SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

FIIT-5212-5736

Martin Polakovič

Game for linking resources and metadata in multimedia domain

Bachelor Thesis

Degree Program: Informatics

Degree Course: 9.2.1 Informatics

Workplace: Institute of Informatics and Software Engineering, FIIT STU, Bratislava

Supervisor: Ing. Jakub Šimko, PhD.

2015, May

Anotácia

Slovenská Technická Univerzita v Bratislave

Fakulta Informatiky a Informačných Technológií

Študijný program: Informatika

Autor: Martin Polakovič

Bakalársky projekt: Hra na prepájanie zdrojov a metadát v doméne multimédií

Vedúci bakalárskeho projektu: Ing. Jakub Šimko, PhD.

máj, 2015

Manipulácia a manažment dát vyžaduje, aby stroje rozumeli obsahu spracúvaných dát a tiež kontextu v ktorom daný obsah vystupuje. Toto umožňujú metadáta (dáta o dátach), ktoré identifikujú potrebné charakteristiky obsahu. Získanie metadát však predstavuje problém, pretože samotné vyžaduje znalosť a chápanie obsahu, ktorý metadata budú opisovať. Niektoré metadáta nie je možné vygenerovať automaticky a je nutné využiť ľudské znalosti a schopnosti. Toto je však problém v súvislosti s množstvom dát ktoré treba analyzovať. Pre prekonanie tohto problému existujú tzv. crowdsourcing riešenia, teda riešenie využívajúce schopnosti väčšieho množstva ľudí naraz. V tejto kategórií riešení sa nachádzajú aj hry s účeľom, ktoré sú schopné využiť vedomosti alebo schopnosti samotných hráčov, ktorí tak vedome alebo nevedome ponúkajú riešenia pre problémy ktore boli transformované do hernej mechaniky hry. Hry s účeľom sú uznávané ako fungujúce riešenie pre získavanie metadát a našli svoje uplatnenie v praxi.

Táto práca sa zameriava na získavanie multimediálnych metadát, špecificky metadát sémantiky obsahu obrázkov. Cieľom práce je zhodnotiť niektoré zdielané charakteristiky obsahu naprieč množinou obrázkov. Ďalší cieľ je získat dáta o relatívnej podobnosti jednotlivých orázkov. Tieto dáta sú potom použiteľné pri odporúčacích systémoch a systémoch, alebo systémoch, ktoré kategorizujú a manipulujú s danými obrázkami.

Práca prezentuje hru s účeľom ako riešenie problému získania sémantiky pre užšie zamerané množiny obrázkov, kde je nutné hlbšie rozumenie obsahu a správne identifikovanie detailov, ktoré odlišujú obrázky na jemnejšej úrovni. Daná hra je dosková hra pre viacerých hráčov, kde hráči manipulujú s obrázkami v rámci dvojrozmernej plochy. Toto umožňuje hráčom vložiť do hry ich znalosti o obsahu obrázkov. Hráči hrajú na rovnakej hracej ploche a nevedomky si navzájom validujú vlastné akcie. Analýzou absolutnych a relatívnych pozícií obrázkov počas a na konci hry potom vieme extrahovať sémantiku obrázkov. Pri dostatočnom množstve dát je možné zachytiť vlastnosti obsahu naprieč množinou obrázkov.

Annotation

Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

Degree Course: Informatics

Author: Martin Polakovič

Bachelor thesis: Game for linking resources and metadata in multimedia domain

Supervisor: Ing. Jakub Šimko, PhD.

2015, May

Data management and manipulation requires machines to understand the data content and its context. For this to be possible, metadata (data about data), which identifies content properties and relationships must be generated on top of the content. Acquiring metadata can be no easy task, as it requires intricate knowledge and understanding of the very content in the first place. Some metadata can only be acquired manually – by a person, which puts severe limits to scope of the acquisition. Attempting to cross that limit are crowdsourcing solutions, harnessing the capability of a crowd of people. In this category of solutions, there are games with a purpose, which have players play and indirectly solve a problem transformed into a game mechanics. Games with a purpose are now acknowledged solution to metadata acquisition.

This work focuses on metadata acquisition of multimedia content, especially for specific datasets. The aim is to evaluate some characteristics of the content, common for images within the dataset. Another goal is to identify similarity between individual images. This data could be then used for recommender or image management systems.

This work presents game with a purpose as solution for semantics acquisition for more narrowly focused image datasets, which requires deeper understanding of the content and identifying of fine-grained details. Devised game is a 2D board game - minimalistic multiplayer output agreement game with a purpose. Having players manipulate position of images on a 2D board essentially allows players to input image semantics into the game. Players unknowingly validate each other's actions by playing on the same board simultaneously. Estimating image similarity and semantics is then possible from the relative and absolute position of the images throughout or at the end of the game. With sufficient amount of data, evaluating characteristics of the image contents for whole datasets is possible.

I Martin Polakovič, student of Bachelors in the subject of Informatics session 2012-2015 hereby declare that the matter printed in this documentation is my work and has not been printed, published and submitted as research work, thesis, documentation or publication in any form in any University.

Bratislava, May 2015

Martin Polakovič

I want to give my thanks to everyone who helped me or contributed somehow to this work. I also want to thank my advisor, Ing. Jakub Šimko, PhD., for giving me the chance to work on this interesting project and leading me along the way.

Thank you.

Contents

1 Introduction

The emergence of Web 2.0 technologies and their wide adoption has transformed internet from a static medium into dynamic and quickly changing environment. The notion of Web 2.0 denotes a web paradigm beyond static websites. Rather than a standard, it is the way the web content is generated and interacted with. Users, not limited to reading the content, can participate in social exchange within virtual communities. They are no longer simple consumers, instead, they actively engage in a consumer-producer model, where the consumed content itself is created by other users. This form of content is known as user-generated content. User generated content is any kind of data, created by users online. Some of the most common forms are blogs, wikis, forums, chat, audio, images or video.

The continuing, rapid increase in computer hardware capacity goes hand in hand with amount of data generated. In conjunction with web-related technologies and increasing variety of generated data, there is growing importance to efficiently filter, navigate and recognize these data in order to process them. Be it user reading a website, search engine indexing the content or bot processing a wiki page, it is essential to correctly identify the data for what they really are and do this in the least amount of time possible. The user generated content especially is known for its heterogeneous nature, short-lived cycle and containing a lot of waste product. As such, it calls for particular attention. In addition to its plentifulness, it interconnects and references other content, often in many forms and converges those into new forms of content. Internet memes are a good example of new form of content in the age of user generated content. This type of content even necessitates or drives adoption of new technologies[1] and business models. Ultimately, the connection and relatedness between information is of importance, expected only to rise.

To enable content browsing with satisfactory results, it is critical to employ machines and automation to preprocess the content for human access. In order for machines to search and process the content, they must first understand the content and the context of the data they process. Unfortunately, it is virtually impossible to process large quantity of general content in real time to satisfactory results, due to its heterogeneous nature and high semantic complexity, which proved problematic for machines to overcome. In the absence of technologies for recognizing the meaning of the content, the solution is to fall back to auxiliary information about the content carrying resources, metadata. This alleviates the need to analyze the resource multiple times. Once the metadata are gathered, they become representative of the resource instead.

Metadata provide a helping hand in domain of information and resource discovery. They can supplement or represent the resource with necessary information, as a quickly accessible "pre-cached" form. Commonly known metadata are multimedia file tags. Using metadata introduces variety of problems (their structure, format, storage, validity, etc.), but has tremendous benefits. For example: the mitigation of the resource heterogeneity and introduction of a standard for interpretation. Therefore, metadata are utilized ubiquitously, ranging from relational databases and web site indexing to image tags. There is a notion of Semantic Web, a conception of leveraging metadata to transform currently unstructured web into a web of data[2], a machine-understood meta-layer above the human-

[1] Consumer-producer model is compatible with open-source ideas and less strict licensing that promote collaboration and content reuse.

[2] With the help of Resource RDF (resource description framework) and OWL (web ontology language)

readable web. The term Semantic Web[3] is sometimes used as a component or even synonym for Web 3.0.

An important aspect of metadata is the method of their acquisition. The acquisition can be automated or manual (intentional or as a by-product of an activity). The automated methods use machines to acquire the data, but despite the raw computing power of the machines, they are limited by the algorithm. The advancements in the field of artificial intelligence and media recognition promise good results, but generally not yet enough to deal with the problem. On the other hand, manual work can provide good result, but is costly and does not scale very well. The scaling problem is addressed with crowdsourcing solutions (making use of crowds to do the work). However, as these scale up, the participants' expertise can be expected to decrease. This then puts large importance on the design of the crowdsourcing solution.

Attempting to solve the acquisition problem unorthodoxly are games with a purpose (GWAPs). Harnessing the power of crowd, they pass the manual work of individuals off as playing the game, which is supposed to cater to a larger crowd. The games log and analyze player's actions, transforming them into a purposeful result. Naturally, this requires carefully crafted design of the game mechanics and be prepared to tackle some of the problems inherently bound to the approach. This includes the correctness of the result, cheating, insufficient number of players, cold-start problems, problem transformation into game artifacts, etc. The purpose of the game is not necessarily transparent to the player and can even influence more people to play the game, to help realize the game's purpose.

This thesis describes a crowdsourcing solution to metadata acquisition in the image domain, specifically, a game with a purpose for image resource linking and image metadata acquisition. The game attempts to use minimalistic (but extensible) game mechanics to gather semantics of the image content. The player's goal is to reorder small amount of provided images on a 2-dimensional board based on their content. He is then rewarded for matching of image positions with his player-partner, who plays with the same set of images. Game provides game artifacts for annotating the image contents. By analyzing the state of the board and annotations, we can draw conclusions about the relationship between the image resources and moreover, about the semantics of these relationships.

The chapter 2 comprises the problem domain analysis and deals with the notions of semantic metadata, their acquisition and games with the purpose. In the chapter 3 we decompose some of the existing GWAP solutions, show how they relate to our game and point out some of the conclusions about them. Chapter 4 focuses on the specifics of our game, its goals, specifications, game mechanics and implementation. Chapter 5 describes the experiment, including the hypothesis, set up, results and possible modifications for the future. We conclude the thesis in chapter 6.

[3] Semantic Web homepage www.w3.org/2001/sw

2 Multimedia metadata acquisition analysis

2.1 Metadata, semantics and image domain

Metadata are literally "data about data"[4]. Metadata are structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource (1). The metadata referred to in our work is mostly semantic metadata – metadata related or identifying the meaning and content of the data. According to NISO, there are three types of metadata (1):

- Structural: Describes the structure of compound objects, often data about containers for data. They also promote interoperability[5].
- Descriptive: Describes resource for discovery and identification purposes. Elements such as author, title, keywords, etc.
- Administrative: Information to manage resources. E.g. creation date, file type, etc.

A simpler, but perhaps more powerful way to look at what metadata are is: *Metadata are a key to ensuring that resources will survive and continue to be accessible into the future* (1). This is a crucial characteristic for maintaining user-generated content. Semantic metadata is mostly descriptive metadata. Acquiring descriptive metadata is different (and much harder) than in case of administrative metadata, which are mostly generated automatically. Descriptive metadata are tightly related to the notion of semantics and ontologies.

Semantics, the study of meaning, focuses on the relation between signs and symbols (generally called signifiers) and decomposition to their meaning (denotation). In the context of this work, the symbols, whose meaning is denoted are represented by the image content. When we refer to image content, we don't refer to the raw digital data, but to the visual perception of the image, e.g the shapes or objects on the picture. When we refer to semantic information of the image or image content, we refer to the concepts the image content represents and its characteristics. This concept is separate from the raw data or even the content (thus it can not be extracted from it), and is impossible to acquire without solid understanding of the relevant context – external semantic information. For example identifying cat and a ball on the same image is not sufficient to identifying the action of a cat playing with the ball.

Ontology is formal definition and naming of types, attributes and relationships between entities in particular ontology domain. The basic form of ontologies is that of predicate triplets: oriented binary relationships comprised of source entity (subject), target entity (object) and connecting relationship (predicate). These triplets represent facts and can also be composed into more complex ones as well. Ontologies can be used to infer facts from a knowledge base, i.e. for problem solving. The concept of Semantic Web is also based on ontologies.

The value of semantics and ontologies lies in the fact, that they are machine understandable. They can then be utilized in knowledge management and categorization, information visualization or recommender systems. Many web sites such as online stores or wikipedias make use of *taxonomies* (classification of content) to bring structure and hierarchy into their content. Particularly relevant in our case are folksonomies[6] – classification systems for categorization of content en masse. These

[4] Sometimes also "information about information"
[5] XML plays a big role in exchanging structured data on the web
[6] Blend of the words folk and taxonomy

collaboration systems utilize crowdsourcing to annotate content and are also referred to as collaborative tagging or social classification. The idea of folksonomy is actually very close the concept of GWAP. It has both purpose and crowdsourcing behind its back, although instead of playing a game, the users directly annotate the content. Hopefully the underlying intention behind GWAPs as finding synergy between purpose and gaming reward is palpable here. Folksonomies, like wikis, are part of Web 2.0, for their social and collaborative aspects. They are particularly employed in the domain of images and there are several notable websites to mention. To name a few: Flickr[7], Delicious[8], Last.fm[9].

Storing metadata is a problem of ongoing debate. Metadata databases can be used to store the information in a centralized way and can be very efficient. The disadvantage of the approach becomes the separation of the metadata and the resource, meaning a link must be carefully maintained. Media library applications usually use this method. Embedding the information in the very file of the resource gets rid of the link problem, but introduces information duplicity. Memory consumption can become a problem for large resource collections (which the web literally demands) or under naive attempt to use other resources as metadata itself, e.g. album cover of audio files, which is flawed by design (Resources themselves should not be used as metadata, there is usually a relationship between them). Plus, consistency of the data has to be maintained and updating metadata becomes a complex task, requiring the update of all relevant resources. Approach leveraging both distributed and centralized storage can use a unique per-file metadata key identifying the resource and serving as the key in metadata database.

An image is an artifact, which provides visual depiction of an object. In our work, we will stay in the domain of two-dimensional images, more specifically digital images. To disambiguate the term properly, by referring to an image, we will strictly refer to a two-dimensional graphics). A content of a digital image file is represented by matrix of individual pixels, each having its three colors represented by numbers and eventually bytes in the file. The image file also contains a header, where various descriptive metadata can be located.[10] This most often involves technical information about the image and resource (e.g. color depth, resolution, etc.), enriched by the information put in by the device or software that created the resource (e.g. name and model of the device or various configurations of the device at the moment of creation), but also more content-oriented data, such as title, description, artist or rating, as well as custom tags. Obtaining and using descriptive metadata of technical nature is certainly useful for image categorization, search and filtering, but does not present a problem requiring a GWAP. On the other hand, annotation/tag metadata acquisition constitutes a viable problem. However, per-image annotation does not provide nearly as much value for us as putting this information in context of the domain dataset.

Aside from explicit metadata, there are implicit data retrievable by analysis of the bytes representing the image. For example color channels, color histogram, brightness, saturation and other characteristics depending on the color model we are interested in. This can be accomplished very easily (and fast) with simple statistical and mathematical algorithms. A more computational-heavy algorithms are already capable of tracing lines and objects or recognizing faces. As a result, categorizing images based on their color characteristics (or even simple shapes) does not require us to

[7] www.flickr.com
[8] www.delicious.com
[9] www.last.fm
[10] Exchangeable image file format (EXIF) standard is quite common. www.cipa.jp/std/documents/e/DC-008-2012_E.pdf

rely on complicated metadata acquisition methods and utilizing crowdsourcing would be a waste of effort. Among the most approaches to analyzing image content are neural networks and artificial intelligence. With these it is even possible to recognize objects, scenes and context, but this requires a lot of computational power and the technology is not mature, nor widely available. Nevertheless, in the last years (2010-2014), there has been a tremendous algorithmic progress in the field of object recognition (2). Unfortunately object recognition is only a part of the problem of semantic data acquisitions, which poses a much more difficult challenge.

2.2 Ways to acquire metadata

An effort to acquire (in more narrow sense generate) metadata is manual or automated. We can arrange the metadata acquisition approaches into several categories:

Expert driven are manual approaches relying on the knowledge of domain experts, who annotate existing resources or in some other way generate or put to use metadata (e.g. by creating ontologies). Experts are capable of generating very specific and detailed, thus more valuable data. For example an image annotation by an expert can capture the content more concisely because the expert understands the underlying features of the content, that separate it from other content. The number of experts can be as little as 1. The expert driven approach can produce very high result, but is generally very expensive as well. Because it is impossible to cover the very large amount of resources, this approach is employed for specific domains, small datasets or in situations requiring highly correct data. It is useful to use this method to generate an initial sample for data validation or building top layers of ontologies (3).

Crowd driven (crowdsourcing) are manual approaches harnessing the knowledge of the "crowd" of people. Many individuals contribute with effort to common goal, which does not have to be same as the motivation compelling them to work. Crowdsourcing often produces the metadata as a byproduct of other activity. The provided result varies depending on the specifics of the domain. Because this approach leverages the ability of a human brain, it can be very effective. However, the generality of the harnessed knowledge leads to unsatisfactory results for specific domains, where experts' knowledge is required. This method is unavoidably prone to mistakes and incorrect results. Therefore, further validation of the partial results is required. These can be cross validated with results of other individuals (multiple user agreement principle (3)), experts or facts known to be correct. The correctness often remains statistically approximated. Because the crowdsourcing stands and falls on the number of participants, the motivation to participate in the activity is particularly important. We recognize motivation through personal interest or reward. Games with a purpose, which also utilize crowdsourcing, try to motivate participation through motivation to play games (which themselves provide reward in form of gratification).

Machine driven are automated approaches relying on the computational power of the machines to handle algorithmic solutions. The machines use various forms of data mining or content recognition techniques. For machines, heterogeneous information spaces pose a problem, although it can be overcome. Machine learning or artificial intelligence often provides a leading edge in the area. Automatic approaches, like crowdsourcing, are prone to incorrect results and need to be supervised. They can be efficient for small datasets, and concrete domains. However, machine driven approaches are rapidly evolving, and with the growing availability of computing power, they are capable of solving increasingly complex tasks.

2.3 Games with a purpose

Games with a purpose (GWAPs), also called human computation games (HCG), denote game applications which focus on solving computationally difficult problems exploiting the knowledge of the player base. It is a form of human-based computation, specifically crowdsourcing, where the crowd (the players in this case) is motivated to participate on solving the problem by playing the game.

GWAPs are based on the idea of different computational cost between humans and computers. They try to minimize the computing costs needed to solve a task, or solve a task impossible to solve with computer (AI-complete [11]). In human-based computation model, it is the computer that assigns the task to a human (usually quickly solvable). It then collects the results from many such tasks and transforms them into solution of the original problem. GWAPs employ human-based computation model while attempting for gamification[12] of the human interacted elements. GWAPs are capable of delivering high quality of data. Some of the domains where they prove to be cost effective solutions are: semantics generation, human language processing, object recognition or scientific problems.

There are several problems that concern GWAPs. GWAP must be able to cater to sufficiently large crowd of people by providing effective motivators to engage the people in the game. The game must effectively incorporate human interaction into the game mechanics such that their effort is properly utilized to solve the problem, without sacrificing the entertainment factors. It must be able to correctly validate activity of the players and prevent malicious behavior. GWAPs should be able to discern quality of the human input and leverage the quality of the expert knowledge, even in a large crowd pools. And even well-designed game can still fail to reach its potential if it doesn't gain enough popularity and recognition. It can be seen that there are many obstacles to employing GWAP concept successfully. In general, there is not yet known methodology for designing games with a purpose and designers must often face these problems on their own. Fortunately, designers can borrow many of the existing and well known rules and strategies for creating real games. (4)

Increasing the player's satisfaction requires us first identify the design aspects responsible and understanding the reasons why people play the games in the first place (3).

- Challenge. The games have always been about challenging oneself. The player can challenge an increasingly difficult task, himself (self-improvement and overcoming own result) or other people's achievements.
- Competition. A form of challenge. Player can compete with other players ether simultaneously by (multiplayer games) or by comparing their measure of success (score), leading to the well-known concept of leader boards. Attempting to enlist or top the leaderboard can prove to be a worthy challenge for many players.
- Self-improvement. Aligned with self-challenge. The player is likely to engage in a game that will improve and hone his ability at certain tasks (likely to be useful in the context of other games). The positive effects of games on human cognition are well known and there is group of players who will play games in order to learn, expand their knowledge and improve their ability at tasks.

[11] Group of difficult problems, solving which requires making computers as intelligent as humans – creating a true artificial intelligence

[12] Gamification is the introduction of game-like elements into the design, mostly to increase the entertainment factor

- Aesthetic experience. Most of the games employ elements of art, aesthetics and various concepts borrowed from many fields. There are players who like to immerse themselves in virtual worlds, endeavors and discovery of new concepts and information.
- Entertainment. Most of the players play games purely for immediate gratifications as a form of relaxation. The immersion of the games and concentration required makes people "forget" or "escape" from the external reality and provide relaxing experience, perhaps due to flow[13] (5).
- Social aspects. The games often indirectly (singleplayer games) or directly (multiplayer games) provide a form of socialization and player interaction and this aspect is gaining even more importance in the recent years.

Player's motivations like competition, challenge or social participation are better understood and more easily leveraged in game design, while notion of entertainment and discovery is often subjective and difficult to grasp. An example of a framework dealing with the issue of aesthetic experience is MDA (Mechanics-Dynamics-Aesthetic) (6). In conclusion, designing a successful game is a difficult task involving taking different perspectives in order to evaluate and plan the various aspects of the game contributing to player's drive to play. This is even more important in regards to GWAP design, which carries the additional complexity and limitations posed by the game's purpose - the inclusion of problem solving in the game mechanics. Because of this, many GWAPs remain very simple games. However simplicity of design does not necessarily equate a disadvantage. Despite, it is often the case that GWAPs end up less successful than normal games do in purely entertainment value. Good design can overcome this. In the end game design remains a difficult task, for which no known correct methodology exists.

2.4 Multiplayer GWAPs

One of the additional problems of GWAP design is the validation of player's actions and making sure it produces correct data. Game mechanics must reward players' actions and the rewards must be properly aligned with game's purpose, in order to guide players into actions that produce the answers to the problem the game is trying to solve. The problem is that if the rewards are based on already gathered data, then the game inhibits players from arriving at different answers that the ones already established. The less data there is, the more problematic this effect becomes. Ultimately, this results in the so called cold start – where absence of any data makes it impossible to arrive at any sensible reward strategy.

One way to avoid cold start problem is to produce a minimal amount of data by other means, for example by expert driven approach. An expert would manually derive limited amount of correct results, which would be then used initially to jumpstart the game validation. The best way to tackle this problem is however, to not rely on any data whatsoever for player rewarding. This can be accomplished by using a mutual output agreement model, where multiple players playing simultaneously, but independently from each other, validate their results by comparison. This has implications for player reward, discovering extreme and unwanted values, noise, malicious behavior and cheating etc. This is why many GWAPs are multiplayer games, despite the added development complexity. However there are other advantages to using multiplayer mode, for example making the game more engaging or lacking sufficient algorithm for computer to play instead of real person. Some

[13] Flow (also known as zone) in psychology refers to the mental state in which person performing an activity is immersed and focused to the point of complete absorption, which results in enjoyment and cognitive boost

games provide single player modes by having the computer simulate player behavior using the gathered game data or recorded actions of a player in the past.

Another notable problem of GWAP design is malicious player behavior and cheating, which not only breaks the fun of the game (for the other party, if there is one), but also corrupts data acquisition. TO solve this problem, developers can make as much game logic as possible run on the game server and making the game client as thin[14] as possible leaving the client only with the purpose of user interaction and networking. This type of server, which takes control of the logic is called authoritative server. It separates what the player wants to do and what happens. Authoritative game server can be thought of as a server which runs all of the game simulation and procession of all the user inputs. On the other hand, if the server fulfills more of a synchronization role, it is called non-authoritative. Authoritative server comes with its own disadvantages. It is more difficult to develop (due to the actions being asynchronous), has a processing performance overhead and produces time delay because all user actions must be sent to the server and wait for reply. This poses a problem for real-time multiplayer games, which must employ prediction algorithms[15] to eliminate the lag (7).

2.5 Conclusion

Metadata acquisition in the image domain remains an open problem that can not be sufficiently addressed by automated approaches. Not only is the object recognition still in its infancy, it poses only part of the problem, which extends to complex scene recognition and understanding of the context of the image content. The number of resources in the domain is also much larger than in case of audio or video, which renders manual expert approaches all the more inefficient when compared to crowdsourcing. Therefore, semantic analysis of an image content, particularly in context of another image or image domain is sufficiently complex task for us to use as a basis for metadata acquisition with GWAP and can lead to results, hard to obtain by different methods of metadata acquisition.

[14] Thin client, also called a lean, zero or slim client, is an application characterized by relying on server to fulfill its role and can be thought of as a terminal.

[15] Prediction algorithm run on the client and is responsible to predict and perform a course of actions before server reply or update arrives. There is an additional problem of merging the predicted state with the game update in the end. Prediction algorithm can for example make player movement more natural.

3 Existing games in the domain

Human computation games still remain relatively young phenomenon (about 10 years old), but various attempts has been made to harness its potential. Interestingly, the games were immediately used in multimedia domain, but their coverage expanded since, finding use not only as simple content annotators, but also in web semantics, building ontologies and knowledge base. Our work focuses on the image domain and we present several well-known and relevant existing solutions.

ESP game

ESP Game (8) is one of the first and most successful GWAPs. Conceived by Luis von Ahn[16], it produces annotation for images. It is multiplayer game, where two players attempt to describe the image by associating it to words (tags) within the time limit. Players score when they use the same word. The game was later adopted by Google and incorporated into Google Image Search feature under the label of Google Image Labeler, with the aim to improve its image search. The game faced a serious abuse and counter-measures had to be taken (some words have been black listed). The game was deployed in the years 2006 – 2011 and produced good results. Its success resulted in a wider GWAP recognition and spawned many derivatives.

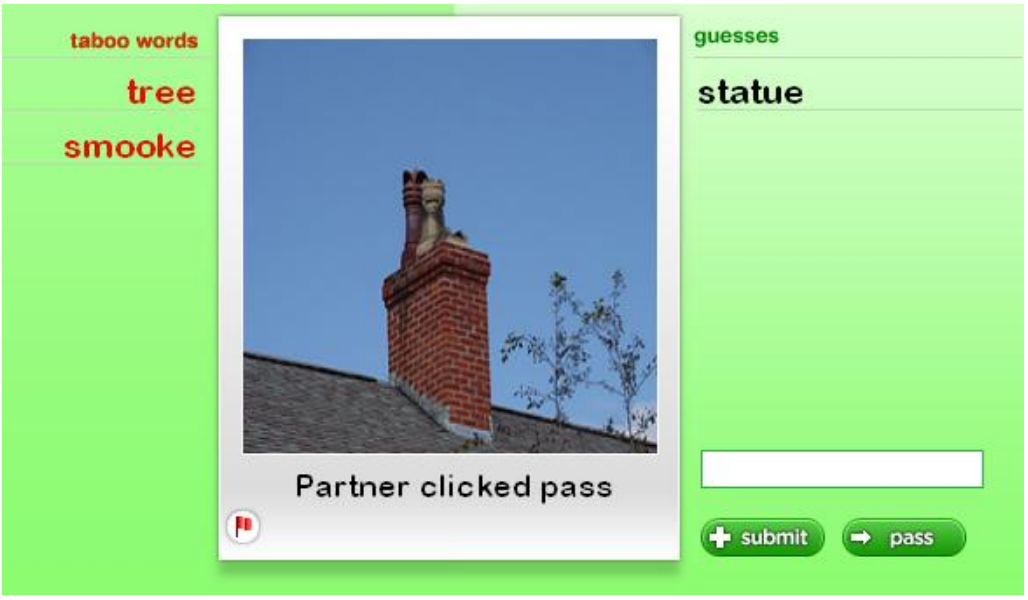

Figure 1: The looks of ESP Game

Peekaboom

Peekaboom (9), another game by Luis von Ahn and considered an extension of ESP Game, also has players evaluate the content of the images. Where ESP Game has players tell what object is on the picture, Peekaboom aims to tell where it is. Players of Peekaboom take on asymmetric roles (unlike ESP Game) - Peek player starts with black screen, while Boom with an image. Boom's role is to successfully communicate the content of the image to the Peek by gradually revealing the content of the image to him. Boom chooses the position of circle area that is revealed. In return Peek tries to find out what is on the picture by providing tags back to Boom, who validates them. The advantage of this

[16] Pioneer in crowdsourcing, first to use the term "human computation", author of ESP Game (among others) and reCAPTCHA (CAPTCHA that helps digitize books)

mechanics is that position of the most prominent object on the picture is revealed. The game produced good results.

Figure 2: Both screens of the Peekaboom game

Curator (10) is a two player output agreement online game and a prototype collection game. Randomly paired users are shown two sets of items and asked to group them into collections. Players are rewarded for matched collections. At the end of the game players see which collections matched and each other's choices. This helps players understand each other's strategy without direct communication and helps them form their strategy in the next game, because each player must act not only based on his own but also the other player's preference.

Figure 3: Curator game

SeaFish

SeaFish (11) is a single player game for image linking. The game is centered about one particular concept taken from DBpedia, represented by an image. Player see this image and also several floating images - a result of a query for this concept. Player's task is to fish the images and put them into baskets *related* and *unrelated*. Players are rewarded for choices aligned with other player's choices in the past. The annotation output of the game is exported as RDF triples in form <image> <http://xmlns.com/foaf/spec/depiction> <concept>, which follows Linked Open Data principles[17] and can contribute into Linked Open Data cloud[18].

Figure 4: SeaFish game

[17] www.w3.org/DesignIssues/LinkedData.html
[18] www.linkeddata.org

4 Game for semantic image metadata acquisition

Our goal is to create, implement and test a game with a purpose, capable of identifying semantic relation between multimedia resources. Furthermore, we would like to identify the nature of these relations. The purpose of the game should not be required to be disclosed to the players. The game should boast a satisfactory entertainment value.

The initial concept had player evaluate relative characteristics of various multimedia (either audio, video or image) displayed on the screen. The game would allow player to interact with the screen consequently leading to series of simple decisions on player's part. The decisions would be of comparative nature. This has several advantages in terms of game design. Rather than force player to evaluate the entirety of multimedia at once, the game would allow the player to do this naturally in chunks. This results in better workload distribution as well as smoother and less tiring gameplay. At the same time, player is not asked to provide absolute values at any point. Even if this complicates the evaluation of the output in absolute values, e.g. the position of the image on pretty/ugly scale, this eliminates player evaluation errors. Sorting the set of images by characteristic leveraging cumulative comparisons reaches better accuracy than sorting the images using values obtained individually.

The nature of multimedia has been narrowed down to image domain. The domain has been selected after decision process that shaped the primordial design of the game and image domain was most compatible. Because the cumulative decision process requires series of decisions instead of one, it would be inconvenient to require the player to hear an audio track or watch a video repeatedly each time it is evaluated against a different one. Images can however be comprehended very quickly compared to audio or video. This allows us to scale the amount of comparisons in single game very high, which is very important.

To have gameplay reach positive entertainment level, we decided to design the game as close to classic games (games with no purpose, i.e. pure games) as possible. The aspect of data acquisition should not severely impact gameplay and inhibit entertainment factors. Therefore, we tried to come up with a rough game design first and then insert the purpose. The idea of a board game for image manipulation came naturally, building on the success of simple games like Concentration. Then, the goal became to enrich the game with elements for semantic data acquisition noninvasively, as a natural part of the game. We tried to exclude elements such as manual annotation, writing tags or other task that are repetitive or lasting. Game tasks that don't directly contribute to either competitive or collaborative aspects of the game end up uninteresting and boring (12).

4.1 Specification

The presented game is a 2D board human computation game. Player's role is spatial image manipulation within the provided board. The board consists of fixed rectangular tiles which identify possible image positions. The default board size spans 9 tiles in a shape of rectangular 3x3 matrix. The alteration of matrix dimensions as well as shape of the tiles (to create different number of neighboring images) is realizable, but the overall idea remains the same. The effects of such alterations will be discussed later.

The game starts by the player being shown the board with image tiles and being asked to reposition them (by swapping each other) as he sees fit. Bearing in mind the presence of another player with the same initial board and same goal, the players are rewarded equally, for the amount of matches between their boards. This marks the collaborative nature of the game (as opposed to competitive

type) because the players must collaborate towards shared goal, not compete. The players are regularly notified of this in short intervals and are cumulatively assigned point for the period. The notification period should be as low as possible (as to not disrupt the player's process of thinking), in order to introduce the concept of urgency which speeds up the player's time perception and increases his engagement with the game. We contemplate about game version that does not provide any hints as to what exactly should the player be doing and forces him to figure it by himself, from the periodic point rewards. Because there is only single action the player can do (swapping images), he should be able to figure out what the game expects him to do. Note that a match does not necessarily have to be a complete match, it could be a fuzzy[19] value, e.g. computed as Manhattan's distance[20].

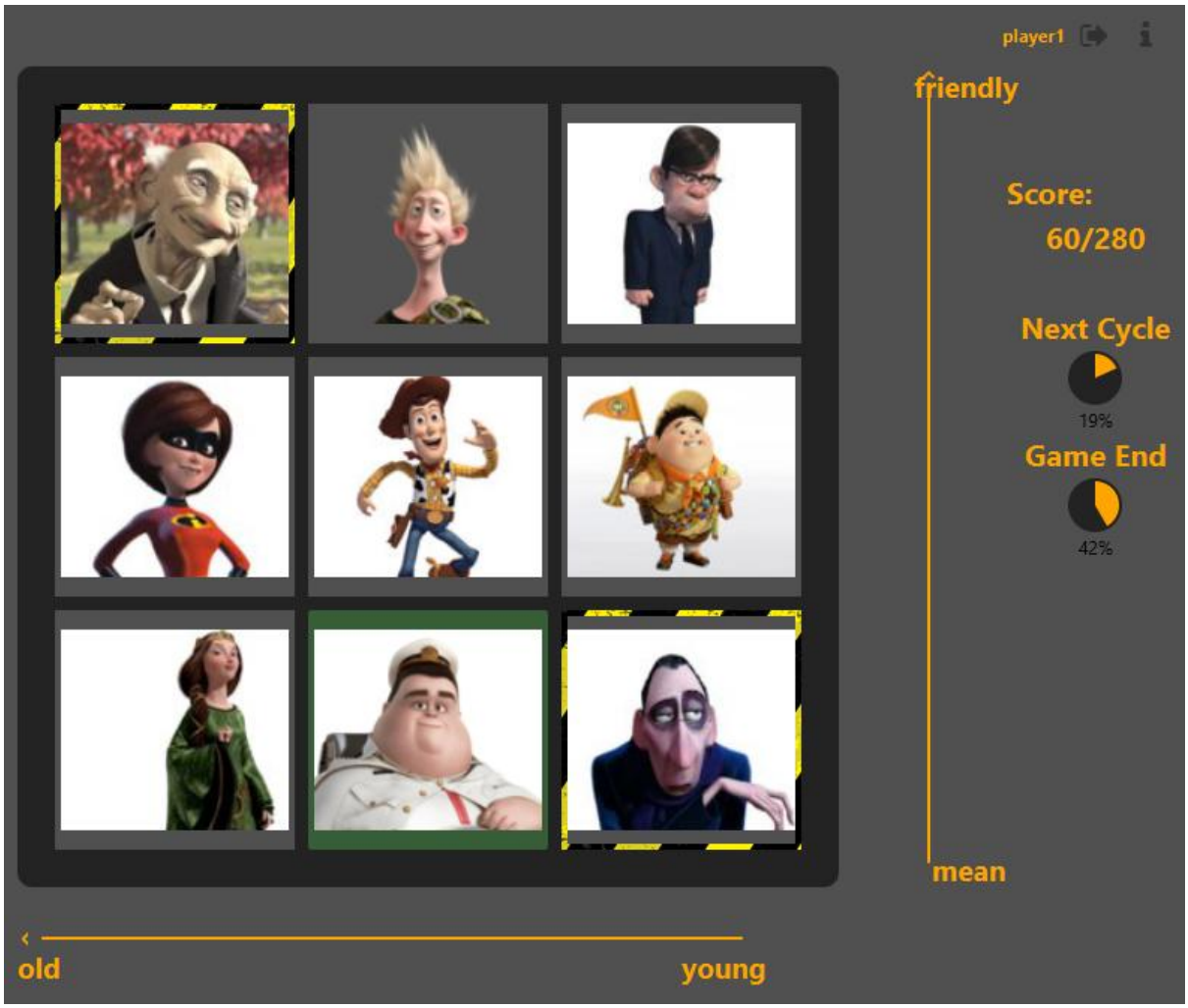

Figure 5: Screenshot of the game

The purpose of the player is to realize the greatest point reward by creating as many matches as possible with the other player. This is conceptually the same as a simple image tagging GWAP. However, the player is being guided thorough the game by the points he is being assigned. This creates a sense of dynamics as the board state is expected to change fast, as well as sense of interaction with the other player. The player is required to be in constant awareness of the state of not only his own board, but also board of the other player. He must be trying to discover the mind of the

[19] The concept of fuzzy value is borrowed from fuzzy logic – many valued extension of formal logic where the values are any from the interval <0,1>. This allows representation of "fuzzy" statements, by assigning them probabilistic true/false values, as opposed to "crisp" 1 or 0 value.

[20] Manhattan's distance of two points (or L_1 distance) is sum of the (absolute) differences of their coordinates.

other user in order to make his own decisions, regarding the recommended position of individual pictures. He must make assumptions and apply prediction of behavior of the other player based on heuristics of his own and the other player's.

This indirectly forces the player to analyze the content of the images and discover their relation. Because, ultimately, the player does not know the heuristic pattern of the other player, he must discover and make use of the most obvious and natural semantic relation between the images. This applies to all players, who as a consequence transform the semantic relation between the images into state of the board, which converges into the state with the most precise and correct semantic information. In reality, the player is reducing the heterogeneity of the image board by ordering it semantically. All players are attempting to reach the board state of the highest semantic homogeneity, which is also the purpose of the game. Player, who wants to succeed, must employ strategies of heuristic analysis on the semantic context between the images. This shows how we managed to create a game mechanics, successfully exploiting player's work to solve unrelated problem, by transparently aligning the purpose of the game with the goal the player must be trying to achieve in order to succeed.

Multiplayer

Note, that the multiplayer aspect of the game is still important here. Although the game would work in a very similar way in single player mode (ignoring the reward point assignment), by leading 2 players to align their boards with each other, it cross-validates player' results (de facto in real-time). In a single player mode, the absence of the validation would pose problems, such as point rewards not being completely aligned with the game's purpose and corrupting results, because the rewarding would have to rely on already acquired data. In case of multiplayer, the final state of the game is not the result of single user's action, rather of a shared effort, with several important characteristics, helping us validate and evaluate the importance of the game result. Analyzing the amount of closeness between the player's final board state, enables the determination of the quality of the result. If the players managed to produce very similar boards, we can increase the weight of the found semantic interrelations of the evaluated images. This allows us to filter out the "noise" and weak results and identify expert players, whose future activity could consequently be assigned more weight as well. For this, it is imperative for the players to not communicate or know each other.

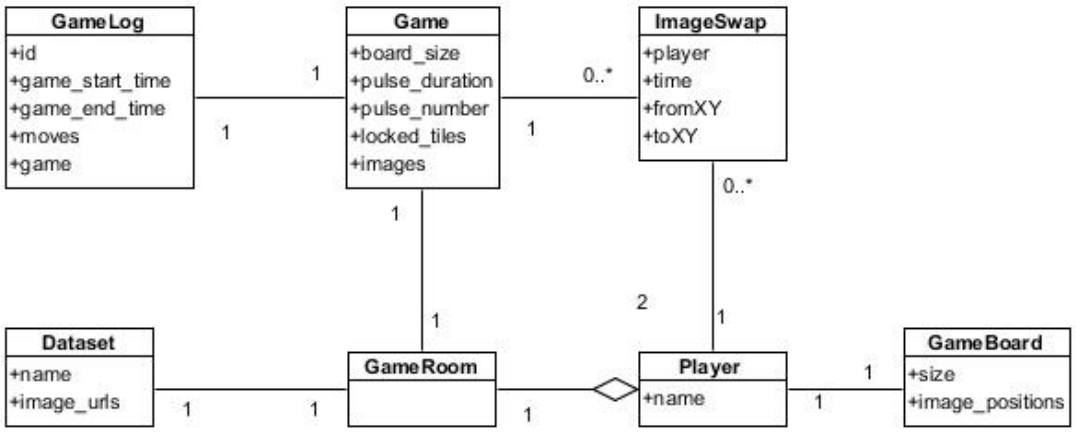

Figure 6: Logical data diagram of the game

Similarity to existing solutions

Just like ESP Game and other image tagging solution, our game also has the power to annotate images, however the number of annotations is limited by number of characteristics examined - axes used. The number of axes that can be used depends on number of games required to produce correct image evaluation and popularity of the game. But, even in the best conditions, this number remains rather limited. In addition, each axis must be manually assigned, which demands prior analysis of viability of the given characteristic within the image dataset. The strength of the game however lies in the evaluation precision, because image position reveals how much the given characteristic applies to the image and gives us fuzzy value (13). This, combined with the relative image positions, produces relative values which can be used to build more accurate semantic networks that from simple statistical analysis of tags gathered by annotation games. Even better, the effectiveness of our solution for very narrow and highly specific datasets is expected to diminish much less than in case of annotation games, which rely on absolute difference between the images. Even strong tag correlation can not capture simple but often important nuances between similar images. Our solution can, in theory, capture this more effectively.

4.2 Game mechanics

Game board consists of the tiles, uniformly spread across it. Each tile represents one position where image can be placed. The board can be represented in form of graph, where vertices represent the tiles and the edges symbolize the adjacency with other tiles. The degree of the vertex is constant and denotes the number of adjacent images. We can easily generalize the solution for any vertex degree (from 2 to n) and will refer to this number as *arity*[21] of the board (b_a). The arity of the board interests us, because it defines the number of relations per image i.e. the number of semantic data we generate from game. By increasing the arity, the game becomes increasingly complex (the amount of semantics the player has to evaluate grows).

When talking about board, we will always implicitly refer to quaternary board (4-ary, i.e. with arity of 4), unless explicitly stated otherwise. The board is also implemented as such. Quaternary board is

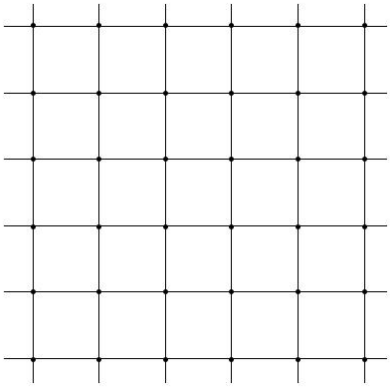

Figure 8: Graph representing board with rectangular tiles, $b_a = 4$

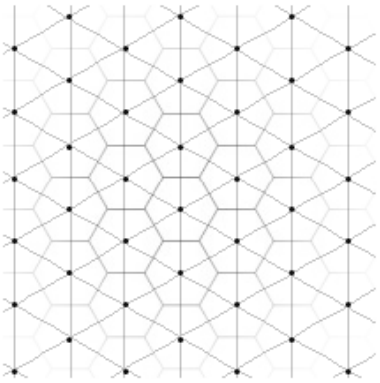

Figure 7: Graph representing board with hexagonal tiles, $b_a = 6$

characterized by rectangular tiles. It is generally common type of board in 2D games and intuitive for new players.

[21] Also called rank (mathematics), adicity, degree or valency (linguistics)

Game pulse is the game loop. The game is reevaluated periodically at specified intervals. Each pulse has the same duration. Each pulse is equal, and the same set of action on the server is executed – game state (state of all players' boards) is evaluated and user feedback is generated.

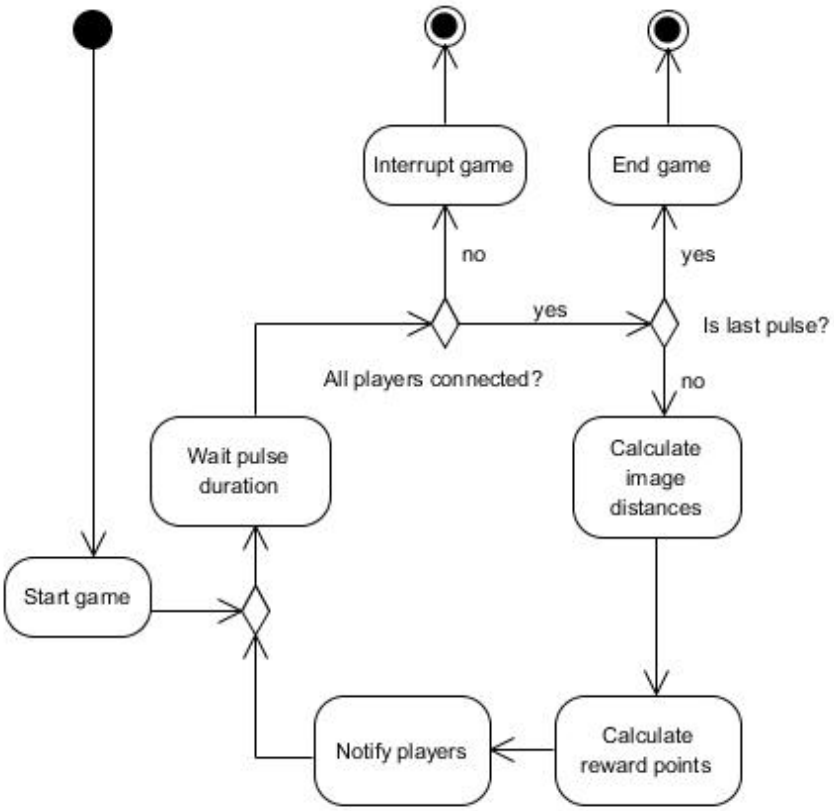

Figure 9: Activity diagram of game server logic for game pulses

First pulse is actually ignored to give players time to analyze the board at the beginning of the game. This makes first pulse twice as long as normal pulse.

The game signals **time progress**. During the game, the players are constantly notified of the time. There are separate progress indicators for game pulse and total game progress. Game pulse indicator moves very fast (in accordance with game pulse duration) in order to give a sense of fast game pace. The game progress indicator simply notifies how far in the game the player is, leaving the player more in control.

Image distance (d_i) refers to distance between the positions of the image on boards of the players. If players place the image at the same position, the distance is 0. The maximum distance depends on the method of calculation, but ultimately it is transformed to 1 to produce distance values independent of implementation[22]. First x distance (d_x) and y distance (d_y) are calculated. These refer to horizontal and vertical dimensions of a distance vector.

1. Manhattan distance = $d_x + d_y$
2. Diagonal distance = $\max(d_x,d_y)$
3. Euclidean distance = $\text{sqrt}(d_x^2,d_y^2)$

[22] This allows modular approach where distance calculation module can be easily changed without affecting the functionality of the program.

Although Euclidean comes out as more accurate, we chose to use Manhattan distance calculation for its simplicity and because it gives decimal numbers (before the transformation).

Axes are an important game element. Along the board, there is horizontal and vertical axis. The axes are static (non-interactive) and represent certain (hand-picked) characteristics of the image content within the dataset. Each axis is directed and has annotated minimum and maximum. The labels are usually opposites, e.g. pretty-ugly. The axes fulfill two roles. First and foremost, they make it possible to evaluate image characteristics, besides relative image similarity. Second, they are an important hint for the player to form an image placing strategy. The use of axes became apparent very early when people reported being confused as to what strategy they should form and what basis they should compare the images on. Relaying this information to the player is an issue we noted as perhaps the most important to the design of this game. Thus the axes became very natural part of the game.

It is important to realize that using the axes completely change the perspective on the game. Because the game requires the players to refrain from communicating between each other, it is very likely their image analysis will be concentrated on very different aspects of the content, even when using very good dataset. With the axes, the content analysis problem is mitigated. Player would gradually realize that the problem of placing the images semantically is reduced to placing them correctly within a 2D gradient defined by the axes.

Another element is **0-distance hint**. Each cycle evaluates image distance for each image and displays position hint to every player. 0-distance hint is a green glow around the image and it signals that image distance for this image is 0 – the image positions match. This signals the player what images are placed identically and allow player to concentrate on the remaining images. A generalization of this hint that shows for non-0 image distance is called **position hint (**or **image distance hint).**

The problem with this mechanism is that it can lead to abuse. There are several ways to mitigate that:

1. Game pulse duration is significantly large.
2. Game pulse number is not significantly large.
3. There is sufficient number of images/combinations.

Using this combination, players do not get the chance to wait and try reposition the remaining images. No matter how fast the player tries another combination, only one combination will be tested at the end of the cycle and if there is a lot of combinations to try, the technique won't be effective. In conclusion, if the game is quick and complex enough, there is no room for exploitation here.

4. Position hint is displayed for images with distance 0 or 1

We avoided this strategy because it would cause problems at the end of the game. It would be very difficult to achieve identical boards, unless we differentiated the different value, e.g by color.

Point hint is another helping element. Point hint is a number of points players would get if the game ended this cycle. Each cycle point hint is displayed. Because points are calculated solely from image distance, the point hint essentially tells the player how much similar their boards are, without giving away the locations of the images of the other player. Point hint has no effect on the point result of the game and servers only an informational value.

Reward points are points rewarding the player at the end of the game. The number is calculated as cumulative image distance of every image in the game. The amount of points per image is inversely

proportional to the image distance, however the exact function does not have to be linear. We used a simple implementation:

- 1 point if image distance = 0
- 0.5 point if 1
- 0 otherwise.

A continuous function could be used as well, for example:

- 1 - distance
- $(1 - \text{distance})^n$

From player's perspective, the goal is to score the most reward points at the end of the game. The players only need to pay attention to the state of the board in the last cycle and how to reach it. Initially, the rewards points gradually accumulated per each cycle. We decided against this method because it was practically impossible to get maximum points (since it is impossible to agree on the position right from the beginning), which does not pose a good reward mechanism. Even worse, this can serve as an inhibitor, where player will take caution against further image swapping if the current point hint is large enough. Making only the last score relevant promotes removes this problem and improves player focus during the game.

4.3 Game parameters

- Board size (b_s)

Board size is the length of the board in tiles. The board is always rectangular (N x N). The size can range from 2 to 6 tiles. Increasing board size increases the complexity of the game exponentially. The most natural size is 3, producing 9 tile board, following 4 for more experienced players. Bigger boards tend to be confusing and become overwhelming. However, for long session games relatively large boards could be used, e.g. 10 or even more. Because our game is focused on gathering data as quickly as possible, we prefer short game sessions and thus small boards.

- Board arity (b_a)

Different values require different implementations of core game parts, thus the parameter was decided early on to be 4.

- Locked tiles

In order to supply users with some common ground, some tiles can be locked at specified position from the start of the game. Locked image can not be manipulated and all players agree on its position. Locked images are not evaluated in the session, nor attribute to point reward. Initially, the central image was to be locked, to further hint to the players that they are manipulating the immediate surroundings of the image. Unfortunately this severely restrains the freedom of image manipulation, particularly in 3x3 game. Therefore we decided to lock either 1 or 2 images in the diagonally opposite corners. Hypothetically, any number of images can be locked (e.g. the board could be generated randomly). This carries potential for increasing game replayability.

- Empty tiles

Some image tiles can contain no image. This gives more freedom to manipulate the images, as well as room to create 'aggregates'. Using empty tiles increases the average distance between the images, which in effect amplifies the results and removes noise data. On the other hand, this introduction decreases the complexity of the game. We have noted players wanting to discard empty tiles in 3x3

games. The concept of empty tiles has more prospect for larger boards. It is possible to produce various board "shapes". By combining them with locked tiles, it is possible to provide various "levels" or "maps". This carries potential for increasing game replayability. Exploring this further did not become part of this work.

- Pulse duration (p_d), pulse number (p_n), game length (g_d)

The pulse duration can be set to arbitrary duration. After initial duration of 5 seconds, we increased this parameter to 6 seconds. The number should slightly correlate with board size, because swap operations in larger boards require more time to analyze and carry out. Pulse number represents number of pulses per game session. Normally, this simply determines length of the game, but this number should carefully reflect board size, rather than being set arbitrarily. Larger boards require much more swap operations to attain final state. For games with board size b_s = 3, we set pulse number to 10 and for b_s=4 15.

- Number of players

Because the game is multiplayer by design, the minimal number of players is 2. The game is designed with 2 players per instance in mind, although the number does not have upper limit. Having more than 2 players play a single game would strengthen the aspect of cross validation. Real time hints in regards to player's board similarity with other player's boards could also be tailored per player to hint at specific player (e.g. using different color). Requiring many players to play a single game however limits the number of games played and game playability in general. Calculating results and handling the data would also become increasingly difficult.

4.4 Game characteristics

- Game length/duration (D)

Game duration is duration of single game session from beginning to end. Game duration equals number of pulses times pulse duration. Using our selected values for b_s = 3 and 4 puts the game length to 60s and 90s respectively, using the formula:

$$D = p_d * p_n$$

In conclusion, because both pulse number and pulse duration depend on board size, game length (depending solely on these two parameters) also depends on the board size.

- Pace

The pace of the game is determined by pulse duration. We aimed to produce a fast pace game. Shorter games can lead to more games played during the period. Fast pace also produces sense of urgency. This is important for better immersion and game experience and also because it manipulates the player into taking more greedy approach. If the player is hurried to take action, his actions will tend to be more intuitive and thus more reflective as the first actions are generally the ones most obvious to take, something a slow player could actually miss out on.

- Number of tiles (T)

Number of image positions. For standard board (rectangular, b_a=4):

$$T = b_s^2$$

- Number of images (I)

Number of images equals number of tiles minus number of empty tiles.

$$I = T - ET = b_s^2 - ET$$

- Game complexity (C)

Game complexity represents the difficulty in capturing, analyzing and understanding the game board state in context of achieving the goal, i.e. the semantic complexity of the board. It can be estimated using several methods. It clearly depends on board size and board arity as well as other parameters, such as number of locked tiles, their positions, empty tiles, axes, etc. On the other hand the nature of the images, player's familiarity with their content and its context (likely depending on dataset used) as well as his experience with the game also come into play.

In a simple model, the complexity roughly equals the amount of semantic information player has to take be aware of.

This can be expressed as number of all possible comparisons between images on a board:

$$C_1 = I! = (b_s^2 - ET)!$$

But also (less correctly) as a number of connections on a single board:

$$C_2 = I * b_a$$

- Game difficulty (D)

Game difficulty stems directly from game complexity, but there are additional factors at play, such as game duration. Very short game would prove very difficult, while overly prolonged game would allow players to eventually agree on identical positions for all images. Additionally, because each game pulse provides score and visual aid for images with distance 0, the pulse length also has effect on the difficulty of the game. Difficulty can also change during the game depending on player's ability and state of the board. For instance, as the player becomes increasingly familiar with the board, his decisions become faster. In practice the number of images he needs to evaluate also decreases with time, because images placed identically or with high level of confidence can be largely ignored.

Game difficulty can be expressed as a function of familiarity with the board. Player begins with zero familiarity and finishes (ideally) with complete one. Familiarity with the images grows roughly linearly with time. Time (t) is expressed as 0-1 value in terms of progress within the game, 0 signifying the beginning and 1 the end of the game.

$$D = 1/t$$

We can approximate the difficulty in 0-1 bounds, taking accelerating change into consideration:

$$D = 1 - t^2$$

The difficulty is at the highest at the beginning of the game and then slowly recedes. The easier the game becomes, the faster the difficulty drops, eventually falling to zero at the end. The advantage of this is that the game remains relatively difficult during the majority of its run. And because the game only rarely ends with a completely identical state of the board, marginal difficulty drops are rare as well.

Alternatively we can try expressing game difficulty as a function of remaining images (I_r):

$$D = I_r$$

Factoring for complexity as a function of remaining images:

$$D = C(I_r)$$

Do note, that these completely ignore the comparative decision making aspect of the difficulty. Even if the player remembers all of the pictures, comparing the images to each other can still pose a problem.

Gameplay

From the player's point of view gameplay remains very simple. Regardless of the multitude of decisions behind, the only action player can take is to swap images of two tiles. He can repeat this

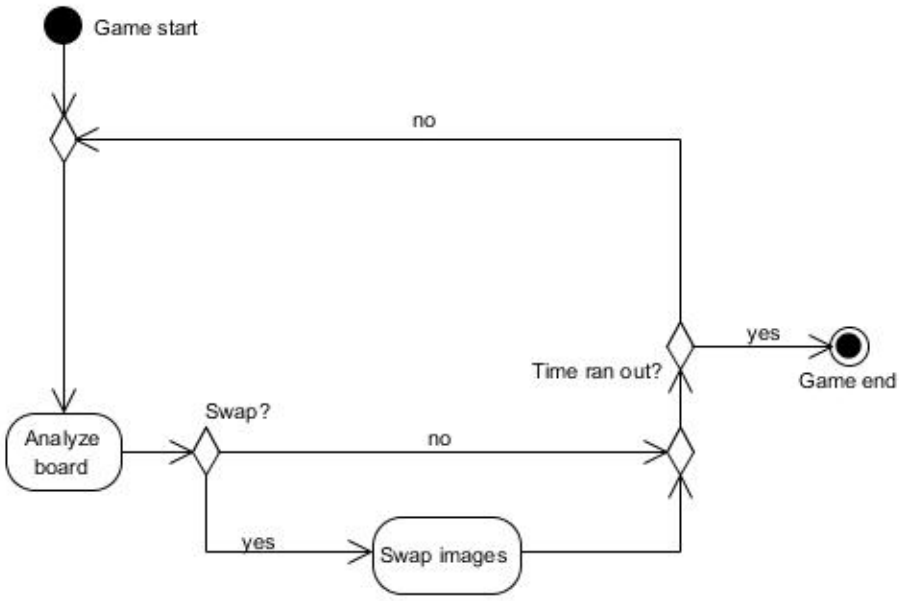

Figure 10: Activity diagram of the gameplay from player's point of view

action until the time runs out. Majority of the time is spent by analyzing the board, which maximalizes the knowledge harnessing aspect of the game.

Game purpose and its transparency

The purpose of the game is to acquire the semantic relations between images and possibly identify them and their characteristics. This is effectively equivalent to creating semantic network of the domain. As such, the game can be viewed as an expert system that infers the semantics between the resources. The only difference is, that in our case, the ability to infer and the knowledge are both harnessed from the crowd of players.

The game's design allows its purpose to be completely transparent to the player. The game itself should prove sufficiently entertaining to not require additional motivators. Additionally, the game mechanics efficiently transforms the problem the game tries to solve into game artifacts, which the player must work with or around. It is then capable of transforming the player's actions back into results for the original problem. Hence, the players do not need to be familiar with the game's purpose to motivate them to provide correct results for the game to process.

Game input

The game is applicable to any image domain, i.e. image dataset. Not only does the game's design allow evaluation of any image in the context of other images, the evaluation itself always brings an added value. However, this value can differ based on the content domain characteristics. In the context of used datasets we recognize the following possible domain strategies:

- **Single domain**. By using a single domain dataset for the game to evaluate, we can discover interrelatedness in the context of this domain. In order for players to effectively evaluate the images, the domain should not require expert knowledge. Players should be able to recognize the content and clearly identify the similarities and dissimilarities between individual elements. For example, it makes more sense to use dataset from a domain of animals then domain of dogs, because average player may find himself unable to properly differentiate between two breeds of the dog, while he might find it easy to recognize two different animal species. Generally, the domain should not be overly general, or specific. The former case suffers from too many distinctive features (thus making it hard to find similar elements), while the former has the opposite problem. Using proper domain can help players identify key features of the image content and ultimately provide better result.
- **No domain.** Effectively a single domain with no explicitly shared characteristics between the images. Such domain would suffer the problem of being too general, eliminating which would require substantial increase in the size of the dataset and even more the number of game instances to run to get satisfactory result
- **Multiple domains**. With enough evaluation, the game could be able to identify the inter-domain characteristics.

The chosen domain has also effect on player's game experience and motivation to play. The player's interest in the domain increases his motivation to play and very likely also the quality of his knowledge in the field. It would therefore be optimal to provide several datasets the player can choose from. Such move would split the amount of gathered data and was avoided for experimentation. However it remains feasible for long term experiment, given enough popularity.

Game output

The immediate (raw) game output comprises the game logs. Each finished game produces single log file. The logged information includes:

- Initial position of images of both players
- Final position of images of both players
- All image swap moves. Each includes time, player, and coordinates
- Start time and end time
- Board size and position of locked tiles, if any
- Pulse duration, pulse number

Essentially all game parameters are stored along with the process of the game and its result. The most important information remains the final state of the boards as it allows the easiest data procession and it alone contains enough semantic data to extract.

The real output of the game is provided by log analyzer application, which extracts the semantic information from the game logs. This process is carried out offline and independently of the game log accumulation stage.

4.5 Implementation architecture

The complete solution incorporates:

- game client – thin application client for players to play the game
- game server – backend server, executing the game logic
- dataset storage – storage that provides access to the images used in the game
- logging module – logging facility and storage
- log analysis software – offline log analysis tool

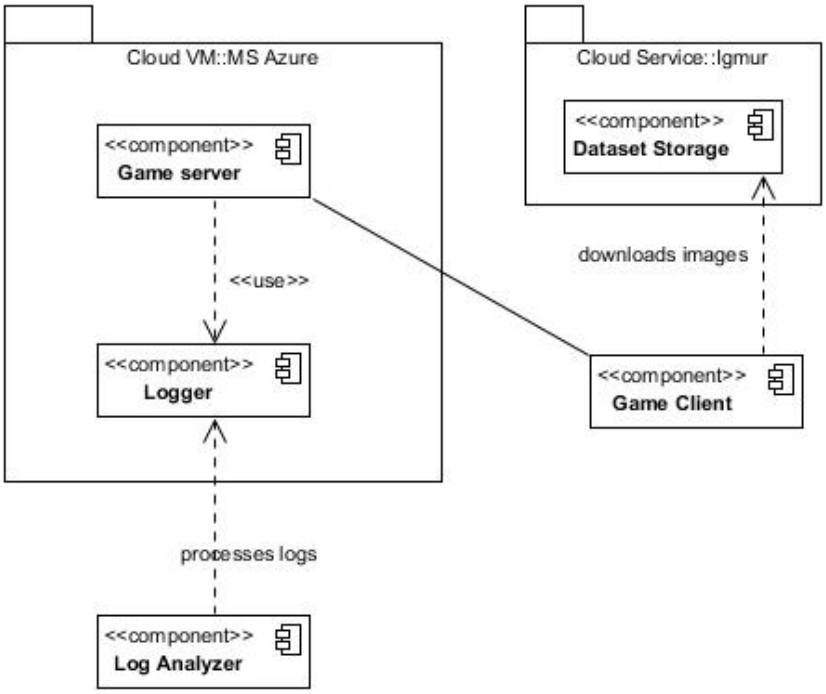

Figure 11: Component diagram of the whole solution

Game Client

The client is developed as JavaFX application (using java 8), which can be deployed as a java application, native desktop application (Mac, Windows or Linux) or webstart. The client runs locally on player's computer and connects to the server. The connection is required for being able to play (there is no single player mode). The client side implementation is not dependent on the server. Because of the authoritative server model employed to prevent malicious behavior, no game logic execute on client's side. Instead, the clients sends all requests on the server, which responds accordingly. The communication uses TCP and UDP protocols to deliver json messages used to connect users with the server, establish game sessions (called rooms) and facilitate real-time game actions.

The graphical user interface (GUI) is comprised of views – individual pages, each with different purpose, which are displayed based on the activity of the player within the application. Only one view is visible at any time. The views' visibility and transitions can be modeled using a state (**Error! Reference source not found.**). The application displays login view at the beginning, asking the player to fill in an alias (username). After connecting, lobby view is displayed, where players can create, manage and join rooms, which function as game sessions. Lobby ended up too advanced for this simple game, lead to confusion and was removed. Instead autojoin was used which is a feature

that connects the user and starts a matchmaking process. Matchmaking automatically joins user to the first available room. Available room is room which contains exactly one joined player. If there is no room available, a new one is created and player joins it. Room view displays joined users and basic game settings (disabled for experiments). When 2 players join the room, the game can start, displaying the game starting view. At this moment players are waiting while the image resources are being downloaded. Process can be cancelled in the initial stages. Once the game starts, game view is

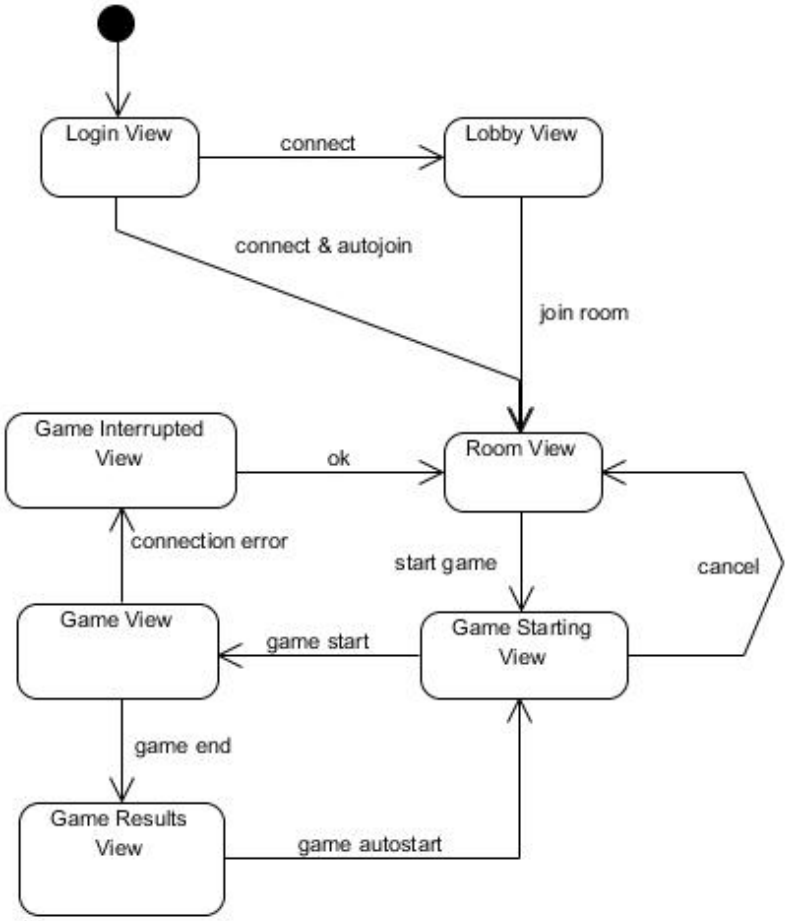

Figure 12: State diagram showcasing the separate views comprising the GUI

displayed and players can play the game. If a connection error occurs anytime throughout the game, it is interrupted and players are notified and eventually moved back to room. Upon successful game finish, another game is automatically started. In any time, the players can disconnect and close the game.

Game Server

Because AppWarp game engine is on premise[23], normally there is no requirement to build custom game server. This means a default minimalistic server provided by AppWarp could be used as a proxy to rely information between clients and any game logic would execute on the clients themselves. This however, is not compliant with authoritative server model which dictate the execution of the logic on the server. Custom server was therefore developed in Java language exploiting the features of

[23] On premise software (also on-prem software) is software that is installed and ran on the premises of the person using the software. This is different for example from an application running in the cloud. In the case of GWAP presented in this work, that means any game logic executes on the client.

AppWarp S2 API. The server is developed as standard java application with no graphical user interface. Instead of being deployed as a webserver or application server, the application is ran on a Windows 8 operating system on virtual machine[24] in Microsoft Azure[25] cloud[26]. The server is accessed through remote desktop application. The server communicates with clients through TCP and UDP endpoints[27] on port 12346.

Logging module

Logging module is part of the game server. Upon game completion a game log is created and stored in a server's file system. The logs accumulate and can at any point be analyzed. The logs and their content are not used in the game in any way. Logging is part of the server application, written in java and utilizes org.json[28] library to produce json log files.

Dataset storage

For storing the image datasets the Igmur[29] image service is being used. The service can store large number of images publically or privately. It provides constant and fast access. Upon uploading the set of images a collection of urls[30] (url per image) can be generated and then used by the server, which sends the urls to the clients when the game starts. This means the server sees no image data traffic at all, nor it contains any copy of the image files. This guarantees the absence of any potential performance bottleneck on our part.

Log analysis module

Simple java application capable of analyzing the json log files locally. The logs are first copied from server machine and then examined.

[24] Virtual machine (VM) is a software emulation of a computer that can run complete operating system.
[25] www.azure.microsoft.com – Microsoft cloud platform
[26] Cloud computing is a form of virtualized computing, which involves application systems executed within the cloud (in virtual environment) and operated through internet.
[27] Endpoint is a communication channel defined by protocol and port
[28] www.json.org
[29] www.imgur.com
[30] Uniform resource locator - a means to access a resource over computer network

5 Experiments

5.1 Playtesting

Throughout the development stage the testing was limited to usability tests of the graphical user interface and game mechanics problems. We tried to simplify game client application and make it more intuitive for use, find discovering the biggest obstacles to player experience and understand player's thinking to improve the overall design.

Usability was tested mainly to create satisfactory user interface. Testing was done by 3 different users (at different point of development) who were given tasks according to prepared scenarios and had their actions observed, followed by questions to identify the reasons behind them. The scenario tasks were:

- starting the game after the application starts
- finding out how to play the game.
- leaving active game to change the username

The testing revealed major problems for all three of them. First task, starting the game, required too many steps and revealed inadequate and confusing user interface. For this reason game lobby was completely removed (see Figure 12) and the concept of autojoin created, which makes use of simple matchmaking logic to eliminate the player's partaking in the game session creating process. Following this, each button responsible for transition between views (again, see Figure 12) is differentiated by color. Simply following the buttons without complete understanding will guide user through. Second scenario, finding out how to play the game, was important, because initially we assumed the game would be easy enough to figure out by no external knowledge provided. Instead the player should arrive at the point of the game himself. The idea proved too optimistic and we added a how to play guide accessible from the login screen as well as top right corner the whole time. On top of this, hints are shown during game loading, which also gives the player an activity while he is waiting for the game. Third task, leaving the game, was designed because it involves interrupting the game and rebuilding the game session. The most problematic action turned out to be finding the disconnect button for the first time. Interrupting the game disrupts the other player as well, which is why another scenario was added:

- recovery after game interruption

We had players play the game and then interrupted it from the other's player side, presenting the players with connection error screen. Players had no problem recovering from connection errors.

Testing of game mechanics concentrated on finding the best values of several game parameters. First game pace was being tested by observing whether players can keep up with the game pulse duration. 5 second interval turned out to overload players a little, particularly when they were new at the game so the parameter was pushed to 6 seconds (board size 3), which did not lead to any observable problems. The number of cycles (and duration of the game) was estimated based on when the players stopped providing useful information. This behavior is symptomatic of swapping images without considering their semantics, which occurs after the player puts most of the images to according to him proper position. Then he would for example try different positions multiple times hoping to match other player and score more points. The number of cycles was eventually put to 10 (board size 3). Later, during the experiment, some players gained enough experience and familiarity with the game

mechanics and dataset, that the game could be cut short of few cycles. In fact, there was a rare case of players getting maximum score (thus matching the whole board) in almost middle of the game. Players would then report dissatisfaction at having to wait for the time to run out. As for the board size, value of 3 proved to be ideal for new players, while players who completed about 10-20 games wanted to increase the size. Few games tested board size of 5 and unexpectedly, players were able to attain position matches given relatively quickly. However, there was a lot of try-and-see attempts signaling the image distance hint system is fundamentally broken. Lastly, we observed that the score per cycle supposed to hint the partial matches is largely ignored.

The players were also asked question about their overall experience, whether they had fun and to point out the parts of the game they liked the most and those they did not.

5.2 Experiment

In the scope of this work, we managed to carry out a closed experiment to test the rough validity of the game output and generate sample of semantic data.

People partaking in the experiment were explained the concept of games with the purpose and received simple instruction on the game mechanics, and were referred to the game guide for further information. They were all in their twenties and had intermediate to high experience with both computer technology and videogames. In total 6 people participated. There was no communication between the players during the game session.

Dataset used for the experiment was created manually and features 27 handpicked images featuring various cartoon characters found on Disney Wikipedia page[31]. See Figure 5 and Figure 17[32] for sample of the dataset images. We defined two observed characteristics:

- Age: represented by *young-old* axis
- Likability: represented by *mean-friendly* axis

We evaluated each characteristic for each image by single expert before the experiment began. This involved assigning a number from <0;1> range per each image-characteristic couple. The evaluation was done separately per characteristic by first ordering all images into the line by comparison and then assigning a value thought to best fit particular image.

6 players played the game in 4 sessions in the span of 4 days. Players played from 10 to 30 games each. In total N games were played. We analyzed the gamelogs with the log analyzer, which produced the following data[33]:

- mean image distance: mean(i_d)
  Average distance between the positions of the image on the boards of the players in the game. This per-image value indicates how easy it is for players to characterize the image. If the players tend to put the image on a similar position, it is likely that players are more confident in their choices and that the image content is easier to evaluate. Further, because of the axes representing the characteristics, it is possible to break the image distance vector into vertical and horizontal value and observe mean image distance per image characteristic indicating how easy it is to evaluate the given characteristic for the image.

[31] www.disney.wikia.com/wiki/Category:Pixar_characters
[32] Complete dataset can be found on the attached cd
[33] The data can be found on the attached cd

This value is calculated simply as statistical mean:

$$\text{mean id} = \frac{\text{sum}(id)}{games}$$

Where *games* is the number of the games, in which the image appeared in. There is absolute (i_{ad}) and relative (i_{rd}) image distance. Absolute i_d takes value from <0, max(i_d)>, while the relative i_d is normalized value in the range of <0,1>, and is board size agnostic.

$$i_{rd} = i_{ad}/\max(i_d)$$

Max(i_d) refers to the maximum image distance and depends on the board size and method of distance calculation. For Manhattan distance:

$$\max(i_d) = 2*(board\_size-1)$$

- image characteristic fitness: c(i)
  Per-image value of the fitness of the characteristic for the image – how much can the image be described by the characteristic. Essentially, for each observed characteristic a fuzzy set is produced, where c(i) refers to the grade of membership of the image *i* in the set[34].

- image similarity index: s(i1,i2)
  Value calculated for each image-image pair - element of a Cartesian product D x D, where D represents the input dataset. The value takes on value from range <0,1> and represents the similarity between the images. The value is calculated as mean relative distance between the two images in all games.

$$\text{s i1,i2} = \frac{\text{sum}(1 - dist(i1, i2))}{games}$$

  Where *games* is the number of the games, in which both *i1* and *i2* images appeared in, and *dist*(i1,i2) returns relative distance between the images. Once again, it is possible to break the value apart and calculate it separately per observed characteristic.

In order to evaluate game output correctness, we compared the image characteristic fitness c(i) to the manual values produced by the expert. The correlation coefficient for evaluation of age characteristic was 0.7634 and for likability was 0.7894. Although the scope of the experiment remains rather limited, the findings suggest the potential of the game in producing valuable results.

[34] A fuzzy set is a pair (U,m) where U is a set and m is a membership function m: U ->[0,1]. Grade of membership refers to the value of m(x), where x is an element from U. In this case, set U is a set of images characterized by characteristic C, i.e. our whole dataset in the context of the C.

6 Conclusions

In this work, we analyzed the problem domain of semantic image metadata acquisition and described various classes of solutions to the acquisition problem, leading us to crowdsourcing and the reasoning behind existence of games with a purpose.

We then focused on designing a game with a purpose in order to achieve proficiency in semantic data acquisition and discovery of resource similarity and interrelatedness in area of crowdsourcing. Our devised solution takes into consideration numerous game design aspects influencing the success of the game among players and as a data acquisition tool. We aimed our attention on learning from research in the domain, successful transformation and encapsulation of the purpose (semantic data acquisition) into game mechanics and validity of data generated. We also examined feasibilities of various possible implementations from technical and architectural standpoint and followed to fully implement the game.

Our devised game is collaborative player output agreement multiplayer board game for generating semantic metadata of image content. Players manipulate images on the board and indirectly insert semantic information into the game while making use of their knowledge of the context of the image content. The game monitors players' activities and stores them as data logs. We then inspect the data and extract the semantic information with our log analyzer tool.

The game is characterized by simple mechanics and conceptually builds upon the pioneer works in the field, such as Luis von Ahn's ESP game. However, compared to other games with a purpose, our work attempts to further hide away the game's purpose and create more entertaining game, similar to standard board games. Players reported enjoying the game, namely the thinking process of transforming the content of the picture into position on a 2D grid and the social aspect of trying to understand and predict other player's behavior. Compared to standard online games on the web, the our game tries to distinguish by combining the fast-paced board game mechanics with semantic manipulation of image resources, which presents an added value to the gameplay experience. It has potential to motivate crowds normally not interested in board games or games in general. With successful integration, the game could work in a real-world environment, aiming to generate practical data necessary for semantic graph generation or recommender systems, particularly in the area of specialized domains. Single player version of the game could also be used for personalized image management system leveraging faceted search and semantic similarity. The game could also be used to approximate fuzzy sets.

Our hypothesis was that it is both possible and practical to employ crowd-sourcing solutions (such as GWAPs) to generate accurate semantic information for highly specific set of image resources. We carried out simple usability tests and moved on to the experiment, where we gathered gameplay data. Small dataset of cartoon character images was created with handpicked image characteristics for the game to evaluate. Analysis of the data produced semantic information on image similarity, tag fitness of image content and its ambiguity – difficulty to evaluate. The data correlated with those created manually with an expert.

Future work involves carrying out an open experiment, which was not conducted within the scope of this work. We aim to increase the scope of the experiment and evaluate the effect of various game parameters (e.g. board size) on game results. The open experiment will have the game publically available for a prolonged time period in an attempt to gather large amount of data.

Possible game extensions

- **Premature game ending**
  If the players reach the ideal board and score maximum number of points before the time runs out, the game should end automatically, preferably with a bonus.

- **Image distance hint**
  Image distance hint (green glow fading out) was reported to be insufficient and clumsy. The effect was strengthened in contrast and duration, but it may be fundamentally broken by design, particularly for larger board size, where it leads to frequent try-and-see move attempts. Player would shuffle the board and wait for the pulse to hint him which images match their position. This behavior is unwanted, but for large boards it is inevitable because the hint is the only means of verifying the correct position. The goal of the game is for players to agree on some positions and them slowly build around them until they match the whole board – in such process the semantic information is used the whole time. If the players start shuffle the board randomly (which can not lead to sufficient point gains anyway) they will eventually find some matches, but they are worthless both for the players and from metadata acquisition standpoint.

  Thus, the hint system should be enhanced to show the image distance for each image, not just complete matches. We propose two methods to achieve this – either using the glow effect on every image and adding intensity or duration parameter to it to represent the image distance or show point gain per image (for example by having the points pop out of the image at every cycle).[35] This actually solves the problem of the cycle score being largely ignored.

- **Player login**
  This is one of the features that were omitted. Player identity is imperative for recognizing recurring players, which in turn enables drawing additional information. Besides the player statistics, it can also help identify expert players to improve data gathering efficiency. This modification requires use of player database. For player's convenience support for using existing accounts like Facebook or Google+ could be leveraged.

- **Number of players**
  The game can also be modified to allow more than 2 players per game session, leading to an increase in value of player action validation. The game client requires no modification, while the server has been created with this option in mind. The only noncompliant parts are logging and image distance calculation during game pulses. Multiple player solution requires method capable of combining image distances of each player-player element of the Cartesian product of players into single statistical value.

- **Board arity**
  Very interesting modification is the use of senary (arity of 6) board, which boasts hexagonal tiles. This requires extensive modification of both game client and game server.

- **Annotations**

[35] This attempt was partially actualized in the initial development stages, but was met with technical difficulties and abandoned due to uncertainty about its usefulness.

Very promising from data acquisition standpoint is the inclusion of annotations into the game. Players could be given limited opportunity to annotate tile with an image and notify the other players of the annotation and annotated tile position. Optionally, the notified annotation could move along if player decides to change position of the image later in the game. This feature would serve as a hint to other players to better understand each other's intentions, while producing image annotations as by-product. To eliminate the need for players to write annotations, there could be predefines set of annotations available for players to choose from. This does decrease the uniqueness value of the annotations.

- **Rich output analysis**
  One of the future works which require no modifications is more advanced analysis of the game output. Besides final board state, times of individual moves could be examined and different values could be assigned to them, eventually contributing to confidence factor per image. It stands to reason that moves at the beginning of the game and very extreme moves better reflect the semantic value of the images involved. Confidence factor could also be drawn from player's ability (estimated from his past game records), final game score (better score implies more correct results), number of moves required to reach final state of the board, number of moves required to reach final image position, etc.

- **Data visualization**
  Application could be developed that displays semantic graph of the dataset.

- **Service incorporation**
  Incorporation into an image service is one of the potential modifications as well. It would allow real life utilization of the game and possibly building publicly available semantic networks. This however requires a service compliant with the requirements of storing the generated data.

- **Dataset variety**
  The game client is completely independent of dataset used, while the game server allows defining multiple datasets. Simple modification could allow players to pick the dataset of their choice. This carries the advantage greater player customization and increased interest in the game.

7 Bibliography

1. **NISO.** *Understanding Metadata.* s.l. : NISO Press, 2004. p. 1. ISBN 1-880124-62-9.

2. **Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei.** ImageNet Large Scale Visual Recognition Challenge. 2014, pp. 20-21.

3. **Šimko, Jakub.** *Semantic Acquisition Games Harnessing the manpower for creating semantics.* Bratislava : s.n., 2013. 978-3-319-06114-6.

4. **Schell, Jesse.** *The Art of Game Design: A Book of Lenses, Second Edition.* s.l. : CRC Press, 2014. 1466598646.

5. **Mihaly Csikszentmihalyi, Isabella Selega Csikszentmihalyi.** *Optimal Experience Psychological Studies of Flow in Consciousness.* 1992. pp. 29-31. 9780521438094.

6. Player Acceptance of Human Computation Games: An Aesthetic Perspective. [book auth.] Dion Hoe-Lian Goh, Ee-Peng Lim, Adrian Wei Liang Vu Xiaohui Wang. *The Emergence of Digital Libraries - Research and Practices.* 2014.

7. High Level Networking Concepts. *docs.unity3d.com/Manual/net-HighLevelOverview.html.* [Online] Unity Technologies, 2014. [Cited: 2014.]

8. *Labeling images with a computer game.* **Luis von Ahn, Laura Dabbish.** Pittsburgh, PA, USA : s.n., 2004.

9. *Peekaboom: A Game for Locating Objects in Images.* **Luis von Ahn, Ruoran Liu and Manuel Blum.**

10. *Curator: A Game with a Purpose for Collection.* **Greg Walsh, Jennifer Golbeck.** 2010.

11. **Stefan Thaler, Katharina Siorpaes, David Mear, Elena Simperl, Carl Goodman.** SeaFish: A Game for Collaborative and Visual Image Annotation and Interlinking. *The Semanic Web: Research and Applications.* 2011, pp. 466-470.

12. **Dion Hoe-Lian Goh, Chei Sian Lee.** Understanding Playability and Motivational Needs in Human Computation Games. *Digital Libraries:For Cultural Heritage,Knowledge Dissemination,and Future Creation.* s.l. : Springer Berlin Heidelberg, 2011, pp. 234-235.

13. Applying Fuzzy DLs in the Extraction of Image Semantics. [book auth.] Ioannis Kompatsiaris, Michael G. Strintzis Stamatia Dasiopoulou. *Journal on Data Semantics XIV.* pp. 105-132.

Appendix A – Technical documentation

A.1 Implementation possibilities & picking the platform

The devised game is a multiplayer online game. For game activity processing and validation, we use an authoritative server-client model. The application will therefore consist of three parts: the server containing all the logic, the client and the networking for communication between the two. Below, we describe some of the possible development environments we came across and our evaluation in context of our requirements.

Html (Hyper Text Markup Language) is the publishing language of the World Wide Web[36]. It is the leading and most widespread language for creating websites and simple web applications[37]. It is independent of platform and designed to work with other languages (namely CSS and Javascript), which can greatly extend its capabilities. With the rise of HTML5, which adds better support for multimedia and dynamic content, it is a good candidate for creating the client side of simple games. HTML is natively supported by web browser applications and is accessible on wide range of devices. It has great penetration rates and its availability is important for crowdsourcing solutions like GWAP. The language itself does not provide any networking support, but there are several technologies available that can add such support to a html web application. In summary, html provides the necessary capabilities for simple interactive games. We chose it as the best alternative to develop client side of our game.

An alternative solution is **Rich Client Application** (RIA). RIA is a web application with characteristics of a typical desktop application. They leverage a custom run-time environment and as such are not dependent on the browser. Because of this, RIAs are mostly unsupported by web browsers natively, instead, they rely on plugins or virtual machines, which is their major strength, but simultaneously greatest drawback. The most common platforms for RIA development are Adobe Flash[38], Microsoft Silverlight[39] and Java. RIA and particularly Adobe's and Java's (developed by Sun, lately Oracle) solutions are criticized for security issues.

Adobe Flash depends on Adobe Flash Player browser plugin and is mostly used for embedding multimedia content into html pages. Despite its high penetration rate Flash is on the decline (especially because of HTML5) as it failed to capture the market of mobile devices.

Silverlight depends on the widespread .NET framework (both developed by Microsoft) and requires browser plugin to run. It is designed for creating interactive web and mobile applications. It also allows running programmed logic. Despite its high adoption rates, it is now on the decline and Microsoft even announced end of life for the version 5 in 2021.

JavaFX[40] [41] is part of core java libraries for both desktop and rich internet applications. It depends on the Java Runtime Environment to run on Java Virtual machine and requires a Java plugin to work in

[36] www.w3.org/html
[37] Web application is a term generally used for software that runs in a web browser.
[38] www.adobe.com/products/flashruntimes.html
[39] www.microsoft.com/silverlight
[40] Java is repeatedly trying to capture the RIA market. Java applets are java aplications embedded and deployed within a website. Requiring Java plugin, they were often used for interactive or 3D visualizations, that requied more high level language to implement. Once widely accepted, java applets are disappearing and even Oracle abandoned the concept. It instead turned to JavaFX Script – scripting language compiling to bytecode

browser. It borrows a lot from HTML. The graphics can be separated into XML-like markup language (thus utilizing the MVC[42]) and even be completely styled with CSS. JavaFX unifies desktop and RIA development and intends to provide full cross-platform support (including Android and iOS). It is mostly open source and is receiving a big push from community. It is ideal for GWAP development as it offers high-level language for web deployed applications. Unfortunately, the low penetration rates would play against the adoption of the game. Regardless, we seriously considered this platform against HTML5, because of prior experience with JavaFX, simplified development and native support for networking.

Elm[43] is a functional reactive language for interactive applications which compiles to HTML, CSS and Javascript. The language could satisfy the needs of GWAP development, but because of its novelty and potential problems, we decided against using it as a solution.

There are emerging frameworks for rapid web application development like **Ruby on Rails**[44] (backed by Ruby language) or **Play**[45] (backed by Java and Scala language), which also come as an alternative to developing the game. These frameworks also simplify the server deployment. Ultimately, we decided against web application frameworks, which favor convention over configuration.

Our attention also focused on proliferation of game engines and web frameworks, enabling development of multiplayer games and web applications. The term game engine can refer to a heavy-weight engine for cutting edge games (e.g. CryEngine3, Unreal Engine), but also light-weight frameworks facilitating the creation of simple HTML –based games. In the field of web game engines there are many alternative to choose from (especially web game engines)[46]. Some of the HTML5 game engines we inspected include ImpactJS[47], Isogenic Game Engine[48] and LimeJS[49].

Unity 3D[50] is a robust game engine bringing features of a fully-fledged game engine into browser and mobile games. It allows multiplatform development of wide range of games. It supports 2D and 3D games, provides high performance, server-client models and many much needed features for serious game development. However, the engine clearly provides much more than we need. It only natively suports C++ and .Net languages and mainly, it does not provide any implementation for authoritative server models. Its usefulness for us remains at the client, development of which can do away with the complexity of heavy-weight game engine.

AppWarp S2[51] is an on-premise Real-time Multiplayer game server solution. It provides Java API[52] for development of game server (and specifically authoritative model). It offers rich features like

---

(thus being interoperable with standard java code) that was intented to provide better device compatibility and easier use. JavaFX Script eventually evolved into java code. It received a major redesign and called JavaFX is now a part of JRE (Java Runtime Environment)and JDK (Java Development Kit).

[41] www.docs.oracle.com/javafx/2/overview/jfxpub-overview.htm

[42] MVC – Model View Controller is a way of developing application graphical interface, that focuses on separation of concerns, namely the Model (business logic), View (the graphics) and Controller (the behavior of the graphics)

[43] www.elm-lang.org

[44] www.rubyonrails.org

[45] www.playframework.com

[46] www.html5gameengine.com

[47] www.impactjs.com

[48] www.isogenicengine.com

[49] www.limejs.com

[50] www.unity3d.com

[51] www.appwarps2.shephertz.com

matchmaking, lobbies, rooms, authentication as well as game cloud management, leader boards, avatar management, notifications, in-game analytics, etc. It can support variety of game types, serious games (e.g. MMOs), but also very simplistic, which is what we need. AppWarp S2 is also client platform agnostic and provides multitude of SDKs[53] for concrete platforms. It comes with firm documentation and tutorials that help starting the development. The author's experience with Java language also favors this solution. This platform satisfies our requirements and we chose it for developing the server side of game.

For server deployment there is the possibility of using Java Servlet, e.g. **Tomcat**[54] or deploying the server application on a virtual machine in the cloud, either **Amazon AWS**[55] or **Microsoft Azure**. Alternative hosting for server solutions are **Heroku** and **Openshift**.

A.2 Installation

Client

Game client is portable and does not require installation. It can be run as desktop application by executing the jar. There is also index.html, opening which allows to open the client as webstart. In both cases java must be installed on the system, at least version 8. For successful client-server communication, a server IP address must first be specified in the config.txt file. It is possible to run the whole solution locally (with the exception of image dataset storage), using 127.0.0.1 localhost IP address and running multiple instances of the game client at once. It is however not possible to control those instances simultaneously. It is also possible to use a simple local area network (LAN) for this purpose.

Server

Game server is in form of standard java application, also runnable by executing the jar. This application must be executed on virtual machine or wrapped into a server container and deployed. The application must be accessible with static public IP of choice and allow TCP and UDP communication on port 12346. The server contains AppConfig.json file where these settings must be correctly set up, before the server is executed.

Logging module is part of the game server. The logs will be produced as files in the application's /gamelogs directory.

Analyzer

Log analyzer is standalone application, which requires no installation. The application is executable jar. After it is executed, it analyzes all logs in the location of the jar, produces single file called statistics.txt and exits. Before execution, copy the logs to be examined into the application directory.

A.3 Dataset preparation

Image dataset storage can be both centralized or distributed. All the game server requires is to have list of publically accessible urls for images that comprise the dataset. In the servers application directory, each datasets has its own directory: server_app_location/datasets/datasetname, where

[52] API – Application programming interface
[53] SDK – Software Development Kit
[54] www.tomcat.apache.org
[55] www.aws.amazon.com

datasetname text is arbitrary and will be used by the application. The name should reflect the nature or intention of the dataset.

Each dataset directory must contain the following files:

- axes.txt
  each line should contain a key-value pair defining minimum and maximum value of the axes as text. An example: “young : old”.
- characteristics.txt
  contains list of key-value pairs defining image characteristics defined in the axes file. Format: image_file_name : list of numbers separated by ‘,’. The numbers represent how much is the axe characteristic applicable to the image. The numbers must be in range <0;1> and their number and order must reflect the axes records in the axes file. For example if the axes contain 2 records, each line could look like this: “WDCWQWN.jpg : 0.17,0.42”
- urls.txt
  contains key-value pairs, defining the respective urls for the image in the /images directory. The format is: “image_file_name : url”, each key-value pair must be on separate line. An example: “FSrySIc.jpg : http://imgur.com/FSrySIc.jpg”. It is recommended to have the name of the image resource be the same as the last part of the url, but it is not necessary.
- images (directory)
  the directory should contain all image resources

The files axes.txt can be empty, in such case a 2 default characteristics will be created (although the output will not be relevant). Therefore the characteristics.txt file must contain at least 2 numbers per image record. For practical use, refer to the contents in the attached medium.

The recommended way to create dataset follows these steps:

- choose dataset name
- create dataset directory using the name in the /datasets directory on application server
- upload the images on an image service
- have the service generate urls for access
- download all the images into the dataset_directory/images so that the file names reflect urls
- define image characteristics as axes in the axes.txt
- evaluate the characteristics manually by expert and define them in characteristics.txt file
- put records in urls.txt

A.4 Source code

Server

Game server is structured into adaptors. Adaptor classes handle all communication and management of the games on server’s part.

ChatServerAdaptor manages the creation and deletion of zones. Zones are important for scaling the game and for distributive game models.

ZoneAdaptor manages zone. Involved are handling user connection, user authentication and room management. Only one zone is required and created by our game server.

RoomAdaptor manages the whole game session and all communication between joined players. Entirety of the server logic is aggregated at two places – the loop task which is a timer responsible for the game loop and executing the pulse logic and hangleChatRequest() method. This is method that handles all requests sent to the server. We refer to the requests as messages and they represent game events. The messages are Strings parsable from/to json objects and can be thought of as collections of key-value pairs. The networking part is handled automatically by the AppWarp library. The structure of the messages varies depending on the event. Every event message contains event value, which determines the type of event. The hierarchy of the events is below:

- ROOM
  Event that will be broadcasted to all players joined in the room. This is useful for player-to-player events.

- START_READY
  Event signifies that particular user waiting in the room is ready to participate in game.

- INIT_COUNTDOWN_READY
  Event requesting to start the game starting process, during which the clients download images and prepare the game.

- INIT_COUNTDOWN_UNREADY
  Event requesting to cancel game starting process. A result of player canceling the game before it starts.

- READY
  Event notifying the server that particular player has finished game initialization and all image resources are ready. Everytime server receives this event, it checks if all joined users are ready. The server waits for the last users to finish the initialization phase upon which a START event is broadcasted to all players and game loop is started. First pulse is not fired immediately, but after pulse duration.

- START
  Event notifying the players to start the game. Game clients display game view and the game starts. This event also carries additional information about game properties , including locked tile positions and character of the axes (their labels).

- GAME
  Event pertaining to the gameplay itself. Encompasses all actions during the game. The event contains action value which more closely specifies the type of this event. The most notable action is MOVE. It specifies an image swap operation.

Other events like joining the room, leaving the room, connecting and disconnecting, connection errors and the rest is handled in separate methods and separate adaptor classes. The structure of handling all this on client's part is roughly identical and therefore will not be further described.

PointEvaluator is responsible for calculating reward points per image. It is a function of image distance. Because point evaluation implementations is depend on the way image distance is calculated, PointEvaluator must also fulfill this function. In order to allow implementation change, PointEvaluator is an interface that defines the following methods:

```
public double eval(int distance, int board_size);   // calculates reward for given distance and board size
public int distance(Point p1, Point p2);  // calculates image distance
```

Below is the standard point evaluator we used:

```
public class SimplePointEvaluator implements PointEvaluator {
   public double eval(int distance, int size) {
      return distance==0 ? 1
                         : distance==1 ? 0.5
                                       : 0;
   }
   public int distance(P p1, P p2) {
      return abs(p1.x-p2.x)+abs(p1.y-p2.y);
   }
}
```

Client

The client is an JavaFX application[56] and follows JavaFX' MVC[57] and best practices. Therefore, we leverage two new java technologies. One is fxml[58] - a scriptable XML-based markup language for constructing Java object graphs to rapidly build some of the graphical user interface and to separate it from the game logic. The other is JavaFX CSS[59] - CSS accustomed for use with JavaFX to produce custom look and feel of the application easily and with clear separation from the game logic or graphics. All of the CSS code is contained in the Skin.css file.

A.5 Game output log specification

The output of the game are JSON files, one per finished game session. JSON files contain a JSON[60] object with the structure below (key:value) :

axexmin : String specifying minimum of the characteristic of an x axis
axexmax : String specifying minimum of the characteristic of an x axis
axeymin : String specifying minimum of the characteristic of an y axis
axeymax : String specifying minimum of the characteristic of an y axis
size: int specifying board size
image_count : int specifying number of images used,
maxcycles : int specifying number of pulses

[56] www.download.java.net/jdk8/jfxdocs/javafx/application/Application.html
[57] MVC – Model View Controller
[58] www.docs.oracle.com/javafx/2/api/javafx/fxml/doc-files/introduction_to_fxml.html
[59] www.docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html
[60] more about the JSON structure and elements at www.json.org

id : java UUID, rangom unique game identifier
timestart : long specifying the time of the game start measured in milliseconds, between the current time and midnight, January 1, 1970 UTC
timend : long specifying the time the game ended
cycle_dur : int specifying pulse duration in seconds
startboard1url : JSON array of Strings - urls of images in board1 at the beginning of the game, if the url is in format "ex", where x is any decimal number (e.g. e1 or e14), it represents an empty tile and should be ignored
startboard1pos : JSONarray of int - position indexes of the images in the startboard1url array, position index is calculated as board_size*y+x. To read the board state from the log, it is recommended to iterate both arrays and
startboard2url : see above
startboard2pos : see above
endboard1url : see above
endboard1pos : see above
endboard2url : see above
endboard2pos : see above
moves : array of JSON objects representing moves – image swaps operations
- **time** : time specifying the time of the move (same unit as timestart/timeend)
- **fromx** : int specifying x position of starting point of the move
- **fromy** : int specifying y position of starting point of the move
- **tox** : int specifying x position of ending point of the move
- **toy** : int specifying y position of ending point of the move
- **board** : int specifying board (1 or 2 in game for 2 players).[61]

The files are named in the format: "id.json", where id is equal to the value stored in the file. To analyze the output files, list all json files in the /dataset_directory/gamelogs directory. Each dataset contains its own logs, to enable using multiple datasets.

[61] Unfortunately, our experimental logs miss this information due to a bug

Appendix B - How to play

The game starts with a login screen asking user to write his alias as a player. Player can read short manual by clicking on the “How to Play” button. Once player click “Connect & Play” or presses ENTER key, he is connected to game room. Player is successfully connected if the IP server address is properly configured in the config.txt file, the username is unique (no user with the same name is connected) and does not contain any special characters.

Figure 13: Login screen

In the game room, player waits for 2nd player to join and selects checkbox to indicate he is ready. There are game settings on the right to customize the game, however these served a purpose only for the developer and are overriden by server during experiments. Once all players are ready “Start” button becomes available. Clicking on the button starts the game. The green is colored differently to indicate its importance nd to guide player intuitively. This method is used for all buttons that let user transition between views.

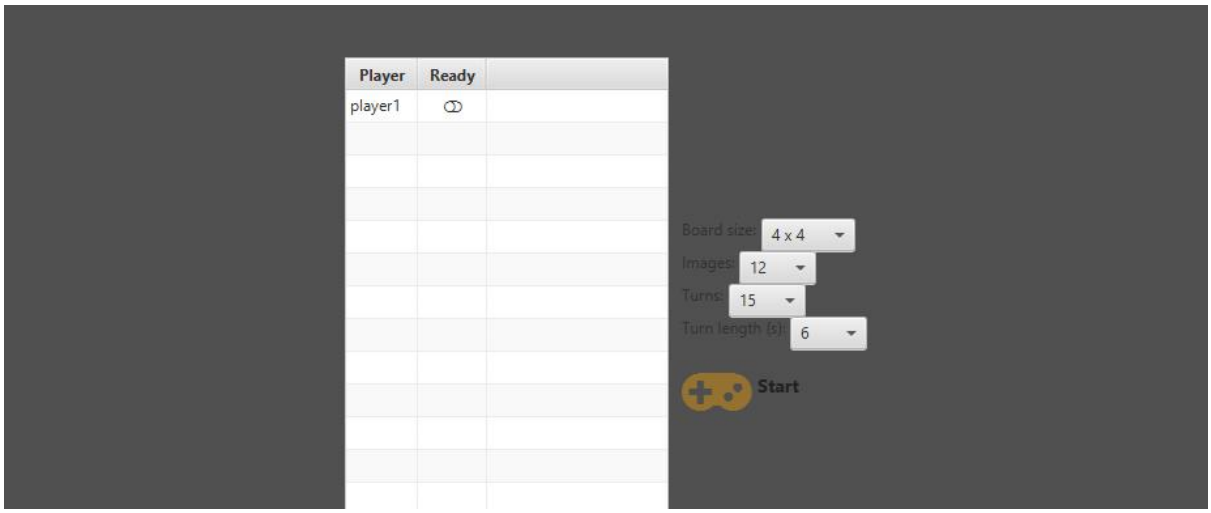

Figure 14: Room screen

Players now wait until the game is ready. The game starting proccess can be canceledby pressing ESC key. If the game starting is canceled, the players return back to room.

Figure 15: game starting screen

Once the game starts, the board is displayed and players can manipulate the images. The opposite tiles in the left top corner and right bottom corner are locked and players can not move the images. The locked images has their characteristics aligned with the axes, which requires prior data on the images available – this is done by the expert when dataset is being created. Theoretically, not all images need to have this information available.

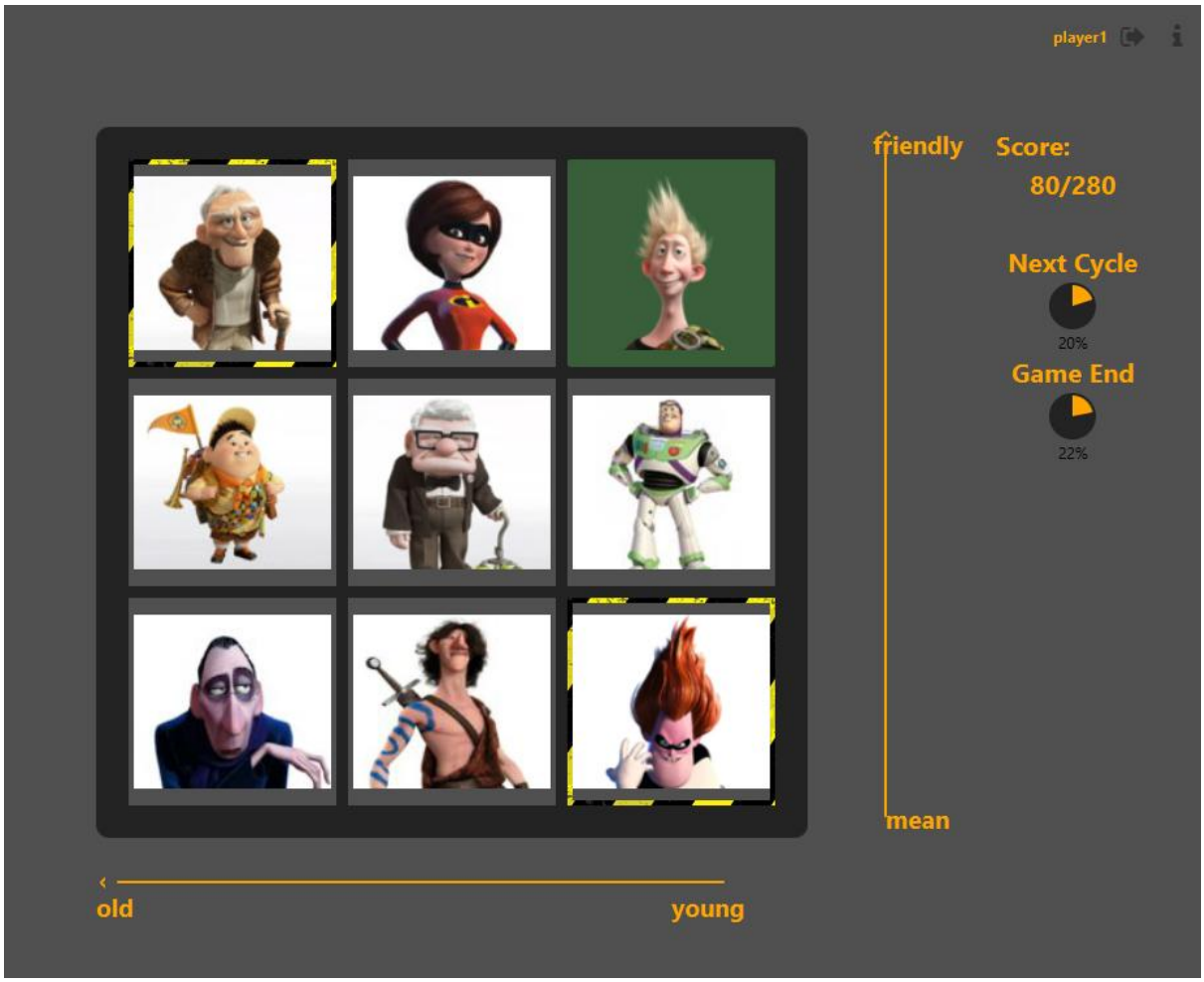

Figure 16: Game screen

Players must consider the locked images and axes and figure out the placing strategy for the images. Dragging the image will move it across the board and allow player to swap it with the image the dragged image currently hovers above. On the right side, timers display time game progress and cycle progress under latest score. Correctly placed (meaning the images placed on the same location) images have a green glow effect that slowly fades out on every pulse.

In the end of the game, all correctly placed images are permanently glowing to show players the final state. Score is then evaluated and shown to players and next game starts automatically. At any time user can disconnect or open guide using buttons in the right top corner of the game.

Figure 17: Board at the end of the game. Notice the green effect. In this particular case players get maximal score.

Appendix C – Content of attached CD

/application_projects: contains the complete projects of game server, game client and log analyzer

/applications: contains the executables of game server, game client and log analyzer

/experiment: contains the dataset, gamelogs and statistical results of the experiment

/thesis: contains electronic version of this document

Resumé

S rozmachom fenoménu Web 2.0 a množstva používateľmi generovaného obsahu sa čoraz viac dostáva do povedomia potreba efektívnej manipulácie a manažmentu multimediálneho obsahu. Automatizácia manipulácie dát strojmi si však žiada, aby stroje rozumeli obsahu spracúvaných dát a tiež kontextu v ktorom daný obsah vystupuje. Pretože stroje sú schopné spracovať iba homogénne dáta, ktorých štruktúra je vopred určená, heterogénna podoba dát produkovaných reálnych podmienkach a nutnosť poznania sémantiky dát vyžaduje použie tzv. dát o dátach - metadáta. Tie umožnujú manipuláciu dát, o ktorých identifikujú potrebné charakteristiky. Získanie metadát však predstavuje problém, pretože samotné vyžaduje znalosť a chápanie pre rozoznanie skúmaných charakteristík obsahu, ktorý metadata budú opisovať. Napriek výraznému zlepšeniu rozpoznávania multimediálneho obsahu, sa jedná o problém vyžadujúci kognitívne schopnosti človeka, ktoré je v súčastnosti nemožné simulovať umelou inteligenciou ani inými pokročilími algoritmami. Niektoré metadáta teda nie je možné vygenerovať automaticky a je nutné využiť ľudské znalosti a schopnosti.

Problém sémantiky multimédií a hier s účeľom

V súvislosti so získavaním metadát rozoznávame tri kategórie riešení. Riešenie spoliehajúce sa na znalosti expertov získavajú metadáta manuálne prácou ľudí, ktorí maju dostatočné znalosti s daným obsahom. Experti sú schopní generovať kvalitné metadáta, no za cenu slabej škálovateľnosti. Strojové riešenia sa snažia dolovať metadáta plne automaticky a sú schopné analyzovať obrovské množstvá dát, no kvalita výstupu pokulháva a to najmä na úrovni sémantiky. Sediac medzi týmito dvomi kategóriami sú riešenia využívajúce masy – veľké množstvo ľudí, poväčšine bez špecializovaných znalostí v doméne. V tejto kategórií riešení sa nachádzajú aj hry s účeľom (game with a purpose - GWAP), ktoré sú schopné využiť vedomosti alebo schopnosti samotných hráčov, ktorí tak vedome alebo nevedome ponúkajú riešenia pre problémy, ktoré boli transformované do hernej mechaniky hry. Hry s účeľom sú uznávané ako fungujúce riešenie pre získavanie metadát a našli svoje uplatnenie v praxi. Použitie hry s účeľom je však náročné a vyžaduje si dobrú analýzu problému, existujúcich riešení, vytvorenie a špecifikáciu schopného dizajnu hry a jej často netriviálnu implementáciu.

Táto práca sa zameriava na získavanie multimediálnych metadát, špecificky metadát sémantiky obsahu obrázkov pomocou hry s účeľom. Sématika a sémantické data sa čoraz viac objavujú v praxi, často v súvislosti so Sémantickým Webom a budovaniím taxonómií, folksonómií a ontológií, ktoré slúžia na organizovanie, vyhľadávanie alebo manipulovanie dát, tak ako sú na to ľudia prispôsobený. Termínom sémantika obsahu obrázkov referujeme k informáciam obsahu obrázka v istom kontexte. Pokročilý algoritmus detekcie obrázkov je schopný detegovať napr. auto, táto informácia sama o sebe však nie je úplne dostatočná a dostáva úplne iný rozmer, ak vieme použiť znalosti o autách a rozhodnúť o akcií ktorá sa na obrázku odohráva, typ auta o ktorý sa jedná (napr. športové) a množstvo iných aspektov, ktoré sa dajú formulovať iba za prítomnosti externých znalostí - kontextu. Dané informácie sa teda nielen že nedajú získať mechanickou analýzov surových dát obrázka (rgb raster) no kompletná analýza jednotlivých objektov na obrázku zlyháva. Pre účely extrakcie sémantiky obrázkov je teda nevyhnutné použiť masy ľudí.

Dizajn hier s účeľom si žiada preskúmanie a porozumenie aspektov, ktoré tvoria hru zaujímavu a zábavnú pre hráčov. Niektoré z nich sú: estetický zážitok, súperenie, sebazlešenie a interakcia s inými ľuďmi. Hra s účeľom navyše pridáva do hernej mechaniky účeľ (v našom prípade generovanie sémantiky), čo dizajn komplikuje zvýšenou komplexitou a limitáciou hernej mechaniky.

Existujúce riešenia

Hru s účeľom zostávajú pomerne mladým fenoménom, ktorý spopularizoval Luis von Ahn so svojou hrou ESP Game. Hra predstavuje azda najznámejší prípad hry s účeľom. Jedná sa o online hru pre dvoch hráčov, ktorých úlohov je opisovať spoločný obrázok tagmi. Herný aspekt pozostáva s hráčov snažiacich sa opísať obrázky rovnakými tagmi, za čo sú odmeňovaní. Pretože hráči nemôžu navzájom komunikovať, musia pre čo najviac bodov opísať obrázok čo najpresnejšie. Toto vedie

k vedľajšiemu účinku hry (z pohľadu hráča) – generovaniu sémantiky. Hra bola v rokoch 2006-2011 adoptovaná spoločnosťou Google a úspešne nasadená pod názvom Google Image Labeler pre zlepšenie výsledkov ich obrázkového vyhľadávača. Hra sa teší množstvu derivátov a iných hier, ktoré sa nechali inšpirovať. Medzi iné existujúce riešenia v doméne obrázkov patrí hra Peekaboom, ktorá rozoznáva pozíciu objektov na obrázku alebo SeaFish, ktorá sa snaží o linkovanie obrázkov podľa otvorených špecifikácií prepojených dát a RDF.

Špecifikácia hry

Naša hra s účeľom je kolaboratívna dosková hra pre dvoch hráčov. Hráči sú prezentovaní plochou so štvorcovou 3x3 maticou obrázkov a majú za úlohu s obrázkami manipulovať (vymieňať ich medzi sebou) aby dosiahli zhodu v pozící. Hra núti hráčov rozmiestňovať obrázky na základe ich sémantiky. Hráči si musia vybudovať stratégiu rozmiestňovania obrázkov, k čomu ich navádzajú niektoré herné elementy. Toto umožňuje hráčom vložiť do hry ich znalosti o obsahu obrázkov. Hráči hrajú na rovnakej hracej ploche s odlišným počiatočným rozmiestnením obrázkov a nevedomky si navzájom validujú vlastné akcie. Analýzou absolutnych a relatívnych pozícií obrázkov počas a na konci hry potom vieme extrahovať sémantiku obrázkov. Pri dostatočnom množstve dát je možné zachytiť vlastnosti obsahu naprieč množinou obrázkov. V rámci hernej mechaniky rozoznávame niekoľko dôležitých elementov:

- Hracia plocha obsahuje dve osi (vertikálnu a horizontálnu), ktoré reprezentujú skúmanú charakteristiku obrázkov a signalizujú minimum a maximum na opačnych stranách dosky. Herná plocha sa tak mení na akýsi gradient, v rámci ktorého musia hráči preusporiadať obrázky. Hráč najprv analyzuje obrázky na ploche a potom ich porovnáva medzi sebou a postupne vymieňa.
- Ďalším elementom je dočasná nápoveda zhôd, ktorá vizuálne signalizuje hráčom, ktoré obrázky majú identickú pozíciu. Hráči sa tak môžu sústrediť na zvyšné obrázky a postupne dosahovať vac aviac zhôd.
- Veľkost dosky predstavuje rozmer hracej plochy meraný v počte obrázkov. Štandartná hodnota parametra je 3.
- Prázdne pozície predstavujú pozicie, na ktorých sa žiadne obrázky nenachádajú. Prázdne pozície sú súčasťou hry nakoľko hráč na ne stále môže presunúť obrázok. Jedná sa o dôsledok zníženého počtu obrázkov na ploche.
- Uzamknuté pozície predstavujú pozície, s ktorými hráči nemôžu manipulovať. Dané pozície sú rovnake pre oboch hráčov a obsahujú rovnaké obrázky. Obrázky sa nerátajú do bodového hodnotenia. Tento herný element poskytuje hráčom počiatočný odrazový mostík na začiatku hry.
- Priebežné skóre predstavuje mieru počtu zhôd k počtu obrázkov, vyhodnocovanú ako celé čislo. Do skóre sa započítavajú aj nezhody, ak sú obrázky dostatočne blízko.
- Herný pulz referuje k periodickému vyhodnocovaniu hernej plochy. Hra je charakterizovaná pulznou periódov a počtom pulzov. Každý pulz vyhodnotí pozície a vzdialenosti jednotlivých obrázkov, ukáže nápovedu zhôd a prepočíta priebežné skóre. Skóre je prepočítavané každý pulz a indikuje mieru úspešnosti v danom okamihu. Hra končí posledným pulzom.
- Celkové skóre je priebežné skóre na konci hry.

Implementácia

Architektúra riešenia pozostáva z viacerých častí (modulov) v snahe o zvýšenie modularity implementácie:

- Herný server predstavuje ťažisko hernej logiky. Z dôvodu zamedzenia neférového správania (angl. *cheating*) sa jedná o autoritatívny server, čo znamená že činnosti hráčov sú validované serverom a modifikácia hry neumožnuje neférové praktiky. Server je aplikácia vyvinutá v programovacom jazyku java a umiestnená na virtuálnom stroji v rámci služby Microsoft Azure. Herný klienti sa na server pripájajú a komunikujú s ním počas hry, na čo server posiela odpovede. Herné akcie sú teda vyvíjané asynchrónne.
- Herný klient je aplikácia vyvinutá v programovacom jazyku java (špeciálne JavaFX)

a ponúka možnosti nasadenia ako klasická aplikácia alebo ako webštart.

- Úložisko datasetu ponúka prístup k obrázkom použitým v priebehu hry. Toto úložisko je oddelené od herného serveru, ktorý iba pozná URL linky k daným obrázkom. Herný server sa tak nijako nepodieľa na dátovej komunikácií v súvislosti s obrázkami, čo bol náš cieľ z dôvodu minimalizovania množstva dátovej komunikácie.
- Logovací modul je súčasťou herného serveru a má za účeľ generovať herné záznamy. Tie obsahujú informácie o priebehu samotných hier. Herné záznamy sú reprezentované JSON súbormi.
- Analyzačná aplikácia je samostatná aplikácia určená pre štatistickú analýzu herných logov po skončení experimentu.

Experiment

V rámci tejto práce sme vykonali uzavretý experiment na získanie dát a overenie funkčnosti hry. 5 hráčov odohralo spolu 41 hier s rôznymi parametrami hry. Použili sme vlastný dataset rozprávkových charakterov o veľkosti 27 obrázkov. Sledovali sme dve charakteristiky definované ako osi mladý-starý a milý-nepríjemný. Na začiatku boli obrázky manuálne vyhodnotené expertom zoradením podľa danej charakteristiky a nasledovným priradením hodnoty. Vyhodnotením herných záznamov sme boli schopní vyhodnotiť mieru vhodnosti každej charakteristiky pre každý obrázok, mieru obtiažnosti určenia danej charakteristiky hráčmi a podobnosť kazdej dvojice obrázkov. Vygenerované hodnoty charakteristík korelovali s expertnými údajmi.

Závery

Naším cieľom bolo navrhnúť a zrealizovať hru s účeľom pre získavanie sémantiky a linkovanie obrázkov. Hru sme navrhli na základe analýzy existujúcich riešení a zohľadnením aspektov pre dizajnovanie hier s účeľom. Niektorí hráči považovali hru počas experimentu za zábavbú, jeden pre možnosť odhadnúť úmyseľ druhého hráča a druhý vďaka samotnej hernej dynamike (obaja hráči sa často venujú hraniu vo voľnom čase), považujeme preto cieľ vytvrorenia zábavnej hry ako splnený. Uzavretý experiment rovnako naznačuje, že hra je schopná generovať správne sémantické data. Škála experimentu je však nedostatočná pre definitívne závery a testovanie tiež odhalilo niekoľko nedostatkov hry. Preto je súčasťou budúcej práce otvorený experiment s vylepšenou verziou hry a ostrým nasadením s cieľom získať väčšie množstvo dát a použiť väčší dataset.