Rudolf Wagner, Bsc

Myth Hunter - Design and Implementation of a Location-Based Learning Game

Master's Thesis
to achieve the university degree of
Master of Science
Master's degree programme: Software Engineering and Management

submitted to

Graz University of Technology

Supervisor
Assoc.Prof. Dipl.Ing. Dr.techn. Christian Gütl
Dipl.Ing. Dr.techn. Johanna Pirker

Institute of Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Graz, November 2017

Rudolf Wagner, Bsc

Myth Hunter - Entwurf und Umsetzung eines standortbezogenen Lern-Spiels

Master Arbeit
zur Erlangung des Akademischen Grades
Diplom Ingenieur
Master Studium: Softwareentwicklung und Wirtschaft

an der

Technischen Universität Graz

Betreuer
Assoc.Prof. Dipl.Ing. Dr.techn. Christian Gütl
Dipl.Ing. Dr.techn. Johanna Pirker

Institut für Informationssysteme und Computer Medien (IICM)
Leitung: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Graz, November 2017

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Abstract

Motivating learners is of vital importance in education, especially when educating children. But what motivates people? When looking at video game and mobile-game statistics one realizes that this kind of applications manages to drive many people for long timespans. Also, the recent hype for engaging mobile apps such as Pokemon Go[1] showed that with strong incentives players can be motivated to execute activities, they wouldn't do otherwise (e.g., walking several kilometers to hatch an egg). Studies show that successful interweaving playing and learning not only motivates pupils within class but they are also more likely to learn about a subject in their free time (Kuo, 2007).

How can these motivation capabilities of video games and the recent hype for mobile applications be used for education? Many different approaches to mobile game-based learning have been developed over the past years. These games often use geolocations as part of their game concepts and allow to connect educational content with physical locations and objects. However, because of this location-connection, the scope of such games is often insufficient. Some applications try to broaden their reach by using user-created scavenger hunts but fail to connect these independent quests to create long-term motivation.

The *Myth Hunter* project described in this thesis aims to overcome these limitations by interweaving educational digital scavenger hunts with the long-term motivational aspects of a collection-card game. Basically for each completed scavenger hunt users gain rewards in the form of play cards which can be collected and played with. These cards are also needed within some of the scavenger hunts to defeat enemies. The scavenger hunts, as well as new cards, can be created by users.

This thesis includes background information and related work. It covers the requirements, design, and development of the *Myth Hunter* mobile application and a server functioning as global data storage. Used tools and frameworks will be described as well as specific technical design and user interface decisions.

Finally, two evaluation studies conducted with *Myth Hunter* are described. The results show that Myth Hunter is perceived very motivating and easy to use. Its educational capabilities, however, depend heavily on the structure of the scavenger hunt and how the content is presented. Based on feedback received during the evaluations possible enhancements of *Myth Hunter* to further strengthen its educational capabilities are mentioned.

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

Kurzfassung

Lernende im Zuge einer Ausbildung zu motivieren ist von entscheidender Bedeutung, speziell wenn es sich um Kinder handelt. Nun stellt sich die Frage: was motiviert Menschen? Videospiele, sowohl am Heim-PC als auch auf Smartphones, schaffen es viele Menschen über lange Zeitspannen hinweg zu motivieren. Speziell der momentane Hype um mobile Smartphone Apps, wie etwa Pokémon Go, zeigt, dass starke Anreize Spieler zu Aktivitäten motivieren können die sie ansonsten möglicherweise nicht durchführen würden (z.B. mehrere Kilometer gehen um ein Pokémon-Ei auszubrüten). Studien zeigen, dass erfolgreiches Verknüpfen von Spielen und Lernen Kinder nicht nur während des Unterrichts motiviert sondern dass sich diese auch mit höherer Wahrscheinlichkeit nach der Schule mit dem Lern-Stoff beschäftigen (Kuo, 2007).

Wie können diese motivierenden Aspekte von Videospielen und der momentane Hype um mobile Apps für Bildung genutzt werden? Viele verschiedene Methoden um mobiles, spiel-orientiertes Lernen zu ermöglichen wurden in den letzten Jahren entwickelt. Einige dieser Spiele nützen Standort-Daten als Teil des Spielkonzepts und verknüpfen Lerninhalte mit realen Orten oder Gegenständen. Durch diese Ortsbezogenheit sind die Einsatzgebiete solcher Apps jedoch meist sehr beschränkt. Manche Anwendungen versuchen ihre Einsetzbarkeit zu verbreitern indem sie benutzer-generierte Schnitzeljagden nützen. Diese unabhängigen Schnitzeljagden stehen aber innerhalb der App in keinem Zusammenhang miteinander wodurch wenig bis keine Langzeitmotivation aufkommt.

Das *Myth Hunter* Projekt, welches in dieser Arbeit beschrieben wird, umgeht diese Schwachstellen indem es digitale Bildungs-Schnitzeljagden mit den langzeitmotivierenden Aspekten eines Sammelkartenspiels kombiniert. Für jede abgeschlossene Schnitzeljagd erhalten Spieler Belohnungen in Form von neuen Spielkarten. Diese Karten können einerseits gesammelt werden und werden andererseits in einigen Schnitzelgaden genützt um Widersacher zu besiegen. Sowohl die Schnitzeljagden als auch neue Spielkarten können von Benutzern erstellt werden.

Diese Arbeit beinhaltet Hintergrundinformationen und Beschreibungen ähnliche Projekte sowie die Anforderungen, das Design und die Umsetzung der *Myth Hunter* App und eines dazugehörigen Servers der als globaler Datenspeicher fungiert. Eingesetzte Werkzeuge und Frameworks sowie spezielle Entscheidungen bezüglich Design und Benutzeroberflächen werden beschrieben.

Zwei Evaluierungs-Studien wurden für *Myth Hunter* durchgeführt. Diese Studien und ihre Ergebnisse werden im Detail beschrieben und disskutiert. Die Resultate zeigen, dass die App als sehr motivierend aufgenommen wird und sie leicht zu bedienen ist. Die Effektivität der Wissensvermittlung durch *Myth*

Hunter hängt jedoch sehr stark von der Struktur der Schnitzeljagden sowie von der Art der Informationsaufbereitung ab. Basierend auf diesen Studien wurden mögliche Verbesserungen abgeleitet welche am Ende dieser Arbeit genannt sind.

Acknowledgements

I wish to express my deepest gratitude to my supervisors Christian Gütl and Johanna Pirker who not only constantly supported me during this work but suggested to transform the rough idea of Myth Hunter we had at the time into a master thesis in the first place.

Special thanks to Armin Hutzler who is one of the heads behind the idea of Myth Hunter and creator of the authoring tool which allows creating scavenger hunts and game-cards easily. For very valuable input regarding the Myth Hunter concept, I want to thank David Kastl and especially Caterina Nahler who also provided all textures used in the Myth Hunter App.

I want to thank Ulrike Steiner, Christiane Tanzmayr and their whole secondary school class for their enthusiasm and patience during the release evaluation. Also thanks to all participants of the prototype evaluation. Your feedback and ideas were greatly appreciated.

Finally I want to say "Thank you!" to my family and friends for their continued support and source of motivation throughout my whole study.

Graz, Austria, October 2017
Rudof Wagner

Contents

List of Figures

1 Introduction

In today's society computer games tend to play an important role, especially amongst children and adolescents. Already in 2003 Prensky estimated that an individual had spent approximately 10,000 hours playing video games when reaching the age of 21 years (Prensky, 2003). A study by Gentile (2009) showed that 88% of Americans between eight and 18 play at least occasionally video games. Especially in recent years the use of mobile games on smartphones and tablets has increased each year steadily according to statista[1].

It is safe to say that intrinsic motivation towards digital games is very high among young people and also tends to increase among adults. On the other hand, traditional education is often experienced as rather dull and even meaningless compared to the rich and challenging environments of games. "Digital Game-Based Learning" tries to combine educational content with motivating game elements to create a more motivating and therefore more efficient way of education (Prensky, 2003). Many different approaches to game-based learning have emerged over the years. Recently a focus on mobile location-based learning games is notable utilizing their ability to connect educational content with physical locations or objects. However, their scope is usually very limited because of this connection to physical locations. Some applications overcome this issue by allowing users to create scavenger hunts wherever they want but fail to make meaningful connections between these scavenger hunts and therefore lack longtime motivational features.

This thesis presents the application *Myth Hunter* which aims to overcome these issues by connection digital educational scavenger hunts with the longtime motivating features of a collection-card game by offering additional cards as a reward for completed scavenger hunts.

1.1 Goals and Objectives

This work describes the design, development, and evaluation of the location-based mobile game *Myth Hunter*. This application allows the execution of user-created scavenger hunts including a collection-card game which is also expandable by user-created cards. During a field study Myth Hunter's usability as well as motivational and educational capabilities are evaluated. This work focuses on the mobile application itself whereas a separate application used for the

[1]https://www.statista.com/statistics/234635/number-of-mobile-gamers-forecast/

creation of mentioned scavenger hunts and cards exists. This editor is explained in the thesis "Design and Implementation of a Web-based Authoring Tool for Location-based Gamified Applications" by Hutzler (2017, under review).

The goal is the creation of an easy to use, fun to play application which supports game-based learning. Any educational content can be included in such a scavenger hunt if it has connections to physical places, objects or the story of the scavenger hunt. Furthermore, the user's ability to navigate through unknown train using a map should be increased.

1.2 Structure of this Thesis

This thesis about the design and development of the location-based learning game *Myth Hunter* is structured as follows: in Chapter 2, the theoretical background and related work is discussed. Different types of educational software are mentioned focusing on how to introduce game elements into digital education. Several examples, showing different approaches, are mentioned and discussed. Taking the rise of smartphone usage into account location-based games and their applicability for educational purposes are discussed. Additionally, to broaden the game experience of a location-based application, popular collection-card games are discussed with the aim of extracting requirements for a new game of this genre. Finally, artificial intelligence algorithms and their applicability for such a collection-card game are discussed.

Chapter 3 describes requirements the *Myth Hunter* application has to fulfill. In a first step functional requirements of the three main areas of the application: (1) the scavenger hunt, (2) the card game and (3) additional gamification are formalized. Afterwards, several non-functional requirements are defined including usability and data usage. Finally, based on this requirements a conceptual design is created and the communication between components of this design is explained using the use case of a quest execution as an example.

In Chapter 4, details about the development of client and server application are provided. Tools and frameworks used for the development are mentioned as well as technical decisions made during this phase. The description of these decisions is again split into the three primary application areas and extended by a general implementation section. Finally, a user-centered view on the application is given including user interface and general usage explanations.

Chapter 5 covers two evaluations of the *Myth Hunter* application. A first prototype evaluation was done early in the development process focusing only on the scavenger hunt as proof of concept study. A second evaluation was done after the development was finished including 17 participants of the target group of secondary school children. Both evaluations are explained in detail ant their results discussed and interpreted.

In Chapter 6, experiences made during the work on this thesis are discussed. This includes ideas and problems raised during the theoretical background research as well as strengths and flaws of the design and development process. This chapter also discusses possible enhancements of the *Myth Hunter* application and future work that could be done to strengthen its educational capabilities further.

2 Background and Related Work

This chapter provides background information about software that aims to help pupils learn including software-aided teaching tools as well as dedicated learning games. The evolution of such software, as well as their effectiveness, is discussed. Additionally, the possibilities of location-based learning-games are discussed by taking a look on some examples in this field. Finally, the popular branch of collection-card games is mentioned as a successful example of motivating game features and their applicability for learning games.

2.1 Motivation

"A sine qua non of successful learning is motivation: a motivated learner can't be stopped". This quote from Prensky's article about digital game-based learning indicates that motivation is of vital importance when aiming to create a successful learning environment (Prensky, 2003). Studies show that students in game-based learning environments are both more motivated to learn about a new subject and more likely to come back to the subject after school hours (Kuo, 2007). A lot of software tools have emerged in the last decades which combine these game-based aspects and educational content to help children learn playfully.

This chapter takes a closer look at computer, or more specifically, software-aided teaching. The possibilities and results of adding game-like aspects to such software are discussed. Furthermore, the use of dedicated learning games is examined.

This thesis presents the software "Myth Hunter" which combines the hands-on experience and teaching possibilities of a location-based application with the motivational aspects of collection-card games. This chapter gives insights into these two domains by focusing on the strengths and weaknesses of existing location-based games and learning applications as well as popular collection-card games and their motivational features.

From a technical point of view, the implementation of an artificial intelligence (AI) algorithm for collection card games is a difficult task, but many different AI algorithms are already known in the literature. This chapter also gives an overview of traditional game-AI approaches and discusses their applicability on a collection-card game such as "Myth Hunter".

2.2 Educational Software

This section covers educational software by giving background information on its history as well as different types and examples of such software.

2.2.1 Overview

Computer-aided education is no new field of study. "*Since the early 1960s educational technologists have been developing programs of computer-based instruction (CBI) to drill, tutor, and test students*" (Chen-Lin Kulik, 1991). Already back in the 1980s Niemiec, Sikorski, and Walberg (1989) concluded that computer-assisted instruction was considerably more cost-effective than traditional tutoring. The primary use of computer technology in these years was focused on practicing basic math and language arts skills. Starting from the 1990s, usage of computers and also the variety of tasks executed on computers during education increased significantly (Henry Becker, 1999). Cuban (2001) stated that especially in preschool and primary education computer usage is integrated into the educational process quite extensively saying that "*The computers are left on all day, and they are in constant use*"

2.2.2 Types of Educational Software

.Nowadays, a vast amount of software tools aiding teachers in their everyday work is available:

Course and learning management systems such as "Moodle"[1] or "Edmodo"[2] are mostly used for online or blended learning and provide functionality for placement of course materials online, associating students with courses, tracking student performance, uploading student submissions as well as moderating and mediating communication between the students and their instructors (William R. Watson, 2007).

Another group of software solutions designed to help children learn is the so-called **reference software** which was already used in classrooms in the mid-90s and is still common in today's education. Educational book publishers like the Austrian publisher "Helbling Verlag"[3] ship learning DVDs alongside their study books to help pupils practice the skills learned during lectures or even at home.

Together with the rise of smartphone popularity also the field of **mobile learning** was created. Already in 2007 when the first Ipone was presented Traxler (2007)

[1] https://moodle.org/
[2] https://www.edmodo.com/
[3] http://www.helbling.at/

tried to define mobile learning by various aspects. From a technical point of view, all learning that happens on mobile devices such as phones or PDAs can be considered as mobile learning. From user perspective the term is connected with word such as *'personal,' 'spontaneous,' 'opportunistic,' 'informal,' 'pervasive,' 'situated,' 'private,' 'context-aware,' 'bite-sized,' and 'portable.'* Furthermore, smartphones allow the shift from learning 'just-in-case' to 'just-in-time' meaning that information can be retrieved just when needed instead of remembering it beforehand in case it is needed later on. However mobile devices also foster 'just-in-case' learning by enabling users to learn whenever they have time wherever they are at the moment. Unique to mobile learning is the ability to connect educational content to physical locations and object. This element of location-based learning are discussed later in this chapter in further detail.

2.2.3 Discussion

Educational software is used very widely in schools nowadays to help pupils learn or structure their educational content. The field of mobile learning also fosters learning outside of classrooms and schools. However, educational software tools often lack motivational features and are perceived as somewhat dull and tedious to use. On the other hand, children like to play and are usually very motivated to do so again and again. This lead to the development of game-based educational software.

2.3 Game-based Education-Software

Game-based educational software tries to combine learning and playing. This section describes different types of game-based educational software and discusses example applications of this field.

2.3.1 Overview

Motivation is of vital importance when aiming to train or educate people. In early childhood, children learn about their surrounding by playfully exploring it. Nowadays educational software developers try to integrate this motivational factor of playing into their software to allow also older children and even adults to learn while playing as they used to in their early years. Several different types of game-based education-software have been developed in the last few decades. One of the first approaches in this field were so-called "Serious Games".

Serious Games: The term serious games was first used by Abt (1970) saying that serious games *"unite the seriousness of thought and problems that require it with the experimental and emotional freedom of active play."* Peña-Miguel Noemí (2014) state that serious games can and should be entertaining but their main purpose is to train or educate users. A popular example of serious games are digital flight simulators combining the objective of training pilots exploratory nature of games. Also the game "America's Army"[4] developed by the U.S. Army in 2002 is considered to be a serious game because of it's primary purpose to recruit new soldiers. Most serious games are developed for (young) adults and teach or train subjects or skills outside of traditional school education. Games with the purpose of teaching a particular subject respectively skill, expand concepts or reinforce development are called educational games (Spaulding, 2012).

Educational Games: This kind of software products is usually used either in class or at home as an addition to traditional education. As several studies (Facer et al., 2004; Schwabe & Göth, 2005) concluded, game-like features in a learning environment have an enormous potential if done correctly. These studies state that most students enjoyed such learning games and perceived them as very motivating. Several studies (Al-Tarawneh, 2016; Maria Virvou & Manos, 2005; Shiue, Hsu, & Liang, 2016) conclude that in general, a performance increase is notable and that students would be quite happy to use educational games instead of conventional educational software. Especially poor performers in traditional education environments benefit the most by the use of educational games in classrooms while the overall performance of good students stayed the same. This may be explained by the fact that good students perform well under any conditions whereas poor performers need the extra motivating environments found in games. During these studies, teachers confirmed that pupils who were usually not easily disciplined in class had been absorbed by the game decreasing their willingness to take time out and talk to other students. Also, a decrease in cheating attempts by these students was notable. These studies also revealed that there is no significant difference in students' knowledge acquisition attributed to the gender variable. However, not every software can be a game or be designed as such. In the last few years, researchers and companies tried to extract motivation features usually found in games and introduce them in non-game context as well. This approach is called gamification.

Gamification: Deterding, Dixon, Khaled, and Nacke (2011) describe gamification as "*the use of game design elements in non-game contexts*". Werbach (2014), on the other hand, describes gamification rather as the process of making an application more game-like then just sticking in some game-like features. Deterding et al. (2011), as well as Werbach (2014), agree however that the purpose of gamification is to make an application more fun to use, encourage users to voluntarily interact with it in a repeated manner. Some gamification features that learning applications frequently use are achievements/badges that encourage students to keep learning to earn the next badge for their collection. Points and leaderboards aim at the competitive nature of children. They enable the users to compare

[4]https://www.americasarmy.com/

themselves to their colleagues and beat the high score.

2.3.2 Examples of Game-based Education-Software

This section discusses some recent developments in the area of game-based education-software including prototypes and their evaluation results. These examples show the variety of gamification approaches in different fields of education.

The project **Racing Academy** by the Bristol: Futurelab[5] is a prototype for a *massively multiplayer online engineering and racing car simulation*. By explaining complex engineering and physics principles, it allows users to modify cars and virtually test-drive them on a straight track representing a quarter-mile drag-race against an artificial intelligence driver. The game includes options to change the cars engine as well as tire and gear ratios to find the optimal setup and beat the best time. Additionally, a message board was integrated allowing users to discuss different setups and help each other. Even though it is possible to beat the game by trial and error, the big number of different possibilities encourage users to make effective engineering decisions.

The prototype was evaluated by 40 students of two secondary schools in Briston. It revealed that many of the participants found the game engaging, challenging and rewarding even though it was considered to be very tough at first. Despite the significant amount of possible alterations many participants used a trial-and-error approach and often retested different setups. Still, a post-questionnaire, as well as observations after the experiment, revealed that many of the students began to grasp some of the presented principles (Sandford & Williamson, 2004).

Lin and Wei (2011) created the **Interactive Game-based Learning System (IGLS)** arguing that e-learning and serious games *"not only trigger users' intrinsic motivation, but also arouses peer cooperation and coordination"* especially for underachievers. The system allows students to review their learning as well as compete against other participants in the form of quizzes. This quizzes are performed as group-versus-group challenges where each team has a start amount of hit points and a pool of randomly selected questions about a subject. Players can now attack the opponent team by selecting one of these questions. If the defending team gives a wrong answer, their hint points are reduced. If the answer is correct, the attacking team's hit points will diminish. The game is over when one loses all hit points.

After evaluating IGLS in a Taiwanese seventh-grade history-class, Lin and Wei (2011) concluded that IGLS benefits academic performance in junior high history learning and arouses notable improvement in both motivation and concentration.

[5]https://www.nfer.ac.uk/futurelab/

Figure 2.1: Screen-shot of Jingnan Campaign (Yu, Yang, Chen, Yu, & Hou, 2015)

The serious game **Jingnan Campaign** created by a research group of the National Taiwan University of Science and Technology integrates the historic facts about the Jingnan Campaign, a civil war in the early Ming Dynasty in China, into a strategy game. During the game campaign where players can command armies into battle (see Figure 2.1) historical information is presented to the user as intelligence clues. Before each battle a test named "Jiaozhen" is performed in which the the generals of each side shout questions about the Jingnan Campaign to their opponent. Correct answers boost the armies morale while wrong answers result in an attack from the opponent and corrective feedback. The users, however, do not need to know the answers by hard but can analyze the intelligence clues gained earlier and reason the answers which enhances the higher-order cognitive thinking.

A study including twenty-nine college students from Taiwan revealed that perceived usefulness was above the median and also a significant progress in history knowledge could be observed. However, the dimension of technology acceptance did not reach the median which suggests, that the usability of the game did not reach the required level (Yu, Yang, Chen, Yu, & Hou, 2015).

2.3.3 Discussion

These applications are only a few representative examples of game-based education-software, but they give an idea of possible approaches. Most of these software solutions manage to motivate the users and can educate learners about their specific domain but are not applicable to any other area. Furthermore, these applications usually do not include any physical illustrative material or require any player movement which helps some people to absorb information better. The physical action itself can also play an essential part of a motivating game experience. Especially many smartphone games nowadays include physical movement and the player's actual physical location in the game concept. This kind of games is called location-based games.

2.4 Location-Based Games (LBG)

This section explains the definition of location-based games (LBG) and the different types from technical point of view. Furthermore it mentions and discusses some example applications of this field.

2.4.1 Overview

Location-based Games (LBG) are played on, or with the help of mobile devices, supporting physical location determination. Ejsing-Duun (2011) states that " *in order to be considered an LBG, there must be a link between locations in the physical world and game-play. The outcome of the LBG always depends on the player's movement and locations in the physical world.*" Rashid, Mullins, Coulton, and Edwards (2006) define LBG as games that "*allow users to extend their game-playing outside cyberspace by incorporating interaction with the physical world*. These definitions are quite general, as long as a game of any kind has any connection to physical locations or movement of the player it is considered a location-based game.

Sotamaa (2002) divided LBG from a technological point of view into three groups:

The first LBG were not played on smartphones or PDAs but just used **Global Positioning System** (GPS) receivers. These games were usually outdoor treasure hunt games and could be played on the entire planet.

Another method of determining a players location is to use **local area networks** (WLAN etc.) and proximity sensors. Such games can use physical sites, objects or even other players location as fundamental game elements. The game area, however, is limited by the range of the network.

"*The third category of location-based games consists of the ones taking advantage of* ***cell identification in GSM networks****. GSM network based locating is not as accurate as other alternatives but the advantage is that cell identifi cation does not require any new hardware or additional cards but the games can be played by using standard GSM phones*" (Sotamaa, 2002).

In modern smartphones, mostly GPS and local area networks are used simultaneously to calculate the precise physical location of the device, but for some LBG, like Geocaching, a simple GPS tracker is also sufficient.

2.4.2 Examples of Location-Based Games

The following section describes a view early examples of LBG as well as some top-rated recent games using physical location as a game element. The vast amount of players of some of these games as well as the fact that also early LBG are still played by a lot of people show the enormous motivational possibility of combining physical exercise with gaming.

Geocaching is basically a modern, high-tech version of "hide and seek" and was the first game considered to be a location-based game. To play Geocaching only a web browser and a GPS-unit are needed. The principle is simple: one user hides some kind of treasure, the so-called "cache" in a particular place and published the GPS coordinates on the geocaching website. These caches are usually placed in waterproof and camouflaged containers and consist of items like toys, stickers, music recordings or pictures (Matherson, Wright, Inman, & Wilson, 2008; Lary, 2004). All other "geocachers" can now try to find the hidden treasure. In general, there are 3 rules to follow after one found a cache:

1. Take an item from the cache.
2. Leave another item in the cache container in the old item's place
3. Record your visit in the accompanying log book within the cache container

The first cache was hidden by engineer David Ulmer near his home in May 2000 a few days after the GPS signal was made available for civilian usage. He posted the location on the web and challenged people to find it. A few months later programmer Jeremy Irish build the first official geocaching website. Since then over half a million caches have been hidden in more than 180 countries around the globe (Schlatter & Hurd, 2005).

Additionally to the traditional caches a view variants of Geocaching have been established:

- Multi Cache: Participants have to find multiple caches an collect information to work out and find the final cache which contains the log-book and treasures (O'Hara, 2008).
- Puzzle Cache: A puzzle must be solved to work out the location of the cache. On the Geocaching website, each puzzle cache has a dedicated web page containing "*contextual information about the cache and the site, maps of starting points, a puzzle to work out the coordinates, and useful tips or things to look out for*" (O'Hara, 2008).
- Virtual Cache: These type of caches focus on landmarks or historical places. A Geocacher must locate the site and let the owner of the cache know that the location was found by answering a question about it (Matherson et al., 2008).

Can You See Me Now? (CYSMN) developed by the Mixed Reality Laboratory at the University of Nottingham, a partner in the UK's Equator project, and Blast

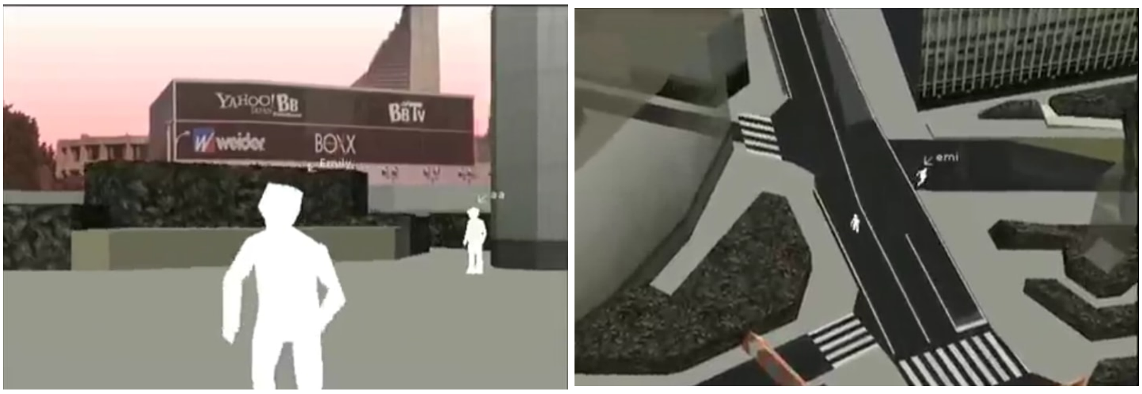


Figure 2.2: The different view-options of an CYSMN online player trying to hide from the runners ("CAN YOU SEE ME NOW?" 2017)

Theory, an artists' group, was an experimental location-based game deployed at two new-media festivals between 2001 and 2003. The purpose of the experiment was to study possibilities of how to work with GPS uncertainty in a location-based game.

The game consisted of two types of players: Online players control an avatar moving with a fixed speed around the virtual representation of a city on different zoom levels of an abstract 3D view (see Figures 2.2). Runners, like in Figure 2.3, are equipped with GPS modules and mobile screens showing the virtual representation while running through the real, physical city. Both player groups could see each other within the virtual representation and the runner's goal was to catch the online players by moving to the physical location represented by the online player's virtual position. The runners were special performers hired by the research group while the online players were random people joining the experiment.

"*Overall, CYSMN was well received, and the players seemed to feel genuine engagement and even tension, especially when the game was working smoothly.*" However, problems concerning GPS accuracy or even complete disappearance of the runners in the virtual city were quite frequent which lowered the players' acceptance factor (Benford et al., 2003).

The game **Ingress**[6] developed by Niantic Labs[7] in 2012 is a massively multi-player online augmented reality game. The game is played using Android or IOS smartphones showing a virtual representation of the physical world (see Figure 2.4). Part of this map are so-called portals, which mark important sites like buildings and statues. All Players are divided into two groups with the aim of capturing those portals. As soon as a team is able to control at least three nearby portals these sites are linked and create a virtual triangular field. The size of the field and it's geographical coverage as well as the statistical number of

[6] https://www.ingress.com/
[7] https://www.nianticlabs.com/

Figure 2.3: A CYSMN Runner with the hand-held computer device and GPS tracker (Benford et al., 2003).

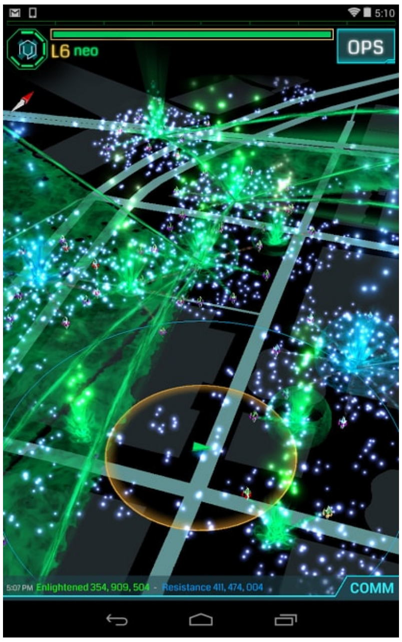
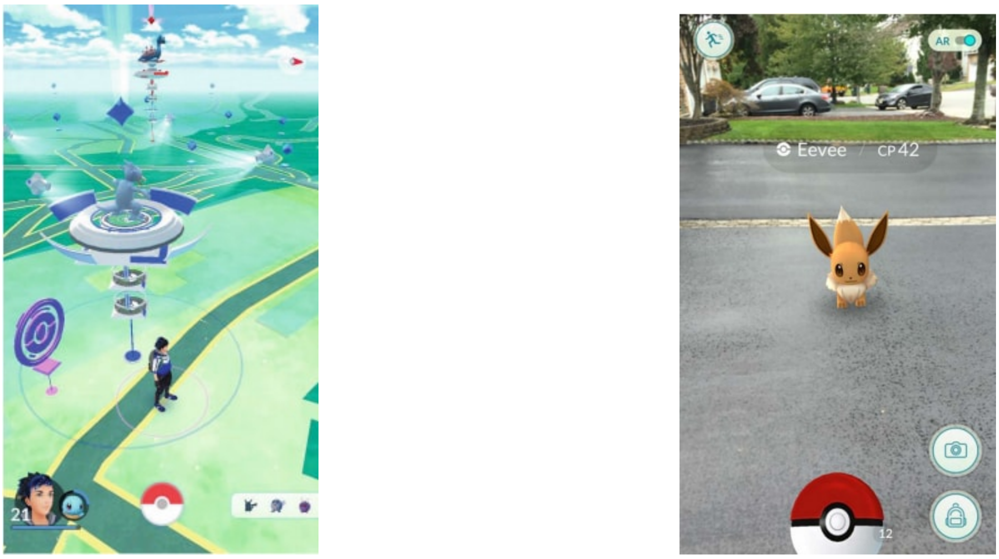


Figure 2.4: Ingress in-game map ("Ingress," 2017)

people living within that field determine the amount of points the team receives for capturing those portals. Portals can be placed on the map by using a web browser and accessing the Ingress Intel Map and identifying important sites. Descriptions and pictures of the real location can be added so games can also learn basic facts about the visited sites during the game (Sheng, 2013; Chess, 2014).



a) Map representation of the nearby area and the player (Meschtscherjakov, Trösterer, Lupp, & Tscheligi, 2017).

b) Camera picture with Pokemon overlay (Huang, 2016).

Figure 2.5: Pokemon Go screen shots

The basic concept of Ingress is straightforward and therefore easy to grasp by new players. The division in two big teams makes it easy to connect with other people and work towards a common goal. Even though it includes information about some sites and therefore could also be used to learn about famous places of a city, it is not necessary to read this information to play the game. The primary focus lies on capturing the portals hence it can not be categorized as learning application.

The location-based game **Pokemon Go** quickly became a global phenomenon. The gaming app for iOS and Android was released in July 2016 and features a customized version of Google Maps to visualize the current geographical location of the player as shown in Figure 2.5a. The goal of the player is to catch virtual creatures called Pokemon which appear randomly on the map if the player is close by. As can be seen in Figure 2.5b, " *after tapping on a Pokemon on the map, it is visualized as an overlay on top of the smartphone camera image creating an augmented reality (AR) and the player may catch it by throwing a PokeBall at the it with a flick gesture on the smartphone*" (Meschtscherjakov, Trösterer, Lupp, & Tscheligi, 2017).

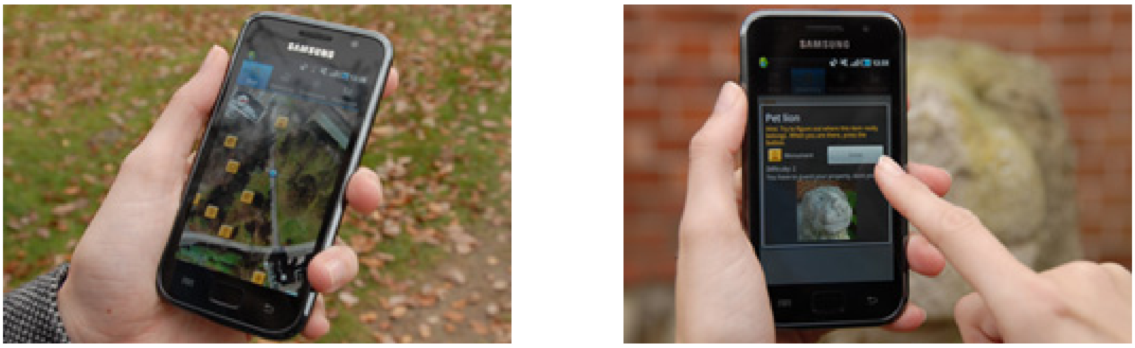
Even though Pokemon Go's primary purpose is entertainment, several studies conclude positives side-effects of playing the game. Meschtscherjakov et al. (2017) investigated the increase of physical workout of Pokemon Go players. The study revealed that " *participants walked on average 383 km actively playing for 14 weeks*". Players also stated that they not only used the app during their usual walks but also walked for longer distances, and deliberately took detours to proceed in the game. Ching-Yu Huang took a look at Pokemon Go regarding its possible use in computer science education. He recognized several aspects of the game which can be used to explain and discuss specific computer science fields in classrooms like human-computer interaction or character modeling (Huang, 2016).

The popularity of Pokemon Go results rather from the well-known franchise than the fact that itself is a particularly good or innovative game. However, the extent of augmented reality used in the game is very entertaining, and the possibility to catch all available Pokemon is a good long-time motivator. Pokemon Go uses popular or interesting sites for special in-game location like markets but does not provide any information about the physical location or object on its position and can therefore not be used for any educational purpose other than computer science as mentioned above.

Tidy City is a simple scavenger-hunt-like location-based game for Android devices developed as part of the German-French research project "TOTEM – Theories and Tools for Distributed Authoring of Mobile Mixed Reality Games". "*The goal of the project is to provide a flexible and powerful framework for the creation of mobile mixed reality games, ranging from experienced users to novices.*" (Wetzel, 2011). In Tidy City users are presented with a map of their surroundings upon which riddle-markers are placed as illustrated in Figure 2.6a. Each riddle consists of a name, a difficulty rating, a clue text and a clue image. These clues describe a hidden location in an area near the marker, and the player's task is to figure out the place and physically move to the site where the riddle can be solved by pressing a button. This process is shown in Figure 2.6b.

Such riddles cannot be copied into other cities or be automatically generated because of their tight connection to the physical location. Tidy City relies on users to create new riddles in different areas. Therefore a separate mobile app the Tidy City Scout was implemented to help users create puzzles by first taking pictures, capturing GPS-coordinates and storing them onto the Tidy City server. Afterwards, this stored data can be accessed via a web-interface, and a complete riddle can be created and published to be solved by other users. Tidy City is free to use for non-commercial purposes and was staged at several events in Germany, New Zealand, and Ireland (Wetzel, Blum, & Oppermann, 2012; Richard, Lisa, Feng, Leif, & Michael, 2011).

Tidy City and the Tidy City Scout are proper tools to create and solve single, independent location-find riddles. Though the puzzles are usually rather short, they can include information about the cite in question and have limited possibility of educating the users. However such riddles cannot be connected to each other to become a full scavenger-hunt and tell a story or give information about similar sites.



a) Tidy City map with marked riddles (TOTEM, 2011).

b) A riddle is ready for completion (TOTEM, 2011).

Figure 2.6: The Tidy City mobile application in action

2.4.3 Discussion

The mentioned location-based games show different approaches of how to use physical location as crucial game-element. Being able to capture a portal like in Ingress or finding hidden Geocaching-treasures by actually going to a physical place can be very motivating. Combined was the possibility to collect items or creatures like in Pokemon Go a whole new game experience can be created. The motivational boost of achieving a goal by going to a physical location and can also be used for educational purposes. Tidy City already has some possibilities to educate the users, but it was not the primary goal of the developers. The next section discusses how to integrate educational content into location-based games or create such games with a primary focus on education.

2.5 Location-based Learning Games (LBLG)

This section covers location-based learning games (LBLG). It describes how such a game is defined and mentions successful examples of this genre. Finally, it discusses these examples with the aim of identifying possible enhancement capabilities.

2.5.1 Overview

Location-based learning games (LBLG) use ideas and concepts described in the last section and use them specifically to create games in an educational context. Even though some of the above-mentioned application can be used for learning as well, this was not the primary intention of the game design. Location-based learning applications are designed to foster pervasive mobile learning that supports contextualized learning in non-formal learning situations (Avouris & Yiannoutsou, 2012). Location-based games have been proven to improve learning experiences by extending the education beyond traditional classrooms. Virtual layers of information can be added to physical items and the positive learning effects of exploration, cooperation and analysis can be utilized (Melero & HernÁndez-Leo, 2017).

2.5.2 Examples of Location-based Learning Games

The following section mentions some examples of how LBLGs can look like in different fields of education. These cases include research prototypes as well as fully released education games. Each sample is discussed concerning its effectiveness in the chosen field, its applicability in other educational areas and included motivational aspects where possible.

The game **Frequency 1550** developed by a Dutch ICT research foundation called Waag Society is a LBG with the aim of teaching pupils about medieval Amsterdam in a single school day. The players objective is to gain citizenship in the city of Amsterdam by acquiring 366 points or 'days of citizenship' which represent the medieval year-and-a-day rule, the period a citizen has to reside within the city walls to earn civil rights.

The pupils are split into teams of four or five pupils which will again be divided into two groups: one city-group which will be walking around Amsterdam using hand-held devices and one headquarter-group operating with computers in the main building. Both groups can communicate with each other and can see a map of medieval Amsterdam while only the headquarter group sees the map of present Amsterdam and indicators showing the city-teams current position. Therefore one essential task of the headquarter team is to give instructions to the city-team to help them navigate. The map of medieval Amsterdam is divided into 6 regions each having an own theme like labor, trade, and religion. As soon as the city-team enters one of these areas they automatically receive a message including a description of their task like collecting yarn for the holy-sisters from a different region. This requires to first navigate to the right area and afterward finding the exact hidden location. Sometimes pupils have to answer questions, take pictures or even participating in and filming a small role-play activity. During these activities, the headquarter-team searches through the Internet or other sources for additional information needed to fulfill the assignments. The accuracy of the given answered as well as the required time determine the amount of points the whole team gets for the completion of a task. After lunch the two groups switch places allowing each pupil to gain the full experience of the game. The winning team is determined at the end of the day by comparing the acquired points.

The evaluation after such gamedays showed a learning effect concerning knowledge of medieval Amsterdam by playing Frequency 1550. In comparison to pupils who learned about the same content in regular project-based lessons, Frequency 1550 players attained higher scores on knowledge tests about the subject. However, no significant differences in motivational effects for History in general or the Middle Ages, in particular, could be found (Huizenga, Admiraal, Akkerman, & ten Dam, 2009).

Frequency 1550 is specifically designed to teach about medieval Amsterdam and can therefore not be used to inform about other content without extensive modifications. The game features group-work, communication between two team-parts and riddle-solving as well as healthy competition between different teams. The tasks each side has to fulfill are quite diverse never to become boring. Because of this features Frequency 1550 is suspected to be very entertaining and motivation, however, these aspects were not explicitly evaluated.

ItoScramble is a location-based English vocabulary learning game developed by TANG et al. (2017) to help students improve their English vocabulary. The game concept is similar to Ingress. The players are divided into two teams, and their objective is to capture specific locations in the real world. To do so the participants have to answer 10 English vocabulary questions when they reach a place. Whether the site can be captured is depending on the accuracy of the answers and the time taken to solve the question.

During an evaluation-session, 63 first-year students of the university in Kyushu (Japan) played IntoScramble for an hour. Afterwards, their learning improvements have been measured by comparing a pre- and post-vocabulary test. TANG et al. (2017) stated that " *although the mean of the post-test (28.21) was slightly higher than that of the pre-test (27.76), there was no significant difference.*". Regarding the question, if students liked to play ItoScramble and were motivated to continue the team received better results and many suggestions how to improve the game (TANG et al., 2017).

This game focuses on evaluating what participants have learned beforehand rather than help them learn. This explains the somewhat poor performance shown during the evaluation, but it can help motivate people to increase their vocabulary to capture more locations. Even though the developers of ItoScramble focused on English vocabulary, the concept can be applied to various other domains as well. Summarized ItoScramble is a universal tool to motivate learners but does not educate on its own.

The LBLG **The Early Success of the Puppet Master Li Tian-Lu** is designed to be played in Taipei City, Taiwan, to learn of the famous Puppet Master Li Tian-Lu. "*The main goal of this game is to allow learners to experience in retrospect the disappeared historical scenes of puppet Master Li's early performances*". Groups of 3 students are required to visit historic sites and are provided with retrospective scenes, old pictures, narratives and questions within an augmented reality setting as seen in Figure 2.7. To answer the in-game questions, learners are encouraged to discuss the presented pieces of information within the groups or even consult the local population.

An experiment including 20 first-year students from National Taiwan Normal

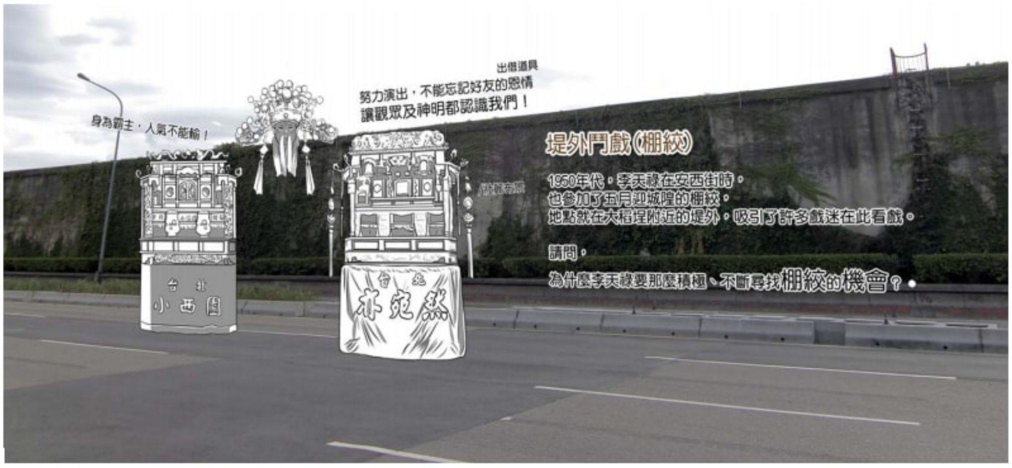


Figure 2.7: Puppet Master Li Tian-Lu: AR presentations of historic information (Liu & Wang, 2016)

University was conducted to evaluate the games educational performance. The participants were split into to groups where one uses the mobile game to learn about Li Tian-Lu history while the other group used traditional rote learning. Both groups were given two tests about the subject, one right after the learning process and the other one month after the experiment. While in the first test no significant difference in learning achievement between the two groups could be found the result of the delayed test showed *"that students who were assigned discovery strategy had better learning retention compared to those assigned rote strategy"* (Liu & Wang, 2016).

The use of augmented reality to recreate historical scenes is a very motivation approach in comparison to learning the same content from books or pictures. The evaluation shows that the participants remembered the content significantly longer than usual. This indicates that richly presented data is not only more motivating but also easier to remember. However, the game's area is limited to Taipei City and the subject of puppet Master Li and cannot easily be extended.

Pirker, Gütl, Weiner, and Garcia-Barrios (2015) present their **Location-based Mobile Application Creator (LMAC)** which allows the creation of individual scavenger hunts for children. This scavenger hunts can be executed on any smartphone within a standard web browser. The aim of the project was primarily to develop a framework which can easily be used by non-programmers to create engaging, mobile-aware applications. Users can create routes by adding markers on a map which are either "fixed routes" with a pre-set order or "undefined routes" where the player can decide the order of the execution.

"Each target point is linked to an activity which is triggered when the user of the mobile app is close to this point" (Pirker et al., 2015). The prototype includes two types of activities: (1) *Location information* like textual or media data and (2) *Location Interactivities* like multiple-choice or open-ended questions.

As the result of an initial field study including 8 to 10-year-old primary school children several improvement possibilities were found: Children at that age are seldom experienced in reading maps. Therefore, additional map-interface elements, like compass or direction guides would improve the design. Also, several security issues were discovered because participants were more focused on the mobile device than on roads and traffic (Pirker et al., 2015; Pirker, Gütl, Weiner, Garcia-Barrios, & Tomintz, 2014).

LMAC's design allows it to be used in several different educational fields as long as they have any connection to physical locations and objects. Geocaching and several other application have proven that digital scavenger hunts can be very motivating and the association of informational content like text and pictures with the actual physical places and objects helps users to remember facts more natural and more extended. The scavenger hunts have to be selected beforehand which means spontaneous looking for available scavenger hunts in the current area is not possible. Each scavenger hunt in LMAC is independent, and no gamification elements like statistics, badges or collectibles are included.

MILE[8] is a project funded by the Baden-Wuerttemberg State Ministry for Rural Areas and Consumer Protection[9] and aims to teach a deeper understanding of regionally and seasonally grown agricultural products and sustainable value chains in adolescents' everyday life.

The project consist of two software tools:

- The **MILE.designer** is a web-based content management system designed to help educators create location-based games. Such a game consists of a series of mini-games, so-called "Marbles", which can be such as multiple-choice questions, puzzles and sorting challenges for example. In total MILE supports 10 different Marble-types. The content of each Marbel is tightly connected to a physical location. Therefore, a geolocation is assigned to each Marbel using a map-editing tool. Within the definition of a Marbel educators can use written text as well as images, audio, and video.
- The **MILE.explorer** is an Android application on which a game created in the MILE.designer can be played. Games can be downloaded one-by-one from the official website and afterwards be played offline. The interface of the application features a road-map displaying the player's current position as well as all available Marbels. The players need to be physically at the location of the Marbel to start it. After a user has finished the task of the Marbel feedback is given, and the player earns points. The game ends when all Marbels have been solved.

MILE is designed for use as an extension to traditional classroom education, therefore, it is usually used by whole school-classes while their teachers monitor

[8]http://www.mile-bw.org/

[9]https://ec.europa.eu/growth/tools-databases/regional-innovation-monitor/organisation/ministry-rural-area-and-consumer-protection-baden-wuerttemberg

their progress. It is even possible for educators to write messages directly to a specific MILE.explorer instance to e.g., help their pupils if they got stuck on a task (Brosda, Bartsch, Oppermann, & Schaal, 2016).

Even tough MILE is designed for educating children about agriculture and value chains in the food industry it can easily be applied to various other fields as well. In comparison to the Location-based Mobile Application Creator discussed before MILE offers more different task types. However also in MILE the scavenger hunts have to be selected and downloaded beforehand to be played. Again no gamification elements or collectibles are included in the game.

2.5.3 Discussion

Different types of location-based learning applications were discussed in this section. Evaluations indicated that the richer information is presented and visualized the easier it is remembered. However, this usually means that an application is designed for a single purpose and can not easily be used in any other education domain. Several scavenger hunt applications that allow content creation for different fields exist but these scavenger hunts are usually treated independently. No gamification elements or collectibles were used to motivate users further after one hunt is completed. Furthermore, none of the presented applications allow to spontaneously execute a scavenger hunt in the nearby area without downloading it beforehand.

To motivate users to continue executing quests after finishing the first one the *Myth Hunter* application presented in this thesis combines digital scavenger hunts with a Collection-Card game. This offers a second game experience as well as collectibles as rewards and incentives.

2.6 Collection-Card Games (CCG)

This section describes the characteristics of a collection-card games, the different game aspects and the possible us of such games in an educational context. Popular examples of such games are mentioned and discussed afterwards.

2.6.1 Overview

In general Trading- or Collection-Card games (CCG) are card games which combine the three aspects of collection, creation and community in one game.

- **Collection:** A Trading-Card game usually consists of an, theoretically, unlimited amount of cards. When players want to start playing such a game they acquire a so-called "Starter Pack" including enough cards to play the game but by far not all available cards. A big side activity of such games is to buy, win or trade additional cards. A vital part for most CCGs are so-called "Booster-Deck" which contain a few random cards and can be bought or won. The more boosters players acquire, the higher are the chances to receive powerful and rare cards.

- **Creation:** Using this cards players can create decks. A deck is a subset of the player's cards which is used in a single game against an opponent. The amount of cards one deck can contain is usually limited therefore players have to choose with which cards to play prior to the actual game. Players have to think about which kind of strategy they want to follow in the game and build their decks accordingly.
- **Community:** "*Most CCGs involve conflict of some kind. So, they require two or more players at a time. While the players may be enemies during play, they may be friends, mentors, or collaborators in the broader context of the play community*". As Adinolf and Turkay (2011) stated there are more communication aspects to a Collection-Card game than simply while playing against each other. Players often meet up to compare their decks, gain advice from experienced players or just talk about the last tournament they played in.

One major point of criticism often voiced against CCG is that by putting more money into the game players can effectively improve their chances of victory (meople's magazine, 2011; Adinolf & Turkay, 2011).

Trading-card games are often used as reward systems. Chen, Kuo, Chang, and Heh (2009) introduced the concept of collection-card games as a reward system in classrooms. Pupils collect cards to play while teachers can give away boosters or unique cards to children for right answers or participation (Chen, Chang, Kuoc, & Heha, 2016). Turkay, Adinolf, and Tirthali (2012) argue that using CCGs in education not only helps to motivate pupils but encourages them to develop various skills, including analytical thinking, empathy, social manipulation, iterative design, and communication.

2.6.2 Examples of Collection-Card Games

In the following section, famous examples of CCG and their game mechanics are presented. Afterwards, similarities and differences in these games are discussed to extract relevant game elements for the *Myth Hunter* card game.

Magic: the Gathering[10] is considered the first CCG and is still one of the most popular games of this genre. It was created by mathematics professor Richard Channing Garfield and first published by Wizards of the Coast[11] in 1993. It quickly became "*one of the big master strokes in modern gaming*" (meople's magazine, 2011). In 2011 the game consisted of close to 12.000 different cards. According to Wizards of the Coast approximately 12 million players in over 70 countries are still playing Magic the Gathering today.

[10] http://magic.wizards.com/
[11] http://company.wizards.com/

Figure 2.8: A typical game of Magic: The Gathering (of the Coast, 2010)

In a game of Magic: The Gathering players have the role of powerful mages, so-called "Planeswalkers" who try to defeat their opponents armed with a deck of magic cards representing lands, creatures, and spells. Lands produce "Mana" which is the primary resource in the game. Mana is needed to cast spells or summon creatures. The creature cards represent the wide variety of beings in the world of Magic. These cards are an essential part of each deck and are used to attack the enemy or block incoming attacks. Spell cards have a huge variety of effects from preventing incoming attacks healing damage, to killing the enemy's creatures. The final goal is to get past the enemy's monsters and deal damage to reduce the life points to zero. Figure 2.8 shows a typical game of Magic: The Gathering (Trammell, 2010; meople's magazine, 2011).

Yu-Gi-Oh! Trading Card Game[12] is based on the fictional game called "Duel Monsters" of the manga and animation series Yu-Gi-Oh! created by Kazuki Takahashi. Yu-Gi-Oh! is on the market since 1999 and with 25 billion cards sold worldwide the game is holder of the Guiness World Record for best-selling trading card game since 2009 (Konami, 2009).

A Yu-Gi-Oh deck consists of Monster, Spell, and Trap-cards. The use of monster and spells is similar to Magic: the Gathering except no mana is needed to summon a monster or cast a spell. Instead, the use of spell-cards per turn is unlimited while only one monster can be summoned each turn. Additionally, monster-cards have a level which determines how these cards can be played: Level 1-4 cards can be played immediately while other monsters have to be sacrificed to summon stronger monsters. Trap-cards can be placed face-down on the board and are triggered by specific events like the summon of an enemy creature. The effect is usually negative for the opponent (Konami, 1996).

[12]http://www.yugioh-card.com/

Figure 2.9: Screenshot of the game Hearthstone (Zopf, 2015)

Hearthstone[13] is a free-to-play digital CCG developed by Blizzard Entertainment[14] and was released on March 11, 2014. Six months later in September 2014, Blizzard announced that there are over 20 million registered accounts worldwide (Zopf, 2015). In May 2016 the 50 million accounts mark was reached making it the most popular online CCG (Bursztein, 2016).

In Hearthstone two players compete online against each other in an alternating manner. In contrast to offline CCG like Magic: The Gathering each turn is strictly limited to 30 seconds. Additionally to the choice of which cards to put in a deck in Hearthstone players have to choose a hero as well. Depending on the hero different cards are available during deck creation, and special abilities can be used during the game. Similar to Magic: The Gathering Mana is needed to play cards but no land-cards are required to gain Mana. Instead, each player receives a certain amount of Mana at the beginning of each turn while unused mana is discarded at the end of the turn. In the first round, both players have only one mana, but the amount increases each round till the maximum of ten is reached.

A Hearthstone deck generally only consists of Minions and Spell-cards but depending on the chosen hero special cards like traps or weapons can be added. In Hearthstone many minions have unique lead or combo effects which can affect other minions or the whole game board. Figure 2.9 shows the game board and the most essential components (Goes et al., 2016; Zopf, 2015).

[13]http://eu.battle.net/hearthstone
[14]http://eu.blizzard.com/de-de/

2.6.3 Discussion

Many other CCG other than the three mentioned exist, but these are the most popular and sufficient enough to extract relevant game elements for *Myth Hunter*. All of the mentioned CCGs have in common that there exist different types of cards. At least some kind of monster and spell cards are present in each game but other types like traps or land cards do not appear in each of the examples. The total amount of cards available is usually quite big and also the number of cards a player can own is unlimited. However, in order to play players must choose a subset of this cards as their current deck. Also, a card lead limitation is part of every rule set, either a strict maximum amount of cards that can be played each turn like in Yu-Gi-Oh! or a limited resource (e.g. Mana) is needed to lead cards. Other strategic elements like heroes in Hearthstone can be added to increase the possible game strategies even more but are not strictly necessary.

Because the CGG within *Myth Hunter* is only an additional game element and not the primary focus, only a small subset of the mentioned game mechanics apart from the strictly necessary ones will be implemented. However, the users will not play against each other but against the application, and therefore an artificial intelligence (AI) algorithm is needed which is strong enough to keep players entertained. In the next section possible AI algorithms and their applicability for CCG are discussed.

2.7 Artificial Intelligence AI Algorithms for Collection Card Games

Collection-Card games (CCG) are usually played as a two-player game. In such games, players move respectively play their cards in turns which can be modeled in a structure known as game tree. Each node in a game tree represents the state of a game; for example, in CCGs this includes the game board with all currently played cards as well each players deck and dead cards (Hong, Huang, & Lin, 1998).

Several algorithms operating on such game trees to find optimal or near-optimal moves have been introduced over the years. This section gives an overview of popular algorithms and their strengths and weaknesses when applied to a CCG.

2.7.1 Minimaxing

Minimaxing is one of the oldest but still widely used algorithms in game theory. It is most often used in two-player perfect information games such as chess. Perfect information, in this case, means, that both players can see all the information of the current game state.

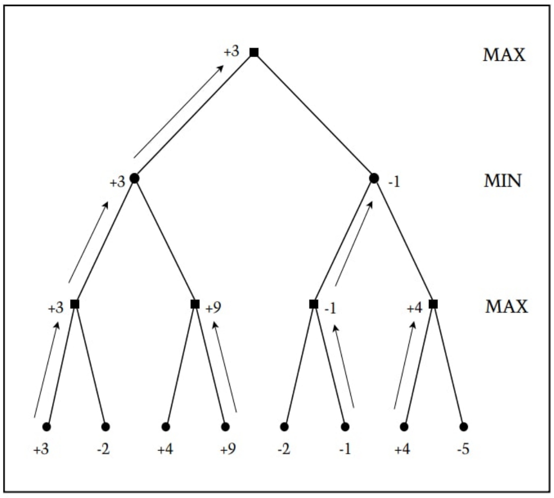


Figure 2.10: An Example of Minimaxing (Kaindl, 1988)

In general minimaxing can be divided into three phases: (1) building up a game tree starting at the current game state till a pre-defined depth, (2) calculating the win-likelihood of the player in question for each leaf and (3) propagating the likelihood back to the root to choose a move. To calculate the win-likelihood of a game state heuristics for the particular game have to be defined beforehand. Let the player move be called MAX and the enemy MIN then back-propagating is done as follows: At each node in the tree one possibility is chosen and its likelihood is copied to this node. As can be seen in Figure 2.10, if the node represents a turn of MAX the highest value is chosen wheres at a MIN turn the lowest value will be selected. This simulates that at each turn the player to move selects the optimal move while ensuring the maximum outcome for MAX at the end of the look-ahead tree (Kaindl, 1988; Campbell & Marsland, 1983).

Given accurate heuristics, this simple algorithm is sufficient for many games of perfect information not including any chance elements, e.g., rolling dice. Ballard (1983) introduced the *-Minimax Algorithm which can cope with a certain amount of certainty by including chance-nodes into the game tree which obtain the average likelihood of their successors when back-propagating.

Depending on the game the number of possible moves or chance outcomes can be quite extensive which limits the depth the minimaxing algorithm can search in a limited amount of time. Reducing the search depth also reduces the quality of the chosen move. Therefore, this algorithm is only feasible for perfect information games with a limited amount of choices per state.

2.7.2 Alpha-Beta Pruning

Alpha-Beta Pruning is an improvement of Minimaxing by reducing the amount leafs to be checked by cutting off tree parts while still ensuring the same result.

Minimaxing executes the three phases of building a game tree, evaluation

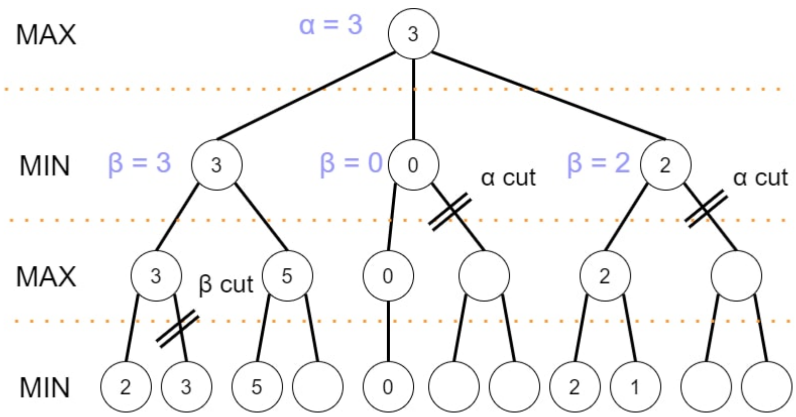


Figure 2.11: An Example of Alpha Beta Pruning (Jez9999, 2007)

the leafs and back-propagating one after another for the whole tree. Alpha-Beta Pruning, however, uses depth-first search till it reaches a termination state (e.g., a leaf of the predefined depth), calculates the likelihood and directly propagates it back to its predecessor before creating another leaf, evaluating it and updating the predecessor value according to the Minimaxing rules. Additionally, it stores in each node the best, already examined option for MAX on a path to the root(Alpha) and the worst for MIN(Beta). If the algorithm reaches a MAX node with a current value bigger than Beta it can cut off all unexplored successors because MIN will never choose that path. Accordingly, successors of a MIN node can be cut off if the current value is lower than Alpha. An Example of Alpha-Beta Pruning can be seen in Figure 2.11 (Knuth & Moore, 1975; Campbell & Marsland, 1983).

Using Alpha-Beta Pruning, the performance of an AI for perfect information games with a high amount of choices can be significantly improved in comparison to pure Minimaxing. However, both approaches rely on accurate heuristics to evaluate a game state which are hard to come by for some games like Go (Enzenberger, 2004; Bouzy & Cazenave, 2001). Additionally, most trading card games include hidden information, like the opponents deck or hand cards, with which both Minimaxing and Alpha-Beta Pruning cannot cope efficiently.

2.7.3 Monte Carlo Tree Search (MCTS)

Monte Carlo Simulation is "*a numerical experimentation technique to obtain the statistics of the output variables of a system computational model, given the statistics of the input variables*" (Mahadevan, 1997). The random input variables are sampled in each experiment based on their distributions while the output variables are calculated using the model. After a given number of operations are carried out the probability of each output variable can be calculated.

Multi-Armed Bandit Problem: "*Bandit problems are a well-known class of sequential decision problems, in which one needs to choose amongst K actions (e.g. the K arms* *of a multi-armed bandit slot machine) in order to maximise the cumulative reward by consistently taking the optimal action"* (Browne et al., 2012). Which action to choose is difficult because the underlying reward probabilities are unknown and can therefore only be estimated using past observations. This leads to the so-called "exploitation-exploration dilemma" where one must balance the two opposing strategies of (1) exploitation of the currently (believed to be) most promising action and (2) exploration of alternatives which no seem inferior but may later turn out to be superior in the long run.

The most widely used algorithm to solve this problem is the "Upper Confidence Bounds" algorithm USB1 proposed by Auer, Cesa-Bianchi, and Fischer (2002) which dictates to play the arm j that maximises:

$$USB1 = \overline{X_j} + C * \sqrt{\frac{ln(n)}{n_j}} \tag{2.1}$$

where $\overline{X_j}$ is simply the average reward of j, n_j is the number of times j was chosen in the past and n is the total number tries so far.C is a constant used to alter the exploitation verses exploration rate (Browne et al., 2012).

Abramson (1990) was the first to propose the use of Monte Carlo simulations to evaluate game states. However, this approach was widely ignored till the introduction of Monte Carlo Tree Search (MCTS) algorithms in 2006 (Kocsis & Szepesvári, 2006; Coulom, 2007). This family algorithms uses Monte Carlo simulations in combination with the concepts of exploitation and exploration to build a search tree which guides simulations (Whitehouse, 2014).

The basic concept behind MCTS algorithms is rather simple as illustrated in Figure 2.12 and one iteration can be divided into four steps (Browne et al., 2012; Zopf, 2015):

- **Selection:** Starting at the root node of the tree, a child node is recursively selected using a child selection policy. This policy can be seen as a variation of the multi-armed bandit problem and is usually solved using UCB1 (Auer et al., 2002).
- **Expansion:** At least one child node is added to the tree representing an available action.
- **Simulation:** A simulation is run from the new node(s) choosing random actions till one of the players wins.
- **Backpropagation:** The result of the simulation added to the statistics (wins and visits) of each visited note.

After a predefined amount of iterations or any other interruption criteria is met, the algorithm chooses one of the root's children based on their statistics. According to Schadd (2009) one of four criteria can be chosen to select a winning action:

1. **Max child:** The root-child with the highest winning rate.

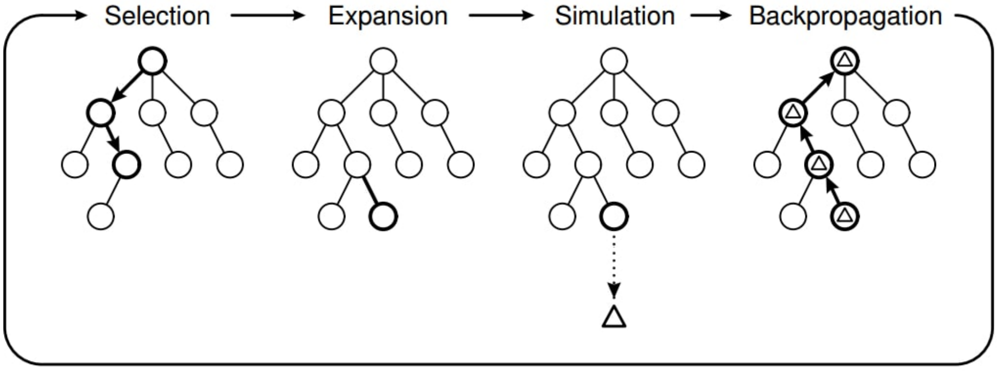


Figure 2.12: One iteration of the general MCTS approach (Browne et al., 2012)

2. **Robust child:** The most often visited root-child.
3. **Max-Robust child:** The child with both the highest winning rate and the highest visit count. If no such child exists at the moment continue MCTS till the requirements are met.
4. **Secure child:** The child which maximizes a lower confidence bound.

One huge benefit of MCTS over algorithms based on Minimaxing is that no domain-specific knowledge (e.g., heuristics) is required (but can be beneficial) beyond the rules of the game. Therefore MCTS can easily be applied to any domain that can be modeled using a decision tree. "*MCTS is also an iterative algorithm which means that all values are always up-to-date following every iteration of the algorithm. In practice this means the MCTS algorithm is able to return an action at effectively any moment in time since each iteration is usually short compared to the overall run time of the algorithm*" (Whitehouse, 2014). Traditional MCTS is, however, only applicable for games of perfect information and not suitable for games with hidden information. Ignoring the hidden information (e.g. cheating) often leads to unsatisfying experience for the gamer and should therefore be avoided.

2.7.4 Information Set Monte Carlo Tree Search (ISMCTS)

The **Subset Armed Bandit Problem** is similar to the Multi-Armed Bandit Problem described in section 2.7.3 except not all arms are available at all times. In each iteration, different arms are available and can be chosen from.

Two approaches are possible to solve this problem:

- Treat each subset independently by creating statistics for each subset and arm and apply the regular UCB1 algorithm.
- Use only one set of statistics but add a value representing the times an arm was available in the past. The modified UCB1 algorithm for Subset Armed Bandits is defined as:

$$USB1 = \overline{X_j} + C * \sqrt{\frac{ln(\overline{T_j}(n))}{n_j}} \tag{2.2}$$

where $\overline{T_j}(n)$ is the number of times arm j was available in previous trials. The other parts of the formula are the same as in the regular USB1. This ensures that an arm which is only seldom available is not over explored which would be the case with standard UCB1.

In order to overcome the restrictions of MCTS and apply it to games with hidden information Cowling, Powley, and Whitehouse (2012) presented a family of algorithms called Information Set Monte Carlo Tree Search (ISMCTS). In this approach nodes in the game tree represent information sets rather than actual states. The conventional MCTS algorithm is changed in the following way:

0. **Determinization:** All hidden information is randomized in the root game state. After this step, all hidden variables have defined, but random values.
1. **Selection:** As in regular MCTS, but only children that are compatible with the determinized state may be chosen. Also instead of regular UCB1 the modified version for the Subset Armed Bandit Problem is used to choose a child.
2. **Expansion:** Only children compatible with the determinized state can be added to the tree.
3. **Simulation:** Excatly the same as in regular MCTS.
4. **Backpropagation:** Excatly the same as in regular MCTS.

This results in nodes having several children, but only a subset is available for a given state depending on the initial determinization. However, one child can be available in several different determinizations which allows the algorithm to reuse information in upcoming iterations (Edward Powley & Cowling, 2013; Whitehouse, 2014; Cowling et al., 2012).

Depending on the amount of hidden information and their complexity the branching factor of the game tree can get very high. Therefore a certain minimal amount of iterations is needed for ISMCTS to work properly. To reduce computation time Sephton, Cowling, Powley, Whitehouse, and Slaven (2014) introduced three approaches to allow ISMCTS to be executed parallelized on multiple cores:

- **Root Parallelization:** The initial game state is cloned, and each agent runs ISMCTS on its own tree. After an interruption criteria is met, the main thread merges the trees and selects the best move.
- **Tree Parallelization:** A single tree is shared among all agents and locks ensure that *"no two agents attempted to write to the same memory at the same time, or read from memory that was being altered"*.
- **Leaf Parallelization:** A single "parent" agent maintains a single tree and hands the simulation step to child agents. Child agents are created until a predefined amount is reached and afterwards reused.

Sephton et al. (2014) applied all three approaches on the card game *Lord of War* [15]. *"Of all approaches attempted, root Parallelization across high numbers of agents was*

[15] https://www.kickstarter.com/projects/388956994/lords-of-war-fantasy-battles

the most time efficient approach, although the speed increase became mostly negligible when adding more than 4-5 agents" (Sephton et al., 2014).

2.7.5 Discussion

Several AI algorithms and their strengths respectively weaknesses when applied to Collection-Card games where discussed. Of these the Information Set Monte Carlo Tree Search (ISMCTS) algorithm was detected as most efficient and promising for the *Myth Hunter* card game. To further strengthen the algorithm and using the available computation power of modern smartphones three approaches to allow parallel execution of the ISMCTS where mentioned. Evaluation of all three methods showed that tree parallelization is the most efficient and will, therefore, be implemented in *Myth Hunter*.

2.8 Summary

Motivation has always been a critical element in education. If people are not motivated to learn about a specific subject, it is tough to teach them anything. Games, on the other hand, have proven to be very motivating. Already in early childhood playing and learning are very connected, so it makes sense to combine games and educational content also for older children and even adults. Nowadays especially computer and mobile games are very popular therefore more and more applications with the aim of teaching in a playfully way are developed.

This chapter has discussed several software approaches of how to interweave learning and playing. Educational software with gamification elements as well as dedicated learning-games were mentioned focusing on mobile-location aware applications. Approaches like Tidy City or MILE using digital scavenger hunts to guide users through a specific area have proven to be very motivating. Furthermore, they allow the connection of physical places or objects with educational content. These approaches, however, lack special incentives for users to continue solving new scavenger hunts. Also, the variety of tasks and game elements is very limited

To overcome this issues collectibles as reward system have been discussed. Especially Collecting-Card games (CCG) were mentioned because they not only work as incentives but also allow to add a completely different game experience to a location-based game. Minimal requirements for such a CCG have been identified and include:

- monster cards
- spell cards
- great amount of cards in total
- subset of these cards for one game (deck)
- lead limitation

Finally, several artificial intelligence algorithms (AI) and their applicability for such Collecting-Card games were discussed. Such an algorithm must be able to cope with uncertainty as well as hidden information. Also, the number of possibilities in one turn can be quite extensive and must be taken into account when implementing such an AI. The Information Set Monte Carlo Tree Search Algorithm was found to be most promising and will, therefore, be implemented in the *Myth Hunter* application presented in this thesis. The next chapter discusses further requirements to implement *Myth Hunter*, the location-based scavenger-hunt game with an interwoven Collection-Card game and the aim of educating children.

3 Requirements and Design

This chapter covers the requirements the implementation of *Myth Hunter* has to fulfill. Not all of these requirements were known or formalized before the implementation and evaluation of the first prototype. After the requirement section, the architectural design of the application is presented by taking a deep insight look at the quest execution process.

3.1 Motivation

In the previous chapter different approaches how to interweave learning with playing were discussed. Especially location-based games have proven to have high potential to be used in educational context while still be highly motivating. Also, collection-card games are perceived as highly motivating allowing the players to collect cards and strengthen their game over an extended period. *Myth Hunter* connects these two types of games in a single learning-application to create a long-term motivating learning-while-playing experience. This chapter will discuss which requirements need to be fulfilled to develop such an application. This discussion is split into functional and non-functional requirement analysis while the functional requirements are further categorized into the three main aspects of *Myth Hunter*: scavenger hunt, collection-card game, and further gamification-elements.

3.2 Main Concept

One of the main ideas of Myth Hunter is to be a complete, fun-to-play mobile game including learning opportunities instead of being another learning-application with added gamification features. Therefore the application consists of two parts tightly connected to each other. One part is a collection-card game similar to the ones mentioned in section 2.6 where the player can collect a various amount of different cards, create customized decks and use this decks to fight against opponents controlled by an artificial intelligence. Wrapped around this card game location-based quests are created which present an entertaining story and allow the player to visit the real locations of this tale. Entangled in this stories is content the player should memorize like historical facts about the visited places, the story of a statue or any other information which can be linked to the tail itself or a visited physical location. As part of these quests, the players have to fulfill several tasks like answering questions, finding hidden places or fighting against an opponent using their decks. Upon completion of a quest, the players receive a reward which, in turn, helps to improve their decks. Apart from presenting informational content and linking it to activities and actual locations the usages of this application also enhances the user's ability to navigate through unfamiliar places by use of a map. Due to the need of walking from one location of a quest to the next also a slight improvement of the user's stamina and overall health can be a positive side effect of the use of *Myth Hunter*.

3.3 Functional Requirements

This section will describe in detail which functional requirements need to be met by the Myth Hunter Application to provide the desired experience of fun while learning for a user.

3.3.1 Scavenger Hunt

The following section will describe the requirements of the scavenger hunt game. It will explain basic user interface functionalities as well as different types of stations which can appear within a scavenger hunt.

R-1 Display of a Road-Map and the Current Location: One crucial task a player has to fulfill to play Myth Hunter is to find the places of a quest in the real world. Therefore it is necessary for the Myth Hunter Application to feature a road-map of the user's surroundings which indicates the player's current position as well as the positions of the quest targets. This map must be available in two modes:

- **R-1.1 Orientation-Following:** The map automatically follows the user's orientation which means that the map rotates in such a way that the cardinal direction the user looks towards is shown going upwards on the screen. This makes it very intuitive for inexperienced players to reach a target on the map because if, for example, the user looks at the map and sees a mark to the right of the current position the player just has to turn right in the real word to go towards it regardless of the cardinal direction.
- **R-1.2 Static:** The second map mode is the same as if the user would be equipped with a traditional paper map and a compass. The map doesn't rotate at all but indicates which direction is north. At the same time, a compass shows in real time the cardinal direction the user is heading towards. By aligning the north-indicator of the map and the needle of the compass the client device can be oriented according to the physical world.

Onto the map different markers will be placed: new quests, stations of already started quests and random enemies. To not overfill the map with markers always only one type of quest markers is displayed, either new ones or active, already started quests. The default option upon launching the application is to show new quests.

R-2 Quest-Marker and Quest-Description: Quest markers have to be placed at their start locations on the map representing new, not yet started quests. Quests are categorized in 3 difficulty levels (easy, medium and hard) which should be directly distinguishable by just looking at the markers. Regardless of the player's actual position, a tap on such a marker opens a window showing a description of the quest including the possible reward for completing this quest. The user has now the option to leave the description and go back to the map or start this quest. Upon beginning a quest the map automatically switches into the "active quests" mode hiding all not-started quests and displaying the first station marker of the started quest.

R-3 Station-Marker and Task Execution The second type of markers shown on the map are station markers representing tasks a user has to fulfill in order to complete a quest. Contrary to quest markers a player has to be at the location of a station marker to open it's description and fulfill the task. Right after starting a quest always only one station marker of type "Information" is visible. This marker is placed at the same location as the quest marker was placed. The marker's description, which can be opened upon tapping on it if the user is near it's position, leads into the story of the quest and describes what the user has to do next. From this point forward one or more station marker can be visible at the same time on the map. A quest can consist of several different types of station markers:

- **R-3.1 Information Marker:** An information marker represents, as the name indicates, some information to the player. This information can include simple text as well as pictures, video clips or hyperlinks to web pages. The content can vary from fictional story parts to hard facts a player should memorize. The user's task to complete this kind of station is to merely reach the location of the marker and absorb the given information. This is the simplest type of station marker, and everything mentioned above is true for all upcoming station markers as well.
- **R-3.2 Quiz Marker:** In addition to the information marker's functionality, a quiz marker can include questions the player has to answer correctly in order to complete the task. This questions can be raised in the form of single or multiple choice quizzes as well as free text questionnaires like fill-in-the-blank texts. After committing the given answers, they are checked and the results presented in an understandable form. The task is not complete until all questions are answered correctly. The number of wrong attempts is stored for reward calculation in the background but is not displayed to the user to not demotivate them. After all questions are answered correctly a second screen is displayed to the player representing

information about the next steps of the quest.
- **R-3.3 Find Marker:** The find marker consists of two markers at different locations. One of the markers is visible on the map and describes the place of the second, invisible one. This description is supposed to be a riddle giving hints of how to find the hidden location without specifying the exact position. This information can be reviewed as often as desired. The second marker only becomes visible at the map when the user is nearby. After tapping on the now visible marker, viewing the presented information about the upcoming task and acknowledging it the task is completed. If the user can not find the location by following the hints, there exists the possibility of displaying an indicator showing if the user is going in the correct direction and how far away the target still is in regard to the start position exists.
- **R-3.4 Fight Marker:** At the fight marker's location, the player finds an opponent which must be defeated. This fight is executed as card game like described in section 3.3.2. The player can restart the match as often as desired but similar to giving wrong answers in a quiz marker the number of restarts is taken into account when calculating the quest reward. The user also can skip the fight entirely which will reduce the possible reward. This consequence, however, will be presented to the user beforehand. Rewards are explained later in this chapter. After winning or skipping the fight information about the aftermath of the fight and the upcoming tasks will be presented.

R-4 Quest Structure: As mentioned before a quest, consists of several stations a player has to complete. How these locations are connected to each other can be very different depending on the story of the quest. Every quest has one start point, but afterwards, the user might have to make decisions. These decisions can range from merely choosing which task to complete first if multiple stations have to be completed to move on, to complicated, story changing choices where a user must decide which task to execute. This means markers might disappear on the map after completing a different one. Even different endings for one quest are possible.

The information of how stations are connected must be presented within the previous markers or the active markers itself. Past markers can be reviewed by either adjusting a filter on the map or taking a look into the quest-log which is described in section 3.3.1.

R-5 Quest-Log: To review already finished stations of a quest the user can take a look into the quest-log. It lists all the started quests and their tasks. By taping on a complete task, the user can retake a look at its information. Active tasks can only be reviewed if the player has already visited their locations.

Usually only the last started quest is marked as active, and its markers are visible on the map. This setting helps the player to focus on one quest rather than mixing up parts of different quests. In the quest-log, however, the player can change which quests are shown on the map and can directly jump to a markers location on the map.

3.3.2 Card Game

This section covers all functional requirements regarding the collection-card game within Myth Hunter. It also explains the game elements and rules. The collection-card game is a significant part of the Myth Hunter Application. Two players, or rather the player and an artificial intelligence, play against each other in a turn-based card game. Each player starts with 30 life points and 3 cards on hand. The goal is to bring the enemies life points down to zero while protecting the own. To this end, the players need to draw cards from their decks, lead them onto a playing-board and attack the enemy. Each turn starts by drawing one additional card from the deck. If the deck becomes empty, all destroyed cards are shuffled and put into the deck again. To lead a specific card the player needs a certain amount of power indicated by stars on the card. This power is restored every turn and increases each turn by two starting with two power in the first turn up to a maximum of eight. Therefore the most potent cards can only be played later in a game. Additionally, the player can choose to sacrifice up to 6 life points each turn to gain additional power at an exchange rate of two life points for one power. Each deck can hold twenty to thirty cards consisting of Monster- and Spell-Cards.

R-6 Monster-Cards: As the name indicates this kind of cards represent some kind of monster, minion or any kind of person in the broadest sense. As illustrated in Figure 3.1 a Monster-Card consists of several attributes:

- **R-6.1 Name:** Each card has a unique name by which it can be recognized and remembered by a player.
- **R-6.2 Picture:** A visualization of the monster further improves the recognizability of a card.
- **R-6.3 Power-Stars:** The amount of stars on a card indicates how much power is needed by a player to lead this card onto the board. The more powerful a monster is the more stars are required. Table 3.1 shows how the star-amount is calculated.
- **R-6.4 Attack Value:** This number indicates how much damage this monster can inflict by attacking an opponent which can be either another monster or the enemy player.
- **R-6.5 Life Points:** This represents the durability of a monster or in other words the amount of damage a monster can take before it dies and has to be removed from the board.

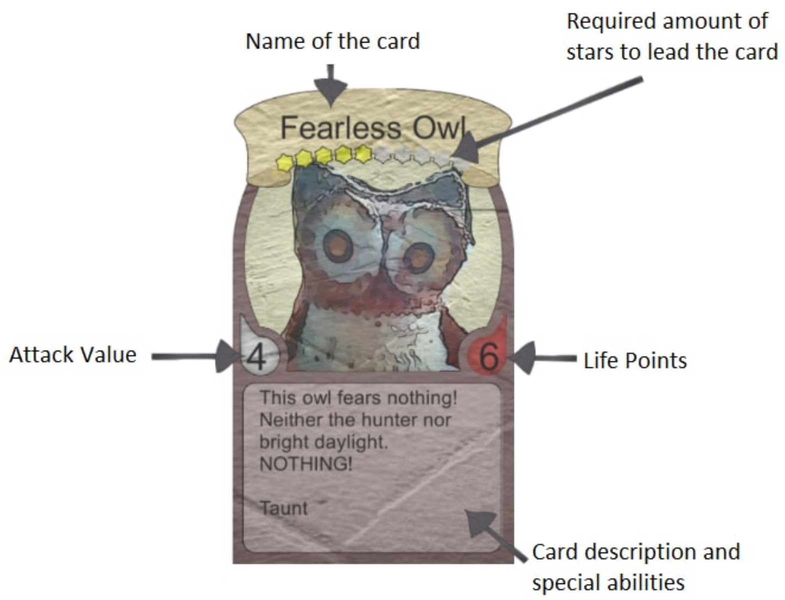


Figure 3.1: An example Monster-Card of the game Myth Hunter

- **R-6.6 Description:** The card description includes two kinds of information. The first is a short characterization of the monster itself, for example, a background-story. It has no impact on the game itself but adds some information of the character.
- **R-6.7 Abilities:** The second part of the description shows which special abilities a monster can use in the game. There are to types of skills a monster can have: Passive abilities need no activation and act always. Active skills, on the other hand, need to be activated by the player and cost one Life Point of the acting monster per activation. Usually, also the amount of possible activations is limited like shown in Table 3.1.
  These abilities are:
  - **R-6.7.1 Taunt:** This is a passive ability which means no activation is needed. If a monster has the taunt ability no other friendly monster without taunt nor the player owing this monster can be attacked while this monster remains on the board.
  - **R-6.7.2 Sleep:** This active ability can be used to attack an enemy monster but instead of inflicting damage it lulls the opponent into sleep for 1 round making it unable to attack. Also, the Taunt ability of a sleeping monster is blocked till it wakes up again.
  - **R-6.7.3 Confuse** The active ability Confuse attacks an enemy monster and confuses it for two rounds which means that if this monster attacks there is a 50% chance that it hits itself instead of the aimed target.
  - **R-6.7.4 Weakness** is an active ability which can be used to attack an

enemy monster and halve its attack value. This reduction of attack power is permanent.
– **R-6.7.5 Damage over Time (DoT)** A monster can attack an enemy (monster as well as player) by adding a DoT. It inflicts six damage split equally over three rounds.

Attribute / Action	Amount / Usages	Stars
Life Points	1	0.5
Attack Value	1	0.5
Taunt	-	2
Sleep	1	2
Sleep	2	3
Sleep	3	4
Confuse	1	2
Confuse	2	3
Confuse	3	4
Weaken	1	2
Weaken	2	3
Weaken	3	4
Damage Over Time	1	2
Damage Over Time	2	3
Damage Over Time	3	4

Table 3.1: Monster-Attributes to Stars Rate

Except in the very first turn of a game, a monster can be led to the board and be used in the same turn. A led monster stays on the board till it is defeated or till the end of the game. At any given time only a maximum of five monsters can be placed at the board simultaneously. A player cannot remove one of its own monsters.

R-7 Spell-Cards: This type of card can represent a tool, an item or some kind of magic. Unlike Monster-Cards Spell-Cards can only be used once and do not stay on the board after leading. Each Spell-Card can just have one effect. A list of the available effects and their costs can be seen in Table 3.2. The effect of the card is executed immediately upon leading, and afterwards the card gets destroyed. As can be seen in Figure 3.2 a Spell-Card has only a few attributes:

- **R-7.1 Name:** The name of the Spell-Card by which the player can recognize it.
- **R-7.2 Picture:** A visualization of the Spell-Card's effect further improves the recognizability and understandability of the card.

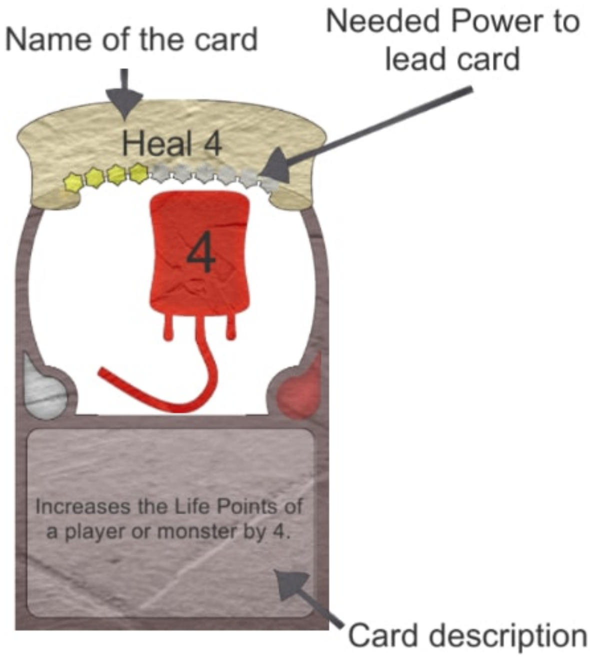


Figure 3.2: An example Spell-Card of the game Myth Hunter

- **R-7.3 Power-Stars:** The amount of stars on a card indicates how much power is needed by a player to lead this card onto the board. The more powerful a spell is the more stars are required. Table 3.2 shows how the star-amount is calculated.
- **R-7.4 Description:** The description of a Spell-Card explains the effect of the card and can include some trivia about the card respectively the item it represents.

R-8 Deck Editor: A card deck within Myth Hunter consists of a user given name and all the cards within it. To customize one's deck and create new ones a Deck Editor needs to be included in the Myth Hunter Application. It allows players to quickly add cards from their card pool to the deck and vice versa. A single card of the player's card pool can be used in several decks but only once within a single deck. Multiple usages of the same card within a deck is only allowed if the player possess more than one instance of this card. A deck can be used in a fight if it consists of minimum 20 and maximum 30 cards. It is essential that a player can always see the amount of cards included in the deck without needing to count them.

3.3.3 Gamification

Additionally to the card- and scavenger hunt game *Myth Hunter* includes several gamification features like rewards and statistics to motivate the player further. These features are explained in this section in further detail.

Effect	Amount	Stars
Immediate Damage	1	2
Immediate Damage	2	4
Immediate Damage	3	6
Immediate Damage	4	8
Immediate Damage	5	9
Immediate Damage	6	10
Damage of Time	6 damage over 2 rounds	5
Damage of Time	6 damage over 3 rounds	4
Damage of Time	9 damage over 3 rounds	6
Reduce Attack Value	x 0.5	5
Reduce Attack Value	-2	4
Reduce Attack Value	-4	8
Increase Attack Value	x 1.5	5
Increase Attack Value	x 2	10
Increase Attack Value	+2	4
Increase Attack Value	+4	8
Draw cards	2	3
Draw cards	4	8
Remove all Effects	-	3
Increase Life Points	1	1
Increase Life Points	2	2
Increase Life Points	3	3
Increase Life Points	4	4
Increase Life Points	5	5
Increase Life Points	6	6
Increase Life Points	7	7
Increase Life Points	8	8
Increase Life Points	9	9
Increase Life Points	10	10

Table 3.2: Spell-Card Effects to Stars Rate

R-9 Rewards: As mentioned in previews sections a player can gain rewards by completing quests. The amount and type of such a prize depends on the quest itself as well as the player's performance while executing the quest. In general, a reward can consist of up to 3 parts:

- **R-9.1 Booster-Decks:** Regardless of the quest and the player's performance the reward always includes one Booster-Deck. A Booster-Deck is a collection of three cards randomly picked out off all the available cards of the game. This can consist of new cards as well as instances of cards the player already owns. It is even possible but rather unlikely that a Booster-Deck contains the same card multiple times. The user can afterwards use these cards to strengthen its own decks.
- **R-9.2: Money** A reward also always contains in-game money which can be used to buy additional Booster-Decks at a market. The amount of the money depends on the quest as well as the user's performance. The difficulty level of the quest sets the maximum amount of possible money. This amount will be reduced by a certain percentage correlating to the amount of lost fights and wrongly answered questions while executing this quest. The resulting amount will always be greater than zero though.
- **R-9.3: Special Cards** Additionally a quest can also contain cards which are unique for this quest. Usually, these are cards representing a defeated opponent or one of its minions. If the player has not skipped any of the fight stations during the quest also one of these unique cards will be included in the reward. If the quest does not include special cards or the player has skipped fights no unique cards are added to the reward.

Due to this constellation a player is always rewarded for completing a quest but by performing very well the amount can be increased.

R-10 Player Statistics: Additionally to the road-map the Player Statistics screen functions as a central navigation point through the application. From this screen any other functionality of Myth Hunter is reachable. The primary purpose of this screen, however, is to present statistics about the payer. These statistics include:

- Amount of in-game money
- Number of active quests
- Amount of cards in total
- Quantity of decks
- Count of finished quests and stations
- Amount of started and won fights
- Number of answered questions
- Count of found hidden locations

By checking these numbers, players can compare their results and engage in a healthy competition which in turn motivates to complete further quests and learn their embedded content.

R-11 Random Enemies: If a player is in need of a stronger deck to defeat an enemy or is just stuck in an area with no or too few quests, it is possible to just fight against enemies which randomly appear on the map. Each enemy can be defeated once by a player and disappears afterwards for this player only. Other players can still fight against this enemy. Once per day all enemies are removed from the map and new enemies are generated at different locations.

Defeating such a random enemy rewards the player with in-game money which can be used to buy new Booster-Decks.

R-12 Tutorial: Due to the complexity of the card game and the different types of markers on the map, the Myth Hunter Application provides a beginners tutorial. This tutorial will first explain the rules of the card game by leading the player through an example fight. During this battle, the player will experience the basic functionality of the different cards and gain an overview of their abilities. The tutorial will also introduce explanation screens for the most crucial game mechanics which can be reviewed at any time even after the tutorial has finished.

Following this introduction of the card game the player will be presented with the needed information to start and complete the first quest. The Myth Hunter application will also include a glossary screen listing all icons used within the application and explaining each in detail. This glossary can be consulted at any time after the tutorial.

3.4 Non-Functional Requirements

Additionally to the described functionality, several non-functional requirements have to be taken into account while implementing the Myth Hunter Application. These requirements focus on one hand on the user's primary experience when using the application, like usability, and secondary side effects like the amount of mobile data usage and portability to other devices. On the other hand design decisions regarding the softwares maintainability are taken into account in this section.

3.4.1 Usability

NIELSEN (2012) describes usability as "*a quality attribute that assesses how easy user interfaces are to use*". Kim and Eom (2002) state that usability is a critical factor in achieving user satisfaction. Especially in games user satisfaction is needed above all otherwise players will just switch the application for one of its alternatives. Therefore it is necessary that the application never overwhelms the user by too much information. Especially on the roadmap, only a minimal number of absolutely needed markers should be shown by default. The user, however, should be able to enable the visibility of additional markers in an easy way.

The same should be true for the card game: Only the most important information required to understand the current game state should be displayed by default but all available information is accessible for the user to inspect if one wishes to.

3.4.2 Localization

Myth Hunter shall be playable all around the world. Therefore the language of the user interface must changeable by the user. In the first version at least English and German must be supported but an extension to other languages must easily be possible. This requirement applies only to static text defined in the application itself. Quests as well as cards are user-generated content and will, therefore, appear in their original language in the application regardless of the chosen display language.

3.4.3 Portability and Efficient Data Usage

For a player of Myth Hunter there should be no difference in the game's appearance regardless of the device on which the game is played. Furthermore, except during fights, users should be able to switch devices and continue exactly where they left off without the need of syncing the two devices in any form. The probability of game progress loss upon malfunctioning of a device should be minimized as well. To ensure this requirement global storage of user data and game progress will be necessary. This storage will be accessible through the Internet capabilities of modern smartphones. Due to limited access to Wireless LAN access point in the open, efficient usage of mobile data must be considered when implementing the application. Data should only be loaded from the global storage if needed while ensuring the application's responsiveness upon network problems.

3.4.4 Flexibility and Extensibility

The content of the application, namely quests, stations, and game-cards will not be stored within the application but at the global storage mentioned in the previous section. This allows the flexible addition of further content without the need of an application update. This content will be created within a Web-Editor which is described in detail in the thesis "Design and Implementation of a Web-based Authoring Tool for Location-based Gamified Applications" by Hutzler (2017, under review).

Though the functional requirements mentioned in this chapter already cover several types of station markers the design of the Myth Hunter Application should be open to the addition of further marker types. Especially interface changes to the global storage should be avoided. Also an extension of the card ability list (see Table 3.1 resp. 3.2) should be easily possible.

3.4.5 Security and Privacy

Within the Myth Hunter application as well as on the server no personal data will be stored. Users are only identified by their chosen user-name and password combination. Furthermore, the password will always be hashed at client side and just it's hash will be transferred to and stored on the server. For each user, the already completed quests are stored at the server but without any time-stamp. Therefore it is very roughly possible to determine where a specific user has been in the past but no other location data will be stored at the server once a quest is completed.

Because of the limited time available and the prototype status of the application no dedicated security measures, other than generally best practice coding to avoid code injection, will be implemented.

3.5 Design

This section covers the architectural design of the mobile application and a global data storage accessible through the Internet. Based on this architecture a more in-depth look at the quest execution process is provided.

3.5.1 Conceptual Components

Figure 3.3 illustrates the conceptual components of *Myth Hunter*. It contains the client application which features the scavenger hunt and the card game. The main interface of the scavenger hunt is the world map on which quests and tasks are placed on. The player uses the world map to execute quests and earn new cards.

The player can add newly earned cards to an existing deck or create new decks using the deck editor. Afterwards, this cards can be used within the card game against random enemies or quest-opponents.

The requirements state that the client is supposed to be lightweight and shall store content only temporarily. Therefore each instance of the mobile application is connected through the Internet to a global data storage on a web server. This server offers a web service which loads, manipulates and stores data into an underlying database management system (DBMS). This DBMS stores information such as users, quests, cards, and decks.

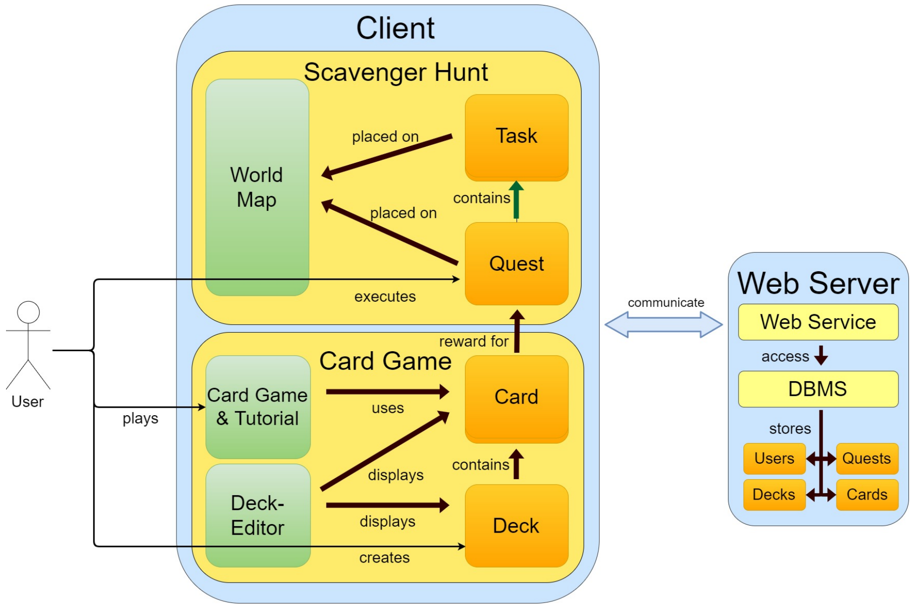


Figure 3.3: Conceptual Components of the Myth Hunter Software Product

3.5.2 Execution of a Quest

The mobile application is designed as lightweight client displaying and manipulating content but not storing it. Figure 3.4 shows how and in which sequence a user, the mobile application, and the global data storage interact with each other during the execution of a quest.

After a user has started to the mobile application a login with username and password is needed to identify and link the physical person to the user information stored in the global data storage. After the successful validation of the provided login credentials on the web server, the user's information and active quests are sent to the mobile application. Afterwards, the mobile application requests all not yet started quests within the user's area from the server and displays them as markers on the roadmap.

The player can now start a new quest which will be processed on client side resulting in the display of the first task on the map. The global data storage is only informed asynchronously about the quest start to update the information within its database.

The user now executes task after task until the quest is finished. After each execution, the server is informed about the status update but again the calculation of the next task(s) is done by the mobile application. Upon execution of the last task, the server is also informed about the quest finalization, and a Booster-Deck is requested. The player is notified of the successful completion of the quest and the earned reward including the Booster-Deck.

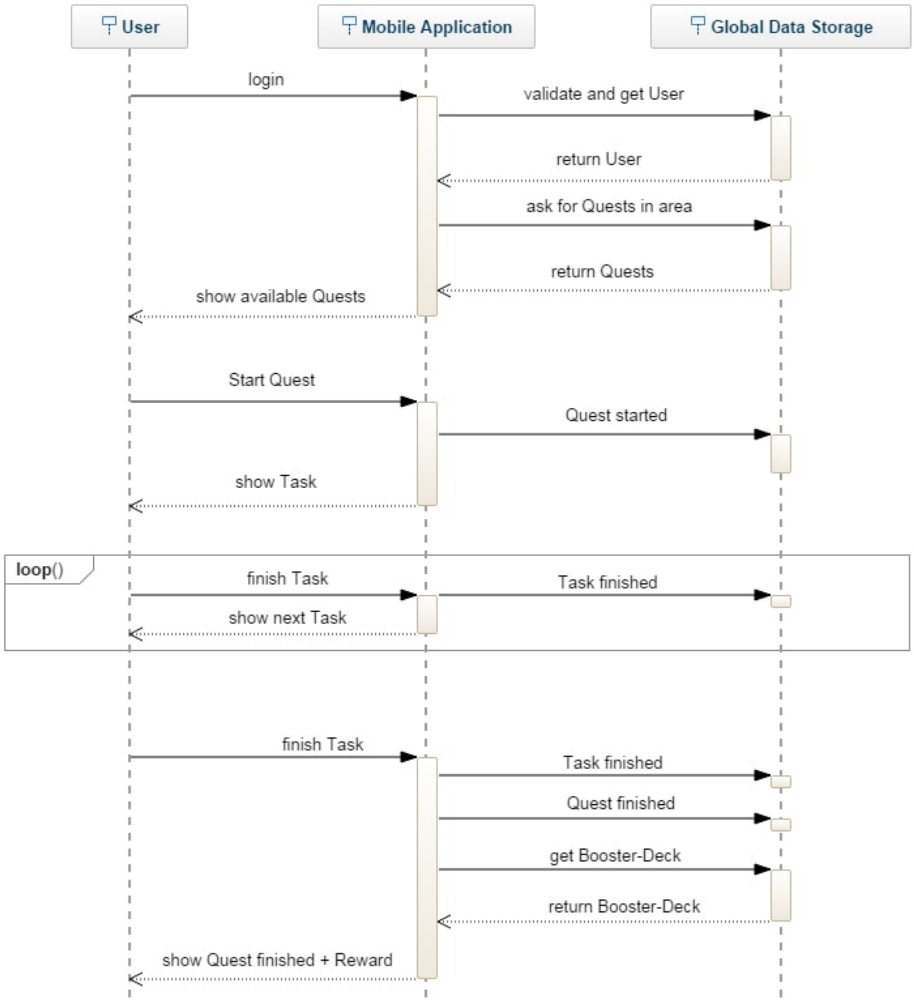


Figure 3.4: Sequence diagram of a Myth Hunter quest execution

3.6 Summary

In this chapter requirements for Myth Hunter were formalized. These included functional requirements for the collection card game as well as the scavenger hunt and additional gamification features. The game mechanics and rules were explained in detail and some design decisions were mentioned. These decisions were formalized as non-functional requirements in this chapter. Finally an architectural overview of *Myth Hunter* was given including server and client and how communication between them will look like in detail.

Now that all requirements are known the next chapter will describe how the actual development of *Myth Hunter* looked like and which tools and frameworks were used.

4 Development

This chapter covers the development of the Myth Hunter Project based on the requirements described in the previous chapter. First, an overview of the Myth Hunter components is given. Afterwards, the client implementation is presented in detail including general implementation approaches as well as the realization of the scavenger hunt, card game, and gamification elements. The development of the server is presented focusing on the general data structure and used tools. Finally, Some examples of use cases from a users perspective are given including in-game screenshots.

4.1 Motivation

In the previous chapter, the requirements Myth Hunter needs to fulfill were stated. This chapter describes how the development was done in detail. Used tools are mentioned as well as how specific requirements were realized from a technical point of view. The data model on both client and server side are presented explaining their use in the application and the communication between client and server. After the technical description, the client application's usage from a user perspective is explained describing the user interface of different screens and use cases.

4.2 Component Overview

As displayed in Figure 4.1 Myth Hunter uses a client-server architecture. All frameworks and applications mentioned in the figure respectively in this section are explained in more detail later in this chapter.

The Myth Hunter client application can be installed on any Android device. The application was developed using Unity 5[1] therefore the Unity UI is the primary interface with which a user interacts during playing. The application does not only display quests and cards but also contains the game logic. However, quests and other data are just stored temporarily which means they must be reloaded through a web service-client upon each start of the application.

Data used within Myth Hunter is permanently stored in a PostgreSQL database hosted on a Linux server. This database is connected through Java Hibernate[2]

[1]https://unity3d.com/
[2]http://hibernate.org/

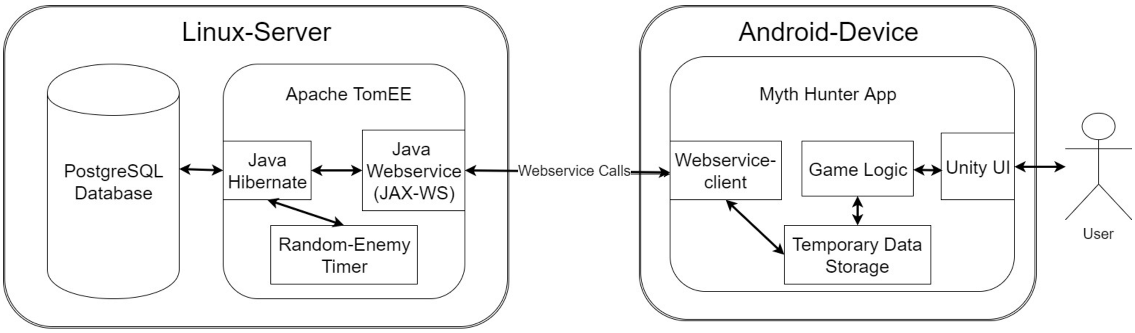


Figure 4.1: Overview of the Myth Hunter components

to a Java web service hosted within an Apache TomEE webserver[3]. Within this web server, a timer runs with the purpose of creating new random enemies and storing them into the database. The development of this components is now explained in detail starting with the client application.

4.3 Client Application

The Myth Hunter App for Android was developed using Unity 5.5 and several other tools and frameworks. These is now explained in detail followed by a description of the Myth Hunter client development.

4.3.1 Tools and Frameworks

This section mentions tools and frameworks used for the development of the Myth Hunter Android client.

Unity 5: The Myth Hunter Android Application was developed using the Unity Game Engine[4]. Unity is written in C++ and "*integrates a custom rendering engine with the nVidia PhysX physics engine and Mono, the open source implementation of Microsoft's .NET libraries*" (Jeff Craighead, 2008). It includes a Graphical Development Environment as can be seen in Figure 4.2 which allows developers to manipulate game scenes and create levels directly. In general, Unity can build games for several platforms such as Windows, Linux, Android, and IOS. Within Myth Hunter, however, some native Android code is used which is explained later in this chapter.

Web Services Description Language-Tool Wsdl.exe: The tool Wsdl.exe developed by Microsoft[5] is part of the .Net Software Development Kit and allows

[3]http://tomee.apache.org/apache-tomee.html
[4]https://unity3d.com/
[5]https://www.microsoft.com/

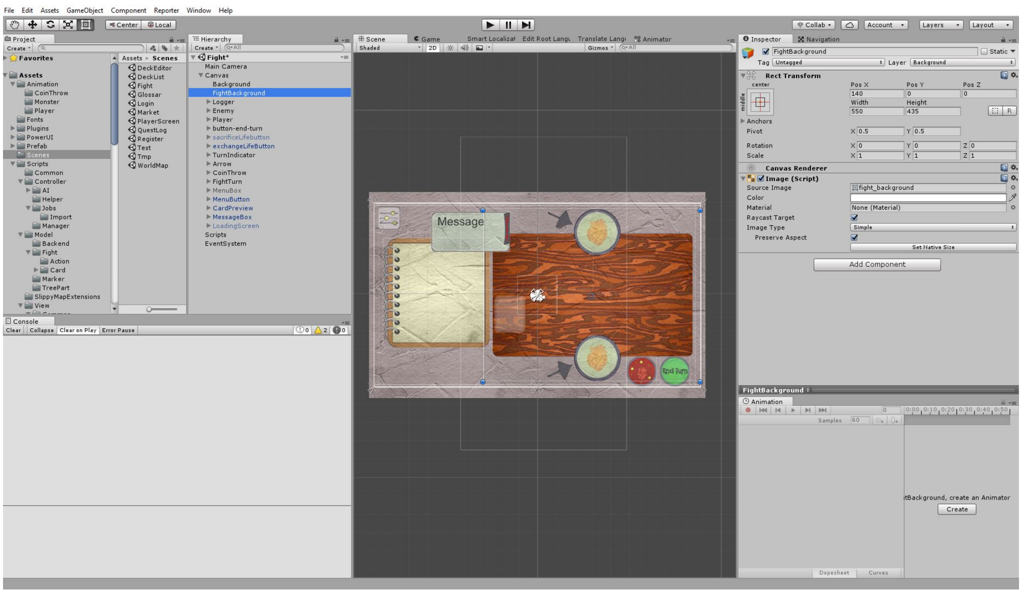


Figure 4.2: Screen-shot of the Unity Development Environment

to *"generates code for XML Web services and XML Web service clients from WSDL contract files, XSD schemas, and .discomap discovery documents"* (Microsoft, 2017). Within Myth Hunter the tool is used to generate C# classes representing objects which can be sent and received from the Myth Hunter server via web service calls.

UnitySlippyMap: UnitySlippyMap[6] is a framework for Unity developed by Jonathan Derrough. It helps developers integrate 2D and 3D maps into their games working with a variety of online tile providers like OpenStreetMap[7] or Bing Maps[8] as well as offline sources such as DBMap or MBTiles. It is released under the "GNU Lesser General Public License" 3.0 but still in alpha stage. Myth Hunter uses UnitySlippyMap to display a road map and quest markers onto this map as are further explained in section 4.5.3.

PowerUI 1.9: PowerUI[9] developed by Kulestar is a Unity GUI framework available at the Unity Asset Store to display user interfaces using standard HTML and CSS easily. The interface can be displayed as a full-screen overlay atop the actual game as well as in-game in 3D world space. The HTML content can be manipulated using Unity c# code as well as embedded "Nitro" code. Nitro is a scripting language within PowerUI similar to JavaScript. Even though PowerUI can display simple web-pages, it is not intended to browse the web and does not support standard JavaScript.

[6]https://github.com/jderrough/UnitySlippyMap
[7]https://www.openstreetmap.org/
[8]https://www.bing.com/maps
[9]http://powerui.kulestar.com/

Quest and task-informations within Myth Hunter are stored as HTML pages on the server. PowerUI is used to display this pages within the Android application.

Smart Localization: Smart Localization[10] is a Unity plugin developed by Niklas Borglund and Jakob Hillerström which helps to make an application multilingual. In key-value pairs, words and phrases for different languages can be stored and displayed by referencing the key in the code. The tool includes import and export functionalities to translate the needed phrases into another language easily. Unfortunately, as of September 2017, the tool is no longer supported and therefore no longer available in the Unity asset store.

Unity-Logs-Viewer: The Unity framework Log Viewer [11] provides functionality to view log messages within the game on the target device as they could be seen in the Unity Development Environment. Additionally, it displays statistics such as CPU and memory usage. This eases debugging on the target device, but the framework should be removed before a release.

4.3.2 General Implementation

This section describes general design decisions made during the development phase.

Webservice Connection: Upon start of the Myth Hunter application, no data is stored on the device but must be loaded from the global data storage of the Myth Hunter server. This communication is done via a WSDL-based web service. Using the Web Services Description Language-Tool entity classes and a web service client implementation were created. A wrapper-class around this implementation was used to abstract the method calls and convert the internally used entities to the ones used by the web service.

Data Model: Figure 4.3 shows the general data model used in the Myth Hunter application. On the bottom right-hand side of the figure the *User* and it's *UserStatistics* is shown. A user, in general, contains information about the player like the username, hashed password and the amount of in-game money the user owns. Additionally, a user can have several decks as well as cards not present in a deck. For the cards, a map containing the card itself and the amount of owned instances of this card is used. This representation is used several times in different entities.

The *StaticCard* object contains all information about a certain play card like described in section 3.3.2. The implementation of what a spell- or monster-card

[10] https://www.assetstore.unity3d.com/en/#!/content/7543
[11] https://www.assetstore.unity3d.com/en/#!/content/12047]

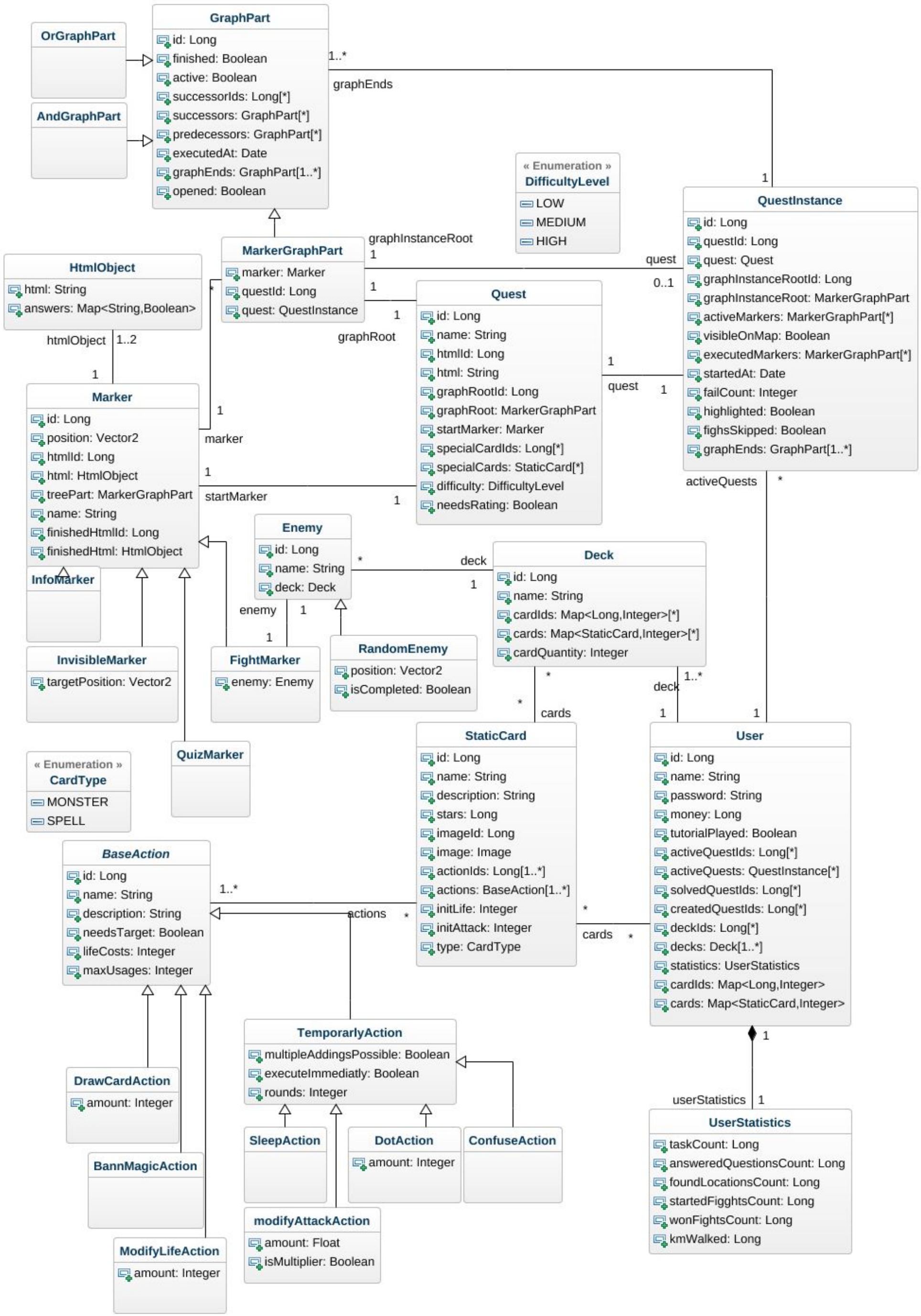


Figure 4.3: The Myth Hunter Application Data Model

can do is done in *BaseAction* and all its derivations.

The *User* entity also stores references to finished and active quests. The entity *Quest* is used as a template containing all the information about a certain quest but no information about a users progress which is stored separately in the entity *QuestInstance*. This means a certain *Quest* only exists once globally on the server but several *QuestInstances* referring to this quest can exist. However per user always only one *QuestInstance* referring to a certain *Quest* can exist.

A quest itself consists of *Markers* where the derivation describes what a user has to do to complete this marker. This can for example be to defeat an *Enemy* of a *FightMarker* which has an own *Deck* of cards. The structure of a quest and the order of the marker is stored within a graph using *GraphParts*.

Lazy Loading: In Figure 4.3 can be seen that in several entities an object reference as well as it's id is present. This is used for two kinds of lazy loading: (1) data is only loaded from the server when it is needed reducing traffic overhead and (2) additional data is loaded asynchronously in the background if the data will most likely be required in the future but not at the moment. This allows shorter response times and quick scene changes.

Entity Manager: All loading of entities from the server is done via a class called *Entity Manager*. Its purpose is to ensure that always only one instance of an entity exists in memory. This is especially important when loading a quest graph because in the XML representation one *GraphPart* can appear several times if it is referenced several times by different parts. Therefore the *Entity Manager* holds all loaded entities and their ids in lookup maps and only requests data from the server if no object with the given id is present and if no open request is found for that entity. Additionally, when receiving data from the server, it checks if parts of it already exist within its maps and discards the data parts accordingly favoring the objects in the maps.

4.3.3 Scavenger Hunt

This section covers development decisions which were made specifically for the scavenger hunt.

GPS Connection: One important requirement for Myth Hunter to work properly is the ability to determine the user's position precisely. Unity provides functionality for retrieving the devices current geo-location but during our prototype evaluation, we found that the position retrieved from unity is not precise and stable enough for our purposes. Unity allows to integrate native code into the project, therefore, the location determination was implemented in a native android plugin. A Unity-plugin class must derive from *UnityPlayerActivity* but can implement any functionality.

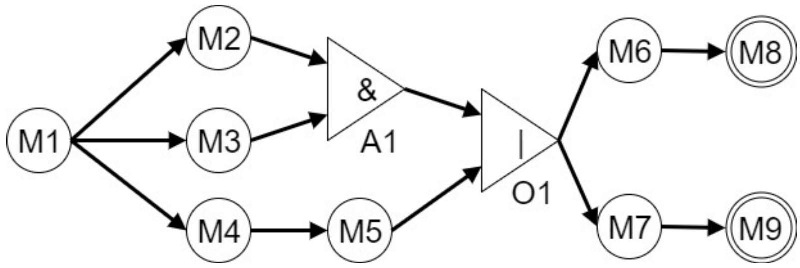


Figure 4.4: An example graph of a quest in Myth Hunter

Within the app's c# code a plugin class can be loaded by its class-name using *AndroidJavaClass* which provides methods to invoke plugin -methods by stating its name and the return type.

The location plugin used in Myth Hunter is rather simple and only provides methods to check whether GPS is enabled on the device and retrieve the current location. GPS and network provider determine the location and on both the accuracy is set to maximum. Due to this plugin the Myth Hunter application is only available for Android devices at the moment.

Quest-Graph: A quest in Myth Hunter consists of several objects: the *Quest* containing the general information about the quest and its content, *Markers* describing the stations and tasks a user has to complete and connecting *GraphParts* for structure. Figure 4.4 shows an example of how the structure of a quest could look like.

Circles represent *MarkerGrapParts* at which a user has to fulfill a task. Each quest must start and end with *MarkerGrapParts* whereby only one start point is possible, but multiple endpoints (double-circle) are allowed. When a *MarkerGrapPart* is completed it's successors become available. After M1, in the case of Figure 4.4, is completed M2-M4 become available.

A1 is an *AndGraphPart* which means that all its predecessors have to be completed in order to reach the successors. In our case, M2 and M3 have to be finished to reach O1.

An implicit *And*-connection is also possible when connecting two *MarkerGrapParts* directly like M4 and M5. In this case, however, M4 has to be completed before M5 becomes available while M2 and M3 are available at the same time.

The third possible *GraphPart* is the *OrGraphPart* which evaluates to true and activates it's successors as soon as one of its predecessors is completed. In the given case either M2 and M3 **or** M4 and M5 have to be completed to reach M6 and M7. This also means that if one path (e.g., M4 and M5) reaches the *OrGraphPart* all other paths are no longer available.

An or-condition can also implicitly be defined when a quest has more than one end-point. In our case, a user can choose to complete M6 or M7 however if M6 is completed M7 and M9 are no longer available. In general, if a user completes a task all end-points which are not reachable from this *GraphPart* are disabled as

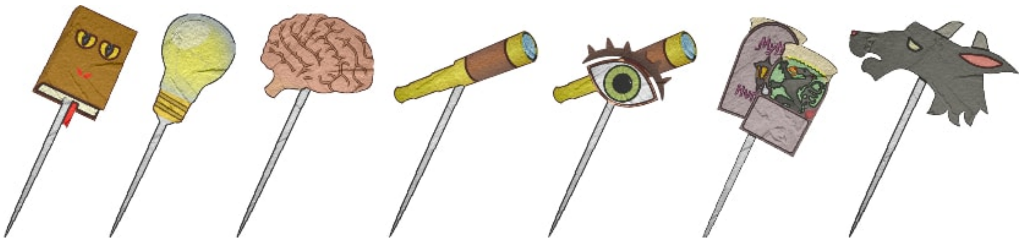


Figure 4.5: Markers as shown on the map. From left to right: New Quest, Information, Quiz, Find-Start, Find-End, Fight and Random-Enemy- Marker

well as the paths leading there.

Quest Execution: During the execution of a quest, task-markers are shown on the map, and a user has to physically reach the location of a marker to open it and execute the associated task. Depending on the graph structure one or more stations become active after a previous station is completed.

A quest can consist of several different kinds of tasks. These tasks are displayed on the map as markers. Figure 4.5 shows the most common marker including a marker for new, not started quests on the left and the Random-Enemy marker on the right. The other markers represent the different types of quest-tasks. New quests and Random-Enemies are only loaded for the currently visible map-area and only until a certain zoom level is reached not to overfill the map. When the map has moved a request for new quests and enemies in the new area is sent to the backend-server including the identifiers of already loaded entities. The server only returns entities which identifiers were not present in the list to avoid duplicate sending of information.

Usually, a station has to be finished to move on(if the quest structure doesn't say otherwise using an 'OR'-connection). However, Myth Hunter allows Fight-markers to be skipped so players, who are just interested in the story but do not like the card game, do not have to play it. This action reduces the possible quest reward, however.

4.3.4 Card Game

This section focuses on the development of the card game. It describes how the card game works on a technical level.

Data Model: For the execution of the card game, additional entities have been created. These entities are shown in Figure 4.6. Upon start of a fight a *Fighter* is created representing the user and another one for the enemy. This entity stores all cards in the different stages (in hand, in the pile, on the board or dead) and other attributes changeable during the fight like the amount of stars a fighter currently has available.

For each card in the user's deck as well as for all enemy cards a *FightCard*,

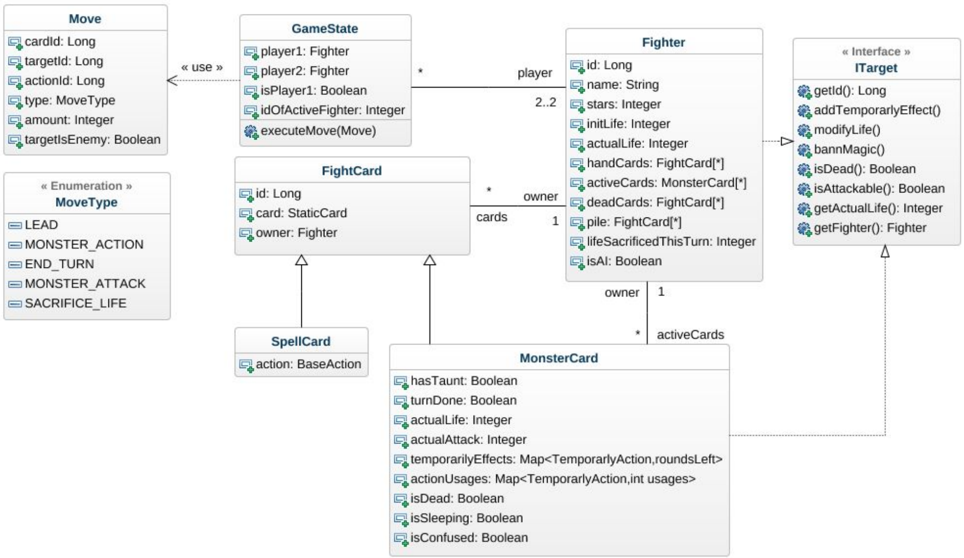


Figure 4.6: The Card game execution entities

either *SpellCard* or *MonsterCard*, is created. These entities consist of a *StaticCard*, a unique id and several other attributes storing possible changes of the card during a fight like loss of life points.

Both, a *Fighter* and a *MonsterCard* can be targeted by an attack or spell and therefore implement the *ITarget* interface. If a *MonsterCard*'s *actualLife* reaches zero the *isDead* flag is set and it will be removed from the *Fighter*'s *activeCards* and added to it's *deadCards*. If a *Fighter*'s *actualLife* reaches zero the fight is over and it's opponent wins.

An entity called *GameState* stores the *Fighters* and therefore all information of the specific state the game is in. Executing a legal *Move* on the *GameState* alters it and the underlying *Fighters* and *FightCards*. In general five different types of *Moves* exist:

- **Lead:** A monster or spell-card is played from the hand-cards onto the board.
- **Monster-attack:** An active monster on the board attacks a target using it's attack-points.
- **Monster-action:** An active monster on the board attacks a target using a special action.
- **Sacrifice life:** A fighter chooses to exchange life-points for additional stars.
- **End Turn:** A fighter signals the end of it's round and the opponent's turn starts.

During a single round several *Moves* can be executed by the same fighter in a sequence which always ends with an end-turn-move.

Fight Coordination: A central singleton class called *FightManager* coordinates the execution of a fight. It is responsible for creating a *GameState* at the beginning and keeping it updated by executing *Moves*, received either from the user interface or the AI, on the *GameState*. The *FightManager* also creates one background working-thread for computation-intensive tasks not to block the UI. The communication between the UI-thread and the background-thread id done via *Jobs* stored in an import and an export queue within the *FightManager* depending on the direction. All AI-moves are executed using the background thread creating several jobs to update the UI accordingly or play animations. On each frame, the main UI-class *FightScreen* checks if jobs are available in the queue and executes one after another ensuring the right order of events and animations. Likewise, the background thread checks and executes jobs of its queue periodically. One important import-job is the *PlayerTurnDoneJob* which signals that the user has finished the turn and that the AI can start calculating its moves.

Myth Hunter AI: The card game AI of Myth Hunter is based on the Information-set Monte Carlo Tree Search algorithm. Additional Myth Hunter specific knowledge is included in the algorithm when collecting possible moves. Some moves don't make much sense even though they are legal like healing an enemy. Therefore the AI does not take such moves into account reducing the number of possible moves and therefore the branching factor. Myth Hunter has no move order which means within a turn a player can execute moves in arbitrary order. In the game tree of the AI this would lead to a higher branching factor and redundancy in the tree. Therefore the Myth Hunter AI executes moves always in the following order within a turn:

1. Sacrifice life
2. Lead moves
3. Monster moves
4. End Turn

Move-types can, of course, be skipped but if e.g., a monster move is executed no lead or sacrifice live move will be executed afterwards in this turn.

During the Determinization phase of the algorithm all hand cards of the human player are put back into the deck, shuffled and the same amount drawn again. In other words: the AI knows which cards are in a player's deck and which cards are already played but the hand cards and the order of the deck cards is randomized. To increase the performance of the algorithm the deck of the AI is not shuffled and remains in the same order during the whole game.

To reduce calculation time moves which do not require a target, and therefore do not alter depending on the game-state, are created once and stored within the associated entity rather than destroyed and recreated in each AI iteration.

Most smartphones nowadays include multi-core CPUs allowing multiple threads to be executed simultaneously. The Myth Hunter AI implements Root Parallelization by creating one thread per logical core during an AI turn. The

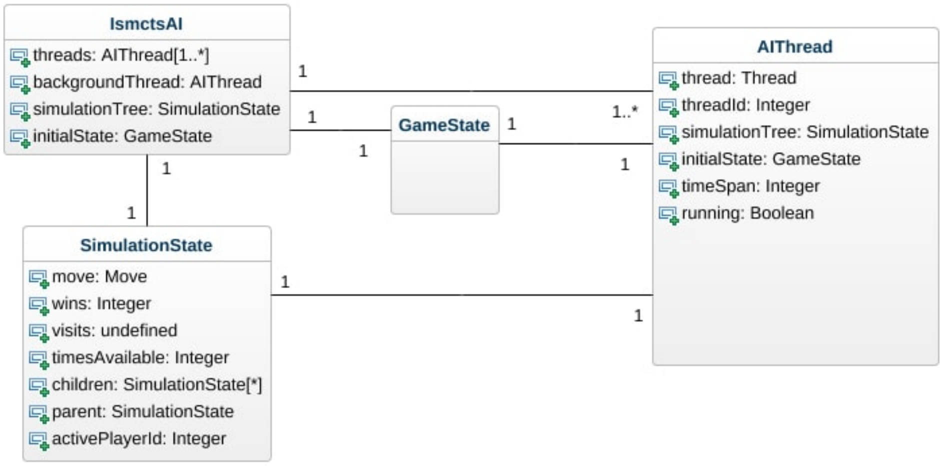


Figure 4.7: AI entity Class-Digram

structure of the AI classes is shown in Figure 4.7. *IsmctsAI* is the base class being called on the start of an AI turn. It holds the current *GameState* and the game-tree of the previous turn consisting of *SimulationStates*. Upon the start of a new turn it updates the *GameState* and tries to find the *SimulationState* which corresponds to the current *GameState* within the game-tree. This allows reusing some of the calculations done earlier in the game. Afterwards it distributes the current *GameState* and, if possible, the according *SimulationState* to its working threads. Each thread know calculates simulations for a predefined timespan and returns its simulation tree in the end. To avoid similar trees each thread uses its own seed for randomization during the Determinization phase and also when selecting random moves during simulation.

After all threads have finished their calculated trees are merged, and the best moves for the turn are selected. One of the threads remains calculation even during the player's turn to maximize the possibility of finding a matching *SimulationState* when starting the next AI turn.

4.3.5 Gamification

Reward: Upon finishing a quest the user receives a reward. The amount of the reward depends on the quest difficulty as well as the user's performance while executing the quest. The more difficult a quest is the more significant is the possible reward. During the quest execution, the game counts how often the user gives wrong answers in a quiz task or loses a fight. At the end of the quest, this count determines how much of the possible reward a user receives. The reward always includes in-game money as well as a booster-deck containing three random cards. If a quest has a special card attached and the user has not skipped a fight during the quest, also this special card is included in the reward.

Player Statistics: During the game, the application keeps track of the user's actions and calculates statistics like the total number of finished quests and tasks. As soon as a certain action occurs, (e.g., a task is finished) counters in the entity *UserStatistics* are increased accordingly on the client as well as on server side. In a special player screen, all this statistics can be vied by the player.

Tutorial: After registering the user is asked to play a tutorial explaining the card game and its rules. The tutorial leads the player through an example card game using text messages to explain what to do next and why. The cards, as well as the order in which they are drawn, is predefined. During the tutorial only certain actions are allowed at a given time to prevent, for example, ending a turn without leading a card. This would be possible in a normal game but is not in the tutorial's intention. The tutorial will always result in a win for the player. Afterwards, key elements of the scavenger hunt game are explained leaving the user at a glossary screen to further explore further the icons used in the game.

4.4 Server

The Myth Hunter prototype server is installed on a Linux operating system and consists of two parts: (1) a PostgreSQL Database and (2) an Apache TomEE web server. The web server hosts a XML-web service implemented in Java which uses Java Hibernate to communicate with the database. This web service provides functionality to add, update and delete data on the server but seldom manipulates data itself. Additionally, a timer, responsible for the creation of random-enemies, is implemented and is further explained in section 4.4.3.

4.4.1 Tools and Frameworks

PostgreSQL: The main responsibility of the Myth Hunter server is to store data like user profiles, cards, and quests and provide functions to add or retrieve such data. In order to meet this requirements a PostgreSQL[12] database is used. "*PostgreSQL is a powerful, open source object-relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness.*" PostgreSQL was chosen for Myth Hunter mostly because of its easy installation, good documentation and it's open-source license.

Java Hibernate: Hibernate[13] is an open-source object-relational mapping tool for the Java programming language. It is used to map an object-oriented domain

[12]https://www.postgresql.org/
[13]http://hibernate.org/

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <class name="gen.entities.DbRandomEnemy">
        <id name="id" column="id" type="long">
            <meta attribute="use-in-tostring">true</meta>
            <generator class="identity"></generator>
        </id>
        <version column="version_column" name="version" />
        <component name="position" class="gen.entities.MapPosition"
            lazy="false">
            <meta attribute="use-in-tostring">true</meta>
            <property name="longitude" type="float" column="position_longitude">
                <meta attribute="use-in-tostring">true</meta>
            </property>
            <property name="latitude" type="float" column="position_latitude">
                <meta attribute="use-in-tostring">true</meta>
            </property>
        </component>
        <many-to-one name="enemy" column="enemy" class="gen.entities.DbEnemy"
            lazy="false" cascade="merge">
        </many-to-one>
        <list name="completedByUserIds" lazy="false" cascade="merge,delete">
            <key column="randomEnemyId" />
            <index />
            <element type="java.lang.Long">
            </element>
        </list>
    </class>
</hibernate-mapping>
```

Figure 4.8: An example Hibernate mapping file

model to a relational database. Moreover, the usage of Hibernate Tools within the Eclipse-IDE[14] allows developers to define entities, constraints, and sequences in XML annotation and automatically generate Java classes as well as the required database tables.

In Order to use this feature first, the entities must be defined in XML representation using Hibernate mapping files. An example file can be seen in Figure 4.8. Objects defined in such files can consist of basic data types (e.g., long ”id”) as well as complex components (e.g., the MapPosition) and relations to other objects which are defined in other mapping files (e.g., DbEnemy). This files can afterwards be used to generate the according Java classes and their data access objects.

Data access objects are used to persist, retrieve and merge the objects to and from the database. Hibernate always operates within a so-called transaction. Within a transaction, data can be manipulated in memory without affecting the underlying database. Only after a transaction is complete, the changes are automatically committed to the database or can be rolled back. It is also possible to detach object which means that they are loaded in one transaction but committed in another. In this case, however, hibernate must be explicitly told to merge the changes into the database using the data access objects.

If configured correctly Hibernate will automatically create the needed tables in the configured database schema. Basic data types and components will be stored in one table representing the defined entity while only the primary key of a related object will be stored. The related object itself is stored in a separate

[14]www.eclipse.org/

table.

Hibernate also provides it's own Data Manipulation Language called Hibernate Query Language (HQL) which allows the collection of data from the database at object level rather than referring to tables as in basic SQL. Depending on the object structure and fetch policy one simple HQL select statement can be translated to a complex SQL query joining several tables at the database level.

Apache TomEE: Apache TomEE[15] is the Java Enterprise Edition of Apache Tomcat, an "*open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies*". The Tomcat project started at Sun Microsystems and its code was donated by Sun to the Apache Software Foundation in 1999. Since 2005 Tomcat is an own top-level Apache project, which means, that it left the Jakarta umbrella and oversight structure to manage itself.

Within Myth Hunter an Apache TomEE instance is used to run a java web service building the Myth Hunter backend as further explained in section 4.4.

4.4.2 General Implementation

Figure 4.9 shows the Myth Hunter server's internal data model. Due to the primary purpose of storing data the overall structure of the server data model is very similar to the data model of the Android application. However, while most objects are directly referenced in the Myth Hunter application, on server-side only lose referencing using ids or even list of identifiers is used. This allows to lazy-load data via the web service. Otherwise, lots of possibly unnecessary data would be loaded. For example when loading a *Quest* merely to show it's position on the map there is no need to load all *GraphParts* and *Markers*. If the user chooses to play this quest the data can be loaded on demand.

Another difference to the client's data model is that on the server no derivations are used. In the *Action* entity for example the information for all action types is stored using the enum *ActionType* to distinguish the actual type of the action.

4.4.3 Random Enemy Creation

Random Enemies are opponents randomly paced at the map which can be defeated by a user using the card game. These enemies are created once per day and store each a list of users who have already defeated this enemy. This ensures that one enemy can only be defeated once by a certain user but is still available to other users. Because of insufficient server resources in our prototype, roughly one enemy per km^2 is created only for the state Styria in Austria.

[15]http://tomee.apache.org/apache-tomee.html

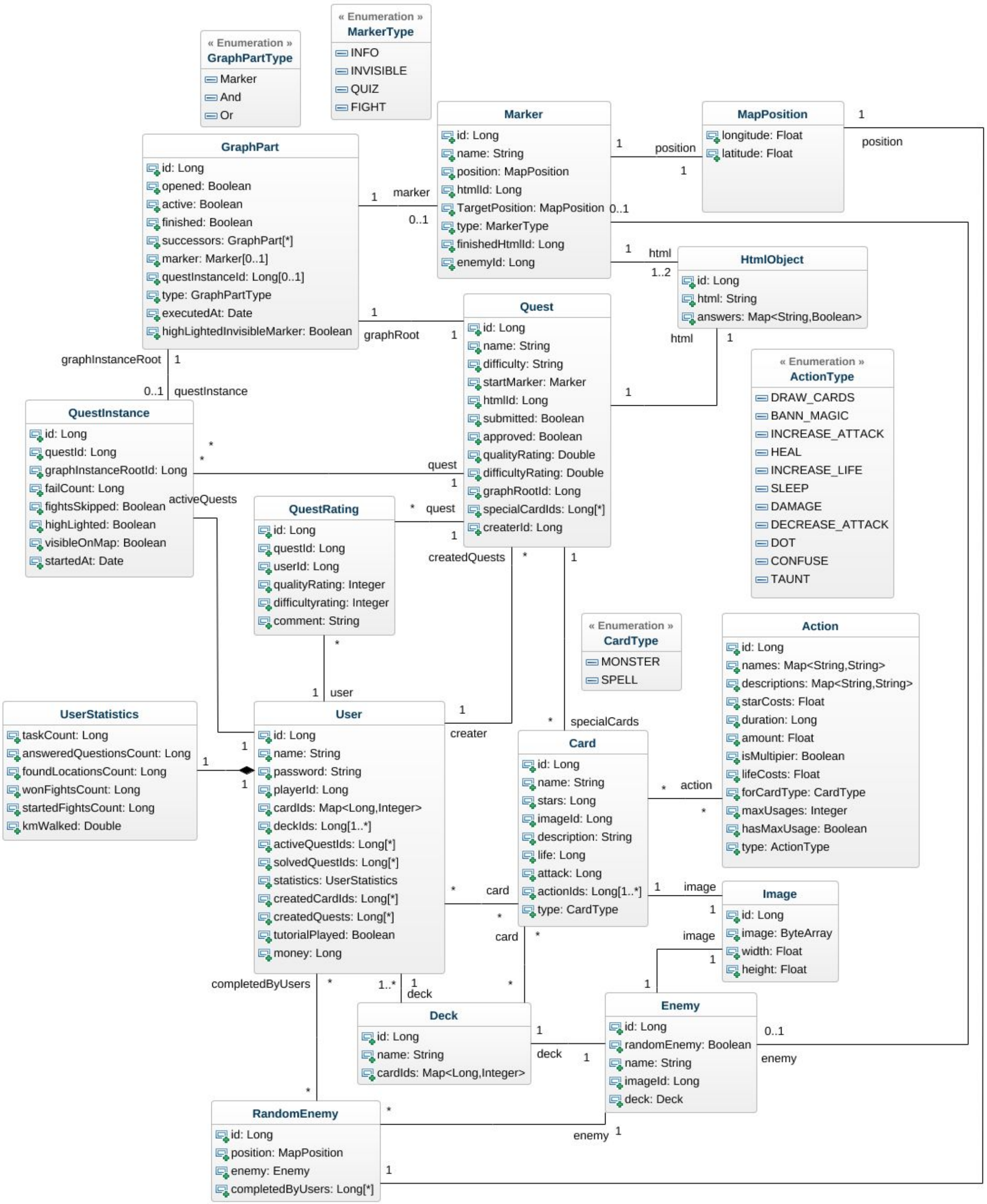


Figure 4.9: The Myth Hunter Server Data Model

a) The Tutorial message upon first login

b) The Glossary screen includes descriptions about all used icons.

Figure 4.10: Tutorial and Glossary

4.5 User Perspective

This section explains the game interface from a user's perspective.

4.5.1 Tutorial

After the first login, the player is asked to play the tutorial as can be seen in Figure 4.10a. The user can choose to skip the tutorial, but the message will be displayed again on the next login till it is either finished or hid permanently using the option of the message. During the tutorial explanation-screens of the game-board as well as of the cards are shown to the user. This explanation screens can be reviewed at any time even after the tutorial using the menu-button of the card game interface. After the card game is won, the tutorial returns to the road-map explaining the scavenger hunt interface using a glossary screen. This screen can be seen in Figure 4.10b and includes descriptions for all icons used within the Myth Hunter application.

4.5.2 Card Game

Figure 4.11 shows the main screen used to play the Myth Hunter card game. In the middle of the screen, the player's and enemy's active creatures are located. Due to limited space, a small representation containing only the creature's picture and its life and attack points were chosen. By tapping on the creature the user

Figure 4.11: Screenshot of the Myth Hunter Card Game Interface

can open a big view of the actual card containing all information like in Figure 3.1.

Above, respectively beneath, the monsters the life and current star amount of player and enemy are located. On the bottom left-hand side, the user's current hand cards are shown. Again, a tap on the card opens a big view. Above the hand cards, a log-book shows additional information about everything that has happened since the start of the fight. The bottom right-hand side provides buttons to sacrifice life and end the turn. The menu button in the top left-hand corner allows opening of several explanatory screens as well as to abort the fight.

To play a card onto the board or attack an enemy the player has to draw a line from the source to target on the screen. An arrow appears indicating the source and the current position of the finger on the screen. During this process, valid targets are highlighted. Alternatively, it is also possible to start this process by tapping and holding the big view of a card for about one second.

4.5.3 Map Interface

The road-map is one of the most important screens of Myth Hunter and probably the one users will spend the most time on. It is the first screen a user sees right after login. In Figure 4.12 one can see the map with a marker and several buttons on the edge of the screen. These include:

- On the bottom left-hand side of the screen, two toggle buttons are placed. The top one known as **New-versus-Active Toggle** is used to switch between displaying new, not yet started, quests on to map and active quests. In Figure 4.12 the toggle is set to display active quests indicated by the lightbulb while a closed book would indicate new quests.

Figure 4.12: Road-Map Interface with Information marker

- The purpose of the second toggle button is to **show/hide finished markers**. In default settings, neither finished quests nor already finished tasks of an active quest are shown on the map. By tapping on this toggle button, this can be enabled to play a quest again or review a finished task.
- On the bottom center the **Player button** is located which navigates to a player screen which functions as central navigation point of the application. This screen is described in further detail in section 4.5.4.
- The **Quest-Book** on the top left corner of the screen directly opens the Quest-Log.
- Next to the *Quest-Book* the **current active quest** is displayed. A tap on this banner opens the quest-log same as the quest-book. This banner, however, is only shown if the user has an active quest and chose to display active quests rather than new quests on the map using the *New-versus-Active Toggle* button.
- The main purpose of the **compass** in the top right corner is to show the direction the user is looking towards. The map itself features two modes which can be switched by taping on the compass: (1) the map rotates in such a way that the if a user looks straight into a street that same street is displayed at the screen facing upwards. (2) the map does not rotate on its own but the user can rotate it manually as one would do with a paper map and a compass.
  Additionally small marker symbols on the edge of the compass show in

a) The Player Menu screen including statistics

b) The Quest-Log screen showing active quests and markers

Figure 4.13: Statistics and Quest-Log Screenshots

which direction a user needs to go to reach the next marker(s) of the active quest.

- The **Position-Button** right beneath the *compass* moves the map to display the user's current position right in the center of the screen indicated by a blue dot.

After finishing a station, the map interface automatically moves and zooms in such a way that both the user and all currently active stations are visible at once. Additionally, the newly added stations flash a couple of times to help the user understand what has changed.

Initial user tests have shown that the concept of the Find-task was not easily understood. Therefore after a Find-marker was opened a message appears explaining the what to do. Additionally, a help functionality is available which displays an eye on the screen. The closer a user gets to the invisible target location the wider the eye will open to indicate that the user is going in the right direction.

4.5.4 Player Screen

Figure 4.13a shows the player menu screen. It is the main navigation point of the Myth Hunter application including buttons to all the major features of the software including the deck editor, market and a glossary explaining all symbols used within Myth Hunter. Additionally it includes statistics about the player as defined in Chapter 3.

a) A List of the user's decks

b) The Myth Hunter Deck Editor interface

Figure 4.14: Deck creation

4.5.5 Quest-Log

In a separate screen called "quest-log" (see Figure 4.13b) all active quests are listed including their active and finished markers. Users can re-view the descriptions of all these markers and adjust which markers shall be shown on the map. One quest is marked as the "highlighted-one" which means that it's markers are shown on the map as well as on the edge of the compass. Other quests can additionally be marked as visible to show it's markers on the map. The user can directly jump to visible markers on the map from the quest-log screen.

4.5.6 Deck Editor

The Myth Hunter application features a deck editor allowing the user to create new decks or modify respectively delete existing decks. The editor consists of two scenes: Figure 4.14a is the first screen showing all decks of a user and the cards within the selected deck at the bottom. A green check mark on the left of the deck signals that the deck is complete an can be used in a fight. A user always has to have at least one complete deck, therefore, deleting the "standard deck" in this case is not possible.

Pressing "Edit" on an existing deck or the new deck button in the bottom right-hand corner opens the deck editor itself (Figure 4.14b). In the middle of the screen, the user's cards are displayed in pages of nine cards per page. The user can scroll through the page or use the arrows to switch to a different page. Filtering the cards by name, type, and several other attributes is also possible. On the bottom, the cards which are already in the deck are shown. By either long-tapping on a card or double-tapping, it can be added to or removed from the deck.

4.6 Summary

In this chapter important decisions made during the implementation process were mentioned. These included which tools and frameworks were used as well as the data structure of client and server. Due to technical complications such as multiple occurrences of identical entities in the webservice's XML representation wrapper and manager classes were introduced. These classes convert data received from the server in a usable form for the client like a graph representation for quests. Additionally, these classes ensure efficient data usage and allow lazy loading. Screen-shots showing the user interface for the scavenger hunt and card game were presented and explained. Especially for the card game coordination between the user interface and the AI calculation in the background had to be taken into account. Therefore a single communication point with import and export queues were implemented.

Even though the implementation of Myth Hunter was an iterative process with included user feedback, a evaluation study with key users of the target group was conducted after the release to ensure the usability and educational requirements were met. This study and its results are presented and discussed in the next chapter.

5 Evaluation

Myth Hunter was implemented in two iterations focusing on the scavenger hunt in the first iteration and on the card game in the second. After the first iteration, a qualitative study was conducted to get first insights in the application's applicability for students and learning scenarios. After the second implementation iteration was complete including all the features described in Chapter 3 respectively 4, a second study was conducted to evaluate the performance and usability of the application by the target group of secondary school children. These two evaluations are explained and discussed in this chapter.

5.1 Prototype Evaluation

The prototype evaluation was conducted early in the implementation phase after most of the scavenger hunt features were implemented with only a spartan user interface. The purpose of this evaluation was to get early feedback to the main concept.

5.1.1 Method Description

This evaluation was done over a period of two weeks with always maximum two participants at a time. This allowed to closely observe how the participants were using the application and minimized the possibility of one participant just following another. All participants hat to complete the same quest guiding them through the old town of the city Graz in Austria. The quest required approximately 1.5 km of walking and consisted of seven tasks including Information, Quiz, and Find-tasks. To complete the Quiz-tasks the users had to search for information in the nearby area on their own.

5.1.2 Material

After completing the quest, the participants were asked to fill out a survey. This survey was roughly split into three parts: (1) personal information and experience, (2) game experience and (3) estimation.

Game experience: This section of the survey focused on their experience with app elements such as fun, motivation, learning effect, easy use and safety risks.

It mostly used Likert-scaling questions between 1 (not at all) and 5 (very strong). The statements used were:

- I had fun doing the scavenger hunt.
- I felt motivated to complete the whole quest.
- I learned something about the history of the Schlossberg.
- I prefer to learn about the history of Graz using the app to reading a book about it.
- I think my ability to use a compass/map increased.
- At any point during the test, I knew where I had to go next.
- It was easy for me to navigate with the app.
- I lost the sense of my surroundings while using the app.
- I think using the app can bring security risks with it (tripping over/ running into something)

Estimation: The last section focused on how the participants would estimate children reacting to the app. Using the same Likert-scaling as above it stated:

- I think children (7-14) would have fun learning with the application.
- I think children (7-14) would prefer using the app to learn over reading a school book.
- I think children would also use the app after school in their free time.
- I think that using the application would pose a security risk for children (e.g., in traffic).
- I think the app is too complex for children to use.

5.1.3 Participants

The study included eight participants (7 female) between 20-35 years old (M=24.63, SD=3.46) with little experience regarding mobile games and next to no experience with location-based games. *"On a Likert scale between 1 (not at all) and 5 (very strong) they rated their experience with mobile games with a mean of 2.88 (SD=0.83) and their experience with location-based mobile apps (mainly Google Maps) with 2.25 (SD=0.89). Asking them about their skills to navigate with a map/compass the average mean was 3.5 (SD=0.93)"* (Hutzler, Wagner, Pirker, & Gütl, 2017).

5.1.4 Results

"The overall Feedback was very positive: 75% of the participants answered the questions about the fun of the game and the motivation to finish the quest with five out of five points. Especially exploratory and narrative story-based elements were noted as motivating" (Hutzler et al., 2017). Only one of the testers would prefer reading a book about the history of Graz to learning with the app.

The results regarding the learning effect were very mixed mostly depending on the knowledge they already had about the area prior the study. " *87.5% answered with four or more points out of five when asked how easy it was to navigate with the app and if they always knew where to go next* "(Hutzler et al., 2017). The participants were asked to estimate how children would react to the application. All participants expected that kids would have fun executing such a quest and prefer it to traditional educational practices of the same content. The question about the likelihood of pupils using the app in their free time as well scored an arithmetic mean of 3.38 out of five points indicating that the gamification elements of our prototype were not enough to keep players motivated.

Finally, we raised the question about safety risks for children while using the app. 75% rated this question with three or more points of a total of five indicating that the risk of children forgetting about their surroundings, e.g., traffic is perceived rather high. Additional comments of the participants included that, depending on the age of the children, the representation of information might need to be less text and more graphic focused. This lead to introducing different quest difficulty levels in the final release. Several times during the study GPS problem occurred leading to users not being able to open a task despite being at the right location. This lead to the decision of implementing the GPS functionality in a native android plugin rather than using Unity's built-in mechanism.

The quest also included one Find-task which apparently wasn't described clearly enough in the task description. Some participants didn't know what to do next or were confused about the marker staying on the map while all other diapered after opening.

5.2 Release Evaluation

After all features were implemented and tested a release evaluation with our target group of secondary school children was conducted.

5.2.1 Method Description

The evaluation of the Myth Hunter application was done during a field trip of a secondary school class to the city of Graz. One day prior the field trip the application was installed on their private phones, and a pre-questionnaire about their experience regarding computer and smartphone games as well as street-map usage was filled out. Afterwards, they were allowed to play the tutorial.

During the field trip, the participants formed groups of three and had to complete a quest, designed by their teachers, using the Myth Hunter Android application. The quest required approximately 1.5 km of waking and consisted of eight stations including all available types (Information, Quiz, Find and Fight-markers) leading them through the old town of Graz giving information about important places and buildings. During the scavenger hunt, the participants had to fill out a questionnaire similar to van Dijk's adaptation of the Intrinsic Motivation Inventory (IMI) suitable for children (van Dijk, Lingnau, & Kockelkorn, 2012).

After all groups had completed the quest, they were given a post-questionnaire asking about their experience with the application focusing on usability, assessability, motivation, and satisfaction.

5.2.2 Material

To evaluate the application three questionnaires were used, one prior the experiment, one during the scavenger hunt and the last one after the experiment was completed. The first consisted of seven questions about their background like age, gender and their experience with games and roadmaps.

The second questionnaire included eight questions using a Likert scaling between 1 (strongly disagree) and 5 (strongly agree). The questions were based on the Intrinsic Motivation Inventory (IMI) created by Ryan and Deci respectively van Dijk's adaptation for children. These adjustments include reducing the total number of questions and reversing negatively formulated statements (van Dijk et al., 2012). The statements used in the Myth Hunter evaluation are:

- I'm enjoying the scavenger hunt very much.
- The scavenger hunt is fun to do.
- I think the scavenger hunt is exciting.
- I'd like to continue the scavenger hunt.
- The scavenger hunt holds my attention very well.
- I would describe the scavenger hunt as very interesting.
- I think this scavenger hunt is quite enjoyable.
- While I was doing the scavenger hunt, I was thinking about how much I enjoyed it.

In the post-questionnaire, ten questions were raised including open-ended questions as well as Likert scaling like in the second questionnaire. The participants are asked what they liked and disliked about the application or the activity in general. Also, two questions regarding the learning performance were included in the open-ended questions. One asked about a specific piece of information given during the scavenger hunt while on the second participants could openly state what they have learned. The questions using the Likert scaling aimed at the usability of the application and willingness to use Myth Hunter again:

- I had fun doing the scavenger hunt.
- I would like to use this app in class more often.
- I think, I would use this app also in my free time.
- I think the app is easy to use.
- I always knew what I had to do next.

Figure 5.1: The Smileyometer by Read and MacFarlane (2006)

- It was easy to find the target locations in the real world.

To make the questionnaires more suitable for children the ”Smileyometer” (see Figure 5.1) taken from the ”Fun Toolkit” was used for all Likert scaling questions (Read & Macfarlane, 2002).

5.2.3 Participants

The study was done in cooperation with a first class of a secondary school near Graz. It included 17 participants (12 female) aged 10 to 12 with mixed experience levels regarding computer and smartphone games. On a Likert scale between 1 (never) and 5 (very often) they rated their use of computer games with a mean of 3.29 (SD=1.16) but when asked about smartphone games 70.59% answered with four or more points out of five. The question about their experience with road maps resulted in a mean of 2.94 (SD=1.14) out of five points with only two participants characterizing them as experts using road maps. Out of the 17 participants, five have already done a digital scavenger hunt before, but all of them stated to be very much looking forward to trying the application.

5.2.4 Results

This section presents the results of the Myth Hunter evaluation. In the first part, general observations during the experiment are described and discussed. Afterwards, the questionnaires are analyzed.

Experiment

The experiment can roughly be split into two parts; tutorial and scavenger hunt.

Tutorial: As mentioned before, the Android application was installed on the participant's phones one day before the field trip to Graz. The application could not be installed for all participants. Roughly one third had either no phone present or the application was not compatible with the operating system. After the installation, their task was to register and complete the Tutorial on their own. The application checks if GPS is enabled upon start and tells the user if not but does not check for internet connection. Some participants had either disabled mobile data or had reached their data limits which resulted in a cryptic error message upon registering. Thankfully the school provided wireless LAN which allowed these participants to continue the experiment. The first negative point the users mentioned was a quite long loading time after registering till the map was shown. During this time the application loads the initial deck and all its cards for the first time. This takes some time in perfect conditions, but due to the unstable internet connection in the classroom as well as the servers limitation, it took significantly longer. Some of the participants grew inpatient and restarted the application during the process and tried to re-register afterwards which led to an error message because the user was already created the first time. Logging in with this username, instead of registering, worked quickly because no cards are loaded. However, to play the Tutorial the cards had to be loaded again which resulted in the same loading time problem as during register.

After some restarts and waiting all participants who had the application on their phones were able to play the tutorial. During the tutorial, it was observable that some of the participants didn't bother to read the explanations trusting that the application will give enough visual hints to complete the tasks. This was not the case, so participants got stuck and didn't know what to do next. After verbally explaining and demonstrating the mechanism to play a card most of the participants were able to complete the tutorial on their own.

Scavenger Hunt: At the beginning of the field trip, the class was split into groups of three ensuring that each group had at least one smartphone with Myth Hunter installed. Their task was to complete a quest with various kinds of different stations starting with a quiz task. All groups were able to find the needed information about the steps up the "Schlossberg" rather quickly but due to the following informational screen somewhere misled and thought they had to go up the stairs. Additionally, the next task was a Find-task. Therefore, no point on the map told them where to go next, just a riddle in the task description and the title gave a clue that the hidden location is on the main square. The groups which misunderstood the hints or didn't read the description well enough had to be told that up the stairs was the wrong direction.

With some help, the groups found the "Weikhard-Watch" and afterwards the "Erzherzog-Johann" fountain. It was observable, however, that some groups had problems regarding the accuracy of the GPS signal especially in narrow streets which reduced the helpfulness of the "Distance-To-Target" indicator. Another thing we noticed was that some groups didn't read the, sometimes quite extensive, information of the targets carefully but rather skipped the text and moved on to the next task. A positive aspect was that each group tried to solve the tasks on their own rather than checking what the other groups are doing. At the fountain, a break was planned and the second questionnaire was filled out.

During the rest of the quest, one group's application stopped working because the mobile data limit was reached. They were able to complete the quest on a different device. We also got complaints that being on the street-map screen a single tap on the hardware-back-button leads to a logout which apparently happened a couple of times unintentionally.

Near the end of the quest, one Fight-task was placed which again caused some frustration due to long loading time. On several devices, the fight didn't load at all within a reasonable time or crashed for unknown reasons. Therefore most groups chose to skip the fight. In the end, only one group managed to complete the Fight-task which shows that the loading time is a big issue which needs to be addressed.

Questionnaires

As mentioned before, the participants had to fill out three questionnaires. The results of the pre-questionnaire were presented in section 5.2.3.

Intermediate-Questionnaire: The purpose of the Intermediate-questionnaire was to measure enjoyment during the scavenger hunt. The results of this questionnaire were very positive. Not a single statement scored an arithmetic mean below four out of five points. For example, when asked if they are enjoying the scavenger hunt 88.24% answered with five out of five points. Also, the question if the participants would like to continue the scavenger hunt got four or more points from each participant. Table 5.1 shows the arithmetic mean and standard derivation for each question of the Intermediate-questionnaire.

Statement	arithmetic mean	standard derivation
I'm enjoying the scavenger hunt very much.	4.88	0.33
The scavenger hunt is fun to do.	4.82	0.39
I think the scavenger hunt is exciting.	4.59	0.71
I'd like to continue the scavenger hunt.	4.53	0.51
The scavenger hunt holds my attention very well.	4.35	0.49
I would describe the scavenger hunt as very interesting.	4.59	0.62
I think this scavenger hunt is quite enjoyable.	4.76	0.44
While I was doing the scavenger hunt, I was thinking about how much I enjoyed it.	4.18	0.95

Table 5.1: Intermediate-questionnaire results

Post-Questionnaire: The questions and statements of the post-questionnaire were supposed to measure perceived usability as well as willingness to use the application again. Furthermore, the application's ability to transfer knowledge was measured.

When asked if they had fun doing the scavenger hunt all participants answered with four or more points out of five. The questionnaire also included open questions about what they liked or did not like about the scavenger hunt. Due to the young age of the participants, these questions were often not answered at all or very briefly. However, some participants stated, that they liked that the activity was done in groups and that they were given the freedom to explore the city on their own guided by the Myth Hunter application. It was especially appreciated, that the scavenger hunt doesn't require to look at the phone all the time. Also, the riddles included in the quest were perceived as very motivating by some participants. Even though we observed that some groups didn't read the information about the targets very careful, one participant stated especially the amount of information as very positive. Only five out of the 17 participants reported negative points about the activity. This statements included the long loading time prior the fight-task as well as walking the same streets more than once. One participant stated that the quest was a bit boring at the beginning but got better till the end.

Regarding the overall usability of the application, 76.47% of the testers answered giving four or more points out of five. The question if they always knew what to do next scored an arithmetic mean of 3.88(SD=0.86) indication that there is still some room for improvement. 88.24% stated, that it was rather easy(four or more points out of five) to find the target locations in the real world. Almost all participants answered that they would like to use the application more often in class resulting in an arithmetic mean of 4.35(SD=1.06). However, the mean for willingness to use Myth Hunter in their free time dropped to 3.94(SD=1.03) indication that either the usage time was too short or the reward system is not motivation enough. Also, the fact that only one group was able to complete the card game could be an explanation because the reward system focuses on the card game. The results of all Likert-Scale questions of the post-questionnaire are shown in Table 5.2.

Part of the post-questionnaire were two questions regarding what the testers have learned. One of them asked specifically for the name of a visited fountain. A lot of information about the fountain and an including statue of Archduke John of Austria was given after the location was found. However only roughly half of the kids (47.06%) were able to answer the question with some help of their teacher. The second question was an open one asking what else they have learned during the scavenger hunt. The question was intended to aim at other information they could remember but was not worded precisely enough. Therefore some of the children's answers were more general like "Teamwork" or how to navigate through Graz but also some pieces of information were included. Interestingly most of this information pieces were given in the task descriptions or as part of a riddle rather than in the actual information-texts.

Statement	arithmetic mean	standard derivation
I had fun doing the scavenger hunt.	4.82	0.39
I would like to use this app in class more often.	4.35	1.06
I think, I would use this app also in my free time.	3.94	1.03
I think the app is easy to use.	4.12	0.93
I always knew what I had to do next.	3.88	0.86
It was easy to find the target locations in the real world.	4.24	0.66

Table 5.2: Post-questionnaire results

5.3 Discussion

All in all the release evaluation went quite well for the Myth Hunter application. In the Intermediate as well as in the Post-questionnaire the participants stated that they liked the application and had fun doing the scavenger hunt. Also, the usability could be increased compared to the prototype evaluation, especially for Find-tasks. However, the Application still focuses on textual description and explanation of features which is not as suitable for children as expected. Especially during the Tutorial, it was noticeable that kids tend to skip textual descriptions. Also during the scavenger hunt participants skipped informational text content but because this content was not directly needed to complete the quest it had no impact on the tester's opinions towards the game. The Post-questionnaire, however, revealed that the participants could not answer questions about the presented information either because they skipped the informational content or could not remember it. On the other hand, pieces of information given in the task description or within a riddle were remembered significantly better. This leads to the assumption that if the purpose of a quest is to transfer information or knowledge it must be directly included in the task. Or in other words, the user must need this information to continue the quest either by searching for it on their own and typing it into a Quiz-task or as part of a Find-task description. Additional information for interested users can be given after the task. Summarized: the evaluation revealed that Myth Hunter, in general, is capable of transferring knowledge but the extend is dependent on the structure of the quest at hand.

Participants stated that they learned how to navigate with a map and also now know the location of several places of Graz. Therefore the Myth Hunter application is a useful tool to make users familiar with new sites in a playful way or teach them how to navigate with a map on their own. However, we noticed that the participants tended to look on their phones rather than an on the streets at the beginning of the scavenger hunt which can be dangerous in areas with heavy traffic. Therefore especially young children should always be accompanied by an adult when using the application.

6 Lessons Learned and Future Work

This chapter describes lessons learned during the realization of this master thesis. It includes new insights in the domain learning through games gained during the literature study as well as experiences made in the design and implementation phase. Finally, the evaluation outcome and possible enhancements of the software product are discussed.

6.1 Theory

In the chapter 2 Background and Related Work of this thesis different approaches of computer-aided teaching were introduced. Focusing on educational location-based games several prototypes were analyzed, and their strengths and weaknesses discussed. One problem many of these prototypes faced was the inaccuracy of the device localization using GPS especially in urban areas with high buildings. Prototypes with similarities to Myth Hunter were found, but in most cases, an evaluation towards user acceptance as well as learning efficiency was missing.

To choose an appropriate AI algorithm for Myth Hunter different algorithms were examined according to their applicability for a Collection Card Game. Traditional AI algorithms are very efficient for games of total information but lack effective modifications for games of hidden information or chance events. In the end, Information Set Monte Carlo Tree Search was chosen.

6.2 Development

In this project, a lot of tools and frameworks were used to implement the required functionality. When using tools and especially game engines like Unity, it is essential to get a good overview of the functionalities a tool offers before starting to develop an own application. This functionality research was not done at the beginning of the Myth Hunter implementation phase and some features of Unity or other tools were only discovered in the middle of development. This resulted in several design changes, and some parts of the software were rewritten multiple times during the development.

When looking for new tools respectively frameworks to implement a specific functionality, it is also favorable to check if there exists an active community around them. Getting to know a new environment without proper documentation and the possibility to ask questions can be quite tiresome and time-consuming.

When implementing parts of a software product independently, it is beneficial to test the communication and compatibility of these parts as soon as possible. During the Myth Hunter implementation most of the Android Application tests where done using mock-up test data rather than actual data provided by the Editor which led to misconceptions and additional code changes once the two parts were tested together. Especially the framework PowerUI which was responsible for rendering HTML pages created by the editor on the client device had some restrictions which were not known before and had to be fixed late in the project.

6.3 Evaluation

The questionnaires used for the evaluation were based on the Intrinsic Motivation Inventory as well as on the System Usability Scale. The amount of questions, however, was greatly reduced to fit for children. Also the standard point Likert scale was dismissed for a more child-suitable scale using smileys.

The evaluation showed that Likert scale questions are suitable for children aged 10 to 12, but open-ended questions should be raised in a face-to-face interview rather than on paper because children in this age tend to skip such question or only give concise answers.

The results of the evaluation show that Myth Hunter succeeded in motivating the participants and was rather easy to use. However, the teaching potential of Myth Hunter depends heavily on the right structure of the quest and the information placement within the quest at least for children aged 10 to 12. The secondary aim of Myth Hunter, increasing navigation skills and teaching the location of specific places within an area, was achieved according to the statements of several participants.

6.4 Possible Enhancements

App Modifications: Myth Hunter needs to reduce the amount of textual description in favor of short visual clues to explain the functionalities to children. Especially within the card game tutorial, the upcoming steps a user has to take to continue must be presented in a visual way like short video clips for example. Because quests and tasks are user-generated content ensuring that videos and pictures are used instead of masses of text is not possible. However, to support young children, a text-to-speech option could be included as well.

Another big issue identified during the evaluation was the long loading times prior a Fight-task. One possibility to overcome this weak spot would be to locally store the cards and their images in a database on the client device once they are loaded. Such a local database would also allow storing all quest data upon starting the quest. A quest could then be executed offline, and loss of internet connection would not block the quest execution. Map-tiles for the quest area need to be stored offline in this case as well.

To further motivate users to collect new cards and strengthen their deck by executing more quests a player-versus-player mode can be added to allow gamers to compete against each other.

Server Modifications: For a release open for public usage security measures must be taken. Each web service call must include username and password and must be evaluated according to the user's rights before executing it. For example altering a *QuestInstance* must be only allowed for the user who owns this particular *QuestInstance*. Also, access to the Linux server itself must be restricted using only SSH connections.

Further studies: To further improve Myth Hunter's usability and educational possibilities further studies should be conducted. So far the release version was only evaluated using children aged 10-12. Evaluations of Myth Hunter with different age groups would give more insight in its applicability for a broader target group. Especially because older children or even adults tend to answer open-ended questions more willingly, further enhancement possibilities could be identified.

7 Summary

Prensky (2003), as well as Kuo (2007) and many others, identified that motivation is of vital importance when aiming to educate people. Furthermore, they stated that children, as well as adults, are usually very motivated to play. Therefore modern education should seek to interweave learning and playing. With the rise of computer usage learning software and dedicated learning games emerged. Especially with the resent upcome of smartphones and their location awareness lots of new possibilities to connect educational content playfully with physical locations are available. Several educational scavenger hunt applications were introduced recently, but most of these games treat their quests independently and therefore lack incentives to finish further quests.

The location-based learning game Myth Hunter tries to solve this incentives-problem by merging an educational scavenger hunt with a collection card game in which players earn rewards in the form of additional cards for each finished quest. Myth Hunter consists of an online-editor which allows the creation of new quests and cards, a server which stores all quests and other needed data and an android application to execute the scavenger hunts and play the card game. In this thesis, the design, implementation, and evaluation of server and Android application were presented. A quest in Myth Hunter can consist of several different task types including Quiz- and Find-tasks as well as Fight-tasks which require the player to defeat an enemy by winning a card game. In this tasks, educational content can be linked to physical locations and objects allowing players not only to read about a subject but actually engage with it. This content can include text as well as pictures or even short video clips and hyperlinks into the web.

The implementation of Myth Hunter was split into two iterations each followed by an evaluation. The first iteration focused on the scavenger hunt without the card game. Its evaluation was a qualitative study with eight participants aiming to get early feedback regarding the main concept and usability suggestions. The overall feedback regarding fun and motivation was very positive, but concerns about safety risks especially for children were raised. The results concerning the learning effect were very mixed depending on the knowledge the participants already had about the presented subject. Additionally, some technical problems like inaccuracy of the GPS signal were observed during the study.

In the second iteration, all features concerning the card game were implemented followed by a release evaluation was conducted. This study was done with the help of a secondary school class as part of a field trip to Graz. The participants had to execute quest in groups of three leading them through the old town presenting information about particular places and objects. Prior, during and after the scavenger hunt the participants were asked to fill out questionnaires each focusing on different aspects, like the participant's background, fun and motivation during the scavenger hunt and usability as well as learning effect after the last task. Again the results concerning fun and motivation were very positive however some participants tent to skip informational content which resulted and weak learning effectiveness. The study revealed that the learning effect heavily depends on the right placement of educational content within the quest. Additional information after a task was finished tends do be skipped especially if it solely consists of text. However, facts placed within a riddle or as part of the task itself were remembered significantly better. Overall Myth Hunter was perceived as a fun and motivating scavenger hunt game with educational potential if the quest creation was done right, but it still has potential for improvements.

Appendix A

Content on the DVD

- Prototype Evaluation
  - Questionnaire
  - Results
- Release Evaluation
  - Questionnaire
  - Results
- Myth Hunter APK
- Myth Hunter Unity project
- Server sources
- Thesis Latex project

Bibliography

Abramson, B. (1990, February). Expected-outcome: a general model of static evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(2), 182–193. doi:10.1109/34.44404

Abt, C. C. (1970). *Serious games* (First). New York: Viking Press.

Adinolf, S. & Turkay, S. (2011). Collection, creation and community: a discussion on collectible card games. In *Proceedings of the 7th international conference on games + learning + society conference* (pp. 3–11). GLS'11. Madison, Wisconsin: ETC Press. Retrieved from http://dl.acm.org/citation.cfm?id=2206376.2206377

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002, May). Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* *47*(2-3), 235–256. doi:10.1023/A:1013689704352

Avouris, N. & Yiannoutsou, N. (2012, August 1). A review of mobile location-based games for learning across physical and virtual spaces. *j-jucs*, *18*(15), 2120–2142. http://www.jucs.org/jucs_18_15/a_review_of_mobile.

Ballard, B. W. (1983, September). The *-minimax search procedure for trees containing chance nodes. *Artif. Intell.* *21*(3), 327–350. doi:10.1016/S0004-3702(83)80015-0

Benford, S., Anastasi, R., Flintham, M., Greenhalgh, C., Tandavanitj, N., Adams, M., & Row-Farr, J. (2003, July). Coping with uncertainty in a location-based game. *IEEE Pervasive Computing*, *2*(3), 34–41. doi:10.1109/MPRV.2003.1228525

Bouzy, B. & Cazenave, T. (2001, October). Computer go: an ai oriented survey. *Artif. Intell.* *132*(1), 39–103. doi:10.1016/S0004-3702(01)00127-8

Brosda, C., Bartsch, S., Oppermann, L., & Schaal, S. (2016). On the use of audio in the educational location based game platform mile. In *Proceedings of the 18th international conference on human-computer interaction with mobile devices and services adjunct* (pp. 1049–1054). MobileHCI '16. Florence, Italy: ACM. doi:10.1145/2957265.2964198

Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P., Rohlfshagen, P., . . . Colton, S. (2012, March). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, *4*(1), 1–43. doi:10.1109/TCIAIG.2012.2186810

Bursztein, E. (2016, September). I am a legend: hacking hearthstone using statistical learning methods. In *2016 ieee conference on computational intelligence and games (cig)* (pp. 1–8). doi:10.1109/CIG.2016.7860416

Campbell, M. & Marsland, T. (1983, July). A comparison of minimax tree search algorithms. *Artificial Intelligence*, *20*(4), 347–367. doi:10.1016/0004-3702(83)90001-2

Chen, P., Chang, M., Kuoc, R., & Heha, J.-S. (2016). Trading card game.

Chen, P., Kuo, R., Chang, M., & Heh, J.-S. (2009). Designing a trading card game as educational reward system to improve students' learning motivations. In Z. Pan, A. D. Cheok, W. Müller, & M. Chang (Eds.), *Transactions on edutainment iii* (pp. 116–128). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-11245-4_11

Chen-Lin Kulik, J. K. (1991). Effectiveness of computer-based instruction: an updated analysis. *Computers in Human Behavior*, *7*(1-2), 75–94.

Chess, S. (2014). Augmented regionalism: ingress as geomediated gaming narrative. *Information, Communication & Society*, *17*(9), 1105–1117. doi:10.1080/1369118X.2014.881903. eprint: http://dx.doi.org/10.1080/1369118X.2014.881903

Coulom, R. (2007). Efficient selectivity and backup operators in monte-carlo tree search. In H. J. van den Herik, P. Ciancarini, & H. H. L. M. (Donkers (Eds.), *Computers and games: 5th international conference, cg 2006, turin, italy, may 29-31, 2006. revised papers* (pp. 72–83). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-75538-8_7

Cowling, P. I., Powley, E. J., & Whitehouse, D. (2012, June). Information set monte carlo tree search. *IEEE Transactions on Computational Intelligence and AI in Games*, *4*(2), 120–143. doi:10.1109/TCIAIG.2012.2200894

Cuban, L. (2001). *Oversold and underused: computers in the classroom*. Harvard Univerity Press.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th international academic mindtrek conference: envisioning future media environments* (pp. 9–15). MindTrek '11. Tampere, Finland: ACM. doi:10.1145/2181037.2181040

Edward Powley, D. W. & Cowling, P. (2013). Reducing the burden of knowledge: simulation-based methods in imperfect information games. http://www.aifactory.co.uk/newsletter/2013_01_reduce_burden.htm. (Accessed on 05/12/2017).

Ejsing-Duun, S. (2011). *Location-based games* (Doctoral dissertation, Aarhus Universitet).

Enzenberger, M. (2004). Evaluation in go by a neural network using soft segmentation. In H. J. Van Den Herik, H. Iida, & E. A. Heinz (Eds.), *Advances in computer games: many games, many challenges* (pp. 97–108). Boston, MA: Springer US. doi:10.1007/978-0-387-35706-5_7

Facer, K., Joiner, R., Stanton, D., Reid, J., Hull, R., & Kirk, D. (2004). Savannah: mobile gaming and learning? *Journal of Computer Assisted Learning*, *20*(6), 399–409. doi:10.1111/j.1365-2729.2004.00105.x

Foundation, F. S. (2017). Gnu lesser general public license. http://www.gnu.org/licenses/lgpl.html. (Accessed on 05/18/2017).

Gentile, D. (2009). Pathological video-game use among youth ages 8 to 18. *Psychological Science*, *20*(5), 594–602. PMID: 19476590. doi:10.1111/j.1467-9280.2009.02340.x. eprint: https://doi.org/10.1111/j.1467-9280.2009.02340.x

Goes, L. F. W., da Silva, A. R., Rezende, J., Amorim, A., Franca, C., Zaidan, T., . . . Martins, C. A. P. S. (2016). Honingstone: building creative combos with honing theory for a digital card game. *IEEE Transactions on Computational Intelligence and AI in Games*, *PP*(99), 1–1. doi:10.1109/TCIAIG.2016.2536689

Henry Becker, Y. W., Jason Ravitz. (1999, November). *Teacher and teacher-directed student use of computers and software. teaching, learning, and computing: 1998 national survey.* (tech. rep. No. 3). Center for Research on Information Technology and Organizations, Irvine, CA. Minnesota Univ., Minneapolis.

Hong, T.-P., Huang, K.-Y., & Lin, W.-Y. (1998, May). A genetic minimax game-playing strategy. In *1998 ieee international conference on evolutionary computation proceedings. ieee world congress on computational intelligence (cat. no.98th8360)* (pp. 690–694). doi:10.1109/ICEC.1998.700123

Huang, C. Y. (2016, December). An innovative proposal for young students to learn computer science and technology through pokemon go. In *2016 international conference on computational science and computational intelligence (csci)* (pp. 246–251). doi:10.1109/CSCI.2016.0055

Huizenga, J., Admiraal, W., Akkerman, S., & ten Dam, G. (2009, August). Mobile game-based learning in secondary education: engagement, motivation and learning in a mobile city game. *Journal of Computer Assisted Learning*, *25*(4), 332–344. Retrieved from https://www.learntechlib.org/p/103931

Hutzler, A. (2017, under review, November). *Design and implementation of a web-based authoring tool for location-based gamified applications* (Master's thesis, Technical University Graz).

Hutzler, A., Wagner, R., Pirker, J., & Gütl, C. (2017). Mythhunter: gamification in an educational location-based scavenger hunt. In D. Beck, C. Allison, L. Morgado, J. Pirker, F. Khosmood, J. Richter, & C. Gütl (Eds.), *Immersive learning research network: third international conference, ilrn 2017, coimbra, portugal, june 26–29, 2017. proceedings* (pp. 155–169). Cham: Springer International Publishing. doi:10.1007/978-3-319-60633-0_13

Jeff Craighead, R. M., Jenny Burke. (2008). Using the unity game engine to develop sarge: a case study. In *Proceedings of the 2008 simulation workshop at the international conference on intelligent robots and systems*. IROS 2008.

Jez9999, W. (2007). Alpha-beta pruning example. https://commons.wikimedia.org/w/index.php?curid=3708424. (Accessed on 05/05/2017).

Kaindl, H. (1988, September). Minimaxing: Theory and Practice. *AI Magazine*, *9*(3), 69. Retrieved May 11, 2017, from https://aaai.org/ojs/index.php/aimagazine/article/view/944

Kim, E. B. & Eom, S. B. (2002). Designing effective cyber store user interface. *Industrial Management & Data Systems*, *102*(5), 241–251. doi:10.1108/02635570210428276. eprint: http://dx.doi.org/10.1108/02635570210428276

Knuth, D. & Moore, R. (1975, December). An analysis of alpha-beta pruning. *Artificial Intelligence*, *6*(4), 293–326. doi:10.1016/0004-3702(75)90019-3

Kocsis, L. & Szepesvári, C. (2006). Bandit based monte-carlo planning. In *Proceedings of the 17th european conference on machine learning* (pp. 282–293). ECML'06. Berlin, Germany: Springer-Verlag. doi:10.1007/11871842_29

Konami. (1996). Yu-gi-oh! trading card game official rulebook. http://www.yugioh-card.com/uk/rulebook/Rulebook_v9_en.pdf. (Accessed on 05/05/2017).

Konami. (2009). Yu-gi-oh! card sales set new world record. http://www.konami.jp/topics/2009/0807/index-e.html. (Accessed on 05/05/2017).

Kuo, M. J. (2007, March). How does an online game based learning environment promote students' intrinsic motivation for learning natural science and how does it affect their learning outcomes? In *2007 first ieee international workshop on digital game and intelligent toy enhanced learning (digitel'07)* (pp. 135–142). doi:10.1109/DIGITEL.2007.28

Lary, L. M. (2004). Hide and seek gps and geocaching in the classroom. *Learning and Leading with Technology*, *31*(6), 14–18.

Lin, K. C. & Wei, Y. C. (2011, July). Online interactive game-based learning in high school history education: impact on educational effectiveness and student motivation. In *2011 fourth international conference on ubi-media computing* (pp. 265–268). doi:10.1109/U-MEDIA.2011.22

Liu, S. W. & Wang, C. H. (2016, July). Exploring the effects of discovery learning strategy for location-based historic retrospection mobile game. In *2016 ieee 16th international conference on advanced learning technologies (icalt)* (pp. 95–99). doi:10.1109/ICALT.2016.65

Mahadevan, S. (1997). Monte carlo simulation. *MECHANICAL ENGINEERING-NEW YORK AND BASEL-MARCEL DEKKER-*, 123–146.

Maria Virvou, G. K. & Manos, K. (2005). Combining software games with education, evaluation of its educational effectiveness. *Educational Technology and Society*, *8*(3), 54–65.

Matherson, L., Wright, V., Inman, C., & Wilson, E. (2008). Get up, get out with geocaching: engaging technology for the social studies classroom. *Social Studies Research and Practice*, *3*(3), 80–85.

Melero, J. & HernÁndez-Leo, D. (2017, January). Design and implementation of location-based learning games: four case studies with questinsitu: the game. *IEEE Transactions on Emerging Topics in Computing*, *5*(1), 84–94. doi:10.1109/TETC.2016.2615861

meople's magazine. (2011). Collectible card games. http://www.meoplesmagazine.com/2011/07/21/collectible-card-games/. (Accessed on 05/05/2017).

Meschtscherjakov, A., Trösterer, S., Lupp, A., & Tscheligi, M. (2017). Pokémon walk: persuasive effects of pokémon go game-design elements. In P. W.

de Vries, H. Oinas-Kukkonen, L. Siemons, N. Beerlage-de Jong, & L. van Gemert-Pijnen (Eds.), *Persuasive technology: development and implementation of personalized technologies to change attitudes and behaviors: 12th international conference, persuasive 2017, amsterdam, the netherlands, april 4–6, 2017, proceedings* (pp. 241–252). Cham: Springer International Publishing. doi:10.1007/978-3-319-55134-0_19

Microsoft. (2017). Web services description language-tool. https://msdn.microsoft.com/de-de/library/7h3ystb6(v=vs.100).aspx. (Accessed on 05/18/2017).

Nic Borg, C. H., Jeff O'Hara. (2017). Edmodo. https://www.edmodo.com/. (Accessed on 04/18/2017).

NIELSEN, J. (2012). Usability 101: introduction to usability. https://www.nngroup.com/articles/usability-101-introduction-to-usability/. (Accessed on 04/18/2017).

Niemiec, R. P., Sikorski, M. F., & Walberg, H. J. (1989). Comparing the cost-effectiveness of tutoring and computer-based instruction. *Journal of Educational Computing Research*, *5*(4), 395–407. doi:10.2190/3RH7-3T7A-4JXU-H6CA. eprint: http://dx.doi.org/10.2190/3RH7-3T7A-4JXU-H6CA

of the Coast, W. (2010). 2010 magic: the gathering world championships top photos. https://magic.wizards.com/en/articles/archive/event-coverage/2010-magic-gathering-world-championships-top-photos-2010-12-11. (Accessed on 05/05/2017).

O'Hara, K. (2008). Understanding geocaching practices and motivations. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1177–1186). CHI '08. Florence, Italy: ACM. doi:10.1145/1357054.1357239

CAN YOU SEE ME NOW? (2017). http://www.blasttheory.co.uk/projects/can-you-see-me-now/. (Accessed on 08/09/2017).

Helbling Verlag. (2017). http://www.helbling.at/. (Accessed on 02/16/2017).

Ingress. (2017). https://www.ingress.com/. (Accessed on 04/20/2017).

Moodle. (2017). https://moodle.org/. (Accessed on 04/20/2017).

Peña-Miguel Noemí, S. H. M. (2014). Educational games for learning. *Universal Journal of Educational Research*, *2*(3), 230–238.

Pirker, J., Gütl, C., Weiner, P., & Garcia-Barrios, V. M. (2015). Application domains for a location-based mobile application creator. *International Journal of Interactive Mobile Technologies*, *9*(3), 52–57.

Pirker, J., Gütl, C., Weiner, P., Garcia-Barrios, V. M., & Tomintz, M. (2014). Location-based mobile application creator creating educational mobile scavenger hunts. In *Interactive mobile communication technologies and learning (imcl), 2014 international conference on* (pp. 160–164). IEEE.

Prensky, M. (2003, October). Digital game-based learning. *Comput. Entertain.* *1*(1), 21–21. doi:10.1145/950566.950596

Rashid, O., Mullins, I., Coulton, P., & Edwards, R. (2006, January). Extending cyberspace: location based games using cellular phones. *Comput. Entertain.* *4*(1). doi:10.1145/1111293.1111302

Read, J. & MacFarlane, S. (2006). Using the fun toolkit and other survey methods to gather opinions in child computer interaction. In *Proceedings of the 2006 conference on interaction design and children* (pp. 81–88). IDC '06. Tampere, Finland: ACM. doi:10.1145/1139073.1139096

Read, J. & Macfarlane, S. (2002). Endurability, engagement and expectations: measuring children's fun. In *Interaction design and children, shaker publishing* (pp. 1–23). Shaker Publishing.

Richard, W., Lisa, B., Feng, F., Leif, O., & Michael, S. (2011). Tidy city: a location-based game for city exploration based on user-created content. In M. Eibl (Ed.), *Mensch und computer 2011: übermedien—übermorgen* (pp. 487–496). München: Oldenbourg Verlag.

Sandford, R. & Williamson, B. (2004). *Racing academy: a futurelab prototype research report*. Bristol: Futurelab.

Schadd, F. C. (2009, December). *Monte-carlo search techniques in the modern board game thurn and taxis* (Master's thesis, Maastricht University).

Schlatter, B. E. & Hurd, A. R. (2005). Geocaching: 21st-century hide-and-seek. *Journal of Physical Education, Recreation & Dance*, *76*(7), 28–32. doi:10.1080/07303084.2005.10609309. eprint: http://dx.doi.org/10.1080/07303084.2005.10609309

Schwabe, G. & Göth, C. (2005). Mobile learning with a mobile game: design and motivational effects. *Journal of Computer Assisted Learning*, *21*(3), 204–216. doi:10.1111/j.1365-2729.2005.00128.x

Sephton, N., Cowling, P. I., Powley, E., Whitehouse, D., & Slaven, N. H. (2014, July). Parallelization of information set monte carlo tree search. In *2014 ieee congress on evolutionary computation (cec)* (pp. 2290–2297). doi:10.1109/CEC.2014.6900583

Sheng, L. Y. (2013, October). Modelling learning from ingress (google's augmented reality social game). In *2013 ieee 63rd annual conference international council for education media (icem)* (pp. 1–8). doi:10.1109/CICEM.2013.6820152

Shiue, Y. M., Hsu, Y. C., & Liang, Y. C. (2016, November). Investigating elementary students' epistemological beliefs, game preference by applying game-based learning to a history course. In *2016 international conference on advanced materials for science and engineering (icamse)* (pp. 460–462). doi:10.1109/ICAMSE.2016.7840358

Sotamaa, O. (2002). All the world's a botfighter stage: notes on location-based multi-user gaming. In *Cgdc conf.*

Spaulding, J. (2012). Technology enhanced learning and education. In V. Wang (Ed.), *Encyclopedia of e-leadership, counseling and training* (pp. 642–655). Hershey. doi:10.4018/978-1-61350-068-2

TANG, F., WANG, B., KANEKO, K., GODA, Y., OKADA, Y., & YAMADA, M. (2017, March). Design and assessment of a location-based game to support english vocabulary learning in university. *Society for Information Technology and Teacher Education International Conference 2017*.

Al-Tarawneh, M. H. (2016). The effectiveness of educational games on scientific concepts acquisition in first grade students in science. *Journal of Education and Practice*, *7*(3), 31–37.

TOTEM. (2011). Tidycity. http://totem.fit.fraunhofer.de/tidycity/. (Accessed on 04/18/2017).

Trammell, A. (2010). Magic: the gathering in material and virtual space: an ethnographic approach toward understanding players who dislike online play. http://meaningfulplay.msu.edu/proceedings2010/mp2010_paper_42.pdf. (Accessed on 05/05/2017).

Traxler, J. (2007). Defining, discussing and evaluating mobile learning: the moving finger writes and having writ . . . . *The International Review of Research in Open and Distributed Learning*, *8*(2). Retrieved from http://www.irrodl.org/index.php/irrodl/article/view/346

Turkay, S., Adinolf, S., & Tirthali, D. (2012). Collectible card games as learning tools. *Procedia - Social and Behavioral Sciences*, *46*, 3701–3705. doi:http://dx.doi.org/10.1016/j.sbspro.2012.06.130

van Dijk, E., Lingnau, A., & Kockelkorn, H. (2012, October). Measuring enjoyment of an interactive museum experience. In *Proceedings of the 14th acm international conference on multimodal interaction, icmi 2012* (pp. 249–256). 10.1145/2388676.2388728. ACM. doi:10.1145/2388676.2388728

Werbach, K. (2014). Persuasive technology: 9th international conference, persuasive 2014, padua, italy, may 21-23, 2014. proceedings. In A. Spagnolli, L. Chittaro, & L. Gamberini (Eds.), (Chap. (Re)Defining Gamification: A Process Approach, pp. 266–272). Cham: Springer International Publishing. doi:10.1007/978-3-319-07127-5_23

Wetzel, R. (2011). Tidy city: learning about cultural heritage through a user-created mobile game. https://ercim-news.ercim.eu/en86/special/tidy-city-learning-about-cultural-heritage-through-a-user-created-mobile-game. (Accessed on 04/20/2017).

Wetzel, R., Blum, L., & Oppermann, L. (2012). Tidy city: a location-based game supported by in-situ and web-based authoring tools to enable user-created content. In *Proceedings of the international conference on the foundations of digital games* (pp. 238–241). FDG '12. Raleigh, North Carolina: ACM. doi:10.1145/2282338.2282385

Whitehouse, D. (2014, August). *Monte Carlo Tree Search for games with Hidden Information and Uncertainty* (phd, University of York). Retrieved May 12, 2017, from http://etheses.whiterose.ac.uk/8117/

William R. Watson, S. L. W. (2007). An argument for clarity: what are learning management systems, what are they not, and what should they become? *TechTrends, Springer Verlag*, *51*(2), 28–34.

Yu, H. H., Yang, J. K., Chen, H. W., Yu, T. F., & Hou, H. T. (2015, July). Jingnan campaign - using game-based assessment with the mechanism of strategy games for history teaching system development and learning evaluation. In

2015 iiai 4th international congress on advanced applied informatics (pp. 727–728). doi:10.1109/IIAI-AAI.2015.199

Zopf, M. (2015, April). *A comparison between the usage of flat and structured game trees for move evaluation in hearthstone* (Master's thesis, Technische Universität Darmstadt).