Mutually Reinforcing Systems

John Urquhart Ferguson
B.Sc. (Hons)

Submitted in fulfillment of the requirements for the
Degree of Doctor of Philosophy

School of Computing Science
College of Science & Engineering
University of Glasgow

July 4, 2011

Abstract

Human computation can be described as outsourcing part of a computational process to humans. This technique might be used when a problem can be solved better by humans than computers or it may require a level of adaptation that computers are not yet capable of handling. This can be particularly important in changeable settings which require a greater level of adaptation to the surrounding environment. In most cases, human computation has been used to gather data that computers struggle to create. Games with by-products can provide an incentive for people to carry out such tasks by rewarding them with entertainment. These are games which are designed to create a by-product during the course of regular play. However, such games have traditionally been unable to deal with requests for specific data, relying instead on a broad capture of data in the hope that it will cover specific needs. A new method is needed to focus the efforts of human computation and produce specifically requested results. This would make human computation a more valuable and versatile technique.

Mutually reinforcing systems are a new approach to human computation that tries to attain this focus. Ordinary human computation systems tend to work in isolation and do not work directly with each other. Mutually reinforcing systems are an attempt to allow multiple human computation systems to work together so that each can benefit from the other's strengths. For example, a non-game system can request specific data from a game. The game can then tailor its game-play to deliver the required by-products from the players. This is also beneficial to the game because the requests become game content, creating variety in the game-play which helps to prevent players getting bored of the game.

Mobile systems provide a particularly good test of human computation because they allow

users to react to their environment. Real world environments are changeable and require higher levels of adaptation from the users. This means that, in addition to the human computation required by other systems, mobile systems can also take advantage of a user's ability to apply environmental context to the computational task.

This research explores the effects of mutually reinforcing systems on mobile games with by-products. These effects will be explored by building and testing mutually reinforcing systems, including mobile games.

A review of existing literature, human computation systems and games with by-products will set out problems which exist in outsourcing parts of a computational process to humans. Mutually reinforcing systems are presented as one approach of addressing some of these problems. Example systems have been created to demonstrate the successes and failures of this approach and their evolving designs have been documented. The evaluation of these systems will be presented along with a discussion of the outcomes and possible future work. A conclusion will summarize the findings of the work carried out.

This dissertation shows that extending human computation techniques to allow the collection and classification of useful contextual information in mobile environments is possible and can be extended to allow the by-products to match the specific needs of another system.

Contents

List of Figures

Acknowledgments

I would like to extend my deepest and warmest thanks to my supervisor, Matthew Chalmers. This research simply would not exist without his invaluable guidance and advice. He has been a true inspiration and his support and friendship have meant the world to me.

I thank my secondary supervisor, Phil Gray, for his encouragement and guidance. Particularly his advice for the best ways of writing for computing science.

Working within the Equator group at Glasgow was a great education. I was lucky enough to work on some interesting projects and to have talented people help me with mine. I would like to thank everyone who was part of the Equator project for helping to guide my path. Throughout this dissertation, the systems which are presented will be credited with all those who worked on them, but I would like to take the time now to thank everyone for their time and effort in helping me to produce the systems necessary for my research.

In particular, I would like to thank Marek Bell for his early work on *EyeSpy*. Not only did he lay down the ground work for this system, but he spent a great deal of time tutoring me in its design. Marek also helped me greatly by teaching me many of his own programming techniques. He was vitally important in helping me gain the technical skills required for my research and I could not have done it without him.

One of the first iPhone versions of *EyeSpy* was worked on by Owain Brown and Alistair Morrison, with a particularly large effort from Don McMillan. Their help was much appreciated and it contributed heavily in getting an early trial of an iPhone version of the system.

Patient help in programming mobile and web environments was given to me by Malcolm Hall (as well as some electric guitar lessons) for which I am very grateful.

Stuart Reeves had long discussions with me on the best way to trial and evaluate the systems which I am sure prevented me from making rookie mistakes. He also orchestrated the initial validation tests on the by-products for EyeSpy. For this, I thank him.

The participants of the user trials were obviously of great importance to my research and their comments on the systems helped in the designs of future versions. I am indebted to each and every one of them who took part.

While they did not contribute directly, I would like to thank John Nicholson and Joseph Maguire for having theoretical discussions about my research and system design. It led me to some novel ideas and some interesting solutions to weird coding problems. I thank them for always being open to these discussions, no matter what time of night it was. Although he made no significant contributions to my work (quite the opposite in fact), I would like to extend my thanks to Scott Heron. At my lowest points, Scott was always there to cheer me up and take my mind off the bad things of the world. He kept me sane in a time of insanity. I could not ask for a better friend.

I would like to extend a special thanks to my big sister, Dr. Alexandra Laura Ferguson, who led the way for me in postgraduate study. As in so many things, she was the one that inspired me to go for it. Being so willing to share her experiences kept me on a straight path. There are no words that can suitably express my thanks for a lifetime of friendship.

Lastly, I would like to thank my parents, Laura and John. They have made me who I am today and given me all the best things in my life. Their moral, emotional and financial support have made the last few years possible. I could not possibly have achieved any of my successes without their love and care and I will always be grateful for everything they have done for me.

The work described within this dissertation has been funded by EPSRC grant GR/N15986/-01 as well as an EPSRC Doctoral Training Award associated with that grant. Other people who contributed programming time to this research were funded by EPSRC grants EP/E04848X/1 and EP/F035586/1.

Author's Declaration

The contents of this dissertation are the author's personal work. However, some of the systems discussed were designed and implemented with help from both the Equator group and the Social/Ubiquitous/Mobile group at the University of Glasgow.

The author has attempted to make clear when and by whom systems have been designed and implemented with others. However, the author has been one of the main designers and programmers of *EyeSpy* and was the sole designer and programmer of *Realise*. While the system design and implementation of *EyeSpy* was worked on by others, the work and research related to the dissertation itself are entirely the author's own.

Chapter 1
Introduction

Computer games have an important cultural significance and their popularity is easily seen, having produced a $10 billion per year industry [1]. The enjoyment received from gaming has led people all around the globe to spend billions of hours each year in this pursuit [2]. In essence, people enjoy playing computer games and will go out of their way to do so.

If people are going to play games anyway, why not take advantage of this and give games a second purpose? Games could be educational or become tools to improve one's life. They might even benefit other people or organizations.

People are not only playing games on their home computers; they are playing games on mobile devices which provide new ways to interact with our world. Despite such devices being revolutionary, they are also becoming more common in the form of 'smartphones.' Over 45 million smartphones are in use in the United States [3] and over 47% of those devices are used to play games [4]. While this may not be the sole use of such devices, it is a growing market with a 60% boost of games played on smartphones between 2009 and 2010 [4].

In recent years, location technologies, such as GPS, have enabled a range of new mobile applications. Digital cameras can now automatically tag photographs with their location allowing images to be browsed and arranged geographically (e.g., Eye-Fi).[1] A number of photography websites, such as Flickr[2] and Panoramio,[3] provide large collections of

[1] `http://www.eye.fi`

[2] `http://www.flickr.com`

[3] `http://www.panoramio.com`

publicly available, accurately geo-referenced images. The ability of these mobile devices to use their context in the world provides a level of interaction that goes far beyond the home computer.

As with games, these devices need not be restricted to personal tools and entertainment devices. Their contextual capabilities could be harnessed to allow analysis of the world in a way that would be computationally difficult by automated means. These devices would only be useful in this way because of people. Humans can provide the necessary physicality and thoughtfulness to allow a new analytical power which was not previously available.

Human computation [5] is a method of integrating real people as a working component of a computer system. There are many benefits to this: humans can perform a number of tasks better than computers (such as describing an image), humans are better at adapting to changing environments (such as city streets), and it has been demonstrated that 'human computation' games (such as the ESP Game [6]) can be effective in producing by-products as a side effect of play. In this context, a by-product is any useful data that is produced as an incidental outcome of playing the game. That is, the player is not solely aiming to produce useful data, but it happens anyway as a result of play.

However, there are some problems with human computation: the games often produce generalized by-products and are not designed for the benefit of a specific system. Although this approach may maximize the potential usefulness of the games, there will be occasions when specific data is required that may not have been produced.

A second issue is the monotony of these games. Players are being presented with similar tasks again and again and people may feel like they have solved these problems before.

Games with by-products have largely been restricted to the World Wide Web. But we have just discussed the great potential in extending human computation to address the computationally difficult task of collecting and classifying good contextual information in mobile environments. This research aims to explore this potential in order to improve the results of human computation in mobile games with by-products.

1.1 – Research Questions

In order to further the field of human computation, mobile pervasive gaming and games with by-products and to allow such systems to work together to improve their results, this dissertation investigates the following issues:

RQ1. *How can the results of human computation be improved to match the specific needs of other systems?*

RQ2. *How can human computation be extended to collect and classify useful contextual information in mobile environments?*

To begin exploring solutions to the research questions, a review of existing literature will give details of games and how they can be beneficial beyond entertainment. This will largely focus on human computation and games with by-products. The literature review will highlight several problems in existing games with by-products concerning their ability to create desired by-products and in maintaining users' interest in playing the games. This will contribute to ***RQ1*** by identifying some of the issues which need to be addressed. The literature review will also discuss mobile games in order to begin addressing ***RQ2***.

Chapter 3 will use the results from the literature review to develop a new mobile game with by-products. The chapter will briefly cover the testing and evaluation of that game in order to develop an improved design which can match the specific needs of another system. This will work toward answering ***RQ1***. By developing a mobile game, the chapter will also explore issues that will be necessary in answering ***RQ2***.

Chapter 4 will introduce the first mutually reinforcing systems. These systems will include an augmentation of the system that will be described in Chapter 3. The augmentations will begin to address ***RQ1*** as well as further exploring the success of a mobile game with by-products that will contribute to answering ***RQ2***. As with Chapter 3, a brief discussion of the testing and evaluation of the systems will be presented, showing how their problems led to an improved design.

Chapter 5 will describe the addition of a subjective reward system into the concept of mutually reinforcing systems. This will be a further augmentation of the systems described in Chapters 3 and 4. The subjective reward system is a refinement of mutually reinforcing systems which improves the principal aims of the two systems described in Chapters 3, 4 and 5. As such, it will be helping contribute toward answers for ***RQ1*** and ***RQ2***. Once again, the testing and evaluation of the systems will be presented, showing their successes and failures.

Chapter 6 will provide a more detailed description of how the systems in Chapters 3, 4 and 5 were tested and evaluated showing what was found from these results. While some of the testing and evaluation will be discussed in Chapters 3, 4 and 5, this will only cover the findings that could contribute to technical improvements in the systems. Chapter 6 will evaluate the systems on a more conceptual level to show their higher level successes and failings. This will be necessary in discussing how well the systems have addressed ***RQ1*** and ***RQ2***.

Chapter 7 will be a discussion of the findings of Chapter 6 and how well the systems created in Chapters 3 to 5 address the problems of ***RQ1*** and ***RQ2***. Potential future work in mutually reinforcing systems will also be discussed. This discussion will result in a final analysis of mutually reinforcing systems and what they can do.

Finally, Chapter 8 will review the dissertation and the contributions it has made, particularly in respect to how well it has addressed ***RQ1*** and ***RQ2***.

Chapter 2
Background

Using people to increase the capabilities of computer systems is becoming increasingly common. This 'human computation' [5] is often used where the task would be difficult for computers alone. One example is when product selling websites, such as Amazon,[1] compare the usage history of their customers in order to recommend appropriate products. Computers could not make these recommendations themselves without comparing the actions of people because of the subjectivity and complexity of deciding what products would be interesting.

Human computation often uses computers to generate tasks and gather the results. This can work well for broad problems which have limited variations and are trivial for humans to solve. But there can be situations which require complex solutions and greater expertise from the workers. An example can be seen in the question and answer service, Mosio,[2] which allows users to submit problems to a large group of people in order to 'crowdsource' [7] a solution. In crowdsourcing, computers can be used to facilitate communication between worker and beneficiary, as well as helping administer the results. In some cases the problem can be split up into smaller tasks and the Internet can allow people to work together to solve larger problems.

The rise of Internet enabled mobile phones with increasingly sophisticated sensors allows these ideas to be extended to mobile environments where people can work with contextual information as well.

[1] http://www.amazon.com

[2] http://ask.mosio.com

While people can be used to increase the capabilities of computers, computers can also be used to help people improve themselves. Through the use of computer games, we might find it easier to learn a new subject or improve our health. These 'serious games' [8] may not have entertainment as their primary purpose, but they can make an arduous task more enjoyable.

Improving the enjoyability of a task is also a principal aim of games with by-products. These are a form of human computation which encourages people to do work that is phrased as a game. In this sense, games with by-products might be seen as both increasing the capabilities of a computer system and making work tasks more enjoyable.

While there have been many great strides in these areas, they are not free of flaws and there are still ways we could improve the techniques. The research questions of this dissertation focus on finding a means to achieve such improvements. This chapter will explore crowdsourcing, human computation, serious games and games with by-products. This will show the background to the research questions in Section 1.1 and why they have been posed.

2.1 – Crowdsourcing

A potential problem in business is trying to solve a problem for which you have no qualified personnel. The usual method of approaching this problem is to hire someone who can do the job, or to outsource the job to a consultant. While both of these methods would tackle the issue at hand, they each require financial investment. If solving the problem will not recoup the costs of hiring or outsourcing, it may not be worth spending the money. What options, then, are we left with?

One solution was coined by Jeff Howe as 'crowdsourcing' [7]. Howe's proposal was that you could take the problem and open it to a large group of people. Those people could then work on the solution, either individually or by forming groups, and submit answers. The best solution could then be applied to the original problem.

In some cases, a financial reward is given to the creators of the best solution, but people may also compete altruistically or for intellectual satisfaction. No matter what their motivation, a wide pool of people with different talents have the opportunity to apply themselves to the problem. If a financial reward is given, it is only after a solution is found and only to the group which came up with it. You never pay the losers and you do not pay anyone if no solution is found. Furthermore, if you can convince people to participate for non-monetary reasons, it will cost you nothing.

Howe's discussion of crowdsourcing is focused on getting people to directly contribute their time to solve a problem. But there are other problems which people can contribute toward in less direct ways. The earlier example where Amazon suggests products to its visitors is called a recommender system. A similar system, called Recer, was discussed by Chalmers et al. [9]. Recer compares people's usage histories. If your most recent usage is similar to an episode in someone else's usage history, then their next step might be relevant to you as well. Taking this assumption allows Recer to suggest something which might be relevant to you. However, this shows that people are being used to solve a problem without consciously working toward a solution. In the case of Amazon, people simply use the site to browse and purchase products because they want to buy things. However, their history of use is valuable to Amazon because it allows the website to suggest products to other users, resulting in more sales.

Such commercial use of passive crowdsourcing can also be seen in the Internet music site, Last.fm.[3] Users of Last.fm are able to listen to a personalized 'radio' station, saying whether or not they like the music being played. The listeners' likes and dislikes are compared with other users, allowing the site to play music to the user which was enjoyed by people who have similar musical tastes. There is also a feature called 'audio scrobbling' which takes note of the music you listen to in other music software. This information can then be added to your profile to further refine what music you like. The hope is that the system will only recommend music which you will enjoy, even if you have never heard it before. While the users are not actively trying to improve the site for other listeners, this happens passively as they use the site for their own benefit. This is especially true of the scrobbling mode where users simply listen to music as they would normally in their preferred software, without needing to explicitly say that they like what they are listening to.

Last.fm also encourages use of its service to people who have no interest in the 'radio' sta-

[3] `http://www.last.fm`

tions by promoting the scrobbling technology as a means to see statistics of your listening habits over time. There are also live 'widgets' which people can publish on their own websites to show what music they are listening to. By promoting the use of the audio scrobbling software in this way, Last.fm is still getting useful information from people who may not even use their main service.

In this sense, the data being collected on music listeners is a by-product of something which they are doing anyway. People are already listening to music in their own software, and Last.fm offers them a nice statistical overview of their listening habits in return for the data. While this data is clearly useful to the listeners (so they can see who they really choose to listen to the most and who they consistently listen to over time), the data itself would not be worth much on its own. It is perhaps for this reason that the listeners feel that the widgets and statistical information they receive is payment enough. But for Last.fm, having large amounts of information from many users is of great use to their goal of creating personalized music stations.

But how does Last.fm hope to make any money from their service? One way is to charge people who want to promote their own music. Last.fm is in a unique position to target new music to the people who are most likely to enjoy it. By initially trying the music on a broad spectrum of their users, they can determine which profiles are likely to enjoy it using the like/dislike voting system. In addition to promoting the music to the people most likely to enjoy it, they can also sell this information to the artist so that the music can be marketed to the right segment of society.

Another form of crowdsourcing, which is in contrast to Howe's original vision as well as the passive crowdsourcing that was just discussed, is that of the Amazon Mechanical Turk.[4] The Amazon Mechanical Turk is a service which allows users to post open calls for problems to be solved. These problems are known as HITs (Human Intelligence Tasks) because they are supposed to consist of problems that humans find easier to solve than computers. This active crowdsourcing is very similar to Howe's original description, but the difference comes in the organization of the problem solvers. Howe had suggested that people would be willing to compete in order to solve the problem either for the satisfaction of winning or because of a reward bestowed on the creator of the best solution. In this instance, you

[4] `http://www.mturk.com`

simply pick the best solution and pay the one group who produced it but in the Amazon Mechanical Turk, you must pay for every valid solution, and there are no instances where people will work for free.

While many of the benefits are shared between Amazon Mechanical Turk and Howe's original description of crowdsourcing (such as opening the task up to a potentially large range of people and talents), there are some drawbacks to the Turk system. The most obvious being the requirement to spend money on all the solutions which are returned. This means you will have to spend a larger amount of money if you want to see many solutions in order to pick the best one. Alternatively, you can just hire one person at a time until a correct solution is returned, and then pay that successful party. This second option means you may spend less money, but you will not be exposed to multiple correct solutions in order to pick the best.

However, in some cases this may not be such a problem. For example, if the task you need solving can be split into a number of small tasks, you could make each HIT one of these smaller tasks instead of one large task. This means lots of people will still take part, but you will only receive one larger solution (the conglomeration of the HITs). You can then set the payment for each HIT to a much smaller value. This may then cost less than requesting larger HITs from lots of people, but you will still have the benefit of a broad range of people and talents solving your task.

Because of the need to pay all acceptable solutions, the Turk system allows qualifying tasks to be posed before workers are allowed to tackle a HIT. This step is there to try and ensure that acceptable solutions will also be of an acceptably high standard. Workers also gain reputation points that demonstrate the level of satisfied requesters for whom they have carried out HITs. This reputation system is important because if workers want to continue to get HITs assigned to them, they will need to do good work and maintain a good reputation level over time.

Another popular use of active crowdsourcing based on the principle of getting lots of people to individually carry out small tasks is question and answer services. These are services where you submit any question you like and in a short time you will receive an answer. In this case the answer will only have been worked on by one person. Some services (such as 63336[5] and Aardvark[6]) claim to have a system in place which ensures that the person providing the answer is an expert within the field of the question.

A similar service is forthcoming from the social networking site, Facebook.[7] However, rather than send the questions to hired professionals, Facebook relies on its existing users to answer the questions. In particular, it focuses the questions on your existing friends so that you might get answers which are more personalized to you.

A different tack is taken by a company called 'txteagle' which targets the problems to people in developing countries.[8] This is achieved by turning the questions into SMS text messages which are more easy to distribute in developing countries where mobile phones have a higher penetration than the Internet. However, the workers are still required to go through a training and certification procedure. In many cases, this is to the benefit of the workers who are gaining new skills and getting paid to take part and answer questions. The use of mobile phones and SMS text messages imposes the constraint of questions and answers needing to be in plain text and less than 160 characters.

In each of these different services, the background to the service may affect the answer that is given. Hired 'experts' in the same country as the person posing the problem may be able to answer culturally relevant questions which would not be possible for people in other countries. Similarly, your friends in Facebook may give you different answers than strangers because they can tailor the answer to you (perhaps with a more detailed knowledge of what you are really asking).

Commercial uses of active crowdsourcing need not be generalized services like the ones above, especially if a task recurs often enough to justify a service in its own right. One such system focuses on getting the public to search for prior art capable of invalidating a patent or patent application.[9] Large bounties of up to $50,000 are offered for successfully finding such conflicts. The task creators are mainly large companies who want to do their best to ensure that their own patent applications will succeed and that there are no existing

[5] `http://www.aqa.63336.com`

[6] `http://vark.com`

[7] `http://blog.facebook.com/blog.php?post=411795942130`

[8] `http://txteagle.com`

[9] `http://www.articleonepartners.com`

patents that are likely to cause issues.

Not all crowdsourcing is carried out for commercial purposes though. As was already discussed, there are occasions where people will work for the intellectual satisfaction or the enjoyment of winning. But there are also cases where crowdsourcing groups will form in order to benefit the community as a whole.

The earlier examples of question and answer based services have equivalents in the non-commercial realm. Services such as 3form,[10] Mosio[11] and Quora[12] all provide a free Q&A service, but the answers are taken from within the community rather than by hiring dedicated people to answer the questions. This means that the time taken to receive answers may take longer and that answers may intentionally be incorrect or inappropriate. It is also possible that a question will never be dealt with, either because no one can answer it or because they do not want to. There could be various reasons for this, but ultimately the system relies on people giving their time for free to the community, and it may simply be too much effort to answer certain questions. This would be less likely to happen in a commercial service because people would worry about not being paid if they do not answer questions or answer them inappropriately.

Conversely, there is also the possibility that people may ask questions that are inappropriate or abusive. There may even be people who purposely try to waste other users' time by asking questions for which there are no answers. For example, "Who starred in the 1955 remake of *Back to the Future*?" 'Back to the Future' was a film released in 1985 but is about a time traveler who goes to the year 1955. If people are unfamiliar with the film, they may not get the joke and will waste time trying to track down information on a 1955 film that does not exist.

We also cannot ignore that people may simply lie when giving answers. As has been shown in on-line dating profiles, people are more willing to tell lies through computers than they are face to face [10].

Community Q&A services tend to rely on their users to report and deal with such misuse of

[10]http://3form.org

[11]http://ask.mosio.com

[12]http://www.quora.com

the system, perhaps banning the responsible party. This is the approach taken by 3form and Quora. This openness in the system affords the potential for answers to be developed over time by anyone who feels they can improve them. This openness can also lead to potential privacy concerns as there are some questions you may not want anyone to know you are asking.

In the case of Mosio, the questions and answers only get passed among members of your 'question universe.' This means that you can pick and choose who you receive questions from and who is able to see your questions. If people are being abusive in this instance, you can simply remove them from your question universe. This also means you can be more certain of your privacy if you trust the people in your network. The downside of this is that it takes longer to start using the service because you need to build up your question universe. A question universe, being smaller than a completely open call, will probably have a smaller range of people and talents.

Just as with commercial crowdsourcing, there are specific as well as general uses of the model in the non-commercial field. One such example is Open Street Map.[13] This service encourages the public to create street maps of the world that are free to use and under an open license (see Figure 2.1). While the creation of such maps benefits the community as a whole, a certain amount of altruism is necessary for these systems to exist. For instance, people could just use Open Street Map's data without contributing, but for the system to improve, people have to be willing to give back.

There are some mapping projects which are particularly altruistic, focusing on information dissemination during crisis situations.[14][15] These services invite people to collect information about a crisis, then add it to a map which can help visualize the information for the people involved in the crisis who may not have the time or resources to aggregate the information themselves. This can be invaluable information in a crisis, allowing people to avoid disaster areas or help aid organizations get to the places where they are most needed.

Crowdsourcing has also been used by large organizations, relying on people's altruism to take part. The United States Geological Survey has an Earthquake Hazards Program which

[13] http://www.openstreetmap.org

[14] http://crowdmap.com

[15] http://www.ushahidi.com

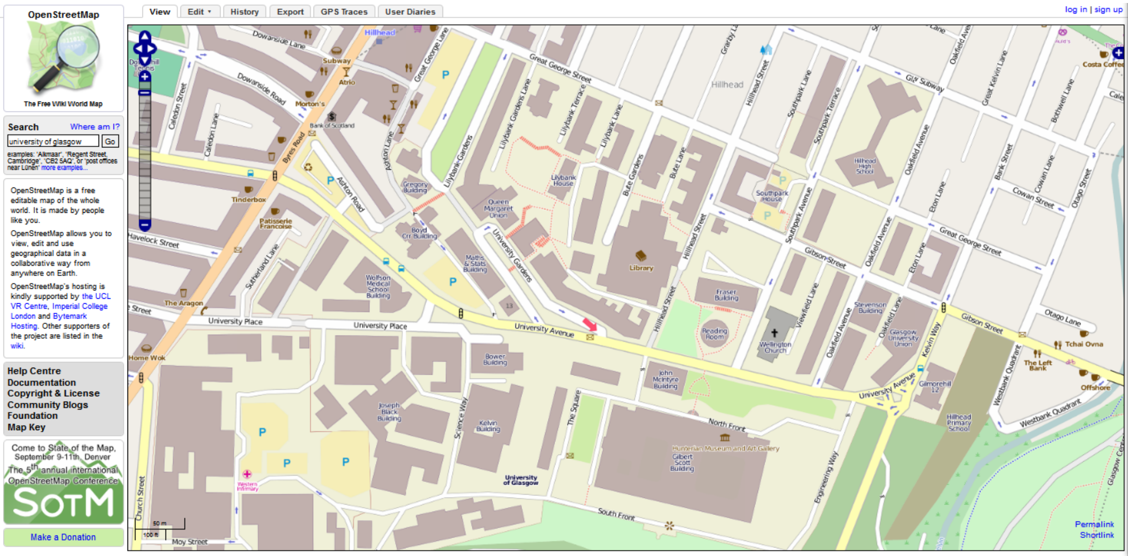

Figure 2.1: The Open Street Map website, showing a map created by the site's users. This map shows the University of Glasgow.

encourages citizens to submit information about earthquakes which may otherwise go unreported.[16] Stardust@Home is a project run by Berkeley University which encourages people to locate interstellar dust particles on pictures of samples collected by the Stardust spacecraft.[17]

The Stardust@Home project is interesting in that it requires a minor certification before people are allowed to take part. Users are trained to identify the required dust particles, then they must pass a test to identify dust in some control pictures. This is understandable as taking the word of unqualified individuals could lead to false negatives and dust particles might never be found. The chance of false negatives being dismissed can also be minimized by having duplication of effort, ensuring that a certain number of people agree that there are no dust particles in a picture.

The issue of training in Stardust@Home is important because it demonstrates the possibility of using non-commercial crowdsourcing techniques when expert knowledge is required. Ultimately the solution is the same as in the commercial realm, as demonstrated with the earlier discussion of Amazon Mechanical Turk: get the responders to prove that they are qualified to deal with the problem.

[16] `http://earthquake.usgs.gov/earthquakes/dyfi`

[17] `http://stardustathome.ssl.berkeley.edu`

The motivations for people to take part in crowdsourcing efforts where there is no 'winner' has not been discussed in much detail. It has already been suggested that people might take part for intellectual satisfaction, but there are other reasons as well.

Some people may choose to take part in crowdsourcing efforts that represent some form of activism because they want to improve their community. There are websites which let you report graffiti, public safety and environmental issues to those who govern their city.[18] Another system allows people to report speed traps where police are monitoring for drivers who are going above the speed limit. This then allows other users to be especially careful of their speed in these areas.[19]

Another potential reason for giving time to crowdsourcing efforts is that people want to contribute to cultural programs, such as writing reviews for movies[20] or producing their own Internet 'radio' stations.[21]

Another important point is that people might give up their time just to know that they took part. One example of this is a band who wanted to feature their fans in one of their music videos. They filmed themselves performing various poses, then took each frame and asked their fans to copy the pose and take a picture. The band could then swap the original frame for one with the fan copying the pose. Many people took part because they were fans, or simply because they wanted to take part in a music video.[22] Figure 2.2 shows a frame that has been submitted by one of the website users.

Perhaps one of the most successful examples of crowdsourcing is the encyclopedia website, Wikipedia.[23] Wikipedia hosts a collection of articles on a wide range of topics which are written, edited and improved by the general public. No one is paid to write any of the articles, and there is no incentive given for contributing other than to improve the resource for others. Despite this, a study has shown that Wikipedia has comparative accuracy to more established, professionally produced encyclopedias like Britannica [11].

[18] `http://www.citysourced.com`
[19] `http://trapster.com`
[20] `http://fflick.com`
[21] `http://songza.com`
[22] `http://oneframeoffame.com/19726`
[23] `http://www.wikipedia.org`

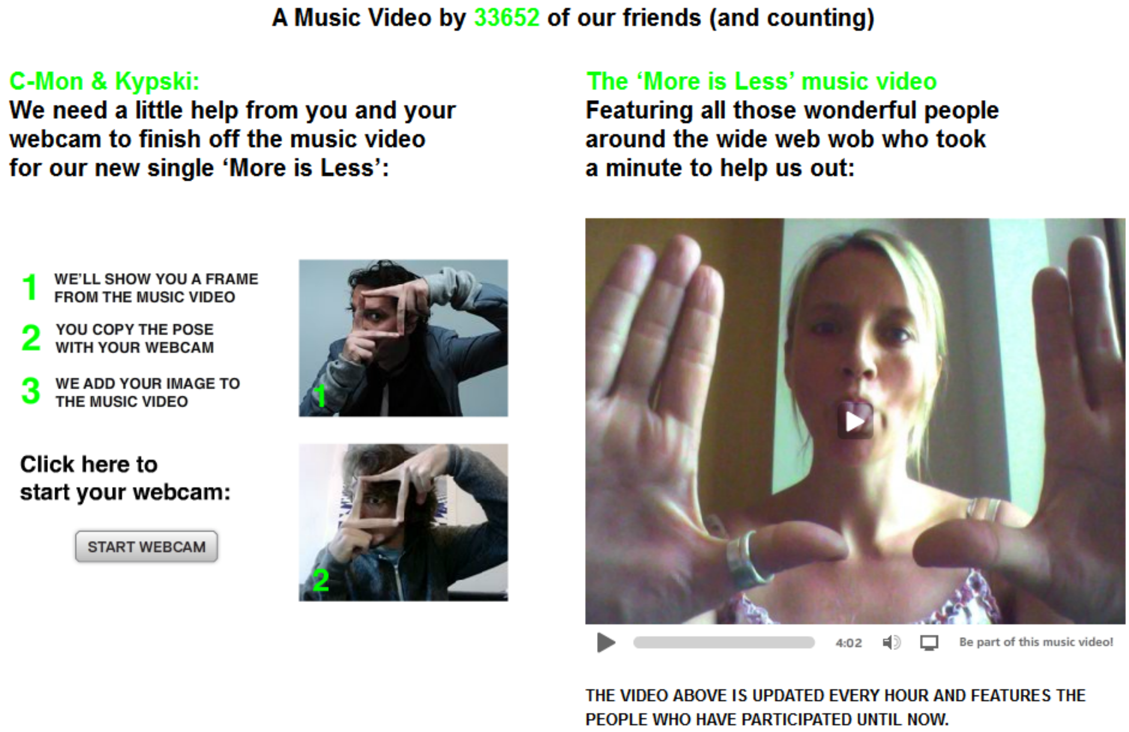

Figure 2.2: The 'One Frame of Fame' website showing a user submitted photograph to match one of the frames from the band's music video.

The motivations of the people who work on Wikipedia may vary greatly, but a study has shown that they generally open themselves up more to their friends than people who do not contribute [12]. While people's openness may play a part in the creation of an open access resource of knowledge, we cannot say that this is the only reason, or even the main reason why people are willing to give their time and energy to this task. It is just as possible that someone with a vested interest in an article will edit it to suit their own needs.

There is, as with other forms of crowdsourcing that have already been discussed, the possibility of vandalism due to the relative lack of accountability in such open systems where anyone can edit a page, even anonymously. It is also pointed out by Denning et al. [13] that with an ever changing live site which anyone can edit, you might try to cite a page that is later changed so that it no longer backs up the point you were making. You could choose to cite a specific version of the article, but then you may lose out on future improvements that are made. The lack of a formal peer review process has led Wikipedia contributers to organize themselves into an editorial body, but this still does not compare with a full time staff of credentialed employees whose work can be trusted. All the same, it should be noted that most vandalism on Wikipedia is fixed so quickly that most users never see it [14].

Although there are many potential weaknesses, crowdsourcing has been shown to be successful in many systems as the basis for large groups of people working together and accomplishing goals which would be difficult or expensive to achieve by other means. However, there are some fundamental problems with crowdsourcing and these will be discussed in the next section.

2.1.1 – Problems in Crowdsourcing

While it has been demonstrated that there are many potential uses for crowdsourcing, both in the commercial and non-commercial realm, there are fundamental issues that stop it being suitable in all situations.

One issue is that of privacy. There may be situations where a company cannot use open calls because it would release company secrets to their competitors. The work itself may also invade people's privacy. One such example was a proposed system to reward the public for monitoring Closed Circuit Television (CCTV) footage through the Internet.[24] The system launch was delayed because the concept of paying the public to spy on each other was found to be distasteful and privacy rights organizations felt it was too invasive to allow non-trained people to view CCTV and that it may lead to the system being used for entertainment.[25]

Another potential problem with crowdsourcing is that it can be difficult to automate if you have a large number of similar problems to be solved. However, recent toolkits such as TurkIt [15] and Crowdflower have made this process much easier.[26] But even with this ease in automation, there is still the problem of automating quality assurance. Crowdflower attempts to deal with this by adding an additional layer of crowdsourcing where people validate the work of others. Another method is just to employ more workers for each task so that there is redundancy. This means that the most reported solutions are more likely to be correct because different people have independently proposed the same answer.

Silberman et al. [16] also point out that there are a lot of workers on Amazon Mechanical Turk who do the minimum effort for the maximum amount of money, which often means

[24] `http://interneteyes.co.uk`
[25] `http://news.bbc.co.uk/2/hi/uk_news/8485056.stm`
[26] `http://crowdflower.com`

not doing their best work, but simply submitting answers that meet the minimum requirements to get paid. Depending on the task, this may even allow them to submit guesses which are hard to detect. In an automated system, these people will probably still get paid. It may be easier for people to do more of these tasks in the same amount of time (because less effort is required to fake an answer than actually work it out) and make more money because it is difficult to pick them out of the crowd. A solution was presented by Ipeirotis et al. who used redundancy to rate the success of workers and then favored employment of workers who produce better results [17].

The issue of people doing the minimum amount of work in order to get paid is also pointed out by Mason and Watts [18] who show that financial incentives seem to increase the quantity of work produced, but not necessarily the quality. This seems to be particularly true when the difficulty of a task increases: people would rather do a larger number of easy tasks than a smaller number of difficult tasks if the amount of money is the same at the end.

Even when people are not trying to abuse the system, there is still the possibility of errors. Recommender systems, for example, try to predict what information will be useful or relevant to someone in the future based on their previous actions and the actions of others. However, as has been discussed by Albers [19], the past usage of a system cannot predict its future use. It is simply impossible to guarantee that recommender systems will correctly predict useful or relevant information.

'Utility' itself is not always objective. Crowdsourcing workers may feel they are providing useful information that others would disagree with. For example, a system was created which allowed people to share music in an urban area to allow an aesthetic account of the area to be developed by the collective community [20]. However, the study showed that there was great variety in people's aesthetic experience of the space. Context affects interpretation of information and this means that the same information may mean different things to different people at different times and in different places [21].

The problem of getting people to agree on labeling something can also be seen in 'folksonomies' where people try to index content on the Internet [22]. These systems often suffer from 'noise' which dilutes the labels by introducing too many that are not widely agreed on.

Many of the problems in crowdsourcing seem to stem from a lack of structure to herd the people and their answers. Crowdsourcing relies on people doing the work themselves, as well as self-governing so that potential abuses do not affect the outcomes. In the next section, the concept of human computation will be discussed as a potential means to guarantee that these outcomes will take place.

2.2 – Human Computation

While crowdsourcing tends to gather workers through open calls which are created by humans, human computation tends to automate this process as part of a computational process. While the motivation for crowdsourcing is often to save money because it is a cheaper method of getting lots of people to do work for you, human computation tends to focus solely on things which computers find difficult to do. This might include adapting to changing environments, providing subjective opinions or taking advantage of human sensory capabilities and reasoning.

Before giving an example of human computation, another crowdsourcing system will be discussed so that its human computation equivalent can be fully understood. The system is called Distributed Proofreaders[27] and is related to Project Gutenberg.[28] Project Gutenberg is an attempt to digitize all the public domain works of fiction that are available and a lot of this work is done by scanning physical copies of the books and then using optical character recognition (OCR) software to turn them into digital text.

However, due to the large scale of the project, it is difficult to find all the OCR errors that occur. Distributed Proofreaders was an attempt to crowdsource this problem so that people could find and correct the errors. Users of the Distributed Proofreaders website can see a tutorial on how to proofread and correct errors, and are then given a random page from a book that is suspected of OCR errors and asked to try and correct them.

[27] `http://www.pgdp.net`

[28] `http://www.gutenberg.org`

Figure 2.3: The input system for reCAPTCHA.

This process has been turned into a human computation system called reCAPTCHA.[29] A CAPTCHA (Completely Automated Public Turing-test to Tell Computers and Humans Apart) [23] usually involves asking users to type in the letters that appears within a warped image. Due to the warped effect on the text, computers should find it difficult to automatically determine what the letters are, but it should be possible for humans. CAPTCHAs are often used in preventing automated form fillers on websites, particularly when people use forms to register for new accounts. Von Ahn et al. proposed that you could use CAPTCHAs to solve the same problem as Distributed Proofreaders by taking words for which there were OCR errors and then turning those words into CAPTCHAs [24]. This makes sense because CAPTCHAs should be problems which humans can solve, but computers cannot. Since there have already been OCR errors, we know that a computer has already failed to solve the problem.

Part of the problem with this is that we will not know if humans are solving the problem because we do not know the correct answer yet. ReCAPTCHA solves this problem by asking the users to read two different words at the same time (see Figure 2.3). One word has already had enough users agree on an answer that it can be assumed to have a known solution. The other word is then unknown, but users only have to get the known word correct in order to pass the test. However, there is no way for the user to know which word already has a known solution. Once enough users have agreed on answers for the other word, it too can be considered to have a known solution.

[29] `http://www.google.com/recaptcha`

A similar system was proposed by Faymonville et al. which designed CAPTCHAs to show images of real things which included words (such as the brand name on a packet of crisps [25]). As with reCAPTCHA, these images had already been flagged for requiring human involvement to accurately achieve the labeling of the image. The system used to flag them is called the soylent grid and was set up to enable the labeling of images by people [26]. The relationship between the image labeling CAPTCHA and the soylent grid is similar to the relationship between reCAPTCHA and Distributed Proofreaders. Another similar system (though based on describing general images rather than ones with text in them) was used by Morrison et al. [27].

Not all human computation tasks are carried out by isolated users. In the Monolingual system, two users are paired together, each of whom cannot speak the other's language [28]. A machine translation of a text from one user's language into the other is then created. The first user attempts to highlight parts of the translation which look incorrect. Their annotated version is then machine translated back into the source language and presented to the other user. This user then attempts to correct the meaning of the sentence by introducing redundant language to make it clearer. This back and forth between the two users continues until they both feel that the meaning of the sentence has been preserved well enough in both languages. In this way, two users who cannot speak each others' language can work together to form a more accurate translation of a passage of text.

A similar process of iteration has been used with Amazon Mechanical Turk in getting users to describe images [29]. Workers would build upon the description of an image given by previous workers instead of starting from scratch each time. In some cases this would involve changing or correcting earlier descriptions. This is in contrast to the approach of having many users independently construct descriptions and then seeing how many independently agreed. In this instance, the descriptions appear to be better in an iterative approach to labeling. However, not all tasks will lend themselves to such a system, such as problems which have yes or no answers.

It has been suggested that human computation is comparable to an algorithm because they both have inputs and outputs, they both have levels of accuracy and their efficiency can be measured by looking at average output over time [30]. However, unlike algorithms, you cannot guarantee that the output in human computation will be what you want or that there will be any output at all within a given time. In human computation, we need to think of the systems as being part of interaction machines because the user input will change how programs run each time [31]. For example, humans might add new instructions to a program during runtime which will change the way the program operates.

It has also been suggested that human computation can resemble a distributed process if you use lots of humans together to solve small parts of a larger problem [2]. This would mean that the more humans you get to take part, the larger the problem which can be addressed with human computation.

This section has described some methods of incorporating human computation. However, there are potential problems with human computation which will be discussed in the next section.

2.2.1 – Problems in Human Computation

Human computation can be seen as the means to incorporate crowdsourcing more directly as part of a computational process. Because of this, human computation shares some of the same problems as crowdsourcing. The privacy issues, for example, still apply. However, human computation also addresses some of the problems of crowdsourcing, such as making it easier to automate the process.

Human computation, while addressing some of crowdsourcing's issues, also introduces some of its own. One problem is that human computation efforts, such as reCAPTCHA, rely on many humans agreeing on a solution before it is accepted as accurate. While this may generally be true, there are occasions when the majority may be in the wrong [32] [33]. For example, there are many areas where you may need to be an expert to answer a question. In this instance, the one expert's opinion should have more weight than a large number of people who are merely guessing about the answer.

In general, human computation focuses on small tasks which are easy to carry out. However, these sorts of problems may be boring and repetitive to solve. Both human computation and crowdsourcing can be set up to pay people to do the work, but this may become a requirement if problems are not enjoyable enough for people to find solutions for free.

Huang et al. show that the solution quality for a human computation task can be affected when the final goal is achieved in different ways [34]. This demonstrates that the phrasing of human computation tasks is important in getting people to work as hard as possible.

Another potential problem can come from the demographics of the people carrying out the work. Cultural differences between two demographic groups could potentially change the solutions given to tasks. This problem is highlighted by Ross et al. who point out that most work on Amazon's Mechanical Turk is produced by young, well educated Indian workers [35]. This could lead to cultural disparities in the work. There may be cases where this is an advantage, especially if you are seeking information about a particular demographic, but if you are not aware of these issues, you may be surprised by the solutions your workers produce. It is best to target the work toward workers whom you think will produce the solutions that are of most use to your goal. This highlights that the usefulness of data depends on who is using it. The same data may be useful to multiple groups in different ways, and there may be groups who will not find the information useful at all.

Paying people money to work on problems can be expensive for large or ongoing problems, so how do we encourage people to work for free?

One possibility is to encourage altruism, but this depends largely on the type of work people are being asked to do. While it has been shown that people are willing to give their time toward some crowdsourcing tasks for free, they were usually tasks where no one made money and the outcome was for the benefit of some community, or mankind in general.

Another possibility is to phrase the problems in a more entertaining way so that people will enjoy their work. The next sections will explore this possibility.

2.3 – Games

As has already been discussed in Chapter 1, there are a lot of people playing computer games. The time people spend playing games could be harnessed for more purposes than just entertainment. Additionally, games can be a good way to help us learn things [36] and can also be a good vehicle for conducting research, as was stated by Bell et al. [37],

> *"Games have wide social and financial impact, and form an interesting application area in themselves, but we chose a game because one can design a game to explore specific technical issues raised by wider research, and adapt it with ongoing findings relatively easily."*

Human computation, discussed in the previous section, can also be applied to games. In some games, like 'Sleep Is Death,' human computation is the method by which the game content is generated.[30] In the game, one person is the player and another person generates the game content in a turn based scenario. This creates the illusion that the game can react to any input from the player, in contrast to most computer games which rely on pre-programmed game content which has a finite limit of possibilities.

This system of players reacting to each other's play is common in many games, such as chess. But often, the rules put limits on what the players can do. For example, Takhtamysheva et al. describe a game where one player must create descriptions of animations and then the other player must try to match the descriptions to the correct animations [38]. While the descriptions themselves may be quite varied and it can be argued that one player is generating the game content, the limited number of animations in the system limit the game's possibilities.

There are further games which use human computation in quite direct ways, making it clear to the players that they are trying to get them to do some work, but hope that they will find it fun. For example, NASA created a game called 'Be A Martian' which allows users to play games while also sorting through and organizing images of Mars.[31] There are similar games to classify and organize celestial information as part of the 'Zooniverse' project.[32]

A puzzle game called 'Fold-it' involves the players interacting with protein structures to try and optimize the computed energy [39]. Since players are told they are interacting with protein structures and that the purpose of the game is to locate the native conformation of proteins, it is a very thinly veiled task of human computation. However, since it is presented as a puzzle game, people still find it enjoyable.

[30] `http://www.sleepisdeath.net`

[31] `http://beamartian.jpl.nasa.gov`

[32] `http://www.zooniverse.org`

In addition to entertainment, games can be used for promotion. This has been adopted by the American military who commissioned the game 'America's Army' as a way to provide insights into the way that the military carry out their missions and what they require of their soldiers.[33] A similar system has been developed by NASA called 'Moonbase Alpha' which was intended to inspire people to take interest in NASA and space exploration.[34]

These examples demonstrate some of the vast potential of games and how they can be used to achieve various goals. However, the next sections will focus on specific uses of games that relate to this dissertation: serious games and games with by-products.

2.3.1 – Serious Games

While many games are created for the purposes of entertainment, there are other areas in which games can be used. One such area is that of 'Serious Games.' Serious games are an established concept but there is some ambiguity over an exact definition [40]. One possible description of serious games is given by Susi et al. [40],

> *"Serious games usually refer to games used for training, advertising, simulation, or education that are designed to run on personal computers or video game consoles."*

Another definition is given by Michael [8],

> *"... games that do not have entertainment, enjoyment, or fun as their primary purpose."*

Both of these definitions address issues that are important to serious games. Susi et al. show that some useful outcome is a necessity for a serious game. Michael shows that entertainment is not the primary reason to create a serious game. A definition that incorporates both of these ideas might be a game which is not designed with entertainment as its primary purpose, but which does have a useful outcome beyond that of having fun. However,

[33] http://www.americasarmy.com

[34] http://www.moonbasealphagame.com

perception of these games is not limited to design values. A player's perception of the primary purpose of such a game may differ from the game designer's primary purpose for it. A player may indeed see a game as fun and play it for this reason. However, the game designer might not have made this the principal goal of design. This will also vary from person to person.

According to Susi et al., these outcomes can be beneficial either to the person playing, or some other party or business [40]. In order to better demonstrate the possibilities of these outcomes, some examples will be discussed.

One serious game, called 'REXplorer' [41], is comprised of a mobile device used for tourism within a historic village. Tourists are able to rent a mobile device that can track their position using the Global Positioning System (GPS). It is intended to make tour guides less boring by having the users 'cast spells' in specific places in order to interact with 'ghosts' that appear to them. The tourists are not explicitly made aware that they are carrying a GPS device. They are told that it is a device for detecting paranormal activity. Thus, when the GPS unit detects that the device is in a place where information is available, the user is shown a high level of 'paranormal activity.' The device also contains a mobile phone that has a camera. The phone sends data back to a server which creates a log of what the users did. This can then be made into an interactive journal that can be accessed by players after their visits. This concept of post-visiting has been described by Brown and Chalmers as an important aspect of tourism [42],

> *"Post-visiting is [...] a powerful way of extending the enjoyment of a tourist visit out beyond the visit itself."*

The game is linked to real artifacts which the users are given knowledge of in order to play the game. A genuine gravestone with symbols that have never been deciphered are used as the basis of the spell casting. The players must use their device to 'draw' the symbols from the gravestone in mid air in order to cast spells. If this is done correctly in a suitable location, a ghost will appear and give the user some historical information. The information is structured so that it will leave dramatic endings to each monologue to entice users to play further. Camera based motion tracking is used to determine whether a spell has been cast successfully.

The makers state that using a device that is rented is a great benefit because it reduces the normal "lowest common denominator constraint" [41] of many pervasive applications. Also, pretending the device is for paranormal detection would be less feasible if people could use their own devices.

The outcome of this game is that it makes the idea of a historical tour more interesting. Thus, it is hoped that the user will learn more because they will find it more entertaining. The effect of having the journal that one can refer to after the event also means that the users will be more likely to retain the knowledge and perhaps spread it to others.

Another system called 'Fish'n'Steps' was created by Lin et al. [43]. Fish'n'Steps makes use of a step counter to determine the health of a virtual fish. The more the user walks, the healthier his or her particular fish will be. The point of this system is that the users should be able to better identify the effects of their lifestyle by seeing the affected health of their fish. It is then hoped that, in an attempt to keep the fish healthy, the users will become more active. Having changed their behavior in this way, the users would retain these new characteristics after they stop playing the game. A competition element between users is also established by a points system in order to further motivate players.

The system's primary focus is to help people lose weight. The authors show that one of the problems of relying on lifestyle change to prevent weight gain is that people seem to find it difficult to permanently change their lifestyle in this way. There is also the problem that improvements are only seen gradually in a person's weight or physique and this is often discouraging. Fish'n'Steps is an attempt to combat this by taking advantage of the Transtheoretical Model (TTM). This is described by Lin et al. [43],

> *"TTM argues that individuals change their behavior gradually, by advancing along a series of steps. These steps vary from pre-contemplation in which individuals have not realized the need for change, to termination in which the new behavior has become so habitual that there is no longer any danger of relapse."*

Thus, by using the Fish'n'Steps game to encourage exercise until the player has reached the final TTM step, the chance of a permanent, healthy lifestyle change is improved. This may be due in part to the game fitting well into people's daily routines.

The Fish'n'Steps game provides an incentive to do something about one's health. While the game's focus is not that of entertainment, it tries to have something compelling about it that keeps users playing. However, it is admitted by the author that the users lost a lot of their enthusiasm for the game after the first couple of weeks [43]. Fish'n'Steps does show that serious games can give users incentives to do things that they would not find enjoyable. However, this is not the same as getting entertainment. In this instance, the user will benefit physically, so one might suggest that this is the ultimate pay off for playing this serious game.

While Fish'n'Steps took the approach of being a passive game that would fit well into a daily routine, there are other games which require the player to participate more actively. One such game is 'Dance Dance Revolution.'

Dance Dance Revolution (DDR) is a game in which players dance to try and match a given sequence of moves, supplied by the computer, which suitably follow a song that is being played. Hoysniemi successfully convinced 556 people from different age groups and countries to fill in a questionnaire about their experiences of playing Dance Dance Revolution [44]. The results of the questionnaire are summed up as follows,

> *"The results show that playing DDR has a positive effect on the social life and physical health of players, as it improves endurance, muscle strength and sense of rhythm, and creates a setting where new friends can be found."*

It is perhaps important to note that there is some potential bias from the people who answered the questionnaires as they were people who regularly played the game. Therefore, it is likely that they are mostly people who enjoy it. To get a true view of DDR, it would be necessary to get answers from people who did not regularly play the game or who did not enjoy it. This would help show how likely DDR was to create a healthy lifestyle amongst all people.

DDR might be considered a serious game in that it has the obvious side benefit of improving people's general fitness. However, it could be argued that this is the case because it provides more than a simple motivation: the people who want to play it find it genuinely entertaining. It is then possible that they are not playing it because it is a means to help them become fitter, but because they enjoy it. This is shown by Hoysniemi [44],

> *"The health benefits associated with DDR did not play a significant part in the decision to begin playing the game. Only 1.5% of respondents reported health as a reason to play the game, and even fewer because it helped them to lose weight (0.9% of the respondents)."*

However, Hoysniemi also showed that after playing regularly, 55.2% of players mention health benefits as one of the reasons to continue. This is compared to 65.6% saying that entertainment was a major factor in continuing to play. Therefore, one might suggest that DDR is not a serious game as entertainment does seem to be the primary reason to play. However, it does fulfill some of the criteria in that it provides benefits to the players by improving their fitness level. The fact that regular players consider the fitness improvement to be one of the reasons to play, although not the primary reason, also shows a close connection to serious games. This perhaps shows the possibility for serious games that are also genuinely entertaining and enjoyable rather than only being motivational incentives.

It is also worth noting that mobile games might fit better into people's daily lives than an arcade style game like DDR. This is an important issue for serious games that are trying to improve a player's lifestyle choices. As shown by Lin et al. [43], the Transtheoretical Model involves a player learning a behavior through habitual reinforcement. Fish'n'Steps achieved this by supplying an incentive for players to change their lifestyle such that, when the incentive was taken away, the player should maintain the learned behavior. However, with Dance Dance Revolution, the game was not intended to change a player's behavior in their daily life and the game was not being played all the time. This does not mean that DDR will not produce a lifestyle change, but it implies that the change would come from specific periods of exercise rather than a more generally active lifestyle.

This comparison between Fish'n'Steps and Dance Dance Revolution might suggest that Fish'n'Steps would be played more than DDR as it fits better into a player's daily routine. However, Lin et al. showed that many players of Fish'n'Steps lost interest in the game after only a few weeks of playing [43]. This is not enough time to go through the habitual learning that is described by the Transtheoretical Model. Thus, DDR might actually have a longer lasting effect than Fish'n'Steps because people still choose to play it after a longer period than those playing Fish'n'Steps. The benefits of DDR as an exercise tool are shown by its acceptance in school exercise programs, as shown by Sall and Grinter [45],

> *"… the state of West Virginia adopted DDR in all of its schools with the purpose of encouraging children to exercise and lose weight."*

This illustrates a major problem with serious games. When there are games that are genuinely entertaining, why would a person choose one that is not? In the case of REXplorer, it is likely due to people wanting to learn, but wanting it to be more interesting. It is also important to note that REXplorer was only played for very short periods of time in the special context of giving visitors a richer tourist experience. In the case of Fish'n'Steps, it is likely due to people wanting to improve their fitness, but struggling to find a strong incentive to do so. This issue is discussed by Susi et al. [40],

> *"Hardcore gamers generally want the richest possible experience from their games. For serious games, however, it is more important that the model or simulation can be used to solve a problem, than providing 'rich experiences' of the kind sought by hardcore gamers."*

It might then be the case that people will play a game that is less entertaining in order to help them solve some problem of their own. If solving the problem will benefit them in some way but is rather dull or tedious to carry out, serious games might be an ideal solution. The trade off for not choosing to play a more entertaining game is that a serious game will directly benefit the player. However, the benefits for playing serious games are uncertain, as shown by Susi et al. [40],

> *"… there is no conclusive answer to the question of evidence for the acclaimed benefits and potential consequences of games and game play."*

Serious games can do more than benefit a player's health. EpicWin is a game for mobile phones that turns a to-do list into a game.[35] The to-do list allows you to add things that

[35] `http://www.epicwinapp.com`

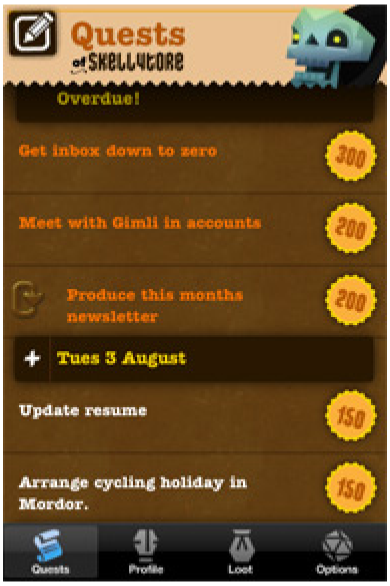

Figure 2.4: The interface for the EpicWin game, showing the tasks that need to be completed and their associated difficulty ratings.

you need to do in your daily life and cross them off when you complete them. However, in EpicWin, you will also assign difficulty ratings to each task. Then, when you cross them off after you have done them, you will see your avatar beat up the task and grow stronger. The interface for the task list can be seen in Figure 2.4. The more tasks you successfully complete, the stronger your avatar will become. There is also a map which your avatar can travel in, and the stronger it becomes, the more areas it can access. Many areas also feature new items which your avatar can find and use to become stronger.

The game requires the user to make honest judgments about the difficulty of the tasks they add to the system. It would be very easy for users to assign high difficulty ratings to easy tasks so that their avatar could gain strength more quickly. However, since the users do not compete against anyone and there is no way to 'win' the game, this would cheat the purpose of the avatar: a way to visually represent how much you are achieving in your daily life. However, players might prefer to see more areas of the map (or more successful animations of their avatar beating up tasks) than have an avatar which represents their actual achievements. However, since the purpose of the game is to make daily tasks more fun, cheating in this fashion would not detract from the game's purpose. As such, the ease of being able to cheat might not necessarily be seen as a design flaw.

Serious games are commonly used as educational tools. There have been serious games ranging from teaching parenthood to disaster preparation. The former of these is a commercial game.[36] The latter is funded by the Federal Emergency Management Agency (FEMA) of the United States of America.[37] This shows that these games are entering the commercial world and are being backed by government agencies. However, a problem that must be overcome when using games as an educational tool is finding a good balance between fun and learning. This is highlighted by Van Eck [1],

> *"A balance between the needs of the curriculum and the structure of the game must be achieved to avoid either compromising the learning outcomes or forcing a game to work in a way for which it is not suited."*

The ideal solution would be to create a game which taught people something, but had enough entertainment to be played on this merit alone. However, creating educational titles of comparable quality to successful commercial games requires the same investment of time and money. It may even be more time and money because of the additional effort involved in merging the curriculum with the entertainment.

Examples have now been given of games which benefit the player, but what about games that benefit a third party? One such serious game is called the 'YouLicense Rating Game.'[38] The YouLicense website is a place where musicians can post their music such that other people can browse for music that they wish to license for use in their own projects. However, the descriptions of the music on the website are given by the musicians themselves. This affects how well a search will work on the website for a licenser looking for music for a specific purpose. Rather than the system administrators going through all the music on the website and judging whether or not it has been described accurately, they constructed a serious game to encourage the site's users to do it for them. They describe the game at their website,

> *"The Rating Game will help YouLicense users identify which songs are best and should be shared with the YouLicense community. There are three categories for rating a*

[36] `http://www.babysittingmama.com`
[37] `http://www.disasterhero.com`
[38] `http://www.youlicense.com/RatingGame`

> *song: overall rating, how well the song is described and the quality of the recording. As an added bonus, at the beginning of each month we will feature on our front page the YouLicense user who has rated the most songs truthfully."*

As shown from this quote, the only real reason given to users to play the game is that if they do it more than anyone else, their artist page will be featured on the website. This would make it more likely that their own music would be heard and licensed. While this reward is one that the users might seek, the likelihood of achieving it is small. Players are not told how much music they will need to review in order to win or how many people they are competing against. Since it is only the top player that can win, most players will not gain the benefit of being shown on the front page. The only people that always win are the system administrators who ensure that the descriptions of the music on the site are as accurate as possible, thus providing more accurate search data.

It should be noted that these are short term goals. Users of YouLicense might recognise that the more successful the site becomes, the more likely is it that their own music will be licensed from it. Accurate search data will allow a licenser to find more relevant music. The better such a search works, the more likely it is that licensers will use the site. Therefore, it would be prudent for users to play the game for this benefit as well as the possibility of being featured on the front page. However, this may not be a trade off that the site users feel is fair.

The problem with the YouLicense rating game is not that is has been designed with a purpose other than entertainment. It is that it has been designed to benefit someone other than the player. This situation might still be viable if the player got something out of the game as well, but that does not seem to be the case. In this instance, one must ask why anyone would choose to play such a game. The game would work much better if the pay off for the user was higher. Despite its shortcomings, the game does demonstrate the idea of using games to entice players to carry out work for the benefit of the game designer, rather than the player. This is an idea that can be described as 'Games with Human Computed By-products.' However, for these games to work, the user must still receive some sort of benefit from playing. It is for this reason that the YouLicense Rating Game will probably be unsuccessful.

Figure 2.5: A screen shot from the America's Army game showing the point of view of the player's character.

One game which perhaps encompasses all these ideas is America's Army.[39] Originally created to show civilians what it was like to be a member of the United States army, this game has become very popular. A screen shot from the game can be seen in Figure 2.5. The game's website describes it as,

> *"… one of the most popular computer games in the world."*

The game is distributed for free from the website and is a well developed 3D simulation of military battle and training. Due to its zero cost for players and its enjoyable nature, players choose to play this game for the purposes of entertainment at no cost to themselves. However, the game serves two other purposes. Firstly, it is showing civilians what the military is like. This could be called marketing on the part of the Department of the Army. Furthermore, the game is used internally as a training system. This is shown by a quote from the website,

> *"As the game's popularity continued to grow with each of its dozens of new version*

[39] http://www.americasarmy.com

> *releases, the Army has expanded its brand through a variety of products including […] training applications for use within the military and government sectors …"*

This shows that it is a serious game. However, it has also proven to be very entertaining. Since players can download and play the game for free, it could be argued that they are also getting a certain amount of free military training. Thus, the game benefits the Department of the Army by providing a marketing and training tool and also benefits players by providing entertainment and mild training. This fulfills the criteria for a serious game in that it benefits the player, but is also entertaining and benefits the game designers. This shows the great potential of games for use beyond the entertainment industry.

2.3.1.1 – Problems in Serious Games

As has been shown, serious games can be used to add incentives to tasks that are unattractive to users. In general, entertainment is not their primary purpose. For example, REXplorer [41] makes a learning experience more interesting. Similarly, Fish'n'Steps [43] provides an incentive for a lifestyle change. This shows that users are willing to play a less entertaining game if they can see that it will be beneficial in some way. These benefits may be for the player, or the player may have more altruistic motives. Hoysniemi showed that the benefits of serious games might be better achieved if the player finds the games genuinely entertaining [44]. This way, it is more likely that the games will be played.

The YouLicense rating game showed that games can be used to benefit people other than the player.[40] This also demonstrated that, if the player does not benefit from the game's design goal, a player should always gain something from playing the game, even if it is just entertainment. Otherwise, such a game would not be likely to succeed. This is a significant problem with serious games that are designed to benefit a third party, rather than the player. While users may be willing to play a less entertaining game because they know it will also provide them an additional benefit, it is less likely that they will do this if it benefits someone else instead. While people might be altruistic in this regard, people will still be more likely to play games that benefit a third party if the games themselves are genuinely entertaining.

The problem of creating serious games is that users require some benefit. If the game is

[40] `http://www.youlicense.com/RatingGame`

not entertaining, the user is not likely to play unless there is some other benefit for them. Although it is possible that users will play a game for altruistic reasons, even if they do not enjoy it, this is not likely to be as successful as a game that was genuinely entertaining. Conversely, if a game is entertaining enough, a user may play it for that reason alone, even if it does have an ulterior purpose.

This section has shown that people are more willing to do certain things if they are phrased as games. However, this does not mean that people will play serious games instead of others which are more fun. Serious games might only be called fun when compared against the task they are masking. However, people will still play them as a more enjoyable way to achieve certain tasks. When this benefits the player directly, the reason for playing is obvious: while this may not be the most entertaining use of their time, people will play the game because it is profitable to them in some other way. Also, it is likely to be a more pleasurable way to achieve these benefits than without a game. When the game benefits a third party, but is still not as enjoyable as other games, the reasons for someone to play become less clear.

It is possible that someone will play altruistically, knowing that they are helping the third party in some way. Another possibility is that the game itself is free to play, making it a cheaper option than paying for a game that is more entertaining. Whatever the reason, people will be more likely to play a game if it is enjoyable and will be more likely to return and play it again. If we construct a game that benefits a third party, it is therefore more important that the game be fun than if it benefited the players directly.

2.3.2 – Games With By-products

One way in which playing a game can be beneficial to a third party is by having the games create a by-product during the course of play. This means that playing the game will produce some sort of data which is valuable to a third party. This is a specific type of serious game that is constructed both to be enjoyable enough to play on its own merits, and which also produces by-products that are of use to a third party. The reason these games need to be enjoyable is because more by-products will be produced each time the game is played. The more enjoyable the game is, the more likely people are to play it multiple times and create more by-product data.

Section 2.3.1 showed that while serious games are designed for a specific purpose other than entertainment, people will still be willing to play them if they can see a benefit to themselves or others. Furnas et al. showed that altruism toward a community is one of the reasons that tagging systems work well [46]. However, games which incorporate tagging as a by-product can provide unique benefits such as a game dynamic which ensures the tags are useful to a broad community of users [47].

While there are people who might be willing to put time into these systems for altruistic reasons, as discussed by Kuo et al. [48], people will be more likely to take part in these tasks if it is genuinely enjoyable for them to do so. This is described by Krause and Aras [49],

> *"As opposed to manual or supervised approaches of expert-like annotations that have high costs for generation and maintenance – in particular for frequently changing web content – social tagging can be regarded as a simple and easy to use tool to classify and describe relevant web content by human collective intelligence leading to daily meta-data generation."*

These have been referred to as 'Games With A Purpose' [50] due to having a secondary design goal beyond entertainment. However, this dissertation will take the view that most games have a purpose, whether they produce by-products or not. This purpose is quite often to provide entertainment, though this may not always be their primary goal. As such, the term 'Games With By-products' will be used in this dissertation to refer to this specialization of serious games. The definition used will be a game designed to be entertaining enough to be played on that merit alone, but which also produces a by-product as a side effect of play that is of value to a third party.

These by-products are often the result of human computation, as discussed in Section 2.2. However, in some cases it may simply be more practical to use a person's physicality and senses than it would be to create a robotic means of capturing information. In either case, these games are designed to harness the power of people by constructing entertaining games around a problem which humans are good at solving. Such problems tend to be ones that computers, or other automatic methods, are currently incapable of solving. These games with by-products have great potential in harnessing humans in a successful, lasting way. Von Ahn discusses the idea of harnessing humans to do things that computers tend to find difficult [2],

> *"Despite colossal advances over the past 50 years, computers still do not possess the basic conceptual intelligence or perceptual capabilities that most humans take for granted. If we treat human brains as processors in a distributed system, each can perform a small part of a massive computation."*

This quote claims that many humans could be used together to solve problems that mankind finds simple, but computers find difficult. Von Ahn also points out that games are a particularly good way of achieving human computation because they allow the designer to present problems to the user within the game itself. Since users enjoy the challenges of games, making the solutions to those challenges useful outside of the game seems a logical step. People will play games as they already do because they enjoy them, and the program designer will get the benefit of human computed by-products.

Of course, it is reasonable to ask if the humans in this model are doing the 'work' for free. One might consider that they are being paid in happiness or entertainment. Serious games have already shown that people are willing to do work in the form of a game if it benefits them. However, that benefit might be entertainment. A problem with some serious games is that they are not very entertaining. However, if the designer can construct a game that is entertaining and supply it to the user for free, the player may choose to play the game for that reason. We know that people already play games. By making good games that are free, the designers can direct those game hours to their own games. The players may think that they are only playing the game for 'free' entertainment. Whether or not this is true, the human computation they carry out while playing may be of extreme value to the designer. This seems like an ideal situation as the player does what he or she does anyway and the designer now benefits from it.

Amazon's Mechanical Turk has already been discussed in Sections 2.1 and 2.2 as a means of paying people to carry out tasks rather than computers. In this model, the people carrying out the human computation are being paid. However, the tasks to carry out will probably not be as enjoyable as playing a game. Also, paying users for each successful solution or answer is not always viable, especially if the ultimate goal requires a large amount of polling of many different users or having many users doing a large number of small tasks. Thus, in

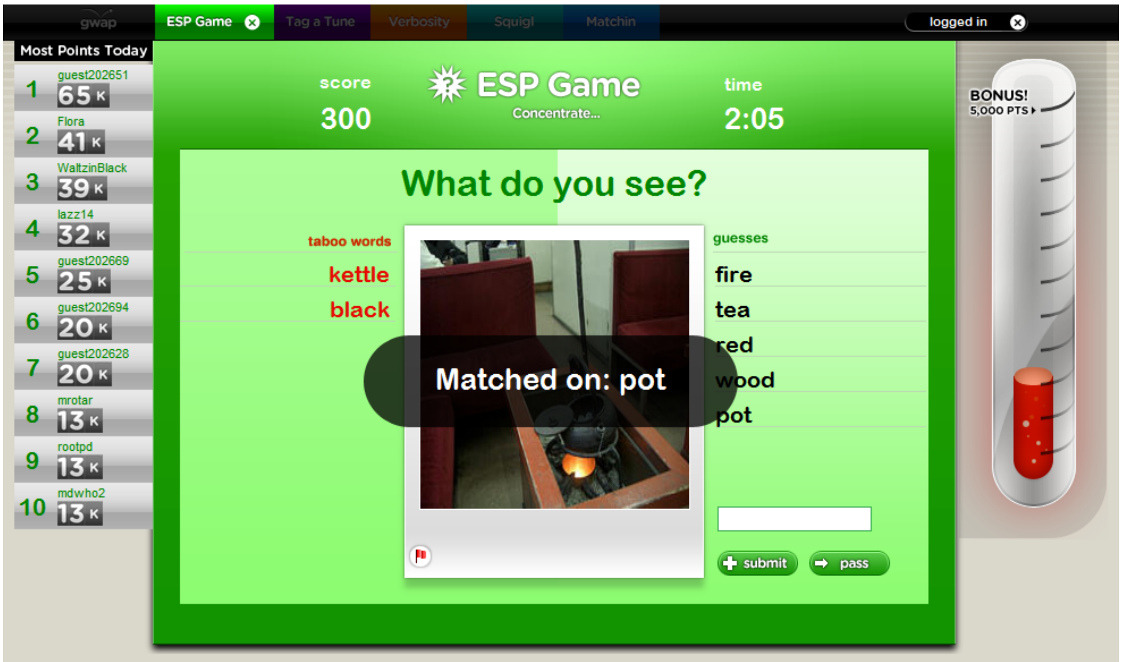

Figure 2.6: The interface for the ESP Game, showing one player having just matched on a word with the other player.

certain situations, it may be more economically viable to spend time constructing a game that people will play in return for entertainment alone.

In order to further demonstrate the concept of games with by-products, several existing games will be described which have these features. Then, common and unusual design features of the games will be highlighted and discussed to show the benefits and limitations of this concept.

One of the earliest games with by-products was the ESP Game [6]. In this game, a user is paired with another player and told that both of them can see the same image. Both users are instructed to try and guess what the other player is typing. In this way, both users try to guess what they are each typing. If there is a match between their guesses, the users gain points and the image is replaced with a new one. There is also a list of 'taboo' words for each image. This list is displayed to the users and indicates words that the game will not allow to form a match, even if both users guess one of those words. The users can also pass on an image if it is too hard. This might imply that the image is of poor visual quality or too abstract to describe. If users repeatedly pass on the same image, then it is removed from the game altogether. The interface for the ESP Game can be seen in Figure 2.6.

Although it is not a necessity for playing the game, the only sensible strategy for trying to guess the other player's typed words is to use the one stimulus that both players have in common: the image that is presented to both users. This will result in both users guessing words that relate to the image. The benefit of this for the game designers is that when the users match on a guess, that guess is usually a good description word for the image. If enough users pair the same word, then it will be allocated to that image as a good description of it. This can then be used as a means for searching images. Users can search for images by entering key words and the images that have matching description words should provide a good result. This is because many people have independently come up with the same word for the same image.

The reason for tagging images in this way is due to the inaccuracy of automated methods. This is described by the game designers [6],

> *"Current techniques to categorize images for these applications are insufficient in many ways, mostly because they assume that the contents of images on the Web are related to the text appearing in the page. This is insufficient because the text adjacent to the images is often scarce, and can be misleading or hard to process"*

This would imply that, for the current techniques to work, an element of interaction between the categorizing system and the website which contains the image would be necessary in order to determine if the surrounding words were actually a good match for the image. However, computers alone cannot achieve this because interactive tasks cannot be described as algorithms. This is discussed by Wegner [51],

> *"Interactive tasks, like driving home from work, cannot be realized through algorithms. Algorithms that execute automatically without taking notice of their surroundings cannot handle traffic and other interactive events."*

This problem might, therefore, be better solved with a hybrid approach that extended beyond traditional algorithms. Turing said that such hybrid solutions to computing problems were more computationally powerful that computers alone [52]. This might suggest that including human computation as part of a system has the potential to be more computationally powerful than a computer alone.

In the case of the Internet, von Ahn points out that there are no guidelines about the labeling of images on the world wide web [6]. It is also worth noting that, even if there were such guidelines, there would be no way to make people follow them.

The taboo words show a clever design by making the game harder (and therefore a little more compelling), and by being generated from the game itself. However, they also provide a needed purpose by preventing the creation of the same labels again and again. If a word is matched enough times to pass a threshold, then it is used as a label for the image and also marked as a taboo word. This encourages users to come up with less obvious words for an image. The threshold idea is a good solution to getting good quality labels. If a perfect image label is one where everyone in the world would agree on it (i.e., describing images with words is very much a subjective thing, but ideally it will work better if all humans agree), then having a threshold of humans who come up with a label would probably make for a good labeling system. Only words that pass the threshold are used as tags for the image (and thus become taboo words).

Another clever mechanism of the game is being able to pass on an image if it is too hard. This will usually occur if there are a lot of taboo words. However, it might also occur if the image is itself a bit too abstract to have a generally accepted label. In any case, images that are frequently passed will be considered fully labeled. Erasing the labels and re-inserting the images into the game again after a few months allows for changes in cultural meanings and relevance to take affect with the image labels. This situation is described by von Ahn [6],

> *"… an image of Michael Jackson twenty years ago might have been labeled as 'superstar' whereas today it might be labeled as 'criminal.' "*

In order to ensure that the game is always available for play, users' responses in the game are recorded so that they can be used to compete against new players when they are the only ones on-line or there is an odd number of players. This has the same effect of having two real people play each other but will perhaps give recorded users more influence over which words make good labels. However, since the other user is always new, it ensures that the results of each game are still valid.

Von Ahn also has a solution to potential cheating. This is done by checking if the average time for agreement drops very quickly. In this scenario, the number of recorded user actions being played against live users is increased. This prevents collaboration between players that try to play at the same time. It is worth noting that some cheating or bad labeling can be allowed if searching images is the main application domain for the data produced by the ESP Game. As long as some good labels appear for an image, the bad ones do not matter so much.

Another solution to the problem of cheating is that the players are kept anonymous from each other. Although users can create a user name and password for the system (so that they can gain points for all games played), this is never displayed to other players during the course of the game. These names are all that is stored about a user and provide no means of contacting users outside of the game. Users with the same network address are never allowed to play against each other so that the likelihood of users in the same physical location being able to cheat is reduced. Also, users are not paired together in order but are randomly put together. This prevents users trying to start a game at the exact same time in the hope of being paired together.

Another game called 'KissKissBan' [53] adds another element to the game to help prevent cheating by having an opposition player. In the ESP Game, both players are working together to gain points rather than competing. In KissKissBan, two players work together to try and agree on image labels while a third player tries to block them. The third player does this by trying to guess the words the other players will come up with, which means they are not allowed to match on these blocked words. If he is successful and the players cannot provide a matched word, he will gain points rather than them. Having players both compete and work together in this fashion provides for a more interesting game dynamic and may help prevent players cheating since they are not always on the same side. The blocker player in KissKissBan also has a similar role to the 'taboo' words in the ESP Game. By coming up with obvious words, he forces the players to create less obvious labels for the images.

Two other potential options to cheat in the ESP Game are discussed by von Ahn. The first is that a massive group of users could all agree on a labeling scheme and then all play the game at the same time. This would increase the chances of being paired with another member of the colluding group. Two solutions to this are presented. Firstly, if the strategy was simply to label all images with the same label, thus ensuring guess matching, this can be easily detected and the system can simply increase the number of recorded user action players in the system. This will make the likelihood of pairing within the colluding group less likely. If problems persist, it would be increased so that live players could only play recorded user actions. Another solution could be to make any agreed word become a taboo word across the whole session for the two users playing against each other, meaning that only the first image would be tagged with an erroneous label. However, the cheaters could still get round this by having a more complicated labeling scheme, such as marking the first image as 'one,' the second image as 'two' and so on. While this could still be stopped by detecting that users were agreeing on words too quickly and then increasing the number of recorded action users, this is less of a problem. Since the images themselves are displayed randomly to the users, what was tagged as 'one' in an earlier game might appear as 'five' in another. Thus, the chances of such labels reaching the threshold amount for an image are greatly reduced because of the large number of images in the system.

The game designer also discusses the idea of theme rooms which allow for expert users in a particular area to label images that are within that area. This might lead to more sophisticated labels for these images. However, the system has no means of checking that a user is an expert, so unless two experts are paired together this system would appear to be unlikely to succeed.

The Internet search company, Google, have licensed the ESP Game to make their own version called the Google Image Labeler [54].[41] The Google Image Labeler's interface shares all the same functionality as the ESP Game except for having the new ability to zoom into images, and the disappearance of the thermometer of the ESP Game which showed your total amount of matches for the current game. The rules and system of play for the two games are identical.

As with the ESP Game, the Google Image Labeler is used in order to provide better tags for an image search utility. However, in contrast to the ESP Game, the Google Image Labeler makes clear to the user the overall purpose of the game. It would be interesting to see if this knowledge affects the users in their motivation or willingness to play one game compared to the other, though it appears that no such research has yet been carried out.

[41] `http://images.google.com/imagelabeler`

Figure 2.7: The interface for the Peekaboom game. The left picture shows the interface for the peek player and the right shows the interface for the boom player.

It is important to note that a large company such as Google has seen the potential of such systems. This suggests that they believe it to be a viable model for accomplishing tasks that computers find difficult.

An inversion of this idea can be seen in the web-based game 'Picture This' [55] [56]. In Picture This, the players are shown a list of images and asked which one best matches a search query. If the players independently agree on an image they receive points. While the ESP Game is used to produce better labels for images, Picture This is used to improve the relevance ordering for image search results.

Another game that builds on the success of the ESP Game is 'Peekaboom' [57]. In this game, two users are paired together. The first user is shown a full image and a word associated with it. The second user is shown only a black image. The first user has to highlight the part of the image that he or she believes that the word refers to. As this happens, the second user begins to see the highlighted area, but nothing else appears. The second user is then asked to guess what word might refer to what can be seen. If the second user can match the word on the first user's screen, both players receive points. In addition, the smaller the area that is uncovered, the more points are gained for a successful guess. After each image is resolved, the users switch roles. The interface for each player can be seen in Figure 2.7.

The first player can also give additional hints. This can take the form of suggesting that the word refers to text in the image, whether the word is a noun or a verb, a related noun can be given and the first user may also send a 'ping' to the part of the image that is especially important. The last case is useful if the word refers to small part of a larger object. For instance, the first user might have to uncover a large amount of a picture of an elephant for the trunk to be recognizable for what it is.

The by-product of this game is that it allows the designer to know which pixels in an image are relevant to which of that image's tags. The images and tags are taken directly from results of the ESP Game. In this sense, the ESP Game feeds into Peekaboom in the form of content generation. An interesting point to be made about Peekaboom is that users are actually rewarded for using the hint mechanisms by receiving more points. This is because the hints actually give additional useful information about the images. However, to the player, this might seem a little counter-intuitive.

The designers of Peekaboom state that the results are designed to be used as training data for automated systems that would pick out objects within an image [57]. They point out that since the percentage of a picture that refers to a tag is known from Peekaboom, it could be used to sort image results by relevance. Von Ahn also states that Peekaboom has potential uses in improving accessibility [2].

The designers of Peekaboom may feel that their game's by-products are only a stepping stone toward a fully automated system for gathering the data [57],

> *"Some day computers will be able to segment objects in images unassisted, but that day is not today. Today we have engines like Peekaboom that use the wisdom of humans to help naïve computers get to that point."*

However, it should be noted that these solutions seem to work well and work quickly. In describing the ESP Game, von Ahn stated [2],

> *"Within a few months of initial deployment on 25 October 2003, the game collected more than 10 million image labels; if hosted on a major site like MSN Games or Yahoo! Games, all images on the Web could be labeled in a matter of weeks."*

Restricting these ideas to the point of a stepping stone seems wasteful. People will always form part of the system in some way, whether it is explicit or not. This would make von Ahn's ideas more generally applicable than just a stepping stone until better automated technologies exist.

Apart from the two players having different roles, many features are carried over from the ESP Game to Peekaboom. Two users are used to provide a validation system for results (in Peekaboom, if the uncovered area is guessed with the correct word, it implies that the word accurately describes the uncovered area). Implementing 'bot' players using recorded user actions is carried over. Also, all the same anti-cheating methods are included, though a few additional ones are introduced in Peekaboom.

Peekaboom uses seed images to prevent third party programmed automated players from diluting the results. These are images for which the object location data has already been discovered. If a bot player is used, it will likely make incorrect guesses a lot of the time (i.e., guesses that do not match the known answers). If this happens repeatedly, the bot players network address will be banned from playing the game.

Another potential for cheating comes from both users being able to see the guesses for the current image. One user could enter his user name for an internet messaging system. Then, when the next image is shown and the user roles are swapped, the other user could do the same. This would then allow the users to communicate through a different channel than the game itself. This can be resolved by only allowing dictionary words to appear as valid guesses. However, the existing anti-cheating mechanisms (such as changing to user recorded action players when the average time to a correct guess is too quick) can still deal with this as well.

An important point to note about Peekaboom is the way in which it builds upon the ESP Game. The ESP Game supplies the image/word pairs to Peekaboom as the basic content for the game. This makes a lot of sense as Peekaboom is essentially a system for making the ESP Game results more fine grained. However, nothing is said about Peekaboom feeding back into the ESP Game. Peekaboom lets a player pass on a word if it is too hard to guess. If this happens repeatedly for several users, that image/word pair is removed from the game. This should have an effect on the same image/word pair in the ESP Game where it originated. Perhaps that image/word pair is not valid. Perhaps it is just a weak pairing. It

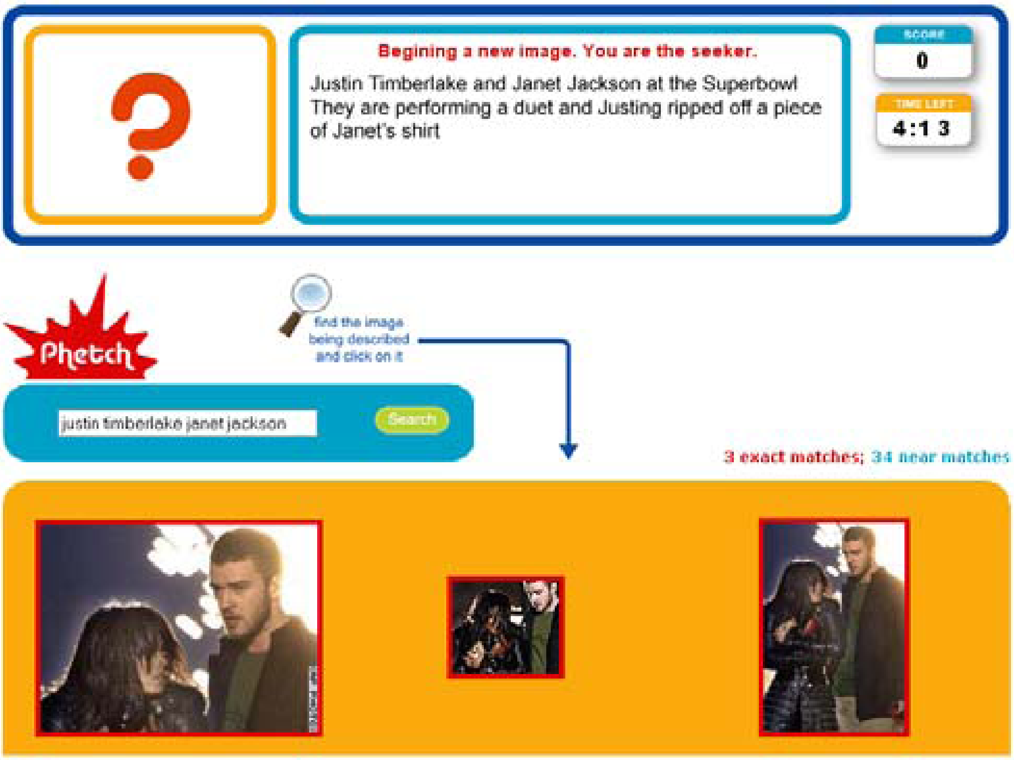

Figure 2.8: The interface for the Phetch game, showing a seeker attempting to find an image that matches the given description.

may simply be that the cultural significance has changed. However, nothing is mentioned about Peekaboom informing the ESP Game in some way that this is the case. It can only be assumed that this result has no effect on the ESP Game or the image search engine which it powers.

Another game which demonstrates human computation is 'Phetch' [58]. In Phetch, there is one describer and a number of seekers. The describer is shown an image and told to describe it to the seekers. This is a one way communication. The seekers cannot communicate with anyone at all and the describer cannot tell what they are doing. The seekers then use an image search engine to try and find an image that matches the one described by the describer. The first seeker to find the same image as the describer wins and becomes the seeker in the next round. A round will last only five minutes. The describer will also get points for a successful match, so it is in the describer's interest to come up with good descriptions. The seeker interface for Phetch can be seen in Figure 2.8.

The ulterior purpose of this game is to get descriptive text for images. While the ESP Game gathered good single words to describe an image, Phetch gathers full textual descriptions. This has obvious uses in accessibility as it would create a description of an image rather than just a keyword tag.

An interesting point to note about Phetch is that the search engine used is the one that is powered by the ESP Game. This provides a good benefit to Phetch. As shown in the ESP Game and Peekaboom, it cannot always be guaranteed that there will be enough players on-line for a game to take place. This is solved by recording user actions then playing them back to the user in order to create the effect of a second player. In Phetch this is very easy to do for describers, but what if only one person was playing and he or she was the describer? Playing back a previous seeker's actions might not match the description and would, therefore, provide unrealistic results. However, by using the keywords from the ESP Game, Phetch simply says that an image was found if the description matches an appropriate proportion of the existing ESP Game keywords. This again shows how these systems can feed into one another. As with the ESP Game and Peekaboom, users can pass on an image if it is too difficult. However, the ESP Game and Peekaboom remove images where this repeatedly happens. It seems that the ESP Game would benefit from Peekaboom informing it that a new label is needed for those images. This would appear to be true of Phetch as well. No mention is made of what happens to images that are repeatedly passed on in the Phetch game.

As with the ESP Game and Peekaboom, this game makes use of anonymity and has similar rules to prevent cheating. However, the nature of this game means it is not susceptible to some of the mass user attacks that users may have attempted in those games.

A potential problem with Peekaboom is that it does not correct poor use of language. However, seeing as the use of language would have been good enough to find the correct image, it is probably better than having no description at all. Ensuring good descriptions in Phetch is perhaps more difficult than in the previously mentioned games. Aggregation is achieved through the single player version of the game. Many seekers are given the recorded user actions of a past describer. If a high enough number find the image, then the description is seen as a good match. A second level of verification is achieved by comparing the descriptions to the keywords achieved from the ESP Game. Descriptors which feature these keywords might be considered better matches for the image.

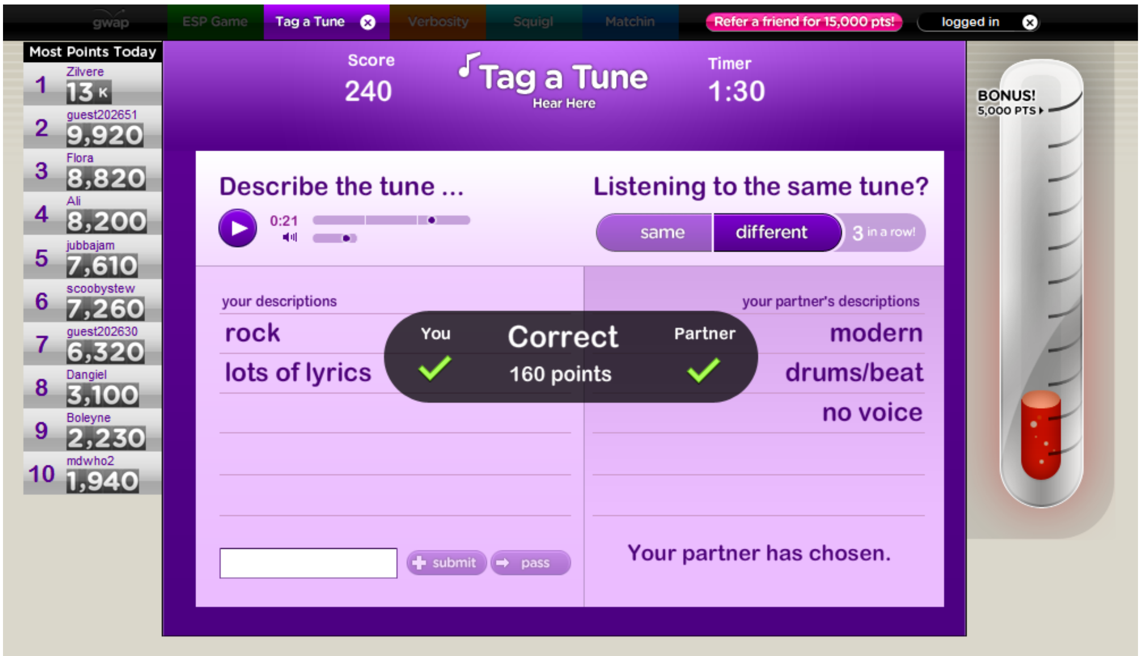

Figure 2.9: The interface for the TagATune game, showing partners agreeing that they are listening to a different tune.

Another game with a similar theme to the ESP Game is 'TagATune' [59]. Whereas the ESP Game was based around images, TagATune is based around audio. Essentially, the game operates in a similar way to the ESP Game with each player shown a stimulus and then given the opportunity to describe it. However, unlike the ESP Game, the stimulus will not always be the same for each player and the players will be able to see each other's descriptions. The players must then guess whether or not they have received the same stimulus. If a player is correct, he or she will receive points. The interface for the game can be seen in Figure 2.9. While the game-play sounds similar in style to the ESP Game, Law et al. point out [54],

> *"A main difference between TagATune and ESP is that what people hear in a sound is often more subjective, ambiguous, and imaginative than what they see in an image."*

There is also the issue that sounds have an inherently more temporal nature than images. While a user could type words that related to a particular part of an audio clip, the time taken to type and that it often takes multiple guesses to agree on a label would mean that it would be harder to match the correct descriptor to the correct part of the audio. However, not including such a feature means that only tags that relate to the entire clip can be entered. This would mean that the user should not really be allowed to enter words until the whole clip has been heard. In a sense, this level of granularity about the audio clips is similar to the way that the ESP Game and Peekaboom are directed at different areas of image labeling.

Another problem with audio in this context is its origin. This is described by Law et al. [54],

> *"While natural sounds can be described in terms of their (imagined) source, music is more abstract, and even the descriptive terms for music are ambiguous."*

A similar game to TagATune called 'MajorMinor' tries to tackle this temporal problem by restricting the clip length to ten seconds ensuring that users will be more likely to enter tags that refer to the entire clip [60].

The purpose of the TagATune game is also very different from that of the ESP Game. While the results could be used to create a good audio search engine, the imagined use is in creating a new form of CAPTCHA [23]. CAPTCHAs are a means of testing whether or not someone is human by asking them to carry out a short, simple task that users find easy to do but computers cannot. This often involves reading a warped version of text that exists within an image. However, TagATune describes the idea of using an audio clip and asking users to describe it. This would perhaps improve the accessibility of current image based CAPTCHA systems which rely on users having the ability of sight.

TagATune, in addition to having similar game play to the ESP Game, uses the same recorded user actions system for the same reasons that have previously been described. Perhaps an interesting diversion for TagATune is that it tries to determine what sounds are more enjoyable to listen to in order to stimulate longer play from the users. This is done by prompting users to rate the audio clip. The closer their ratings are, the more points they each get. Thus, a general level of enjoyment about a piece of audio is aggregated over many users. Sounds that are less enjoyable to hear are included in the game less often to keep the game more enjoyable to play. This shows that human computation is actually being fed back into the game itself to ensure that it continues to be fun.

The largest difference that TagATune has from previous games is that it has a comparison round. In this round, players are asked to what extent sounds are similar to each other, which ones they prefer and which ones make them feel a certain way. This adds a more emotional tagging to the audio than simply having descriptive words. It allows users to compare how they feel about the audio clips.

While TagATune is used to create a new type of CAPTCHA, 'Magic Bullet' is a game used to test the robustness of existing CAPTCHAs [61]. Magic Bullet differs from the other games mentioned so far by being team based. Teams are randomly created at the start of the game with all players being unable to communicate. The players are presented with two shooting targets and an image with a letter on it. As the time increases, the letter moves toward the targets (where there is one target for each of the two teams). Whichever team correctly types the letter in the image first will have it hit their target and they will receive the points for that round. Playing the game creates training data for machine learning systems which can then be used to test the robustness of existing CAPTCHAs.

The games discussed so far have focused on ways to label content so that it could be found by search engines. There are also games which try to improve existing search results using games with by-products.

One example is the 'Search War' game [62]. Two opposing players each receive a web query and some results that the web query produces. Each player must then choose a selection of the results to be presented to the opposing player such that it will be hard to guess what the search query was. This has the effect of determining which web pages are the least relevant to the search query, as well as suggesting better words for these results (based on what the opposing player guesses). In this way, the game will help to improve existing results from a search engine.

While Search War uses an opponent based game-play, the 'Thumbs Up' game [63] [64] uses a cooperative approach similar to games which have already been discussed. As with previous games, this provides human validation. Each player is presented with a search query and two image results. The player has then to decide which result is most relevant. If the players agree on the choice, they both receive points. The game-play allows for better relevance sorting for the image search engine. The interface for Thumbs Up can be seen in Figure 2.10.

Yet another approach is taken to improve search result relevance by the game 'Page Hunt' [65]. This is a single player game in which the user is presented with a web page then

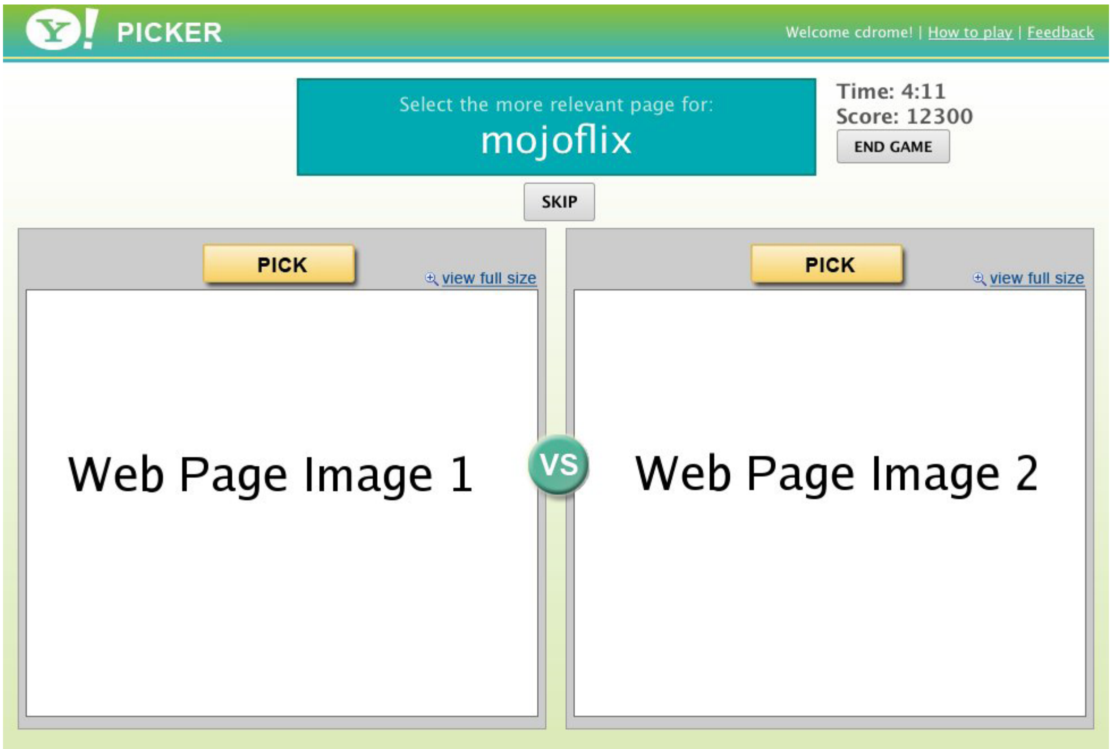

Figure 2.10: The interface for the Thumbs Up game, showing how web pages are presented to the player.

asked to construct a search query that will return it. If the query is in the top results, the player will receive points proportional to the page's position in the results. If the page is not returned, the user is told to modify the query and try again. The changes made to the queries to get them into the top results provide data that can then be analyzed to improve the relevance ordering on search results. While most games use humans to provide some sort of validation, this is not so important in Page Hunt because the queries are measured using an actual search engine. If a player never gets the web page to appear in the top results, their game-play by-products will not be included in the output data. However, there is still the possibility that people will enter purposefully irrelevant words within their search queries but still have enough relevance in the rest of the query to bring up the required page. This could make the by-products harder to analyze later on when trying to improve the relevance of search engine results.

Most of the games described so far have revolved round some form of labeling. However, there are other areas where games with by-products can be used. 'Verbosity' is an attempt to use human computation to gather common sense facts such as, "milk is white" [30]. The game involves two players with different roles. One player is a guesser and the other is a

Figure 2.11: The interface for the Verbosity game, showing one player about to correctly guess the word being described by the other player.

narrator. The narrator is given a word and a number of possible sentence templates that can be used to describe it. The narrator simply needs to select an appropriate template and fill in a blank with an appropriate word. However, the narrator is not allowed to the use the word itself. The guesser sees these filled in sentences from the narrator and must guess what word is being described. All the guesses are seen by the narrator who can then mark them as hot or cold. When the guesser correctly comes up with the word, the players swap roles and are given a new word. The interface for Verbosity can be seen in Figure 2.11.

The game features similar anti-cheating mechanisms to the previously mentioned games. However, a new possibility for subverting the game presents itself with Verbosity. Players could enter lots of useless pieces of information about a word, then enter one or two good ones so that the other player guesses the word correctly. This would mean that all the bad information would be included too. Although the solution to this problem is not mentioned in the paper, it seems reasonable to assume that the validation system would eventually filter out these erroneous sentences.

The single player mode is also slightly different. The narrator is based on previously collected facts about a word rather than on a specific user's recorded actions. This is because it allows the system to differentiate which sentences led to the correct guess and which ones did not. The guesser is much harder to re-create, as shown in by von Ahn et al. [30],

> *"Emulating a Guesser in a convincing manner is more difficult. If a real player enters useless descriptions of the word, we do not want the emulated Guesser to guess correctly."*

However, there is no real way to know that the words entered are good or bad. For example, just because a sentence has not been used to describe the word before does not make it wrong. The solution used is to collect a series of related words for each word used in the game. Although not specified by von Ahn et al., this list is presumably constructed from the words that have previously been filled in to the template sentences to describe the current word. If enough related words are used in the template sentences, then the automated guesser will guess the correct word.

The breadth of by-products created by these games is more diverse than has been covered so far. Other web-based games with by-products include a game for creating question/answer pairs [66], a game to determine the true sense of words within their context of use [67] and games to create annotations for the Semantic Web [68] [69].

So far, all the games with by-products which have been discussed have been web-based. However, the concept is not restricted to the world wide web and there have been examples of these games being used in mobile settings.

One such example is the 'Gopher Game' [70]. In the game, players have hand-held computers and can go around searching for digital 'gophers.' When they find one, they can choose to help it achieve its tasks. At this point, the gopher is not available to other players. The player can release the gopher without completing the task, or the gopher can leave itself from 'boredom.' After completing all the tasks, a user can submit their solutions for the task to a trial by jury where other players say whether or not the answers are good. If passed, the original player can create their own gophers. This means that they can get the other users to carry out tasks for them. The human computation here is very open to what an individual user wishes it to be.

Of course, in reality the people are the gophers, carrying out desired tasks created by the other players. While they may think they are helping a digital gopher achieve a task, they are really doing all the work. This is a clever way of generating content for a game but also for getting people to do things you cannot or do not want to do by putting it in the form of a game.

The trial by jury seems like it might not work very well. Since this is a leader board based game, players might not be honest in saying whether or not they thought the results were good. However, players on the jury are rewarded more points for being closer to the average vote for the jury as a whole. There does not seem to be a strong reason why all the players would not just vote against the other players all the time. This way they would guarantee that they would all get more points for being close to the jury's average vote and the player submitting to the jury would get less.

This raises an important issue about mobile versions of these sorts of games where human validation is involved. The greatest problems with human validation are in constructing a system where all users gain by telling the truth and where players cannot collude to bypass the human validation mechanism. This does not really seem to have been solved with the gopher game. Telling the truth about whether or not a player successfully achieved a task does not necessarily seem the most beneficial way for a juror to vote, especially if jurors could communicate with each other. There seems to be no mechanism in the game to prevent players meeting up with each other. It would seem sensible that jury members should all be spread out from each other and the player submitting the completed task.

However, users will probably congregate around similar areas as they will all be trying to find gophers in order to play the game. An area with a lot of gophers will gather a lot of players. It will also make sense to leave gophers in similar areas in order to increase the likelihood of other players carrying out your human computing task. Thus, the game seems somewhat counter-intuitive in preventing cheating of this kind as the game encourages players to go to the same areas. It should be noted that this sort of cheating did not happen in the game, but it is still a valid issue that must be taken into account for mobile games that aim to use human computation.

The Gopher Game was not wholly successful in user trials. The game designers describe the situation [70],

> *"As a whole the game was undoubtedly too complex for beginner players. It has demonstrated some of the pitfalls of mobile games development, particularly concerning the shared comprehension of a set of rules between isolated players."*

The Gopher Game showed that there is potential for developing a mobile game that uses human computation, but that some of the design principles from the web based games are important and must still be maintained in this new setting.

One final game that is worth mentioning is 'Manhattan Story Mashup' [71]. Although this game was not designed to take advantage of human computation, it shares some design principles with the other games that have been mentioned. Manhattan Story Mashup involves a number of Internet players constructing a story together. This takes the form of users adding new sentences or re-using existing ones which already have illustrations. The illustrations are gathered by a number of ground users with camera enabled mobile phones. The ground users receive single words from new sentences and must find something which would make a suitable illustration for that word. When they do, they take a photo of it. There is a 90 second time limit for taking photos for each word.

In order to validate the photos, two other ground users are shown the photograph and asked to guess what the word was. If they both guess correctly then the word is matched to the photo and everyone gets points. The result of this is a story constructed by a large number of users with matching illustrations. However, since the illustrations are word based rather than within the context of the story, the illustrative photos will often not fit with the story but will still pass the human validation. For example, a picture of a dog might pass for the word 'dogma.'

The game is a good demonstration of adaptable game play working well in mobile situations where environments are unpredictable and impossible to control.

While the ultimate goal of Manhattan Story Mashup was to mix different technologies in order to create a compelling game, it demonstrated the potential use of human computation in the form of gathering images for use in a story. It also faced similar issues of using humans for validation and having to prevent users from colluding. However, the important point to note is that this game was created purely to be as entertaining as possible. Human computed by-products were not a goal of the system. However, human computation did in fact form part of the system. This demonstrates that a genuinely compelling game that has entertainment as a top priority could still incorporate human computed by-products. In fact, since this was not a design goal of the system, it could imply that the design solutions for some of the issues for human computed by-product games actually add to the enjoyability of these games.

2.3.2.1 – Common Designs

Similar designs appear in many games with by-products. There are rules that pair players together and make them agree on submitted data, rules that only accept data when a specified number of people have independently submitted it, rules that prevent players communicating directly or knowing each other's identity and rules that encourage broader data to be submitted. For example, in TagATune [54], players must independently type the same word to describe a piece of audio before it is used as a by-product. In Peekaboom [57], the pixels chosen by a player to represent an image will not be accepted until multiple people have independently submitted them. Verbosity [30] and Phetch [58] use a random pairing system to prevent two users who know each other trying to play together and collude. In the ESP Game [6], there were 'taboo' words that the users were not allowed to submit. These were actually words that had already been used to tag the images in question (the by-product of the game being images tagged with appropriate words). Preventing the players from submitting these words again caused a broader set of data to be collected. The similarity in the designs of many games with by-products means that a successful method of making specific requests for data could be adapted to work on many games.

The games that have been discussed share a lot of similar design elements. While these games have differences in their purpose and game play, there are issues that come up repeatedly and lead to a set of principles that should be considered in the creation of any game with human computed by-products.

Von Ahn et al. consider these types of systems to be analogous to algorithms [58],

> *"A traditional algorithm is a series of steps that may be taken to solve a problem. We consider Phetch as a kind of algorithm. Analogous to one, Phetch has well-defined input and output: an arbitrary image from the Web and its proper description, respectively."*

This is perhaps the first clear goal that all games with human computed by-products should have. While the rest of the issues described may not apply to all future games in this area, they should all be considered to determine whether or not they do. In the case where any of these issues are applicable, they should clearly be addressed in the design.

Payment Through Entertainment

All the games with human computed by-products have had one goal for the player: to be entertained. While Amazon's Mechanical Turk[42] introduced the concept of paying users to carry out human computation, games with by-products try to create an enjoyable experience that is its own reward. As stated by von Ahn and Dabbish [50],

> *"The [...] approach is characterized by three motivating factors: an increasing proportion of the world's population has access to the Internet; certain tasks are impossible for computers but easy for humans; and people spend lots of time playing games on computers."*

If people spend a great deal of time playing games it makes sense to harness that time and put it to a second use beyond that of entertainment. By creating compelling games that users will enjoy playing and go out of their way to continue playing, the users get the benefit of entertainment, while designers get the benefit of human computation. While it may not be fair to say that the players get the game for free, if they enjoy playing it then that seems like a reasonable reward.

Game Types

Many games with by-products fit into one of three game types [50]. These are output-agreement games, inversion-problem games and input-agreement games.

In output-agreement games, players are given the same input and must independently agree

[42] http://www.mturk.com

on an output. The ESP Game [6] is an example of this type because both players are shown an image (the input) and must try and come up with the same word (the output).

Inversion-problem games are characterized by having one player be a describer so that another player can guess what is being described. Only the describer receives the original input. If the guesser can correctly determine the describer's input, both players will receive points. A common feature of these games is to have some level of transparency in the guesses. This means that the describer can see the guesses being made by the other player and give hints about whether a guess is close to the correct answer or not. The two players will alternate their roles at the end of each round. Peekaboom [57] is an example of this type of game because one player, the describer, must decide what area of an image to reveal to the other player based on what is relevant to a given word. The guesser must then try and determine the given word based on the areas of the image that are available to them.

The third type is input-agreement games. In this scenario, two players may or may not receive the same input and must then describe their inputs to each other. The players must then decide if they have the same input. A correct assessment from both players wins them points. The TagATune game [50] worked in this fashion. Each player was given an audio clip to describe and based on their descriptions, the players would then guess whether or not they had been given the same clip. If both guessed correctly, they would both gain points.

Human Validation

Most games with by-products use some form of human validation. One of the issues with human computation is predicting the performance level of players. Users may perform badly, purposely answer incorrectly in order to 'beat the system' or simply get the answer wrong. Human validation allows a system with human computation to ensure that answers to problems are at least viable. Moderation and quality control are vital components in keeping these systems running successfully [72].

The greatest problems with human validation are in constructing a system where all users gain by telling the truth and where players cannot collude in subverting the system in a means to bypass the human validation mechanism.

Some of the systems which have been discussed have tried to construct an element into the game in which two players validate something. Then, once enough pairs of users have also validated that same thing, it can be used as a by-product. This element in the games was usually constructed so that it was to both players' benefit to have a valid answer and that it be validated correctly. In the case of the ESP Game [6], both users are giving valid answers and validating each other in the same step. In Peekaboom [57], Phetch [58], Verbosity [30], TagATune [54] and the Gopher Game [70], validation was largely based on repeating user recorded actions to new players to see if they also agreed with the answers. This was disguised in these cases by making it seem like the user was playing against another live user on the system. This solution seems to apply to a wider scope of problems as it was used in most of the systems which have been discussed. It also has other benefits to prevent cheating and subverting the system's results which is of great importance in ensuring reasonable human validation.

The final method used was simply asking another human if the results seemed correct. This was the approach taken in Manhattan Story Mashup [71] and Amazon's Mechanical Turk.[43] This works well too, but does not provide the secondary benefits of the recorded user actions approach.

Using Recorded User Actions

Many of the human computation games that were mentioned include a means to record user actions during play and then use them later for other purposes. In the case of the ESP Game [6] and TagATune [54], this was used to ensure that if there was an odd number of users on-line (or only a single user) then the remaining user could play against a recorded session instead of a live person. This also supplied a mechanism to prevent users subverting the system by using naming schemes rather than play the game properly. If users were detected to be colluding, then the players were split up and smoothly moved to playing against recorded sessions instead of each other.

As well as these uses, Peekaboom [57] and Phetch [58] also used this system as part of the validation technique. In Peekaboom there is no obvious way to match the validation of many different user pairs as the pixels chosen will all be of a slightly different range. Thus, it is easier to play back the same pixels revealed by one user to many others and see how many correctly guess the word. If this is done with different sets of pixels from different

[43] `http://www.mturk.com`

users, then the one with the highest percentage of correct guesses might be considered the most valid pixel range for the object in the image. Phetch also used a similar method by playing back sessions of the narrator to new live seekers in order to determine if the given description was a good one.

While this may not be viable in all games of this type (such as the Gopher Game [70] and Manhattan Story Mashup [71]), using recorded user actions seems to have many benefits and is an option well worth considering if it can fit within the game successfully.

Keeping Play Competitive

There are also several common game-play features which keep the games competitive for the players. This is an important aspect of designing games with by-products because if people feel competitive, they will play more often. And the more often people play, the more by-products will be produced.

One way to do this is to keep score so that players can tell how well they are playing and if they are improving. These scores can then be kept and used to make leader boards so that players can compare their abilities to that of the other players. These aspects may encourage players to play more often because they want to improve their skill level and be placed higher on the leader board.

A player's position on the leader board can also be used to rank them into different skill levels. This makes it easier for players to quantify how good a player they are and how much better they will need to be to reach the next level. Keeping score and having leader boards are common in many games with by-products such as the ESP Game [6] and Peekaboom [57].

In certain games, skill levels may also be useful as a pairing mechanism, keeping players at the same rank playing against each other. This would keep the game competitive because the players will be playing against people who are close to their own abilities.

Timing the game-play can also be effective in making games more competitive and challenging [50]. This could take the form of requiring tasks to be completed within a given time or no points are given. Another option could be that completing tasks quicker means that players will receive additional points.

We will now go on to discuss some other common designs to deal with specific issues related to cheating, multi-player games and mobile games.

2.3.2.2 – Designing Against Cheating

The games shown have incorporated a number of methods to prevent cheating and subverting the systems. Cheating is an especially important issue for games that incorporate human computation. Subverting the system can lead to bad data degrading the system's results and cheating will most likely lead to this. Since the human validation techniques that have been described can prevent most bad data from being entered into the system, the main goal in preventing cheating is to make sure that users cannot bypass or subvert the human validation systems.

The easiest way for users to bypass the human validation mechanisms is to collude on their answers. This could be in the form of sending messages to each other, either through the game in some way or via some external means. Another possibility is that a group of users could use a naming scheme and all start playing the game at once in order to increase the possibility of being paired with each other. Different games present solutions to these problems in a number of ways. Firstly, there is no way that users can tell who they are playing from within the game. Including this level of anonymity is an important step in preventing collusion. This also extends to preventing any unnecessary communication between players in the game. This may involve preventing non-dictionary words from being sent so that instant messenger user names cannot be passed between players. Users are also not paired in the order in which they visit the site. Random pairing is included to prevent users trying to start the game at the same time in order to be paired together. Lastly, to prevent the likelihood that users are playing from the same physical location (and are thus able to communicate verbally), users with the same network address are not allowed to be paired together.

There is an additional technique used to prevent cheating as well. This involves creating the impression that the players are playing with other live players when they are playing with previously recorded player data or seed data. For example, if the average time for users to agree or get through a human validation technique is too quick, then the system can split up those users and have them play against recorded data instead. This system can deal with most types of colluding as well as the use of third party programmed automatic players that might degrade the quality of information in the system results.

While these are all viable solutions, they might not all be successful outwith the web platform that most of these games have existed in. In a mobile game setting, for instance, it could be much more difficult to prevent users from meeting up and colluding, especially if social proximity is important to the mobile game. Techniques to avoid this could involve having validation spread between users who are not near each other. Since this would reintroduce the anonymity of the human validation, recorded user actions could be used as well. This would introduce all the benefits that such a system would create.

A game that includes much of these anti-cheating mechanisms is Peekaboom. But it should be noted that the designers claim it is rare for users to try and cheat [57],

> *"Although a minority of players might obtain satisfaction from 'gaming the system,' the majority of them just want to play the game honestly. […] Nevertheless, Peekaboom does have a full set of measures to prevent collusion."*

It may be that this is simply good design, but it might be that even a small amount of cheating can have a significant impact on the collected data. Either way, the designers do not seem to discuss their reasoning, though they point out that for the ESP Game, made by some of the same people, there would need to be a very large proportion of bad data to affect successful use of the results in an image search system.

Games that exploit human computation have been effective in producing good quality by-products as a side effect of play. Determining what makes a 'good' by-product is not straightforward. It is largely measured by a by-product's usefulness to the system which takes advantage of it. However, a game might produce some by-products that are better suited to one system than another, even if this is not the system for which it was designed.

Games with by-products generally use a system of rules that reward players for producing by-products that match the system design. However, these rules tend to make many assumptions about the players. For example, assumptions are often made about why the users play the games in the first place. It is generally assumed that users play the games because they are fun and because they want to win. But they might also play for a sense of status by being placed on a leader board. They might play because they enjoy a sense of sociality provided by the game. It might be that there is some monetary reward associated with winning. There is even the possibility that the players know about the by-products and just play the game to be altruistic, as can be seen with other human computation projects such as distributed proofreaders.[44]

However, we cannot discount the idea that people will play because they want to disrupt the system, whether for the challenge of doing so or for some unseen reason.

Whether the players are purposely trying to disrupt the system or simply behaving in an unanticipated manner while trying to play normally, the end result is much the same for the game designers: the by-products will not be what was expected and may not be what the designers built the game rules to produce.

However, there are many ways we can set up the games to minimize the number of by-products that are not 'good.'

Cheating implies that a player is choosing to subvert the nature of the game in order to gain an advantage over the other players. Some methods could be very direct. A player might attempt to examine the packets that are sent to the game server in order to artificially change the scores. Since these methods go outwith the game rules presented to the user, it can be said that the players are 'cracking the system.' Addressing these problems is a technical issue rather than one of game dynamics. Since this is less to do with the design of the game itself and more to do with its technical implementation, this area will not be discussed.

The second general form of manipulating the system is to use the game rules in unanticipated ways without leaving the interface that is presented. Players may do this to give themselves an advantage or to simply sabotage the system. It is arguable whether or not this can be considered cheating, as the players are staying within the rules and presentation that is supplied to them. For this reason, the more general term of 'gaming the system' will be used.

The first point to note is the confirmation step. Many games with by-products require a

[44] `http://www.pgdp.net`

step like this. In the ESP Game [6], players must independently type the same word to describe an image before it is used as a by-product. In Peekaboom [57], one player must identify which parts of an image a word describes and another player must guess the word based on which parts of the image the first player reveals. This dissertation shall refer to this design element as an *agreement rule*.

This rule has its own limitations. For one, it requires that games with by-products are inherently multiplayer. If a game is not time dependent and does not have direct pairing between two players, this does not pose a great problem. However, many games such as Peekaboom, Verbosity [30], TagATune [54], Phetch [58] and the ESP Game have needed to include a bot mechanism so that there is always someone to play with, even if there is no one else ready to play at the time.

A second issue with this rule, which must be acknowledged, is that the common consensus might be wrong. It is possible for many people to incorrectly agree on something. The concept of popular misconceptions has been seen throughout history [32]. In some cases, games have used a training round to ensure that players know how to play the game correctly and will contribute to good by-products. This can be in the form of telling the players that play has begun, but really they are playing against recorded user actions where viable responses can be compared to the player's own [50]. If they pass this test, they can then be allowed to play real players and their responses will join the pool of potential by-products.

However, games with by-products require a general consensus across many players before a by-product can be considered a good one and be used in other systems. Such confirmation steps can allow for this by having a required number of agreements for the same by-product. For example, the by-product will not be considered 'good' until a minimum number of players have independently agreed on it. This type of threshold system is used in many games with by-products. The ESP Game uses it in a direct way by requiring the exact label for an image to have been agreed upon by several pairs of players before it is added to the set of by-products. A slight variation is seen in Peekaboom where the pixels chosen by a player to represent an image are averaged out across multiple players. This is because the game deals with a less precise by-product. This dissertation shall refer to this design element as a *consensus rule*.

Even though multiple people will be playing against each other, it is still important that a consensus rule includes a degree of anonymity. If the players know who they are playing against, then they could collude with players and attempt to only confirm each others' work. Conversely, they could choose not to confirm the work of players who were ahead of them in the score board. Players should not be able to pick whose work they receive and should not be able to tell who made any of the by-products. This problem can be tackled by randomly assigning jobs to players and not displaying who created them. This dissertation shall refer to this design element as an *anonymity rule*.

The anonymity rule is used in many web-based games with by-products. The ESP Game, Peekaboom, Verbosity and Phetch use a random pairing system to prevent two users logging in at the same time in an attempt to play together and collude. They also check to make sure that players are not paired together when their network address is the same. This prevents players from playing against themselves in order to game the system.

The 'taboo' words in the ESP Game prevent users from tagging an image with words that are already associated with it. This sort of rule encourages more diversity within the by-products and the game play. This is in delicate balance with the consensus rule. One is made to ensure that people agree on things and the other is created to ensure that diversity takes place. However, if the balance is maintained well, it will make the game more challenging and fun to play and will keep the by-products more varied and of a good quality. Taboo words can also prevent naming schemes from a mass invasion of colluding users because the system can detect words that are being used too often and being agreed upon too quickly. The game can then add the words to the taboo list but without including them as potential by-products. More complex naming schemes might prevent this tactic from working, but for simple naming schemes, taboo words can prevent cheating. This dissertation shall refer to this design element as a *diversity rule*.

For ongoing games that are played over a longer period of time, the players might only be permitted to play for a limited amount each day. This rule has no direct benefit to creating by-products. However, it does keep the game competitive for players by imposing a limit on the amount that the game can be played each day. If there are multiple players and we want to keep as many of them playing as possible, we need to keep the game competitive for all of them. Allowing one player to race ahead would make the other players feel like they would not be able to keep up. This might make them stop playing altogether. Also, if one player was too far ahead, that player might not feel any sense of achievement in continuing to play and might also stop playing. Keeping things competitive for people may prevent the need to try and cheat to win.This dissertation shall refer to this design element as a *competitive rule*.

The rules for designing against cheating are not always detrimental to game-play and this section has discussed how, in some cases, it may even make the games more competitive and enjoyable. In this sense, these rules can also be seen as common design elements. So far, common designs in games with by-products have only been discussed at a broad level but there are two areas worth talking about in more detail: multi-player games and mobile games. These areas introduce topics of relevance to the research questions of this dissertation and so will be discussed in a little more depth.

2.3.2.3 – Multi-player Games

This section will discuss some general issues to do with multi-player games. Many games which feature human computation or crowdsourcing elements are also multi-player. This means that certain features of multi-player games will be important to games which encourage players to carry out other tasks.

Multi-player games are played together by more than one person. However, there are some games which, though played together, are not necessarily played at the same time. For example, chess might be played through the postal system with each player sending his or her next move to the other person.

It should also be noted that multi-player games need not necessarily be competitive. There are many games which are played cooperatively. This may involve all players trying to beat the game together, or may involve people playing in opposing teams. A common feature of multi-player games is that people compete to improve their standing in the community through some sort of leader board.

However, the defining feature of multi-player games is that people interact with each other through play. Sometimes these interactions are complex enough that they can mimic a microcosm of life outside the game. An example of this was seen in a multi-player on-line role-playing game where a virus was injected into the world without the game designers' knowledge. The virus spread from 'physical' interactions between the game-players in the world in much the same way that a virus would be spread outside the game world. Because people's actions and movements in the game were believed to be similar to life outside the game, researchers felt it would be of interest to study the way the virus spread as a means of predicting how an epidemic might spread outside of the game world [73]. This then led to other researchers contacting the game designers to allow them to affect the game world in other ways to study the effects [74].

It has been suggested that a game can itself be a reason why people might interact or collaborate [75], leading to large and complex social groups [76]. Conversely, people's interactions within a game might lead the designers in the way they develop a game world. This claim is made by Crabtree et al. [77],

> *"... playing of games is [...] inseparably intertwined with their orchestration."*

This has been demonstrated in the multi-player mixed reality game 'Can You See Me Now?' where players added to the context by commenting on the game itself [78]. Players commented on the disparities caused by the location sensing equipment, providing other players with better context to their actual location.

Recent popularity in social networking websites, such as Facebook,[45] has led some designers to create games that work on those platforms. One such game is 'Herd It' [79] which presents players with music which they must try and describe together by correctly agreeing on an adjective that best describes the music. This game also demonstrates human computation as the collected data is used to label the music so that a search engine can find songs based on user entered descriptions. However, the game adds demographic context to the music descriptions that it collects by also collecting some user information from the players' Facebook accounts. These demographics can then be used in researching the system to determine the kinds of people who are playing the game together. This context of use then allows the game designers to develop the game further to match the requirements of those who are playing it, hopefully making it more enjoyable. This is especially important

[45] `http://www.facebook.com`

in human computation games as enjoyable games will be played more often, meaning more work will be done.

Context of use is also of great importance in mobile gaming where the environment can factor into the game design.

2.3.2.4 – Mobile Games

Most games discussed so far have been web-based, but the concept of 'games with by-products' is not limited to the web. As computer games are increasingly being played on mobile platforms, they are dealing with environments that can be neither predicted nor controlled. A mobile game has the potential to involve highly changeable game-play that reacts to the environment in which it is played. It is also possible that the people themselves can react to the environment in unpredictable ways which must also be dealt with by the games. Such context is important in mobile gaming, but what constitutes context? Context can potentially cover a huge number of things, but Dey and Abowd suggest that some elements of context should be considered more than others [80],

> *"There are certain types of context that are, in practice, more important that others. These are location, identity, activity and time."*

The philosophy that some aspects of context are more important than others may indeed be true, but it should also be noted that what is relevant to one player or one game may not be relevant to others. We cannot say that the four elements of context suggested above will always be more important that others, even if they prove to be most of the time.

Perhaps an important point to note is that the context of collected data does not change significantly whether it is gathered implicitly or explicitly [81]. That is, a user's perception of what data means to him or her generally correlates with their actions as well. This means that we need not necessarily rely on human reporting to find out what context is important, it can be measured from their behavior as well.

An example of these perception issues can be seen in urban planning where noise levels are more than just a scientific measurement of volume and the type of noise being created can have varied effects on the people who hear them [82]. This contrasts with the contextual viewpoint of recommender systems which do not try to analyze the content or meaning of past actions. They simply record the actions and compare them to those of other users. However, there may be occasions when past actions do not necessarily reflect an affinity for those actions. In the case of a mobile environment, just passing through an area does not mean that you like that area. Similarly, people in the same area may attribute different meanings to sounds based on their past experiences [83]. This idea is used in shopping malls where music is played that creates a mood of happiness and well-being that influences customers to spend more money [84].

The perception of sounds can be seen in the mobile game 'The Songs of North' which used sound heavily as part of the game-play [85]. However, players felt that the noises were embarrassing to them as it would attract negative attention from non-players in the same area.

Perception of an area can also be affected by the way users interact with it. For example, some mobile games are played entirely within mobile environments, but others can be played with a mixture of both mobile and non-mobile play. For example, the 'Can You See Me Now?' game was played with mobile players and web based players [86]. The interaction between these players was quite direct, with on-line and street players showing avatars on the same map. The on-line players would move their avatars to 'catch' the street players. Successful street players would use spatial and geographical features to their advantage.

The 'FIASCO' game also featured web and street players [87]. In FIASCO, the street players perform 'stunts' which must involve an object, an action and a theme. The location where a stunt takes place is important because the amusement of a stunt might be tied to a specific location. The street players take photos of their stunts and upload them where the web players can rate the amusement value of the stunt. Having the highest rated stunt for an area means that the player 'owns' that area. The point of the game is to own as many locations as possible.

Another example is the CityExplorer [88] game which exploits the players' local knowledge to tag urban areas. However, the verification step was done on a website, separately from the mobile play. This involved players judging the correctness of each others' tags. However, Matyas et al. reported that the players found this “cumbersome” [89].

This introduces an important issue when the point of a mobile game depends on the input of web users: what if the players are also web users? This would introduce a conflict of interest where web players might favor themselves such that they gain more points. In the case of FIASCO, there is no real reason given for the web users to be honest in their opinion so this could potentially be a serious problem.

Conversely, there are communities which rely heavily on trusted on-line users. Geocaching is an activity where people hide containers and take note of the location coordinates. They then publish these coordinates on-line so that other people can use mobile geo-positioning devices to find the container. This ‘treasure hunt’ game only works because the users trust each other not to lie about the location of the caches [90].

People’s physicality can play an important role in mobile games. A game called ‘Treasure’ [91] involves two teams of users with hand-held computers trying to find digital gold coins randomly dispersed around a real area (though these coins were only visible on the users’ devices) and deposit them in a treasure chest. Users have the ability to pick-pocket others who are already carrying coins. Players can shield themselves from this for one minute. After that minute the ‘shield’ needs time to recharge. Mines are scattered around that the players must avoid, but if they activate one they lose whatever coins they are carrying and their device is disabled for 20 seconds. It is important to note that shielding, pick-pocketing and putting coins in the treasure chest can only occur when a player is within Wi-Fi network coverage and close enough to the appropriate object (such as another player or the treasure chest). The interface for Treasure can be seen in Figure 2.12.

Treasure exposes the underlying infrastructure of the Wi-Fi network to the players as they are forced to develop tactics which use it to their advantage. This idea of turning the limits of technology into a feature (such as turning the absence of Wi-Fi into a game dynamic) was a design feature of Treasure known as ‘seamful’ design [92].

Although not intended, a potential by-product of this game might have been to get users to map out areas of Wi-Fi coverage. Since this is not something that the players were consciously doing, this by-product would have taken advantage of people’s physicality rather than their intelligence. We cannot classify the game as human computation because Wi-Fi

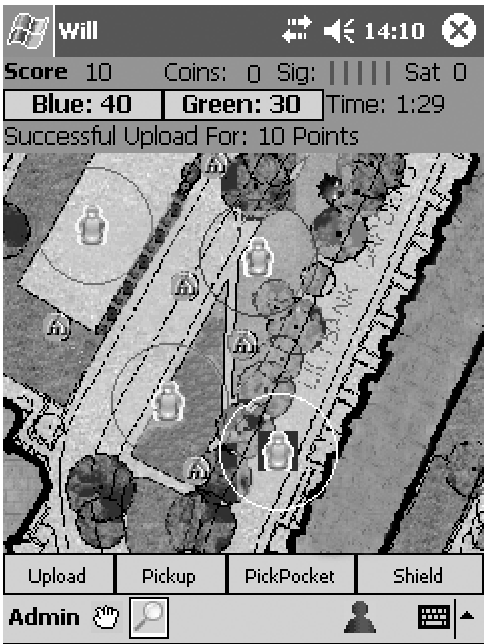

Figure 2.12: The interface for the Treasure game, showing the location of other players and coins on the map.

strength and coverage is an objective measurement which could have been attained with automatic means. However, it may have been cheaper and quicker to do so with the Treasure game.

This system also shows some possibilities for feeding back into itself. If there had been a required by-product of mapping Wi-Fi coverage, the server which acted as the treasure chest and determined the coin placements could have concentrated coins in areas where more information was needed. This would have forced users into those areas and would have had the effect of a more focused mapping of the Wi-Fi coverage.

It should be noted that this system was unable to make any assumptions about the environment in which it would be used other than there needed to be some level of Wi-Fi coverage. The players were forced to develop strategies based on the underlying technologies of the system, but the system design would not have been able to account for these strategies. This shows a great difference to non-mobile games which can often predict the settings for use and act accordingly.

Another game that exploited underlying technologies was ‘Feeding Yoshi’ [93]. In this game, users are encouraged to walk around cities in order to find little animals and planta-

Figure 2.13: The interface for the Feeding Yoshi game, showing items the creature is hungry for and what has been gathered to feed him.

tions. The animals inform the user of what food they need (each animal is able to supply one type of seed which may or may not match what they are hungry for) and the user can then plant seeds in the plantations, pick the food then bring it back to the animal and feed it. The interface for feeding the creature can be seen in Figure 2.13. The users receive points for this and are competing against other players in a leader board to get the most points. In reality, the animals are marked by secure wireless access points and the plantations are marked by open wireless access points. Users had to pull from their local knowledge to determine which places would have dense Wi-Fi coverage.

This has similar potential to Treasure in that it could have the by-product of mapping open and closed wireless access points. This could again be directed by an outside program by making sure that all the animals wanted food that was only available in the area that required more detail. The actual implementation of Feeding Yoshi did not include location data for the wireless access points and merely brought them to the user's attention, but this information could simply be added to the system's design.

Feeding Yoshi was designed so that it could easily fit into the everyday lives of the players. Players could then play in small amounts over the course of many weeks. In this way, the game did not demand intense bursts of play with longer individual sessions. Players could fit the game into short periods during their regular day and their movements became an important part of the game dynamic. It can be seen from these systems that pervasive games can involve the ability to fit into a player's everyday life, the ability to take advantage of a player's local knowledge and the ability to take advantage of a player's location.

As with Treasure, it would be hard to stop collusion in this game. To cheat would only require two users to meet and inform each other of where certain fruits and animals were.

An important point to note about Treasure and Feeding Yoshi is that while both games show potential to have a by-product within the mobile area, neither game would require human validation. This is because the data being gathered does not rely on opinion. While is it technically feasible to build some robot to carry out the operation of mapping out these areas, it is much cheaper to design one of these games to encourage humans to carry out the task.

This idea of using humans for their physicality rather than their opinions was also demonstrated in the 'PhotoCity' game where players were put into teams and then assigned buildings and landmarks and told to take pictures of them [94]. The game interface can be seen in Figure 2.14. The pictures were then used to construct three dimensional models of the buildings. Whichever team had taken the largest number of pictures (contributing the most to the three dimensional model) would then 'control' that area. Whichever team controls the most areas is then the winner. While the pictures themselves were set up by the players, the locations they were to take pictures of were determined by the game designers. In this sense, they did not have a great deal of choice in where to take photos.

A number of mobile games have now been discussed which involved taking pictures as part of the game dynamic. In some cases this produced geo-tagged pictures and allowed the players to determine which areas were relevant. In others, the location was determined for them and the players had less control in their image subject. However, even when the location of an image is known, we still cannot fully attribute which parts of the image are relevant to the intended subject matter. A game described by Arase et al. tried to address this

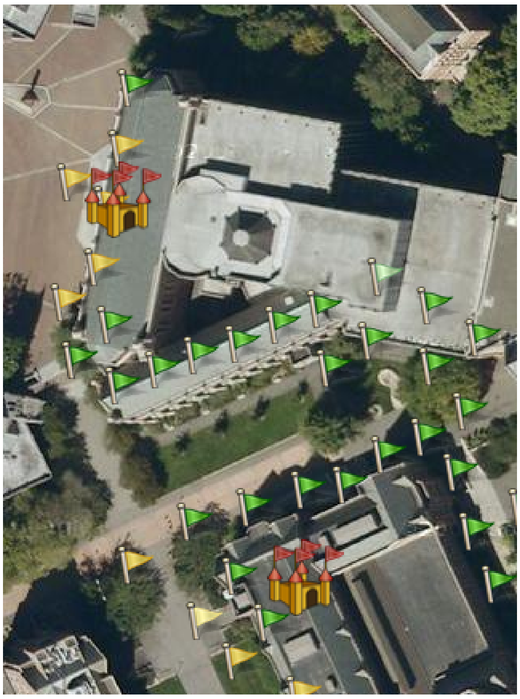

Figure 2.14: The interface for the PhotoCity game, showing which parts of the map are controlled by each team.

issue [95]. In the game, web players reveal parts of an image (taken at a known location) and then mobile players must try to guess the correct location based on the parts of the image that are revealed. This indicates that the mobile players could find the location where the original image was taken based only on the parts that were revealed, indicating that these parts of the image were more relevant.

Another issue with mobile games in which players are told what areas are relevant is that it can be difficult to express the difference between 'space' and 'place' [96]. For example, in setting up the area which mobile players are to take pictures of, a web user might be presented with a map. This was the case in the game described by Arase et al [95]. But the on-line players in that game could only click a point on a map to suggest the area of play. While this may be fine for small areas or buildings, people may conceive of large areas in general ways. For example, a person might think of a complexly shaped, segmented area (perhaps a University) as being one single place, but this same area would take some time to define as a coordinate path and people may not think of it in such strict boundaries. This problem was described by Hightower [97],

"Manual definition does not scale. Instead, ubiquitous deployment requires auto-

matically learning significant regions and semantically labeling them as places."

Hightower has also proposed a method of addressing this problem in which a mobile device gathers continual information from GPS, Wi-Fi and mobile phone towers in the area, allowing it to infer further information, such as when places are indoor or outdoor [98]. By recording which radio signals can be seen and how strong each one is, the system can match an area rather than a specific set of coordinates. However, this still relies on a technological specification of what constitutes an area, rather than what a person might perceive to be a single place.

A large difference between the Feeding Yoshi and Treasure games described earlier is that Treasure was played in a group at a specific time, whereas Feeding Yoshi was played as part of a daily routine and involved no communication with other players. Feeding Yoshi also demonstrates that people will be willing to change their daily routine in order to play a game. This was reported by Bell et al. [93],

> *"... participants [...] would take a different route to their destination, either for work or leisure, in order to play Yoshi."*

This sort of behavior was also demonstrated in a game called 'Mogi Mogi' [99]. Mogi Mogi was a commercial game in Tokyo that involved collecting 'in-game' objects with mobile phones by being in the same physical location that corresponded to the object's in-game position. The players of Mogi Mogi also reported changing their regular routine in order to play the game.

This raises an important point about game by-products being related to how a game is played. In the case of using one application to gather data to be used in another, the data might not be what is expected, or indeed accurate. This could have nothing to do with the capabilities of the technology, but rather the users' behavior while making use of it. In the case of the Mogi Mogi game, people's behavior changed in reaction to the in-game objects they were presented with. This was described by Licoppe and Inada [99],

> *"Active players multiply detours for the sake of the game. In their accounts these detours appear to be opportunistic and improvised displacements. A player 'notices'*

> *a coveted object on her screen during her usual displacements. She leaves her regular path or her home to pick it up. Detours are made often, every day, and concern all types of player. Many alight at an unusual tube station on the way home if they notice an object on their mobile screen, even if this means walking much further to get home."*

Although the Mogi Mogi game did not have a secondary purpose of gathering data, it does demonstrate that the system itself changed people's behavior. Therefore, if the application had been logging data about user behavior, it would have only been accurate in the context of the program. It would not have been an accurate representation of people's behavior who did not use the application.

The concept of tying in-game objects to physical locations can also be seen in augmented reality games where players see the world through a digital display which will add additional digital elements to what is captured by a camera [100]. While these types of games may make the in-game objects feel more real than simply seeing them on a digital map, they often require a larger investment in system programming and equipment.

Conversely, you might have the game react to physical entities that are out of the game. Using mobile systems to react to physical locations was demonstrated by 'comMotion' which reminds users of relevant information when they are near the thing or place that the information pertains to [101].

Another issue in mobile gaming is that of social proximity. This can be seen in a mobile game called 'Castles' [37]. In this game, users have to build up an army on their mobile device using a number of resources which are made available to them. These include things such as the means to produce food, manufacture weapons and train different kinds of soldier. The users can then use their armies to battle each other. This is achieved by being in the same physical location as your opponent and challenging them to a battle. A picture of the interface for Castles can be seen in Figure 2.15.

After the battle, a user may be given hints from the game as to what else they might build or use in order to improve their chances of winning. In some cases, these items will not have been available to the user before.

Figure 2.15: The interface for the Castles game, showing the user about to construct a new building.

The underlying technology is that the behavior of other users is compared to your own and where there is a close match, differences in behavior are turned into suggestions. For example, if your behavior in building your army was similar to that of your opponent but he or she did one or two things differently and beat you in the battle, those one or two differences might make you more likely to win if you adopt them as part of your own strategy.

Not all users start off with the same capabilities. The devices within the game, such as building types and soldier types, are actually software modules which can be exchanged between players. These exchanges are also based on the same recommender system as before.

An important point to note about Castles is that it is based on the concept of social proximity. Users cannot battle each other unless they are near enough for their devices to communicate via Wi-Fi. Also, if a particular class of soldier only exists amongst a group of users and you never battle any of them, you may never get that class of soldier. Of course, a chain of battles between users might eventually lead the soldier class back to you, but the game design encourages users to actively seek out new players.

Social proximity is used in Castles because the designers felt that players who occupied the same location may share other things in common [37]. One example might be a social group where people come together in the same location to spend time with each other, which might involve sharing information that they have acquired since the last time they met.

Even outwith social groups, people who occupy the same area will, by traveling there, have at least some shared experience. However, that does not necessarily mean that they will have the same perspective on that area. In addition to subjective opinion, experience also plays a part in perception of place, such as the difference between local knowledge of a place and a tourist trying to navigate. This situation was described by Brown and Chalmers [42],

> *"… the tourists' problem is in moving from the guidebook to the street they are on. Although they find the house on a map, its street name ('Lawnmarket') is not enough for them to find the house without some work. The street they are on is labelled 'Lawnmarket', it is often simply called 'the High Street' by locals since it is a*

continuation of that street. Confusingly, 'Castle Hill Street' is very close by too."

In this instance, a positioning system would not have helped the tourists because they were in the correct location, but could still not identify which building they were looking for due to disparities between their guidebook and information received from locals.

Social proximity applications suggest that location tracking is not the only way of taking advantage of people's locations, and it may make more sense to identify locations by the people that are there [102]. In this sense, people may define a 'place' that exists within a 'space.' Places cannot be considered independently from the people that inhabit them, even if the people are only there for a moment [103].

Another game which used social proximity was 'Pirates!' [104]. The Pirates! game attempted to encourage people to interact and communicate directly without using the game interface as a middleman. A relative positioning system was used on hand held devices such that the events in the game were triggered by people's physical location in relation to each other. Social interaction is an important aspect of mobile gaming because people are more likely to have a direct and spontaneous connection rather than through a game interface. This ability to 'by-pass' a game interface must be taken into account in the design of mobile games so that it does not allow players to undermine the game dynamic.

Such a system as this with players being in close proximity would not work very well within the common design elements of the games with human computed by-products. These games largely rely on anonymity to ensure good human validation and to prevent collusion between users. While Castles might still be used as a game with a human computed by-product (perhaps to analyze how viruses spread), this may only work where no human validation would be required, as with Treasure and Feeding Yoshi. However, this does not mean that all mobile games with human computed by-products require a system in which human validation does not occur. It just shows that these games were not designed with such issues in mind. The Gopher Game [70] and Manhattan Story Mash-up [71], for example, were both mobile games that included human validation.

It has been shown that mobile games must deal with an environment that is constantly changing, which cannot be controlled and which cannot be predicted. As has previously been shown, human computation is well equipped to deal with such situations of uncertainty and change. This might show that mobile systems would be a particularly good use of human computation. Since it has been shown that human computation is harnessed well in the form of entertaining games, mobile games might be another good means of exploiting this idea.

It has also been shown that mobile games with by-products have their own limitations and constraints which may not appear in other systems. However, these problems (such as preventing collusion and preserving anonymity between players for human validation) can be solved by detecting whether players are near each other and always randomly assigning players together during human matching. The other issues might be solved with human computation itself by introducing a second system that validates the game's by-products.

Both broad and specific design issues that apply to games with by-products have now been discussed. While these issues might affect a large number of games, they will not account for all eventualities of all games. The next section will show some examples where the designers had to leave the common rules to achieve their tasks and how we might learn from their methods of improvisation.

2.3.2.5 – Unusual Designs

Each game with by-products will have its own unique elements. This means that, while the general rules which have been described can be applied to many such games, they will not necessarily apply to all of them. In addition, each game will also have to implement its own unique solutions to encourage good quality by-products.

In the ESP Game [6], it was realized that if a large enough group of players all agreed to play the game at the same time and always 'guess' the same word, they could all get high scores from successfully gaming the system. The solution that was proposed was to detect this behaviour by measuring the time taken for players to correctly guess the same word for an image. If the average time for all players suddenly dropped very quickly, this would imply that the players were gaming the system. The ESP Game uses the recorded actions of previous players to ensure that current players can always be paired with someone, even if it is only a recorded player. By increasing the number of these 'bot' players in the game, the designers could then dilute the number of players who could collaborate to game the system.

In the Phetch game [58], bot players are also used as a means to apply the consensus rule. The game is designed to produce natural language descriptions for images. The is achieved by having one player describe an image that is presented to them. Then a group of between two and four players are required to use an image search engine to try and find the same image. By recording the descriptions of the first player, the game creates a bot that can be used for other players trying to find the image. If multiple people find the correct image based on the description, then it is considered to be accurate. This is a good mechanism to ensure good quality by-products. A similar approach was taken in the Verbosity game [30]. In this game, one player was given a word and template sentences they could fill out to describe it. The other player then had to guess what the word was. To ensure that common sense facts relating to a word were accurate, the descriptions used by one player would be recorded and made into a bot player to be used with other human players.

This system of using 'bot' players could also be used as a means to game the system. For example, in Peekaboom [57], it could be possible for bot players to guess random words or to reveal random parts of the picture. To address this problem, Peekaboom uses a set of "hand-verified metadata" [57]. This means that if a player consistently fails to agree with this metadata, they will be added to a blacklist. Any player on this list will not have their by-products added to the result set. This also prevents malicious players from not playing the game properly in an attempt to create bad metadata. The Peekaboom designers also note that external bot players generally do not work well in the game. This is because they can only make random guesses. If a bot player could successfully play the game, then it would suggest that it was not necessary to create a human computation solution to the problem.

Phetch and Verbosity also try and determine the quality of their by-products by measuring the time taken for the second user to correctly guess the image or word being described. This could also be useful in determining if bot players are being used because the descriptions they submitted would be of a lower quality, making it harder for the other player to complete their task.

While most of the games which have been discussed involve a certain degree of subjectivity from the players, the consensus rule should ensure a reasonably objective set of by-products. However, there are degrees of subjectivity and some games may have to take stronger measures to ensure a relatively objective set of by-products. TagATune [54] is such a game. Its style of play is very similar to the ESP Game. But it is audio, rather than images, to which the players are exposed. The designers of the game describe audio content as being more subjective than what is seen in an image [54]. This can make it very difficult for the two players to agree on a word that describes the audio which would mean the number of by-products produced would be low. To combat this, TagATune uses a category for each piece of audio. This is displayed to both users to help them focus the words they are guessing. Although this is a good solution to ensure that more by-products are produced, it does go against the diversity rule that was discussed earlier in the paper. However, when the diversity occurs naturally, as in TagATune, there is no need to design rules to encourage it. Because the answers can be so diverse in TagATune, using a rule to limit the scope of the words which are guessed will probably produce more by-products and make the game more successful.

Each of these game specific designs show that the game makers determined a special case where the existing rules of design did not fully resolve an issue facing them. In each case, the game makers identified a problem then determined if the existing design rules could be extrapolated in order to solve the problem and, if not, to create a unique solution. But identifying the issue is always the first step and a careful analysis of the game design will be important when creating games with by-products.

The benefits and design rules of games with by-products have been discussed and the benefits of the concept have been demonstrated. However, there are still problems with the idea and some of these will be discussed in the next section.

2.3.2.6 – Problems in Games With By-products

These games often produce generalized by-products and are not designed for the benefit of a specific system. This means that broad by-products are created that can potentially be used in a large number of systems. While this approach may maximize the potential usefulness of the by-products, there will be occasions when specific data is required that may not have been produced. As most games with by-products require validation between players, these players will generally create the most generic data because it is the most strategic way to ensure agreement between players, thus reducing the diversity of the by-products [105].

There is no way to guarantee if required data will be contained in the existing dataset and no way to know how long it will take for that data to be included, if at all. Even if the required data is included within the current by-products, there may be a need for multiple results or more fine grained results.

Another issue with the approach of creating general by-products is that users often come up with the same solutions for similar problems. This means there may be duplication of effort in creating by-products rather than creating a more diverse dataset, though this can partly be solved with systems similar to the 'taboo' words as described in Section 2.3.2. It also means that players may see generalities in the game-play and feel they have solved the problem before. This can lead to a feeling of monotony.

A major problem with the concept of games with by-products is constructing a fun game that can compete against other free games that are created purely for entertainment. Another problem is maintaining the required user base for an application that requires an extended period of human computation. The first issue is mentioned briefly by Law et al. [54],

> *"Fun is a vague concept that is difficult to characterize, since many different elements in a game come together to create the specific experience."*

This describes an element of risk involved in creating a game with the purpose of human computation. If the game does not prove to be successful, then the human computation may also fail to work as no one will be playing the game. While this is a major problem, the success of existing games with by-products have shown that the common elements of human computation games actually seem to be compelling in themselves. This is demonstrated further with Manhattan Story Mashup [71] which includes many similar design ideas to games with by-products (such as human validation and anonymity), but the designers were simply trying to create a game that would be enjoyable to play.

The issue of maintaining interest over time is also difficult to talk about with much authority. Early computer games are not played as much as more recent ones, yet extremely old games such as chess and backgammon still prove popular. While some of the existing games with by-products proved fairly popular and successful on their initial launch, there is no data available on how much these games are still being played. However, as von Ahn et al. have stated [6],

> *"... most popular games on the Web have more than 5,000 players at any one time."*

Therefore, it might simply be the case that a designer has to expect that a game may not always be popular and new games will need to continue to be developed in order to continue the human computation that is needed.

So far, most games with by-products have not been action packed or involved detailed stories and characters. While games with by-products introduce a unique game-play of their own, they might achieve mainstream success more easily if they are able to fit within existing genres. This may be possible and has already been attempted [106] but there are still a great number of genres that are not represented by human computation games. Fitting the games into more common genres may also make it easier to hide the human computation tasks so that people are just enjoying the game. This would naturally be beneficial and fits well with the reasoning behind games with by-products.

Another issue that must always be addressed with human computation is that humans are fallible. Even with human validation methods, it is still possible that two people will erroneously make the same mistake and agree upon it. This is why aggregating the validations is an important step. Rather than rely on a single pair of users to reach validity, having more pairs before accepting validation will increase the chances of viable answers.

It must still be noted that entire sections of society might be wrong: it was once a common belief that the world was flat. This mass fallibility is not always the result of false science spreading but can come about through a concept called the madness of crowds [32]. While the likelihood of this affecting human computation will largely rely on the ultimate goal, it is still an important issue. For example, in the case of the ESP Game [6] the ultimate goal was to construct a more accurate image search engine. If it is a similar user group that play the game and use the search engine, then it does not really matter if the general opinion of the crowd is wrong. In fact, in such an instance the search engine would work better for more people if it matched the erroneous opinion of a majority user group.

In the case of Phetch [58], however, if the majority of the people mistake a picture of Queen Elizabeth II for a lady in a costume, then the blind users that would use the descriptions from Phetch would be very confused by a picture for a fancy dress party being included in a web page about the Queen.

Another example might be in specific user knowledge rather than general knowledge. If a user wanted to use the ESP Game's resultant search engine to find an image for the chemical composition of plastic, it is not likely that they will have much success unless a sufficiently large number of the ESP Game's users were at the same academic level as the user of the search engine.

This shows that if the human computation group and those who use its results are different enough in their beliefs and opinions, the effect of the madness of crowds could have an impact on a system's viability.

A potential benefit of human computation is that it can use human judgment. However, this might be subject to opinion and the mood of the person involved [107]. This is a problem that might be a greater issue within games where people are focused on winning and are feeling competitive. While the games which have been discussed usually incorporate methods to achieve a generally valid opinion across multiple players, it is possible that the game dynamic itself could change the mood of all the players which might in turn affect their subjective opinions. This should be considered when designing the games and when using the by-products.

Games with human computed by-products have been shown to be a successful method for solving problems that computers find difficult. Since many users already play games on computers, it makes sense to make use of that time for human computation. While it may not be true that every human computation task can successfully be turned into an entertaining game, it seems that some of the features required to allow for successful human computation might make a good design for enjoyable games. There seem to be common features that appear in most human computed by-product games such as human validation and anti-cheating mechanisms. These play an important role in keeping the system results of a viable nature and should be used or adapted for any new game system with the focus of human computation.

2.4 – Conclusions

Humans can increase the capabilities of computer systems because of their opinions, mobility and senses. These are things which people find it easy to use, but for which computers generally have no capability whatsoever.

People can be called upon to solve problems, both actively and passively. This is often a cheaper way of solving problems than more direct methods and has many benefits. This system is called 'crowdsourcing.' You are benefiting from the experience of a boundless group of people instead of a specified team. Also, despite many people working on the problem, you may only have to pay the people who construct the best solution, or perhaps not pay anyone at all. People might contribute because they want to win, they want to make money, they enjoy the challenge, they want to be altruistic, they want to benefit their community, they want to improve their standing in the community, they want to feel they are part of something, or simply because it is incidental to another task they are trying to achieve. It has been shown that crowdsourcing can be used in a broad variety of domains, but it must be acknowledged that people might lie or be incorrect.

There are other issues with crowdsourcing as well. Since the call for solutions to problems is open, trade secrets within the business world might prevent it being a viable option. Even though crowdsourcing might be cheaper than hiring people directly, it could still cost money. It has also been shown that people would prefer to tackle easier problems. For the same amount of money, people would rather do lots of easy tasks than one hard task. While more money might increase the quantity of work that people are willing to do, it does not necessarily increase the quality.

The idea of crowdsourcing can be applied as part of a computational process. Using humans in this way to increase the capabilities of a computational system is called 'human computation.' This technique is generally used for things which humans can do but computers cannot. The range of these abilities need not be limited to one person. A large task can be split into smaller problems and distributed amongst a large group of users. The Internet is a useful infrastructure for achieving this collaboration. While human computation is folded into a computational process, it differs from an algorithm because there is no guarantee of when a person will finish a task, if ever, or that it will be what is needed.

Human computation shares many of the same problems as crowdsourcing. Additionally, due to its automated nature and focus on things which computers cannot do, human computation often relies on validation by other humans. But if both parties are wrong, then incorrect information will be produced. Another problem is that the tasks generally do not benefit the users directly and no reward is given. Unless the task is an incidental part of something the users want to achieve for themselves, we may need to encourage the users in some way.

The discussion of 'serious games' showed that we can design games for reasons other than entertainment. Games can add an incentive to a task that is unattractive to the player. People will play games that benefit them, even if the games are not that fun. Common uses include education and fitness, but there is a broad spectrum of applications, including ones that do not benefit the player.

However, games are more likely to be played if people find them genuinely entertaining. While games might benefit a third party, a person is less likely to play them if they do not receive a direct benefit or find it genuinely entertaining. Serious games can encourage people to do certain tasks, but they might only be considered fun when compared to the task, rather than to other games.

Since a major issue with human computation is how to encourage people to carry out the human computation tasks, perhaps serious games can provide a solution. We refer to this as 'games with by-products.' Human computation in games provides a potential variety in game-play that would not be possible with computers alone because it can take advantage of people's creativity.

Games can have alternate goals beyond entertainment and people might play games altruistically if they know their actions are helping solve a problem. The more enjoyable a game is, the more likely people will be willing to carry out the associated human computation tasks within the game. It also means they are more likely to play multiple times, thus carrying out the human computation tasks multiple times and creating more by-products.

There are similar designs in many games with by-products. These help to maintain good quality by-product production as well as keep the games entertaining for the players. Cheating is an important issue because it can affect the quality of the by-products, thus it must be kept to a minimum. There are also many common designs to prevent cheating.

Since human validation is a common requirement of games with by-products, many of the games are multi-player. However, we can still incorporate multi-player elements that help to produce by-products, but which do not center around validation. For example, demographic information and team based group opinion could also be beneficial reasons to develop multi-player elements in games with by-products.

Most games with by-products have been web based, but with the rise of Internet enabled phones, we can extend these techniques out from the web and into mobile environments. Mobile games can take advantage of context and environment, opening up a new domain of use for games with by-products. The game-play can be quite different, often involving travel and being played throughout the day rather than in one continuous period. Social proximity and location can also affect game-play and design. Some of the common designs to prevent cheating and colluding might not work as well under these conditions. We must keep in mind that even with common designs between games with by-products, each game is unique. The particular design issues must be taken into account to ensure that the games will be enjoyable and incorporate good quality by-products.

Even though games with by-products have proven to be successful, there are still several drawbacks. It is possible that the by-products could be incorrect if players are consistently wrong during their generation. Since the content of these games is usually quite similar, there is also the possibility that the games will become monotonous for the players. Additionally, the by-products are generally broad, creating a data set which may not include what is needed in the future. It has also been shown that, while humans are good at adapting to changeable environments, there has been little effort to create games with by-products in the mobile domain. Due to the differences in this domain, many of the existing designs might not apply.

We can try to solve these problems by creating a game with by-products that exists in the mobile realm. The game should incorporate as many of the common designs as are relevant, but it is likely that new designs will need to be determined in order to deal with the new challenges of the mobile environment. This would begin to address ***RQ2***, which asked how can human computation could be extended to collect and classify useful contextual information in mobile environments. In order to prevent the game-play becoming monotonous for the users, we might try having people pose the problems to be solved instead of relying on a data set. It is then possible that the people posing the problems could validate the solutions themselves. In this scenario, it might matter less if the solution is accurate or not, as long as it is what the person who posed the problem wanted. This would begin to address ***RQ1***, which asked how the results of human computation could be improved to match the specific needs of other systems. In the next chapter, the first steps will be taken to producing such a game.

Chapter 3
EyeSpy

The previous chapter introduced the concepts of human computation and games with by-products. It was also shown that there are common design elements to many of these games. So far these systems have largely been web-based. While this has been successful, human computation could be well suited to mobile environments because humans are good at adapting to changeable situations. It was also shown that games with by-products could become monotonous because of the lack of diversity between the in-game tasks. This problem might be solved by having the other users of the game generate the tasks themselves. In this chapter, the designing, building and testing of a game will be discussed which is both mobile and has player generated tasks.

In order to begin, a human computation task was needed that would allow for a game that was mobile and which would allow user generated tasks. In recent years, location technologies have become more common in mobile phones. Similarly, many mobile phones now include a camera. A number of photography websites, such as Dermandar[1] and Panoramio,[2] provide large collections of publicly available, geographically referenced images that were created with phones. Some websites have taken this a step further to produce photographic encyclopedias[3] and geographically referenced historical photographs.[4]

Photographic information about locations is being used in navigation and on-line exploration services. One example is 'Google Street View' [108] which provides street level panoramic views of major cities. The images are obtained from a camera mounted to the

[1] http://www.dermandar.com

[2] http://www.panoramio.com

[3] http://www.fotopedia.com

[4] http://www.historypin.com

top of a car which drives through the cities taking pictures. However, the system only creates panoramas in measured spots around the city and only of places accessible by car. This should not be confused with a contiguous three dimensional representation of city streets. In Street View, users can only see panoramic photographs wrapped into bubbles. This means there will occasionally be details of locations that are not accessible from Street View because that spot is between bubbles. A similar system to Street View is 'Street Slide' [109]. Street Slide attempts to solve the problem of non-contiguous panoramas by creating side views of streets (as if looking out of a car's side window). However, the side views are also constructed from panorama bubbles, so anything that lacked in detail in the bubbles is still not shown in this side view. If a panoramic scene of a location was needed but not covered by Street View or Street Slide, there are commercial companies which provide a similar experience if you are willing to pay the cost.[5]

It must also be noted that the images in Street View and Street Slide are not live and are only updated infrequently. Regular updates to the photographs are probably not a priority for the systems because they are only intended as navigational aids. However, if a property changes hands and is remodeled or a street's layout changes, the lack of updates will have a negative impact on helping to navigate a location.

Street Slide is used as part of an on-line map service called 'Bing Maps.'[6] Another feature of this service is showing geographically referenced photographs from the public. These photos are pasted on top of the Street Slide panoramas, perhaps providing more detail or images from a different point in time. The photos are taken from the photo sharing site, Flickr.[7] Since the photos come from the public, it is possible that they could be more up to date than the panorama, which could potentially mitigate the issues associated with a lack of updates. However, the system provides no way to encourage users to take photos of particular locations or to keep existing photos up to date with what is there.

A last feature of Bing Maps worth noting is 'photosynth.'[8] Photosynth allows users to upload pictures of landmarks and buildings from different angles, then creates a three dimensional point model based on the photographs. This allows users to browse the photos,

[5] `http://www.everyscape.com`

[6] `http://www.bing.com/maps/`

[7] `http://www.flickr.com`

[8] `http://photosynth.net`

stitched together to allow viewing from many angles, as well as see a rough three dimensional model of the location. However, as with Street Slide and Street View, the photos are not live and there is no way to encourage users to take photos of a particular location. The PhotoCity game (discussed in Section 2.3.2.4) uses the photosynth technology to power its generation of three dimensional models. This game can be used to encourage players to create new photosynth models, but in the current version it is only the game designers who can set up which locations and landmarks are used.

The commonality of cameras on phones has led to new uses beyond photography. School children have been shown to appropriate the camera functions of their phones and combine them with their location as part of their play [110]. Combining photographs and location technologies supports new interactions that extend everyday photographic practices. There is the activity of geocaching (as discussed in Section 2.3.2.4) as well as games that are designed to take advantage of mobile phones with cameras and location technologies [111].

While these technologies allow people to see where and when a photo was taken, the photographs themselves often lack textual descriptions. One solution to this problem could be the use of the ESP Game [6] which has the by-product of labeling images. However, the ESP Game does not show the context of the images to the players. People will not know about nearby landmarks or how an image relates to other ones from the surrounding area.

Therefore, a good human computation task for a mobile setting could be the labeling of geographically referenced photographs. Photosynth and Street Slide have shown that the relationships between photographs can be used in interesting ways, therefore, the human computation task should maintain a sense of relationship between photos in the way that they are labeled. We also want the human computation tasks to be generated by the players themselves so that we might relieve some of the potential monotony of the tasks becoming too similar when coming from a database.

The game designed to incorporate this human computation task is called 'EyeSpy.' In the game, players use mobile phones (equipped with location sensing technologies and cameras) to take photographs and create text labels for local landmarks. The players decide themselves what locations and landmarks to photograph or describe with text. The photographs and text labels are swapped amongst the players using the Internet connections on the phones.

The players must then, based purely on seeing a photo or reading a text label, decide where the other players were standing when they photographed or described the landmark. If they can find the correct location, they will receive points, as will the player who took the picture or created the text label. A side effect of play is the production of a corpus of photos and text labels that can be easily located.

The game dynamic encourages 'good' photos of 'what everybody can find' in a particular area. If the location of a photograph cannot be found by the other players, it will not receive points. This encourages players to only choose landmarks that are recognizable and to photograph them in a way that will make them easy to find. The same can also be said for the textual descriptions of locations.

If the text labels and photographs are from the same location, this relationship can be used as a means to search for photographs using text, perhaps in a location based image search engine. A textual search would therefore supply images not just of what matched the search terms, but also of pictures taken from the surrounding area. For example, a search for 'Big Ben' might return pictures of the Clock Tower at the Palace of Westminster as well as pictures of the River Thames, Westminster Bridge, Westminster Abbey and the inside the Houses of Parliament. This could be beneficial when trying to get the feel for an area rather than just a specific landmark.

Because this is a many to many relationship, a single text label could apply to many photographs from the same area. Similarly, a photograph could have many text labels that apply to it. It is also possible that it could simply be used on a photo sharing site, with the users able to see nearby photographs and all the text labels that apply to the area.

EyeSpy thus generates collections of geographically referenced images, but unlike websites such as Flickr, these are photographs designed to be of recognizable locations and landmarks, rather than for more personal purposes. This could make the photos suitable for navigation as they will be easily findable due to the nature of the game's design. As a game, EyeSpy also explores how photography can be seen not only as a hobby or personal interest, but manipulated into new forms of leisure (such as photography games) or as a method of bringing a social group together.

EyeSpy's design makes use of players' local knowledge. This approach has been partially applied in mobile settings before. For example, the Treasure game [91] was designed to create maps of Wi-Fi coverage in an urban area as a by-product of game-play. Treasure's by-product takes advantage of human movement but does not involve human computation as such, because Wi-Fi strength is an objective measurement straightforward to achieve with computers. The CityExplorer game [88] involves exploiting common sense and local knowledge. In mobile play, urban areas were tagged with categories such as "church" and "beer garden." Verification of the tags was done in a later web-based phase, in which players judged the correctness of one another's tags via a web-based interface (a process which the players found "cumbersome" [89]). EyeSpy builds upon such prior work in that it exploits participants' common sense and local knowledge to produce a useful image set, and incorporates a verification mechanism as a fundamental and ongoing part of the game.

Pervasive games take place over a geographical area, and usually over a long period of time, attempting to break the user experience away from the desktop or hand-held and push it further into the everyday world. One early pervasive game, Can You See Me Now? (as discussed in Section 2.3.2.4), was played on city streets, combining on-line and physically present players.

The Feeding Yoshi game (also discussed in Section 2.3.2.4) showed that a mobile game could be incorporated into players' everyday lives. The game-play was designed so that it could be interspersed with everyday life and work over a period of weeks, rather than demanding concentrated use for minutes or hours. Feeding Yoshi enabled play when players had free moments, but also made use of players' everyday movements as a key game dynamic. Thus, an important aspect to pervasive gaming is the way in which players' lives, knowledge and location become key parts of the game. In Feeding Yoshi it was knowledge of areas expected to have a high density of Wi-Fi access points.

In designing EyeSpy, a game was produced that could both be played dynamically over a long period of time over a city's streets (like Feeding Yoshi and Can You See Me Now?), and made use of geographically referenced photographs as a key part of the game, producing beneficial by-products (like the ESP Game).

This chapter will detail the initial work that was carried out on EyeSpy and some of the outcomes from that work.

3.1 – System Design

Designing games with by–products, like any game or user experience design, involves considering how formal rules will be used and interpreted in practice. A game designed to create by-products harnesses the enjoyment, intelligence and creativity of players. We need players to enjoy such games so that they are motivated to make by-products, but we also want them to play only in ways that create what we consider to be useful by-products. We want them to be creative but not 'too' creative; that is, players' enjoyment and engagement may encourage them to find ways to play that help them win, or help them enjoy the game more, but which do not create useful by-products. EyeSpy's rules orient players toward a strong concern for two aspects of navigation, recognizability and find-ability, but nevertheless the rules encourage rather than enforce such an orientation (as shown by Benford et al. [112]).

The key design goal of EyeSpy was to produce a game that would generate geographically referenced photographs and text labels suitable for map annotation. Orientating maps to the environment is a challenge for many map users, and literally just finding where you are on a map as you stand on a street can be a challenging task [113]. Providing a photograph on a map at the right place could greatly assist this task, as well as overlaying the map with the 'texture' of the area [114].

While there are a number of collections of geographically referenced photographs already available, one key problem with these sources is knowing which photographs are usable for navigation. For instance, Flickr[9] photographs are taken and uploaded for a wide variety of reasons (e.g., art, amusement, emotion) and while these are an important part of the value of sharing photographs, it can conflict with using photographs for navigational purposes. Even amongst repositories that are more utilitarian in design, such as the photographs collected by Google Street View [108], one is left with the problem of selecting which photographs to use from a stream of millions. While selecting appropriate photographs by hand is practical for small areas, overall it depends upon employing local knowledge of what are good and familiar local landmarks, and this is difficult to do over a large area.

[9] `http://www.flickr.com`

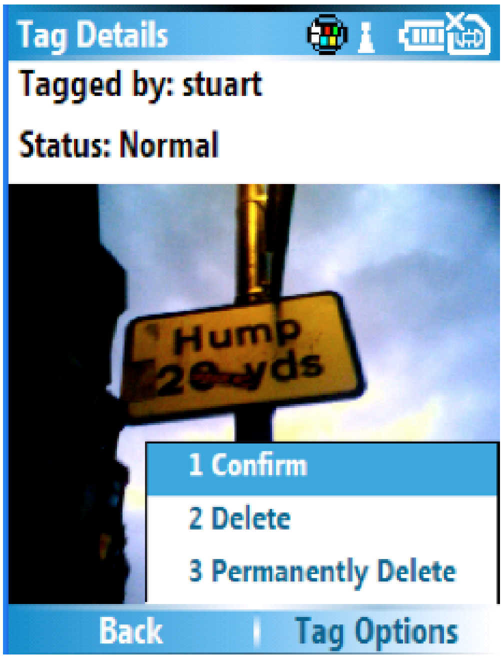

Figure 3.1: EyeSpy as seen on a Windows Mobile phone. A photo is about to be confirmed.

Therefore, our key design goal in EyeSpy was to reward players for producing geographically referenced photographs of good landmarks or easily found objects. In the eventual game design these became geo-located 'tags' that could be easily found by other players: players score points by having their tags confirmed. Players had no control over who could confirm these tags. While the game can be played in small groups of friends, it was also designed so that it could be played amongst groups of strangers. Apart from blocking collusion among players, it was hoped that this lack of control would mean that players would take photographs and write labels that would be sufficiently generic to be findable by any other player. It was reasoned that these photographs would be potentially suitable for navigation since, if they could be easily found by other locals, they might be easily findable by those unfamiliar with the area.

The EyeSpy game was created such that players would create a useful by-product during play. This by-product was a set of photographs that were geo-tagged and had text associated with them. However, the text would not necessarily describe the photographs themselves, but the area in which they were taken. This would mean that searching the set of images for a specific term would return not only images that the term described, but also images from the surrounding area. This could be helpful in allowing users to locate a place by supplying a 'path of images' that lead to it.

In the game, at the start of each day, every player receives a set of ten tags (five photo tags and five text tags) to confirm. This is achieved using the phone's ability to connect to the Internet. The set of tags is randomly chosen and anonymously presented. When the players receive the tags of others, their task is to 'confirm' them. From nothing more than viewing the picture or text, the player must use their local knowledge to determine where it was created. The players then visit that place and go through the confirm step (Figure 3.1). If they are in the correct location, the player will receive points. Players are also able to create new photo or text tags (five of each per day). When a player's tag is confirmed by another, the creator's score is increased. Confirming a tag and having one confirmed is worth the same number of points.

These rules reward players for creating tags that are easier for other people to find. Since the tags can be received by many players over time, it is beneficial to make them as recognizable to as many people as possible so that the player receives the maximum number of points. This is because, even though confirming a tag is worth the same as having one confirmed, a tag might be confirmed many times if a lot of players receive it.

Certain assumptions were made about the reasons people would play the game. The most fundamental assumption is that people will play the game in order to win or do better than other players. However, there are other reasons someone might choose to play. People might not care about their position in the score board at all and simply play because the action of playing is entertaining. It is also feasible that people will play the game to be altruistic, knowing that their play will make the needed by-products.

Another basic assumption is about the cultural background of the players. It was assumed that the people playing EyeSpy would be English speakers with a reasonable local knowledge of the play area. Without these skills, it would be difficult to join in the game.

However, imposing this kind of restriction might not always be beneficial. For example, in the Peekaboom game [57], it could have been beneficial to make the game available in multiple languages for the same set of images. Then the result sets could be compared for the objects in the images. This could allow Peekaboom to function as the basis for a language translator. Of course, this is really multiple versions of the game and in each one the players will use the same language.

An example of designing for this flexibility in play can be seen from the game *Treasure* [91]. In this game, users would pick up coins from places shown to them on a digital map and try and return them to a central store to get points. However, they had to do it without being pick-pocketed by other players. For the game to identify that a player was in the correct location to pick-pocket someone, all the players had to be in sync. This was achieved by using a Wi-Fi connection to the game's server. This meant that when you were not in Wi-Fi coverage, you could not be seen.

The game was designed to make players aware of the Wi-Fi coverage in the play area by showing them where they could and could not go in order to play the game. But some players realized that they could use the holes in Wi-Fi coverage to hide from other players and then make sneak attacks. Similarly, players realized that if they could get close to the central store without entering Wi-Fi coverage, then they could not be pick-pocketed. This was an unanticipated behavior, but it may actually have been more likely to make players aware of the Wi-Fi coverage in the area.

This observation is important when designing these game systems because it shows that not every hole in the game dynamic needs to be filled. If players do something unanticipated, the other users will have some influence over whether it is considered legal. This could range from telling off a player if they see them carry out the action, to refusing to play anymore. This means that we must design rules that the players will consider fair in the first place. While it may seem beneficial to close all holes in the game dynamic and prevent all cheating or gaming the system, the costs of doing this may outweigh the negative impact that such game-play will have on the by-products. It may be better, instead, to design a system where the other players can monitor the system themselves and have some means to punish cheaters.

Additionally, it could be easier to allow bad tags into the system initially, but have a promotion scheme whereby bad tags are never promoted as high as good tags. In EyeSpy, for example, we might only use tags as by-products when they have been confirmed by more than three players. This is an example of the consensus rule described in Section 2.3.2.2.

The game rules in EyeSpy were designed in order to limit the ability of the players to produce 'bad' by-products and to encourage the creation of 'good' by-products. However, we can see from these examples that it may not be necessary to design for every eventuality. In fact, leaving an element of flexibility in the game-play can lead to unexpected benefits which can then be factored into future revisions of the game design.

In Section 2.3.2.1, some common designs of games with by-products were discussed. These common designs, having been successful in several existing games, were applied when designing EyeSpy. These common designs serve to promote the creation of 'good' by-products (that is, by-products that best fit the human computation goals of the game's design).

Payment Through Entertainment

Since the players of EyeSpy will be doing work by creating photo and text labels of locations that are easily recognizable, it is up to the game designers to make this task as entertaining as possible through the creation of an enjoyable game wrapped around the task.

In designing the game, the designers were lucky that the required by-product lent itself to a novel experience of play. The location based nature of the system, as well as the novelty of using photography as a fundamental part of play was expected to keep the game entertaining for the players.

It was thought that the players' ability to choose what locations to tag would also add to the enjoyment of the game by allowing them to be tactical and creative. The potential downside of this system is that EyeSpy would not be used to tag existing photos. While this might be technically possible if the existing photos had location information, they may not necessarily be designed to be easily findable. This might make the game too difficult for the players. However, since the textual labels that are created are applied to an area rather than a specific picture, the same labels could potentially be used for any existing photos that were also taken in that area.

While the design of EyeSpy tries to incorporate many of the common designs from Section 2.3.2.1, most of the earlier games were web based. As such, it is difficult to predict how well these designs will work in the mobile environment in which EyeSpy is played. However, starting off with designs which have been successful in existing areas of games with by-products seems a sensible approach to this new domain, while reasoning in new concepts and dropping existing ones as appropriate.

Game Types

EyeSpy is an example of an inversion-problem game. This means that there is a describer and a guesser. In this case, all players are both describers and guessers depending on which action they are using in the game at the time. If the players are creating tags, they are describers: creating a photograph or a textual description so that they other players can find the correct location. When attempting to find the location, the players are taking on the role of guesser.

In contrast to the inversion-problem games described in Section 2.3.2.1, the describer chooses the input by deciding what to tag. In this way, the players are generating the game content themselves rather than relying on a database of existing content. Another peculiarity is that the describer is not paired with a single guesser. When a describer creates a tag, it could potentially be received by all the other players at some point for them to try and guess where the picture was taken. However, even with these slight differences, the game fundamentally fits into the design of an inversion-problem game.

Human Validation

As in other games with by-products, EyeSpy needed to be designed to tell which by-products were satisfying the human computation goals of the system. The game is designed to reward players who put in the most effort and create the best by-products.

If you put in effort by making more tags, then there is more chance that your tags will be received by the other players. In a limited pool of tags, the player with the most will benefit best when the tags are randomly assigned to the other players. However, creating a large number of by-products will not ensure points, it will just ensure that people are more likely to receive that player's tags. For them to confirm them, they need to be as easy to find as possible because players will surely favor tags which they can confirm quickly and easily as this will be the quickest and easiest way to get points for confirming tags. Since all tags are worth the same number of points, it seems reasonable to assume that players will pick the easiest tags to confirm first and only get to harder tags if they have time. They will also be able to pass on tags that they find too difficult.

There is a fine balance in the game between creating and confirming tags. If no one confirms tags, then no one will receive any points. However, creating tags seems to be worth potentially more points than confirming. This might lead some players to only create tags. However, since players must wait till the next day for their tags to be distributed and confirmed, they do not get immediate feedback on the success of creating a tag. Also, even though it is possible that every player will receive and confirm their tag, it is possible that the random choice will only share their tag with a small number of players and possibly over a long period of time. It is also possible that the players will not be able to confirm the tag because they do not know where it is, or simply because there are other tags nearby they would prefer to tag. However, confirming tags is immediate and points are immediately added. Users can immediately see their status in the game improve. In a sense, creating tags is a longer term gamble which could lead to a larger number of points if a tag is well made, while confirming tags results in immediate points but is worth less overall. It was hoped that this balance would lead people to both create and confirm tags, thus providing a good level of human validation to the number of tags being produced.

Using Recorded User Actions

Section 2.3.2.1 showed that recorded user actions could be used to allow odd numbers of players to be paired together by having one player play against the recorded actions of a previous player. It was also shown that this could be used as an anti-cheating measure as players would not be able to collude with a recording. A final benefit was to improve validation by repeating play with several players to see if the results were the same each time.

While these are all strong benefits, the concept does not fit well with EyeSpy. This is because there is no one-to-one relationship with the EyeSpy players. Tags are not paired up in a direct sense, they are merely swapped around the players. This is perhaps unusual for an inversion-problem game because such games generally involve some back and forth between players. However, since we wanted this game to fit into people's daily lives with them playing a little here and there, rather than being played in single longer sessions, such direct pairing seemed inappropriate.

The method of swapping tags around rather than direct pairing means that there is never an odd man out player. As such, we do not need recorded user actions for anyone to be able to play the game. The existing game dynamic works equally well for an odd and even number of players. This swapping method also means that tags are frequently validated by several players. This mimics the concept of 'repeated' play from recorded user actions meaning that it would provide no real benefit for this purpose.

The last issue of preventing cheating is also not necessary here. Since players are not directly paired and tags are randomly assigned, it is unlikely that people would be able to collude effectively. However, if this was detected, the system could simply ensure that older tags were sent to those players to be confirmed and that their created tags were isolated from the other users. It should noted, however, that this mechanism was not included in the game due to time constraints in its construction as well as the relatively unlikely circumstances that would allow it to be required.

Keeping Play Competitive

Keeping play competitive is important because it encourages the players to play more and makes the game more enjoyable. EyeSpy's approach to keeping play competitive was to reward points for successfully creating tags that were confirmed, as well as for confirming other people's tags. The score for all the players was also displayed on a scoreboard within the game interface so that players could always seen their current position amongst the other players. These techniques were previously incorporated into other games with by-products, as discussed in Section 2.3.2.1.

In addition to the score and leader board, which was intended to encourage people to play more, two additional rules were added to the game to prevent people from getting too far ahead of the other players. To encourage by-products to come from a wide variety of people and to allow the game to fit into people's daily lives, rules were created that would make it difficult for a small number of enthusiastic players with more free time to race ahead in the leader board. If people were able to race ahead in this manner, people at the bottom of the leader board may feel that it would be too difficult for them to catch up, causing them to give up playing altogether.

The first rule to prevent this was putting a limit on the number of tags that could be created and confirmed each day. Players were allowed to create five text tags and five photo tags each day. Additionally, players would only be delivered five new text tags and five new photo tags to confirm each day. However, if players still had tags to confirm from the previous day, these would remain instead of being replaced. Players were also given to option to delete tags. The first option to delete would simply remove the tag from their phone in order to free up a slot so that it could be replaced. However, this did not mean that the player would not receive that same tag again at a later date. This option was intended to allow players to remove a tag because they did not want to confirm it at that time, not because they did not know where it was. The second option was to permanently delete a tag. This meant that the tag would be removed from the phone and that the player would never be assigned that tag ever again. This option was put in place if the player had no idea where a tag was or if the tag itself was badly made and impossible to confirm.

For example, if two text tags were left from the previous day, players would only receive three new tags that day. However, as soon as the older tags were confirmed or deleted, they would be replaced with the new tags for that day. This meant that all players would receive the same amount of tags for the duration they had been playing and had to make a conscious decision to confirm or delete tags, rather than just ignoring them. This last point was important, because tags that were frequently being deleted could be removed from the game as being 'bad' tags that no one was able to confirm. If the game replaced them automatically, we would not be sure whether they just had not had time to deal with them yet or if they really could not confirm them.

The second rule to prevent people racing ahead in the leader board was to prevent people from tagging the same place twice. For example, if people found a good spot, they should not be allowed to create the same photo five times. This meant that players would be forced to move around and find new locations and would have to consistently find good locations to be successful players, rather than relying on one really good location. However, the players were allowed to create both a photo and text tag in the same location.

Section 2.3.2.1 also discussed the possibility of using skill levels and timers during game play to keep the game competitive. However, neither of these ideas were incorporated into the initial design for EyeSpy. While both of these ideas could have provided a potential benefit to the game design, time constraints on implementation and the need to test the basic idea as quickly as possible meant that they were left out. It was also felt that, in a new game, the ranking system would not provide a huge initial benefit and was geared toward longer term games. Since the initial trials were expected to be quite short, this would most likely not have provided a great benefit compared to the amount of time needed to code in the feature. The concept of timing the game play seemed like it could make the game too complicated for the players and could potentially be anti-competitive. Again, due to time constraints and wanting to test the basic game as soon as possible, this feature was not incorporated.

The common designs which were used to form the basis for EyeSpy's game dynamic largely focus on the creation of 'good' by-products. However, the prevention of 'bad' products must also be taken into account. This issue will be discussed in the next section.

3.1.1 – Designing Against Cheating

Designing against cheating is an important consideration when designing games with by-products. While it may be important in other types of games as well, the potential to affect the quality of the by-products produced makes it especially important to ensuring the success of human computation based games.

Whether the players are purposely trying to disrupt the system or simply behaving in an unanticipated manner while trying to play normally, the end result is much the same for the game designers: the by-products will not be what was expected and may not be what the designers created the game rules to produce. However, there are many ways in which the number of 'bad' by-products can be minimized.

As discussed in Section 2.3.2.2, there is a distinction between subverting the game by going outwith the game rules, and subverting the game while staying within them. While staying within the game rules could be referred to as 'gaming the system,' this behavior will be referred to as 'cheating' for the rest of this dissertation. Subverting the game outwith the game rules is a technical issue and a more relevant discussion for this dissertation is the higher level design of the game dynamics.

In this section, the designs that were used in EyeSpy to prevent cheating will be discussed. These designs were influenced by common designs to prevent cheating, as discussed in Section 2.3.2.2. While the common designs of games with by-products serve to promote the creation of 'good' by-products, it might be said that the common designs to prevent cheating serve to minimize the creation of 'bad' products (that is, by-products which do not satisfy the human computation goals of the game's design).

Agreement Rules

There are already sources of geographically referenced and textually tagged images, such as Flickr[10] and Google Street View [108], but EyeSpy is unique because of the type of images that are produced. EyeSpy exploits the players' local knowledge to choose what locations and things to photograph. Google Street View only takes photographs from a car's point of view and because of its blanket coverage of an area, it can be hard for someone unfamiliar with a location to pick out the things that are most interesting to see. Also, the time taken to produce the images in Google Street View means that the images can often be quite old. It would not be helpful for someone trying to find their way to be looking for an empty work site where there is now a new, completed building. Flickr, on the other hand, often has pictures that were taken for personal reasons. As discussed by Arase et al. [95], Flickr contains many "irrelevant images." The rules in EyeSpy encourage photographs that are created specifically for use by other people because the tags must be confirmed. In a sense, the photos are created specifically so that they will be easy for others to find.

Therefore, in the case of EyeSpy, a 'good' tag is one that is well chosen to describe a location such that others can find it. In order to prevent people from introducing 'bad' tags into the system, the game requires that other players confirm the tags. Players confirm tags by going to the same location based only on seeing the tags. It is reasoned that if people can independently find a tag's location from nothing more than looking at it, then it must have been well chosen and well made.

The biggest threat to this system is that players might try and collude. If players can create tags which only they and their partner can locate, it will give them an advantage over the other players. It will also mean that the tags are probably bad because they are designed so that other players cannot find them. However, because the game randomly assigns the tags to other players and tags contain no identifying information about the player who created then, it would be difficult for players to ensure that their tags were swapped amongst each other. While this might happen naturally during the randomization, it is unlikely to happen often enough to give such players an advantage. However, this is based on the assumption that there are only two players trying to collude. If a large enough team of players formed, they could inject enough tags into the system to make it more likely that the tags would be received by more players, but only those within the colluding group would be able to

[10] `http://www.flickr.com`

confirm them.

In this scenario, it is unlikely that the players would to be able to create such a large collection of tags which only they could confirm without some sort of naming scheme within the game. Naming schemes could be included in photos by simply writing something on a bit of paper, then taking a photograph of it. With text tags, they simply type the naming scheme in. The ability for limited communication through a text labeling system in such games was also noted in Peekaboom [57]. However, developing a system whereby a large group of players could decipher a tag's location and no one else could would probably be time-consuming in itself and it seems unlikely someone would go to such lengths. In the case of text tags, we could impose that only dictionary words were allowed, but this prevents the use of proper names which are very likely in location based descriptions. In the case of photo tags, we could include software to analyze the contents, but this seems to defeat the point of the agreement rule which should be based on human computation alone.

A more likely use of 'passing notes' in tags would be signing your name into the tag. This means that players would know not to confirm tags made by people outside the colluding group. However, for this to have an impact on the outcome of play, the group would need to be large enough to dominate the tags in the database. We could of course filter out text tags that include the user names, but this would be harder for photo tags.

In the end, it was decided to ignore these extreme situations and see how common they were in initial trials before investing heavily in time and effort to prevent them. As has already been discussed, filling every hole in the game dynamic may not always be beneficial. The costs of filling such holes may not outweigh their impact on the system. A similar viewpoint was taken by the ESP Game where it was seen that a few bad labels for images would not prevent the good tags from continuing to work [2].

An easier solution, for example would be to take into account how often a player's tags are deleted by other players and how often they are confirmed. If players create tags that are frequently deleted by all players except for the same few, this might indicate a naming scheme in use. The system could then prevent these players from receiving each others tags, and possibly isolate their tags from the other players as well. Again, this system was not implemented due to time constraints, but it shows that a human computation approach to the problem, without resorting to machine analysis of the contents of tags, could provide a reasonable solution.

A subtle way to cheat in the game, and one which can occur naturally and without planning, is to create tags which require extremely localized knowledge. For example, if you wanted to collude with all students of a University, but not non-students, you could take photos of the inside of lecture theaters or other places which non-students were unlikely to be familiar with.

A potential problem that the designers anticipated was how local naming would compare to non-local naming. Because the players of the game are expected to have local knowledge, and they are designing their tags to be easy to confirm by others with similar knowledge, they might not consider things from a world view. For example, tagging a place with the description, "ref entrance," might mean 'University Refectory Entrance' to students of the University where play took place, but would mean nothing to most other people and might not be particularly beneficial as a by-product.

It should be noted that in this scenario, the text descriptions would be useful for tagging photos so that locals could find them. And while it may not be directly beneficial to people without that local knowledge, it is still better than having no meta-data at all. Because the tagging system in the game is location based rather than having one-to-one assignments between each text and photo tag, there might still be other text tags that will return the same photo and will be more beneficial to people without that local knowledge. In this sense, these tags are not 'bad,' they are just beneficial to a smaller group of people.

Consensus Rules

In EyeSpy, we require that three players go through the confirm step for a tag before it is added to the by-products for the system. To better ensure the quality of by-products, we can increase this number when there are more players in the system.

As with the agreement rules above, players might collude such that three or more people manage to confirm a 'bad' tag. As the number of tags in the system increases, this will become less likely, but it could be addressed by comparing the number of confirms against the number of deletions. If it is always the same group of players confirming tags that everyone else is deleting, they may be colluding and the system can address this issue as described previously. Again, this solution was not implemented due to time constraints weighed against the expected level of impact on the game.

Another solution would be to increase the threshold for the number of people who need to confirm a tag before it is added to the list of by-products. However, there would be no obvious way to know when to do this without implementing the solution above.

It is also reasonable to assume that, unless the colluding group was larger than the group of all the other players, 'good' tags would generally have more confirmations than tags created by collusion. This introduces a new idea of having a confidence score associated with the by-products. Rather than simply saying a tag is good or bad; we can show all the tags that pass the confirmation threshold and also show how many times they were confirmed. This might provide and indication of how good or bad a tag actually is. This last technique was incorporated into the game because of the minimal time and effort to include it weighed against its potential usefulness to other systems which might use the by-products.

Anonymity Rules

As has been discussed above, anonymity between players is important in preventing collusion. However, it might be the case that players wish to play the game together, even if they are not trying to collude. They may simply enjoy looking for places to confirm and trying to find other people's tags as part of a group. Unlike the web based games discussed in Section 2.3.2.2, players wanting to join other players in this way (but without wanting to cheat) could be quite common. Unfortunately, it will produce by-products that appear to have been independently agreed upon when this is not the case. However, there is also the possibility that the player could register twice on two separate phones and try and confirm his own tags. However, due to the randomness of tag assignments, this would not be a particularly successful strategy.

While the game design does randomize the assignment of tags to players, it is always possible that a player will be asked to confirm the tag of someone that they are currently playing the game with. In this scenario, that person could simply tell them where the tag was created. It is quite possible that the players will not undertake this action with any malicious intent and will simply consider it an advantage of playing the game together, rather than it being an unfair advantage against the other players. A potential solution to this could be to add social proximity aspects to the game, as discussed in Section 2.3.2.4. The phones could be used to detect other players who were currently in close proximity, or who had often been in close proximity in the past. The game server could then prevent these players from receiving each other's tags. A more extreme solution could be to detect if the player who had created a tag was nearby during a confirm step. If this was detected, the game could respond with a negative message and deduct points from both players as a punishment.

However, these solutions seem a little extreme for what was expected to be a rare (and most likely unplanned) event. Due to the time restrictions of implementing these solutions and their potential negative impact on the enjoyability of play, it was decided to ignore this issue.

Diversity Rules

EyeSpy prevents a player from re-tagging an area which he or she has previously tagged. However, the game treats photo tags and text tags in isolation. This means that a player can tag a location twice as long as he or she uses one photo tag and one text tag. This rule is intended to make the by-products of the system more diverse, as well as making the game more varied for the players. In this system, the players will be forced to think of new areas to tag and potentially explore new areas that they themselves do not know. It was believed that this would be beneficial to the players and the by-products produced by the game.

It was initially considered that this rule might apply across players (making it more similar to the 'taboo' words of the ESP Game, as discussed in Section 2.3.2.2), meaning that each location could contain a single photo tag from just one player and a single text tag from just one player. The reason for considering this is that it was expected that players would, at least initially, choose similar locations to tag in the play area. This would mean that the tags would all be from similar locations. However, since the photos would likely be from different perspectives and the text tags would likely have different descriptions, it was felt this was not such a great concern.

Another possibility was to allow a certain number of tags for each location before it was put off-limits, but it was felt that such a rule across all players would make the game too difficult, especially for new players just starting the game. Had such a solution been implemented, it would probably have required a map of 'untagged locations' to be included with the game.

While this is not necessarily a bad idea, it was felt that it would limit the players' ability to choose locations which it was felt was a major aspect of why the game would be enjoyable. As such, the rule was kept to the player level rather than applying across all players.

Competitive Rules

In order to keep the game competitive amongst all players, EyeSpy puts limits on the number of tags that can be created and confirmed each day. However, it was important to consider how this would affect play over many days. Because the initial version of EyeSpy would be played on phones that used Wi-Fi for their Internet connection, the players would have to manually connect to wireless access points (to enter the network password and such). This would make it possible for players to prevent their phone from syncing with the EyeSpy server for as long as they wished.

One possible application of this tactic could be to save up on the allocation of tags to be downloaded and created. This would mean that players could use their full allocation, but hold off on syncing so that other players became complacent about their place in the leader board. The tactical player could then sync and have their full allowance of tags to create and confirm from the previous days, then have a single long play session and race ahead of players who were not using their full allocation because they felt they were already ahead of the other players.

While such a tactic must be commended for its understanding of the system dynamics and for its strategy against the other players, it does not seem particularly fair that the game rules would allow it. As such, the game does not allow players to save up their tag creation allowance. Each day, the allowance is set to five photo tags and five text tags, no matter how many were created the previous day. In the case of tags to confirm, the allowance system is a little more complicated. Essentially, the allowance is based on downloads, rather than tags on the phone ready to be confirmed. For example, if a player had two unconfirmed photo tags on the phone from the previous day and had not deleted them, a sync would download a further three photo tags, bringing the total on the phone to five. However, that would still leave an allowance of two photo tags which could be downloaded that day if more slots opened on the phone. However, the downloads are reset to a maximum of five every day. The reason for this is that it allows players to take several days to get round to confirming tags but still be able to start fresh with five new tags each day. It also will not detract from other players who have put more effort in and played every day.

So far, the discussion on system design has largely focused on higher level concepts and game rules. The next section will discuss some of the technical aspects of implementing EyeSpy.

3.1.2 – Technology

EyeSpy was designed to run on commodity hardware mobile phones, with Wi-Fi being the only 'high level' feature needed. The trial system ran on iMate-SP5 phones using the Windows Mobile platform.

One of the goals of EyeSpy was to allow play at any urban location, including indoors, so that by-products could be generated for any desired area. However, in selecting a method of locating users we were constrained by the power, storage and processing capabilities of the devices used. We elected to use radio frequency (RF) fingerprints to match the locations of tags users created in the game. Not only did this prove to be an extremely efficient technique on the client devices, it allows for subsequent rapid matching of all the uploaded tags. Through detection of the unique IDs of local RF beacons (in EyeSpy, 802.11 Wi-Fi access points, but could be GSM cell antennae) and signal strengths these beacon transmissions can be used to generate a unique pattern, or fingerprint, which characterizes a particular location. Once this fingerprint is stored a subsequent scan may be used to determine whether the current fingerprint overlaps with the recorded fingerprint, thus ascertaining if the device is at the same location. Using Wi-Fi access points gave the game quite a high granularity and accuracy of locating tags and photos, as 802.11 beacons typically have a maximum range of 100 meters (much lower than GSM) and in most cities are generally distributed more densely. In our tests, we found that the average EyeSpy fingerprint was constructed from data of 7.99 access points. We required at least a 50% overlap before the current scan was said to match a fingerprint, resulting in scans being matched within approximately a 5 to 20 meter range.

In EyeSpy we store fingerprint data—access point MAC addresses and signal strengths, along with text and photo tags. These fingerprints are thus tied to images and textual descriptions on the client device, however we note that the fingerprints are not converted to geographic coordinates. Geo-referencing photographs is becoming increasingly popular, however there are some issues with the use of GPS. Where GPS hardware is available, it often requires time to 'warm up' before it can get a position fix (often several minutes) which does not provide a suitable match with the time frame of taking a photograph (which is often just seconds). Wi-Fi positioning is generally faster, e.g., fingerprinting eight access points takes less than 1 second whereas GPS from a cold start can take minutes. In addition, when using GPS hardware, the device must be powered constantly, thus draining battery power, or be powered on after the photograph is taken, which may lead to the first GPS fix several minutes later being inaccurately used as the position of the photograph. Wi-Fi also favors built-up areas where GPS may encounter considerable problems (e.g., 'shadows'). Our coupling of fingerprints and tags rather than explicitly geo-coding these tags can provide a more suitable method to position photographs on mobile devices. However, GPS does favor a wider availability, and there will always be different areas of applicability for Wi-Fi and GPS positioning.

The procedure for syncing tags in EyeSpy was done over Wi-Fi. Users would need to connect the phone to a wireless access point that had an Internet connection. The phone client could then sync with the game server. While we encouraged players to connect their phone to any access point they liked, we made one available in the play area to be used when needed. The process of syncing was a manual procedure because it required the players to manually connect to a Wi-Fi with an Internet connection.

The majority of the coding work on this first version of EyeSpy was undertaken by Dr. Marek Bell. This included constructing the client and server side code and the general design of the system. The author's contributions included smaller coding roles and fixing the many bugs, including a great deal of time tracking the cause of program crashes and fixing them.

The author also spent a lot of time constructing an Eye Spy experimenter website. This website allowed the experimenters to monitor the by-products from the trial as they were being submitted. The main function of this website was to list fingerprint locations. Since Eye Spy used Wi-Fi positioning, the locations were based on what Wi-Fi access points could be seen by the device. A 'fingerprint' was the name given to this type of location. The percentage of Wi-Fi access points that needed to be seen for the fingerprint to be recognized could also be changed. This means that if two fingerprints had the same percentage of visible

Wi-Fi access points, then they would be considered to be the same fingerprint. The website would then list all the tags in each fingerprint based on this percentage threshold. The website also allowed the experimenters to adjust this percentage to see how it would affect results, but this did not change the percentage that the players required to confirm a tag. The fingerprints could be ordered either by 'most tags' or by 'most confirmed tags.' This allowed the experimenters to see where most play was taking place.

The website proved useful in the initial user trials of Eye Spy as it allowed the experimenters to see the overall activity level in the game, and to show how much each user was submitting. It also allowed the experimenters to quickly determine which players were having difficulty with the game, contact them and resolve the problem. These problems were partly due to personal problems the players had which prevented them from playing as fully as they had intended and partly due to a bug with the Wi-Fi syncing that prevented two of the players continuing to play until the experimenters resolved the problem.

3.2 – Trial Design

To test the game design, it was trialled with 27 participants (8 female and 19 male) over three separate rounds. In the first round, nine participants played for one week. In the second and third rounds, nine participants played for two weeks. In the first round we focused on encouraging as much play as possible, seeding the game with our own photographic and text tags at the start of play. In the second and third round, the game was not seeded and was played for twice as long.

In each round, participants were paid £10 for each week that they played. The winner of each round was given double the amount of money as a prize. This was done to encourage a sense of competition between the players.

In the first round, players were drawn from computing science undergraduates, who knew one another before starting the trial and who had existing social ties. The second round involved a more mixed group of seven non-computing science students plus two non-student participants. Eight of the players in the group were acquainted with each other, although this group on the whole did not have strong social bonds. The third round involved eight computing science students and one engineering student. The players in the third round had no social ties.

The players in the trials worked, studied or lived in the area of the city around the University of Glasgow. This acted as a natural limit on the 'game area.' In addition we asked players to restrict their play to around this area, so as to prevent the game becoming too difficult to play. However, players did at times make text and photographic tags outside this area.

The first round used questionnaires at the end of the trial. This proved relatively ineffective in determining the enjoyment levels of the participants and their other thoughts on the game. The answers from the questionnaires were not especially detailed and although the data was useful, the experimenters determined that interviewing the participants would be necessary in future trials. As such, every player was interviewed at the end of the second and third rounds.

The interview transcripts were coded and analyzed for key themes. Special attention was paid to the reported motivations of players, the different game styles and strategies that players adopted, and the information shared and relationships between players both through the game and outside of the game. Lastly, we looked at where players went and their relationship with those places in playing the game. In particular we were interested in how the game dynamic developed and how players oriented to the rules of the game in producing their photo and text tags.

The next section will discuss the initial findings from the trials of EyeSpy and how they led to improvements for the next iteration of the system.

3.3 – Initial Findings

Broadly, the players created a mix of photo and text tags: out of 632 tags overall there were 369 photos (58%) and 263 text tags (42%). This does not include the seed tags that were included in the first round. The second and third rounds did not use seed tags or the tags from previous rounds. Players produced on average 24 tags during the game: 14 photos and 10 text tags. Figure 3.2 shows a portion of the image tags generated by players from

Figure 3.2: A portion of the photos collected during the trial of EyeSpy (area shown is approximately 400m²). Regions in grey are streets and open areas inside the University of Glasgow.

both rounds. As might be expected, the tags were mainly of geographically prominent landmarks such as statues, street corners, shops and so on.

After buildings, the second most frequent tags were doors or boundaries (i.e., entrances to buildings or boundaries between different places). Interestingly, players also made text tags involving simple riddles, such as creating the tag, "music to my ears on uni gardens," (rather than explicitly stating the tag's location outside the Music Department, on the street called University Gardens). Crucially, there were no tags taken involving 'transient' objects in the environment (such as cars).

Tags submitted by players were also confirmed in the game by other players (a player must physically be where a tag was made in order to be able to confirm it). 43% of player generated tags were confirmed, however, some tags were confirmed multiple times. In the first round, which lasted one week, 21% of player-generated tags were confirmed, although if we include the 'seed' tags that we used to start the game this rises to 43% as nearly all the seed tags were confirmed by players. In the second round, which involved no seeding but lasted two weeks, 40% of tags were confirmed. In the third round, which also included no seeding and lasted two weeks, 59% of tags were confirmed suggesting that the confirmation rate increases with more prolonged play.

Player Motivations in EyeSpy

One of our first concerns with EyeSpy was how well the experience worked as a game. As can be inferred by the large number of tags entered by each player, the game did have some success as an experience, with players reporting that they 'enjoyed' the game, that it was 'fun' as well as 'easy to use.' While an empirical measurement of fun is difficult to find [115], almost all of the users said they enjoyed playing the game.

The game presented a number of different motivations for players. Firstly, the players' scores were visible within the phone application. This proved to be a significant motivation for some players, echoing results from experiences with the ESP Game [6]. Indeed, some players contrasted this motivation with financial rewards for playing the game, such as prizes or payment. Websites that attempt to distribute small tasks over the Internet, such as Mechanical Turk[11], frequently work on the basis of small financial rewards. Yet for our players the ability to compete and win against their friends (and even complete strangers), was more than adequate motivation, in spite of being a small sum paid to participate (which obviously did provide some level of motivation in and of itself). As a pair of players stated:

> *"You've got two very competitive people here, and you've got someone who's first, and we want to move up to first position."*

A second motivation came from the interaction that grew between players around the photographs. On the whole, forums for social interaction in the game were fairly limited, since it was only tags and scores that were shared between players. Moreover, in the second and third rounds of the game the majority of players only knew one or two others well. Yet despite this, players talked about how this narrow channel did provide awareness of other players, and in turn more motivation for play. Players talked about being connected to

[11] `http://www.mturk.com`

other players in that they created tags of similar landmarks. One player mentioned 'walking in the footsteps' of other players, confirming their tags, but also creating tags that were in response to previous tags taken.

Yet the game did lack much in the way of other communication channels to support sociability. A number of players expressed disappointment that they could not easily find the other players, and most expressed a concern about noticing others playing the game:

> *"When I was walking around taking photos I was wondering if I'd run into anyone else with the same phone, or if somebody would spot me with the phone and be like 'ha.' "*

This following at a distance was deliberately played upon by the game's name: 'EyeSpy.' As with many ubiquitous computing experiences that involve tracking, EyeSpy raises a range of issues about privacy, tracking, self-monitoring and the like. While a game played as part of a trial is perhaps an insufficient test of these issues, it is worth noting that none of the players mentioned these concerns while playing the game (perhaps because of the lightweight connections between players). It would be difficult, for example, to be able to gain specific, rather than very general, location information about other players. Photos and text tags were shared without any identifying information, and were selected at random from the pool of contributed tags.

A further issue related to the motivation was how the game encouraged new interactions with the environment. Participants frequently mentioned how the game provided opportunities to explore new areas, as well as experiencing well-known locations in a new light. Players also noted the health benefits of playing the game because it increased the amount they walked around,

> *"I remember walking round for ages thinking, you know, this is probably good for me in some way [...] I like the exploratory part [...] it reminded me of when, you know you go to another country, and you're wandering round for ages."*

Player Strategies in EyeSpy

This notion of 'wandering and exploring' brought us to consider how players oriented to each others' movements when taking part in the trial. Like other pervasive games, in order to play EyeSpy players must leave their homes or workplaces, and travel around the streets to both make and confirm others' tags. However, the act of creating a text tag or taking a photograph involves the player in a specific interaction with their environment. This draws on players' local knowledge as they select where to author a tag from all the potential photographs and textual descriptions that might be made.

Moreover, to be successful in the game, players needed to consider what other players' local knowledge is likely to be. Co-players need to be able to find a given player's tags for that player to receive confirmation points. For photo tags it is important that the photographed object is recognizable and can be quickly found, at a glance, from all the potential places in the game area. For text tags it is in turn important that the text can be used to locate an area quickly and with sufficient accuracy to be registered by the game as the same area. To be successful at the game, players have to consider what a 'general' player might know about the area in which the game is being played.

For players, the key challenge of the game thus came from this problem of how and where to author tags so as to be successful (i.e., to increase their chances of winning the game). They made tags that others could quickly identify and be willing to locate in order to confirm. Authoring such 'good tags' therefore required recipient design [116]. Players creating tags took into account the perceived behavior of other players. This ranged from their knowledge of the area, to their expected route and even to their social role. One participant, who knew one other player socially but not the others, tried to author his tags for what he thought the other players (identified as 'students') would be able to find and confirm:

> *"I thought that a lot of places I didn't know where you were so I'll go for the obvious targets [...] obviously a lot of the places I didn't know where they were because they were Uni names so I thought I'd go for the obvious targets."*

Players consistently referred to the importance of making a photo tag recognizable, identifiable or obvious using what they considered to be a landmark or central place.

What constituted recognizability for players revealed a concern for the navigational experiences of other players. For instance, at one point a player decided to tag a, "... gargoyle thing that's got a unicorn," which she considered to be, "quirky," but then changed her mind since, "… maybe other people won't know about it." This concern is also revealed in another player's comment about making a 'good tag': "… if [other players] can recognize [a tag], that's enough." Producing something recognizable would mean that other players would, "… know exactly where [tags] are when they see them."

Of course, some tags would not be instantly recognized by players. Recognizability involved a design for find-ability. Players, in considering how findable a tag would be, often reported hypothesizing over how other players would go about navigating to a tag. One player reported changing her play strategy in order to achieve this, at first tagging 'random places' and then starting to consider questions such as 'where would [people] walk?' denoting her attention to the navigation practices that other players she assumed would come to engage in when seeking out her tag.

Finally, what determined a 'good tag' also sometimes depended upon a relationship to other tags. In creating tags that were as easy as possible for people to confirm, one key technique simply involved putting tags spatially close together and "… think[ing] about how people were working" in order to decide where to place them. Players in this way produced a trail of tags that could be created in the course of one walk, but also that a prospective fellow player could walk along confirming multiple tags. Unfortunately, because of the game dynamic (only giving a subset of tags to each player to confirm) it would be unlikely that they would receive more than one tag on any trail negating this potentially beneficial strategy.

Players consistently oriented to the concerns of recognizability and find-ability when acquiring photos in order to make 'good tags' for the purposes of navigation within the game. Crucially this involved players designing their tags according to 'what anyone knows' (i.e., shared local geographic knowledge [90]) and the presumed activities and orientation of these fellow players to the game [117]. As one player commented, for example, "you could tag the Mitchell [Building] because everyone knows where it is, but who's going to be [bothered] to tramp across town?" This thought is important when considering how the photos might provide useful navigational tools outside of the game since they are exploited within the game as a form of 'pictorial instruction.'

In turn, confirming a tag demanded some detective work in finding where the tag was, going to that location, and then attempting to position oneself in the same site that the tag was made. Players pointed out that this was easier for photo tags, since with text tags there was typically a much greater ambiguity about exact location. This is perhaps one reason for there being a greater number of photo tags when compared to text tags generated in play, as well as the generally larger number of photos that were confirmed.

In this way, confirmation of tags also reveals a more detailed level of find-ability: when a player had successfully located the general area of a tag, regardless of whether they had experienced it as recognizable straight-away or instead needed to search it out, the player then had to align their phone's current fingerprint with the fingerprint of the tag. This was done routinely by players; they located 'exact spots' by aligning themselves as demonstrated in the photographs. For instance, a player reported being "sure [they] had the right place" given that it was "exactly the same as it is on the picture." Players anticipated others' actions, and chose orientations and alignments to their photos to make confirmation easier for others. As one player stated, it would be 'easier for [another player] to figure out where I was standing.' This aspect is key when considering the navigational qualities of the photos taken. However, we note that this is less the case for text tags, although players did sometimes design them for find-ability through explicit instruction, such as, "boyd orr building facing qm."

Game Area Saturation

In the dynamic and flexible way tags could be created, however, the game did have some shortcomings. In particular saturation could be achieved in a given area quite quickly when most 'obvious' landmarks or easily findable areas had been photographed or tagged. This was due to the players' orientation to taking tags for a generalized 'co-players' route as well as orienting toward designing for find-ability, resulting in players gradually being unable to resolve this orientation for increasingly more 'obscure' landmarks. Saturation also occurred thanks to a low effort threshold, brought about by the game being played as the part of everyday life; players were only willing to sacrifice a limited amount of time going outside of their daily routine.

This situation was compounded due to the nature of the location sensing technology of the phones. Because the phones were using Wi-Fi fingerprinting to compare the location of tags, the phones were not capable of determining where they were. The phones could only tell if their current location was the same as the one where a tag was made. These fingerprints were made from scanning the Wi-Fi signals in the area. However, in some cases, there were no Wi-Fi signals present. This problem was most noticeable in parks or locations away from populated buildings. Unfortunately, a large park existed within the game area, but was essentially out of bounds because the phones could never get enough Wi-Fi information to create a fingerprint.

However, some players did attempt to overcome saturation via increased creativity in constructing tags within highly saturated areas. One player mentioned, for example,

> *"... trying to take a photo of the same monument [which was already tagged] but from an angle that was a wee bit more abstract; I was almost forced to be artistic in the way that I took it because I wasn't really near anything obvious, and I knew there was quite a lot of different signals in the area, so I thought if I stand next to this tree in a certain way that people would be able to figure out which tree it was [and therefore be able to confirm the tag]."*

In addition this further illustrates the concern for recognizable and findable tags that players oriented to, and the lengths they would go to overcome this problem when faced with difficult, saturated areas.

3.3.1 – Unexpected Behaviors

The design of EyeSpy was made to promote the creation of usable by-products. Many of the game rules and dynamics were specifically created to achieve this. As has already been discussed, we realized that there were still some holes in the system that could be exploited to affect the quality and range of by-products, but we felt that their impact would be negligible and so we chose not to implement solutions for them. However, while trialling the system, there were several issues that came up which we had not anticipated. It should be pointed out that the participants were told that they would not be penalized by us for gaming the system, as long as they stayed within the actions presented to them by the game's interface.

Perhaps the most unexpected behavior was players creating junk images. The EyeSpy dynamic of requiring users to take and confirm tags is set up to ensure the viability of those tags as by-products. For example, if a user takes a picture of the inside of their pocket or enters text that is meaningless, the other players will not be able to confirm it. While this did happen during the trial, the users reported that it was by accident and they were just learning how to operate the software.

It was also common in the first day of the trial rounds for the players to create photo tags of people. When the players were asked about these tags afterwards, they said that they were testing the program and showing it to their friends. Since the focus of these tags was people rather than locations, they were hard for other players to confirm. As such, they were not added to the set of by-products. The downside is that these tags took up a slot in other players' daily tag allowances throughout the course of the trial. A flagging system to remove such tags from play could help alleviate this problem. It might also be beneficial to remove tags from the system when they are repeatedly deleted by players. This makes the assumption that the tag is deleted because it is 'bad.' However, it may just be beyond the scope of the players' local knowledge.

Since these junk tags cannot be confirmed, they will never be used as by-products for other systems. There is only one known tag that might get past this barrier, which is to take a picture of text. This was employed by several players when they took pictures of signs, such as those with the name of a building. For example, in the University of Glasgow, where the game was played, there are many buildings which have their name on a sign on the building. This made for an easy location to be found and confirmed. Street signs were also used. However, these are still useful images. Whether they are as useful as less text driven pictures depends on the purpose of the system that wants to use them.

However, it is equally possible that players could simply write some text on a piece of paper and take a picture of it. This could include geographic coordinates or the name of the place the photograph was taken. Again, it is arguable that these images are worthless. But the important point is that the locations in which the were taken could still be found, allowing the tags to be confirmed. As such, they would be considered useful by-products by the current game dynamic.

Software was not the only thing that led to unexpected behavior. Because all the trial phones were of the same model, one user suggested that she might secretly swap phones with one of the other players. This would then allow her to delete that player's tags for the day. EyeSpy only fills free slots with new tags, so the game allows the players to delete tags if they do not know where they are and are unable to confirm them. However, if a player receives their full tag allowance for the day and deletes tags, they will not receive new ones until the start of the next day. This tactic would effectively prevent the other player from confirming any tags that day. However, the trial participant never actually attempted this strategy suggesting that she did not think it would pay off.

A similar idea was attempted by two other friends playing the game together who reported that they had considered swapping phones with the current scoreboard leader (also one of their friends) because he was playing the game more than they were and this might trick him into increasing their points instead of his own. However, they never followed through with this. It is possible that this might have worked because all players in the trial were using the same model of phone to play the game and the Eye Spy interface does not make it immediately clear what player the current client is attached to. In the case of real deployment with people using their own phones, it is unlikely that this would work. Of course, in a real deployment this would constitute stealing and be against the law. This also makes it less likely that such a tactic would be employed.

Another direct approach that was not carried out was to pay the other participants to lose. One player said that he had considered this at the start of the trial but decided that he could not afford the cost required. This strategy would probably not have worked though. In the trial, the participants were paid £10 a week to take part, and the winner was given double that amount. If the game were to run outside of a trial, there would not be such payments or monetary rewards. There would also be a good deal more players. Thus, paying off the larger number of players would become increasingly costly. Also, since there would be no jackpot to win, there would be very little benefit in climbing to the top of the score board without the satisfaction of competition. Even in the trial, the monetary reward was so small that, after paying off the other participants, it would not be worthwhile doing this just to win the jackpot.

One behavior that was suggested by a participant but not attempted, was to create purposely abusive tags. For example, one player suggested they could make a text snippet that read, "I don't like this game." It is interesting to note that such a tag could not be confirmed. Because the player suggested a directly abusive tag without describing a location, the consensus rule would have prevented it from being confirmed and being added to the set of usable by-products. The player who suggested this behavior said that it may have been used to annoy the other players into not playing. Then it would be easier to win because you were competing against fewer people.

However, the annoyance factor to other players could be significant, because this 'unconfirmable' tag has wasted a space in their daily allocation of tags to confirm. A more serious issue could be the use of swearing, sexually explicit photos and text, and otherwise offensive material being used in creating tags. Such a situation occurred on the popular user-generated video sharing website, YouTube.[12] YouTube maintains its community guidelines largely by relying on users to flag inappropriate content. However, the content must be published on the site before users can flag it. This means that at least some people could see the videos first and it is hard to prevent this. Also, when videos are flagged by users, an employee of YouTube must still review it before it can be taken down.[13] A large mass of users conspired to upload as much pornographic video onto the website as possible within a short period of time, hoping that the site would be unable to filter it out fast enough for a large quantity to get through.[14] For the most part, this succeeded. In order to improve the likelihood of success, the uploaders would include up to a minute of content that would satisfy the community guidelines before the pornographic material started. The descriptions and titles of the videos were also kept within the community guidelines to make it harder to track these videos. Since this incident, YouTube has added in a 'Safety Mode' which they advertise as preventing a higher level of objectionable material from appearing on the site, even when it may be within the community guidelines.[15]

Ideally, a flagging system built into the game would also be beneficial here. Even though the consensus rule will prevent the abusive tag from being added to the set of usable by-products, it will still continue to be given to other players to confirm. Should tags be given a lifetime, as was mentioned earlier, this would help with the problem as the tags that were not confirmed would eventually be taken out of the game. However, allowing users to flag

[12] `http://www.youtube.com`

[13] `http://www.youtube.com/t/community_guidelines`

[14] `http://goo.gl/peoyi`

[15] `http://goo.gl/q259B`

inappropriate tags might produce a quicker result. If a certain number of users all flagged the same tag as inappropriate, then it could immediately be removed from the game and no other players would receive it. We are looking to implement this solution into future versions. The Peekaboom game also filters out a known list of inappropriate or abusive words when people are making guesses [57]. However, this only works for words in the blacklist.

Another possible solution to this problem was discussed by von Ahn and Dabbish [6]. They suggested that their game be split into different versions, with each version having a different type of image content. For example, a 'children's room' would not have any objectionable images in it. However, this assumes a prior knowledge of the image contents. This could be solved by initially putting all images into an 'anything goes' room and waiting for labels to be attached. This would allow for some level of automatic categorization into the other rooms.

Players' practical engagement with the rules, and even the language used to introduce and 'frame' the game configured certain expectations about the style of play (e.g., using the name 'EyeSpy' confused some players initially due to the children's game 'eye spy'). By and large, this oriented players toward the production of 'good' photographic by-products. However, in a small number of text tags, a player chose to create riddles that required co-players to engage in some detective work in order to confirm them. Although these tags might have well been more enjoyable for players both in creating them and discovering how to confirm them, they were less useful as by-products, and lacked the recognizability and find-ability that were touchstones for well-constructed 'good tags' for most EyeSpy players. While it may have been more fun for some to engage with this form of tag, the work required of players in locating them conflicted with the need to maximize tag confirmations, which in turn meant that riddles did not earn many points for the creators. Thus, the rules of EyeSpy to some extent discouraged such creative and playful activities. This serves as an example of game rules encouraging one potential style of play, rather than enforcing it; riddles are technically feasible but lose out to a style based on more straightforward recognizability. One can imagine, though, that a determined enough group of players or a slight shift in game rules might lead to a different game style and different byproducts (e.g., EyeSpy's by-product might instead be riddles). This shows that a by-product being 'good' or 'bad' is largely determined by how well it fits the needs of the system, or systems, for which it will be used.

Generalizing from this, we suggest that it is difficult to guarantee that the by–products of this style of game will always be 'pure'. Given subjective tagging and confirmation, players may always potentially find new ways to play, or ways to 'game the system'. Designers can reduce the likelihood of such events but cannot avoid them completely. Careful design and thorough testing should help, but we recommend that designers also consider the ways in which the language used to introduce the game and frame the system will influence the production of particular kinds of by-products.

One element of the game dynamic had an interesting effect. Since the second and third rounds were started without any tags in the system, the initial time spent playing the game only allowed the players to create tags, not to confirm them. Syncing the phone to the game server was left as a manual operation for the players to carry out. The tags on the game server were distributed randomly between the players, but the second player would always get all the tags of the first player to confirm. The third player to sync would have a mixture of the first and second players' tags. The fourth player would have a mix of the first, second and third players' tags, and so on. This means that the earlier you sync for the first time, the more likely it is that the other players will receive your tags. If you are the last person to sync, then no one will receive your tags during the first day of play.

This meant that people who uploaded their tags early in the trial would have an advantage over those who did it later, giving them an early lead. This situation will be combated in future versions by having an automatic syncing feature using the phone's 3G connection so that tags are synced as soon as they are created instead of in a single batch. While this will not alleviate the problem entirely (everyone will most likely receive the first five photo tags and the first five text tags that are created), it will reward people playing the game most rather than those who happen to manually sync first and it will allow the possibility for those tags to be shared amongst the players rather than ensuring that one player dominates. In a public release of the game with more players, each joining at different times, this is less likely to be an issue.

Players who tended to use their full allowance of tag creation each day also tended to gain more points because there were more of their tags in the system, allowing for a greater likelihood that other players would receive their tags. This fits the game's design of rewarding the players who do the most work.

The tags were generally found to represent landmarks and highly recognizable areas. This seems to stem from the game's dynamic as tags which are easier to find will be easier for other players to confirm. Some players said that they started the game trying to make tags which were more challenging to find because they felt it would be more fun for themselves and the other players. However, most players quickly changed tactics when they realized that this was not an ideal way to gain the maximum number of points and win the game. As such, most players quickly adopted the strategy of creating tags that they felt the other players could easily identify and would be willing to confirm. This would ensure that points are received for each tag.

Because the daily routine of the players was in the same geographical area, this acted as a boundary to the game. If players created a tag that was too far away from that boundary, it was not likely that their tags would be confirmed. This is because people tended not to walk too far out of their way to play the game.

The downside to this is that the players who were creating the most tags tended to run out of places to play the game. This is because we prevented players from re-tagging the same location in order to encourage a broader range of tags. Once players had tagged all the notable places in the play area, there was nothing else for them to do. This is directly contradictory to the design goals of the game because we wanted the players who were creating the most by-products to continue to do so.

Even though this proved to be a problem, some players did express that it forced them to be a little more creative with their tagging. They would tag the same notable place from farther away, making sure that other features would show in the photograph. This would let the other players line themselves up with those features in order to determine where the photo was taken.

Ultimately this problem could also be solved by giving the tags a lifetime. It is important that once a tag's lifetime is over, the location should be free to be re-tagged by the player. This should prevent the saturation problem described here.

Since players continue to receive points every time a new player confirms one of their tags, a person who had stopped playing EyeSpy would continue to receive points. It is conceivable that such a person could be hard to overtake in the scoreboard. Other players might consider this unfair because it is not simply beating the high score of a former player. The high score would keep getting higher as well. Removing older tags from play would probably help to alleviate this problem as it would stop players from receiving points for their older tags until they eventually had no valid tags left.

A second issue with the lifetime of tags is their changing nature over time. As the seasons change, a picture of a place might be harder to locate. Statues might be completely covered in snow and be unfindable. It is also possible that buildings could be knocked down, new buildings could take their place and shops and restaurants could change ownership and become something else. All of these issues mean that older tags could be harder to confirm. Also, since users cannot tag the same location more than once, they would not have the opportunity to capture these new developments in the world.

The ESP Game also recognized this issue of by-product lifetime [6]. Once an image was considered to by fully tagged (when all players continually had to pass on an image because the 'taboo' words covered all the good meta-data) it was removed from the game. However, after three months, it was reintroduced with the meta-data removed. This is because cultural differences over time may have changed peoples' perception of the image.

Several times during the trial, there were players who tried playing the game together. Although in each case the players were friends and were partly doing so to be sociable, some of them tried to come up with tactics to take advantage of this. However, the players never successfully came up with a means of gaming the system this way. Other players would meet to play the game together without any intent to 'game the system' because they enjoyed the social aspect of playing the game.

While anonymity was considered an important part of maintaining the consensus rule, some players suggested that they began to recognize styles of particular players. While they did not necessarily know who the player was, they began to attribute certain tags to one person. A thorough test of whether or not these groupings were accurate was not performed. This is because in a public release of the game there would be too many tags and too many players to make this feasible. Secondly, the technique did not allow the players to know who made a group of tags, just that they thought they were made by one person. They could also not say where that person was in the score board so there is no reason to think they would treat those tags differently from any others.

However, one possible exception to this could be that a player might guess the level of play being carried out by another player based on the number of tags they think they are receiving from them. This level might then allow them to guess which score belonged to that player in the score board. They may then choose not to confirm tags from players who were higher in the score board than they were. This is just postulation, however, and no such occurrences were reported by the trial participants. Even in the case where people did attempt such a tactic, they would still be guessing and could never know for sure. In a public release with more players, this scenario would be increasingly unlikely.

Many of the players reported that, although they enjoyed the game initially, they found the static nature of the game structure a little monotonous. This refers to the game having a simple structure that did not change. The users had a limited number of things to do in the game and each of these things was fairly straightforward.

This also led to a problem of being able to overtake other players. In the initial trial, the game was seeded with existing tags so that the game-play would start very quickly because the trial was only one week long. However, in the second and third trials, the game was not seeded. These were longer trials and the experimenters wanted to simulate a game experience closer to a real deployment. However, during the first week of the second trial, one player managed to create more tags than any other player. This meant that there was a good deal more of his tags in the database than anyone else's, making it more likely that it would be his tags that would be assigned to other players to confirm. Had anyone else in the trial played the game to the same extent as this player, this would not have been an issue. This meant that during the first week, whenever one of his tags was confirmed, he would get points as well. But since he was getting the largest number of points from other players, as well as confirming more than anyone else, he created a large lead early on in the game. While the game rules are set up to encourage the person who produces the most confirmed tags, this led to an unusual effect in the second week of the trial.

The player with the large lead barely played the game in the second week because he had reached the previously mentioned saturation point. Some of the other players played harder in the second week in order to beat him, but others continued their same pace. This meant that the leaders tags continued to dominate the database in the second week because no one player was producing enough to overtake. The players trying to win were still confirming his tags in the second week, giving the leader more points. This made it very difficult for other players to overtake the leader, even though one player produced a similar number of tags and the leader was no longer playing to the same extent. Many players expressed that the feeling they could never win was disheartening and made them lose some of their drive for the game. Because of these points of monotony and inability to overtake the leader, several players suggested that having 'bonus' elements in the game would make it more enjoyable to play.

Several of the players also reported that they felt quite isolated when playing the game. Even though players were aware of each other's activities through the changing amounts in the game's score board and guessing who might have made each tag, they felt that an element of direct communication would have been more enjoyable. Most players suggested that a social or team based aspect to the game might have been preferable. A potential solution to this could be to develop more of a community around the game and its results.

Some players purposely made tags that made a logical sequence. This could either be making tags that were along the same road or, in one case, making photo tags of the successive house numbers on a street. One of the examples from the interviews was:

> *"... I just basically walked along Oakfield Avenue, outside where the Stevey building is. So I just took a picture of number 70, picture number 72, picture number 74. And then a couple across the road cause I didn't know how it was picked, if one person would just get all my tags."*

These players had hoped that others would receive the tags as a group and it would make them easier to confirm. However, since the tags for confirmation were randomly assigned, it is unlikely that this tactic would ever pay off.

While it was not done on purpose, another potential for cheating occurred due to a small group of three players being friends during one of the rounds. They also happened to work in the same building at the University of Glasgow. Because they all knew that they were playing, they took some pictures of the inside of the building. This fits with the design of the EyeSpy rules which encourages the players to design their pictures for other people rather than themselves. However, the game makes the assumption that the players will not know each other. In the case where the number of players was small (there were only another 6 players in that particular round of the trial), players that know each other might forget about designing for the larger group. One player who was not part of their group reported:

> *"But you could so tell what were his tags out of, sort of, if you know what somebody's doing, cause they're always in that building they're gonna tag obscure lifts. Like 4 or 8 of them and you're like yeah, whatever [. . .] just pain in the arse that it was always down in engineering bit. It's like, god I have to walk around there. Like a weirdo taking pictures of signs and stuff."*

In actuality, there were only a small number of tags that were made inside the groups' building. However, taking these kind of pictures is fairly obscure to people who do not use that building as shown by the quote above. While this does not seem to be a premeditated attempt to cheat by this group, it shows one method by which a subgroup of players could try and control the scoring. If the number of players in the subgroup was a majority, it might end up excluding those players who are not in the group by making it impossible for them to confirm tags.

A similar problem was how local naming would compare to non-local naming. Because the players of the game are expected to have local knowledge, and they are designing their tags to be easy to confirm by others with similar knowledge, they might not consider things from a world view. One player tagged a building with the text snippet, "qm entrance." To students of the University of Glasgow, which most players were, this clearly means the entrance to the Queen Margaret Student Union. But to non-locals, this might not mean much at all.

One final method of cheating was not employed by any players, but one player had considered the possibility. This was to leave messages within the tags themselves:

> *"… the only thing I ever thought of and I wouldn't class it as cheating was that you could leave messages on this."*

Such a technique could be used to 'sign' the tags that were made. Then, if two or more players were in league with each other, they could agree to only confirm tags that had been signed by other members of the group. While this would be easy for text tags, it would be more difficult for photo tags. However, players could still write some text on a piece of paper and then include it in the frame when taking the photograph. It is possible that such tactics could backfire if other players became aware of it. They might stop confirming tags that had been signed. However, this is all speculation as this tactic was not employed by any players.

In fact, none of the players considered any of their tactics to be cheating. Some even reported that although they had considered ways that they could 'cheat,' they did not consider the methods to be worthwhile:

> *"I didn't play it tactically, sort of from that point of view, like oh I don't want him to win so I'm not going to tag ones that I think are his. I just went and if there was 10 tags I would trying get all the 10 tags every day … "*

The players who spoke about ideas they had come up with for cheating either did not try them or were unable to gain an advantage using them.

Ultimately, no successful attempts at cheating were found in the initial trials. Some players chose to play the game together, meaning that they would end up with similar tags. While this is not cheating, it will produce duplicate by-products which is less useful than a greater number of unique tags.

While many players reported that they could not think of a way to cheat within the game, this discussion has shown that it was possible and would have affected the quality of by-products. As such, it is an important design consideration when making games with a purpose.

3.3.2 – Possible Improvements

Following on from EyeSpy, we investigated ways to address issues such as sociability and saturation. The most obvious way to improve sociability is to make it a team game. For example, teams may find ways to coordinate and combine their tagging, and enjoy the social interaction of collaboration. One approach to dealing with the saturation of popular areas is to have tags fade over time, so that players will have to revisit (and thereby keep up-to-date) the tags in areas they find convenient to play in. Alternatively, the pay-off for players who explore or 'open up' new areas for play could be increased. Areas that have not been tagged enough could be marked as being worth extra points, or tags could be distributed only for a given area so as to convince new players to play there. A difficult issue here is how to keep the game balanced despite changes to its structure. For example, a player who has saturated an area and so built up a lead in the game may object to new and distant areas opening up that may draw players away from confirming his or her tags in the older area.

Perhaps a more important issue is to do with the type of by-products produced by these games. Because the games themselves do not answer direct requests for by-products, they are designed to return a broad result set. While this is a good preemptive measure when the exact needs of other systems are not known, in more specific or unusual requests the result set might not be able to return suitable data. A solution to this would involve directing the players in some way toward creating by-products that were specified by other systems.

Another possibility for improving the quality of by-products would be to use multiple systems that contribute to the same data set. This could broaden the range of tags by using the strengths of the different systems in gathering different sorts of tags.

A further area of interest is how the level of play changes over time. Does the amount an individual plays the game get more or less over an extended period of time? How long will users play the game before they stop altogether and is there a way to prevent this?

The games with by-products that have been created thus far have been relatively simple. However, the by-products have also been relatively simple. It may be possible that more complex games could be constructed to solve more complex problems. For example, it has been shown how an artificial virus spreading through a large scale multi-player virtual world could be used to model such outbreaks in a controlled environment with examples of how people might behave in such situations [74]. It seems feasible that such an environment could be used to generate a complex dataset of by-products as well.

Sociability might also be improved by creating more of a community around the game, rather than changing the game rules and dynamics. One potential method of doing this will be discussed in the next section.

3.3.2.1 – EyeSpy User Website

One possible solution to the problems of junk tags and the isolated feeling of the players would be to construct a user website for the players of EyeSpy. The EyeSpy user website could allow EyeSpy users to sign in and see their current score and which tags are making them the greatest number of points. In addition, users could be allowed to browse the tags they have received and confirmed.

A 'thumbs up or thumbs down' system to vote for tags could be used, with people receiving points for contributing. Each tag could have a commenting system attached to it as well with a similar voting mechanism. A map showing the location of the tag could also be displayed to put it into some sort of context. Nearby tags which the user has received and confirmed could also be displayed to help with this. A user would always receive points for these votes but would receive more for voting with the crowd and less for voting against it. This means that if more people vote a tag good than bad and a new user agrees with this vote, then she would receive more points than if she had voted against them. These points would be added to the user's total for the EyeSpy game.

In each case, users would not know how popular a tag or comment is until after they vote on it. This could also be used as a means to determine which tags should be by-products. In addition to requiring a certain number of confirmations within the game, tags would need to reach a certain level of popularity on the website to be used as by-products. The most popular comments may then be displayed with the tags. This would hopefully filter out some of the junk tags and could also have the benefit of providing an ordering mechanism for the tags when used as a by-product. That is, the tags for a location could be ordered by how popular they were.

This could lead to more quirky tags being included as by-products. This is because locals might find things more interesting than would be expressible through the game alone. This additional local knowledge could prove useful to, for example, a tourist website. It must be remembered that tourists might want this information in addition to, or instead of, the more regular tourist information. There is no reason to assume that tourists have not been to the location before or would only want the regular tourist information. They may also want more obscure information that only locals would know.

This voting and commenting system could also foster a sense of community to the players. This could alleviate the problem of players feeling isolated.

The EyeSpy user website could also have a 'hot spots' map. This would be a map of the playing area with a subsection marked as being 'hot.' When users create tags within this area, they will be worth more once confirmed. The hot spot location could be updated on a daily basis. The hot spot location would be determined by places in the game area which have seen the least amount of tag creations in the last few days. This would hopefully encourage the EyeSpy users to focus their efforts on areas which had fewer tags.

In order to make this fair to all players, hot spot areas would need to temporarily reset the users' inability to make tags in places where they have done so before if they were in the hot spot. However, they would still only be able to make one tag in a location during the reset period. After the hot spot is no longer in effect, the original conditions for taking tags will come back into effect for that area. The temporary removal of this barrier in the game-play would encourage as much new data as possible for the hot spot area. However, removing the barrier entirely would allow players to continually create the same tags over and over. This will create duplicate images which defeats the purpose of creating as many unique and useful by-products as possible.

While it seems clear that an EyeSpy user website could provide many benefits, a different concept was used to try and solve EyeSpy's problems. Due to time restrictions, only one solution could be built and this solution will be discussed in the next chapter. However, the points made here show that it is one solution to some of the minor problems with the system.

3.4 – Conclusions

In this chapter, a mobile game with by-products was created and tested. The game involved user generated tasks in order to alleviate the monotony seen in some other games with by-products. By bringing verification into the mobile game, we were able to produce one of the first full examples of human computation in a mobile context. The design of the game was based on mobile phones which had location sensing technology and the ability to take photographs. The game, called 'EyeSpy,' involved users tagging areas with textual and photographic descriptions. These descriptions were swapped between the players anonymously. Players had to then confirm the tags by trying to find the location in which they were made from nothing more than viewing the textual and photographic descriptions. This had the additional benefit of both creating and labeling geographically referenced photographs. However, the labels were attached to the location of the photographs, not the photographs themselves. This meant that a label could apply to any photograph from the same location and that each photograph could be described by any label from that location. This could be useful in tourist websites because a search of the text labels could return areas rather than specific photographs, allowing the users to get a feel for an entire area rather than just photographs that were relevant to a specific landmark. In each case, these by-products were proven to be recognizable because other players had successfully carried out the confirm step. This recognizability would also be important if the tags were to be used for a tourist website because it would help them navigate an area with which they were unfamiliar.

The common design rules from Sections 2.3.2.1 and 2.3.2.2 were applied to the design of EyeSpy. These helped to ensure the game was enjoyable to play and would produce good quality by-products. In addition to showing the general design techniques involved, we showed that there are issues which were specific to our game and to the mobile field, demonstrating that there will likely be unique design rules for each game with by-products. The general rules show that designing games to encourage the creation of 'good' by-products is important, as is designing elements that discourage the creation of 'bad' by-products. It was shown that making assumptions about the players or the styles of play can lead to mistakes in game design and that keeping the game dynamics more flexible in early designs can lead to unseen benefits that can be carried through to later iterations. We also discussed the importance of creating additional game rules to keep the games fun and how it might be beneficial to implement a means for the players themselves to monitor and deal with people trying to sabotage or 'game' the system. If designed correctly, these elements could allow for better by-products from games designed to harness human computation. We believe that games with by-products are a particularly good way of harnessing human computation and that determining what makes the current, simpler games successful will be important in the design of more complicated games that might create more complex by-products.

The EyeSpy trials have shown that human computation can be successfully applied to the mobile domain. This is a good domain for human computation as using local knowledge to get recognizable photographs of an area would be difficult to achieve with automatic means.

Our trial of the game demonstrated players' orientation to certain navigational qualities of the tags they created (i.e., recognizability and find-ability), as well as revealing the careful design balance between game rules, how rules work out in practice, and the character of the by-products that are produced. We also reflected on the human issues of designing for fun, motivation, maintaining interest and accountability. The game used Wi-Fi for positioning and to connect to the Internet. The game successfully produced the intended by-products and the players reported that they found the game enjoyable.

However, the game had a saturation problem where a game area would become 'used up' and people who were winning the game would find it too difficult to find places they could play without going too far outwith their normal routine. It was also shown that the game dynamic made it too difficult for players to overtake the leader in the game, even when the leader stopped playing. A last point was that, even with users generating the tasks, the game still became monotonous after a while for some players.

The trials also raised some issues about the way that cheating might affect by-products. While it did not have a significant impact on these trials, it shows the importance of designing a mechanism that makes cheating hard or non-beneficial.

The game dynamic of EyeSpy also led to broad tags being produced. This is because the players themselves created the tags and wanted to make them as easy as possible for others to find. However, since EyeSpy is designed to produce useful by-products, this may not always be desirable. EyeSpy has no mechanism to deal with specific requests for images. For example, there is currently no way for a second application to tell EyeSpy that it wants a picture from a specific location when there are no existing pictures in the database. If EyeSpy was the source of all that application's pictures, then having such a mechanism would be important to addressing this situation.

It was suggested that multiple systems working together in this fashion might make the by-products more usable. Another potential improvement would be to include bonus elements in the game to help players overtake one another. It was also suggested that giving a lifetime to tags could help with the saturation problem. Another solution was to make certain areas open to tagging, even if they had been tagged before. A solution was presented to make the game more sociable by introducing a player website.

By creating EyeSpy, we have demonstrated how to apply the common rules of web-based games with by-products in the mobile domain and incorporated some new ideas to deal with the challenges of mobile environments. We have also highlighted problems in that design which may need to be addressed in future systems. This has helped in addressing ***RQ2***. The creation of EyeSpy has also provided us with the ground work necessary to begin addressing ***RQ1*** by creating a game in which users generated the human computation tasks. The lessons learned from this will help inform ways in which we might augment the game to match the needs of another system to produce more specifically required by-products.

These augmentations to EyeSpy and the additional of a second system which can make specific requests of the game will be the subject of the next chapter.

Chapter 4
Mutually Reinforcing Systems

The EyeSpy game has now been introduced and some of its strengths and weaknesses have been discussed. While EyeSpy was successful in achieving its goals, there was room for improvement. In particular, the game had problems that made it less fun over time and the by-products produced were fairly broad. In this chapter, we will discuss some augmentations that were made to EyeSpy, as well as discussing the introduction of a second system that was used to work together with the game to help address these problems.

In EyeSpy, producing a mobile game allowed the designers to take advantage not only of the players' mental capabilities, but also of their physicality. EyeSpy was designed to be a pervasive game that can be played over a geographical area for a long period of time, rather than at a desktop. The game necessitates that players are mobile. The game did not demand intense bursts of play with long individual sessions: players could fit the game into short periods during their regular day and their movements became part of the game dynamic. EyeSpy had the ability to fit into a player's everyday life, the ability to take advantage of a player's local knowledge and the ability to take advantage of a player's location. The goal was to create a pervasive game with by-products and to ensure that the by-products were of the highest quality.

One possible use of the by-products was in a location based image search engine. Location based searching has been shown in the IDeixis system [118]. This used mobile camera phones to perform location based searches by taking photographs. The photographs were used to find similar images on the World Wide Web and then the matching web pages which contained the images were returned to the user as results. A similar system is used by Google

Goggles which allows the user to take photographs with a mobile camera phone.[1] These photographs are then sent to Google's servers and automatically analyzed for identifiable information, returning relevant search results to the user. In both these cases, an image is used to produce location based information that is presented to the user.

These systems show that the power of Internet services can be used to work with mobile information. This presents a possible means to approaching the problems with EyeSpy by including a stronger Internet focused component. This led us to consider creating a tourist website which was called 'Realise.'

The design of the Realise website was influenced by IDeixis and Google Goggles. However, rather than use photographs to return information about a location, Realise uses location based information to return photographs.

Realise is a website that allows users to browse the geographically referenced photographs produced by EyeSpy, see their location on a map and search them based on nearby text tags. It also allows users to make requests for new photographs by describing their desired photograph to the EyeSpy players. These requests appear in the game as a special tag type that requires the players to create a photograph that matches the description they have been given.

Realise is essentially the location based image search engine that was discussed in Chapter 3. However, we have chosen to frame the search engine as a tourist website for pre-visiting and post-visiting locations, as described by Brown and Chalmers [42]. As such, the Realise website was intended to let users search for areas so that they could get a feel for them, either before or after their intended visit. Pre-visiting using the Realise website allows tourists to see what is nearby in the locations they plan to visit and what the locals consider recognizable. This could help them prepare for their visit or perhaps to help them navigate an area they had not been to before, especially in showing the geographic relationships between landmarks. Post-visiting could be used to further explore the area beyond what was achieved at the time of the visit, or it could be used to acquire additional photographs of a place after a visit.

EyeSpy's by-products might be particularly helpful for these purposes because the content is

[1]http://www.google.com/mobile/goggles

created by people using their local knowledge of an area to determine what is recognizable. A common problem with electronic tourist guides is that they do not pick out things that are directly helpful to orient a user and generally include some sort of blanket coverage [119]. But because the images and descriptions of places in EyeSpy have been confirmed as recognizable for the locations they are attached to, they can be used to orient users in a more selective way. This is, of course, assuming that orientation will be useful to tourists. It has already been discussed in Section 3.3.1 that the usefulness of a by-product is largely determined by what it is used for. While a system may be designed to produce by-products for a particular purpose, this does not mean that the by-products which do not match the design are not useful in some other way.

Using user-generated information for tourism has already seen some success on web sites. Dopplr's social atlas allows tourists to share their own travel tips and information for places they have visited.[2] This means that you get different viewpoints and experiences about a place rather than a guide book written by one person. However, in general, this means that your information comes from other tourists rather than locals. While this is not a disadvantage, there may be interesting places to visit that are not on the normal tourist path and which only locals are likely to know about. The difference between what tourists find important about an area and what locals find important can be seen in on-line 'heat maps.'[3] These are based on the period of time over which people take photos. For example, if people take photos of an area all within the same month, they are probably tourists. But if they take photos over a period of many months or years, they are probably locals. While it cannot be proven if this is an accurate way to identify locals and non-locals, it is interesting to see the disparity between users who photograph a place over a short period of time and those who continue to photograph it for a longer period. Assuming that the maps do show a division between short term and long term occupants of places, these maps could let people find or avoid common tourist spots depending on how they want to spend their time. Some people may want to know where the popular tourist spots are, and others may want to go off the beaten track but still see something interesting.

Brown et al. demonstrated a system called 'George Square' that allowed a mobile user and a web user to co-visit a place, despite the differences in their physical locations [120]. This

[2] http://www.dopplr.com/socialatlas

[3] http://goo.gl/rqi5g

demonstrates that physical location is not a prerequisite for having a shared experience and supports the concept of pairing a tourist website with the EyeSpy game. In such a pairing, the web users can share the experiences of the mobile users by browsing and searching the photographs and textual descriptions of the various locations. Bell et al., talking about the same 'George Square' system, also showed that the flow of information need not be unidirectional and that web users can help direct mobile users to points of interest [121]. This suggests that taking a similar approach and using a tourist website to direct the EyeSpy players would have similar success.

Using systems together in this way might lead to a general improvement in games with by-products. A system that is good at producing one kind of data can share that strength with other systems, or two systems that produce the same type of by-product can be used to form a larger dataset. If a system is bad at gathering one kind of data, it makes sense that it should be able to request that data from another system which is better suited to the task.

This might be a missed opportunity in some of the systems mentioned in Section 2.3.2. By working together, these systems might have benefited from each others' strengths. For example, Peekaboom [57] uses an input dataset of tagged images to allow it to produce its own by-products. Peekaboom's output is a log of the pixels in an image that are associated with a particular word. The tagged images which are used in Peekaboom are the by-product of the ESP Game [6]. In this sense, the ESP Game is used to create game content for Peekaboom, even though the ESP Game was not designed for that specific purpose. This set of tagged images is something which Peekaboom could not have produced on its own. However, Peekaboom could have also provided a useful function to the ESP Game. The ESP Game's primary purpose was to create a dataset that would power a search engine for images. Since Peekaboom gathers the percentage of an image's pixels that refer to a given word, a benefit could have been provided to the search engine. By determining what percentage of an image matches a search term, the results could have been sorted by relevance. By outsourcing this aspect of gathering data for an image based search engine, the ESP Game would be an even stronger system, reinforced by this additional data.

Peekaboom also has the ability to let users pass on an image. If enough users did this for an image/label pair, it might suggest that the pairing is not as good as it should be. However, currently Peekaboom does not communicate this to the ESP Game. If this communication had been possible, the ESP Game could have concentrated more users on these images to produce better labels. This would then improve the quality of the dataset and make the search results better in the image search engine. And, since these image/label pairs are used by Peekaboom as well, it would mean that Peekaboom's game content would be better and less people would pass on images because of bad labels. Since the Phetch game [58] also used the ESP Game's results as part of its own operation, these effects would feed into that system as well. More accurate by-products from the ESP Game would lead to Phetch having better input.

This demonstrates that if the systems worked more closely together then they could benefit from each others' strengths. Peekaboom could request that the ESP Game provide it with better image/label pairs when necessary and, in return, it could supply the ESP Game with relevance percentages for image/label pairs. The ability to communicate direct requests and provide reinforcing data so that each system can benefit from the strengths of others has led to the idea of mutually reinforcing systems.

Mutually reinforcing systems have the ability to receive and act on requests and the ability to make requests of each other. In the scenario that was just described, the ESP Game and Peekaboom would become mutually reinforcing systems. By working together to improve the efficiency and quality of by-product creation, they would each do a better job of achieving their goals.

The loop that they would form, where each system can make requests of the other, as well as responding to requests and sharing data, is a strong element of mutually reinforcing systems. This loop is the reinforcing element that allows the improvement of function. However, the image search engine that uses the dataset from the ESP Game would not be part of the described loop with Peekaboom. It may make sense for the image search engine to join the loop or to form a separate loop with the ESP Game. This would make the ESP Game part of two separate mutually reinforcing system loops, which might make more design sense that creating a single larger loop.

The reason for this is that mutually reinforcing systems are, by their nature, coupled to other systems. In software maintenance terms, coupling is not a good thing because it makes code hard to maintain and creates strong dependencies between systems, making them harder to replace in the future. However, the negative effects of coupling can be minimized by limiting the complexities of the loops. The simplest way to do this is try to reduce the number of systems that form a loop. The technique of sending and receiving requests also limits the level of coupling as it allows a system to be replaced with one that is capable of handling the same requests.

It should be noted that if the image search engine did form a mutually reinforcing system loop with the ESP Game, it would probably be of benefit to both systems. The search engine could allow its users to say whether an image matched the search criteria they wanted. If it did not, then a request could be sent to the ESP Game for a better label. The ESP Game could then remove that text label and concentrate users on that image to produce better labels. However, the bad label could still be included in the 'taboo' words of the ESP Game. These are words which the game prevents the users from submitting. This feature was designed to prevent users entering the same labels again so that the users would enter more diverse labels. However, in this instance, it would prevent the users from entering the same bad label again. The ESP Game could then create a mode for more challenging images (those for which the search engine users said the existing labels were poor). If the users were making poor labels for these images, it would suggest that it is harder to create good labels for them. This more challenging mode could make the game more fun for experienced players. In this way, both systems benefit. By answering direct requests for by-products, this should also eliminate another problem of current games with by-products, which is their production of fairly broad by-products, rather than specifically needed ones. A diagram showing the proposed mutually reinforcing system loops and their interactions can be seen in Figure 4.1.

Another example could have been seen from the Treasure game, as discussed in Section 2.3.2.4. If there had been a required by-product of mapping Wi-Fi coverage (something implicit in the game's design but not recorded or used), then the server which acted as the treasure chest and also allocated where coins were placed could have concentrated coins in areas where more information was needed. This would have forced users into those areas and would have the effect of better mapping the Wi-Fi coverage. As part of a mutually reinforcing system, there may have been a second application which displayed Wi-Fi coverage to a user. If the user wanted to know about a different part of the map or have an existing part defined with better granularity, this might have fed back into the Treasure game as

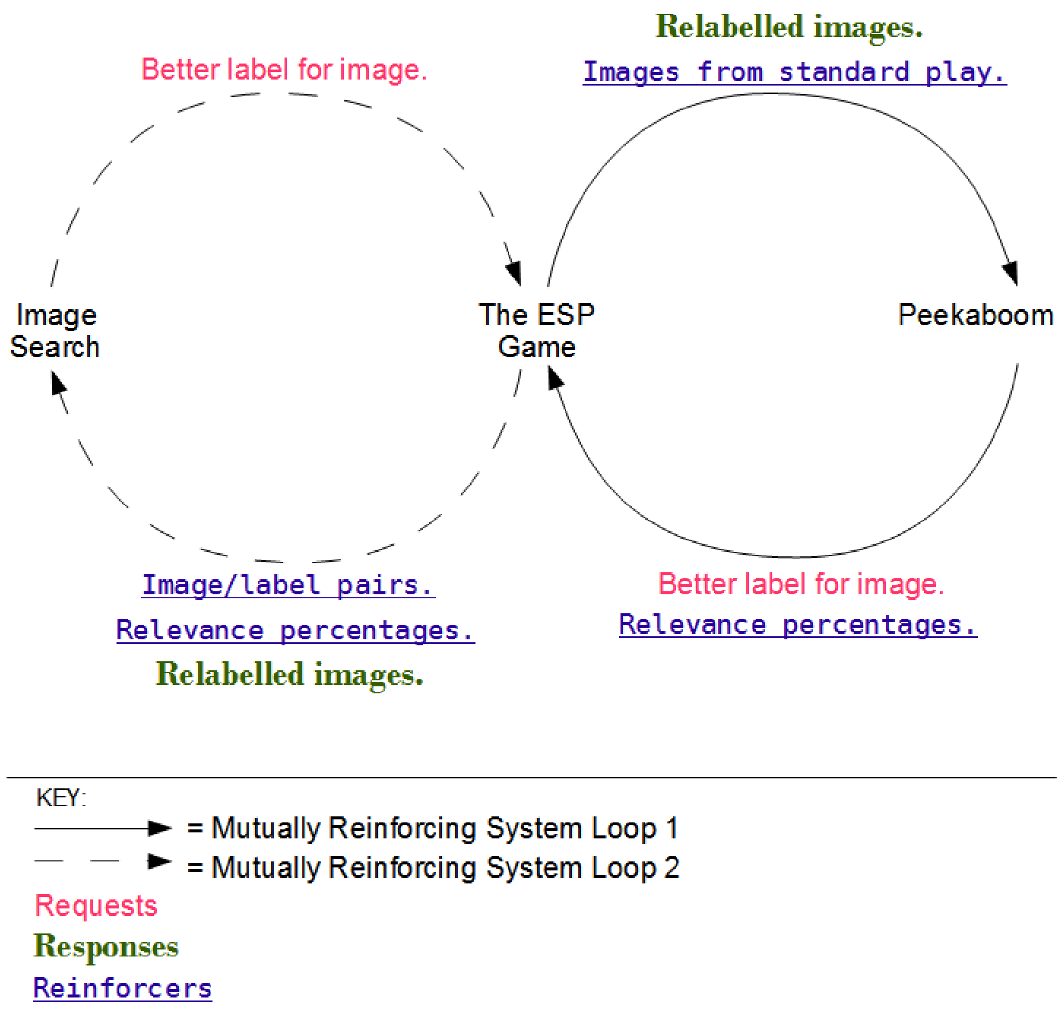

Figure 4.1: Two proposed mutually reinforcing system loops and their interactions. The requests, responses and reinforcers are indicated by the corresponding fonts shown in the key.

well.

Feedback loops of this nature are also seen in human constructed systems like law. A good example of this can be seen in the game 'Nomic,' as proposed by Peter Suber [122]. This game is based on a set of initial rules where the point of the game is to propose and pass new rules in an effort to allow you to win the game rather than the other players. Of course, even this basic principle can be changed during play. By having the players do the human computation of creating the rules, the system in Nomic can become very complex, but is still able to maintain its stability. In a sense, the players are all reinforcing the system of rules, but are also changing it in ways that cannot be predicted or controlled. Games with by-products have already been discussed as being well suited to changeable environments requiring high levels of adaptation. However, players may adapt in ways that do not suit the original system design and the required by-products. Nomic demonstrates that it is possible to place restrictions on a system without making it less stable. This opens the possibility for Realise to restrict game-play to fit its own needs, but still maintain a stable game dynamic. While the EyeSpy players adapt to their environment, the Realise users adapt to the EyeSpy players and encourage changes in the game dynamic to ensure that the required by-products are created.

As has been said, there are two main problems with the EyeSpy game: the produced by-products are broad and are not guaranteed to be those which are needed, and the game-play becomes less fun over time due to monotony and to difficulty in overtaking other players. Mutually reinforcing systems try to tackle these problems by allowing the final system which uses the dataset to make direct requests of the games so that specific data can be gathered when needed. It is expected that these requests will generally represent holes in the dataset that the game is not filling on its own. Thus, game-play based on filling these holes should provide a different experience for players because there are new types of task which the game did not have before.

In order to explore this idea, EyeSpy was augmented and a second system, Realise, was created. For this to work, EyeSpy needed to allow the Realise website to make requests for specific information from the game. EyeSpy could then turn these requests into game elements to encourage the users to gather the required information.

It was hoped that these design changes would address the monotony of the game by having an additional tag type that would be random, and not guaranteed. Since these tags would need to be given priority (so that they can be sent back to the other system as quickly as possible), they would be worth more points and have a time based points system associated with them. This would mean that the quicker a player dealt with a priority target, the more points he or she would be likely to receive. The variable nature of the points might help alleviate the problem of users having difficulty overtaking the leader. Also, because it was important that these pictures be returned quickly, the limitation of not being able to re-tag a location was removed for this tag type. This is still fair because the users themselves are not choosing the location, they are just answering a request. This would go some way to alleviating the saturation problem as well.

Lastly, it was hoped that these design changes would allow other users to make specific requests of the EyeSpy system so that the exact information that was needed could be gathered. This means that not all of EyeSpy's by-products need be as broad.

This chapter will detail the initial work that was carried out on Realise and the augmentations that were applied to EyeSpy, as well as discussing some of the outcomes from that work.

4.1 – System Design

In Section 3.1, we were able to design the EyeSpy game around a set of rules that were taken from the discovery of common elements found in existing games with by-products. The results of the trial show that this method proved to be successful for the most part. However, in constructing the Realise website we did not have a strong model to follow because such a system does not seem to exist. As such, the design of Realise was based on satisfying the needs of its users as well as to try and solve specific problems in EyeSpy.

As has already been discussed, Realise was intended to be a website for pre-visiting and post-visiting locations. This takes the form of allowing users to search for a location then photographs of that location are returned. However, it is possible that several locations will match the search criteria and so photographs from all matching locations will be returned.

While it might seem sensible to return just locations rather than all the photographs within them, it is very possible that a photograph might be in the right area, but not of the landmark which users are searching for. If only locations were returned, this would mean the users would not find what they were looking for.

The photographs in Realise are exclusively taken from EyeSpy. This ensures that the photographs within Realise have been proven to be recognizable by players of the game. It was hoped that this selectivity of photographs would be beneficial to tourists because it would show only places that were recognizable. It was also hoped that the relatively impersonal nature of the EyeSpy photographs (the photographs concentrated mostly on recognizable, non-transient objects rather than people or situations) would be of benefit to tourists as well. This may not always be true, of course. For example, if there was some sort of festival, tourists might want to get a feel for it, but Realise's focus was mainly on exploring areas, not activities within them.

In addition to the photographs being taken from EyeSpy, the textual descriptions from the game are what power the search system. When users enter search terms, all the textual descriptions from EyeSpy are searched and when any of them contain one or more of the search terms, they are returned as correct results. Since each description and photograph in EyeSpy is geographically referenced, we can then return all the photographs that are in the same location as matching descriptions. A results page showing the basic Realise interface can be seen in Figure 4.2.

The default ordering of the results in Realise was by relevance to search terms. This mode of operation was intended to return photographs from the location that were most relevant to the search terms. The location descriptions were compared with the search terms and, if there was more than one term, descriptions that contained a larger number of them would be returned first. If descriptions were found to contain the same number of search terms, then those that had a fewer number of other words were returned first. While further optimizations might have made this a more powerful feature, the priority for the amount of development time available was on returning valid results and it was believed that this solution would work well enough.

The second ordering option would simply return the photographs in the order they were

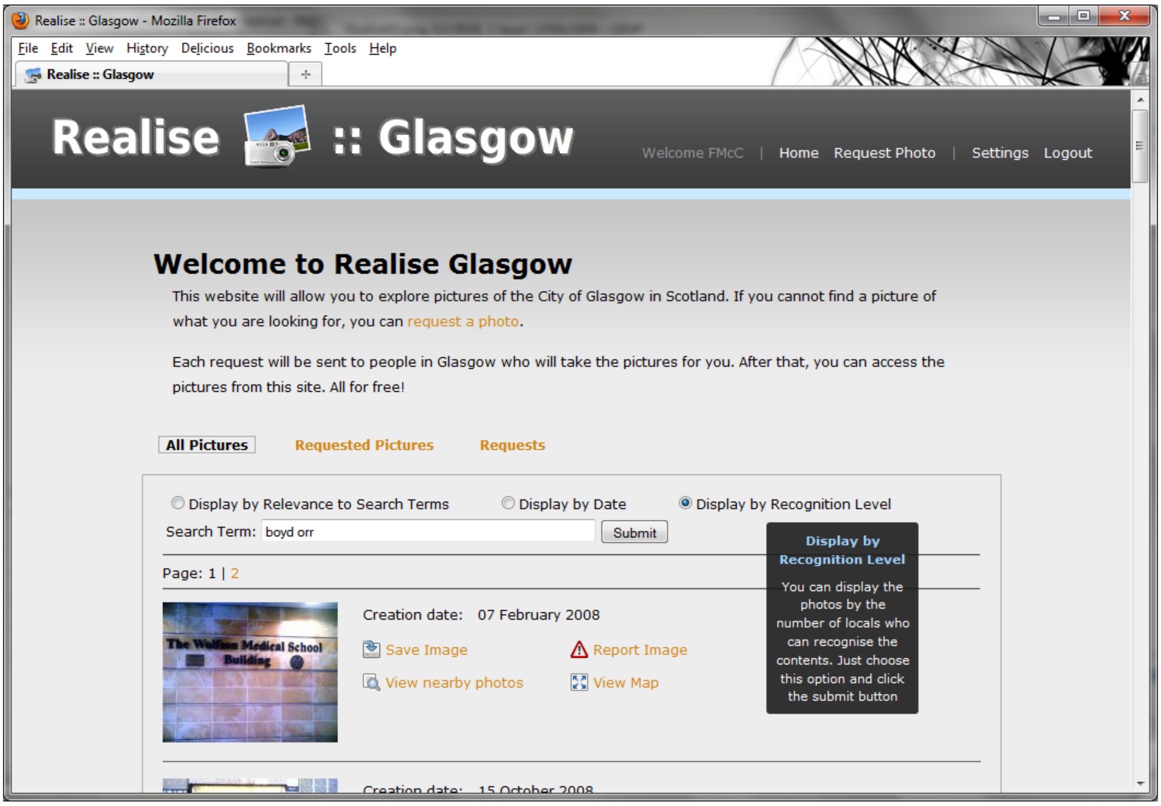

Figure 4.2: The main results page for the Realise website showing results for a user entered search. For each result, users have the option to see the full sized image, save the image, report the image, see the image's location on a map or see photos that are near it. They can also order the results by relevance, date or recognizability (the number of times the image was confirmed by EyeSpy players).

created, with the newest being returned first. The photographs on the Realise website always have their creation date displayed along with them.

The third ordering option would return the photographs based on the number of times they had been confirmed by the EyeSpy players. This ordering was called 'recognition level' because it was believed that the more times a photograph had been confirmed, the more recognizable it must be. An example of this effect can be seen in Figure 4.2. Although the search is for, "boyd orr," which is one of the buildings in the University campus, the first image returned is for the Wolfson Medical School Building which is across the road from the Boyd Orr Building. However, the Wolfson Medical School Building photograph has been confirmed more times than any photographs of the Boyd Orr Building. Because of this, the recognition level ordering has returned this photograph first because it is in the same area but more more easily confirmed by the EyeSpy players than photographs of the Boyd Orr Building.

Unlike traditional search engines, Realise allows the users to enter a null search term, and this is what is presented when a user first visits the website. This simply means that nothing is entered for the search term and all photographs in the system are returned, ordered in whichever way is currently selected. This has the benefit of allowing the users to browse the photographs for interesting locations if they have no specific search term in mind. In some senses, the search terms act more as a filter, reducing the number of photographs returned from the entire set.

When photographs are returned in a list, the users have several options open to them for each one. The thumbnail that is presented can be clicked to reveal a full size version of the image and the users can choose to save a copy of the photograph to their computer. Users are also able to see the location of the photograph on a map.

An important option is the ability to report images that contain inappropriate content. While the agreement and consensus rules of EyeSpy should prevent 'bad' tags from being included, as discussed in Section 3.1.1, these may still allow tags that will be 'bad' for Realise. For example, in Section 3.1.1 we said that a 'good' tag in EyeSpy was one that was well chosen to describe a location such that others could find it. However, this is a very general description and there may be specific exceptions to it which cannot be predicted.

For example, an EyeSpy player may take a picture of a well known landmark which just happens to have some graffiti on it of offensive language. This may allow the location to be confirmed within EyeSpy, but it may still be considered inappropriate by Realise users. Rather than try to predict all the unlikely possibilities, it makes more sense to have a fail safe where the Realise users can report inappropriate content. This use of human computation to add another check to the validity of by-products will strengthen the data set produced by EyeSpy. When a user reports an image, it is immediately removed from their view of the results in Realise. However, it will still be viewable to other users in the system until a threshold is reached. When this threshold is reached, the photograph will be removed from the system entirely. Although Realise is not a game, this can be considered an application of the consensus rule.

A further possibility could be to alert the EyeSpy game to the inappropriate content. The game rules could then be augmented to discriminate against the creation of such tags by negating the points gained from them or by applying some other penalty to the players who created them.

A novel option for each photograph returned by Realise is to view nearby photos. Selecting this option will return a new page showing other photographs in the same location. In essence, this is the location view to allow users to get the feel for a place. This allows the users to distinguish which photographs returned from a search are for the same location, rather than being from different locations which may have matched the search terms. The nearby photos option works by using the selected photograph as a base point, then returning all photographs within a specified radius. While this would be possible with other geographically referenced photographic databases, the selective quality of the EyeSpy database allows Realise to present only recognizable images from the location.

A last major feature of Realise is the ability to request new photographs. In the case that the existing dataset from EyeSpy does not cover the needs of a Realise user, a new photo can be requested from the game. To request a photo, Realise users simply enter a description of what it is they want a picture of. This is sent to EyeSpy and incorporated into the gameplay. When a photograph is returned by the EyeSpy players, the Realise user who requested it will receive an email to inform them of the success. However, photographs will not be returned until they have been validated by EyeSpy. The returned photographs are displayed in their own sections of the website, unique to each user. In order to give all the Realise users a chance for their requests to be satisfied by EyeSpy, requests were restricted to ten per day for each user, and this was reset at the start of each day.

The designers settled on ten requests per day because it was believed that less than this amount would be unhelpful if a new area needed to be discovered for pre-visiting. However, it was believed that more than ten requests per day would flood the system and prevent everyone from having their requests satisfied.

The first section to display the returned photographs is almost exactly the same as the section for displaying all images. It has all the same features and options, but only gives results from the set of returned photographs for the current user.

The second section to display the returned photographs is simply a list of all the requests that have been made and the photographs that have been returned in response to each one. While the photos have the same options as the other sections (view nearby photos, view map, report photo and save photo), the results in this screen cannot be searched or ordered. This decision was largely due to time constraints rather than technical restrictions.

It should be noted that the design of Realise was not to create a great search engine, but one with novel features. Had time not been a factor, the results of searches based on the same data might have been improved with more effective algorithms but no time was available to research this complex subject as it was not considered a priority for answering the research questions.

In order for the 'request photo' feature of Realise to work, some augmentations needed to be made to the EyeSpy game. The first change that was made was to change the name of 'tags' to 'targets.' In interviews with participants of the initial trials, it was found that people found the term 'tag' confusing. As such, it was changed to something that the designers felt would be more easy to understand. As such, the interface would now show the options to 'create targets' and 'confirm targets.'

A new target type was also added to the game in addition to the existing photo and text targets. This target type was called a 'priority target.' This name was chosen to encourage the users to give priority to these targets over the others in the game. It was felt, since the Realise users were waiting for these specific photographs to be returned, that the game should encourage the return of these photos faster than the existing target types.

The priority targets were shown to the users as textual descriptions (similar to the existing text targets). However, during the confirm step, users were asked to take a photograph that matched the description. Although this was presented to the users as a confirm step, it would always be successful, unlike confirm steps for the other target types which required users to be in the correct location. This is because there is no way for the game to know if the user is in the correct location. This potentially means that the photograph could be taken anywhere and of anything and the confirm step would still allow it. Obviously, this is not an ideal solution. While it may have been possible for the Realise users to have defined where they wanted the photograph to be taken using a digital map, such a solution would prevent the EyeSpy users taking advantage of their local knowledge to determine the best spot to take the photograph. The designers felt this ran in contradiction to the spirit of the game and, as such, the common rules of games with a purpose were applied in order to use a human computation solution to this problem.

Payment Through Entertainment

Because the Realise users were waiting for their requests to be fulfilled, the designers wanted to encourage the EyeSpy players to confirm new priority targets as quickly as possible. Even the choice of name and placement within the EyeSpy interface was intended to encourage users to address these targets before anything else.

A second means to make the priority targets popular to the players was to make them worth more points than the other target types. The system for achieving this worked in three ways.

Firstly, the points for confirming a priority target were based on the time from when the player first viewed the target description until it was confirmed. The quicker a player confirmed a priority target, the more points it would be worth. The scale for diminishing the worth of priority targets was not linear, with the number of points a priority target was worth dropping quickly at the start, then more slowly until 24 hours had passed.

The second means to making the priority targets worth more points was to ensure that even after 24 hours from the first viewing, priority targets were still worth 50% more than the other two target types.

The third aspect to making the priority targets worth more was to do with the photograph that was produced during the confirm step. This photograph was put back into the game like a regularly created photo target. This was necessary to ensure human validation would occur and determine if the photograph could be included in the regular results for the website. However, a second benefit of this is that players would receive confirmation points whenever their photograph was confirmed by other players. This ensures that even though the photographs are designed for specific website users, they still must be recognizable by the other players.

While priority targets are worth a lot of points to the players, the players will not immediately receive these points when they go through the confirm step. This is because we require other players to independently select the same location for that priority target before it can be confirmed that this is the correct location. We also need to ensure that the photograph itself is findable by other players to ensure that it is not just a bad photograph in the correct location. Only once both these steps have been achieved will photographs be sent to the website. At this point, the Realise users must decide whether or not the photograph matches their description. If so, they can accept it and the EyeSpy players will finally receive their points. However, in order for players to keep track of how many points they are due, a separate score is shown to them, marked as the 'potential score.' This lists the points that the players are waiting to receive assuming that they pass the validation steps.

The Realise website allows users to make requests for photographs when the existing dataset does not satisfy their needs. This most likely happens when users search for an image and do not get the results they want. They can then request that better results are produced. These requests can be very specific, but they can also rely on local knowledge or human decision making, such as "a tree" or "the nearest subway to Glasgow University." In this sense, the website is no longer just an image search engine with the option to request that 'holes' in the dataset be filled. An entirely new function has been added where labeling of images is descriptive in a subjective way. This ability would not have been possible without being part of a mutually reinforcing system loop. This is generally avoided in games with by-products because objective data is considered more valuable (leading to agreement rules between randomly paired players in order to create such objectivity).

However, it is not always the case that objective data is more valuable. In this instance, where the person who requires the photo is the one making the request, we can see that there is a need for subjective opinion if they have asked for it. This can be seen even more clearly when the Realise user requests something like, "the best pub in the west end of Glasgow." However, this need not necessarily be included in the general search engine until it has also been confirmed by other players. This is an important design point. While the Realise user may be happy with a photo because it fulfilled their particular needs, that does not mean it is labeled in a useful way for other users. The existing game mechanic can be helpful in this instance. Since the photo that was taken for the priority target had location information, it can still be used with the existing text targets for that location. However, we still need to know if the image is generic enough to be part of a general search for that area. This can be achieved by putting the image back into the game as a regular photo target. This also means that players who make priority targets will have the opportunity to get points from the photo a second time when it becomes a photo target. If enough players confirm the photo as a photo target, it means it is recognizable enough to be used as a search result on the Realise website. In the end, this validation step was required both to include the photograph in the general results and before it was returned to the requester. This was done to ensure that junk tags were not returned.

Allowing the EyeSpy players to fulfill the requests of the website users creates a different game dynamic than when the EyeSpy players are creating and confirming targets amongst themselves. The website users will likely make requests for photographs that are not already in the system. This means there is a greater likelihood that the requests will be for things that the players would not pick on their own, which will make the game more varied. Additionally, the website users may make requests which call upon the EyeSpy players to give subjective opinions which are not supported by the existing game dynamic.

Game Types

While the principal game type of EyeSpy is largely unchanged, the priority tags do not fit the rules of an inversion-problem game. The priority tags would be more aptly described as fitting the rules of an output-agreement game, as described in Section 2.3.2.1. This is because the players are given the same input (the description of a place) and then asked to create a photograph for that description. While the photographs themselves may vary a great deal, the location in which they are taken needs to be the same before a photograph is accepted. It is this location output that must be agreed on, and for this reason the priority targets can be considered to operate as an output-agreement game.

This means that EyeSpy now operates within two different game types. This should help to add variety to the game-play and hopefully make the game less monotonous for the players.

Human Validation

As has been discussed already, an additional layer of human validation was incorporated for the new priority targets. Essentially, priority targets for the same request from Realise must be made in the same area for the EyeSpy players to gain points. However, in the initial trial, we only required two players to agree on a location before both photographs were accepted. This was not ideal and the threshold would normally have been made higher, but the time taken to go through this validation step, and have the photographs validated as a regular photo target was expected to take a long time in a trial with a small number of players. As such, it was felt that lowering the requirements of this first human validation step would allow Realise requests to be returned more quickly.

It should be noted, however, that priority targets were not limited to one location for validation. It is entirely possible that two or more locations could be used successfully, as long as at least one other user agreed on each location. Since each player only had one chance to confirm a priority target, they would have to choose their location carefully and try to predict what location would be most popular for the target. However, even if they chose a location that no one else agreed on, they could still gain points when their photograph was put back into the game as a regular photo target.

Using Recorded User Actions

As discussed in Section 3.1, EyeSpy has no real need for the benefits of using recorded user actions. Adding the new priority targets to the game has not changed this. However, it is possible that, should there not be enough requests from the website, previous requests could be resent to the game.

As has been discussed, there are many benefits to having priority targets in the game and it should be ensured that they continue to exist, even if the Realise users make no requests. However, due to time constraints, this feature was not implemented since the designers did not expect it to be an issue during the trials of the systems.

Keeping Play Competitive

It was hoped that the more variable nature of priority targets (variable points and irregularity of assignment) would make them feel like 'bonus' elements in the game. This was intended to make it easier for players to overtake one another as it allowed players who could not play as often to at least have a chance to keep up with the leaders. Even if this is not very likely (the top players also have the ability to take advantage of the priority targets), supplying an irregular scoring element introduces the possibility of catching up, and it was hoped that this would be enough to encourage people to keep playing. While it could be possible to restrict access to priority targets to players who are in the lead (thus making it easier for other players to catch up), it was felt by the game designers that this would be unfair to those who were doing the most work by playing the game the most.

Keeping play competitive does not only apply to the players who are falling behind the leaders. The leaders themselves need to be encouraged to continue playing. If a leader is too far ahead, they may see no challenge in continuing to play and thus stop playing entirely. It is beneficial to all players that game-play is kept competitive. The bonus element of priority targets could help with this.

A second issue in keeping play competitive for the leaders is to minimize the saturation problem that was mentioned in Section 3.3.1. The saturation problem occurs because of a diversity rule, discussed in Section 3.1.1, to ensure that players always target new locations instead of repeating them. However, in order to help alleviate the saturation problem, priority targets are exempt from this rule.

A new feature was added to the EyeSpy game in the form of an indicator to let users know that they were near one of the targets in their list. However, the indicator had ten times the range that was required for actually confirming the target, so the users would still need to find the target to get close enough to confirm it. This feature was added to allow people who were less familiar with an area to have a chance of competing with players that were.

The design goals of EyeSpy and Realise have now been discussed, but we must re-address the issue of preventing cheating in the systems and maintaining good quality by-products.

4.1.1 – Designing Against Cheating

It should be noted that the previous trials of EyeSpy saw no successful attempts to game the system. However, one player attempted to examine the packets that the phone sent to the game server in order to artificially increase his score. Two players had the idea of examining how the tag files were stored in the phone's memory in order to discover who made each tag. They planned to use the information so that they might collude with other players by only confirming each others' tags. None of these tactics proved successful.

It is encouraging to note that these tactics revolved round the technology rather than the game rules. This implies that there were no obvious ways to game the system using the interface that was presented. However, the changes made to EyeSpy and the inclusion of a second system means that we must re-address the situation to ensure that we are taking appropriate measures to prevent cheating which might be destructive to fair play and to ensure good quality by-products.

The Realise website introduces an interesting point about mutually reinforcing systems. The systems are supposed to govern themselves and trust each other to do so. For example, the EyeSpy game dynamic is set up to produce the best tags possible. Similarly, the website is presented to the users in such a way that their requests should supply good game data. However, what if EyeSpy did not require the confirmation step? Anything could be sent out of the system. There would be no prevention to stop people from sending abusive pictures instead of legitimate ones. There is also nothing to stop the website users from sending requests that contain swear words or other obscene language.

The solution to this problem is one that has been adopted before in cases of user generated content. For each place that user generated content is displayed, an option will be given to flag the data. This means that EyeSpy users should be given the option to flag a target that contains obscene material. Similarly, the website users should be able to flag photos that do the same. This is not an ideal solution as it does not prevent the content from ever being seen. In the case of text, certain offensive words could be screened for. However, images pose more of a problem. One of the points of the ESP Game [6] was to get humans to label images because computers are not good at it. Therefore, it only makes sense that it would be difficult to screen for abusive images without a human saying that they were abusive. In a way, this legitimizes the solution that is proposed as we are once again getting humans to solve a problem that computers find difficult, but it is still not ideal. The general failures and successes of flagging systems have already been discussed in Section 3.3.1 and so will not be discussed again here. However, due to time constraints, the EyeSpy system still provides no ability to flag targets. This was believed to be a more important feature for Realise because receiving abusive photographs in response to requests was a more probable possibility than the existing target types in the EyeSpy game.

While there is no flagging system in EyeSpy, there was already a mechanism for players to permanently delete targets which they could not confirm (as opposed to deleting them to free up a download slot to get an easier target). This was discussed under the rules to keep play competitive in Section 3.1. This mechanism was extended slightly in the new version of EyeSpy so that a target which was permanently deleted by three or more players would be removed from play entirely. This serves some of the purpose of a flagging system which is to remove 'bad' targets from the game. However, a true flagging system would reprimand the user who created the image in some way, and this was not implemented. As such, the players had no reason to feel guilty about permanently deleting a tag. Even if it was removed from the system, the original creator was unlikely to receive points from it anyway, and that creator was not punished in any way. Similarly, the new priority targets were subject to this same system.

Agreement Rules

Because the priority targets produce photographs that are put back into the game as regular photo targets, this introduces the possibility for people to simply create new photographs and ignore the request description. This would effectively double the allowance for creating photo targets. However, it is unlikely that such locations would be confirmed as priority targets, so the players could potentially be giving up a large number of points, even if they gained regular photo target confirmation points. In any case, even if they did this, the lack of agreement on a location means it is unlikely that such photos would be returned to the Realise users. Therefore, it is expected that the agreement rule to ensure that a location is accurate will be a good incentive to reduce the number of people ignoring the requests and just making their own photographs. And, even if they do, this will not likely get back to the Realise user who made the request and will still produce good quality by-products for the rest of the website.

Consensus Rules

The consensus rules from the previous version of EyeSpy are essentially the same. The new priority targets require a consensus among players to agree that the correct location has been chosen for the request description. However, due to the relatively small number of players in the trial, the threshold number of users required to agree on a location was only set to two. In effect, this reduces the consensus rule to the level of an agreement rule. However, in a larger scale deployment, this threshold could be increased.

Anonymity Rules

The anonymity rules within the EyeSpy game remain largely unchanged. However, the addition of a second system introduces a potential problem. What if the website users are colluding with the EyeSpy players? Or, even more worrying, what if the Realise users are also EyeSpy players? It is possible that the Realise user could send requests to EyeSpy which only a set group of players would be able to confirm either due to some privileged knowledge or use of a coding system. However, the need for other users (randomly assigned) to confirm the location and the photograph means that at least two EyeSpy players would need to be in on it, and possibly more, making it less likely that this would work. However, with the potential payoff being quite high due to the increased points for priority targets, it is perhaps more likely that people will attempt this method of gaming the system. This is a very real problem and the only solution within the current design would be to increase the threshold for people to agree on location before points are returned to the player. However, because this would increase the time it would take for requests to be returned to the Realise users, this was not done. However, while the situation may not be that likely, the problem remains a flaw in the system design.

Diversity Rules

The exclusion of the priority targets from the previous diversity rule, as discussed in Section 3.1.1, is only intended to allow leading players to get round the saturation problem. However, it is possible to simply use this as another way to create regular photo tags in the same location as previous ones. While this is a possibility, actually using the priority tags properly is worth far more points than this tactic. As such, it would not be particularly beneficial for players to use the priority targets to get round the diversity rule. In the end, the designers did not feel this would be a significant issue.

Competitive Rules

A problem of the previous trials that was believed to be anti-competitive was the use of strategic syncing to keep one player's tags dominant in the system. Players who synced their tags to the server early were more likely to have their tags spread between the other players. While this problem was not eliminated completely, the syncing in EyeSpy was changed from a manual to an automatic procedure and was carried out every time that a target was created or confirmed. This regularity of syncing meant that perhaps only one target or so would have an advantage rather than all the tags of a single user. It was expected that this would reduce the problems of manual syncing.

Some changes were made to the technology used in the EyeSpy game and this should be discussed. Also, the technologies used to create the website should be discussed to show how the systems were able to work together. This will be discussed in the next section.

4.1.2 – Technology

The Realise website included the ability to view the location of photographs on a map, showing the location where they were taken. The maps system used the Google Maps API to create a pin showing the location. The Google Maps API was created so that programmers could easily include map-based interfaces into their systems.[4] This can be seen in Figure 4.3.

In order to allow for the map system to work on the Realise website, the target's locations needed to be changed from a fingerprint based system to a system that incorporated latitude and longitude. The EyeSpy game was moved from the Windows Mobile iMate-SP5 phones to Apple iPhones running iOS. The reason for this choice was the iPhone's location sensing technologies which incorporated Wi-Fi, cell tower and GPS positioning in order to get the best possible location fix in a variety of environments. Some screen shots of the EyeSpy game client can be seen in Figure 4.4.

While this solved the issue for new targets that were created, it still did not allow the existing EyeSpy photographs and descriptions from the Windows Mobile version of the game to be included on the website. In order to allow these to be used, they were run through a system

[4] `http://code.google.com/apis/maps`

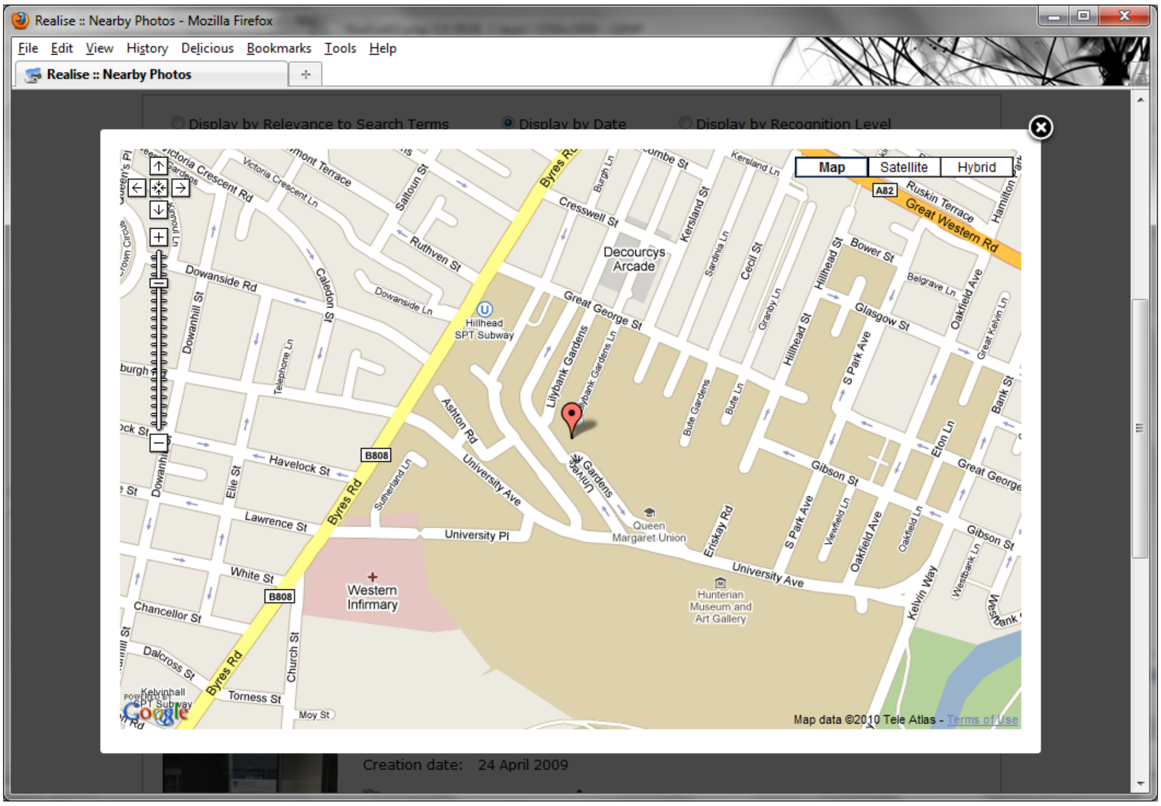

Figure 4.3: A map view of a photo result from the Realise website. This allows Realise users to see where a photo was taken by viewing its location on a map.

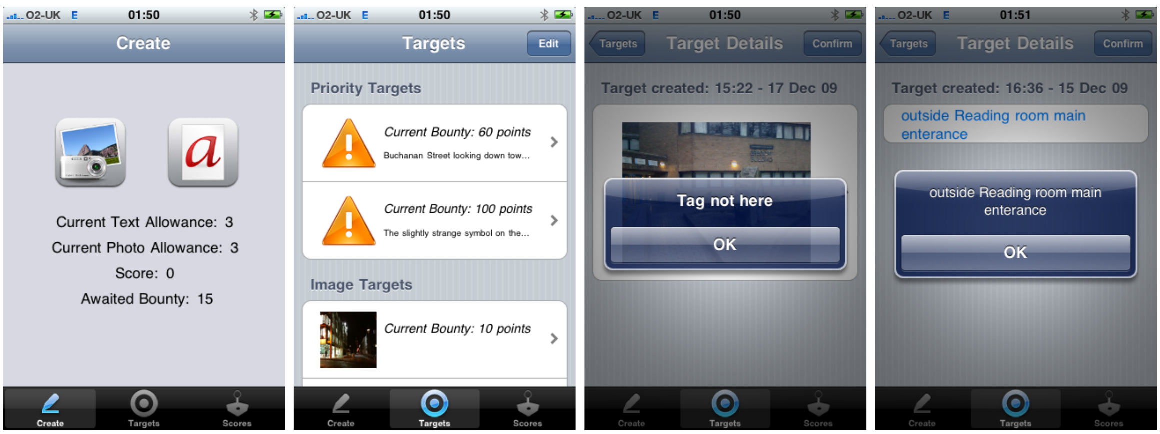

Figure 4.4: Screen shots from the EyeSpy game on the iPhone showing, from left to right: the create screen (where players can make image and text targets), the targets screen (where players can browse the targets which have to be confirmed), an image target screen (where users can confirm an image target, see its full text or when it was created) showing an unsuccessful attempt to confirm the target, and a text target detail screen (where users can confirm a text target, see its full text or when it was created).

called 'Skyhook.'[5] Skyhook is a system that allows programmers to get the best location fix possible from entering Wi-Fi, cell tower and GPS information. However, in the case of the Windows Mobile EyeSpy data, just having the Wi-Fi fingerprints for each photograph and text description was enough information for Skyhook to produce a latitude and longitude for each one.

The syncing system for the new version of EyeSpy used the 3G data connection on the phones. This was important in allowing automatic syncing which kept the game more competitive and prevented the syncing procedure from being used tactically.

EyeSpy and Realise have a similar setup to allow them to connect to each other. The systems each have a client and a database. The EyeSpy game is currently implemented as a phone-based game using the phone's GPS hardware and camera to create geographically referenced photographs and text. The EyeSpy database contains all these photographs and text descriptions so that other systems can access the dataset through a web service.

The Realise website accesses the EyeSpy web service so that it can display the photographs to the website users. It also contains its own database storing user and request information. In addition to this, the EyeSpy web service allows the website to add requests to the EyeSpy game. The Realise website also has a web service to allow the EyeSpy game to alert the website whenever a request is fulfilled by the game. The Realise website can then email the user who requested the photograph to let them know.

Using web services to keep this separation between systems means that other games with different styles of play can be added later on. This could be beneficial if there was a new system which could create a different sort of photograph. This will allow the website to maintain a strong stream of photographs into the system. Since the database for EyeSpy is kept separate from the client and a web service is used to access it from the Realise website, this means that a new game could have its database plugged into the EyeSpy web service, and the website would not have to change any of its code. Similarly, the EyeSpy game could be replaced completely and the website could be pointed to the web service of a new game that implemented the same services as EyeSpy. This limits the coupling between the systems.

[5] `http://www.skyhookwireless.com`

This use of web services also means that systems other than the Realise website could make use of the EyeSpy database. More systems with different goals making requests of EyeSpy could also create more varied tasks for the game-players to carry out which would make the game more challenging and enjoyable.

The Realise website was implemented as a fairly straightforward search engine. When a user searches using keywords, the website sends a request to the EyeSpy web service telling it the keywords and how the results should be ordered. The EyeSpy web service then checks to see if those keywords exists in any of the text labels in the dataset. For every matching label that is found, the EyeSpy web service returns images that were taken in the same area as the matching label.

The initial work on porting EyeSpy to iOS from Windows Mobile was undertaken by Dr. Marek Bell. This rough version was then recoded and cleaned up by the author with help from Don McMillan, Owain Brown and Alistair Morrison. The Realise website was coded entirely by the author.

The design of the two systems has now been discussed in depth, but a trial was required was to test the design and determine any problems that would need to be resolved in future versions.

4.2 - Trial Design

To test the Realise website design and the changes to the EyeSpy game, they were trialled with 15 participants. 5 of these participants were assigned to the Realise website (1 female and 4 male) and 10 assigned to the EyeSpy game (7 female and 3 male). The trial lasted for two weeks. The game was not seeded with existing targets, but the by-products from previous trials were included in the Realise website.

The EyeSpy players were paid £10 for each week that they played. The winner was given double the amount as a prize. This was done to encourage a sense of competition between the players. The Realise users were paid £10 for each week that they took part.

7 of the EyeSpy players were drawn from computing science students, 1 was an engineering student, 1 was a biomedical student and 1 was an environmental chemist. None of the computing science students were friends, but the other three players were. The players all worked, studied or lived in the area of the city around the University of Glasgow were the game would principally be played. As before, this acted as a natural limit to the 'game area.' The players were asked to concentrate their play within this area though there were some tags that were further removed from the main campus. All players were interviewed at the end of the trial.

1 of the Realise users was a computing science student, the other 4 had no current connection to the University. The computing science student did know two of the other users. All the website users were familiar with the University campus where the game would take place. The website users were told that the EyeSpy players would be within the University campus during play and that requests outside of this area were unlikely to be satisfied. All users were interviewed at the end of the trial.

While the interviews for the previous trials had focused on by-products, this trial focused on finding out how people felt about the dynamic between the two systems.

The next section will discuss the initial findings from the first trial of Realise and EyeSpy working together. This will show how the results led to improvements for the next iteration of the systems.

4.3 - Initial Findings

The trial produced 230 images and 133 text descriptions. 63 of the images (27%) were produced in response to requests made to EyeSpy from the Realise website. Out of a total of 363 images and text labels, 63% were images and 37% were text. On average, users produced 24 targets during the game: 15 images and 9 text labels.

Almost a third of all images produced were to fulfill the direct needs of Realise website users. This strong level of directed effort is a major benefit of mutually reinforcing systems as it ensures that the by-products produced are those which are needed.

A total of 66 requests were made by the Realise users. However, only 26 of these requests (39%) were satisfied. Through examining the trial data and interviewing the participants, this relatively poor return on requests seems to stem from EyeSpy not being able to keep up with the number of requests that were made. Essentially, the Realise users made a large number of requests and the current game dynamic took too long to produce responses. Specifically, it seems to be the two step confirmation procedure that is slowing things down. Having to wait for other users to confirm the location and the quality of the photographs means that not enough images are getting back to the Realise users quickly enough. The Realise users themselves reported that they were often waiting several days for a response to requests and that the responses they did get mostly came in the last few days of the trial. However, the Realise users continued to make new requests each day before they had received responses to their previous requests. In some cases, repeating their earlier requests in hopes that they would return a photograph more quickly on a second or third attempt. This also proved to be annoying to the EyeSpy players who reported that they received the same request on two consecutive days and thought it was a bug in the system (when in fact it was two similar requests from the same Realise user). In several cases, the Realise users stopped making requests after the first few days because they had not had any responses and assumed the system was not working. It was not until they started receiving responses that they began to make more requests in the latter half of the trial, at which point there was not enough time in the trial for responses to be made, confirmed and returned.

The players also felt that the timer based scoring mechanism did not feel fair. In some cases, they would be lucky enough to be near a priority target's location and in other cases, simply traveling to that spot would lose them a great deal of potential points. Even though this was still more points than for the other tag types, players felt that it was too much about luck whether or not this tag type would payoff in the long run when other people may always be closer to priority target locations.

The players were also unsatisfied with the confirmation mechanism for priority targets. With the other target types, players would get immediate feedback that they gained points or were in the wrong location. With priority targets, users only had points added to the separate 'potential points' score. When a priority target was validated within the game and by the website users, these points were quietly transfered to the normal score. However, due to the amount of time taken for the validation of priority targets, players felt that their potential points were ever increasing and that they were never being added to their normal score. However, analysis of the data shows that this is not true and at the end of the trial 53 (84%) of the 63 photographs created from priority targets had been successfully validated by the two systems. However, the long time taken for potential points to become real points was still disconcerting to the players.

In general, however, the players reported enjoying the game and, unlike the previous trials, none of the players reported any feelings of monotony during play. The saturation problem, though it did not stop play entirely as in previous trials, was still an issue. The leaders in the game still reported that they struggled toward the end to find places to create new targets.

The game still produced one clear leader (who had double the points of the player who came second). However, the other players' scores where all much closer together than in previous trials and the ordering of these other players changed several times during the trial, showing that it was now easier to overtake other players. The player who came first essentially created and confirmed as many targets as the system allowed him to every day of the trial. This may suggest that the number of targets that can be created and confirmed each day is still too high and should be reduced to keep the game more competitive.

A new method to game the system was discovered by some players in the trial using the nearby targets indicator that was added to the game. This feature was added to make the game more competitive to players that were less familiar with an area by alerting them when a target was nearby. It was intended that the players would then search for the target within a bounded area (where the indicator was shown). However, the players did not know which target was nearby or exactly where it was. Several players reported that they would simply go through all their targets and try the confirm step, even though they still did not know where the target was. Players admitted that, although rare, this had worked in one or two cases and they had successfully confirmed a target without knowing where it was. However, players also said that they had found the feature helpful with especially difficult targets and had been able to find them with help from the new feature. This new feature of the game seemed to provide only marginal positive and negative effects, but it was felt that it might still be beneficial to players who were unfamiliar with the game area (which was not the case with this trial).

All the Realise users reported that they were very happy with the responses they received (even if they took longer than they had hoped) and that they were generally what had been asked for. However, one of the Realise users reported that his requests, though validated by EyeSpy, were not fully satisfied. The requests were to take pictures of a certain street at a particular time of day. While the responses he got back did indeed show the requested street, they were taken much later in the day. In this particular instance, the time of day was important to the Realise user because he wanted to know how many car parking spaces were still available at that time of day.

However, since the scoring system for priority targets was timer based, none of the players wanted to wait to take the photograph at the requested time of day. While it would be possible to enforce that requests are only taken at a particular time of day, there are two problems with such an implementation. Firstly, it negates the purpose of the timer based scoring system and players would be likely to lose a lot of potential points while waiting for the right time of day. Secondly, this opens up a whole new area of abuse from the website users. For example, the Realise users could start making all their requests for times when it was dark (the phones used in the trial had no flash and worked poorly in the dark), essentially making targets that were impossible to get because no one would be able to confirm them (they would just appear completely black and unrecognizable). A more sinister outlook is that Realise users would have the power to get an EyeSpy player to a particular location at a particular time. If the place and time chosen would lead to such a place being secluded, this could be very unsafe for the EyeSpy players. While people should be responsible for their own safety and use common sense when playing the game, it was felt that enforcing a time based request within the game would be irresponsible of the game designers.

This issue is not unique to the EyeSpy game. There have already been games and mobile publishing platforms that reveal a person's location during play or when publishing something on-line [123]. This information could be used, for example, to determine that you are not home and open the possibility for someone to rob your house when you are not there. The website 'Please Rob Me' tries to highlight this issue to users of such systems.[6] This shows a genuine issue with location based systems and privacy is an important aspect of keeping safe. While EyeSpy and Realise maintain anonymity and privacy between all

[6] `http://pleaserobme.com`

users, the possibility for trapping someone with a request is very real. But, as with other systems of this kind, it is essentially the duty of the users to be responsible with their actions and to be safe with their decisions. In the end, the Realise user did confirm the photographs because they were of the correct subject, just the wrong time of day. He felt that this still justified the players getting the points.

One of the potential benefits of the systems was allowing subjectivity to factor into requests. In general, less specific requests seemed popular for the EyeSpy players. They reported enjoying the creative freedom, while still having to fulfill a request. Many players also reported that they tried harder to make more aesthetically pleasing photographs for priority targets because they knew someone actually wanted them. By contrast, they were not sure if anyone really cared about the regular photo targets they created. The Realise users also reported their appreciation for the quality of the photographs which were returned, favoring photographs that had been more creative over ones that carried out the request in more straightforward ways.

However, as has been discussed, Realise users were occasionally dissatisfied with responses when they were very specific and detailed in their requests. EyeSpy players would sometimes not follow complex instructions, suggesting that it was not worth the extra effort as they might still get the points anyway. Some EyeSpy users reported that they felt some of the requests were too demanding and that this made them less fun.

Some Realise users suggested that they would like more control over the distribution of points in the game so that players who made better photographs could be rewarded. This suggests that a stronger link between requests and responses might be needed to ensure the by-products are exactly what is required for another system.

While the concept of mutually reinforcing systems shows potential, we have shown that several issues were raised during this first trial. As such, there are some improvements that could be made to the design of the systems.

4.3.1 – Possible Improvements

Four main problems still remain in the design of EyeSpy and Realise. There is still a saturation problem in the game where players run out of places to create targets; there is still an indication that some players can race too far ahead of the others making the game less competitive; it takes too long to validate priority targets; and the Realise users do not have enough control in saying if responses satisfy their requests.

While the priority targets have allowed leading players to continue playing even after they have run out of places to create new targets, the fundamental problem still remains which is that the players who are creating the most by-products are being prevented from continuing to play. As such, a lifetime should be added to locations in the game. This means that, once a user has created a target at a particular location, he or she will not be able to create a new target there for a specified period of time, rather than forever (which is the current system). Although this goes against the original reasoning behind the diversity rule which led to the saturation problem, it is more important that people be allowed to continue playing the game. It is also important to note that places change over time and even from day to day and documenting these changes can also produce useful by-products. As such, player should be allowed to create a target in the same location after a matter of days.

Another potential solution to this problem was to use the digital compass feature of the iPhones that were used for the trial. This feature allows the phone to determine what compass direction it is facing. This could be used to allow more than one photo target to be created in a location if the phone was facing in a different direction. However, this would likely be a complex idea to explain through the phone's interface and it would only alleviate the problem without solving it. Adding a lifetime to locations is a more permanent solution.

While the inclusion of priority targets has made it easier for players to overtake each other, it is still possible for a user to gain a clear lead over players who have less time to play. As such, the number of targets which can be created each day will be reduced from five photo targets and five text targets to three photo targets and three text targets. Similarly, the number of targets for each type that is downloaded to the phone each day will also be reduced from five to three. It is likely that this will reduce the overall number of targets that are created, but it is important that the game be kept competitive. If one player races ahead, the other players (as shown from the trials) will become less enthusiastic. This will lead to by-products being created by a small number of players instead of a diverse group. Since human computation relies on consensus rules to create broadly usable by-products, it is important that the game is kept competitive.

The priority targets have also proven to take too long to validate. This has left the Realise players waiting too long for their responses and has made the EyeSpy players wait too long to receive their points. A different validation system is necessary to try and reduce these problems.

The last issue is in allowing the Realise users more control to determine what points are received by EyeSpy players when responding to requests. Since the point of the requests is to allow the game to respond to the specific needs of another system, it makes sense that the Realise users shown have control over the points mechanism. As such, EyeSpy and Realise should be redesigned so that the Realise users have more control over how many points the EyeSpy players receive for their responses.

These last two problems might be solved with the same solution. Rather than rely on the game to validate the location and recognizability of priority targets, we could let the Realise users do it themselves. This would ensure that Realise users receive their responses as soon as they are made and can give points as appropriate. It is also likely that this will take less time that the current validation technique, meaning that EyeSpy players will receive their points in less time.

An additional, but minor improvement, might be made to encourage more reinforcement between the two systems. In the current implementation of Realise, the user requests are not used other than to let EyeSpy know what photographs are required. However, these requests could be attached to the same locations as the returned photographs, providing more information for the search engine. This means that other users would benefit during searches and would be less likely to request the same things as other users. This would keep the requests in the EyeSpy game more diverse and less likely to be repeated. Also, if a Realise user did search for something and no results were returned, the website should then encourage the user to make a request to fill the hole in the data set. It might even be worthwhile automatically adding such requests to the EyeSpy game under a dummy user account if enough people search for the same thing but return no results.

4.4 – Conclusions

In this chapter we introduced an augmented version of EyeSpy that fixed some problems of the earlier version. The problems included the use of manual syncing to gain a tactical advantage, the creation of broad by-products rather than what was specifically needed by another system and relieving the monotony of the game play. Some of these augmentations relied on a website, called Realise, that could use the by-products of EyeSpy, as well as request new ones. These systems were used together, using each others' strengths, so that they would work better as a whole. They would reinforce each others' goals and add new capabilities. When the two systems work together in this way, we call them mutually reinforcing systems.

The by-products of EyeSpy proved to be useful to a tourist website like Realise. The local knowledge used to make the photographs in EyeSpy meant that the locations were selectively chosen and were easily identifiable. These were features that would be beneficial to tourists as it would help orient them in an unknown location. Additionally, the Realise website could make requests of the EyeSpy game for photos that had not yet been produced. It was also shown that the website could help direct the play of EyeSpy, injecting a new game dynamic not currently present, which might make the game less monotonous.

A new game element was added to EyeSpy called a priority target. Priority targets were the method used to incorporate the Realise user requests into the EyeSpy game. Priority targets appeared in the game as a description for a photograph. Users would then have to take a suitable photograph to match the description. Once the EyeSpy game had validated the photographs, they were sent back to the Realise users. If the returned photographs were found to be inappropriate, the Realise users were able to flag them. This added a further mechanism to validate the by-products from EyeSpy.

This method of requesting photographs from EyeSpy also allowed requests of a subjective nature that required choices or opinions to be made by the EyeSpy players. This is an unusual feature for a game with by-products, as objectivity is usually promoted in such games. However, in this instance the subjectivity is a benefit.

The priority targets proved popular amongst the EyeSpy players and almost a third of all photographs produced during the trial were created through these targets. It was also noted that the players in this trial did not complain of the game becoming monotonous, as in previous trials before the priority targets were added.

We have also shown that mutually reinforcing systems need to trust each other to do what they claim. For example, if EyeSpy does not take due care to validate by-products before allowing them to be used by Realise, the Realise website would suffer because the results of searches may not be accurate.

Adding EyeSpy to a mutually reinforcing system loop has resolved some of the problems that the game suffered from: the by-products produced are now specifically requested by the system that uses them (so they are closer to what is required and less broad), the game is less monotonous because of the randomness and challenge of the new priority targets (these are like bonus targets because the players cannot guarantee that they will always have them during play) and players feel more competitive because the priority targets are worth a variable number of points depending on the time taken to confirm them. This means that the game encourages finer grained results where required and is now more enjoyable to play because it is less monotonous. By working together in a mutually reinforcing system loop, Realise can now request specific data from EyeSpy, and EyeSpy becomes more diverse and enjoyable to play.

Since there are other games with by-products that share similar rule systems to EyeSpy, the concept of mutually reinforcing systems could be applied to them as well. By examining the system that uses the by-products from the game it might be apparent that there is a need for specific by-products or areas where the currently produced dataset is lacking refined by-products. This can then be turned into a request that can be sent to the game. The game can then present this request to the players as a new feature of the game dynamic.

Once this mutually reinforcing system loop is constructed, opportunities to exploit system strengths can be explored. This is because the designed by-product of a system may not be the only useful by-product. This should lead to by-products that are fine grained and directly needed. It should also allow the games to have more compelling content.

A trial of the two systems highlighted that there were several problems. The procedure to validate the photographs created by priority targets took too long. This left the Realise users waiting a long time on responses to their requests and meant EyeSpy users had to wait a long time before receiving any points. The game was also shown to be too slow to keep up with the number of requests that were made by the website users.

Although the priority targets prevented leading players from stopping play altogether, the saturation problem from the previous trials was still present. It was also shown that some players could still race ahead, making it too hard for anyone to catch up.

Lastly, Realise users complained that they did not have enough say in whether or not a response matched their request.

By augmenting EyeSpy and allowing it to interact with the Realise website, we have demonstrated the concept of mutually reinforcing systems. This has shown how EyeSpy could be used to match the specific needs of another system. This has helped in addressing ***RQ1***. The augmentations to EyeSpy also allowed us to explore potential solutions to the problems suffered in the earlier version of the game. This has further helped toward answering ***RQ2***. The lessons learned will help inform ways in which we might further improve the two systems so that they work better together to achieve their individual goals.

The next chapter will focus on some improvements to the systems that attempted to combat these issues while maintaining the benefits of mutually reinforcing systems and mobile games with by-products.

Chapter 5
Subjective Reward System

Modifications to the EyeSpy game have allowed it to answer the direct requests of another system. These requests were used as new game content for EyeSpy and introduced a new game dynamic that would not have been possible without the requesting system. The Realise website has been introduced to show how the by-products of EyeSpy might be used in a tourist website. It has also been shown that such a website could make direct requests for new information from EyeSpy when there were holes in the data set. In this way, EyeSpy receives better game content and the Realise users get the specific information they require. We called these 'mutually reinforcing systems' because they work together for mutual benefit, reinforcing the goals that each has begun to achieve on its own.

While the basic premise was shown to be somewhat successful, there was a need to improve the systems to make them work together more efficiently. In particular, the procedure to validate request responses took too long and the Realise users did not have enough control over rewarding the players. Some problems from the original version of EyeSpy still remained as well. The saturation problem which prevented the best players from continuing was still present, and the problem of players racing too far ahead was still apparent. In this chapter, we will discuss the final improvements that were made to EyeSpy and Realise in order to tackle these problems.

The previous trial showed that the EyeSpy players enjoyed requests that relied on subjective opinion and allowed them to be creative with their responses. This level of subjectivity is unusual in games with by-products. This is because objectivity leads to by-products that can be used in a broader range of applications. However, as we have shown, EyeSpy's priority targets were set up to respond to the specific needs of another system. In the case of Realise, the specific needs led to subjective responses and the Realise users responded favorably to such activity. In particular, the Realise users said they would like to be able to reward more points to users who produced more aesthetically pleasing photographs. Another problem with the validation system was that it took too long to return responses to the website, leaving the Realise users waiting on their photographs and the EyeSpy users waiting on their points.

A means to solve these problems might be to cut out the existing validation system altogether and put the Realise users in charge of the points for the priority targets. If the Realise users are in charge of the points, it means that photographs can be returned immediately to the website and the website users can say whether or not the points are deserved. While this will be a subjective decision on the part of the website users, the photographs were created to match their exact needs, so it is only them who can really say if it has been done so successfully. This would lead to quicker satisfaction for the Realise users and would allow them to more quickly give or deny points to the EyeSpy players. This should help solve the issue of the existing validation system taking too long to satisfy both players and website users.

Another issue that might be addressed with this solution is that the EyeSpy players did not like the timer based scoring system of the priority targets. The reasoning behind the timer based system was to make the priority targets worth a variable number of points and to encourage users to respond to them as quickly as possible. This is also something that could be put in the hands of the Realise users since it is their requests which need to be satisfied. To achieve this, we could allow the Realise users to choose how much each of their requests will be worth to the EyeSpy players. This would ensure that there would still be a variable points model, though not timer based, and that Realise users could prioritize their requests by giving more points to the requests which were more important to them.

To ensure that Realise users prioritize their requests, we could give them an allowance of points. These points can then be assigned to their requests. Having an allowance means that Realise users will not just be prioritizing their own requests, they will be prioritizing them amongst the other Realise users. For example, if a website user just wants one photo and wants it quickly, he or she can assign all of his or her points to that request. This would mean that EyeSpy players would also prioritize requests that were worth more and deal with them first. Thus, this Realise user would receive a response more quickly than someone who wanted lots of pictures (as the latter would have to divide the allowance between their requests).

This allowance can also be used to ensure that the website users validate their returned photographs appropriately. If the website users have to lose points from their allowance every time they accept a photograph, they will have to choose carefully which one is best. We call this type of validation, in which users are given an allowance to ensure appropriate care is taken, a subjective reward system.

It should be noted that the previous trials of EyeSpy and Realise made assumptions about the players and website users. It was generally assumed that EyeSpy players were locals who were familiar with the game area, and that Realise users need not be familiar with the area, but needed to know enough about it to request information. In order to explore how these assumptions might affect use, this chapter will test the final versions of the systems with both locals and non-locals using them.

5.1 – System Design

Changes were made to EyeSpy to allow the inclusion of the subjective reward system. However, these changes largely involved simplifying the game by removing the previously complicated validation mechanism for priority targets in favor of allowing the website users to take charge of it. The display of targets within the game interface was also changed to show the ‘bounty’ for each target. In the case of text and photo targets, this was always set to 10 points. However, in the case of priority targets, this was set by the Realise user who made the request and the Realise users could not set bounties of less than 15.

The Realise website required more significant changes. Each Realise user would receive an allowance of points each day which could be attached to his or her requests. This allowance was accumulative so that, if it was not used on the day it was received, the Realise user could still use it another day, along with any other points that he or she had received. However, in order to receive his or her daily allowance of points, the Realise user would need to log in to the website at some point during that day. This restriction was put in to prevent users from flooding the system with high value requests after a long period of inactivity.

When a Realise user makes a request, he or she must now indicate how much of his or her points allowance should be attached to that request. The user is unable to attach more than his or her remaining personal allowance. However, the user does not lose the specified amount from his or her allowance when the request is made. The points are only deducted once the user accepts one of the returned photographs. The points are then transferred to the successful EyeSpy player who created the photograph. The website user can decide not to transfer the points if the photograph does not match the original request.

Equilibrium Between Systems

The website users are essentially creating bounties for their requests. The number of points that a Realise website user is assigned is based on the overall system activity for the previous day. If the number of requests with returned images is less than the number of requests made, a larger number of points will be assigned to the website users that day. This means that if a smaller percentage of website requests were dealt with by the game players, the website users can assign more points to their future requests. By increasing the amount of points that requests will be worth, this should encourage the EyeSpy players to devote more of their concentration on the priority targets, which should also help to catch up with the backlog of requests in the system. In assigning requests to players, the system will also favor requests that have larger bounties. This adds more value to the points as it indicates that the more that Realise users assign to a request, the quicker they are likely to receive a response. This should encourage more points being used on a smaller number of requests. The reasoning behind this is to ensure that a balance is maintained between the number of requests being made and the number of responses being created by EyeSpy.

A problem with this is that many users might log into the Realise website every day and receive their points but never make any requests. Then, one day, all those users may decide to use all their points on lots of requests. This would have the effect of flooding the system and the EyeSpy players would take too long to make all the photos. As a preventative measure against this, the system also gives each Realise user an allowance of requests that can be made each day. However, this allowance does not accumulate and is reset every day. This allowance is also adjusted automatically based on the overall activity in the system. If there are greater numbers of requests awaiting photos, there will be fewer requests that are allowed to be made. In some instances where there are too many requests awaiting photos, this may mean that a Realise website user is not allowed to make any requests on a given day.

It should be noted that the subjective reward system sounds, superficially, like an economy model. However, this would not be an accurate description because the points allowance is created by the website. There is no set amount of points within the system and more can be created at will when required. If this were an economic model, it would lead to inflation and the points would become less valuable. While it might be technically feasible to create such a model, it would not make sense because the EyeSpy users can only accumulate points and cannot spend them. This would essentially bankrupt the website and no more requests would be possible. As such, the current model exists purely to put restrictions on the number of requests and to allow Realise users to prioritize them. The secondary benefit is a variable points system within the EyeSpy game and a faster validation of priority targets, which benefits everyone.

These measures mean that each Realise user has a fair chance of getting photos for their requests. This method of valuing the work between systems in a mutually reinforcing system loop is important. The systems should form a partnership without one becoming too dominant, or else the equilibrium will collapse and the benefits of mutually reinforcing systems might be lost.

Payment Through Entertainment

The only significant change to the game dynamic in EyeSpy is the removal of the timer based scoring system, the removal of the slow validation system for priority tags and the switch to letting Realise users determine the points that priority tags are worth as well as whether or not the players receive the points. It was thought that these changes would make the game more enjoyable because EyeSpy players will have to wait less time to receive points for their efforts. Also, the removal of time based scoring should make the game more enjoyable and perhaps more fair as users from the previous trial reported that they disliked this feature and that it might give players an unfair advantage based on luck.

Game Types

In Section 4.1, we noted that the priority targets operated as an output-agreement game.

However, with the changes made to priority targets, they now operate as an inversion-problem game, like the other target types in EyeSpy. This is because the website users are describing something to the players. The players operate as 'guessers' trying to return the correct photograph to the Realise users.

Because the validation takes place entirely between humans with no computer determining if the match is accurate, this is technically outwith the existing inversion-problem model. However, due to the required subjective nature of the new design, this divergence is acceptable.

Human Validation

If a requested photograph is satisfactory, the Realise user will then pay the bounty (giving the EyeSpy player the points). This will allow the Realise user to access the full sized photograph (they must make their determination of whether a photograph is satisfactory based on a smaller thumbnail version). This has the double effect of validating that the photograph matches the request text and giving the player and website user what they wanted (respectively, points and a full sized photo).

The photograph itself will return to the game as a regular photo target so that it can be confirmed by other players as well before it is added to the usable data set. If a photograph is never confirmed by a Realise player then it will never enter the main search results for other Realise users. Each time the photograph is confirmed as a regular photo target, the player who created it will gain points. Thus, even if a user does not have his or her photograph accepted, he or she will still be able to receive some points for the effort put in.

The website users can report inappropriate photographs that appear on the website and any player who has enough photographs reported can then be banned from using the system. This banning is not automatic. Once a threshold has been reached, an administrator is alerted to review the activity and determine if a ban in warranted. In the case of our trials, this would be the game designers' decision. This prevents game players from colluding to try and get another player banned by flooding the system with reports of inappropriate behavior when there is none. It is, however, unlikely that this would happen because only a target can be flagged, not a person. Since the person who creates a target is kept anonymous, it would be hard for other players to target anyone specifically. Admittedly, banning people will not prevent them from creating a new account and repeating their abusive behavior, but they will have to start at the bottom of the leader board in the game. More extreme methods of identification could be included to make people accountable for their actions (such as sending an authentication code in the post to someone's home address, making it difficult to register multiple times). However, unless a serious problem of abusive behavior is demonstrated in the system, this would merely detract from the experience of the users.

In addition to getting targets which are more interesting to confirm, using the second system to validate requests adds a new scoring system to EyeSpy that would not have been possible without mutually reinforcing systems. The confirmation procedure is now much quicker than before and also simpler to implement. Also, because the previous validation system required agreement between players on both location and photograph recognizability, it is possible that EyeSpy players are more likely to get points now as the previous validation system made it quite difficult for a photograph to pass. As before, the priority targets act like a bonus in the game. The number of priority targets received and how many points they are worth will appear to be random to the EyeSpy players, making the game-play less monotonous. Also as before, the requests themselves may include non-location based requests for specific objects, requests which have potentially more than one correct answer and requests which may require the player to give an opinion.

Using Recorded User Actions

The changes to system design have not affected the reasons for not including recorded user actions in the game. This has already been discussed in Sections 3.1 and 4.1.

Keeping Play Competitive

In the previous version of EyeSpy, all goals were set at five: players were allowed to create five text targets each day and five photo targets each day. Additionally, players would receive five text targets and five photo targets to confirm each day. This limit was put in place to keep game play competitive and prevent some players racing ahead. However, as this still became a problem, it was felt that this number should be reduced to three. While this meant that the overall number of tags produced by the system would be less (because the top players would not be able to do as much), it was felt that ensuring the by-products came from a more diverse range of players was more important. However, since the number of targets which could be created was reduced, it was hoped that this would also encourage use of the priority targets as the photographs created by these targets would also enter the game as regular photo targets. It is important that the priority targets are favored to ensure that Realise users have their requests satisfied as quickly as possible as well as to ensure that the game can keep up with the stream of new requests.

In addition, a lifetime was added to locations so that after three days, a user would be allowed to create a new target there. This should still prevent users from excessively creating targets in the same location, but should prevent the saturation problem that was seen by the leading players in previous versions of EyeSpy. As in the previous version of EyeSpy, the priority targets are exempt from this rule and can be made anywhere.

Another important aspect of putting the website users in charge of the points for priority targets is that there is now a game mechanism which does not rely on the other players to achieve points. The other options in the game rely on other players to either confirm a player's targets or create targets to be confirmed before a player can gain points. However, priority targets now gain points due to people outside of the game. This means that players need not necessarily rely on other people to be playing in order to receive points. This is particularly important if requests are made outside of the normal play area where it would be less likely that other players would travel to confirm targets. Similarly, if other players were too busy to play the game for a while, it would still be possible to gain points by focusing on priority targets.

5.1.1 – Designing Against Cheating

Changes in the design of the systems once again require a re-assessment of whether or not the designs appropriately deal with the potential for cheating. Reassigning control for the priority targets' points changes the game dynamic for that feature and we must ensure that the systems can still maintain good quality by-products.

Agreement Rules

Due to the subjectivity of the request/response relationship, the Realise users are the only ones who can agree that a response matches their request. While this is valid for that one user, we must still ensure that the photographs produced are more broadly usable before being accessible to the other website users in the main section of the Realise website. To ensure this, the photographs are put into the game as regular photo targets to be validated.

Consensus Rules

Before a priority target photograph is added to the general body of by-products to be used in the main part of the EyeSpy website, the same threshold of users must agree that the photograph is recognizable by being able to confirm it as a regular photo target. This is the same consensus rule used for the regular photo targets in the game and using the same validation method justifies using the photographs in the same way as general by-products of the game.

Anonymity Rules

As discussed in Section 4.1.1, there is a possibility for EyeSpy and Realise users to collaborate together. Now that the Realise users are in charge of the points for priority targets, this makes this possibility more likely. It will now be easier for Realise users to only confirm photographs from the EyeSpy players they are colluding with.

In order to minimize this effect, the systems maintain their anonymity as much as possible and no identifying information is passed between users. However, the narrow band of communication supplied by the requests themselves and the photographs returned could be enough to allow players and Realise users to identify themselves to each other. However, the Realise users only have access to a small thumbnail version of the photographs returned until they transfer the points, and this may help to minimize such communication. However, this problem remains.

One solution to combat this would be allowing the EyeSpy users to flag Realise users who appear to be sending identifying or coded information in their requests. However, this feature was not implemented due to time constraints.

Diversity Rules

A potential problem with the original validation system for priority targets was that users might respond to priority targets but ignore the actual request. This would extend the number of photo targets that a player could create in a day because the priority targets are put back into the game as regular photo targets for validation.

Such a tactic was very hard to track because of the way the priority targets were validated.

If no other player chose the same location to answer the same priority target, these photographs would never be validated as priority targets, but there was no mechanism for anyone to check on photographs that never passed the validation. They were simply not returned to the website user, but may have successfully passed the second step of validation by being recognizable as regular photo targets. Another reason players might use this tactic is because priority targets are excluded from the rules preventing the creation of a target in the same location. In essence, this was a way to get round the diversity rules included in the EyeSpy design.

Though this tactic was never used, the new validation system might prevent it from happening in the future. Now that all priority target photographs are immediately sent back to the website, the Realise users can flag photographs that bear no resemblance to their request. This will be to their benefit because it will remove such time wasters from the system as, when enough people flag the player's photographs, they will be nominated for removal from the game.

Competitive Rules

As has already been discussed, the number of goals to be achieved each day has been changed from five to three. However, no other anti-competitive behavior was discovered in the previous trial and the changes to the game design were not expected to introduce any new ones.

5.1.2 – Technology

The technology used in this version of the systems was the same as in Section 4.1.2. The changes to implement the subjective reward system only required some additions to the existing web services to allow the bounty information to be sent across systems.

The points themselves were created based on the previous day's activity of the Realise and EyeSpy users. If the number of requests with returned images was less than the number of requests made, then Realise users would have a smaller allowance of requests the next day. However, this would also lead to a larger number of points being assigned to the website users that day (to encourage EyeSpy players to focus more on priority targets).

While the designs of the two systems did not change in terms of how they were played, or how they communicated together, several technical issues to do with performance needed to be fixed. This led to a complete recode of the EyeSpy game to be more efficient and robust (the new design was based on the lessons learned from debugging and fixing the older version). A similar recode was required for the website to improve the efficiency of the search system. Before this recode, an average search in Realise could take between 15 and 50 seconds. After the recode, this average was reduced to between 1 and 4 seconds (the new design placed more of the search code within the database queries, rather than returning larger quantities of data that were filtered down outwith the database).

The design changes and the recoding of the systems were carried out entirely by the author.

5.2 – Trial Design

To test the final versions of the systems, they were trialled with 31 people (6 female and 25 male) over two separate rounds. In each round, the participants got paid £10 a week with winners of the game being paid double. The games were not seeded with existing tags, but the Realise website included all by-products created in previous trials. As with previous trials, the players all worked, studied or lived in the area of the city around the University of Glasgow were the game would principally be played. As with previous trials, this acted as a natural limit to the game area but some players made tags outwith this area.

The first round included 18 participants split into two groups: 9 locals and 9 non-locals. The locals were people who were familiar with the game area and had lived in or near there for over three years. The non-locals were people who were unfamiliar with the game area and who had recently (within three months) moved there. The locals were all computing science students as were all but one of the non-locals. The remaining participant was a linguistics student. Only two of the participants knew each other well.

The second round included 13 participants split into two groups: 7 locals and 6 non-locals. The locals included 4 computing science students, 1 neuropharmacology student, 1 geology student and 1 engineering student. The non-locals were all computing science students. Three of the locals formed one group of friends, and two other locals formed another. Two of the non-locals were friends.

This selection gives a broad range of users with varying levels of local knowledge.

Initially, the non-local users were assigned to the Realise website and the local users were assigned to the EyeSpy game. Half way though the trial, the participants swapped roles. It should be noted that this was run as two separate games so that the locals and non-locals were not competing against each other. The participants spent 11 days on each system.

The trial participants were all interviewed about their experiences. As with the previous trial, these interviews focused on finding out how people felt about the dynamic between the two systems. Particular attention was paid to how locals and non-locals might have had different opinions about the systems.

The next section will discuss the initial findings from the two rounds of the trial. This will show how successful the final systems were in their design goals and what potential improvements could be made to the systems if time allowed for future versions to be made.

5.3 – Initial Findings

The trials produced 197 images and 157 text descriptions. 60 of the images (30%) were produced in response to requests made to EyeSpy from the Realise website. Out of a total of 354 images and text labels, 56% were images and 44% were text.

When non-locals were playing the game, 85 images and 67 text descriptions were produced. 23 of the images (27%) were produced in response to requests made to EyeSpy from the Realise website. Out of a total of 152 images and text labels, 56% were images and 44% were text.

When locals were playing the game, 112 images and 90 text descriptions were produced. 37 of the images (33%) were produced in response to requests made to EyeSpy from the Realise website. Out of a total of 202 images and text labels, 55% were images and 45% were text.

On average, users produced 11 targets during the game: 6 images and 5 text labels. When non-locals were playing the game, users produced an average of 10 targets: 6 images and 4 text labels. When locals were playing the game, users produced an average 13 targets: 7 images and 6 text labels.

In general, locals produced more targets during the game than non-locals, but the ratio of images to text labels produced was about the same for locals and non-locals. However, a larger percentage of the images produced by locals were in response to requests from the Realise website. While such images only accounted for around a third of all images produced, this indicates that locals favored priority targets more than non-locals.

The results indicate that the systems were most efficient when locals were playing the game and non-locals were using the website. This matched expectations because the respective systems were designed for these types of user. It is also not surprising that locals favored priority targets more than non-locals as these were expected to be more challenging than regular photo targets. Interviews with non-locals support this opinion as many suggested that they found many priority targets too hard to confirm because they did not know the area well enough.

In total, 146 requests were made by Realise users. However, only 33 of these requests (23%) were satisfied. This is a significantly poorer ratio than in the previous trial before the subjective reward system was included.

If we break this down to locals and non-locals, we find that locals using the Realise website made 96 requests, of which 16 (17%) were satisfied. Non-locals using the Realise website made 50 requests, of which 17 (34%) were satisfied. While the ratio between requests and satisfaction changes for locals and non-locals, the actual number of satisfied requests is about the same. This might indicate that locals knew more places to request photographs from than non-locals. However, it is possible that the duration of play simply did not allow more than around 16 requests to be satisfied when the ratio between Realise users and EyeSpy players is the same. However, if this is the case, the subjective reward system should have prevented more requests from being made until previous ones had been satisfied, leading to a higher percentage of satisfied requests. Why then, did the system become flooded with requests?

Through examining the trial data and interviewing participants, a flaw in the design of the subjective reward system seems to be responsible. Realise users generally spoke very positively about the request system (more so than in the previous trial) and had a great enthusiasm for it. This may stem from the changes in giving users more control over the scoring system. This can also be seen by the greater number of requests being made. However, the EyeSpy game was not able to keep up with the larger number of requests and the subjective reward system should have dynamically prevented more requests being made to maintain equilibrium between systems. The problem lies in the way that requests are created.

A Realise user must enter a bounty on all requests that are made, and this bounty cannot exceed the points allowance available to that user. However, the points are not transferred from the allowance until the user chooses to confirm a photograph once it has been returned. This means that users can use their full allowance on any request they make, but will not have enough points to confirm all the ones that return. This means that, in terms of making requests, the points become meaningless to the Realise users. It was not until later in the trial, when photographs started returning, that website users realized their mistake and were unable to confirm more than a few photographs. A more appropriate design would have been to lock points when a bounty was made so that, though not transferred to the EyeSpy players, they could not be taken into account when making other requests. While this would not have prevented the flood of initial requests, a similar number of requests would all have had very low bounties and Realise users would have to expect that it would take a longer time to return them than for requests with higher bounties. This would have been the correct behavior for the system.

Even with this issue of bounties, the subjective reward system should still have limited the allowance of requests that could be made. So why did this not work as well as expected? There were two problems that led to the failure. The first was that the the system was not aggressive enough in reducing the number of requests that were allowed. Although the allowance did come down every day (until no requests were allowed at all in the final few days of each round), it should have dropped much faster. This could be solved in future versions by making the system more aggressive.

The second problem was to do with the way that activity was measured. In order to determine the allowance of requests for the current day, the system looked at the previous day and noted how many requests had images returned and how many were still waiting on photographs. However, this is a flawed perspective because it does not take into account that many of these images were not confirmed by the Realise users (in some cases, because they did not have any points left to pay the bounties). And, when a request has no confirmed images, it remains in the system so that EyeSpy players can keep trying until an appropriate photograph is created and confirmed. Additionally, older requests with higher bounties are favored when assigning priority targets to EyeSpy players. This means that many of the first requests were the only ones being circulated in the game until every EyeSpy player had tried to create a photograph for them. This means that the earliest requests with the highest bounties had a fairly high number of photographs, even though they should have been confirmed and closed after the first one was returned because it matched the Realise user's description. This issue could be solved in future versions by basing activity on confirmations, rather than images created.

These trials also saw the first abusive use of the systems. One Realise user made two requests for different parts of the human anatomy and assigned very large bounties to them. However, none of the EyeSpy players returned an image for these requests and all who received it chose to permanently delete it, despite the high bounties being offered. This flagged the user to the game designers and he was politely asked to refrain from such activity, but allowed to continue in the trial. Another user made a request for 'Stone Vagina by Gregory building.' While this might at first appear abusive, this was a colloquial term used by students to refer to a particular statue on campus. As such, players who were familiar with the name and the statue took pictures of it and successfully received points from the Realise user. However, others who were not familiar with the name chose to permanently delete the target, believing it to be abuse in the system. When the person who created the request was asked about it, they responded by saying they had not even thought about it being taken the wrong way, believing that everyone on campus referred to it by that name. This demonstrates that subjectivity can also be an issue in flagging and abuse systems. This, again, shows the importance of consensus rules in games with by-products.

There were also several requests that were not location based and required subjective opinion from the players. Some of these, such as "River Clyde" and "An Open Top Bus" were specific, but open ended. It was left to the players to decided which part of the river Clyde to photograph and which open top bus. There were also some requests that included 'fuzzy' descriptions, relying on the players to figure out what was meant. For example, one request was for "The slightly strange symbol on the Physics building." This request relies on the player to determine that a physics building is a place in which physics is taught and studied, and to determine which of the symbols on that building is the strange one. It should be noted that all three of these requests were successfully confirmed. In general, requests that featured subjective choice or fuzzy descriptions had higher bounties associated with them. When the requesters were interviewed, they suggested that this was because they were asking more of the players, so a higher bounty was suitable.

One request was notable because it was so far from the regular game area. While other trials had seen some targets that were not as close to the main University campus as others, most were within 0.4 miles of the main building. However, one request required a player to travel 1.6 miles from the main building, such that she was significantly far out from the University campus. This request also had a fairly low bounty. The requester indicated that he thought it might be on the way home for someone and wanted to test the limits of the system. The request was satisfied by one player who said that it was not that far out of her way on the walk home and she did not want to disappoint someone if they wanted the photograph. This sentiment of trying harder to make good photographs for priority targets was shown by many of the players, suggesting that points were not the only motivation.

In the previous version of the systems, EyeSpy players produced an average of 24 targets during the game (15 images and 9 text labels). This is higher than the version which includes the subjective reward system. This drop in number from the older versions may be due to the changed limits on game-play in the newer version (previously, players could make and confirm five of each target type, but in the newer system they could only make and confirm three of each target type). While fewer tags were produced by the players overall, it should be noted that each player produced closer to the average amount. In the previous versions of EyeSpy, some users (generally those who were high in the leader board) produced significantly more tags than other players meaning that a small number of users accounted for a large number of by-products. Players who could not keep up with the high scorers' level of play tended to stop playing entirely. This is not the ideal model because we are relying on a small number of people to power the system. It is better if a lot of people do a smaller amount, but always feel competitive and motivated to keep playing. This is more sustainable over a longer period of time and means that the by-products being produced represent a more objective output.

It should be noted that, in these final trials, no users complained of the saturation problem and there was no evidence of a large amount of repeated targets entering the system.

We have now discussed some of the initial findings from testing the subjective reward system as well as solutions to some of the major problems that were uncovered. However, there are still other improvements that might be made to the systems which are more minor or could provide some additional benefit. These will now be discussed.

5.3.1 – Possible Improvements

A great deal of time has been spent in preventing cheating in our systems and one of the methods used to achieve this is maintaining anonymity and minimizing the channels of communication. However, there is a possibility that allowing website users to comment on the photographs that they receive could be beneficial to both systems. Website users could also rate the photographs they receive. This could allow the game players to see why they have not received the bounty for a photograph or to provide them with bonuses when they get a good rating. It might also be beneficial to allow the game players to comment on the requests from the website if they feel they are too difficult to understand or carry out. This will prevent website users wondering why they never get any photographs back for a particular request.

Opening up the requests between website users might also be beneficial. At the moment, Realise users do not know how many points people are allocating to bounties. This makes it difficult for a user to know how many points to use when they want a request to be dealt with before other users, or just what the average bounty is. Allowing users access to other requests would also prevent people making the same requests themselves and would open up the possibility of allowing users to pay the bounty (assuming they could afford it) on other users' requests if they had not been confirmed. This would save time because an immediate response would be possible, rather than having to put a new request into the game and wait for the EyeSpy players to deal with it. If such a system was to be put in place, it would be best if the requests were kept anonymous and it was made clear to all users that their requests could be read by the others.

An additional point is perhaps how the Realise users organize the photographs which are returned to them. A search system is provided that is limited to the returned photographs, but the request text for each photograph is not used by the search engine. Obviously, this would make it easier to find photographs, and it is also possible that the requests should be used by the general search as well. It would also be beneficial for Realise users to be able to tag photographs themselves. The Realise users could then include their own naming schemes to make it easier to search through photographs. If this was extended to the whole site and enough users independently used the same tag for a photograph, it could be attached to the photograph's geographical location for use in general searches for all users.

So far, the EyeSpy game and Realise website have only been designed at a local level. The current implementation would probably not scale to a larger area, or globally. This is important because requests like "University Union" might mean different things outside of the current play area of the University of Glasgow. However, some simple changes could be made to allow this to happen. In the EyeSpy game, the client could inform the game server of its current location, allowing the server to only send targets that were appropriate to that area. In the Realise website, users could restrict their searches to particular geographic regions. Similarly, requests could be augmented to only apply to specific geographic regions. This would allow both systems to be used on a global level. Of course, this has not yet been tested and may introduce problems. For example, will the player densities in different parts of the world limit the requests for particular areas? Will there be a good ratio between players and web site users such that both systems can function adequately? While the subjective reward system's scaling properties could potentially deal with such issues, it too would need to be changed to apply only to specific geographic regions. This means that in one area, the system might allow you to make a lot of requests and in other areas very few or none. The daily points allowance could also be restricted so that each region has its own points allowance for every user, but this may be more conceptually difficult for users to understand.

Another potential problem of scaling could be the leader board. While it would be easy to restrict the leader board to only those playing in the same geographical region, if there were a large number of high scoring players in that region it might be off-putting for new players. One possible solution to this would be to reset the scores every couple of weeks after declaring the top player to be the winner. However, this constant resetting and starting from scratch might put people off replaying the game. One solution to this could be the inclusion of a leveling system where all players start at level one and are only competing against other level one players. Once you have come top in your own level, you will be promoted a level for the next round and only be playing with those people, and so on. This might instill a level of pride and encourage people to continue playing after each round. Of course, there would always be a top level where winning would not promote you any higher.

Another possibility for encouraging continued play would be linking EyeSpy and Realise accounts for each person. This could provide a loyalty system where players are awarded a larger allowance to request photographs by being good players in the EyeSpy game. This would also allow another means to prevent cheating as users could be prevented from getting their own requests, and perhaps from making requests in the same areas they play in.

Another issue which did not come up during the trials is that targets will become less relevant over time. It might be hard to confirm a photograph that was taken at a different time of year or a photograph of a building that has since been knocked down. Since the photographs in Realise have dates displayed with them, they will still be useful to show the changes to a location over time. However, in EyeSpy, it might be best to remove targets from the game after they are a few months old. It also means that players who no longer play the game will not continue to 'win' without playing.

As has been discussed already, the EyeSpy game should really have a dedicated system for flagging targets which are inappropriate. While this was not implemented due to time constraints, the benefits are clear (as shown from the use of the 'permanent delete' feature in the last trials). An important point to note about flagging is that the Realise flagging system currently has no limit on the number of flags a user can raise. Although one user flagging lots of photographs would not affect things too much due to consensus rules, a lot of people working together could effectively prevent all images from being displayed to other users. A simple solution to this potential problem would be to limit the number of flags that users can raise each day.

A last point to make about the Realise website is that the original design called for a greater use of the map component. Due to time restrictions, this was limited merely to displaying a map for each photograph to show where it was located. However, it might be more beneficial to display all the images on the same map, making it easier to see how they relate to each other. The results of a search could restrict the images shown on the map. It would also be possible to only show images from a certain date range or which have been confirmed a certain number of times. Another possibility would be allowing the user to create a path on the map which they expect to travel along, then return a series of the most recognizable photographs along that path showing each one's location on the map. This could help with navigating a new area.

5.4 – Conclusions

In this chapter, the concept of a subjective reward system was used to augment the existing notion of mutually reinforcing systems. The subjective reward system was intended to give the Realise users more control over the points that EyeSpy players received when responding to website requests. This was intended to shorten the time taken for Realise users to receive their photographs as well as shorten the time taken for EyeSpy users to receive their points.

It was shown that mutually reinforcing systems allow for subjective by-products in human computation games and that this was something that should be further explored. The subjective reward system helped with this by allowing Realise users to prioritize their requests and to put the decision of awarding points entirely within their hands and not the game players. The subjective reward system also allowed the removal of the unpopular timer based scoring system for priority targets, while still maintaining the benefits of a variable points system for these targets.

The subjective reward system was based on giving the Realise users an allowance of points to create bounties for their requests, as well as an allowance of requests that could be made each day.

It was also noted that the systems had largely been trialled on people who knew the area well. In order to determine if use of the systems would be affected if this was not the case, the final trials used an almost equal amount of participants who were familiar with the area and those who were not.

The subjective reward system also introduced, for the first time, a means for players to gain points that did not rely on other players. This means that a player could be the only one in an isolated area and still get points. This makes sense for priority targets as relying on other players when there are none would prevent images being returned to the website.

It was also shown that some players worked harder on priority targets because they knew that someone wanted those specific images. This level of motivation did not occur before mutually reinforcing systems were used and this may be an important advantage of such systems.

The trials of the subjective reward system highlighted some major flaws in the implementation of the concept. The principal goal of these augmentations was to shorten the time taken for requests to get responses and to prevent the Realise users flooding the system. However, neither of these issues were successfully solved. Analysis of the trial data and interviews from participants highlighted the reasons for this failure and some potential solutions were presented. Some other potential improvements to the system were discussed which were not focused on fixing these problems. However, due to time constraints, these were never implemented and no further incarnations of the systems were made.

The augmentation of mutually reinforcing systems to include the subjective reward system showed a potential means of tackling some of the major problems that were still present in EyeSpy and Realise. This refinement of mutually reinforcing systems could be used to help the principal aims of EyeSpy and Realise and could potentially be used in similar mutually reinforcing systems. By helping the goals of both EyeSpy and Realise, the subjective reward system has contributed to both ***RQ1*** and ***RQ2***. However, the trials of the systems highlighted some problems that would need to be fixed before a successful demonstration of the concept can be shown.

The next chapter will detail some analysis of the systems that did not directly contribute to future versions, but was still important for measuring their successes and failures.

Chapter 6
Evaluation

The concept of mutually reinforcing systems has now been presented and demonstrated using a mobile game with by-products and a tourist website. Evaluation of this concept is difficult because each system has a goal to improve the other system. However, each system does have its own independent goals. While some of these goals cannot be achieved without the other system, they can still be evaluated without needing to address the success of the other system. If each system is fulfilling its independent goals (including those which require the other system), then we can say that the systems have achieved an overall success by working together as mutually reinforcing systems.

EyeSpy's original goal was to be an enjoyable game that people would want to play and which would produce useful by-products as a side effect of that play. This goal is shared by many games with by-products. However, subsequent versions of EyeSpy expanded on this original goal to include the creation of specifically needed by-products in addition to the broad ones that were already being produced. The by-products themselves were intended to be a set of photographs and text descriptions that were geographically referenced and time stamped to provide contextual information about mobile environments.

The goal of Realise was to allow users to browse and search for images based on location information and to let users request new photos when no satisfactory ones were available. The website concept was aimed at tourists to allow them to pre-visit and post-visit locations.

This chapter will attempt to determine to what level these goals were met and the impact this has on the overall success of mutually reinforcing systems.

6.1 – Evaluating EyeSpy

A method of assessing the success of games with by-products has already been suggested by von Ahn and Dabbish [50]. This focuses on the efficiency of producing by-products, the length of appeal that the game has for players and how much each player contributes to the creation of by-products.

In the case of EyeSpy's general creation of by-products (rather than specifically requested ones), it is difficult to determine the game's efficiency because there is no similar system to compare it to. While there are other games with by-products, there are none that are attempting to solve the particular problem which EyeSpy tackles. If EyeSpy's throughput was compared to other games with by-products which did not operate this way or which dealt with different problems, this would not be an accurate measure of EyeSpy's success because different problems require different solutions, some of which might necessitate slower throughput. Also, different problems may consider different levels of throughput to be efficient. Therefore, in the case of EyeSpy we can only measure the potential usefulness of the by-products rather than the efficiency with which they were created.

A metric called 'system gain' is suggested by Chen et al. [124] for evaluating the performance of games with by-products. System gain considers the average time required to complete each task weighed against the average number of outcomes produced by each task. However, the time taken to complete tasks is not a major factor in EyeSpy because the game is designed to be played sporadically over the course of the day. The time between receiving a task and completing it could be hours or days. Also, the tasks in EyeSpy do not independently create outcomes: it is the matching between the tasks of confirming text descriptions and photographs in the same area that create the outcome. The outcomes will also grow over time because a text description can be used to match many photographs. Any new photographs or text descriptions in an area will add to the existing set creating new matches for all the existing by-products in that location. These two issues mean that system gain would not provide a meaningful measure that can be weighed against other games with by-products which do not operate on these principals.

The enjoyability of EyeSpy is easier to determine as we can simply ask the players how fun they found the game to be. While a quantifiable measure of enjoyment is difficult to produce (it would be hard to say that EyeSpy is more or less fun than other mobile pervasive games due to different styles of play), we are able to determine how many of the trial participants reported enjoying the game and said they would like to play it further. We can also ask players to explain what they enjoyed and disliked about the game in order to better understand the level of the game's appeal. While von Ahn and Dabbish suggest that a game's length of appeal for each player is also important, this is difficult to determine because the trials of EyeSpy were not on-going and existed only for finite periods of time. The long term enjoyability of EyeSpy would require time and resources that were not available during our research. Another issue lies in the method suggested by von Ahn and Dabbish for determining the length of appeal for games with by-products: they suggest that a measure of the total amount of time the game is played averaged across all the players will produce an 'average lifetime play.' However, using this metric to compare games that are played in single, longer sittings with those that are played for short periods spread throughout the day would not produce an accurate measure of which one had better longevity and extended appeal.

How much each player contributes toward the creation of by-products is a measure of the system's efficiency multiplied by the amount of time a player is likely to play the game. However, as has just been discussed, efficiency and appeal are difficult to measure in EyeSpy with the current data that has been collected. But we can still use interviews to measure the level of enjoyability, and we can examine the by-products to determine their level of usefulness. Measuring efficiency is something that may have to wait until systems are built which are comparable to EyeSpy.

The conditions under which the EyeSpy game was played must also be taken into account. EyeSpy was only played as part of a trial and was never given a public release. The players always knew that they were taking part in an experiment. How much this situation may affect the results from these trials is debatable [125], but trialling the game in this way allowed for a deeper analysis of play through monitoring and interviews.

EyeSpy involved drawing on local knowledge in order to successfully capture images that would be useful for navigation. However, usefulness can change depending on subjective opinion and context. For instance, we can imagine how one symbol may mean very different things to different groups of users. For example, the swastika is used as the symbol for the Nazi Party but is very similar to the manji symbol which is commonly used in Japanese maps to mark temples. Human computation is not just about producing 'objective' results, but can also be about using subjective understandings to produce content that draws upon subjective and creative knowledge. Even simple judgments on what people can find in a local area could depend on the cultural positioning of players (e.g., as pedestrians in the city rather than drivers). In EyeSpy, exploiting the local knowledge of participants simply means producing more culturally relevant images. In this sense, EyeSpy is truly adaptive to its context of use and the by-products' usefulness must be considered from this viewpoint.

In order to determine the usefulness of the by-products produced by EyeSpy, the next section will discuss how they were tested.

6.1.1 – Testing Game By-products

Assessing the quality of EyeSpy's by-products requires examining the potential for their use outside the game as stand-alone resources. This section explores the results of further experiments that were conducted, focusing on the images generated by EyeSpy, in order to determine if the photos were recognizable, findable and useful for navigation. It should be noted, however, that recognizability and findability are only two components of navigation and the experiment was configured to test only these two aspects of navigational practices.

The generated photos from EyeSpy could be compared with other collections that could conceivably be used in a location-based service delivering images to users. Of course, there are many different uses and contexts in which a navigation system could be employed [126]. Each different usage could potentially require subtly different types of images. Broadly, however, all images used for navigation share the ability to be quickly identified by individuals in situ. The more obscure an image is, and the longer it takes for an individual to visually link a photo with the scene being navigated, the less useful it is likely to be for this purpose.

In order to test EyeSpy's by-products, a series of tests were conducted by Stuart Reeves comparing EyeSpy's photo set with sets derived from geographically referenced images available on the Flickr website.[1] Flickr was chosen firstly in response to earlier work suggesting the use of Flickr images for navigation [127], but also because of its worldwide coverage. While the EyeSpy image set was small in comparison, it was relatively dense along particular routes and, of the image sites that were looked at, only Flickr had a comparable photo density in order to permit a fair test. Google Street View [108], was also considered as it too has a dense coverage. However, this coverage only includes streets and roads. This would mean that areas that were not drivable would not be included and this would not match EyeSpy's by-products well. As such, Google Street View would not make for a good comparison with EyeSpy's largely pedestrian-generated images.

The tests differed from the navigation tasks within the EyeSpy game because of the specific tasks involved. While EyeSpy validates its images by requiring the players to navigate to them, this is under the assumption that the users already know where the images were taken based on their content. In the case of the tests conducted to determine the usefulness of the by-products for navigation, users were given a route to follow and asked to determine when they had reached the location where given photographs were taken. This is from the viewpoint that users do not know where each photograph is originally from but can easily identify them when in the correct vicinity. This addresses a problem highlighted by Brown and Chalmers that showed that tourists could not determine where a landmark was, even if they were in the correct location and using maps and guidebooks [42].

These tests were carried out on the initial version of EyeSpy before it was ported to the iPhone and before the concept of mutually reinforcing systems was introduced. The tests were carried out at this time to determine the level of success the game had in producing useful by-products. This helped inform the designers of improvements that could be made to the game.

Even though these tests were carried out before its inception, the Realise website shares similar modes of use with navigation tasks. Realise is intended to allow pre-visiting and post-visiting [42] of locations. In the case of pre-visiting, users may want photographs that will help them identify a place or landmark when they come to visit it. In this sense, recognizability and findability are good indicators of the by-products' usefulness for Realise.

[1] `http://www.flickr.com`

Image Retrieval and Coding

The initial version of EyeSpy used Wi-Fi fingerprinting for location comparisons, but actual geographic locations were required in order to compare the images to those from Flickr. To do this, the Wi-Fi fingerprints were resolved to their corresponding GPS locations using the Skyhook system, as described in Section 4.1.2. Based on the location of each of our images (which numbered 257 at the time of the test), a random image was chosen from the ten geographically closest images available on Flickr which themselves contained GPS meta-data. The randomly selected proximal Flickr photos contained significantly higher proportions of people and transient objects. Further to this, the EyeSpy image set contained a higher proportion of signs, shops and doors/boundaries, revealing how the game produced greater numbers of photographs of images likely to be more useful to navigation in terms of the criteria of recognizability and findability.

Navigating with the Image Sets

The first experiment involved the construction of two routes within the game area (of length 530m and 800m respectively). These can be seen in Figure 6.1. 16 EyeSpy and Flickr photos were gathered for each route, resulting in four sets of photos: R_1E, R_1F, R_2E and R_2F. We employed a two factor experimental design with replication in order to test the various permutations of routes. 10 participants (7 female, 3 male) with limited or no knowledge of the game area were recruited, and each was provided with a simplified map of the area (with street names erased) which used circles to indicate an approximate location for each of the photographs for that particular route. Note that, in order to avoid bias, these locations were approximate enough to geographically cover the true locations of corresponding photos spanning the Flickr and EyeSpy versions of the routes. Thus only two maps were constructed.

Participants were also provided with the photos relevant for their routes. Each participant was then sent out individually to walk a selected pair of Flickr and EyeSpy routes, the objective being to locate where they thought the photos had been taken from. Participants were recommended to spend no more than half an hour to complete each route, and were told that the order in which photos were confirmed did not matter.

As a result of this experiment, it was found that participants were able to identify the locations of 95% of EyeSpy photos included in the two routes (91% and 99% for R_1E and

Figure 6.1: The routes used in the testing of EyeSpy's by-products. Route 1 (seen in purple) was 530m in length and Route 2 (seen in orange) was 800m in length.

R_2E respectively) compared to 49% of the Flickr photos (54% and 45% for R_1F and R_2F respectively).

On average, participants took 25 minutes and 11 seconds (standard deviation of 8 minutes and 9 seconds) to complete the Flickr routes, whereas on average it took 17 minutes and 25 seconds for EyeSpy routes (standard deviation of 6 minutes and 16 seconds). These results suggested that photos generated from EyeSpy were more frequently and rapidly located than those from the Flickr set.

Selecting Navigation Images

In a second experiment, a further 16 participants were recruited. Each was presented with a randomly ordered collection of all the photos from the previous experiment (i.e., 32 Flickr photos and 32 EyeSpy photos). Each was then asked to indicate which photos he or she felt would be most appropriate and useful for tourists navigating around a city they were unfamiliar with. An example of some of these photograph pairs can be seen in Figure 6.2.

It was found that 61% of the 32 EyeSpy photos were chosen as appropriate for navigation, whereas only 20% of the 32 Flickr photos were selected, further confirming that photographs generated by EyeSpy were preferable for navigation.

Text Tags as Descriptions for Images

It was originally intended that the game's design would result in successful photographic by-products that offered the possibility of associating those images with relevant search terms culled from nearby text descriptions of the area. Although these experiments mostly focused on the images produced by EyeSpy, the textual descriptions generated by the game were also examined in order to discover whether they could be effectively reused in this way. A lower density of text descriptions compared to photographs resulted in less than 20% of the closest text descriptions providing relevant labels for images. There were also photographs for which there was no nearby textual description. This means that the textual descriptions would have to apply to larger areas in order to cover the majority of photographs that were created. These issues affected the way that text descriptions could be used to search for photographs. Rather than returning specific matching images, the by-products were better suited to returning locations, then showing all the photographs from each one.

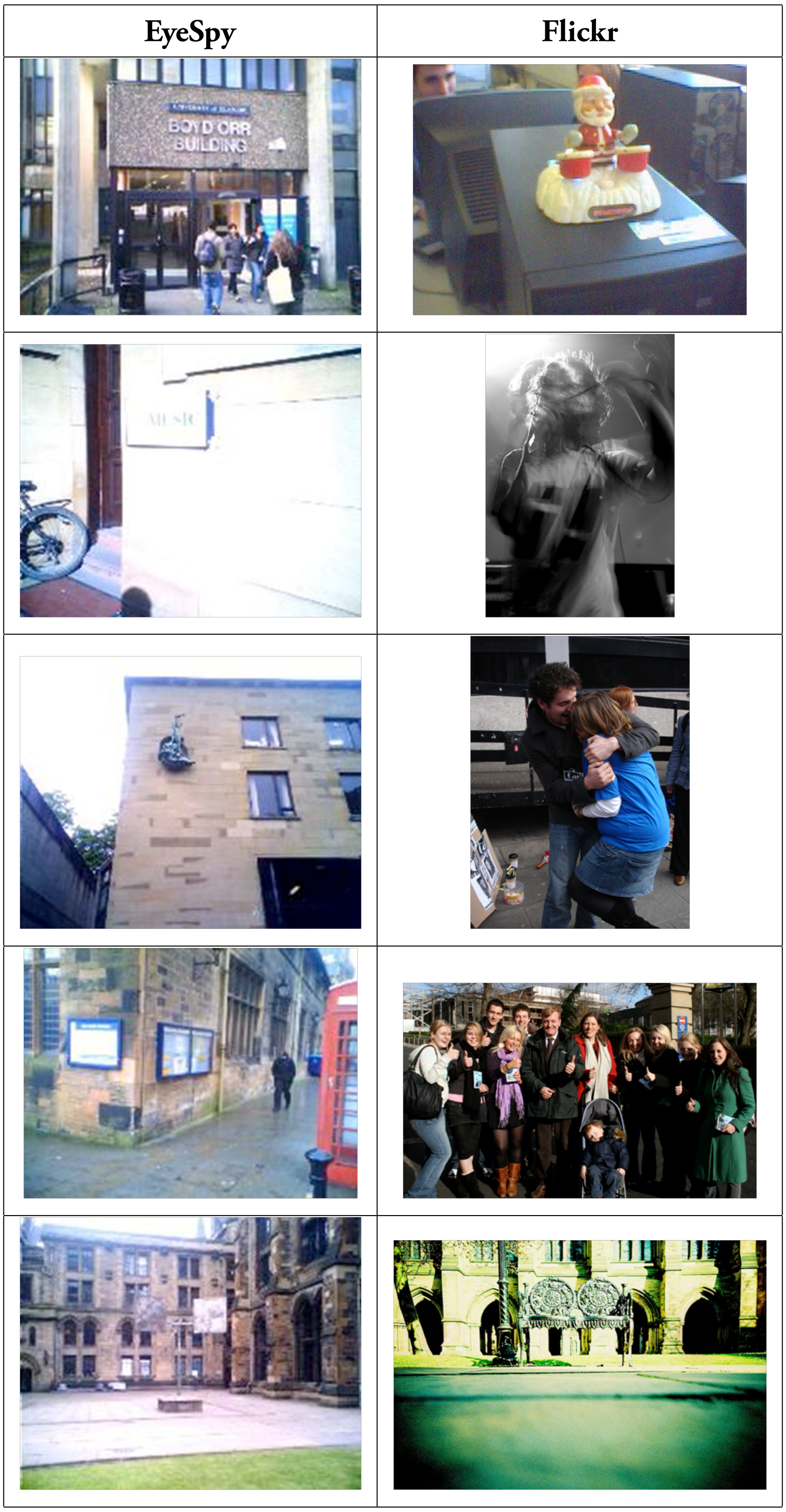

Figure 6.2: A selection of photographs used in the navigation task. On the left are photographs produced by EyeSpy. On the right are photographs from the Flickr website. Each Flickr photograph was reported by the Flickr website to be at the same approximate geographic position as the EyeSpy photograph to its left.

6.1.2 – Testing Game Enjoyment

The enjoyment of EyeSpy is a measure of whether or not the players found the game to be fun. In order to assess this, 54 players were interviewed about their experiences with the game, spanning across 5 separate trials and 3 different versions of EyeSpy. The three versions included one without mutually reinforcing systems, one which included mutually reinforcing systems but not the subjective reward system, and the final version which included mutually reinforcing systems and the subjective reward system. Each version of the EyeSpy game was associated with a standard set of questions that were used when interviewing the trial participants. This ensured that the same ground was covered in all interviews. However, follow up questions were asked when appropriate to learn more about each participant's answers.

The questions were split into several topic areas which included what players enjoyed and disliked about the game, what players found easy and hard to do within the game, whether or not players managed to cheat, the places and times of day that players used the system, how aware the players were of other users, how much players were willing to change their daily routines to play the game, how comfortable the players felt playing the game in public and whether or not the players would choose to play the game again.

Additionally, players were presented with a random selection of the photographs they had created during play and asked to talk about why they made them and if they could recall their motivations for the particular ways they had set up their photographs.

In total, 96% of all the players reported that they enjoyed the game, with 83% wanting to play the game again. Additionally, 91% said that they would recommend the game to other people.

The initial version of the game before the inclusion of mutually reinforcing systems involved 18 players who were interviewed (there were another 9 participants who were not interviewed). 17 of the 18 players (94%) reported that they enjoyed the game, with 11 (61%) wanting to play the game again. There were 13 (67%) who said that they would recommend the game to other people.

The version of the game that included mutually reinforcing systems but not the subjective reward system involved 10 players who were all interviewed. 10 of the players (100%) reported that they enjoyed the game and 9 (90%) said that they would like to play the game again. 10 of them (100%) reported that they would recommend the game to other people.

In the final version of the system, 26 of the 31 players were interviewed. 25 of the 26 players (96%) reported that they enjoyed the game, with 25 (96%) saying that they would like to play the game again. 26 players (100%) reported that they would recommend the game to other people.

Although the number of people involved was small, these results are fairly positive showing a high approval for the game amongst its players. The increase of players wanting to replay or recommend the game after the inclusion of mutually reinforcing systems also indicates that the game had a more lasting appeal because of this feature. This would indicate that mutually reinforcing systems made the game less monotonous.

6.1.3 – Testing the Creation of Specifically Requested By-products

The results from the previous section indicate that EyeSpy was enjoyed by its players and that the inclusion of requests from another system did not significantly degrade that enjoyment. In fact, the results indicate that the game became less monotonous because of the requests system. However, the motivations for players to respond to requests should be discussed.

Of the 36 people who were interviewed after the inclusion of mutually reinforcing systems, 34 of them (94%) said that they felt comfortable with other people making requests of them within EyeSpy. Even knowing that part of the game-play was doing work for other people, 21 of the players (58%) said that they would buy the game if it was publicly released. Furthermore, 21 of the players (58%) said that they would be fine with their photographs being used for some commercial reason, even though they themselves would not receive money. Again, 21 of the players (58%) said that if the requests came from a company rather than the Realise website, it would not prevent them from playing the game or change their attitude during play. This implies either that a significant proportion of the players did not feel a strong sense of ownership about the photographs that they created during play, or that they were altruistic enough to want to help other people. Some of the participants did include exceptions, saying that they were fine with their photographs being used commercially or by companies but not if it included personal information. Others said that they would like to be attributed as the creator of the photograph. A group of players also said that, while they generally supported the use of their photographs for commercial purposes, they would like to know what it was being used for in case it was something they did not support.

It is worth noting that, in a publicly released version of the game, the players might have no knowledge of how the system works. People might play the game and not know that they are carrying out human computation for some other system. It has been stated by von Ahn that the players of the ESP Game were simply told to guess what the other users might be thinking based on a shared stimulus [6]. Users were not instructed to guess labels for images at all, even though that was the ulterior motive. This might be part of the reason for the success of such systems. If someone plays a game for entertainment and that just happens to benefit another system, it may not be beneficial to explicitly tell the player that this is happening. However, as shown above, some players may be motivated by altruism as well as (or instead of) fun and not telling them about the by-products of play might negate this eventuality. It is also worth noting that not informing the players of the by-products could be considered exploitation. Therefore, it would seem that there are arguments to be made both for being up front about the human computation and for hiding the ulterior purpose of games with by-products.

While the motivations for players to respond to requests is important, examining the success of the responses is required to determine how well the concept of mutually reinforcing systems worked. This will be addressed in the next section.

6.2 - Evaluating Realise

The goals of the Realise website are largely about keeping the users happy. This is because it is the users who decide what images they want to search for and request. In determining the success of the website, the subjective opinion of the users is, therefore, of great importance.

While there are other metrics that might be used to determine the success of the website, the research of this dissertation has focused on satisfying the individual users and, as such, their opinion is what matters most in the evaluation of mutually reinforcing systems.

There are specific opinions from the Realise users that contribute toward measuring request satisfaction. One of these is the quality of the request responses and how well they match the requests. The other is how long it took the EyeSpy players to respond. Additionally, the more general idea of whether the users reported enjoying the website is important in assessing their overall satisfaction with the system. The next section will address these issues to determine whether the website met its goals.

6.2.1 – Testing the Realise Website

Although the time taken for EyeSpy players to respond to requests is a requirement for measuring request satisfaction, the incorrect implementation of the subjective reward system, as described in Section 5.3, makes it difficult to measure how long the players would have taken to produce acceptable responses for the Realise users. This means that a full determination of request satisfaction will not be possible without future research. As such, the analysis of the Realise website must focus on whether or not the users enjoyed the website, would use it again and would recommend it to other people. It is also important to assess whether or not the Realise users were happy with the quality of the responses.

The requests all required a certain level of subjective choice from the EyeSpy players in their responses, but some required greater levels of creativity and choice. Requests which required more subjective behavior from the players were either non-location specific, specified an object-type rather than a specific object, specified a location but not a specific subject, specified a very large area as a subject or specifically asked the players to make a choice.

An example of such a request was, “Subway that isnt across the road from Hillhead or in the GUU but is near the uni.” The players had to first determine that this request was for a picture of a fast food chain restaurant called ‘Subway’ and not for a station of the underground train system that exists in Glasgow and which is also called the ‘Subway.’ This first step draws on local knowledge rather than subjective choice to determine what type of ‘Subway’ the requester is referring to. This is made more complicated because “Hillhead” in this context is in fact a Subway Station, though not explicitly described as such in the request. The station is located within an area of Glasgow called 'Hillhead' but used with the phrase, "across the road from," makes it clearer that a specific location is being discussed rather than the larger area. The "GUU" refers to the Glasgow University Union, which does not contain a Subway Train Station but does contain a restaurant of the Subway fast food chain. These pieces of information allowed the players to determine that it was a fast food chain restaurant that was needed and not a train station. However, the request does not refer to a specific restaurant of the chain. Instead, it describes existing locations of the chain restaurant that should not be used and that the location must be near the "uni" (referring to the University of Glasgow). At this stage, the players must now draw on their local knowledge to determine locations of the fast food chain that will satisfy the request. However, they must interpret for themselves how "near" the location must be. In this particular instance, there are two Subway restaurants that satisfy the request and were considered 'near' by the EyeSpy players. Two players opted for one of them and another two players opted for the other. A photo of each restaurant, returned by the EyeSpy players, can be seen in Figure 6.3. It should be noted that one of the photographs returned included both a Subway restaurant and a Subway Train Station. Technically, both the station and the restaurant satisfied the requirements of the request. When asked about the choice to include both in the photograph, the player said, "Yeah, I mean, I realized it was probably for the food place, but when I saw I could get both in the shot I just thought it was better to do both, just in case."

Another example of a request which invoked player subjectivity was, "A nice shot of the pillars of Wellington Church on University Avenue." While the location and focus of this request are fairly objective, the Realise user has specifically included the need for the returned image to be "nice." Exactly what constitutes a nice image of the landmark is left up to the EyeSpy players. Two photographs were returned by EyeSpy players and both were accepted by the requester. These can be seen in Figure 6.4.

A final example of requests requiring subjective choice from the players was, "An Open Top Bus." While this may at first seem quite objective, it does not refer to a specific object or location. There are various open top buses in use in the Glasgow area (even if their basic design is much the same) and their whereabouts changes from moment to moment as they are driven around the city. This requires the players to consider where they are likely to

Figure 6.3: Two photographs showing different locations of 'Subway' restaurants that satisfied the requirements of a request made by the Realise website. The top photo also includes a 'Subway' Train Station which the player included in the shot just in case he had misunderstood what type of Subway would be needed.

Figure 6.4: Two photographs of the same building, each with a different player's interpretation of a 'nice' composition, as required by the request.

see such a transient object and when it is likely to be there. This draws on local knowledge and experience to know what open top buses there are and how best to take a photograph of them. The particular buses that were returned by the EyeSpy players were of different designs and taken in different locations at different times. One player reported knowing that the bus would be at the location at that time every day, but the other player admitted to being more opportunistic and photographed the bus because he happened to see one and knew from his priority targets list that one was needed. These photographs can be seen in Figure 6.5.

It should be noted that transient objects were not commonly used as the subject of photographs within the main EyeSpy game, as discussed in Section 3.3. This is a result of the game dynamic within EyeSpy which requires the photographs to be recognizable and findable in order for players to receive points. Because transient objects will rarely be at the same location, it would be much more difficult for players to confirm these kinds of targets unless the surrounding area was also recognizable and findable. However, because the website users are responsible for validating the photographs from requests, these limitations are no longer present to the players. The same concept also applies to requests for general objects, even when they are not transient (such as a 'squirrel' or a 'tree') as these may not be especially recognizable on their own, but could still be required by other systems wishing to use the dataset of by-products. This demonstrates how the Realise website extends the potential by-products that EyeSpy can produce by removing some of the limitations that are required by the internal validation techniques.

Each of these requests and the returned photographs demonstrates the level of human decision making and subjective choice that goes into responding to some requests, as well as drawing on experience and local knowledge. However, some requests were more specific, requiring less personal input from the players. An example can be seen in the request, "Buchanan Street looking down towards st Enoch centre from the top of the stairs leading up to the Glasgow Royal Concert Hall." This request suggests to the player that as much of Buchanan Street as possible should be captured as the subject and a fairly exact position from where to take the photograph. One of the returned photographs for this request can be seen in Figure 6.6.

It was also found that certain requests returned very similar photographs from multiple

Figure 6.5: Two photographs of different open top buses taken by different players in different locations at different times. Each was responding to the same request.

Figure 6.6: An example of a photograph requiring less subjective input from the player. The photograph shows Buchanan Street, taken from an exact spot that was specified by the request.

players. This could also indicate that these requests required less subjective choice from the players. One example is a request for, "A picture of the Wolfson Medical Building from the steps of the Boyd Orr across the road." As can be seen in Figure 6.7, two of the returned photographs are almost identical despite being taken by different players at different times. A further example can be seen from the request, "ashton lane from entrance to radio." While the photographs returned for this request (as seen in Figure 6.8) are not as identical as the previous example, many elements are shared across the photographs, such as the 'stuff your face' sign, the black chalk board on the left of the street, the reversed 'Grosvenor' sign, the black street posts and the cobbled road. It should also be noted that each photograph is taken within the same three meter radius. While Ashton Lane continues in the other direction and the request did not specify which direction should be used, all the photographs returned are facing the same way, perhaps because this is the more visually appealing part of the street.

Within the parameters of subjectivity that were mentioned earlier, 57 of the total 164 requests (35%) made in the Realise trials called on a greater level of subjective behavior from the EyeSpy players. Before the introduction of the subjective reward system, this amount was 37%. After the subjective reward system was introduced, the amount was 34%. This would indicate that the subjective reward system had little impact on the number of requests requiring greater levels of subjective behavior from the EyeSpy players.

While all requests to EyeSpy players require some level of subjective opinion, this substantial amount requiring a greater level further indicates the importance of subjective opinion in determining the success of the Realise website.

A standard set of questions were used when interviewing the trial participants about the two versions of the Realise website (before and after the inclusion of the subjective reward system). This ensured that the same ground was covered in all interviews for the same system. However, follow up questions were asked when appropriate to learn more about each participant's answers.

The questions were split into several topic areas which included what users enjoyed and disliked about the website, what users found easy and hard to do within the website, how well users felt the website communicated with them and gave them feedback, how conscious

Figure 6.7: An example of the similarity that can exist between returned photographs for the same request. Both these photographs were taken at different times by different players in response to the same request.

Figure 6.8: An example of photographs with common elements, though not identical framing. Each of these photographs is a response to the same request. Many elements, such as the 'stuff your face' sign and black street posts, are seen in every photograph.

the users were of the EyeSpy players and how this affected their use of the system, how users felt about the requests system and subjective reward system, how users felt about the responses they received and the time taken to receive them, what led the users to make each request and whether or not the users found the system useful and could imagine using it beyond the trial.

Additionally, users were presented with a random selection of their requests and the returned photographs that EyeSpy players had come up with for them. The Realise users were then asked to talk about why they made each request and how happy they were with the resulting photographs.

A total of 32 Realise users were interviewed about their experiences with the system. 26 of these users (81%) reported that they enjoyed using the website, with 28 (88%) suggesting that they would like to use it again in the future. When asked whether or not they would recommend the website to other people, 25 users (78%) responded positively. Furthermore, 24 users (75%) said that they were happy with the responses they got for their requests.

There were 5 users who used the website before the introduction of the subjective reward system. Of these 5, all of them (100%) enjoyed the website and 4 of them (80%) said they would like to use Realise after the trial. All 5 (100%) said they would recommend the website to others and 4 of them (80%) said they were happy with the responses they received.

The remaining 27 users experienced the website after the addition of the subjective reward system. 21 of these users (78%) reported enjoying using the website, with 24 of them (89%) saying they would like to use it again. 20 of the users (74%) said they would recommend the website to others and 20 of the users (74%) said they were happy with their received responses.

This drop in user approval after the introduction of the subjective reward system indicates that the incorrect implementation of the concept actually made the system worse. This highlights the importance of preventing the flooding of systems as it leaves the users unsatisfied. This drop would make sense as the incorrect implementation actually allowed more requests to be made instead of less, leading to a worse flooding of the system and a lower likelihood of Realise users receiving responses to their requests.

However, despite these problems, the overall approval from the Realise users is still high with most people enjoying the website and being happy with the quality of the photos they received. This would indicate that the Realise website was successful in its goals, but took too long to satisfy the users. A corrected implementation of the subjective reward system might correct this problem.

6.3 – Conclusions

This chapter has shown that mutually reinforcing systems are difficult to evaluate because they each have a goal to make the other systems successful. However, the systems still have their own individual goals that can be used to evaluate the systems separately. If all the systems are able to achieve their individual goals which may require the use of other systems, this means that the overall mutually reinforcing systems loop has also been a success.

It was shown that existing evaluation methods used for games with by-products were not valid for EyeSpy due to the nature of the game and its differences from other games with by-products. A simpler approach is to examine the by-products to determine how useful they are and to interview the players to find out their opinions on the game. The by-products were evaluated showing that they were useful for navigation. The enjoyability of EyeSpy was then evaluated by examining interviews with EyeSpy players. It was shown that the game was considered enjoyable and that the inclusion of mutually reinforcing systems led to less monotony in the game-play. It was also shown that player motivations involved playing for enjoyment, but may also have had an element of altruism.

The Realise website was evaluated by interviewing the users to determine how happy they were with their experience of using the system. It was shown that all requests from Realise require some level of subjective choice from the EyeSpy players, but a substantial amount require a greater level of creativity and opinion. This showed that subjective opinion is of great importance in examining Realise because the users have chosen to use it to gather responses. There were high levels of users who enjoyed the website and were happy with the quality of the responses they received, but it took too long to receive the responses.

This chapter has attempted to show the higher level findings of mutually reinforcing systems, looking across all versions of the systems and the various trials in order to see changing trends and determine if the systems achieved their goals. Determining the success of Eye-Spy, in particular, was necessary in addressing ***RQ2*** and showing the success of the systems together was important in addressing ***RQ1***. A further discussion of the systems will be necessary before a definitive determination can be reached about how well the research questions have been answered.

Chapter 7
Discussion

Section 2.3.2.6 showed that there are common problems that exist in many games with by-products. These common problems include the generally broad by-products that are produced, as well as the games becoming monotonous over time. A third issue is that most games with by-products cannot respond when the by-products do not contain the desired result because it is not yet covered by the data set.

Chapter 3 introduced a mobile game with by-products that attempted to resolve the issue of monotony by having the players create the human computation tasks. Having the players create the tasks would also address another problem which was how to extend human computation to collect and classify useful contextual information. By giving the users the choice on what information to collect and turn into a task, the players were allowed to decide what information was useful and worth collecting. Having other players confirm the choice allowed that information to be classified as recognizable. Although the game proved successful in being enjoyable and producing useful by-products, the players reported that it still became monotonous over time. Also, having the players create the human computation tasks did not stop the game from producing broad by-products.

In the case of EyeSpy, the monotony of the game seemed to be the result of a lack of variety in the game-play. Even though the tasks were created by other players, they were intended to be easy to solve because this led to an increase in the creator's score. This reasoning meant that the same 'easy to confirm' targets were used a lot of the time and these were not intended to be particularly challenging. This also contributed to the by-products remaining fairly broad. A way to incorporate new human computation tasks with different motivations was required. Such a solution was presented in Chapter 4, which showed that systems could work together in order to help each other achieve their goals.

This solution incorporated tasks from a second system which had different motivations to those which originated within EyeSpy. This difference in motivation introduced more variety to the game-play to make the game less monotonous, but it still needed to be ensured that these new motivations would lead to different types of tasks being created so that the by-products also had more variety. This was achieved in EyeSpy by introducing the Realise website. Users of the Realise website were able to search through EyeSpy's existing by-products as well as make requests for new photographs. This design was intended to encourage the Realise users to primarily request photographs that the EyeSpy game was not already producing. Because a request would take more time to fulfill than searching for an existing image, it would be inefficient for users to request photographs which already existed. While there may be other reasons for Realise users to make requests, the motivations are still likely to differ from the players in the EyeSpy game. Even if some users do not follow the expected path of requesting images that do not currently exist, it would be likely that the desired images would still provide variety within the EyeSpy game and extend the variety of the by-products.

In addition to making EyeSpy more enjoyable and less monotonous to play, the new element in the game also provided a new ability: EyeSpy could now respond to the specific needs of another system. This meant that any eventualities that were not covered by EyeSpy's existing by-products could now be requested by the system that needed them.

While this shows that introducing outside motivations was useful in addressing three of the major problems in existing games with by-products, the solution had problems. While the EyeSpy game and the Realise website worked together to form mutually reinforcing systems, there was an issue of maintaining balance between the two systems. If Realise users made too many requests of EyeSpy, they could flood the game so that it could not keep up with further requests. Chapter 5 presented a solution to this problem that attempted to limit the Realise users' ability to make requests depending on the overall activity level of both systems. This system limited the number of requests per day across all users and also gave users a daily allowance of points that could be allocated to each request. This allowed the Realise users to prioritize their requests (amongst their own and the requests of other users). The points were used to assign worth to the requests once they became tasks within the EyeSpy game. It also allowed the game server to prioritize requests which were worth more points. A final effect of this system was that the Realise users were put in charge of whether or not points were awarded within EyeSpy.

As was discussed in Section 5.3, a flaw in the implementation of this subjective reward system prevented it from functioning as well as was hoped. While a solution to this flaw has been proposed, there was no time left to evaluate whether or not it would be successful. However, it does demonstrate that maintaining balance between mutually reinforcing systems is highly important to their success and ensuring a design that maintains this balance should be a priority. The subjective reward system is one approach to this problem and the general concept of minimizing requests between systems based on their overall activity seems sensible, even if the implementation used during the trials was flawed.

A final point that is greatly important to mutually reinforcing systems is that of trust. Each system is trusting the others to carry out their tasks and take reasonable measures to produce the required results. For example, the Realise website is trusting that the by-products from EyeSpy are of a good quality and that reasonable measures have been taken to ensure this. Both systems incorporate a system for indicating poor or abusive tasks and by-products and both trust the other system to act on these indications accordingly.

There is also an element of trust between the users. While the designs of EyeSpy and Realise rely, to a certain extent, on anonymity between users in order to achieve their goals, this anonymity is also important in maintaining trust between the users as well as maintaining the users' trust of the systems. People may not want to use the systems if they feel that their privacy can be compromised. For example, Realise users may make requests for things that are personal to them which they may not want to be divulged publicly. Of course, various common designs in human computation necessitate the sharing of user generated data in order to carry out tasks and reach consensus. This could lead to problems with privacy due to systems that might be able to find patterns and de-anonymize the data [128]. Such possibilities serve to highlight the importance of only sharing the minimum data that is required for the systems to function, especially when that information might be passed between systems.

As well as maintaining anonymity with information passed between users, there is also an issue of maintaining privacy in the by-products as these may be used in public systems. For example, the photographs taken in EyeSpy might include identifiable information about people who are not involved in the game. This issue has already been raised concerning the Google Street View system [108] which takes pictures of cities from a car in order to allow users to explore the city streets from a browser. However, this blanket coverage might include identifying information about people. Although the system tries to blur out people's faces and the registration plates on cars, some have occasionally gotten through.[1] Google Street View uses a mixture of positioning technology, including Wi-Fi to determine the location of the car when the photographs are taken. However, Google mistakenly downloaded data packets from many wireless routers in using this positioning technology which may have included personal information.[2] This shows that pervasive systems like Google Street View have a strong need to maintain privacy in such settings. While EyeSpy takes no extreme measures (such as blurring people's faces), it is a potential concern for the mobile environment in which the game is played. This is perhaps of a greater concern in the case of mobile games with by-products because there is a necessity to pass information between players and the potential to create by-products that might reveal personal information about people, even outwith the game.

Throughout this dissertation, a solution to various problems in games with by-products has been presented in the form of mutually reinforcing systems. However, a clear definition of what mutually reinforcing systems are and how others might be created has not been given. This is because mutually reinforcing systems are a collection of traits and abilities, making a simple explanation difficult. This dissertation has shown that mutually reinforcing systems work together, benefiting from each others' strengths in order to achieve their goals. Working together takes the form of sending requests and responses between the systems. These are the primary requirements of mutually reinforcing systems because it is these features that allow the systems to work together in improving each others' goals. The focus for mutually reinforcing systems has been toward the improvement of games with by-products. This allows them to deal with specific needs and encourages finer grained results. It also has the benefit of increasing the variability of the game-play and allowing systems to take advantage of each others' strengths.

[1] http://goo.gl/ML3q5

[2] http://goo.gl/Uob2r

The potential benefits that the concept of mutually reinforcing system provides games with by-products make them an attractive solution to some of the problems this dissertation has raised. However, in order to allow other games with by-products to take advantage of this concept, a discussion of how to design these systems is needed. This rest of this chapter will focus on extending the common designs of games with by-products so that they can incorporate mutually reinforcing systems and show how other systems can be built to take advantage of this. A discussion of remaining issues and potential future research will also be presented.

7.1 - System Design

Throughout this dissertation, a focus has been given to the common design elements of games with by-products. These were identified in Sections 2.3.2.1 and 2.3.2.2 by examining several existing games with by-products. The concepts were then used in the design of a mobile game with by-products, as described in Sections 3.1 and 3.1.1. Identifying and applying these common design elements proved successful and this technique was continued when a second system was added. This required changes in the game, as discussed in Sections 4.1 and 4.1.1. Lastly, ensuring these common design elements were being followed was still a priority when the refinement of maintaining balance between the two systems was added in Sections 5.1 and 5.1.1.

It has, therefore, already been shown that the concept of mutually reinforcing systems incorporates these existing design techniques. However, the discussions mentioned above concentrated on specific needs of the systems at the time, rather than discussing how the common design elements applied to mutually reinforcing systems in general. Supplying this higher level discussion could be beneficial in seeing the larger issues in designing mutually reinforcing systems and may provide a guide for those who wish to design their own. To do this, each common design will be discussed from this higher perspective.

Payment Through Entertainment

An essential goal to games with by-products is keeping the players happy. If they enjoy their experience in the game, they will continue to play and by-products will continue to be produced. Mutually reinforcing systems help with this by introducing game-play elements that are different from those produced from within the game itself. Adding this sort of variety to the game is important in reducing the monotony that can come by having the same sorts of tasks continually carried out by the game players. But mutually reinforcing systems can be used to inject new tasks into the game that differ from the internal ones. It is also possible that the people making the tasks will have different motives from each other, producing yet more variety in these tasks. This is in contrast to the internal tasks which are either based on an existing data-set of similar items or which are created by players who all have essentially the same motivations and goals.

The method of using requests to create tasks also provides the possibility of more subjective and creative tasks than would normally be seen in games with by-products. This is because only one person must initially be satisfied by the results rather than being an objectively validated result. Of course, such results will need to be validated objectively if they are going to be used in a more general way, but to satisfy the requester this is not necessary.

It should be noted, however, that while our designs promoted tasks with higher bounties so that they would be dealt with first by the game players, this does not necessarily mean that these tasks were the most entertaining. However, it is still likely to provide variety in the game and should still be entertaining, even if it is not as entertaining as using other requests which have lower bounties.

As was seen in our own designs, it is important not to let confirmation schemes get in the way of user satisfaction. The initial version of priority targets included a complicated and time consuming authentication scheme which annoyed players and website users. Recognizing that it was only the opinion of the website users that initially mattered, we were able to create a change in design that made the system more satisfying and enjoyable for both website users and players. When designing mutually reinforcing systems, it is important to recognize the difference between subjective and objective importance in by-products and not to use the wrong one at the wrong time. For example, using a subjective measure for the recognizability of the by-products in EyeSpy would probably have produced by-products that could not have been used so broadly. But, imposing an objective measure on the return of responses for website requests just kept everyone waiting when this was unnecessary to the needs of the users. Entertainment and satisfaction are both important in mutually reinforcing systems to maintain sustainability and to continue allowing them to work together to achieve their goals. The needs of both systems and their users must always be considered when determining solutions for the authentication of by-products.

Game Types

While the game types presented in Section 2.3.2.1 all have their merits and applications, it is perhaps not especially important which one is used in mutually reinforcing systems. Depending on the needs of the systems, it must be evaluated which game type would be most suitable. However, due to the request/response mechanism of mutually reinforcing systems, there is an argument to be made for using inversion-problem games. The requester would act as the describer and the game players responding to the requests would act as the guessers. Of course, this is slightly different than the conventional design where a computer determines if a guess is correct. With this similarity in design, it might be easier to fit mutually reinforcing systems into games of this type.

It should also be noted that the original implementation of mutually reinforcing systems had a complicated validation system for responses which fit into the design of an output-agreement games, as discussed in Section 4.1. While this solution proved to be too time consuming and was ultimately unnecessary, it meant that EyeSpy was operating under two different game modes. This increased the variety of the game play and this was lost when the new solution was implemented. While the game-play did not change significantly under these two different implementations, the reasoning that players had for choosing locations may have changed because of whether or not they were taking into account what locations other players would choose. While the goals of the system must be taken into account when choosing which game type will fit best, it is worth keeping in mind that having different game modes for the main game and the part that deals with requests might add more variety, which players may find more enjoyable over time.

A last point to note is that the mutually reinforcing systems which were implemented in this research imply that the game must be able to produce its own content, rather that deal with an existing data-set. While this is true of the systems presented here and that the requests were used to encourage the creation of specific content, this is not a prerequisite for mutually reinforcing systems and requests need not always focus on content creation. For example, a requester might simply want a better label to be produced for an existing image (perhaps because it is of a monument they do not recognize).

Human Validation

As discussed in Section 2.3.2.1, human validation systems should be constructed so that everyone gains by being honest and by being correct. This prevents cheating and ensures that the by-products are of a good quality.

Two approaches to human validation were tried in our implementations of mutually reinforcing systems. The first approach was complicated and time consuming, but required users to agree on a location and that the photograph created there was recognizable. This is a fairly objective approach as it requires players to agree on the by-product before it is returned to the website users.

The other approach simply lets the requester decide whether or not the by-product matches their request. Even in this second approach, the by-products did not enter the general data-set of by-products until they had been confirmed in the game as being recognizable, but that did not mean that the requester could not have access to them. This second approach was more subjective because only one game player was involved and only the end user could decide whether or not the by-product was valid for their request.

This second solution seems optimal because the requests are inherently subjective anyway. Putting the photos through the same human validation as the rest of the game ensures they are appropriate additions to the general set, but that should not prevent them from being sent to the requester directly. The only potential problem is ensuring that the website users tell the truth. If the website users could accept all the images, there would be no real reason for them to reject an image. The solution presented in this dissertation was the subjective reward system. This concept meant that users would only see a limited version of a response until they committed to it being correct. This commitment came in the form of paying bounties that the user set up. This meant that game players who created the best response would be rewarded accordingly. However, this would just be based on the subjective opinion of the website user. Also, this does not mean that other responses were not valid, just that they were not the best. But is it right that players giving lesser, but accurate, responses receive no reward?

It is important that a user should not feel that a non-validated by-product has been a waste of their time, even if no one else confirms it. Some system should be put in place to avoid this scenario or provide some consolation if it occurs.

The first implementation of mutually reinforcing systems incorporated this by having request responses fed back into the game as regular photo targets, providing another way to get points even if the location was not confirmed by other players. However, due to the subjective nature of some of the requests, it was sometimes unlikely that the same location would be picked. For example, if someone requested a photograph of 'an interesting tree' then the chances of players both picking the same location are less likely because the request was not specific enough. However, this does not mean that there will not be several accurate responses, but these would never reach the website user. It is also not the fault of the website user because much of the benefit of mutually reinforcing systems comes from the ability to use human computation across systems and taking advantage of human choice is part of that.

The second implementation of mutually reinforcing systems also put the photographs back into the game as regular photo targets, but this time it was providing points for when the requester did not accept the photograph. This is perhaps more important when the acceptance is highly subjective (as in the second implementation).

Another means to reward players if their photo was not accepted was to allow Realise users to buy other's non-accepted photos. This way, if a new user made a similar request to that of someone else, but the original requester had not yet paid the bounty or had not chosen the photograph, then the new user could buy it instead to get immediate access rather than having to request a new one which would take longer. This means that if one user did not subjectively feel an EyeSpy user's response was the best, another user might later on. This would allow another means for the player to gain points.

In summary, human validation is a necessary component of human computation and it is important that all users involved are rewarded for it. Mutually reinforcing systems, and the subjective reward system in particular, introduce the possibility of not being rewarded for correct responses. Because of this, it is necessary to provide other means for people to receive rewards as a consolation for not having produced the by-product which the requester likes the most. The two approaches suggested here (allowing responses to re-enter the regular game to make points and allowing other users to pay the bounty) could be generally applicable to all mutually reinforcing systems.

Using Recorded User Actions

The principal benefits of recorded user actions are allowing an odd number of players where direct pairing is necessary and to be used as a combative measure against mass collusion within the game. This has also been used as a validation technique in some inversion-problem games. The first benefit was not an issue in EyeSpy because direct pairing between players did not take place. However, recorded user actions could have been beneficial to prevent mass collusion, even though this was not seen during trials of the game.

In terms of mutually reinforcing systems, recorded user actions could be used to re-inject requests into the game if there were not enough new ones being made. The requests provide variety in the game-play of EyeSpy, so it would be beneficial to ensure that requests continue to appear in the game, even if there are none which currently need satisfied. However, this would require an automated means of validating responses, as seen in the original implementation of Realise.

Preventing collusion with recorded user actions could also be important for mutually reinforcing systems if there was a large number of abusive or deceptive requests. Older, previously accepted requests could be put into the game instead until it could be established which users were not part of the scheme, gradually allowing them back into the game.

Keeping Play Competitive

Keeping play competitive ensures that players will continue to play over a longer period of time. Competition is not just about encouraging bad players, but keeping the good players from feeling they are getting too far ahead.

In the systems created for this research, introducing the concept of mutually reinforcing systems supplied a means to keep the games competitive by supplying a variable scoring system that made it easier for players to catch up or overtake each other. The tasks which the points were associated with were also variable and might not always be available (depending on the number of requests being made). This makes them act more like a 'bonus' element in the game. Of course, this could have been achieved without mutually reinforcing systems by randomly promoting certain targets to be bonus elements with a random number of points associated with it. This may not have had quite the same effect though as the website users were prioritizing their requests by assigning appropriate points values. This may have meant that requests which were more difficult to fulfill could have been worth more. It may have been more difficult to ensure this was true if the points were assigned randomly, but one method could have been to promote tags which had been confirmed but only infrequently. Again, this might not have been as effective because the small number of confirmations might be due to local knowledge that was only known by a small number of players. In the case of mutually reinforcing systems, the requests might just have required more effort but still have been known in such a way that multiple players could confirm them.

Another means to keep the game competitive could have been to present more statistical resources to the players, showing them where they were earning most of their points and how this compared to other players. This could have allowed the players to be more tactical with their play in order to maximize the number of points for the effort put in. This is perhaps an important issue with the inclusion of mutually reinforcing systems because it takes time for the Realise users to grant points to players. The lack of immediate response makes it difficult for players to see exactly when and where their points come from. This lack of immediate gratification might make the priority targets in EyeSpy less appealing, but a break down of the amount of points received from priority targets and the percentage of priority targets that make points might help people to see their benefit.

While game-play must be kept competitive, it is worth noting that there is an element of competition for the website users with the inclusion of the subjective reward system. This is because the users have to set the bounties for their requests competitively amongst themselves in order to make sure that their requests will not be pushed to the bottom of the list. The current implementation makes this difficult because users have no way to know what bounties the other users are setting. In future implementations of subjective reward systems, such a means to browse bounties should be included. This could also prevent users making the same requests as each other, especially if they were allowed to purchase the responses from other users' requests. Of course, this should all be done anonymously without identifying which users make the requests and bounties.

The sporadic style of game-play used in EyeSpy also necessitated that a limit be put on the daily play. This was important in preventing players from racing too far ahead of others because they had more free time. One of the goals in the game was to allow it to fit into the players' daily lives and not everyone would have as much free time to play. This becomes more important with the introduction of mutually reinforcing systems because the number of requests which a user can respond to must be limited each day to ensure that players cannot race ahead with the larger points values of priority targets. While limiting game-play might seem counter-productive (preventing the creation of more by-products), maintaining a broad user base of players who want to continue playing over an extended period of time is more important in the long run. If only the top players are left, they will eventually get bored and stop playing due to the lack of competition. Also, a smaller number of players will lead to less diversity in the by-products.

Another possibility to keep games competitive is to introduce a leveling system so that players are only playing against people who match their own abilities. Such a system would not be especially hard to implement and could have a good effect on maintaining competitive behavior amongst the players. This was never used during this research due to time constraints and the relatively small scale of the trials.

The methods to maintain competitive behavior need not all be large scale changes to the game rules. For example, later versions of EyeSpy included an indicator that showed players when they were in the vicinity of a target they needed to confirm. This was a minor change that allowed people less familiar with the area to have a better chance of finding targets.

It should be noted that rules to keep the game competitive can sometimes go too far. A balance must be maintained to ensure that top players do not feel their position can be stolen too easily or they may feel it is too hard to maintain their place in the score board.

7.1.1 – Designing For Mutual Reinforcement

The common designs of games with by-products have now been discussed at a higher level, showing how they are affected by the concept of mutually reinforcing systems. However, designing mutually reinforcing systems requires some new common designs. From the experience of designing the first mutually reinforcing systems, some important design issues have been identified that should be important in designing future systems. These will now be discussed.

Clearly Defined Goals

A point that must be noted about mutually reinforcing systems is that each one must have its own, unique goals. If this is not the case, then there is no possibility that the systems can work together to help each other achieve their goals. In essence, each system must have at least two goals where one of them is to improve the functionality of the other systems and one of them is something else. Systems cannot reinforce each other if one of them has nothing to reinforce. This does not mean that helper systems cannot contribute toward mutually reinforcing systems in some way, but the principal systems must have other goals in order to be mutually reinforcing.

Making and Responding to Requests

The role of games with by-products often involves the creation of a data-set for another system. The approach taken by this dissertation is to allow this other system and the game to work together so that they can benefit from each other and work together to achieve their independent goals.

In order to allow systems to reinforce each other's output, they require a means to work together. While there are many ways this might be achieved, the approach taking in this dissertation was to allow systems to make and respond to requests from each other. This approach allows systems to clearly state what they require from each other to help them achieve their goals and also allows a relatively loose coupling, allowing systems to be replaced later on. This is important in mutually reinforcing systems because games may become less popular over time as their novelty wears off and might need to be replaced with new ones.

In addition to requests and responses, systems can also work together by sending reinforcing data, even though it is not specifically requested. This might be useful information that is quite broad (and therefore unlikely to require specific requests), but which the other system is incapable of gathering on its own.

In Chapter 4, it was shown that mutually reinforcing systems acted in loops where each loop contained a minimum of two contributing systems and where each system could be part of many loops. The loops were formed by the messages sent between systems: requests, responses and reinforcers. While it might be possible to have multiple systems in a loop, minimizing the number as much as possible reduces the complexity and maintenance of mutually reinforcing systems, making them easier to work with. Complex interactions might still be modeled by increasing the number of loops of which a system is a member rather than increasing the number of systems in a loop.

In the mutually reinforcing systems presented in this dissertation, the loops generally lead to an increase of data because something additional was being requested. But this may not always be the case. For example, if the Realise website was able to detect that an image was abusive and contained no useful information, this finding could be sent to EyeSpy as a reinforcer. This would allow EyeSpy to remove the offending image and, therefore, strengthen the overall quality of the data-set.

Another possibility is that the by-products are re-interpreted rather than increased or decreased. For example, if a particular location is returned for a certain search term in Realise, but the users feel this is in error, they could flag the result as being incorrect. This information could be sent to EyeSpy as a reinforcer, not so that any data is changed or removed, but simply so that EyeSpy can mark that the photograph does not represent a particular textual description. In essence, the photograph is marked to show that the associated labeling for it is bad. While it would be possible to simply decouple the two by-products, it may be useful in the future to know that the coupling is bad, rather than to remove it and forget about it. This could, for instance, prevent such a coupling of a similar textual description for that photograph in the future. In this sense, even bad by-products can still have value if they are known to be bad.

There is also a fine line between keeping the requests formal and allowing for versatility. For example, the Realise website could have been set up to use a form so that requests used a specific pattern without the freedom to request anything. For example, only specific place names could be requested that must already exist in a database. The benefit of this is that it prevents abuse of the requests system, but it also limits much of the potential for mutually reinforcing systems. In certain areas, such as mobile systems where the environment cannot easily be controlled or predicted, the human players' ability to adapt can be a major benefit. If we give the requesters more freedom to ask for exactly what they want, the human players can adapt to provide as good a solution as they can. An obvious example of this can be seen in requests that require subjective choice from the players such as 'the best pub in Glasgow' or 'people sitting in a lecture.'

Allowing this versatility and giving the human players more freedom to respond allows for a fairly deep level of adaptation, albeit at the cost of possible system abuse. However, this deep adaptation is an important benefit and should not be underestimated. Deep adaptation differs from shallow adaptation in the way that the system changes.

In shallow adaptation, the system itself does not change in a fundamental way, but instead has pre-programmed mechanisms to deal with specific values of data. By appearing to react to this data, the system will display a sense of adapting to its use. This could be in the form of a system informing the user that he or she should dress warmly because it has detected that it will be a cold day. This is a superficial result and the system itself has not changed in any major way.

A deeper adaptation would involve the system changing in a more fundamental way by introducing entirely new functionality or changing core parts of the system. A good example of this is in the 'Nomic' game [122] where the goal is to add to or amend the existing rules of the game in such a way that you win before the other players. The core and every aspect of the game is open to change and this presents an example of a system capable of deep adaptation.

While there may be a sliding scale between these two extremes of adaptation, the difference between them is important. Providing access to contextual data allows for simple programming techniques to make software appear adaptive, but the systems themselves do not adapt beyond a superficial level. However, making humans a core component of the systems allows for a deeper level of adaptation and this is a major benefit of mutually reinforcing systems. Therefore, it is important to find a good balance between using deeper adaptation and opening the system up to potential abuse.

Equilibrium Between Systems

Maintaining equilibrium between systems is important in preventing the flooding of requests. If one system demands too much of the other, it will prevent them from working together in a productive way. This was seen in Realise when too many requests were made. EyeSpy could not keep up with the demand and the Realise users were left to wait too long for responses to their requests. This lack of equilibrium was also seen due to the incorrect implementation of the subjective reward system. Realise users were allowed to make more requests than they were able to pay for, leaving the EyeSpy players with no points for their efforts. Seeing no response to their efforts might make the players give up on the priority targets altogether, thinking that they never receive points for them.

The solution presented in this dissertation to maintain equilibrium was the subjective reward system. However, the success of this solution remains unproven due to failures in the implementation with no time left to test a corrected version. Part of the problem stemmed from not getting users to commit to their bounties when they made requests and it seems logical that fixing this would solve the issue. However, without further testing, it is impossible to say whether or not this would work. Nevertheless, the concept of measuring system activity and preventing or promoting use based on that activity is still a valid way to maintain equilibrium between systems, even if the exact implementation used in this dissertation was not proven. This basic concept could apply to any future mutually reinforcing systems and would allow the systems to scale well. This would also allow the system to detect when recorded user actions should be used if not enough requests are being made.

While receiving an accumulating allowance to spend on request bounties is an important part of the subjective reward system, the non-accumulating limit on requests should not be underestimated. While an accumulative points system encourages people to continue using the system and making more requests (which is beneficial to EyeSpy), limiting the number of requests through a non-accumulating allowance based on system activity is essential to prevent flooding. Again, the implementation used in this dissertation failed in this respect by not being aggressive enough and taking too long to react.

Another issue is what would happen if there were more than two systems in a mutually reinforcing systems loop. In this scenario, all the systems would have to report their activity to each other so that a correct allowance could be given to their users. This may require trust between the systems, which might be an issue depending on the circumstances.

Another possible solution would be to use real money. While the creation of a fake economy system was discounted during the design stage of the subjective reward system because all the 'money' would end up with the players rather than being passed around, using an existing economy would solve this problem because the money would come from an existing economy and the players would be able to spend what they earned within that economy.

This might also go some way toward addressing the trust issue. While abuse might still occur, it would cost people real money to carry out such behavior. The downside is that this would not really maintain equilibrium between systems, just between the richest people. If one system contained predominantly poorer users, it would likely end up under-represented in the loop because the richer system could outbid it for the game players' time. In this sense, using real money would be less effective in maintaining equilibrium between all systems if there were more than two in the loop.

Opportunistic Design

While the design of mutually reinforcing systems should be directed at allowing the systems to achieve their independent goals, there may be other opportunities in which the systems might help each other.

For example, if several Realise users search for something that returns no images but never make a request for it, perhaps an automatic request could be sent to EyeSpy. Another example might be if EyeSpy players frequently delete a request but do not flag it as inappropriate, then perhaps the Realise user should be asked to clarify the request.

While these types of scenario might not directly contribute to the larger goals of the system, they can be used to improve general behavior and performance, perhaps making the systems more efficient or enjoyable to use. Although this is not a requirement for mutually reinforcing systems, paying attention to such possibilities will identify further benefits that may not be immediately apparent.

7.1.2 – Designing Against Cheating

As with the rest of the dissertation, cheating is an important issue in games with by-products and this continues to be true with those that are part of mutually reinforcing systems. Common rules to address this problem have been described throughout the dissertation and these shall be discussed now from the higher level perspective of how they apply to mutually reinforcing systems. The previous sections on system design in this chapter focused on maintaining and promoting the creation of 'good' by-products in mutually reinforcing systems. This section will focus on the prevention of 'bad' by-products, as discussed in Section 2.3.2.2.

A first, technical, point should be noted. Because a second system is being added to games with by-products, there are now requests and responses going between the two systems. This introduces more ways for people to intercept and change messages. However, most games with by-products are multi-player or in some way involve a server which the players must communicate with through the game. This means that there are already ways in which people might try to intercept messages. While there are now more opportunities, the larger issue has not really changed with the introduction of mutually reinforcing systems.

A problem may be introduced if systems were allowed to join and leave mutually reinforcing systems loops through some sort of framework. Since it has already been established that mutually reinforcing systems may require a level of trust between each other, it might not be practical to create such a framework. However, this may be possible if there was some way for systems to monitor each other's behaviors and ensure they were secure. Even if no sensitive information was passed between systems, it would still be important that information was correct and that equilibrium was maintained in order to ensure the viability of by-products. Equilibrium might be attained between systems by only allowing requests to be made once a certain number of confirmed responses had been given. This is in contrast to the earlier discussion of equilibrium which focused on the balance of the user activity between two systems. The issue in this case would be making sure that each system as a whole was not flooding the loop. In all likelihood, some sort of administrator would need to be put in place so that systems could register for membership. If a system did not play by the rules, it could then be banned from taking part.

However, these are technical issues and the focus of this dissertation has been on the conceptual and high level details of the mutually reinforcing systems concept. As such, the rest of this section will try to address these higher level issues.

Agreement Rules

Agreement rules ensure that a by-product is validated by having multiple users confirm a by-product in some way. The final versions of Realise and EyeSpy took this concept and had it work across systems so that the agreement was between a player and a non-player. The player and non-player have different motivations but both want satisfaction. The player wants to create a photograph that will be accepted so that they can gain points. The non-player only wants to pay for acceptable images. These two motivations should lead to an appropriate agreement rule, even if it is only enough to determine that the outcome is valid for these two users. Before the by-product can be added to the data-set for general use, it is still subject to the agreement rules for non-request tasks. This way a requested photograph can still be used by other users as a by-product of EyeSpy.

Another reason for this extra level of validation is that the Realise and EyeSpy users might be colluding with each other, making their agreement invalid. It must be ensured that this will only affect those two users and not infect the general data-set. Using a single account across systems might prevent this (perhaps rewarding game players with a greater allowance on the requesting system), although people could just register twice and make two separate accounts.

The potential for collusion across systems is an important issue. If there is enough freedom in the requests system, this could allow people to pass notes to the game players. This could be addressed by detecting how often a particular user's requests are deleted by game players. If most players delete them, but the same one or two always manage to confirm them, it could indicate a collusion. If this was on a larger scale (perhaps involving a naming scheme), the average time taking to confirm requests or the average number of successful confirms could be detected and then recorded user actions could be injected to prevent the collusion from continuing.

Another potential problem of agreement rules comes from the possible background of the users across systems. In the case of EyeSpy and Realise, the game was expected to be used by locals and the website was expected to be used by tourists. This means that one set of users would be more familiar with the area than the other. This could lead to some cultural problems concerning the descriptions of the area or for EyeSpy players choosing what is most appropriate for the Realise users. This is unlikely to be the case in all mutually reinforcing systems, but it is worth keeping in mind that the different backgrounds of the users of different systems might need to be taken into account during the design stage.

A potential means to detect people who were trying to collude would be to require players to go through a trial period of only playing the regular game until they reached a certain number of points. This might make it less attractive to try and collude. At the very least, it would ensure that something productive came from players before they tried to collude.

The final versions of the systems removed the agreement rule between players of EyeSpy that was needed to validate the location of request responses. This means that players were once again able to use the priority targets to extend the number of regular photo targets they could create each day. However, these will be returned to the Realise users, cluttering up the response page for their request. One way to deal with this could be the introduction of a flagging system so that Realise users could report photographs that are not relevant to their request. If the same players were found to be repeatedly doing this, they could then be reprimanded.

Consensus Rules

While the agreement rules for the request system in Realise and EyeSpy attempt to ensure that the requester and the player will be happy with the outcome, this does not mean that the by-product is ready for general use. Similarly, if a different website user was allowed to pay the bounty and buy the photograph for themselves, this still does not mean that a consensus on that photo has been reached. This is because both these scenarios involve a subjective measure of the acceptability of the photograph. Before the by-product can be accepted for general use, an objective measure is required. This is achieved by having a threshold number of users agree on the acceptability. In the implementation of EyeSpy and Realise, this was achieved by putting the photograph through the same consensus procedure as all the other photographs in the game.

The consensus rule can also help identify colluding players. If there are groups of people who always confirm their members' by-products when no one outside the group does, they are probably colluding. This might also be achieved through a flagging system where the non-colluding players could report by-products that they suspect are made by colluding players. Of course, this flagging will also require a consensus before any action can be taken to make sure that people are not merely abusing the flagging system.

Anonymity Rules

Preventing collusion between systems can be helped by maintaining anonymity between users. This can be achieved by randomly assigning the requests to players. This might even be put on a rota system so that each non-game user must be paired with each game player in the game area at least once before repeating.

There is an issue of privacy here as well. For example, if a Realise user makes a request that is personal or private and EyeSpy players are able to identify the requester, this may prove embarrassing for the Realise user. While it is the responsibility of the requester not to reveal identifying information about themselves in a request, it is the responsibility of the designers to ensure that no unnecessary information is passed between the systems that might identify the requesters or players to each other. If identifying or sensitive information must be passed, this should be done in a secure way so that the data cannot be read except where needed.

While anonymity is important, there may be benefits in opening up the band of communication between players and requesters. This could allow players to ask for clarification of descriptions or otherwise encourage a dialog to ensure that the by-product correctly matches the request. However, this should still be achieved in an anonymous way as far as possible. For example, not allowing actual text to be written, but providing a form of pre-written responses that can be selected. The central issue is that anonymity prevents collusion and maintains privacy amongst all users. In designing mutually reinforcing systems, we must be vigilant to ensure that only the absolutely required information is passed between systems and is kept as secure as possible at all stages.

Diversity Rules

Mutually reinforcing systems introduce new game-play elements which add to the diversity of the games. People are requesting things with a different motivation than the game players, and this is likely to increase the diversity of the by-products. It may be possible to increase this benefit by presenting the existing by-products to the requesters in some way. In Realise, the nature of the website allowed the users to search and browse all the existing confirmed by-products. This should encourage players not to request new by-products which already exist in the data-set (because requesting a new by-product will take longer). Presenting the data-set in this way should encourage unique requests that will lead to a more diverse data-set in the future.

Another way to further encourage unique requests could be the prevention of people making similar requests. While people may do this accidentally, or simply not find the existing responses, it is also possibly that requesters could purposely try and use similar requests in order to disrupt play. If enough people got together to do this, they could flood the system with a single request that players would have to confirm repeatedly. By comparing request text to that of existing requests, this might be prevented. If the match is not certain, the existing request could be presented to the user, showing them the resulting by-products. If the system allowed other users to buy by-products from other users' requests, this could also be presented as an option, being a quicker solution that creating a new request.

An interesting point to note is that the original implementation of mutually reinforcing systems had an internal validation scheme within EyeSpy for by-products that were the result of requests. The results were not passed to the Realise users until they had been validated. However, this validation allowed players to use the priority targets as a means to extend the number of regular photo targets they could create each day (because the images from priority targets were put back into the game as regular photo targets). With that validation scheme, there was no way to catch players who did this. While their photos would never be returned to the website users, they would never be caught for this behavior because it was hard to tell that this was going on. It would be entirely possible that they just kept taking photographs of the request from a different location than everybody else, thus preventing their photographs from being validated in the same location. However, the newer implementation, by removing this validation, actually makes it more likely that these offenders will be caught. It would be simple to include a flagging system within the Realise website that allowed requesters to say that a response has nothing to do with their request. A player who gets flagged for this by several Realise users could then be reprimanded.

Competitive Rules

The addition of the subjective reward system to mutually reinforcing systems, as well as the variability added by requests, helps keep the game more competitive by allowing a variable points model and different game modes. These make it easier for people to overtake each other.

However, a potential problem of this system is that many people may all respond to a request, but the person who made the request may only pay the bounty on the best one. This might happen despite many of the responses being of a high quality and fully satisfying the request. While there may be several valid by-products, not all players in this scenario will be rewarded fully for their efforts. In the EyeSpy game, players should still receive points when their image re-enters the game as a regular photo target, but this will still mean they are losing out on a large number of points. This also opens up the possibility that requesters make requests without any intention of paying the bounties. Assuming that a correct implementation of the subjective reward system was in place (thus requiring requesters to commit their allowance to a bounty as it was being made), users would be limited in the number of requests that could be made. However, the system will continue to give the requests to players until a successful response has been 'bought' by the Realise user. By never confirming, Realise users could create requests that will always be sent out, without ever being satisfied, thus bloating the system with unnecessary tasks.

There are a few ways that such eventualities might be addressed. The most straightforward is to put a lifetime on requests. If a request is not satisfied within a set period of time, it will no longer be sent to the EyeSpy players. It should be noted that the points which the requester has committed to the request should not be returned to them. They can always remain there so that the user can change his or her mind and purchase one of the pictures, but freeing the funds would allow them to potentially save up funds over a long period of time to make it easier to flood the system with similar tactics later on.

Another possibility would be to limit the number of users who receive each request. The implementation used in this dissertation made this a largely random assignment where sometimes more players would receive certain requests. However, it could be possible to ensure that a request would only be sent to some maximum number of players so that it would not forever be in the system. This may limit the potential to get a large variety of responses, but it would prevent abuse and would also make it more likely that players would receive points for their efforts because they would be competing against fewer players for each request.

The design of mutually reinforcing systems has now been discussed, but this has only concentrated on issues that have arisen from the current implementation seen in this dissertation. The next section will discuss potential issues that may appear in future designs of mutually reinforcing systems and some possible improvements that might be made to the concept.

7.2 - Remaining Issues and Future Work

While this dissertation has attempted to present and address as many of the issues relating to mutually reinforcing systems as possible, time limitations prevented every eventuality and idea from being explored.

One issue with mutually reinforcing systems that is not fully addressed is how well they will scale outside of the two systems which are presented in this dissertation. While the subjective reward system is a starting point for maintaining balance between systems, it is only designed to deal with scaling in the number of people using each system. But how might the scaling work if more systems are added to a mutually reinforcing systems loop? The trust based nature of mutually reinforcing systems might suggest that all the systems in a loop need to be designed and maintained by the same people to ensure that all the systems act appropriately and fairly. Another solution could be to incorporate an administrative element to the loops, allowing third-party systems to register for inclusion. If they broke any rules, they could then be removed from the registry.

However, both these ideas are somewhat limiting. The first solution prevents third-party systems from taking advantage of each other and potentially injecting new functionality into a loop. The second solution involves the use of a dedicated administrative staff that monitor the behavior of systems and ensure that they are all acting appropriately. A better solution would be to allow third-party systems in an automated way that still ensured that a fair level of requests and responses were produced by all the systems in a loop. A possible solution could be to use a real economy for the requests system. This would mean that any abuse of the systems would cost the offenders real money, making it less attractive to ignore the rules. However, this could lead to a division amongst the users where people with more money ended up dominating the systems and those with less money would find it too costly. If most of the users of one system were more wealthy than those of another, this may also lead to the poorer system being unable to outbid the richer users and effectively lead it to stop operating.

One solution to maintain trust between systems could be the use of a reputation system, as described by Burnham and Sami [129]. By testing systems to make sure that they produce good by-products or do not abuse the rules of the subjective reward system, it can be known that a system will play by the rules before it is fully introduced into a loop. However, this would not prevent the designers of the new system from changing how it operated once it fully joined the loop.

A potential solution might be to separate the existing subjective reward system out of the individual systems and make it a dedicated system. This system would act as an account manager, keeping track of players' points and non-players' allowances to spend on requests. All transactions involving points exchange would have to go through this separate system, including requests and bounty payments. This, in essence, takes the trust of the requests system away from individual systems and puts it all on the separate subjective rewards system. As long as all the systems trusted this one system, they would not have to trust each other. This could solve the issue of trust for third-party systems, allow for an automated solution and ensure that balance was maintained across all systems in the loop.

This is not so different from how things are already implemented: the subjective reward system is simply moved out of the individual systems and into its own dedicated system. The system for keeping balance would have to be modified slightly to address the issue of some systems being more or less active than others. This was not an issue with the existing implementation described in this dissertation because only two systems were being used. However, if there were several systems and one of them was less active than the others, it could introduce a bottleneck to the loop. Rather than measuring overall activity in the entire loop, activity for individual systems would need to be measured so that users could be presented with how many requests and how many points could be spent on each system each day, rather than there being general amounts across all systems.

Scaling mutually reinforcing systems is not only relevant to maintaining the balance amongst systems and players: the problems being solved by the systems could also become increasingly complex. It has already been shown that games can be used to address relatively complex topics such as tracking the spread of a virus, as discussed in Section 2.3.2.3. The games used to carry out this tracking were fairly complex compared to most current games with by-products which primarily deal with simple tasks that can be solved relatively quickly. But if we wanted to use human computation and games with by-products to address more complex problems, how would this be achieved? One possible solution would be to break the problem down into small component elements. Systems could then be built to address each of these smaller problems, then a larger solution could be assembled. Mutually reinforcing systems might provide a good framework for such an approach as it would allow systems to communicate through requests and ensure that systems worked together in a balanced way, communicating with each about their needs and results to solve the larger problem. This could be an interesting area for future research.

In addition to general issues about mutually reinforcing systems, there are also some improvements that might be made to the specific systems presented in this dissertation. For example, the amount of contextual information used by Realise could be extended by having EyeSpy collect more information when targets are created. This could improve the usefulness of the Realise search engine, as discussed by Lane et al. [130].

Another improvement to the systems could be taken from 'TinEye,' which is a reverse image search engine that finds the web pages than image appears on.[3] There is a potential to do something similar with Realise and EyeSpy. Realise users could submit a photographic image which could then be sent to the EyeSpy players. Once the images were received, players would be given the task of 'confirming' the photograph by going to the location where it was taken. This would be very similar to the existing game dynamic for regular photo targets. The difference would be in requiring multiple players to independently confirm the location. However, once this has been done, the photograph would now be geographically referenced. This method of labeling photographic images with their geographic coordinates could be beneficial to people who did not know where a photograph was taken or where a particular landmark was. This could still fit within the tourist directed design of Realise as it is possible that people might forget where they were when they took their photographs during a trip. Another possibility is that there is a less well known place that they want to visit and have a photograph of it, but do not know exactly where it is.

Another photograph search system is 'Pixolu' which searches images based on a given search term.[4] The search term is compared against Flickr tags and the resulting images are all taken from the Flickr website.[5] What makes pixolu unusual is that the users can filter the results by selecting a few which better match what they are looking for. Once they

[3] `http://www.tineye.com`

[4] `http://www.pixolu.de`

[5] `http://www.flickr.com`

select some images, a new result set is returned with photos that are similar to those which were chosen. This concept might be beneficial to Realise if users want images of a specific landmark rather than a set of images from the same location. By first presenting the location results, Realise could allow users to select the image that most closely matched what they were looking for. An image matching system could then return other images from that location which matched the chosen picture. While Realise is mainly a location based image based search engine, this functionality could help users to locate photographs of a specific landmark from within the location based results.

7.3 – Conclusions

In this chapter, it was noted that there are common problems in games with by-products and that this dissertation has described ways to address these problems. The EyeSpy game addressed the problem of using games with by-products to collect and classify useful contextual information (addressing ***RQ2***) and it was hoped that letting the users create the tasks they had to perform would make the game less monotonous, but this did not prove to be the case. The game also continued to produce broad by-products and could not respond to the need for specific ones.

The introduction of mutually reinforcing systems allowed for more diversity in game-play due to the motivations from non-game users. It also introduced a way to request specific by-products (addressing ***RQ1***). This addressed the principal problems that had been identified in games with by-products, but also introduced its own. The main new problem was maintaining balance between the systems. It had to be ensured that there were not too many requests sent to EyeSpy and that EyeSpy could respond to requests within a reasonable amount of time. A solution to this problem was presented called the subjective reward system. This involved limiting the number of requests that could be made based on the overall activity of the two systems, but also allowed the requesters to prioritize their needs. Additionally, it took much of the validation away from the game and placed it in the hands of the requesters such that they could decide whether or not the game players would receive points for their efforts.

A second problem with mutually reinforcing systems is that the current designs necessitate a lot of inter-system trust. While this may be fine when both systems are designed and maintained by the same people, it might prevent different designers from sharing their systems' benefits with each other. It was also noted that privacy may be a greater issue in mutually reinforcing systems than other games with by-products because of the communication of data between systems, meaning that the data may not be controlled by the system in which it was created.

The common designs of games with by-products were then extended and addressed from the perspective of mutually reinforcing systems in order to discuss issues that others might face in designing their own mutually reinforcing systems. Potential future research was also presented, highlighting problems which had yet to be addressed and which may also affect future mutually reinforcing systems.

While this chapter has shown that our research has addressed ***RQ1*** and ***RQ2***, it has also been shown that there are aspects which still require attention. Although EyeSpy was largely successful in extending human computation to collect and classify useful contextual information in mobile environments, as required by ***RQ2***, the concept of mutually reinforcing systems only went part way to addressing ***RQ1***. It was posed in ***RQ1*** that this dissertation should find a way to improve human computation to match the specific needs of other systems. In a literal sense, mutually reinforcing systems did indeed address ***RQ1***, but they showed that the wider problem went beyond the original question. The issues of trust, privacy and equilibrium between systems become more significant as the number of systems increases. Even within the two systems presented in this dissertation, a fully balanced solution was difficult to achieve (and never perfectly demonstrated).

Although this dissertation has successfully addressed the research questions, it has also highlighted that a true solution goes beyond what was stated in the questions themselves. This indicates that future research would be required to fully answer all the issues which this dissertation has raised beyond the initial research questions.

Chapter 8
Conclusion

This dissertation has discussed the way that computer games have an important cultural significance and can be created to produce human computed by-products as a side effect of play. While these games have proved successful, the by-products are usually produced ahead of time for predicted use in another system, rather than responding to the specific needs of those systems at the time.

This dissertation has also discussed the way that mobile environments provide great potential for such games as they might be used to address the computationally difficult task of collecting and classifying good contextual information. By setting out the common design elements of previous games with by-products, the research described in this dissertation attempted to address the problem of creating a mobile game which could create by-products that responded to the immediate needs of another system.

Using the common designs of previous systems as a guideline, a game with by-products was built called EyeSpy. A particular contribution of this research was that, unlike previous games with by-products, EyeSpy was set in a mobile environment and had the players create, validate and label image by-products that were contextually relevant to that environment. Because these by-products were designed to be confirmed by other players, they were generally of easily identifiable and recognizable places, but the decision of what places to use for creating by-products was left entirely at the discretion of the players. This system proved successful in producing useful by-products and being enjoyable to play. However, it could not respond to the specific needs of another system and became monotonous over time.

In order to address these problems, the concept of mutually reinforcing systems was introduced. This forms the central contribution of this dissertation. Demonstration of the concept required the construction of a second system, Realise, that used the by-products from EyeSpy and allowed its users to request new ones as needed. This required an additional game-play element to be added to EyeSpy in order to incorporate this feature. In this way, Realise users would provide additional game content for EyeSpy and the game would respond to holes in the data-set, as identified by the Realise users. This allowed the systems to mutually reinforce each other and work together to achieve their goals. EyeSpy could now respond to the specific needs of another system, and the additional game-play element helped to make the game less monotonous over time. However, this particular implementation of mutually reinforcing systems had problems of its own which needed to be addressed. The system for validating the new game-play element took too long and Realise users were frustrated that they did not have more control over whether or not the new by-products matched their requests.

To address these issues, the power of accepting responses was placed with the Realise users and a system to achieve this was incorporated into the existing design. This augmentation was called the subjective reward system. This system was designed to provide a faster means of Realise users getting responses to their requests by allowing them to prioritize their requests with points values, and having responses immediately sent to them for approval rather than having the game validate the images first. In order to ensure that users prioritized their requests properly, a limitation on the number of requests was made, as well as a limitation on the points that could be allocated across their requests. These limitations were set dynamically depending on how many requests were currently awaiting responses and how quickly responses were being created for them. However, this final aspect of the subjective reward system was implemented incorrectly and a working demonstration of this element was not built and tested within the time limits of this research. It is proposed that, even though further refinements are desirable, the deployed mutually reinforcing systems implementation forms another contribution of this dissertation, by showing concrete evidence that the mutually reinforcing systems concept can be implemented in a useful way.

It was also noted that the current implementation of mutually reinforcing systems requires trust between the systems and this may prove to be problematic in practical use if all the member systems are not designed and maintained by the same people. Some suggestions for future research to address this problem were posed in Section 7.2.

This dissertation opened by posing two research questions. These were intended to focus the research such that it might lead to potential solutions to the problems. These research questions were stated in Section 1.1:

RQ1. *How can the results of human computation be improved to match the specific needs of other systems?*

RQ2. *How can human computation be extended to collect and classify useful contextual information in mobile environments?*

RQ2 was addressed through the creation of EyeSpy: a mobile game with by-products. This game allowed users to collect information in a mobile environment that they felt was contextually relevant (in that it would be easily recognizable and findable by others). The game also incorporated a means for the users to validate each other's choices and took advantage of the sensor mechanisms on the mobile devices to allow connections to be made between the information that was collected. For example, given a photograph or text description, it would be possible to show other photographs and textual descriptions from the same area. In this way, the game used humans to both collect and classify useful contextual information from a mobile environment, thus addressing ***RQ2***.

RQ1 was addressed through the creation of Realise and the concept of mutually reinforcing systems. By giving Realise users the ability to make direct requests of the EyeSpy players, and by augmenting EyeSpy to incorporate a new game dynamic that would allow them to respond, the game is now able to match the specific needs of another system, which was the requirement of ***RQ1***.

However, there are still aspects of the presented solution that require further attention. While mutually reinforcing systems do allow the results of human computation to match the specific needs of other systems, a method is needed to manage those needs such that they do not overwhelm the human computation component. A finite number of people will be involved in the human computation and too many requests will cause a lengthy wait before all the requests can be dealt with.

The proposed solution to this problem was the subjective reward system. This allowed people to prioritize their needs and prevented too many requests from being made (determined by what the systems were currently able to cope with). However, this concept requires further research and has not yet been demonstrated to be successful. This means that, while this dissertation has achieved the goal put forward by ***RQ1***, future research will need to be carried out before there is a fully viable solution.

By addressing these research questions, the contribution of the research described in this dissertation has been, overall, to extend existing human computation techniques to allow the collection and classification of useful contextual information in mobile environments and to allow the results of this extension to match the specific needs of another system. The dissertation has also highlighted areas of future research that will contribute to making a fully viable solution for other designers who need to create such systems of their own.

By taking human computation beyond the desktop systems of previous games with by-products, this work has opened up a new research area where physicality, local knowledge and environmental adaptability can all be used as we make computer systems outsource tasks to humans. Additionally, by extending human computation to produce valid subjective by-products rather than objectively validated results, this dissertation has demonstrated the possibility for using human computation in areas that are highly subjective, such as those involving art, music, opinion and taste. While refinements and generalizations inevitably have to be done, these outcomes broaden the field of human computation significantly, and extend the potential range of applications of this as yet under-explored and under-exploited research area.

Appendix A: List of Publications

The following is a list of the author's publications that have resulted from or influenced the work in this dissertation.

J. Ferguson, M. Bell, and M. Chalmers, "Mutually reinforcing systems," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 34–37.

M. Bell, S. Reeves, B. Brown, S. Sherwood, D. MacMillan, J. Ferguson, and M. Chalmers, "Eyespy: supporting navigation through play," in *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*. New York, NY, USA: ACM, April 2009, pp. 123–132.

Bibliography

[1] R. Van Eck, "Digital game-based learning: It's not just the digital natives who are restless," *EDUCAUSE Review*, vol. 41, no. 2, pp. p16–18, 20, 22, 24, 26, 28, 30, March 2006.

[2] L. von Ahn, "Games with a purpose," *Computer*, vol. 39, no. 6, pp. 92–94, 2006.

[3] comScore, "comScore reports February 2010 U.S. mobile subscriber market share," Press Release, April 2010. [Online]. Available: http://www.comscore.com/Press_Events/Press_Releases/2010/4/comScore_Reports_February_2010_U.S._Mobile_Subscriber_Market_Share

[4] comScore, "Smartphone adoption shifting dynamics of U.S. mobile gaming market," Press Release, April 2010. [Online]. Available: http://www.comscore.com/Press_Events/Press_Releases/2010/4/Smartphone_Adoption_Shifting_Dynamics_of_U.S._Mobile_Gaming_Market

[5] D. Goldin, S. A. Smolka, and P. Wegner, Eds., *Interactive Computation: The New Paradigm*. Springer Berlin / Heidelberg, 2006.

[6] L. von Ahn and L. Dabbish, "Labeling images with a computer game," in *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press, 2004, pp. 319–326.

[7] J. Howe, *Crowdsourcing - How the Power of the Crowd is Driving the Future of Business*. Crown Business, 2008.

[8] D. Michael and S. Chen, *Serious games: Games that educate, train, and inform*. Boston, MA: Thomson Course Technology, 2006.

[9] M. Chalmers, "When cookies aren't enough: Tracking and enriching web activity with recer," in *Preferred Placement: Knowledge Politics on the Web. Maastricht: Jan van Eyck Academie Editions*, 2000, pp. 99–102.

[10] J. T. Hancock, C. Toma, and N. Ellison, "The truth about lying in online dating profiles," in *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2007, pp. 449–452.

[11] J. Giles, "Internet encyclopaedias go head to head," *Nature*, vol. 438, no. 7070, pp. 900–901, December 2005.

[12] Y. Amichai–Hamburger, N. Lamdan, R. Madiel, and T. Hayat, "Personality characteristics of wikipedia members," *CyberPsychology & Behavior*, vol. 11, no. 6, pp. 679–681, December 2008.

[13] P. Denning, J. Horning, D. Parnas, and L. Weinstein, "Wikipedia risks," *Communications of the ACM*, vol. 48, no. 12, pp. 152–152, 2005.

[14] F. B. Viégas, M. Wattenberg, and K. Dave, "Studying cooperation and conflict between authors with history flow visualizations," in *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2004, pp. 575–582.

[15] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, "Turkit: tools for iterative tasks on mechanical turk," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 29–30.

[16] M. S. Silberman, J. Ross, L. Irani, and B. Tomlinson, "Sellers' problems in human computation markets," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 18–21.

[17] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 64–67.

[18] W. Mason and D. J. Watts, "Financial incentives and the 'performance of crowds'," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 77–85.

[19] S. Albers, *Online Algorithms*, ser. Interactive Computation - The New Paradigm. Springer Berlin / Heidelberg, 2006, ch. 7, pp. 143–164.

[20] A. Bassoli, J. Brewer, K. Martin, P. Dourish, and S. Mainwaring, "Underground aesthetics: Rethinking urban computing," *Pervasive Computing, IEEE*, vol. 6, no. 3, pp. 39–45, 2007.

[21] M. Chalmers, "Hermeneutics, information and representation," *European Journal of Information Systems*, vol. 13, pp. 210–220, September 2004.

[22] N. Auray, "Folksonomy: the new way to serendipity," *International Journal of Digital Economics*, no. 65, pp. 67–88, August 2007.

[23] L. von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," *Communications of the ACM*, vol. 47, no. 2, pp. 56–60, 2004.

[24] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: Human-based character recognition via web security measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, September 2008.

[25] P. Faymonville, K. Wang, J. Miller, and S. Belongie, "CAPTCHA-based image labeling on the soylent grid," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 46–49.

[26] S. Steinbach, V. Rabaud, and S. Belongie, "Soylent grid: it's made of people!" in *ICV 2007, in conjunction with ICCV 2007*, University of California, San Diego, La Jolla, CA 92093, USA, 2007.

[27] D. Morrison, S. Marchand-Maillet, and E. Bruno, "TagCaptcha: annotating images with CAPTCHAs," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 44–45.

[28] C. Hu, B. B. Bederson, and P. Resnik, "Translation by iterative collaboration between monolingual users," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 54–55.

[29] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, "Exploring iterative and parallel human computation processes," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 68–76.

[30] L. von Ahn, M. Kedia, and M. Blum, "Verbosity: a game for collecting commonsense facts," in *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA: ACM Press, 2006, pp. 75–78.

[31] F. Arbab, *Computing and Interaction*, ser. Interactive Computation - The New Paradigm. Springer Berlin / Heidelberg, 2006, ch. 2, pp. 9–23.

[32] C. Mackay, *Extraordinary Popular Delusions and the Madness of Crowds*. London : Harrap, 1956., from the editions of 1841 and 1852, originally published as: Memoirs of extraordinary popular delusions: London: R. Bentley 1841.

[33] H. Orman and R. Schroeppel, "Positive feedback and the madness of crowds," in *NSPW '96: Proceedings of the 1996 workshop on New security paradigms*. New York, NY, USA: ACM, 1996, pp. 134–138.

[34] E. Huang, H. Zhang, D. C. Parkes, K. Z. Gajos, and Y. Chen, "Toward automatic task design: a progress report," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 77–85.

[35] J. Ross, L. Irani, M. S. Silberman, A. Zaldivar, and B. Tomlinson, "Who are the crowdworkers?: shifting demographics in mechanical turk," in *CHI EA '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2010, pp. 2863–2872.

[36] C. K. Olson, "Children's motivations for video game play in the context of normal development." *Review of General Psychology*, vol. 14, no. 2, pp. 180–187, 2010.

[37] M. Bell, M. Hall, M. Chalmers, P. Gray, and B. Brown, "Domino: Exploring mobile collaborative software adaptation." in *PERVASIVE'06: Proceedings of the 4th international conference on Pervasive computing*, 2006, pp. 153–168.

[38] A. Takhtamysheva, R. Porzel, and M. Krause, "Games for games: manipulating contexts in human computation games," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 38–39.

[39] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, and Z. Popovic, "Predicting protein structures with a multiplayer online game," *Nature*, vol. 466, no. 7307, pp. 756–760, August 2010.

[40] T. Susi, M. Johannesson, and P. Backlund, “Serious games - an overview,” University of Skövde, Tech. Rep. HS- IKI -TR-07-001, 2007.

[41] R. A. Ballagas, S. G. Kratz, J. Borchers, E. Yu, S. P. Walz, C. O. Fuhr, L. Hovestadt, and M. Tann, “Rexplorer: a mobile, pervasive spell-casting game for tourists,” in *CHI ’07: Extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM Press, 2007, pp. 1929–1934.

[42] B. Brown and M. Chalmers, “Tourism and mobile technology,” in *ECSCW 2003: Proceedings of the eigth european conference on computer supported cooperative work*. Helsinki, Finland: Dordrecht: Kluwer Academic Press, 2003, pp. 335–355.

[43] J. Lin, L. Mamykina, S. Lindtner, G. Delajoux, and H. Strub, “Fish’n’steps: Encouraging physical activity with an interactive computer game,” in *UbiComp 2006: Ubiquitous Computing*, ser. Lecture Notes in Computer Science, P. Dourish and A. Friday, Eds., vol. 4206. Springer Berlin / Heidelberg, 2006, pp. 261–278.

[44] J. Hoysniemi, “International survey on the dance dance revolution game,” *Computers in Entertainment (CIE)*, vol. 4, no. 2, p. 8, 2006.

[45] A. Sall and R. Grinter, “Let’s get physical! in, out and around the gaming circle of physical gaming at home,” *Computer Supported Cooperative Work (CSCW)*, vol. 16, no. 1, pp. 199–229, April 2007.

[46] G. W. Furnas, C. Fake, L. von Ahn, J. Schachter, S. Golder, K. Fox, M. Davis, C. Marlow, and M. Naaman, “Why do tagging systems work?” in *CHI ’06: Extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM Press, 2006, pp. 36–39.

[47] L. Weng and F. Menczer, “GiveALink tagging game: an incentive for social annotation,” in *HCOMP ’10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 26–29.

[48] Y.-l. Kuo, J.-C. Lee, K.-y. Chiang, R. Wang, E. Shen, C.-w. Chan, and J. Y.-j. Hsu, “Community-based game design: experiments on social games for commonsense data collection,” in *HCOMP ’09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 15–22.

[49] M. Krause and H. Aras, "Playful tagging: folksonomy generation using online games," in *WWW '09: Proceedings of the 18th international conference on World wide web*. New York, NY, USA: ACM, April 2009, pp. 1207–1208.

[50] L. von Ahn and L. Dabbish, "Designing games with a purpose," *Communications of the ACM*, vol. 51, no. 8, pp. 58 – 67, August 2008.

[51] P. Wegner, "Why interaction is more powerful than algorithms," *Communications of the ACM*, vol. 40, no. 5, pp. 80–91, 1997.

[52] A. M. Turing, "Computing machinery and intelligence," *MIND*, vol. LIX, no. 236, pp. 433–460, October 1950.

[53] C.-J. Ho, T.-H. Chang, J.-C. Lee, J. Y.-j. Hsu, and K.-T. Chen, "KissKissBan: a competitive human computation game for image annotation," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 11–14.

[54] E. L. M. Law, L. von Ahn, R. B. Dannenberg, and M. Crawford, "TagATune: A game for music and sound annotation," in *ISMIR 2007: 8th International Conference on Music Information Retrieval*, School of Computer Science, Carnegie Mellon University, 2007.

[55] P. N. Bennett, D. M. Chickering, and A. Mityagin, "Learning consensus opinion: mining data from a labeling game," in *WWW '09: Proceedings of the 18th international conference on World wide web*. New York, NY, USA: ACM, 2009, pp. 121–130.

[56] P. N. Bennett, D. M. Chickering, and A. Mityagin, "Picture this: preferences for image search," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 25–26.

[57] L. von Ahn, R. Liu, and M. Blum, "Peekaboom: a game for locating objects in images," in *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA: ACM Press, 2006, pp. 55–64.

[58] L. von Ahn, S. Ginosar, M. Kedia, R. Liu, and M. Blum, "Improving accessibility of the web with a computer game," in *CHI '06: Proceedings of the SIGCHI conference*

on Human Factors in computing systems. New York, NY, USA: ACM Press, 2006, pp. 79–82.

[59] E. Law and L. von Ahn, "Input-agreement: a new mechanism for collecting data using human computation games," in *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*. New York, NY, USA: ACM, 2009, pp. 1197–1206.

[60] M. I. Mandel and D. P. W. Ellis, "A web-based game for collecting music metadata," *Journal of New Music Research*, vol. 37, no. 2, pp. 151 – 165, June 2008.

[61] J. Yan and S.-Y. Yu, "Magic bullet: a dual-purpose computer game," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 32–33.

[62] E. Law, L. von Ahn, and T. Mitchell, "Search war: a game for improving web search," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 31–31.

[63] A. Dasdan, C. Drome, and S. Kolay, "Thumbs-up: a game for playing to rank search results," in *WWW '09: Proceedings of the 18th international conference on World wide web*. New York, NY, USA: ACM, April 2009, pp. 1071–1072.

[64] A. Dasdan, C. Drome, S. Kolay, M. Alpern, A. Han, T. Chi, J. Hoover, I. Davtchev, and S. Verma, "Thumbs-up: a game for playing to rank search results," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 36–37.

[65] H. Ma, R. Chandrasekar, C. Quirk, and A. Gupta, "Page hunt: using human computation games to improve web search," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 27–28.

[66] H. Aras, M. Krause, A. Haller, and R. Malaka, "Webpardy: harvesting QA by HC," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 49–52.

[67] N. Seemakurty, J. Chu, L. von Ahn, and A. Tomasic, "Word sense disambiguation via human computation," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 60–63.

[68] J. Chamberlain, M. Poesio, and U. Kruschwitz, "A demonstration of human computation using the phrase detectives annotation game," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 23–24.

[69] K. Siorpaes and M. Hepp, "Games with a purpose for the semantic web," *IEEE Intelligent Systems*, vol. 23, no. 3, pp. 50–60, 2008.

[70] S. Casey, B. Kirman, and D. Rowland, "The gopher game: a social, mobile, locative game with user generated content and peer review," in *ACE '07: Proceedings of the international conference on Advances in Computer Entertainment technology*. New York, NY, USA: ACM, 2007, pp. 9–16.

[71] V. H. Tuulos, J. Scheible, and H. Nyholm, "Combining web, mobile phones and public displays in large-scale: manhattan story mashup," in *PERVASIVE'07: Proceedings of the 5th international conference on Pervasive computing*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 37–54.

[72] B. Koleva, I. Taylor, S. Benford, M. Fraser, C. Greenhalgh, H. Schnädelbach, D. vom Lehn, C. Heath, J. Row-Farr, and M. Adams, "Orchestrating a mixed reality performance," in *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI '01. New York, NY, USA: ACM, 2001, pp. 38–45.

[73] R. D. Balicer, "Modeling infectious diseases dissemination through online role-playing games," *Epidemiology*, vol. 18, no. 2, pp. 260–261, March 2007.

[74] E. T. Lofgren and N. H. Fefferman, "The untapped potential of virtual game worlds to shed light on real world epidemics," *The Lancet Infectious Diseases*, vol. 7, no. 9, pp. 625–629, September 2007.

[75] B. Brown and M. Bell, "CSCW at play: 'there' as a collaborative virtual environment," in *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*. New York, NY, USA: ACM Press, 2004, pp. 350–359.

[76] B. Brown and M. Bell, "Social interaction in 'there'," in *CHI '04: Extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM Press, 2004, pp. 1465–1468.

[77] A. Crabtree, S. Benford, M. Capra, M. Flintham, A. Drozd, N. Tandavanitj, M. Adams, and J. R. Farr, "The cooperative work of gaming: Orchestrating a mobile SMS game," *Computer Supported Cooperative Work (CSCW)*, vol. 16, no. 1-2, pp. 167–198, 2007.

[78] M. Flintham, S. Benford, R. Anastasi, T. Hemmings, A. Crabtree, C. Greenhalgh, N. Tandavanitj, M. Adams, and J. Row-Farr, "Where on-line meets on the streets: experiences with mobile mixed reality games," in *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2003, pp. 569–576.

[79] L. Barrington, D. O'Malley, D. Turnbull, and G. Lanckriet, "User-centered design of a social game to tag music," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 7–10.

[80] A. K. Dey and G. Abowd, "Towards a better understanding of context and context-awareness," in *1st International Symposium on Handheld and Ubiquitous Computing (HUC '99)*, Karlsruhe, Germany, 1999, pp. 304 – 307.

[81] J. Froehlich, M. Chen, I. Smith, and F. Potter, "Voting with your feet: An investigative study of the relationship between place visit behavior and preference," *UbiComp 2006: Ubiquitous Computing*, vol. 4206/2006, pp. 333–350, September 2006.

[82] M. Adams, T. Cox, G. Moore, B. Croxford, M. Refaee, and S. Sharples, "Sustainable soundscapes: Noise policy and the urban experience," *Urban Studies*, vol. 43, no. 13, pp. 2385 – 2398, December 2006.

[83] P. Alexanderson and K. Tollmar, "Being and mixing: designing interactive soundscapes," in *NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction*. New York, NY, USA: ACM Press, 2006, pp. 252–261.

[84] A. Butz and R. Jung, "Seamless user notification in ambient soundscapes," in *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*. New York, NY, USA: ACM Press, 2005, pp. 320–322.

[85] I. Ekman, L. Ermi, J. Lahti, J. Nummela, P. Lankoski, and F. F. Mäyrä, "Designing sound for a pervasive mobile game," in *ACE '05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*. New York, NY, USA: ACM Press, 2005, pp. 110–116.

[86] S. Benford, A. Crabtree, M. Flintham, A. Drozd, R. Anastasi, M. Paxton, N. Tandavanitj, M. Adams, and J. Row-Farr, "Can you see me now?" *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 13, no. 1, pp. 100–133, 2006.

[87] M. Chang and E. Goodman, "Fiasco: game interface for location-based play," in *DIS '04: Proceedings of the 2004 conference on Designing interactive systems*. New York, NY, USA: ACM Press, 2004, pp. 329–332.

[88] S. Matyas, "Playful geospatial data acquisition by location-based gaming communities," *International Journal of Virtual Reality*, vol. 6, no. 3, pp. 1 – 10, March 2007.

[89] S. Matyas, C. Matyas, P. Kiefer, C. Schlieder, H. Mitarai, and M. Kamata, "Designing location-based mobile games with a purpose - collecting geospatial data with CityExplorer," in *ACE 2008, Yokohama, Japan, 3-5 December, ACM proceedings*, December 2008.

[90] K. O'Hara, "Understanding geocaching practices and motivations," in *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2008, pp. 1177–1186.

[91] L. Barkhuus, M. Chalmers, P. Tennent, M. Hall, M. Bell, S. Sherwood, and B. Brown, "Picking pockets on the lawn: The development of tactics and strategies in a mobile game." in *UbiComp 2005: Ubiquitous Computing*, 2005, pp. 358–374.

[92] M. Chalmers, M. Bell, B. Brown, M. Hall, S. Sherwood, and P. Tennent, "Gaming on the edge: using seams in ubicomp games," in *ACE '05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment technology*. New York, NY, USA: ACM, 2005, pp. 306–309.

[93] M. Bell, M. Chalmers, L. Barkhuus, M. Hall, S. Sherwood, P. Tennent, B. Brown, D. Rowland, and S. Benford, "Interweaving mobile games with everyday life," in *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA: ACM Press, 2006, pp. 417–426.

[94] K. Tuite, N. Snavely, D.-Y. Hsiao, A. M. Smith, and Z. Popović, "Reconstructing the world in 3d: bringing games with a purpose outdoors," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 41–44.

[95] Y. Arase, X. Xie, M. Duan, T. Hara, and S. Nishio, "A game based approach to assign geographical relevance to web images," in *WWW '09: Proceedings of the 18th international conference on World wide web*. New York, NY, USA: ACM, April 2009, pp. 811–820.

[96] S. Harrison and P. Dourish, "Re-place-ing space: the roles of place and space in collaborative systems," in *CSCW '96: Proceedings of the 1996 ACM conference on Computer supported cooperative work*. New York, NY, USA: ACM Press, 1996, pp. 67–76.

[97] J. Hightower, "From position to place," in *Proceedings of The 2003 Workshop on Location-Aware Computing*, October 2003, pp. 10–12, part of the 2003 Ubiquitous Computing Conference.

[98] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, and J. Hughes, "Learning and recognizing the places we go," *UbiComp 2005: Ubiquitous Computing*, vol. 3660/2005, pp. 159–176, August 2005.

[99] C. Licoppe and Y. Inada, "Emergent uses of a multiplayer location-aware mobile game: the interactional consequences of mediated encounters," *Mobilities*, vol. 1, no. 1, pp. 39 – 61, March 2006.

[100] W. Broll, J. Ohlenburg, I. Lindt, I. Herbst, and A.-K. Braun, "Meeting technology challenges of pervasive augmented reality games," in *NetGames '06: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*. New York, NY, USA: ACM Press, 2006, p. 28.

[101] N. Marmasse and C. Schmandt, "Location-aware information delivery with Com-Motion," in *HUC '00: Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*. London, UK: Springer-Verlag, 2000, pp. 157–171.

[102] P. Tennent, M. Hall, B. Brown, M. Chalmers, and S. Sherwood, "Three applications for mobile epidemic algorithms," in *MobileHCI '05: Proceedings of the 7th interna-*

tional conference on Human computer interaction with mobile devices & services. New York, NY, USA: ACM Press, 2005, pp. 223–226.

[103] C. Licoppe and R. Guillot, *Mobile Technologies of the City*. Routledge, February 2006, ch. ICTs and the Engineering of Encounters: A Case Study of the Development of a Mobile Game Based on the Geolocation of Terminals, pp. 152 – 176.

[104] J. Falk, P. Ljungstrand, S. Björk, and R. Hansson, "Pirates: proximity-triggered interaction in a multi-player game," in *CHI '01: Extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2001, pp. 119–120.

[105] S. Robertson, M. Vojnovic, and I. Weber, "Rethinking the ESP game," in *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2009, pp. 3937–3942.

[106] M. Krause, A. Takhtamysheva, M. Wittstock, and R. Malaka, "Frontiers of a paradigm: exploring human computation with digital games," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 22–25.

[107] B. G. Morton, J. A. Speck, E. M. Schmidt, and Y. E. Kim, "Improving music emotion labeling using human computation," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 45–48.

[108] L. Vincent, "Taking online maps down to street level," *Computer*, vol. 40, pp. 118–120, December 2007.

[109] J. Kopf, B. Chen, R. Szeliski, and M. Cohen, "Street slide: browsing street level imagery," in *ACM SIGGRAPH 2010 papers*, ser. SIGGRAPH '10. New York, NY, USA: ACM, 2010, pp. 96:1–96:8.

[110] P. Jarkiewicz, M. Frankhammar, and Y. Fernaeus, "In the hands of children: exploring the use of mobile phone functionality in casual play settings," in *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*. New York, NY, USA: ACM, 2008, pp. 375–378.

[111] L. Grant, H. Daanen, S. Benford, A. Hampshire, A. Drozd, and C. Greenhalgh, "MobiMissions: the game of missions for mobile phones," in *ACM SIGGRAPH 2007 educators program*, ser. SIGGRAPH '07. New York, NY, USA: ACM, 2007.

[112] S. Benford, B. B. Bederson, K.-P. Åkesson, V. Bayon, A. Druin, P. Hansson, J. P. Hourcade, R. Ingram, H. Neale, C. O'Malley, K. T. Simsarian, D. Stanton, Y. Sundblad, and G. Taxén, "Designing storytelling technologies to encouraging collaboration between young children," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI '00. New York, NY, USA: ACM, 2000, pp. 556–563.

[113] E. Laurier and B. Brown, *Transactions of the Institute of British Geographers*. Blackwell Publishing, 2008, no. 33, ch. Rotating maps and readers: praxiological aspects of alignment and orientation, pp. 201–21.

[114] B. Brown and E. Laurier, *Map-based mobile services – Theories, Methods and Implementations*. Springer Berlin Heidelberg, December 2005, ch. Designing electronic maps: an ethnographic approach, pp. 241–257.

[115] M. Toomim and J. A. Landay, "Measuring utility of human-computer interaction," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 53–53.

[116] H. Sacks, E. A. Schegloff, and G. Jefferson, "A simplest systematics for the organization of turn-taking for conversation," *Language*, vol. 50, no. 4, pp. 696–735, 1974.

[117] H. Garfinkel, *Studies in Ethnomethodology*. Prentice Hall, 1967, ch. Some rules of correct decision making that jurors respect, pp. 104–115.

[118] K. Tollmar, T. Yeh, and T. Darrell, "IDeixis – searching the web with mobile images for location-based information," in *MobileHCI '04: Proceedings of the 6th international conference on Human computer interaction with mobile devices and services*, vol. 3160/2004. Springer Berlin / Heidelberg, 2004, pp. 288–299.

[119] A. Beeharee and A. Steed, "Exploiting real world knowledge in ubiquitous applications," *Personal and Ubiquitous Computing*, vol. 11, no. 6, pp. 429–437, Aug. 2007.

[120] B. Brown, M. Chalmers, M. Bell, M. Hall, I. MacColl, and P. Rudman, "Sharing the square: Collaborative leisure in the city streets," in *ECSCW '05: Proceedings of the Ninth European Conference on Computer-Supported Cooperative Work*, H. Gellersen, K. Schmidt, M. Beaudouin-Lafon, and W. Mackay, Eds. Springer Netherlands, September 2005, pp. 427–447.

[121] M. Bell, M. Chalmers, B. Brown, I. MacColl, M. Hall, and P. Rudman, "Sharing photos and recommendations in the city streets," in *ECHISE '05: 1st International Workshop on Exploiting Context Histories in Smart Environments*, 2005, part of PERVASIVE '05: 3rd International Conference on Pervasive Computing.

[122] P. Suber, *The Paradox of Self-Amendment*. Lang, 1990.

[123] B. Jackson, "Locational privacy and wholesale surveillance via photo services," in *The Next HOPE: The 8th Hackers On Planet Earth Conference*, New York City, July 2010.

[124] L.-J. Chen, B.-C. Wang, and K.-T. Chen, "The design of puzzle selection strategies for GWAP systems," *Concurrency and Computation: Practice & Experience*, vol. 22, no. 7, pp. 890–908, May 2010.

[125] R. Macefield, "Usability studies and the hawthorne effect," *Journal of Usability Studies*, vol. 2, no. 3, pp. 145–154, May 2007.

[126] G. Leshed, T. Velden, O. Rieger, B. Kot, and P. Sengers, "In-car GPS navigation: engagement with and disengagement from the environment," in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 1675–1684.

[127] A. Beeharee and A. Steed, *Human-Computer Interaction. Interaction Platforms and Techniques*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, August 2007, vol. 4551/2007, ch. Minimising Pedestrian Navigational Ambiguities Through Geoannotation and Temporal Tagging, pp. 748–757.

[128] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," *IEEE Symposium on Security and Privacy*, pp. 173–187, 2009.

[129] T. Burnham and R. Sami, "A reputation system for selling human computation," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 54–57.

[130] N. D. Lane, D. Lymberopoulos, F. Zhao, and A. T. Campbell, "Hapori: Context-based local search for mobile phones using community behavioral modeling and similarity," in *Ubicomp 2010*. ACM Press, September 2010.