Designing a Context-Aware Campus Area Gaming Environment for Mobile Platforms

Andrii Sereda

Kongens Lyngby 2013

IMM-MSc-2013-9

Technical University of Denmark
Informatics and Mathematical Modeling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk
IMM -MSC: 2013-9

Summary

Generation of people author of this thesis belongs to is one of the first generations that mature together with the industry of computer games. Computer games have already become a significant part of our lives and different game-worlds continue merging with reality. Different aspects of computer game-mechanics and game-elements are something that is mastered since young age. This generation author of this thesis belongs to was maturing in a merged digital-analog reality that penetrates all aspects of day-to-day activities in a modern digitized world. However, younger generations are already born in this world. This new reality (or realities) that is being created for them now, will be the only reality they will know from their birth. Most of kids that will be born in big developed European, American, Asian and African cities will never know a reality without computer-games.

This thesis is describing one particular kind of modern computer games, appearance of which is closely related with development of portable mobile communication devices. This thesis is dealing with what have been identified as “pervasive games”. Investigation and analysis of reasons behind their appearance, popularization is made. Different trends within classical game-development and theory of classical game-design are investigated to establish similarities and differences, challenges and solutions this newly emerged type of games is introducing.

Knowledge presented in theoretical sections is used to design, develop, deploy and test a game-environment for mobile platforms that is adjusted for campus area.

This designed game-environment consists of two essential components: Front-End, which is represented by Android mobile application and Back-End server logic. Both of components were designed, tested and implemented. Designed game-environment received a name “DTU GoblinsNGold and has been launched. It is made

available publicly worldwide through Google Play market.

All of these aspects and milestones are described in details inside practical part of this thesis.

During game-environment testing valuable data was received that described all aspects of functionality of implemented system together with behavioral patterns of players. These results of deployment together with conclusions are presented in later sections.

Acknowledgments

I would like to thank Jakob Eg Larsen and Sune Lehmann Jørgensen for all their assistance during this project.

I would like to give particular thanks to Aleksandr Egorov, Radu-Vlad Acretoaie, Łukasz Dynowski, Marcos Fuentes, Jorge Perez. Thank You, for your inspiration, enthusiasm and all these useful advices You've shared during these two years. Often your "Hey how are You?" kept my batteries charged.

I owe thanks to all players who responded to my open call for testers. Especially, Massimiliano Curzi.

I would also like to thank Redshrike from Pixel-joint for his set of free goblin pictures I was using and minecraft community for developing free element textures that I used to create my graphical resources.

There is no way I can express my gratitude to my parents. They don't speak English that well, so I will have to translate this sentence for them later.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling (IMM) at the Technical University of Denmark in fulfillment of requirements for acquiring an M.Sc. in Telecommunication.

Work on this thesis was conducted under the supervision of Associate Professor Jakob Eg Larsen and Associate Professor Sune Lehmann Jørgensen.

The thesis deals with subject of pervasive games. More specifically, design of a context-aware game environment adjusted for campus area and designed for mobile platforms is performed.

This game-environment was designed based on theoretical research performed in Chapter 1. Design methodology and design decisions made are discussed in Chapter 2. Chapter 3 describes implementation stage of design together with all decision-changes and iterations. Chapter 4 presents the results of game-environment launch and their assessment.
Conclusions and possible future developments are presented in the concluding sections.

Lyngby, 28-February-2013

[illegible]

Andrii Sereda

Contents

A lot of great things start with File-New project...

Introduction

Initial idea and goal of this thesis is designing a context-aware game-environment that students, workers and visitors of campus area (DTU in context of this thesis) would enjoy, find "fun" and use.

This game environment should be designed for mobile platforms (smartphones in context of this thesis).

Objectives and goals of this thesis are:

- Analysis of concepts of pervasive gaming and game-theory
- Design of context-aware gaming environment for campus area
- Implementation of designed context-aware gaming-environment in real life
- Launch of working implementation and making it public
- Collecting and performing analysis of data that is related with the usage of designed game-environment.

Ideally, designed gaming environment should be interesting, be self driven and exist in parallel with study process without disturbing it, but filling in the time gaps in it.

Ideally, this designed game-environment should not require constant attention or control from any kind of "narrator" and should not be limited in time. However, it should be closely associated to campus area.

Player-to-player interactions should be supported.

Achieving of objectives would represent successful finish of this project.

Chapter 1

Related work

Motivation for playing games

This section will introduce frameworks within games-science, game-research and game-design that present theoretical concepts games and theory of what makes people play games.

As it is noted by different researchers it is not so easy to define game as a concept. This would include discussing rules, elements and players. It is often much easier to define what is a game and what is not a game. Yet, one of the definitions made by Jesse Schell, a famous game designer is as simple as “a game is something we play”.[JS]

Most probably one would answer the main question of this section “why do people play games?” with “because it is fun”. Fun is mostly intangible concept. In order to describe how fun is related to games let’s define different actions that humans tend to find “fun”. These actions would include: winning, problem-solving, exploring, chilling, teamwork, recognition, triumphing (someone loses while you win), collecting, surprise, imagination, sharing, role-playing, customization, goofing-off.[1]

Having these general actions in mind let’s return to concept of games and how fun is related to games. Different studies analyze what make different games enjoyable and playable for players. Further frameworks have been established to describe it.

Nicole Lazarro, a famous game designer, defines 4 Types of Fun:[NL]

1. Easy fun (blowing off steam, something casual, light, chilling out)

1 https://www.coursera.org/course/gamification Course on Gamification by K.Werbach, University of Pennsylvania

2. Hard fun (facing challenges, overcoming obstacles, achieving mastery)
3. People fun (interacting with others, working as a team, socializing)
4. Serious fun (something meaningful outside of game, a “good” or “noble” thing - helping, saving ecology. However, it may also be meaningful just inside game - collecting a full collection of badges in order to unlock special content)

Raph Koster in “A Theory of Fun for Game design” links fun to noticing the pattern in a broad sense and tracing it afterwards to re-occur. Moreover, it is not just about recognizing patterns, but it is a lot about being surprised by a recognized pattern. Fun is closely linked to mastery and teaching thyself. However, teaching in a broad sense as well: teaching about space, relationships between objects, odds etc. and teaching to act, to explore, to collaborate etc.[RK]

Marc Leblanc, educator and designer of video games, famous for his work and involvement in Ultima Underworld, series Thief and System Shock, defined 8 kinds of fun related to games[ML8]:

1. Sensation. Game as sense-pleasure
2. Fantasy. Game as make-believe
3. Narrative. Game as unfolding story
4. Challenge. Game as obstacle course
5. Fellowship. Game as social framework
6. Discovery. Game as uncharted territory
7. Expression. Game as creativity tool
8. Submission. Game as mindless pastime.

Jesse Schell, a famous game designer, in addition to LeBlanc’s framework lists several other things that humans tend to enjoy[JS]

- Anticipation. Knowing that pleasure is about to come and waiting for it is a pleasure in itselfю
- Delight in misfortune of another. Consider example of unjust person getting punished.
- Gift giving.
- Humor.
- Possibility. Having many choices and being able to chose.

These are experiences during shopping or in front of a table full of delicacy.

- Pride in accomplishment.
- Purification. People enjoy order and making something clean. Games often make use of that by tasks “eat all dots”, “clear level”.
- Surprise. Brain likes surprises.
- Thrill. This type of fun is experienced when one experiences terror, but feels secure in safety.
- Triumph over Adversity.
- Wonder.An overwhelming feeling of awe and amazement.

Mihaly Csikszentmihalyi defined a concept of “flow” to describe an optimal experience of a human being. According to this concept flow is the state of mind when person is completely concentrated on a certain task. Most of the other concerns except this task seem to lose meaning and not to matter anymore. However, flow is very fragile set of mind. Meaning that it is easily lost and person shifts to another state. Csikszentmihalyi defines it as a balance between the challenge the task represents and the skill of the performer. If the balance is disrupted in favor of task being more challenging person is shifted to Anxiety. On the contrary, if the actual skills are higher than the required by proposed challenge person shifts towards boredom. Zone of balance in between is defined as a “flow tunnel”.[MCZI]

Sweetser&Wyeth used this model of flow in order to construct a model targeted at describing enjoyability of computer games. The resulting GameFlow model consisted of 36 factors grouped in eight main elements[GFL]:

- Concentration
- Challenge
- Skills
- Control
- Clear goals
- Feedback
- Immersion
- Social.

Pervasive games

This section will explain what pervasive games are in general. It will provide different examples of such games and principles used in design of these games. It will also define main features that differentiates pervasive games from other games.

In order to begin a look into theory of games has to be made.

In 1938 Johan Huizinga, a Dutch historian, cultural theorist and professor, wrote a book “Homo Ludens” (“Man the Player”). This book was investigating the play element of culture and society. Among other concepts he defined a concept of “magic circle” there. According to Huizinga “magic circle” is predefined materially or ideally, deliberately or as a matter of course playground. It is an isolated, forbidden, hallowed spot in space, within which special rules obtain power. He compares playground with ritual place, as conceptually it is hard to distinguish between them. In practice “magic circle” is a temporary world that appears within ordinary world and is dedicated to certain act that is performed within. Simple game related practical examples that everyone is familiar with include the arena, the stage, the card-table, the tennis-court, the football-field. This “magic circle” is what separates the game from the ordinary world.[JHP]

A lot of changes happened in game industry since this definition was made in 1938. One of the most important changes was the invention of virtual digital dimensions, which grant more and more freedom to players. We can “steal”, we can “kill”, we can “destroy” in computer games. Moreover, this is considered not only as acceptable, but as “fun”. This difference between perception of performing a same action inside game-world and real-world is what “magic circle” grants us.

However, the blurring of “magic circle” boundaries is a general trend that is happening inside games-industry as this thesis is being written. Jesse Shell states that “people have hunger for reality”[JS]. Game designers are taking this into account. For example, such systems as Wii MotionPlus, PlayStation Move, Kinect and others make it possible for player to control their avatar in digital world with gestures in real world.[2] Moreover, some games are even designed in a way to blurr these boundaries. They often merge real and game-world by, for example, turning ordinary locations into a

2 http://en.wikipedia.org/wiki/Wii etc.

playground. Such games are defined as “pervasive games” in general. As Montola states: “A pervasive game is a game that has one or more salient features that expand the contractual magic circle of play spatially, temporally, or socially.”[MPG] In this sense spatial extension may mean removing boundaries that limit playground and turning whole world into a playground. Temporal extension would mean blurring of concepts of “game start”, “game finish” and “game process”. Pervasive games are becoming something that takes its time constantly, while players take part in it. Social expanding would mean playing with other players you do not know, while convential games often require certain level of aquaintance or at least real-world contact between players.

According to M.Montola pervasive games are games that exist in the intersection of different phenomena. These include city culture, mobile technology, reality fiction, performing arts, network communication etc. [MPG] Pervasive games may exist in different intersections and, therefore, are often of a very different nature. They may be single or multiplayer, they might aim at short-time playing while waiting on a bus stop, or may be a long-lasting game where Player has to build up his progress for days or months. Some games may be limited in time by game-designers and will have a set starting and finishing point in time. In such games, for example, different teams would start competing in order to reach some kind of destination (actual or metaphorical) in time. Other games would be possible to play any time. Some games are fixed to certain locations, while others may be played from any place of the world. Some games require control and guidance from game-designer or “narrator”, while others are created by players themselves.

Different game genres may be identified within pervasive games. Yet, while some games easily fall for one or more of these genres, some games are “indie” (from “independent”) and are not classified that easily. Stenros and Montola make an attempt to classify them and identify such genres as treasure-hunt, assassination games, pervasive LARP, alternate reality games, smart street sports, playful public performances, urban adventure games and reality games.

In order to familiarize the reader with pervasive games and provide richness of possible design-solution varieties let's take a look at implemented games within different genres. The variety is immense and, unfortunately, there is not enough space to describe all of possibilities, as the only limit here would probably be imagination. Yet, imagination has no limits.

Treasure-hunts are games in which player has to find a certain object at a certain location. This initially was a real object (a code message, a book, a bag, etc.) hidden somewhere in the area (written under the bench, placed behind the poster, hidden in the ground, sewers, caves, abandoned structure or any other building etc.), but later also become a pure virtual object (accessible through smart-phone or mobile PC device with available location aware mechanism). Players may compete in finding object, finding it faster or simply cracking the code. In this sense these games are considered as context-aware games, because they are linked to locations. Games that use this principle include: GeoCaching[3] (searching for an object by GPS), CityQuest (solving logical puzzles around the city to get to next location and find a new puzzle before other teams by foot), AutoQuest (same as CityQuest but using cars and is usually happening in the night), and many many others. They are well known all around the world and are either time based and have fixed game sessions (CityQuest) or are happening constantly (GeoCaching).

Assassination games where players have to eliminate other players and avoid being eliminated themselves. Examples would include game “Killer” described by Montola[MPG], where players have to stalk their victims and “kill” them with water-pistol. In 1999 a similar game took place in Berlin for two weeks. In 2005 and 2006 similar game was organized in Kiev. Different technological or design solutions are possible here, yet, ideas of stalking and eliminating a victim in real life would be central that differentiate these games from others. “DTU Hunter” a game of such genre adjusted for DTU Campus area was developed as a concept by author of this thesis and his team during “Mobile Application Prototyping ” course at DTU in Spring 2012.[4]

Pervasive LARP is, basically, a live-action role playing that is performed in real world. Players have to pretend to be characters and perform some special actions. This is a theater-like approach where players become actors. Players may pretend to be elves, vampires, etc. and pretend that they are surrounded by castles and fortresses etc. This is an old genre, and is very popular due to historical reconstructions and fantasy roleplaying

3 http://www.geocaching.com/ The Official Global GPS Cache Hunt Site
4 http://www.kurser.dtu.dk/02827.aspx?menulanguage=da

games. People dress up, use specific language and follow the script or improvise to reconstruct certain historic event, scene or plot from a fantasy book or movie or follow a self developed script.

Playful public performances are another new way of entertainment where participants are performing not only for their own fun, but for fun of random audience. These may include flashmobs (where people suddenly gather at a certain place in the city, perform a scenario and then disappear) or performances. Example of such game would be a game author of this thesis participated in Kiev, where a spectacle of a fake wedding and a bride running away from groom was performed before random citizens that may or may not have imagined it being a game.

Alternate Reality Games (ARG) are about merging imaginative world with real world. Best examples would be game called the Beast. This game was an advertisement campaign for Steven Spielberg's movie "A.I." (2001) Cast after trailer for the movie included a person that was a "sentient machine therapist in production team". A special blog page and personal web-page for this person was created in order to support curious googlers. (curious people that started investigation on why would there be a sentient machine therapist in movie production team). Search after search, text after text people were involved in solving puzzles, trying to resolve a murder case. This game was not advertised and created an environment of secrecy around it. The project had no name or official web-page. Resulting name "The beast" was just an insiders-joke that turned to nickname. Players had to explore web-pages, send emails, call phone numbers etc. Secrecy coined a gameplay where it was required to recognize the task or puzzle before solving it. Moreover, Microsoft, responsible for game design has not confessed or revealed any details about the game while it was running or even the fact of game existence. Example of a task would be: a player had to find an actual poster advertising the movie, count small notches in letters, this would result in a phone number, calling this number would result in a proper voice message being played to player, using marked letters from posters player would also get messages like "Evan Chan was murdered" or "Jeanine was the key". This game had serious drawbacks and was not ideal. According to later analysis it was really hard to join the game for new players and keep in pace with other devoted players, that were spending more than 40 hours a week on puzzle search and code cracking. [MPG] How to measure a success of a game that is pretending to not to be a game? A social network group called Cloudmaker's was created to collaborate on game solving. This group had thousands of members at its peak and generated over forty thousand messages.[5]

Smart street sports is a combination of a sport game that is supported by technological means (cellular phones, GPS devices, etc). Players often need to combine real physical activities with some kind of tactical approach. These games may be regarded as evolved, advanced versions of old familiar games. Examples would include chasing other players. In the Can You See Me Now? players at computer have to chase with their avatars other players, that are actually traveling around the map with GPS devices[LBG]. However, it is important to state here that often smart street games are results of gamification of sport activities. For example, “Zombies, Run!”[6] game for iOS and Android encourage people to work out and run by comparing their running statistics with “what would be if there was a zombie chasing you” scenario.

Urban Adventure Games are games that include puzzles in city-scapes. These games are usually about providing a player with a playful way of learning something new about a city. They are often used by municipality to introduce city to tourists or increase awareness of both tourists and citizens. Examples would include games launched in Open Air Museum in Brede, Denmark or “Visions of Sara”, a game developed during PhD research on Location Based Games in Aarhus. [LBG] In these games players were required to investigate areas in search for clues that would lead them through the story.

Reality games encourage players to see ordinary world in a new light and change it. This may include actions, not widely accepted per se. For example, changing the street lights color, making a sweater for a statues, changing look of a building. Participants do not imagine or recreate another world within reality or parallel to it. They operate in reality.

It is important to state that the provided classification is but an attempt to classify something that constantly develops and evolves. Provided classification genres tend to describe differences between possible pervasive games. However, new genres may and should appear. Moreover, some genres presented are “broader” than others. Concept of ARG, for

5 http://en.wikipedia.org/wiki/The_Beast_%28game%29
6 https://play.google.com/store/apps/details?id=com.sixtostart.zombiesrun

example, is broader than smart street games. This already results in a possible dividing of ARG in smaller sub-genre groups if required and will apparently result in such division later, as number of pervasive games will grow. Different games due to richness of imagination and used techniques may result as not being within limits of one genre, but be somewhere on intersection of genres. Some of these games are more context-aware than others. In a sense that environment may play a bigger or smaller role within game process itself.

Chapter 2

Design stage

This section will provide description of a design stage of this project. It describes the design of idea that later was implemented and resulted in development and launching of an actual context aware game for mobile platforms.

Initial idea of creating a game-environment was explained in detail as creating a game-environment that students, workers and visitors of DTU would enjoy and find “fun”. This game environment should be interesting for both genders, be self driven and exist in parallel with study process without disturbing it, but filling in the time gaps in it. This designed game-environment should not require constant attention or control from any kind of “narrator” and should not be limited in time. However, it should be densely related to campus area. Game has to have low “entry” barrier for new players.

In order to address problem properly an investigation of potential game-area and players needs to be performed.

Game-area

Campus area of Technical University of Denmark (DTU) was chosen as a tested playground for the designed game-environment for campus area. DTU campus area is located in the northeastern end of the city of Lyngby, Denmark. It occupies more than 375 000 square meters. It is divided into four quadrants of approximately equal sizes that are numbered 1 to 4. These numbers are also used for numbering of more than 50 buildings around campus area. Most of the territory is covered with DTU or Eduroam Wi-Fi network. Field-tests were performed to prove Wi-Fi coverage of these networks in all DTU buildings that are used in study process. Apart from these building campus area also includes dormitories (Kampsax Kollegiet, Villum Kann Rasmussen Kollegiet) and dormitory-alike locations (Campus Village etx), at least six student-bars (bars in 101, 116, 208, 342, Kampsex bar, VKR bar), grocery shop, Oticon Salen used for public activities, research and production locations (Danchip clean-room, DTU Scion, etc) and some other locations. Main building of DTU hosts administration, main library, main canteen, a bar, a sport hall and a gym. :[7]

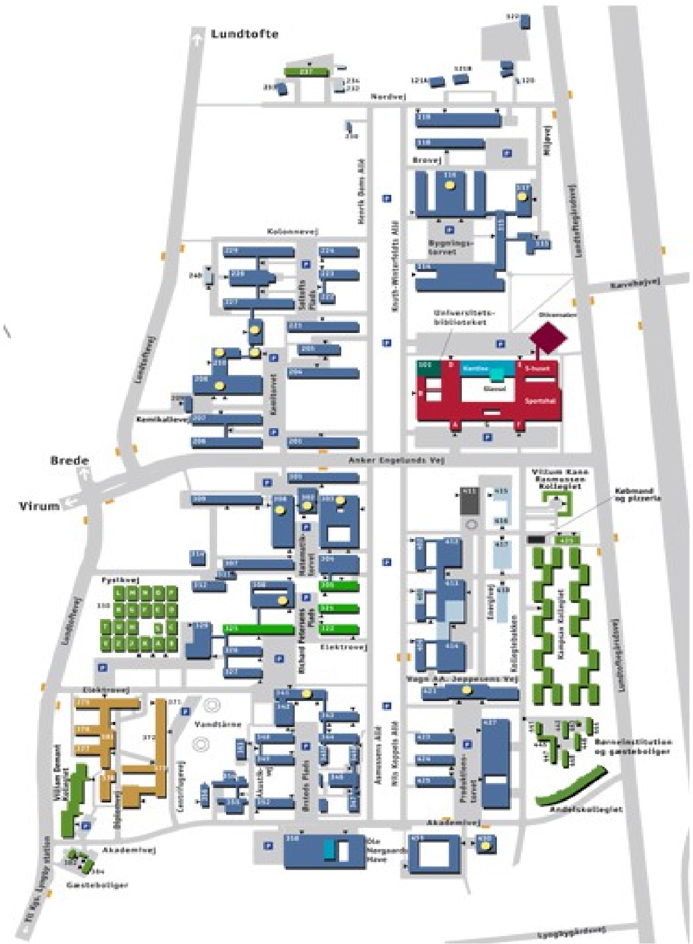


Illustration 1: Map of Technical University of Denmark [7]

One of the objectives of thesis and tasks of the designed game-environment was defined as to turn this campus area into game-area. 4 geopoint vertices were chosen to form a rectangular gamefield. Game grid size was chosen as 14x22, x[0;13] y[0;21] to support cell areas of equal size, in correspondence to accuracy of location sensor. (will be discussed later).

7 http://www.dtu.dk/

Potential player analysis

Potential player would be a student of DTU. However, people that are employed at or are visiting DTU should also be included. “DTU welcomes around 750 international students per year. More than 400 new PhD student enrollments per year. Teacher/researcher to student ratio – 1:4“.[8] “Around one in every eight students is from abroad, which is the highest proportion in Denmark and far above the national average for universities” Other statistical data shows that “The University has approximately 1,500 researchers, 6,500 students, 850 PhD students and 650 international students a year.”[9]

Provided statistics together with personal experience of author of this thesis gained from being an international master student at DTU during two year Msc in Telecommunication program makes it possible to formulate next statements about generalized potential player.

Table 1: Potential player

Feature	**Information**
Gender	Majority male (Optimistic ratio 2 to 1)
Age	18-30 (majority)
Language	All speak perfect or almost perfect English Major language group include Danish. Other major groups vary from year to year.
IT awareness	Very high
Social interaction	Students recognize each others as peers Students like to be distracted from their routines (according to personal experience 8 out of 10 working laptops seen would have Facebook page opened during classes) Students want to be in control of own time and avoid distraction if necessary Partying activity is present in the game-area during 2-3 days per week (bars are opened Thursday Friday, occasionally Saturday)

8 http://www.dtu.dk/

9 http://www.topuniversities.com/universities/technical-university-denmark

General design remarks

Different games may aim for different kinds of fun and different combinations of them. This diversity of possible targets is what results in diversity of games. Finally, beauty is in the eye of beholder. So is fun.

In order to design a game-environment author of this thesis decided to follow advices and instructions of a famous game-designer Jesse Schell [JS].

Jesse Schell, defines fun as pleasure with surprises. Moreover, he emphasizes that different genders tend on average to find different things fun. Men on average enjoy mastering things, competing, destroying things, spatial puzzles, trial and error. Women, on the contrary, on average enjoy exploring richness of emotions, connection with real world, nurturing, verbal actions and puzzles, learning by example.

Therefore, taking into account this statement and player analysis the designed environment should reflect as much of these two groups as possible to address both genders. It was also chosen to implement ideas of Easy and People fun defined by Lazarro[NL] and explained in previous chapter in order to fulfill goals of this project.

Moreover, as it was intended, a game should include player-to-player interactions. Different players tend to express different behaviors in multiplayer games. Richard Bartle, famous MUD (Multi-User Dungeon) game designer, proposed classification of MUD players, which is still applicable to analyze different player behaviours present in most multiplayer games that exist today. He proposed 4 categories of players and assigned each one with a card suit. [RBMP]

1. Achievers (Diamonds) are those players that focus and enjoy overcoming challenges and going for achievements.
2. Explorers (Spades) enjoy being surprised by the game and spend time uncovering its secrets and/or mechanics.
3. Socializers (Hearts) enjoy social activities and try to engage in those as often as possible.
4. Killers (Clubs) enjoy imposing themselves often by defeating other players and disrupting other players' experiences.

However, it is also important to take into account that designed game environment in this project is context-aware location based, which is a subgroup of what is often defined as “pervasive games”. This specifics changes certain aspects of the game mechanics, player’s behavior inside the game etc. This issue was addressed by K.Jegers in his Pervasive Gameflow model.[KJPG] This model was designed as a special version of Gameflow model, but adjusted for pervasive games. In addition to criteria defined in Gameflow model Jegers proposes additional criteria that he places within 8 original groups. Some important changes and additions include changes in Concentration (games should support switching of concentration between game and other matters of importance that surround the player), Challenge (gamers are stimulated to create own gaming environments and move through them with suitable pace; games should keep balance between provided creativity freedoms and imposed constraints), Skills (more flexibility in pace of skill development), Control (game should be able to be picked up and paused easily, quickly explain current status of game and how it changed during player’s absence, gaming in many possible physical settings should be possible), Clear goals (game should help players to form and communicate their own goals), Immersion (games should understand the everyday context and not require player to somehow violate acceptable social norms), Social interaction (gaming environment should create possibilities for creating meaningful interaction between players and create triggers or events that would promote this interaction).[KJPG]

Summing up conclusions made by Jegers, one of the most important difference of pervasive games that sets them aside is that player involved in such game may switch focus between game and out-of-game environment. Moreover, player’s attention focus may switch between two environments quite often. This has to be taken into consideration during game design stage in order to make this refocusing of attention neither disrupting the game experience, nor causing frustration in out-of-game environment. Decreasing the influence of interruptions, taking into account specifics of context and setting where game is played.

Idea development

During the multiple brainstorm sessions a great abundance of ideas that would coin the resulting project were formed. Most of these ideas were disregarded due to different reasons. Full set of ideas would be to big to list it here, but some examples are presented.

Table 2: Disregarded ideas with reasons

Idea	Reason to be disregarded or delayed
Planting bombs and searching for them	Safety concerns and ambiguity issue inside campus area
Killing or Hunting other players	Safety concerns and ambiguity issue inside campus area
Drawing or other creative capabilities included	Technical difficulties, Similar projects, Gameplay uncertainty
Teamplay	Would make game entering barrier for new players higher
Facebook integration	Dependency from 3rd party and limitation for players

The idea that was coined during design: Game-area is divided into set of areas with equal area (grid with cells). Players need to investigate these cells. They need to harvest different resources from different areas to build-up own virtual riches. Different resources of different value are available. Resources may be harvested or crafted from other resources. These resources are used for hiring and training military units that are used to conquer cells. Players compete between themselves for dominance over the map.

Lenses of game-design

Game-design as defined by Schell[JS] consists of looking at a Project through a set of “lenses”. Each of these “lenses” is a question or a set of questions a designer has to answer before moving forward, or returning through iterations of design, as game-design most often is a very iterative process. Below presented most important lenses that were used during the design of the game-environment. All questions are presented as defined by Schell himself and answers are the decisions made by author of this thesis. Some of the lenses were not used, as they were not applicable to current project due to different reasons. These lenses that were disregarded are applicable for commercial projects and projects that involve many parties. As this thesis is an individual research project without any budget and with a defined deadline only the most relevant lenses were used to analyze and formalize the proposal.

Lens #1: The Lens of Essential Experience

"What experience do I want Player to have? What is essential in this experience?"

Designed game environment aims to introduce a new way of looking at campus area. It attempts to merge ordinary campus area with a game world. A task would be to make day spent at campus more interesting and exciting.

Lens#2: The Lens of Surprise

"What will surprise players when they play my game? Does the story/artwork/technology have surprises?"

The idea of playing while studying is rather innovative. In a sense of creating a game-world that will coexist with university and that may be used for promotional, educational activities and/or entertainment. Moreover, randomness that lays within such game events as harvesting resources, actions of other player should have a certain degree of surprise.

Lens#3: The Lens of Fun

"What parts of my game are fun?"

Designed game environment is targeting concepts of easy fun (resource harvesting, unit training) and people fun (player-to-player interaction through conquering and battling over game areas and competition in map dominance) that were described in theoretical research.

Lens#4: The Lens of Curiosity

"What question does my game put in player's mind? What can I do to put even more questions?"

Designed game is all about investigating familiar campus area from a new perspective. Searching for special locations would make player to wonder where are these locations scattered and who is controlling them.

Lens#5: The Lens of Endogenous value

"What is valuable to the players in my game? How can I make it more valuable to them? What is the relationship between values in the game and the player's motivations?"

Resources are divided into tiers. Some are more rare and, therefore, valuable. High tier resources are valuable, because they are used for hiring and training military units that are tools to ultimate goal, which is dominating the game map.

Lens#6: The Lens of Problem Solving

"What problems does my game ask the player to solve? Are there hidden problems?"

Spatial problems of locations investigation and all aspects of competition with other players may be defined as main problems player has to solve.

Lens#7: The Lens of the Elemental Tetrad

"Is my game using all elements of this model?"

An initial design-matrix that is using an "elemental tetrad" model[JS] was formed for this project. It includes four elements: Technology(materials and interactions required for game-play), Mechanics(procedures and rules of the game), Story(sequence of events or general imaginative framework), Aesthetics(the way game looks, sounds, smells, tastes, being experienced in general)

Table 3: Elemental Tetrad for designed game-environment

Technology:	**Mechanics:**
Smart-phone used as Input/Output device Server side required to support interactions **Internet connectivity required for data exchange**	Players play individually Location needs to be identified Relationship between real location and game-data needs to be established (Players can harvest resources from areas) Players are able to interact with each other **(Conquering and battling for areas)**
Story:	**Aesthetics:**
Medieval (Fantasy) atmosphere Story-line is not present in order to make sure that gamers play game constantly It is replaced by an ongoing competition	Map may be used to represent location data to Player Simple and not overweight interface **Navigation is intuitive**

Lens#8: The Lens of the Holographic Design and Lens #45: The Lens of Imagination are connected for this project:

"What elements of the game make experience enjoyable? What can be changed to improve it?" and *"What is the imaginative part and what is the connection to reality?"*

Possibility to see campus area as a game-field and competing with other peer players over real locations might be enjoyable according to described concepts and frameworks.

Lens#9: The Lens of Unification asks *"What is my theme?"*

A medieval fantasy world that exists parallel to campus area was chosen as a theme.

Lens#10: The Lens of Resonance

"What about the game feels powerful and exciting? What get my friends excited if I tell them about my game? What would this game be if there would be no constraints?"

When author of this thesis described concepts of the game and discussed it to third parties and friends - ideas of combination of game and real world was met with most noticeable excitement. People from different universities in Denmark (Copenhagen University, Copenhagen Business School) and from outside Denmark asked if the game is to be expanded to other territories except DTU and were asking for particular details about units, battle implementation, possibilities inside the game etc.

Lens #11: The Lens of Inspiration

"What experience that I have had I would like to share with others?"

Experience of investigation unknown areas. Experience of seeing something familiar from a new angle. Experience of being part of a group.

Lens #12: The Lens of Problem Statement

"What is an actual problem I am trying to solve? How will I be able to tell that problem is solved?"

The problem was stated in Introduction.

Lens #13: The Lens of Eight Filters

Eight very important questions should be answered by game designer before moving forward. These questions and answers may seem subjective, but they show if designer understands and clarify concept.

Table 4: The Lens of Eight Filters

Does the game feel right?	The concept feels right
Will the intended audience like this game enough?	Yes, if designed properly

Is this a well-designed game	It might require adjustment of game balance
Is this game novel enough?	Yes
Will this game sell?	The game and all game aspects are free within this first iteration
Is it technically possible to build this game?	Yes, but proof of concept is required
Does this game meet our social and community goals?	Yes, it does
Do the playtesters enjoy the game?	Answer to this question is impossible to give during design stage

Lens #14: The Lens of Risk Mitigation

"What could keep this game from being great? How to stop that from happening?"

Limitations of technology may prove the initial game design as not working. A proof-of-concept is required. Design of UI may end up poor due to project budget (no budget at all, single person being involved in all game-design and game-creating activities) and lack of proper skills of drawing. Unfortunately, these dangers can not be dealt with currently. An answer would be: to find a team of talented people with different backgrounds to fulfill all aesthetic and technological requirements.

Lens#16: The Lens of the Player

"Who is my Player? What do they like?"

The answer to this question is present in Player analysis.

Lens#17: The Lens of Pleasure

"What pleasures does the game give to players? What pleasures are missing?"

Pleasure of surprise. Pleasure of conquering the area and competing with others, that are regarded as peers. Probably, aesthetics pleasures are going to be limited due to no budget and no visual art skills.

Lens#19: The Lens of Needs

"On which levels of Maslow's hierarchy is my game operating? "How can I make my game fulfill more basic needs than it already is? "

"A theory of human motivation" work by Abraham Maslow[MPN] proposes a pyramid model that describes human motivation through hierarchy of human needs. The main concept is that one can not have motivations of higher levels without fulfillment needs of lower, more basic levels.

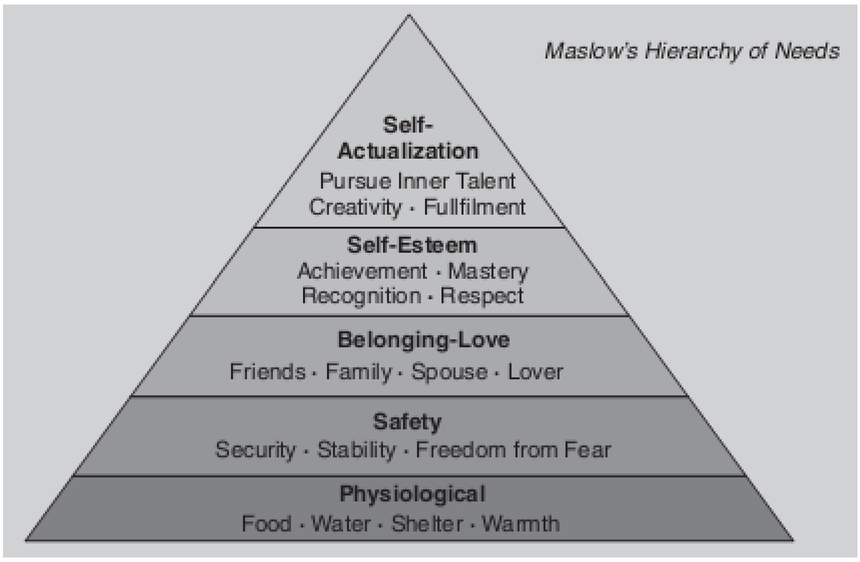


Illustration 2: Maslow's Hierarchy of Needs

The designed game-environment operates at the levels 3, 4 and 5 from bottom. The game takes place in environment that is constituted from peer players, some of which will be friends or even family members. Through making use of this defined social group that is organically created inside campus environment we aim for needs from level of belonging. Conquering areas and rising in chart of top players, on the other hand, addresses needs from self-esteem and self-actualization levels.

Lens#21: The Lens of Functional Space

"Is the space of this game discrete or continuous? What are the boundaries of the space? How many dimensions are involved?"

Every game takes place in some type of space, which is related to "magic circle” concept already described. The visual and aesthetic layer of a game have to be stripped in order to understand the mechanics of space. Some games take place in discrete spaces. For example, Tic-Tac-Toe, where all player movements are defined by a grid. It does not matter in which part of the cell within grid an “X” would be placed while the cell is uniquelly identified. Other games exist in continuous space. For example, football where players and actions may occupy any meaningful place within game area. Changing space mechanics often change the game. Consider how would change of Tic-Tac-Toe size from 3x3 to 9x9 change all player strategies. The designed game-environment is a location based game that uses campus area of DTU as a game-field and, therefore, space. Space is limited by campus area territory. This area is divided into cells with a grid, which makes it a discrete game space.

Lens #22: The Lens of Dynamic state

“What are the objects in game and their attributes? What are possible states? What is known by all Players? What is known by some Players? What is known by game?”

Characters, props, tokens etc. Everything may be considered as an Object within the game. These objects are the “nouns” of what stands behind game-mechanics. These objects are described by attributes. These attributes may be of static nature and never change (color of a checker at a certain location, for example, is static) or dynamic (number of houses at location in Monopoly).

For the designed game-environment next Objects were identified:

Table 5: Game objects with attributes

Objects	**Attributes**
Resource	May be harvested (static) -shows if resource may be harvested from game-field by Player May be crafted (static) -shows if resource may be crafted by Player Tier of resource (static) -represents value of Resource
Player	Player's id (static) -uniquely identifies Player in the game
Creature	Is owned by a Player (static) Is placed at Location (dynamic)

	Strength (static) Health (static) Static in a sense that they might not be changed after Creature is placed at location
Location	Type (static) Coordinates (static) Type and coordinates describe the location on the game-field Is controlled by Player (dynamic) Creature may be placed at location (dynamic)

Different scopes of knowledge are usually defined in the game. Some information is freely available to all Players. Some information comes at a price. Some information should always remain secret in order to preserve the game play. Some information is of purely agnostic nature. (this includes all random events).

Next knowledge-scopes were defined within designed game environment.

Table 6: Knowledge-Scopes for Game-Environment

Public information	**Shared information**	**Private Player information**	**Private Game-designer information**	**Not-known information**
-Game map -Top players	(private player information may be shared by collaborating parties)	-List of areas controlled -List of investigated locations with their types -Amount of resources in stock -Current status of current location	-Mechanisms of Harvesting -Mechanisms of Creature battle -All location types	-Randoms inside resource harvesting -Randoms inside equal creature comparison

Lens #23: The Lens of Emergence and Lens#24: The Lens of Actions are connected

"How many "verbs" do my players have? How many objects players have? How many

subjects are controlled by player? What are the side effects that change constraints? " and *"What are the operational actions? What are resultant actions?"* respectfully.

Actions within game are what creates the game process. They are the "verbs" of the game-play. Actions may be "operative" and "resultant". Operative actions describe what player can take. Example would be player may simply move figures in chess or attack other figures. Resultant actions are seen at a larger scale. They describe how player is using operative actions in order to achieve some goal. Example would be sacrificing a figure in chess to trick opponent into a trap. Number of possible actions defines the experience, so does the ration of "operative"/"resultant"

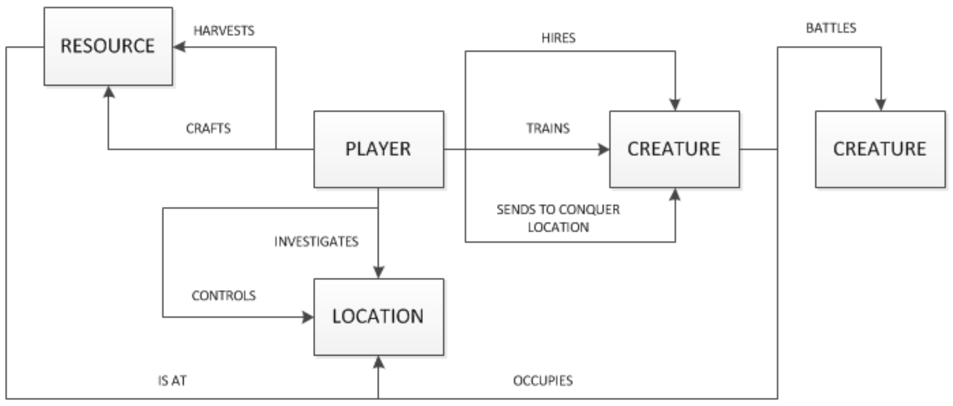


Illustration 3: Operative actions

Resultant actions would include: movement around the map, noticing other players, identifying more and less dangerous or important areas for conquering.

Lens #25: The Lens of Goals

"What is the ultimate goal of my game? Is that goal clear to players? Are there many related goals? Are my goals concrete, achievable and rewarding?"

The ultimate goal of the designed game is conquering and controlling an area of university. This is rewarding through status and authority presented in Maslow's pyramid before. Players would get self-esteem and recognition. This lens works in relationship with Lens #40: The Lens of Reward that poses such questions as *"What rewards is my game giving out now?"* Top chart of players was designed during implementation stage to address it. This Lens is in dense relationship with Lens #66: The Lens of the

Obstacle as goal with no obstacles has no meaning. Player competition is defined as main obstacle in the game.

Lens #26: The Lens of Rules

“What are the rules? Are there “laws” or “house-rules” that are forming in parallel to defined rules? Are there different modes in the game? Are the rules easy to understand?”

After full list of rules was developed (Appendix) an investigation of possible “house-rules” was performed. During design stage these “house-rules' were not identified and it was chosen to check this aspect during game testing. Designed game-environment works as an enforcer of implemented rules. This is the common approach for computer games. Computer is often chosen as main rule enforcer in order to limit players freedom as most of actions have to be simulated and unusual actions may result in strange outcomes. Two main modes were defined: in-gamefield and out-of-gamefield. It was chosen to make certain actions available in both modes (harvesting, crafting, hiring, training), while some were available only in one (conquering an area and, therefore, competing).

Lens #27: The Lens of Skill

“What skills does my game require from player? Which skills are dominant? Can players improve skills by practice?”

Usual games are, as shown before, are based on different skills and developing of these skills by players. The designed game-environment was characterized as “pervasive” and, therefore, this connection with skills here is not that obvious. Designed game does not build upon conventional skills. Most probably, required skill may be defined as “motivation to move around and not spend all time at a single place”. This may be considered as a skill players want to develop. An assumption is made that designed game-environment will be a motivating factor to pursue development of this skill in players eager to devote themselves to this goal. Another possible skill would be “basic strategic thinking” that would be used to define most important game locations, safest locations, locations most commonly attacked by other players.

Lens #29: The Lens of Chance

“What in my game is truly random? What parts just feel random? Does randomness give

players positive or negative feelings? Are there interesting risks?"

Two different concepts were designed as random within the game. Game harvesting and battles between equal creatures. Both of them will be discussed in details in section that describes actual implementation of design.

Lens #30: The Lens of Fairness

"Which is more important: that my game is a reliable measure of who has the most skill, or that it provide an interesting challenge to all players?

Full answer to the question of this lens was developed during implementation stage. This lens was the main reason to introduce such aspects as stock and creature limitations, creature aging at location in order to lower a barrier of entrance for new players with low skill.

Lens #35: The Lens of Head and Hands:

"Are my players looking for mindless action, or an intellectual challenge? Can I give the player a choice — either succeed by exercising a high level of dexterity, or by finding a clever strategy that works with a minimum of physical skill? "

Designed game environment is pervasive and location based. This means that players are required to move more if they are eager to achieve more. Movement is the main physical activity. Additional activities such as harvesting, crafting, hiring, training implement easy-fun concept and may be considered as mindless action. Basic strategic thinking that should be used represents a minimal, yet still present intellectual action within the game.

Lenses #36: The Lens of Competition, Lens #37: The Lens of Cooperation and Lens #38: The Lens of Competition vs. Cooperation describe the players interaction within the game and what is more important competition or collaboration.

The game environment and concept was designed as a competition. However, it might be possible that some players will team up to achieve common goals, such as being in Top players together. Low entry barrier have already been addressed by previous lenses and is discussed in detail in implementation section. Moreover, predefined game location and existence of university community that combines all people spending time at university addresses Lens #86: The Lens of Community and Lens #84:

The Lens of Friendship automatically.

Lens #39: The Lens of Time

"What is the timeflow inside the game? When does it start and when does it end? Is there a time-limit for game actions?"

Most of the game actions require player to be at campus area and are not limited in time in anyway. This is a pervasive game that takes place in parallel with ordinary activities at campus area. Temporak patterns of user behavior were established and are discussed in chapter that assess results.

Lens #32: The Lens of Meaningful Choices: *"What are the choices my players have to make?"* and Lens #33: The Lens of Triangularity: *"Is there a choice of playing safe for low or risk for higher reward?",* Lens #46: The Lens of Economy, Lens #47: The Lens of Balance, Lens #49: The Lens of Visible Progress, that address in-game economical actions, balance issues, visual representation of progress together with Lens #54: The Lens of Physical Interface, Lens #55: The Lens of Virtual Interface, Lens #57: The Lens of Feedback were are addressed in implementation section.

Lens #44: The Lens of Character that asks "Is there anything strange in my game that players talk about excitedly? " , Lens #61: The Lens of the Interest Curve, Lens #62: The Lens of Inherent Interest that are addressing players actual perception of the game are described in later section that describes game-environment testing.

Design stage conclusions

Taking into consideration "pervasive" differences and proposed methods of design assessment and frameworks a proposed game-concept is analysed and next possible fun or flow engagement aspects are established.

Game concept may be identified as Easy fun (game takes place at campus environment and designed to distract without distracting too much), People fun (competing through conquering locations). Collecting different resources in order to spend them may, however, also be identified as possibly related to Serious fun as well. Collection of resources and random chance involved may trigger Collection, Surprize, Chance aspects of fun. Necessity to collect different, often scarce resources may trigger certain aspects of Challenge and Exploration.

Series of meaningful choices provided, clear feedback, unusual short activities at usual places (for example, shaking phone to harvest a resource) and other described techniques are expected to make game fun. Fun game helps to move through the boring day actively and engagingly.

The Game-design Lenses framework was used for assessing design and systematize all proposed aspects and features of game-environment.

Chapter 3

Implementation stage

This section will discuss implementation of design system. It consists of two main parts Front-end solution and Back-end solution that describe all implementation details, reasons standing behind decisions and decision-changes for client and server side respectfully.

Front-end

A Client is interacting with game-world via a Front-end (Mobile Platform as stated in the subject) part of the designed system. This is the system that accepts all Player's inputs and presents Player with resulted output information. This section will discuss possibilities and limitations introduced by nature of designed game-environment. It will present all choices made during design and implementation of front-end, describe possible alternatives and resulting outcomes of choices made.

The mobility of Players is a natural and, moreover, a key-feature within designed game environment. At the time when this thesis is written smart-phones, tablets etc. are conquering the market of mobile portable communication devices.

Table 7: Top Smartphone Operating Systems, Shipments, and Market Share, Q2 2012 (Units in Millions) [10]

Operating System	Q2 2012 Shipments	Q2 2012 Market Share	Q2 2011 Shipments	Q2 2011 Market Share	Year-over-year Change
Android	104.8	68.1%	50.8	46.9%	106.5%
iOS	26.0	16.9%	20.4	18.8%	27.5%
BlackBerry OS	7.4	4.8%	12.5	11.5%	-40.9%

10 http://www.idc.com/getdoc.jsp?containerId=prUS23638712#.USEilaW9Qpc

Symbian	6.8	4.4%	18.3	16.9%	-62.9%
Windows Phone 7/ Windows Mobile	5.4	3.5%	2.5	2.3%	115.3%

These numbers together with personal preferences and tools available resulted in choice of Android as a mobile platform for developing a front end solution. A good alternative would be iOS, which is a strong number two within provided charts. Yet, lack of expertise and ,most importantly, lack of equipment made this choice unavailable.

About Android

Android is currently world's most popular mobile platform. There are more than 600 000 apps and games already available on Google Play to keep users entertained. Android devices are considered smart and are promised to get smarter, enriched with new features.[11] In more than 190 countries around the world Android is already available to customers. According to official page “every day another million users power up their Android devices for the first time and start looking for apps, games, and other digital content”.[12]

The choice of Android and benefits this choice provides are well explained. What are the potential drawbacks of such choice?

Fact that Android is supported by an extra-wide variety of different models of smart-phones and tablets results in great variety of technological configurations of these devices. Variety of hardware components, lack of some elements inside some models, different screen sizes and resolutions, all these differences have to be addressed by developers. One may not expect that all devices will support some features by default. This influences development of solutions that will be constructed in way that will make sure that different configurations support the design. Unavailable features then may be replaced by alternative options or removed gracefully. Wider the target segment of devices is, more different options and possibilities should be taken into account and tested.

11 http://www.android.com/about/
12 http://developer.android.com/about/dashboards/index.html

Possibilities and limitations

Modern Android devices are very different in provided configurations. However, it is possible to make an attempt to generalize what would typical device contain. This typical generalized device would have a touch-screen as a main user interface. This device would be able to access different mobile networks (cellular mobile networks such as GSM and UMTS, Wi-Fi). This device would have a set of sensors that may include accelerometer, magnetic sensor, GPS sensor, etc. Such device runs different applications used for service, entertainment and other various purposes. However, apart from possibilities mobile nature of devices introduces certain limitations. These devices have limited battery life, computational power and limited screen size and resolution. These limitations have to be addressed during design and implementation of systems meant to run on such devices.

Target system

The first choice that had to be made was a choice of a Platform version.

Different versions imply different set of built-in features. However, a certain solution and a roundabout was developed by Google. In order to make certain new features available in older versions of the platform a special support-library was created. Developers may include such library in their projects to make sure that older platform will support required features.[13]

Another choice was related to the target screen size and resolution. It was decided to support majority of devices registered as active based on information provided by Google. Android API 8 Froyo was chosen as a minimal required Android platform. However, a described support-library was decided to be included in the project as well. The typical screen sizes for smart-phones and charts were used as a guideline.[14]

13 http://developer.android.com/tools/extras/support-library.html

14 http://developer.android.com/about/dashboards/index.html

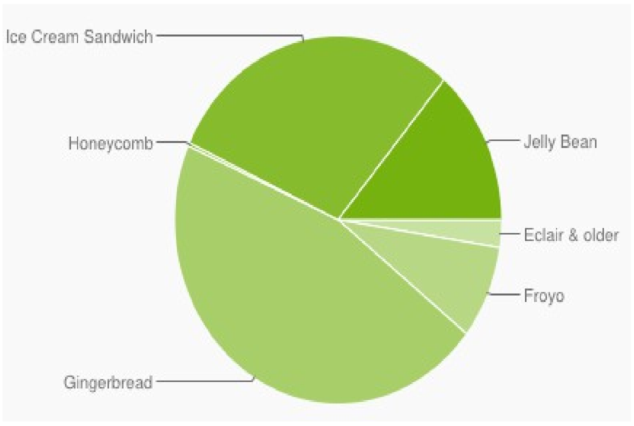


Illustration 4: Android platform distribution (February 2013)

Basics of Android mobile applications

Usually, Android applications are written in the Java programming language. The Android SDK tools freely available are used to compile the code and all resources. As a result an Android package is created. This is an archive file with an .apk suffix, which is considered a single Android application. This file is used by Android devices to install the application. All application are run within own security "sandbox" environment limiting their access and effects on each other.[15]

Typical Android application consists of application components that are basic building blocks of each application. There are four different types of application components. Each of them has a distinct purpose. They have different life-cycle that define how each component is created and destroyed.[16]

Four basic application components include Activities, Services, Content Providers and Broadcast Receivers. Activity represents a single screen hosting UI. Each screen is an independent Activity and they work together to create user experience. Therefore, Activity is a main source for user input and hosts all output data towards user. Services are running in background to perform long-running operations, they do not require UI

15 http://developer.android.com/guide/components/fundamentals.html
16 http://developer.android.com/guide/components/fundamentals.html

and are usually started by or within Activities. Content providers are used to manage sharing of data. Applications use Content providers to access data stored by other Applications, for example (as there is no direct access, due to sandbox implementation). Broadcast receivers are used to respond to system-wide announcements that have been broadcasted. Some of such broadcasts may be originated by the system and inform of system changes ("download complete", "charger connected" message etc). Yet, others may be originated by applications and inform about some app-relative change of states. Activities, Services and Broadcast receivers are activated by asynchronous messages (Intents). Intents are used to define an action and transport information between components. All components need to be registered within special Manifest File. This Manifest File is an .xml file that contains information about all registered components of an application. It also identifies user permissions that application requires, such as Internet access, Account information access etc. It declares minimum and target API version to make sure that correct versions will run the application. It is also used to declare any necessary hardware or software features, such as existence of Camera, Bluetooth on device etc and other for user awareness.[17]

Location investigation techniques

This subsection provides important information on location update strategies.

Android provides access to location services that are supported by device through android.location package. Central component is LocationManager that provides API to determine location of device (if available). As soon as access to LocationManager is received, application is able to query for last known locations, register/unregister for periodic updates, etc. [18]

Android uses a notion of LocationProvider. Each location provider represents a location finding technology that is used to determine a current location[RMAD]. These technologies would include GPS, Wi-Fi, Cell-id received from mobile network operator. Each of these technologies offers different capabilities. They differ in power consumption, accuracy and data they may or may not provide.

17 http://developer.android.com/guide/components/fundamentals.html
18 http://developer.android.com/guide/topics/location/index.html

Simplest way would be to access a LocationProvider directly. However, it is also possible to define a set of Criteria that describe desired accuracy, desired power consumption etc. These Criteria would then be used to find a matching provider.

While dealing with location it is important to take into consideration several aspects: user movement and varying accuracy. Regular updates are used to get the most relevant information time wise. However, each update may not be consistent in accuracy. This means that newest location may have much lower accuracy than the location received a little earlier from a different source.[19]

Taken into consideration best strategies defined by Android and experts in the field and conditions present within DTU campus area and features present at the potential game-field a next algorithm was designed to achieve the most relevant location information.

It is assumed that most of the time while on the game-field potential Player will spend indoors. This is easily checked by average workload expected from a DTU student (1 credit point equals approximately 28 hours, on average 80-90 is required for student at MSc level)[20] PHD students and other employees spend most of their workday indoors as well. This introduces impossibility of using GPS provider, as it does not guarantee adequate accuracy while indoors. Addressing this issue and taking into account fact that most if not all indoor premises within DTU are covered with Wi-Fi networks (dtu and eduroam networks) a decision to chose Network location provider that makes use of Wi-Fi information to obtain a location as priority one provider was made.

In addition, it is important to state that we will be checking data from all available providers in order to determine most relevant location, but rely on Network provider.

A series of field tests were performed to check the accessibility and precision of locations achieved via Android network provider. Reminder: Game grid was chosen as 14x22, x is [0;13] and y is [0;21].

19 http://developer.android.com/guide/topics/location/strategies.html
20 http://www.dtu.dk/English/education/Academic_Calendar.aspx

Table 8: Results of first field-test

Locations at DTU (Building #)	Values for game grid (Coordinates, (row,column))	Locations at DTU (Building #)	Values for game grid (Coordinates, (row,column))
425	21,6	348	19,3
424	21,6	349	18,3
358	21,3	343	18,4 and 18,5
358 (canteen)	21,4	341	16,4
450	**-2,-2**	327	16,3
358 (southern part)	**-2, -2**	325 parking entrance	15,3
352	19,2 and 20,2	322	15.5 and 16,6
canteen in 101	1,1	Library in 101 (second floor)	9,9
S-huset (cafe in 101)	1,11		

First field-test demonstrated necessity to adjust the grid over the game-map. -2,-2 values were used for out-of-game area. However, the initial idea of cell sizes was proven to work. Locations acquired from different building made it possible to get different positions within the grid.

Second field-test after grid readjustments:

Table 9: Results of second field-test

Locations at DTU (Building #)	Values for game grid (Coordinates, (row,column))	Locations at DTU (Building #)	Values for game grid (Coordinates, (row,column))
450	21,5	210	8,5
425	18,5	118	3,11
421	16,6	116	10,3

404	15,6	S-Huset (cafe in 101)	10,8
322	14,6	Library in 101	8,7 and 8,8
321	13,4	224	7,4
341	14,3	306	5,11
325	13,3		

Visualization of last field-test data with accessibility of locations is presented below. Accessible locations are marked with blue color:

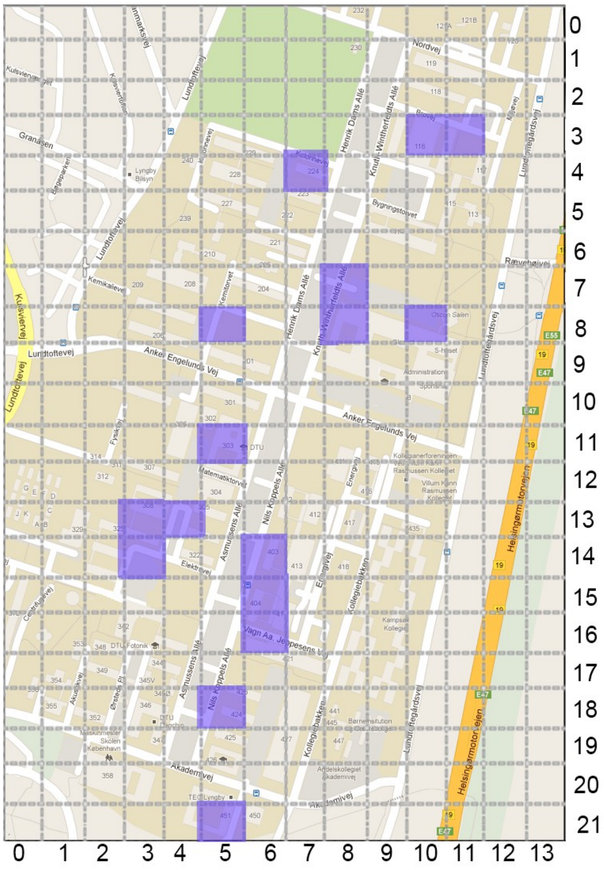


Illustration 5: Accessibility of location data (acquired in field-test experiments)

The location investigation algorithm explanation:

1. Upon starting-up application checks last available location for all providers available inside device. Algorithm checks if locations are within given accuracy and time windows. (2 minutes, 100 meters)
2. A best priority-1 location provider is determined.
3. If there is a location already available it is used as a current location. Otherwise a forced single-update of priority-1 location provider is issued.
4. A regular update is registered for priority-1 location provider with values every 2 minutes in case if 5 meters distance movement was triggered.
5. Every updated location is checked to be within allowed accuracy. Accepted locations are used as current. Other location updates are rejected.
6. The maximum oldness of current location value is used to accept any location update in case if previously stored location is defined as “too old” by the system.
7. All registered listeners are unregistered when/if application is closed.

Chosen values were determined through average speed of movements around campus and resulted size of cells within the game grid. Approximation of cell size within grid is 73x70 m. An average walking speed would be around 3 km/hour, which is equal to 50m/minute. Taking into considerations principles of battery saving for mobile devices (more often updates result in battery drain) this exactly gives us a value of around 2 minutes between location updates to both get recent location, not drain the battery and expect that cell have been changed. It is true that part of students move between locations on campus via bicycles, which gives them higher speed and, therefore, faster inter-cell roaming. However, playing a game while controlling a bicycle would, probably, not be the best idea. Moreover, this means that Player will get less updates, yet the last update will be received from point of destination. In any case, certain inconsistencies or lags caused by exploits are considered tolerable. These possible inconsistencies may be caused high speed movement being close to North or South poles.

From design to prototype

First prototype implemented only several of the core features. Harvesting of resources and Crafting of resources were implemented. All game-map data were stored locally inside device in order to minimize server-client communication.

Player interacted with the application via a main Activity. This main Activity hosted three Fragments, that user was able to swipe through.

Fragment #1:

This fragment contained information about current location presented in simple text format.

Fragment #2:

This fragment presented user with information about resources stored in stock. All available resources with qualities.

Fragment #3:

This fragment informed Player about different Crafting possibilities. It contained crafting recipees and controllers.

A Service was developed and bound with this Main Activity. The main task of this Service was initiating a location update procedures in order to analyze the area user was present at.

A Broadcast Receiver was created and registered within Main Activity to catch broadcasted location updates and propagate information further to fragments.

Prototype supported a local mode exclusively, in a sense that no intercommunication between Players or between Server side was supported. This prototype was used as a proof-of-concept to investigate possibilities of location investigation within DTU and both precision and errors in received location data. This prototype was used during field-tests described above.

Implementation of resource harvesting

During game-design stage a concept of harvesting Resources was formulated. In order to support this idea Resources were created and divided into tiers to support Crafting concept from game-design stage.

Table 10: Virtual game Resources details

Resource tier	Resource name	May be harvested	May be crafted
1	Lumber	Yes	No
1	Stone	Yes	No
2	Coal	No	Yes
2	Ore	No	Yes
3	Magic powder	Yes	No
3	Iron	No	Yes
Ultimate	Gold	Yes	Yes

Moreover, in order to address aspect of “fun” an additional feature was implemented within harvesting. Harvesting was connected to accelerometer sensor of mobile device. Player was introduced with possibility to harvest more than one unit through shaking own mobile device. Accelerometer sensor API was accessed via Sensor manager available in Android platform. A pattern of measured values that corresponds to shake movement was loaded for comparison. Sensor was enabled during harvesting and disabled after harvesting was complete. A scale of shake counters was implemented to get a value from one to ten based on the number of shakes.

This was introduced in order to attempt to create a feeling of immersion that will be directly connected with a game and enrich game-process. In the meaning, while not being something unusual it attracts attention of others and is common within mobile app games. Although, respecting choices of potential Players it was added as a possibility, but not a requirement. Default harvesting without this feature was still possible and would return a default number of harvested resources, which was assigned a one unit.

Within game-design stage a concept of different locations was formulated. In order to support it a set of locations was implemented. Each location type defined a set of resources available for harvesting.

Table 11: Location types

Location name	Location id	Short Description
Ordinary location	0	Higher chances of getting Lumber.
Cave	1	Higher chances of getting Stone
Magic cave	42	Higher chances of getting Magic powder
Gold mine	111	Higher chances of getting Gold

Detailed description of harvesting function is presented below. All areas outside game-field are considered Ordinary locations. These locations are also present within game-field and are considered as most common areas. These locations have in approximation probabilities p=1/10 to get Gold or Magic Powder, p=2/10 to get Stone or p=6/10 to harvest Lumber. Therefore, lumber is most common tier-1 resource. Cave, Magic cave and Gold mine define harvested resource as Stone, Magic powder or Gold respectively.

Theoretical reasons behind such definitions are as follows. First, Player must have a choice to interact with game-environment even while outside game-field. Player must have a possibility to harvest a required necessary resource fast, assuming he knows a location of special location. Gold mines and Magic caves represent very important locations and Players would constantly compete to acquire them, as they are valuable capital rising sources. More adventurous Players ("explorers") would try to find these locations spread around the game map, while "challengers" would be involved in competition to acquire most of locations and easily accessible locations, especially.

These aspects have to be and were taken into account while assigning location types around the map, which may be seen in Appendix.

Implementation of resource crafting

Player is required to Harvest tier-1 resources and use them to craft tier-2 resources. Out of tier-2 resources tier-3 resources are crafted and so on up to ultimate tier-resource, which is used for all in-game trading operations.

Crafting combinations (all in equal units):

Lumber + Lumber = Coal

Stone + Stone = Ore

Coal + Ore = Iron

Iron + Magic powder = Gold

Proof-of-concept testing:

The main difficulty here was the precision of location data. Used methodology to achieve a relevant location information was proven as working. Location data that Android built-in location providers return are described by “accuracy” parameter that is an integer value in meters. Accuracy is defined as a radius of 68% confidence by Google. “In other words, if one would draw a circle centered at received location's latitude and longitude, and with a radius equal to the value of accuracy, then there is a 68% probability that the true location is inside this circle.” Location errors are believed to be random with normal distribution, however, Google warns that in practice they are not. [21]

Empirically it was found that accuracies returned by network provider inside gamefield area were between 40 and 70. This discovered a flaw in algorithm. All new locations were compared between themselves by accuracy. Therefore, in scenario where network provider started supplying locations from time t1 with accuracy value of 50 and higher, while current location's accuracy acquired at t0 before t1 was 45, all new location updates were discarded up until the point when current location was discovered as being too old (10 minutes gap). At that time t2 a next update was chosen as next best location. This meant that at time period t1:t2 while location was already changed system was still using a previous location, because accuracies of new incoming locations were less than current location's accuracy. Although total difference in accuracies was not that noticeable. This resulted in a redesign of comparison criteria from strict “new accuracy smaller or equal than old accuracy” to “new accuracy smaller or equal than old accuracy +30%”. Please, do not get confused by “smaller” in this context, as circle of smaller radius means more precise location.

21 http://developer.android.com/reference/android/location/Location.html#getAccuracy%28%29

From prototype to fully functional application

After the proof-of-concept prototype was tested a fully functional application was developed.

Architecture of the application, therefore, became more complex. New Activities, Broadcast Receivers and in-game classes have been developed.

Network operations:

An Android application runs in one thread by default. This thread is called a “main” or often a “UI” thread. Long lasting computations may block this thread, which will result in a ANR (Application Not Responding) crash. In order to make sure that long-lasting computations do not block this thread it is required to relocate all such operations in other threads. Android provides different possibilities for easy thread management. One of which is AsyncTask that provides developer with possibility to start computations asynchronously and after results are available to inform a main thread of available results. All network operations are performed with help of extending AsyncTask class. A new Broadcast receiver that is listening to results is registered within each interested Activity or Fragment. In order to differentiate between performed actions an identifier is issued for each action. Broadcast receiver checks the identifier when result are available and perform necessary game actions. This includes updating information about location, updating information about creature within location and all network operation in general.

MilitaryActivity:

MilitaryActivity performing as a Command Center within which Player can hire, train and command Creatures was developed. Initial design included a single creature type that Player could hire and place in a certain area to claim it. This concept was changed or rather improved in order to provide more choices, following definition of a game as set of multiple meaningful choices.

Creature-Types available in the final version at the time of this thesis publishing include three different types. Each of them is characterized by values of Strength and Health. Special creature also had ability to steal gold.

Table 12: In-Game creatures

Creature-Type	Creature-name	Default Health	Default Strength	Default price
Basic	Goblin	10	10	4
Advanced	Goblin-Knight	14	14	6
Special	Goblin-Spy	6	6	6

Strength and Health parameters are used by game logic to determine the strongest Creature if two are compared. Comparison of Creatures is performed during stages of attacking another creature in order to claim the area. A turn based strategy is used. A number of turns is calculated that would be required for Creature-A with its Strength-A to defeat a Creature-B with Health-2B. Example: It would take two turns for Creature-A with Strength-A equal to 5 to defeat a Creature-B with Health-B equal to 10. Values of turns are calculated for both Creatures and the one with smallest value is considered a winner. Case for equal Creatures is handled through random decision. Using random here follows another explained game-design principle stating that players tend to like uncertainty and randomness of events.

All creature's stats may be upgraded. Initially both Health and Strength have an upgrade step of value 2 for each next level with level 4 as maximum. This, for example, means that maximum stats for basic creature would be 16-16 in comparison to 10-10. Explanation of motivations behind chosen values are presented below.

The values were designed in a way that Basic creature after being upgraded have chances against Advanced not fully upgraded Creature. Special creature while upgraded will have chances against Basic creature. Fully upgraded Advanced creature will remain as the strongest and, therefore, the ultimate creature. Special creature had an ability to steal a static number of Gold from an attacking Player.

All in all, each creature my be represented by 4^2 combinations of values, where 2 is number of features and 4 is number of levels. This means that, all in all, there are $3 * 4^2$ possible combinations of values of Creature. Therefore, this is $(3 * 4^2)^2 = 2304$ different resulting combinations.

A research of possible 2304 combinations and possible outcomes was performed. A script was written to iterate through possible value combinations and chose a winner or mark result as equal chances for both participants.

Table 13: Results of battles for different Creatures

Special VS Basic			Special VS Advanced			Special VS Special		
Loses: 207	Wins: 3	**Equal: 46**	Loses: 256	Wins: 0	Equal: 0	Loses: 69	Wins: 69	Equal: 118
Basic VS Basic			Basic VS Advanced			Basic VS Special		
Loses: 60	Wins: 60	Equal: 136	Loses: 198	Wins: 3	**Equal: 55**	Loses: 3	Wins: 207	Equal: 46
Advanced VS Basic creature			Advanced VS Advanced			Advanced VS Special		
Loses: 3	Wins: 198	Equal: 55	Loses: 60	Wins: 60	Equal: 136	Loses: 0	Wins: 256	Equal: 0

The preliminary aim of balancing Special and Basic creatures was achieved. As a result Player received a choice of spending less resources and get a weaker Creature with smaller chances of victory, but earlier or spend some more time in order to prepare for the fight and get a stronger creature. This attempts to address different psychological types of careful and more risky players. Moreover, having a Special creature addresses a “fun” aspect and attempts to add surprise.

WorldMap and LocalMap:

Two different maps were added. A World-Map, representing global game area and investigated and currently controlled areas, and a Local-Map, representing closest areas 3x3 grid with player in center, was added to Fragment 1.

TopOfPlayers:

Top of Players representing a leaderbord of all players who control most of areas was added.

Change-decisions:

During development stage concepts were tested, proven correct or wrong. Using empirically achieved results and recommendations from first group of beta-testers certain decision changes were made and are presented below.

General changes and decisions:

Certain UI elements were added or removed. Some of Toast (pop-up messages in Android devices) were removed as unnecessary. Progress-bars were added to illustrate network communication and replace some Toast messages.

A limit for stockpile of resources was introduced. This limitation was considered as necessary to make sure that Players have to make decisions about Resources they want to keep and Resources they need to sacrifice in order to get something more valuable. This introduced a necessity to change Fragment 2 UI to implement a Drop resource ability. This was realized with Spinner UI elements.

In order to make sure that application is not confusing all UI elements were hidden from Player in cases when their functionality was not allowed by game environment or context. This includes: checking harvesting results if harvesting was not allowed, sending creature to claim an area while outside game-field, dropping resource while no resource were in hand, map of closest areas while being outside of game-field etc.

Changes in Creature aspects:

Price for upgrade of Health and Strength initially was the same and was measured in Gold. In order, to make sure that tier-3 resources do not lose they value, in comparison to ultimate tier resource the price for Strength upgrade was reassigned from price measured in Gold to price measured in Iron.

Static value of Gold stolen from other Player by Special creature was reassigned to dynamically calculated value that is a random number between zero and quarter of all available Gold.

Changes in Harvesting of resources:

A time constraint was added for harvesting and stocking new resources. Time constraint was considered as necessary in order to prevent filling of stock instantly. The value of time-constraint was assigned and adjusted empirically to 20 seconds. This value was decided as both equal to perform functions it was designed for and not disturb user experience.

Stone was marked as a Crafted resource. Player had the ability to craft this tier-1 resource from Lumber and Magic Powder. This was introduced in order to balance the out-of-gamefield areas and increase accessibility of this tier-1 resource in a non-direct. Now Players had more choices available, either to spend magic powder and get stone faster to get to Iron or keep it for later to get Gold.

In order to support time constraint a “shake” step was made wider. As a result Player would need to shake mobile device longer to harvest more than one unit of resource.

A game-map containing information about locations was removed from mobile application and migrated to back-end (described in details in back-end section).

Change in map Layout:

It was decided to change the layout of World-map activity from Satellite View (that was aiming at realistic representation) to schematic map view in order to increase readability of this interface.

Back-end

All client-to-client interactions within this project are implemented via Back-End solution. This section will describe the architecture of back-end server. It will mention choices made, discuss alternatives and consequences of choices made.

BaaS and MbaaS concepts

Initial choice was between establishing a server within own premises and using own equipment or sticking to services provided by third parties. At the time when this thesis is written a concept of BaaS (Back-End as a Service) and MBaaS (Mobile Back-end as a Service) are getting popular between developers. These concepts are described in this subsection.

The aim of BaaS/MBaaS providers and value they propose lays in removing all hassle related to designing, setting-up, monitoring maintaining and operating a back-end server from mobile application developers. Providers deliver a stack of services that usually includes storing information, messaging, providing push-notification services, user-management etc. Usually all provided features are accessible via simple Web API and/or included in distributed SDK, which makes management of service easier. This out of box solution makes it faster for mobile application developers to begin providing own service to customers. Moreover, BaaS/MBaaS providers take care of accessibility of resources, scalability of provided service and monitor performance of provided services while promising to take care of security as well[KLRM].

This market segment is developing and developing fast, which illustrates that the value these providers present is indeed appreciated by community. Some numbers illustrating and proving this point are as follows. Kinvey, one of companies that provides BaaS received $5 million of financing in 2012, after receiving $2 million in 2011.[AWKR] Parse, a company that provides MBaaS raised $5.5 million of financing in 2011 and claimed that mobile traffic they serve grew 40% week after week. StackMob, another company providing similar solutions, received $7.5 million in May of 2011[DRMB].

All in all, according to MarketsandMarkets (a global market research and consulting company from U.S.) “The global BaaS market is estimated to grow from $216.5 million in 2012 to $7.7 billion in 2017”. (According to: The report “Cloud Backend-as-a-service (BaaS)/ Mobile BaaS (MBaaS) Market - Global Advancements, Business Models, Technology Roadmap, Forecasts & Analysis (2012 – 2017)”, published by MarketsandMarkets).

Pros/Cons of each choice

The benefits of using services of BaaS/MBaaS providers are explained. Yet, one may ask about drawbacks. Drawbacks of hosting your code inside the cloud may include certain loss of control over the service and certain reliance and/or dependance from service provider owners. Change of conditions of subscription plans, change of pricing policies. All these things may occur and have to be considered. Yet, within project deadlines and budgets, BaaS/MBaaS often remain a good solution for simple proof-of-concept projects, easily migrating projects or just as a mean to start quickly and non-painfully.

Choices between BaaS

A comparison between provided Free-subscription plans taken from official Web-Pages of each company in Autumn 2012/Winter 2013 is presented below.

Table 14: Comparison of BaaS providers

Parse	StackMob	Google App Engine	Kinvey
-Custom code in cloud -Twitter, Facebook Integration -User-management -Crossplatform -Requests: 1 million/month -Pushes: 1 million/month -Files:1 Gb	-Custom Code -(Basic) Access Control Lists -Analytics -Twitter, Facebook Integration -Geoqueries Work Flow (Basic) -Access to the StackMob MarketplaceTM	-Custom code -Analytics -Python/Java RTE -Google Plugin for Eclipse IDE -Image Manipulation -Abundance of Free API -Cron -Logging -Datastore 1G -Outgoing Bandwidth 1G -Incoming Bandwidth 1G -Datastore API: 50k free read/write -month Email API 100 recipients	-Analytics -Security -Facebook, Twitter integration -Crossplatform -up to 1 mil API requests for Web-API -up to 200 active users for mobile -Files: 2 Gb

It may be noticed that advertised features of services provided by competitors are often similar and differences are not easily noticeable. Most go for cross-platform scalable solutions with data-storing possibilities, push-notification, integration with different most popular social media (such as Twitter and Facebook).

However, taking into consideration choice of Android as a mobile platform and design features, GAE was chosen as a solution for hosting the back-end for this project. Main reason for this choice is fact that Android is a platform developed by Google. Therefore, being cross-platform is not even a requirement. Yet, design of the actual implementation will show that cross-platform nature of the service is reached anyway. Moreover, Google constantly develops and enhances both GAE and Android and their integration. Google provides SDK and special plugins for IDE to developers, which results in easy deployment of system. Google has more educational resources to provide developers with, which is very relevant in case of this project. In addition Google is also working on the Google Cloud Endpoints, which is an experimental, innovative, and rapidly changing new feature for Google App Engine to develop mobile back-ends.[22] GCE was still in experimental mode during this thesis preparation and, therefore, was not used, as it was not recommended to use it for actually running public projects due to possible instabilities.

Drawbacks of this choice that were noticed during testing:
In several cases a longer latency of server response was noticed. Investigation showed that this might be caused by several reasons. First, free-subscription within GAE implies that data is hosted in USA. While most of the mobile agents within testing were located in Europe, Denmark. Distance means latency. It is possible to move hosting in Europe, yet only as a paid service. Second, service is not running constantly it is being instantized by "first" request and is kept running while/if next requests are coming. This causes a longer latency for response to this starting request.

GAE detailed description

Out of provided run-time environments (Java, Python, Go) Java was chosen due to author's personal preferences, skills and tools available. GAE uses Java 6 virtual machine (JVM) to run Java applications. Provided SDK supports Java 5 and later. JVM can use classes compiled with versions up to Java 6. Java Servlet standard is used for web applications. Developers provide servlet classes. JSP, data files, static files, other configuration files and deployment descriptor. Application is isolated in own "sandbox" to ensure security and quality of service. This is implemented to make sure that application is not performing actions that interfere with scalability and performance of other apps. For example, application is not allowed to make arbitrary network connections, write data to local file systems, etc.

Two different options for data storage were provided initially: Master/Slave and High replication. Due to budget of the project (no budget at all) a free

22 https://developers.google.com/appengine/docs/java/endpoints/

subscription was chosen. Free subscription implies usage of High-replication option. Moreover, since April 4, 2012 the Master/Slave Datastore has been deprecated in favor of the High Replication Datastore (HRD), in which data is replicated across multiple data centers.[23]

Datastore supports two standard Java interfaces: JDO 2.3 and JPA 1.0 that are implemented using DataNucleus Access Platform, the open source implementation of these standards. Memcache can be used for fast, transient, distributed storage for caching results of queries of datastore and calculations. URL Fetch service is used to access resources over the web and to communicate with other hosts via HTTP/HTTPS. Mail service is available to send email on behalf of administrators of the service. Image service for manipulating Image data is available. Google accounts may be used for authentication.[24]

From mockup to version 1.0

During the design stage server side was assigned to be responsible for all Player-to-Player interactions within designed game environment. Within designed game environment that corresponded to such game actions as placing a Creature at a location, checking if there is a Creature placed at location, updating a Creature at a location. In order to implement this functionality a Database storing Creatures with all required attributes was required.

Choice between JDO and JPA was complex. After performed research it was difficult to state that one is functionally better or more suitable for tasks than the other. Factor that became decisive was of educational nature. Author of this thesis simply found tutorials and educational researches on JDO easier to follow.

Version 1.0 was developed. This version implemented the backbone functionality that is supporting Player-to-Player interaction.

Description of version 1.0:

Having in mind principles of database normal forms Creature was chosen as an object that is made persistent and stored. It includes:

23 https://developers.google.com/appengine/docs/java/datastore/usingmasterslave
24 https://developers.google.com/appengine/docs/java/overview

Table 15: Creature Entity

Creature	
id	Unique id of entry
Name (not used)	String name of Creature
Type	String type of Creature
Location	String location within Gamefield
Health	Integer Health value
Strength	Integer Strength value
Owner_Alias	String alias of player that owns the Creature
Owner_ActualUsername	String actual username of player that owns the Creature
Welcome_Message (not used)	String message a Creature would demonstrate

As it may be seen several of the fields were placed in order to support potential growth and change towards socialization of the game. It is believed that personalization would increase the immersion inside the game-world, so as the social aspect of interchanging messages via placed creatures. A possibility to place a message together with a Creature would change the nature of the game-world towards more social and address Socializers game type. Good example of this would be a Creature displaying “Player 1 is the best!” or “Let's go for a coffee now!” messages. However, due to obvious possibilities of misuse it was decided to keep it out of the game, for now.

In order to access and manipulate Creature objects several Servlets were designed and afterwards implemented. These servlets are triggered with HTTP GET requests with predefined parameters. This actually means that solution is cross-platform as it doesn't matter which device originates an HTTP request. Moreover, as a means of version control, ensuring that mobile application is using the last version of the Server side and as a certain security check a “license key” parameters are included in all requests. Server logic compares the key before all of the other processing in order to check user's authorization to perform actions or inform player about necessity of reinstalling the application.

Two different requests exist: read and write requests. Two different methods of processing were developed:

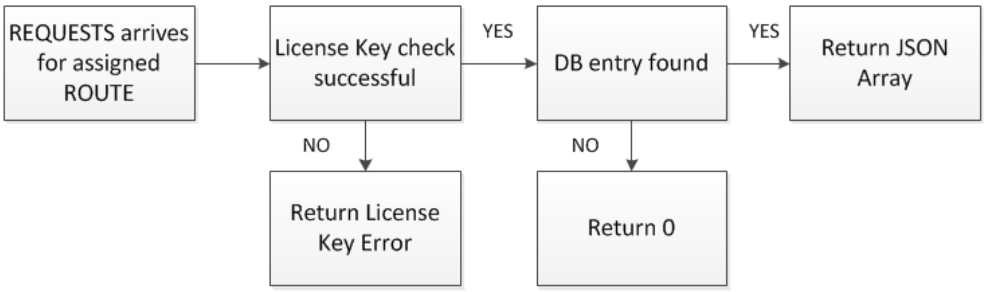


Illustration 6: Read-Queries processing

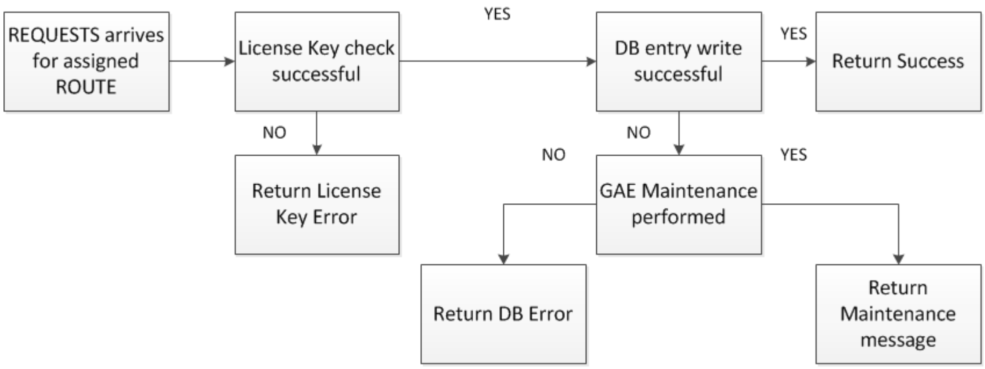

REQUESTS arrives for assigned ROUTE
License Key check successful
YES
DB entry write successful
YES
Return Success
NO
NO
NO
YES
Return License Key Error
GAE Maintenance performed
Return DB Error
Return Maintenance message

Illustration 7: Write-Queries processing

Servlets of version 1 include:

PopulateCreatureServlet

This servlet that is used to insert an entry in the database.

Parameters: “license key”, “type”, “location”, “strength”, “health”, “owner_alias”, “owner_actualUsername”.

Processing: according to Write-request.

GetCreatureAtLocationServlet

This servlet that is used to get an entry of Creature from the database.

Parameters: “license key”, “location”

Processing: according to Read-request.

UpdateCreatureServlet

This servlet that is used to update an entry in the database.

Parameters: “id”, “license key”, “type”, “health”, “strength”, “owner_alias”, “owner_actualUsername”

Processing: according to Write-request.

Descriptor contains routing information to trigger each servlet by requesting a specific URL.

Nature of game-play implies maximum 308 (14x22) entries of Creature type at one time in the database. It is impossible to populate a new creature at location, if another creature is already at this location. An update must be used in this case.

From version 1.0 to version 2.0

Change-Decisions: General errors, such as database write errors, service unavailable errors and wrong-license key errors had initially different Error codes for different servlets to differentiate between and originate a source of error easier. However, it was chosen to replace them by general error codes same for all servlets as a mean to make design prettier, manageable and do not over-complicate the design of front end as all responses had to be dealt and processed by mobile application gracefully.

In order to make game-world easily configurable it was decided to relocate the game-field information from mobile application to server side. This choice resulted in redesigning of the back-end logic. Pros of this choice include: simpler mobile app logic, configurable game-field (different cells could be reconfigured without changes in mobile agents. Cons of this choices include: complication of server side, increasing the amount of request-responses and network communication.

Having in mind principles of database normal forms Cell was chosen as an object that is made persistent and stored. It includes:

Table 16: Cell entity

Cell	
id	Unique id of Entry
type	Integer value that describes the type of location
typeDescribed	String that contains the name of this type of location
location	String location within Gamefield

In order to access and manipulate Cell objects several Servlets were designed and afterward implemented. These servlets are triggered with HTTP GET requests with predefined parameters.

In addition to Version 1.0 servlets next servlets have been implemented:

PopulateCellServlet

This servlet is used to populate a Cell into datastore. This Servlet is used by Administrators of the System and is not revealed to User.

Parameters: “license key”, “type”, “type-described”, “location”

Processing: according to Write-request.

GetCellAtLocationServlet

This servlet is used to get a Cell from datastore with a given location.

Parameters: “license key”, “location”

Processing: according to Read-request.

UpdateCellTypeServlet

This servlet is used to update a Cell entity in the datastore. This Servlet is used by Administrators of the System and is not revealed to User.

Parameters: “license key”, “id”, “type”, “type-described”, “location”

Processing: according to Write-request.

From version 2.0 to version 3.0

Change-Decisions: In order to enhance and adjust gameplay and provide richer feedback to players certain changes were introduced to both back-end and front-end. The concept of Creature dying of old age was introduced. Moreover, the concept of controlled area was generalized from “an area where a Creature of certain Player is placed” to “an area claimed by Player by placing a Creature”. First concept is introduced in order to make easier entrance of new players. Creatures on the map keep losing their health with every day and, eventually, when health drops to 0 are deleted from the map. Moreover, this ensures that Players have to control stats of their Creatures and replace Creatures that are getting older with newer stronger Creatures. Change introduced in second concepts is partially caused by first concept and explains that Player by placing a Creature claims a certain area of the map. However, even if there will be no Creature in this area, and no other Player claimed this area, Player will still be in control of the area.

Two additional fields were added to Cell entity to implement this idea.

Table 17: Cell Entity (remastered)

Cell	
id	Unique id of Entry
type	Integer value that describes the type of location
typeDescribed	String that contains the name of location
location	String location within Gamefield
owner_alias	**String alias of player that controls the Cell**
owner_actualUsername	**String actual username of player that controls the Cell**

In addition to Version 2.0 additional servlets were designed and implemented:

UpdateCellOwnerServlet

This servlet is used to update owner_alias and owner_actualUsername parameters of Cell.

Parameters: “license key”, “id”, “owner_alias”, “owner_actualUsername”

Processing: according to Write-request.

GetCellByOwnerServlet

This servlet is used to return a list of controlled areas that are controlled by Player with owner_alias and owner_actualUsername parameters.

Parameters: “license key”, “owner_alias”, “owner_actualUsername”

Processing: according to Read-request.

RegularDecreaseHealthServlet

This servlet is used to decrease health for all Creatures that are populated in datastore and delete entities in case if health drops to 0. This servlet is linked to a cron task and is performed automatically every day at 03:00 AM. Time was chosen to make as this is a time of minimal expected activity of Players. Servlet is limited to Administrators use and is hidden from users.

From version 3.0 to version 3.1

Change-Decisions: Addressing a necessity to implement such important game element as a leaderbord a new version was released. Leaderbord in this case is a list of Players sorted by number of controlled areas in descending order. This version was the one that encountered a limitation imposed by choice of GAE. What could be easily implemented by SQL COUNT is not easily available in the cloud. In order to implement it author had to develop a simple sort mechanism in the cloud.

GetTopOfPlayersServlet

This servlet constructs a leaderbord explained above.

Parameters: no parametes.

Processing: according to Read-request.

Moreover, problems were noticed with results that contained symbols different than English alphabet. UTF-8 was used in order to address this issue.

Assessments and possible alternatives:

Architecture of database structure is influenced by No-SQL principles of creating a database, where types of queries define the architecture more than relational principles and normalization rules. Author of this thesis was inspired by MongoDB[25] Author of this thesis considers a proper realization of the database based on relational database concepts a possible alternative solution. However, it may also be a possible overkill during both design, implementation and operation stages. Such solution would remove all inconsistencies issues, yet our system can tolerate some inconsistencies. Such system would require such entities as Cell, Player, Creature, Cell-Type, Creature-Type and despite complexity would be solving same tasks as solution that was designed.

Implementation stage conclusions:

This section provided the explanation of what front-end and back-end solutiona are used within designed game environment for. It explained motivations behind choice of Android as a platform for front-end and GAE for back-end during implementation stage and provided list of alternative approaches. This section described the architecture and its development from mockup to last published versions of both parts of architecture with detailed explanations of all change-decisions made.

25 http://www.mongodb.org/

Chapter 4

Results analysis

The overall conclusions about the projects are made based on the results gathered after an actual launch of the game-environment and analysis of these results. This section provides the total overview of all gathered results, research methods used and their assessment.

Methodology

During the testing of developed game-environment multiple different aspects related to this project were tested. They may be characterized and grouped by two main conceptual layers: physical layer and logical layer. Physical layer includes testing of application and server performance, reliability, appearance etc. Logical layer includes identifying the initial users interest pattern, interest retention, analysis of spatial and temporal data related to users activity, etc.

The testing and results gathering was performed in two steps.

First-Step: Closed testing. A small group of users volunteered to test all aspects of physical layer. The primary focus of this step was to analyze physical layer aspects, test concepts, locate and troubleshoot problems and "show-killers" in order to prepare the game-environment for publishing.

Second-Step: Open testing. After publishing of game-environment in public space an open-call for testers was announced via DTU internal electronic communication platform CampusNet. Moreover, a broadcasted message was sent to participants of Mobile Application Prototyping course that took place in Spring 2013 semester at DTU. This message informed about the game and possibility to join the testing.

Some of the results and outcomes of both layers were already discussed in the implementation section as they are reasons for many of the change-decisions that were made during development of the product. Other results are presented below.

Results of the First-Step analysis:

As it was already stated above, this step was more focused on aspects defined and grouped in Physical layer of game-environment functionality. Number of test-users involved: 8. This step have been performed during January-February 2013. All test-users received an application by mail and installed it on their current mobile devices used. This step was finished with application publishing and beginning of Second-step.

The table below proves the variety of Android devices that exist out there:

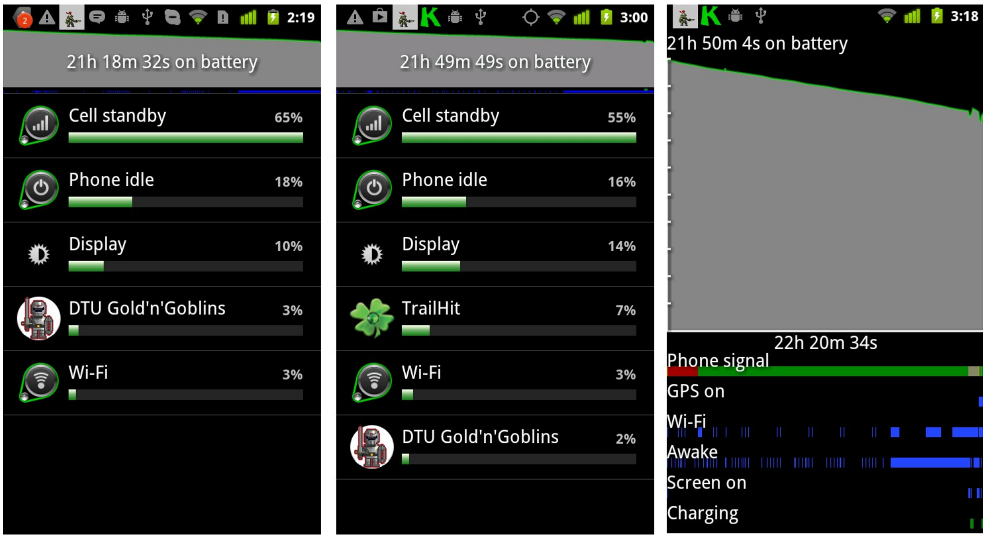
Table 18: Equipment used in First-Step analysis

Test-users	Model of smartphone	Android version	Screen size (in inches)	Type
User#1	HTC Desire	Android 2.3.7	3.7''	Smart-phone with portrait layout
User#2	Nexus S	Android 4.1.2	4''	Smart-phone with portrait layout
User#3	Sony Ericsson Xperia arc S	Android 4.0.4	4.2''	Smart-phone with portrait layout
User#4	Nexus S	Android 4.1.2	4''	Smart-phone with portrait layout
User#5	HTC Wildfire S	Android 2.3.5	3.2''	Smart-phone with portrait layout
User #6	Samsung Galaxy Tab 10.1	Android 4.1.1	10.1''	Tablet
User #7 Maxter	Sony Xperia S	Android 4.0.4	4.3''	Smart-phone with portrait layout
User#8 Kid	Sony Ericsson live with walkman wt19i	Android 2.3.4	3.2''	Smart-phone with portrait layout

Front-end performance:

Front-end performance was assessed by analyzing battery performance, memory usage.Most of devices that were used in testing were smart-phones. Users of these smart-phones haven't reported any significant errors or problems with user interface. However, comments from users regarding interface were used as a base for decision changes presented and discussed in implementation section.

Battery consumption: Battery consumption was measured for developed mobile application. Several measurements were performed in order to investigate the actual battery consumption of mobile application. Moreover, battery consumption of another location based game “TrailHit”[26] that relies exclusively on data from GPS sensors for location discovery was compared with battery consumption of developed game-environment.


2:19
21h 18m 32s on battery
Cell standby 65%
Phone idle 18%
Display 10%
DTU Gold'n'Goblins 3%
Wi-Fi 3%

3:00
21h 49m 49s on battery
Cell standby 55%
Phone idle 16%
Display 14%
TrailHit 7%
Wi-Fi 3%
DTU Gold'n'Goblins 2%

3:18
21h 50m 4s on battery
22h 20m 34s
Phone signal
GPS on
Wi-Fi
Awake
Screen on
Charging

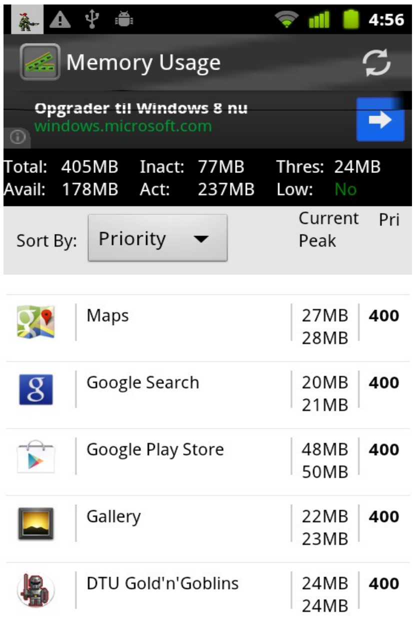
Illustration 8: Battery consumption for DTU GoldNGoblins

Results of this comparison prove the advantages of using data available from Network provider for location discovery in regard to battery saving. Overall, no particular drawbacks were noticed in battery consumption patterns.

Memory consumption: The analysis of application memory consumption have not presented any memory leaks. Before package aligning package occupies 1.96 Mb of space. After relocation to SD it occupies 768 Kb both in phone memory and SD. Aligning of application with zipalign tool[27] decreased consumed space to 1.12 Mb. The consumption of RAM

26 https://play.google.com/store/apps/details?id=com.f5tbl.trailhit
27 http://developer.android.com/tools/help/zipalign.html

according to Android Memory Usage tool[28] is also within comparison to other applications:


4:56

Memory Usage

Opgrader til Windows 8 nu
windows.microsoft.com

Total:	405MB	Inact:	77MB	Thres:	24MB
Avail:	178MB	Act:	237MB	Low:	No

Sort By: Priority

	Current Peak	Pri
Maps	27MB 28MB	400
Google Search	20MB 21MB	400
Google Play Store	48MB 50MB	400
Gallery	22MB 23MB	400
DTU Gold'n'Goblins	24MB 24MB	400

Illustration 9: RAM consumption for DTU GoldNGoblins

Back-End performance:

During First-Step measurements on the back-end were performed to understand the patterns of server-client communication. The measurements for one active player were taken first. One player could not create enough activity on the server to cross at least the 0.5% of assigned limitations by Google for free applications. Latency, however, was reported already at this stage. Reasons for latency were discussed in implementation section.

Results of the Second-Step analysis:

Second-Step began at 20th of February 2013 with publishing of an application. All test-users from First-Step received link to published latest version of application with full sets of instructions. This was required as during First-Step some decision changes were implemented.

28 https://play.google.com/store/apps/details?id=mem.usage

By “publishing of application” making application public and available to all interested users via Google Play market (official distributing system for Android applications that is running and accessible mostly worldwide with some exceptions) is understood. Applications are available either for free or at a cost directly from Android devices via Android Play client application.

Several key-actions are required before publishing.

1. Preparation stage:

This stage includes careful choice of package name, as this name should be suitable for the whole life of application. The name should be unique. It is used to uniquely globally identify application among others and should generally reflect and use domain ownership principle. Name of package was assigned as: uadk.dtu.wdnsd.gng (uadk stands for Ukraine-Denmark, dtu representing DTU as origin and relationship, wdnsd stands for Wednesday and is caused by small fact that surname of author of this thesis may be translated as Wednesday from Ukrainian to English, gng stands for GoblinsNGold). All logging and debugging was switched off. All graphic resources required for application and application information on market checked and uploaded. Version codes required for version control were checked and assigned to 1.0 as a starting point. End-User License Agreement prepared as suggested by Google. Official name approved and chosen: “DTU GoblinsNGold”[29].

2. Signing stage:

“The Android system requires that each installed application is digitally signed with a certificate. This certificate is owned by the application's developer and application holds a private key for it. The Android system uses the certificate as a means of identifying the author of an application and establishing trust relationships between applications. The certificate that is used for signing does not need to be signed by a certificate authority; the Android system allows developer to sign applications with a self-signed certificate.” Certificates need to be valid at least up until 22th October 2033.[30] Certificate was created and application was signed.

29 http://developer.android.com/tools/publishing/publishing_overview.htm
30 http://developer.android.com/tools/publishing/app-signing.html

3. Pre-Publishing stage

A developer account needs to be created and linked to Google Play. The fee for registration was 25$ and this was the first actual investment made in the project apart from personal and DTU equipment used for development. Approval procedures from Google took more than 100 hours of awaiting for decision time. Policies of application being Free for all users and No advertisements were chosen.

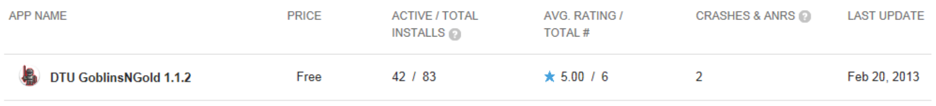
4. Publishing stage

Application was published.

Google Play data analysis:

Logging of data was performed at two points of interest (POI): Google Play and Google App Engine.

Data stored at Google Play POI represents all data about application distribution in time. The most important data and its analysis is presented below.



APP NAME	PRICE	ACTIVE / TOTAL INSTALLS	AVG. RATING / TOTAL #	CRASHES & ANRS	LAST UPDATE
DTU GoblinsNGold 1.1.2	Free	42 / 83	★ 5.00 / 6	2	Feb 20, 2013

Illustration 10: Statistic on application Installs (extracted 28 Feb 2013)

Active Device Installs represent the number of unique active devices where the app is currently installed. Total User Installs represent the total number of unique users who have ever installed this app on one or more of their devices.

The detailed day-to-day Install information up until 25th (26th) February 2013 is presented below. It represents the initial interest curves from potential customers. Description of the application was clearly stressing principles of the game and fact that most of the game-actions required being present at DTU campus area.

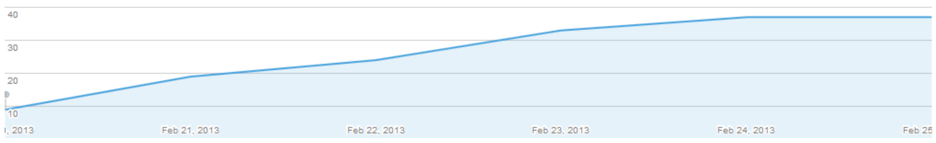
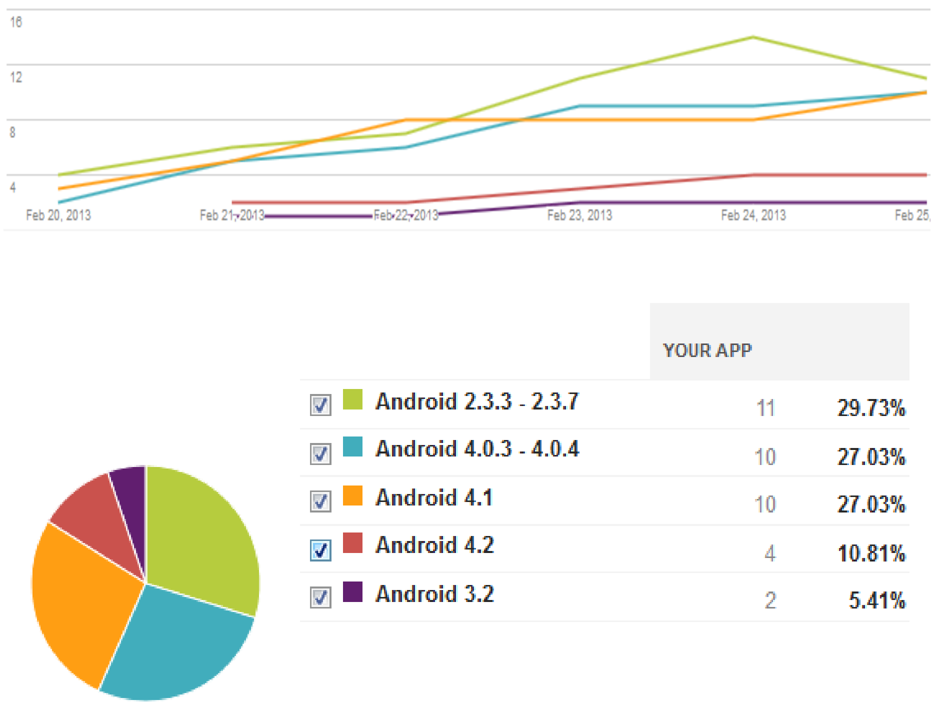


Illustration 11: Active Installs until 25 February 2013

The growth and later stabilization of the Install-number, which is the Initial Interest curve for this case, is explained due to lack of marketing actions. These actions were decided to be postponed until the results of Second-Step analysis would be gathered and assessed.

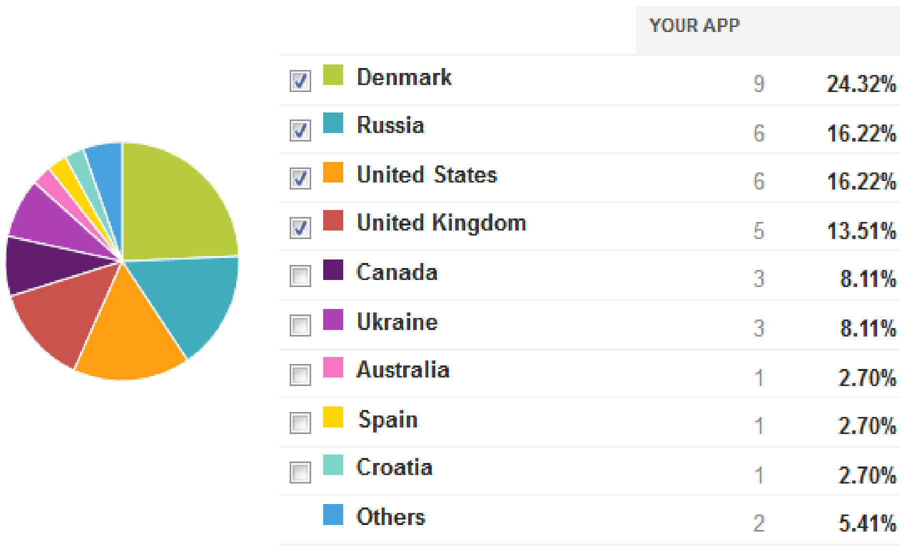
Next chart represents the trends of available mobile platforms at stock. Majority of users still use Android 2.3.3 – 2.3.7 version. This proves statements and assumptions made during implementation stage.



		YOUR APP	
☑	🟩 Android 2.3.3 - 2.3.7	11	29.73%
☑	🟦 Android 4.0.3 - 4.0.4	10	27.03%
☑	🟧 Android 4.1	10	27.03%
☑	🟥 Android 4.2	4	10.81%
☑	🟪 Android 3.2	2	5.41%

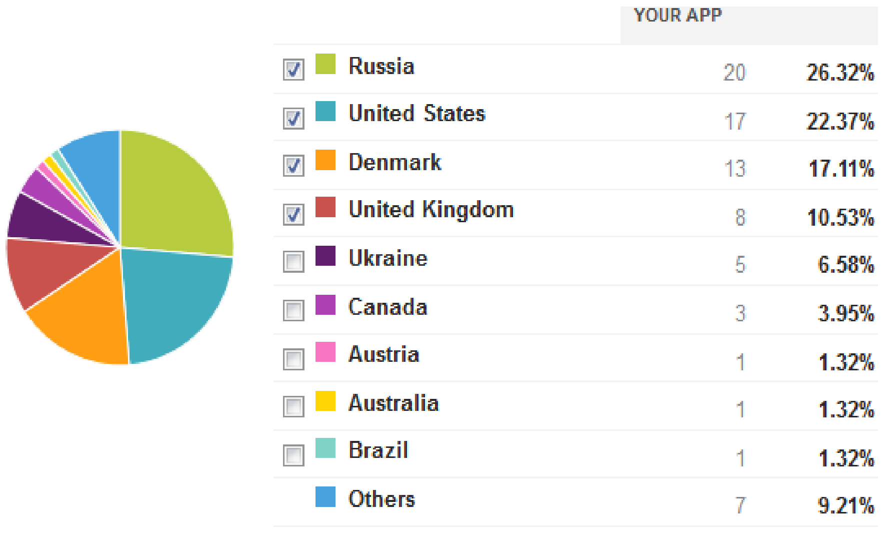
Illustration 12: Active Installs by Platform

The geographical distribution of active installs is not so obvious. Denmark is in the leading position as game-environment was designed and tested in Denmark. 3 of Ukrainian active installs are some of test users from First-Step analysis stage and other interested parties. All other countries represent pure Interest curve research as application was not advertised in any manner. In order to analyze geographical patterns of distribution of potential players the total installs by geographic area is presented.



			YOUR APP	
☑	🟩	Denmark	9	24.32%
☑	🟦	Russia	6	16.22%
☑	🟧	United States	6	16.22%
☑	🟥	United Kingdom	5	13.51%
☐	🟪	Canada	3	8.11%
☐	🟪	Ukraine	3	8.11%
☐	🟪	Australia	1	2.70%
☐	🟨	Spain	1	2.70%
☐	🟩	Croatia	1	2.70%
	🟦	Others	2	5.41%

Illustration 13: Active installs at 26-th February 2013 by country



			YOUR APP	
☑	🟩	Russia	20	26.32%
☑	🟦	United States	17	22.37%
☑	🟧	Denmark	13	17.11%
☑	🟥	United Kingdom	8	10.53%
☐	🟪	Ukraine	5	6.58%
☐	🟪	Canada	3	3.95%
☐	🟪	Austria	1	1.32%
☐	🟨	Australia	1	1.32%
☐	🟩	Brazil	1	1.32%
	🟦	Others	7	9.21%

Illustration 14: Total Installs up until 26-th February 2013 by counrty

If we compare the charts Denmark has moved to third “bronze” position with 13. These 4 users of difference with active installs may be the potential players that have been lost during the week. This represents the Interest retention. In order to attempt to understand what happened a look into daily change of installs have to be made.

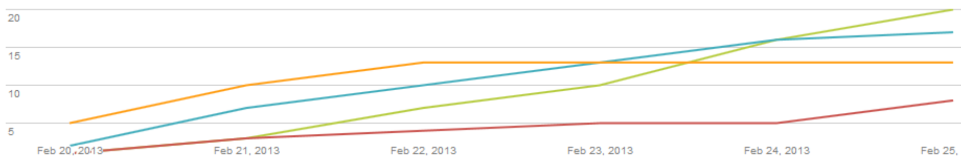


Illustration 15: Total Installs during the February 20 - 25 period by country

The growth and stabilization of installs for Denmark (orange) is explained by the start of Second-Step analysis and information about application being sent to students of Mobile Application Prototyping course at DTU. This may be explained with statement that students got interested by application description and started installing application to check it out. Data from other countries, however, represent a stranger data set. While users from USA (blue) represent a steadily growing pattern it is impossible to state clearly what happened at February 23rd that provoked sudden growth of installs from Russia (green) that resulted in doubling of the number of installs during the next two days.

GAE data analysis:

Data stored at the Google App Engine (POI) represents the actual patterns of Front-End/Back-End communication that represent both layers of analysis. Latencies, patterns of users' requests, reads and writes from/to database represent aspects of physical layer. The actual content of these requests represent aspects of logical layer. Spatial and temporal data shows behavioral patterns of players. All this is represented below.

During the Second-Step analysis all quotas defined by Google to applications hosted as free were not reached.

The logged performance values for one game activity day (February 25-th) are presented below.

Frontend Instance Hours	18%	5.08 of 28.00 Instance Hours

The reached percentage of total numbers of Datastore Write and Read operations from total number is presented below.

Datastore Write Operations	1%	of 0.05 Million Ops
Datastore Read Operations	2%	of 0.05 Million Ops

The detailed analysis for types of Reads and Writes is represented below. As we see the task of minimization of writes was achieved.

Table 19: Reads/Writes to the Datastore

Datastore Entity Fetch Ops		552
Datastore Entity Put Ops		91
Datastore Entity Delete Ops		0
Datastore Index Write Ops		658
Datastore Query Ops		232
Number of Indexes	0%	1 of 200
Code and Static File Storage	5%	0.05 of 1.00 GBytes

Table 20: Types of requests performed:

Type of Requests	**Requests** (during 15 hrs)
Get Info about Location	149
Get Creature at Location	28
Update Location	27
Put or Update Creature	27
Get Top Players	9

Overall performance during all Second-Step Analysis is presented next. On these charts on the timescale “now” mark corresponds to 26-th Feb 2013.

Requests/Second:

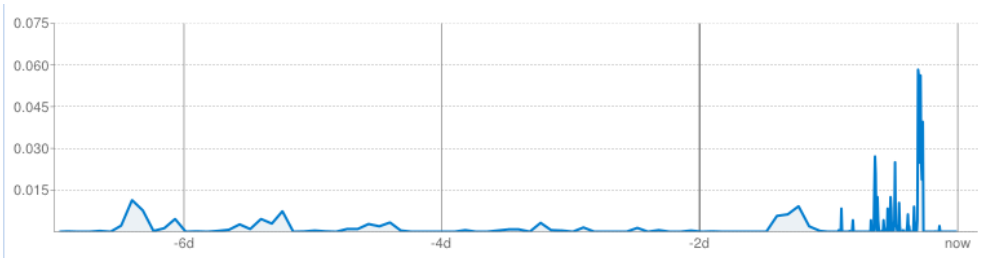


Illustration 16: The number of URIs requested from application every second, including dynamic, static, and cached requests.

These values show that currently intensity of requests are not high, which is explained by the low number of players.

Milliseconds/Request:

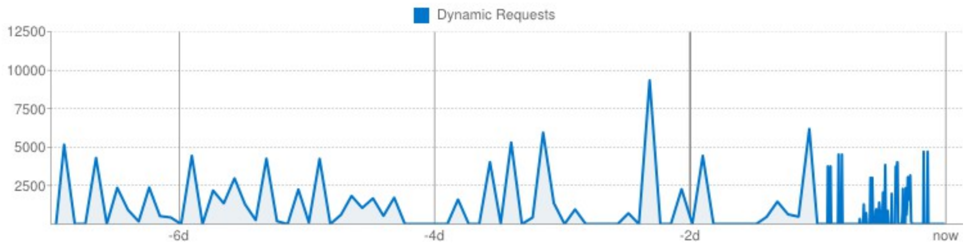


Illustration 17: The average number of milliseconds application takes to service a request (latency measure)

Maximum latency value registered is approximately ten seconds. Reasons for latency were discussed in implementation section. This value is too large for mobile application. The value of around 5 seconds is used by Android system to trigger an ANR error dialog for user input and 10 seconds for broadcast receiver event.[31] These latencies experienced in network communication will not trigger an ANR. However, unfortunately, would be noticed by players and disrupt user experience.

31 http://developer.android.com/training/articles/perf-anr.html

Errors/Second:

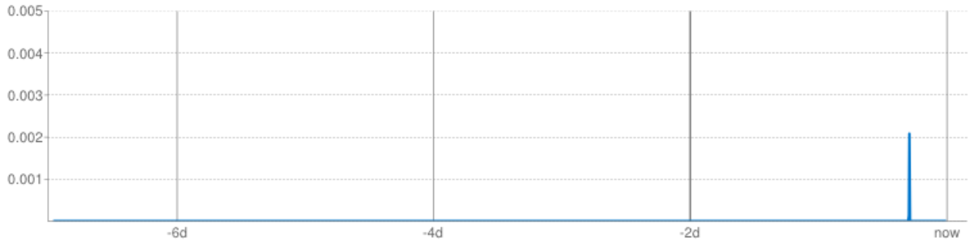


Illustration 18: The number of errors generated by application every second

Only one error of level Warning was generated during Second-Step analysis. Investigation of reasons of this single not repeated error yielded no results.

Memory Usage (Mb):

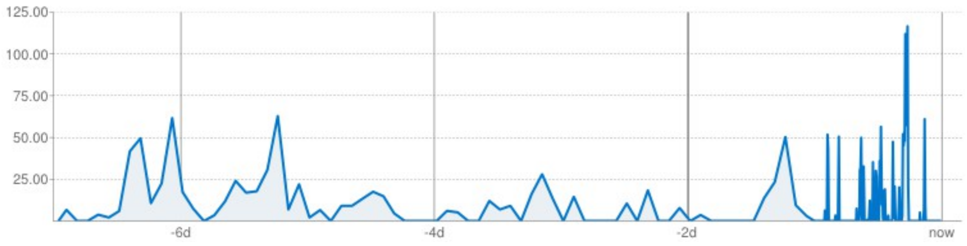


Illustration 19: Memory Usage of application (Mb)

Number of Quota Denials/Second:
No Quota Denials were registered during whole testing period.

Bytes Received/Second and Bytes Sent/Second:

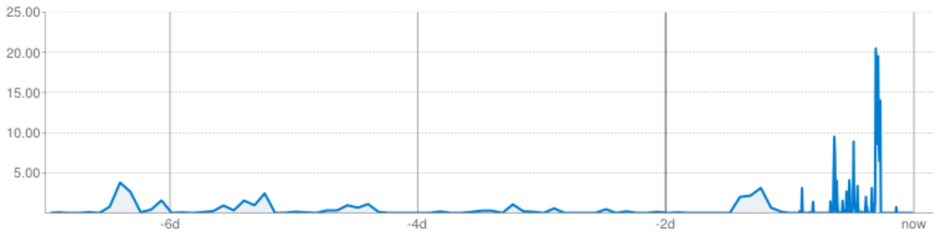


Illustration 20: The number of bytes that are received in a request every second

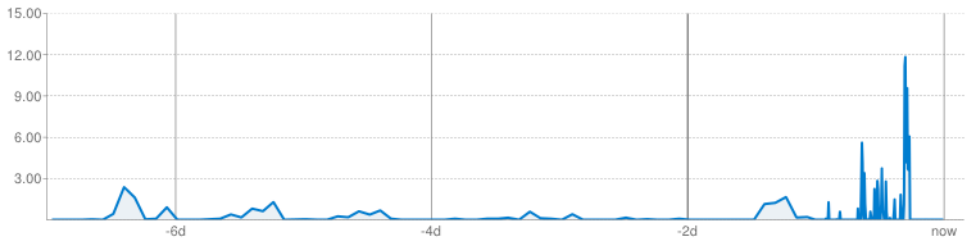


Illustration 21: The number of bytes sent in a request every second

It may be seen that charts are perfectly aligned and peaks in one correspond to peaks in another. Differences in values are related with simplistic structure of replies Back-end returns and necessity to inform more data in requests from Front-end (alphanumeric "license" key is sent with every request, player specific data are sent in order to update datastore entries).

All in all, there were collected more than one thousand log entries of game actions performed by Players during Second-Step analysis. Spatial and Temporal analysis of data is presented below:

Spatial data analysis:

Spatial data analyzed represent players activity during Second-Step analysis. All location related requests to Back-End were analyzed. First set of images shows locations from where requests were originated for every day of Second-Step analysis. An overall overlay is also presented to show the dynamics of players investigation of area.

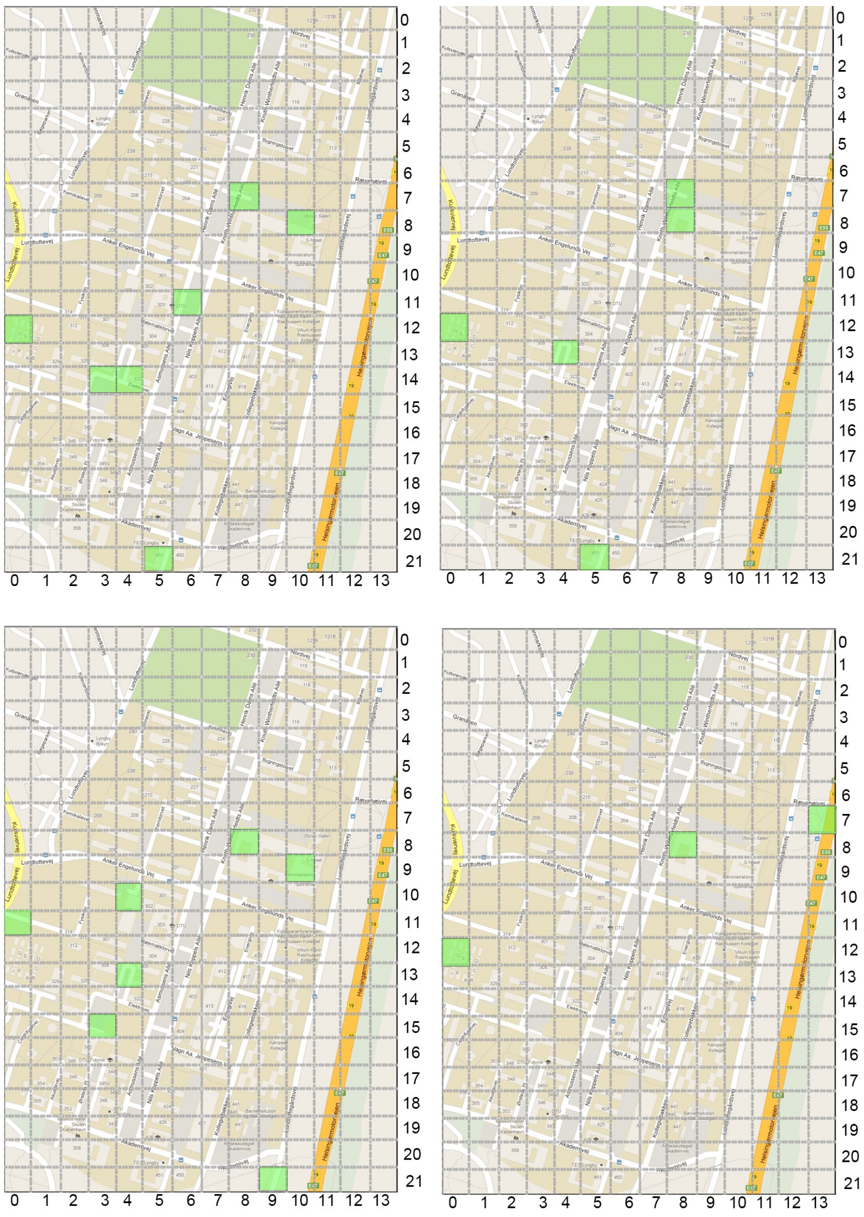


Illustration 22: Locations of all location-related requests from players representing area investigation and conquering: Top-Left: Day one, Top-Right: Day two, Bottom-Left: Day three, Bottom-Right: Day four

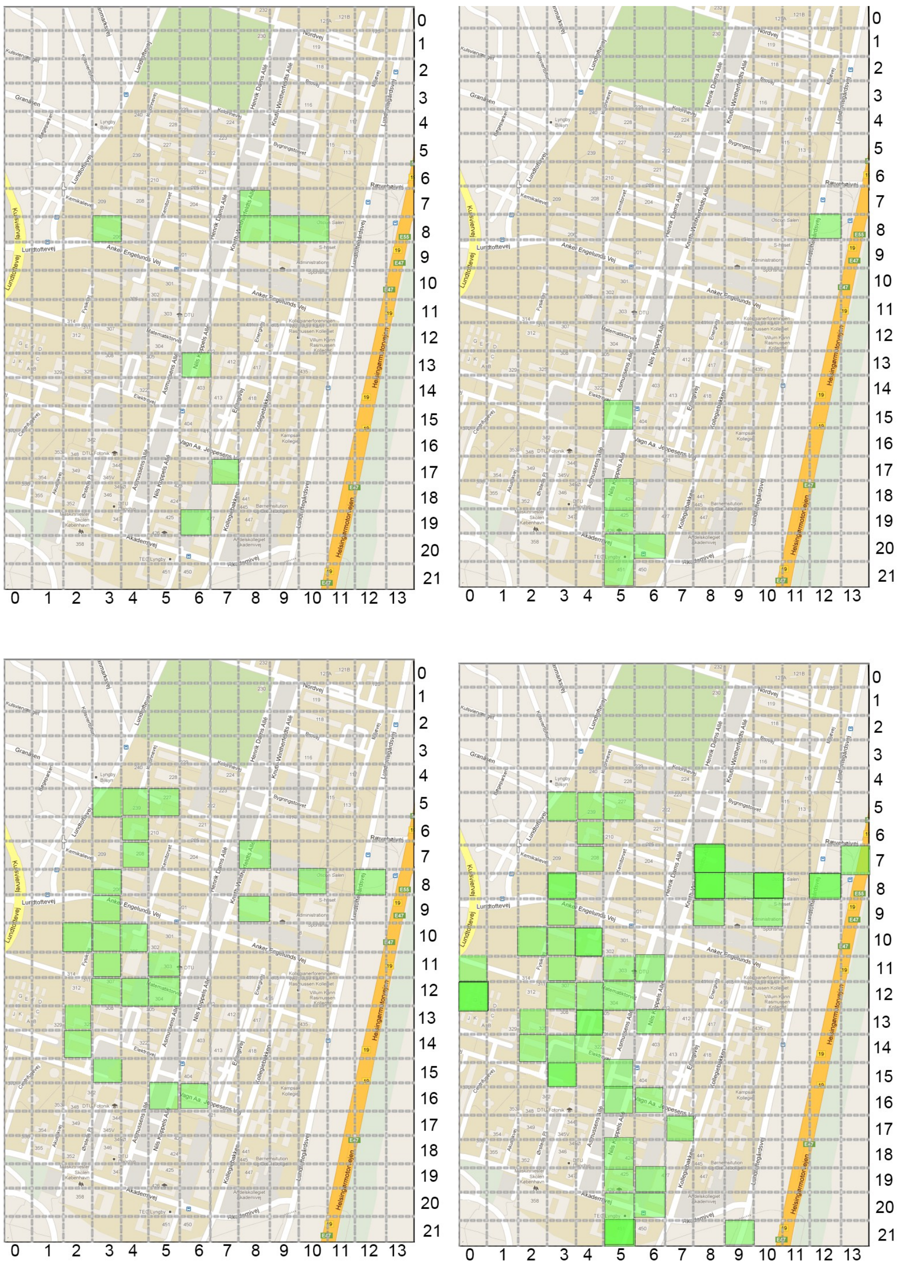


Illustration 23: Locations of all location-related requests from players representing area investigation and conquering: Top-Left: Day five, Top-Right: Day six, Bottom-Left: Day seven, Bottom-Right: Overlay of all data

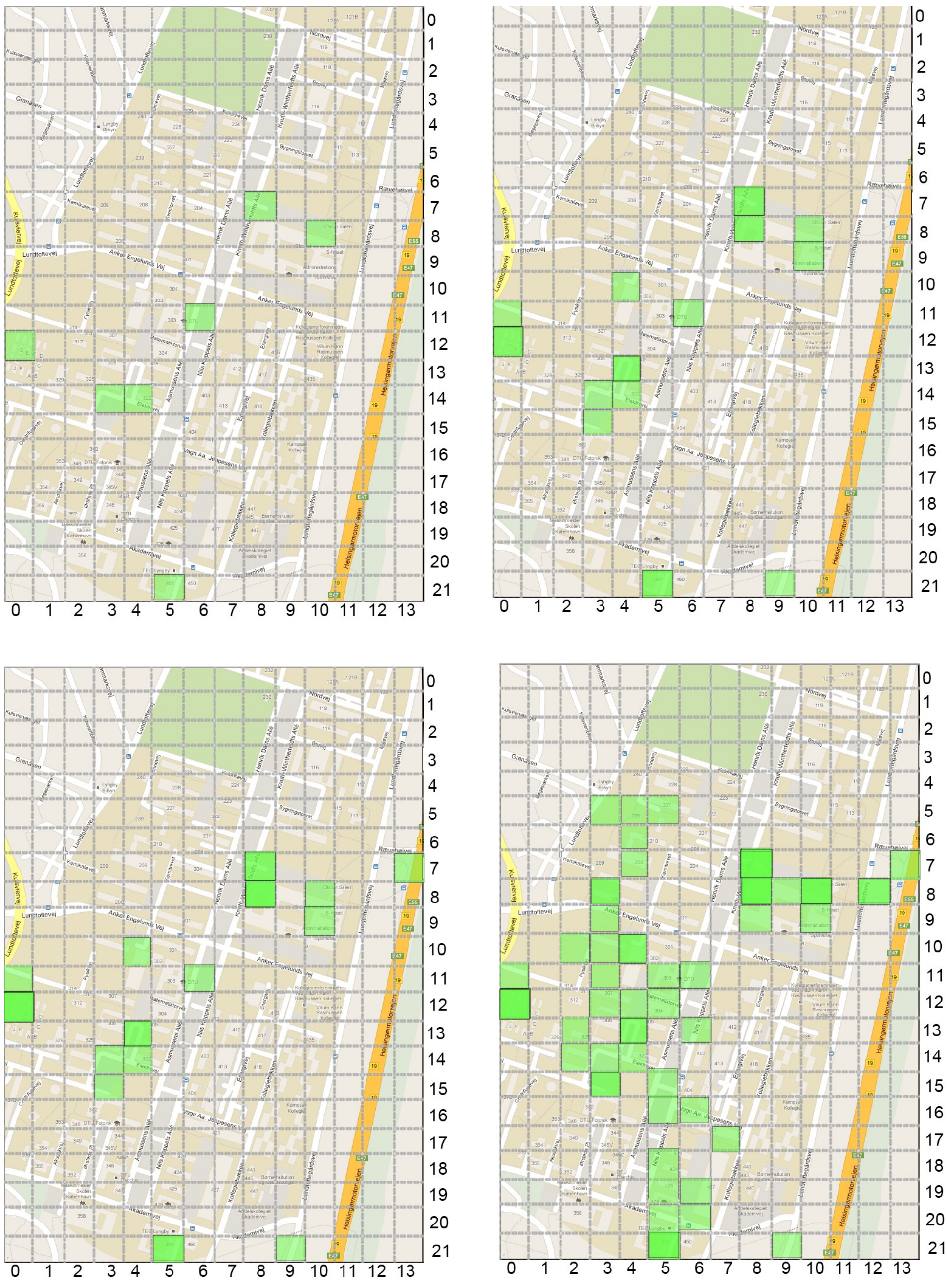


Illustration 24: Dynamics of area investigation by players

The patterns of gradual investigation and conquering of area are visible. Several "center" areas exist where players started their activity. With the flow of time players move around these areas and gradually conquer other areas around.

All location related requests were analyzed in order to construct a heat map representing amount of requests per specific area with the purpose of determining what areas are considered as "hot" areas (in the meaning, where most of game actions take place).

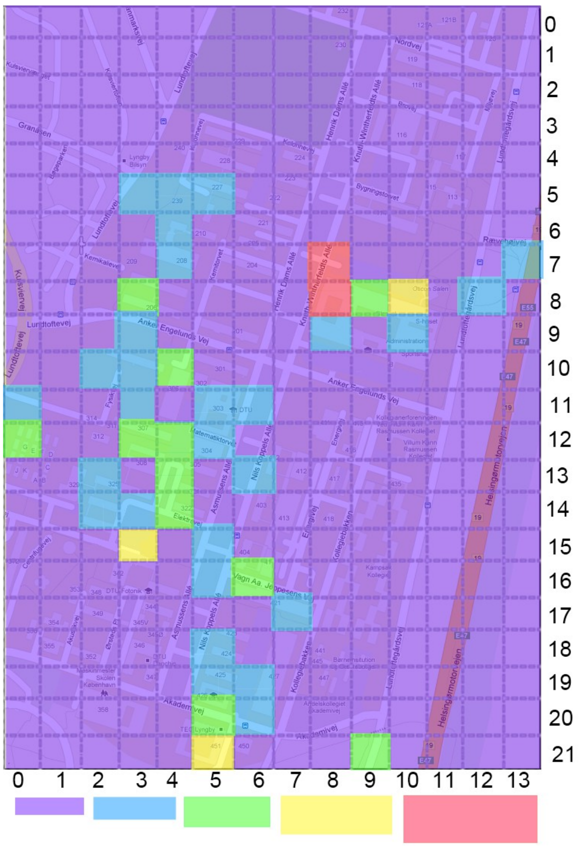


Illustration 25: Heat-map for all location related requests

All areas were divided into very cold areas (0 requests), cold ares (0 – 10 requests), warm areas (10 – 30 requests), hot areas (30 – 60 requests), very hot areas (more than 60 requests). Heat-map directly proves that most of locations being requested for are common and residential areas of

DTU campus. These areas include: library at 101 (very-hot area represented by red), main canteen in 101 (hot (yellow) area at row 8 – column 10). The "L" shaped warm area (green) represents the department of IMM buildings where students of Mobile Application Prototyping take courses.

The last snapshot of area investigation results is: 44 out of 308 areas, which is 14.28 % of all area that was reached in 7 day period.

Temporal data analysis:

Times of all user-requests was investigated in order to locate patterns of usage. DTU workday consists of two equal halves. First half of the workday starts usually from around 08:00 morning and lasts till 12:00. Second half of the workday starts at 13:00 and lasts till 17:00. These halves are separated by a lunch break from 12:00 to 13:00. These time periods were taken for analysis. In order to fill and check other parts of the day several other time periods were defined.

Full set of time periods and data represented below:

Table 21: Requests during different time-periods

Period	**Time-period (hours)**	**Length (hours)**	**Total number of requests**	**Requests per hour**
Night	00:00 – 08:00	8	35	4,37
Morning	08:00 – 12:00	4	106	26,5
Lunch	12:00 – 13:00	1	75	75
Evening	13:00 – 17:00	4	278	69,5
Late evening	17:00 – 21:00	4	370	92,5
Early night	21:00 – 00:00	3	96	32

After approximation to closest Integer a bar chart showing dependancy of Requests per hour (rph) was constructed to visualize the pattern:

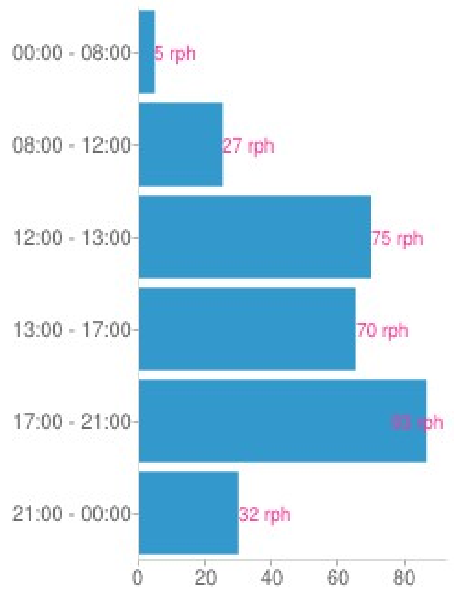


Illustration 26: Requests per hour (rph) for different parts of day

It may be seen that users tent to interact more after lunch break with peak activity during free time.

Other data and user assessments:

A strange, yet descriptive pattern of user's anticipation while using the application was established in players' behavior. At the time of the last snapshot of system-logs (27-th of February 2013) six different users that were uniquely identified by their gmail-accounts refused to enter a valid user alias and were assigned with a default alias.

Received suggestions and recommendations from eager players request additional territories (University of Copenhagen campus area and beyond Denmark), additional creatures and more options, more choices, more potential players!

Conclusions

The goals set by this thesis were achieved successfully. Context-Aware Campus Area Gaming Environment for Mobile Platforms was designed, implemented, launched and tested as a result of this project. The environment received a name “DTU GoblinsNGold” and became public.[32]

This thesis investigated pervasive games as a newly emerging type of game-play activities. The aim of this thesis was to design, develop, test, launch and test again a game-environment that would be designed for mobile platforms, be context-aware and aimed at campus area.

First, thorough analysis of theory was presented in the theoretical research section, Chapter 1. This theory was used to formulate a general concept for future game-environment in Chapter 2. Approach of looking through the Lenses of game-design by J.Schell was chosen and used in order to design the game-environment. However, differences imposed by pervasive nature of designed game-environment were discussed and taken into account, such as Pervasive Gameflow model etc. Architecture of designed game-environment consists of two main parts: Front-End that is built using Android platform and Back-End that is built and hosted on Google App Engine, as discussed in Chapter 3.

Chapter 4 investigated the results of game-environment launch, functioning and performance. Potential of idea was measured by investigating initial interest and interest retention. The data collected from Google Play and Google App Engine were carefully analyzed to support all claims made. More than 1000 log entries from back-end were carefully analyzed to investigate spatial, temporal and other patterns of players' behavior. These data together with all log and administrative information available were also investigated to understand technical aspects of designed game-environment functionality on physical layer of functionality.

32 https://play.google.com/store/apps/details?id=uadk.dtu.wdnsd.gng

Future work

One of the lenses that game-designer should use in development is Lens #93: The Lens of the Crystal Ball. This lens addresses future in general. Game-designer is asked to imagine how would future look like and what changes would future bring.

It is assumed that accuracy of sensors together with computational power and available memory for mobile devices will grow. Moreover, battery consumption is also an issue that is addressed by equipment vendors. These trends will, apparently, remove current technical limitations in a prospective future and open even more possibilities for developers to enhance services provided.

Developed game-environment can easily be used not just for campus areas. It may easily be implemented for entertainment purposes during carnivals, in amusement parks, during corporate events. With certain re-modification of game-rules it can be scaled for larger areas that do not have any boundaries.

During development of this project a pervasive game was launched by Google. The game received a name "Ingress" and was launched at 16th November 2012 without any marketing or information available before launch. Viral marketing concepts were used to advertise the game globally and, therefore, increase awareness of pervasive games worldwide. The game was tested by teams of developers for at least six months before launch of any public information. Last version of Ingress was released at 29th January, which shows that bug fixing together with improving processes are ongoing.[33] "DTU GoblinsNGold" was designed, developed and tested by author of this thesis, solely and in a shorter term. The aspect of attention to subject from such industrial giant as Google together with geographical data on installs of "DTU GoblinsNGold" proves the idea that pervasive games are getting more and more attention from both developers and players globally. It is time to play!

33 https://www.ingress.com/

Afterword

One lens from the used game-design framework was the one that inspired, fueled author with enthusiasm and led this project from scratch to a fully working, public application that is accessible worldwide. This is Lens #88: The Lens of Love. It requires designer to ask thyself, probably, the most important question: ”Do I love my project?”. Game-design is not only about science, but is also about sharing emotions. Watching players downloading, getting acquainted, inspired or disappointed, giving feedback on improvements that they would like to see. There was not a single moment when the answer of author of this thesis and author of this designed game environment was any other than “Yes, I do!”

Bibliography

JS: Jesse Schell, The Art of Game Design, A book of Lenses, 2008,Elsevier Inc,Carnegie Mellon University

NL: Nicole Lazarro, Why We Play Games: Four Keys to More Emotion Without Story, 2004,

RK: Raph Koster, A Theory of Fun for Game Design, 2004,Paraglyph Press,

ML8: Marc Leblanc, 8 Kinds of Fun, ,http://http://8kindsoffun.com/

MCZI: Mihaly Csikszentmihalyi, Flow: The Psychology of Optimal Experience, 2009,HarperCollins e-books,

GFL: P.Sweetser, P. Wyeth, GameFlow: A Model for Evaluating Player Enjoyment in Games, 2005,

JHP: Johan Huizinga, "Homo Ludens,a Study of the Play Element in culture", 1955,Beacon Press,

MPG: Markus Montola, Jaakko Stenros, Annika Waern, Pervasive Games: Theory and Design, 2009,Elsevier Inc,

LBG: Stine Ejsing-Duun, LOCATION-BASED GAMES:FROM SCREEN TO STREET, 2011,Aarhus University

RBMP: Richard Bartle, "HEARTS, CLUBS, DIAMONDS, SPADES: PLAYERS WHO SUIT MUDS", 1996,www.mud.co.uk/richard/hcds.htm

KJPG: Kalle Jegers, "Pervasive GameFlow Identifying and Exploring the Mechanisms of Player Enjoyment in Pervasive Games", 2009,Umeå University

MPN: A.Maslow, "A theory of human motivation", 1943,

RMAD: Reto Meier, "PROFESSIONAL Android™ 4 Application Development", 2012,John Wiley & Sons,Inc,

KLRM: Kin Lane, "Rise of Mobile Backend as a Service (MBaaS) API Stacks". , 2012,

AWKR: Alex Williams, "Kinvey Raises $5 Million For Mobile And Web App Backend As A Service", 2012,

DRMB: Dan Rowinski, "Mobile Backend As A Service Parse Raises $5.5 Million in Series A Funding", 2011,

Appendix

Set of Game Rules:

Campus area of DTU is now a gamefield
Total area is divided into quadrants

The aim is to Dominate the map

Player Investigates the area and Harvests Resources from different areas

4 types of Locations
6 types of Resources

Gold is either harvested or produced from other Resources
Gold is used to hire and train Creatures

Creatures are used to conquer Area
to increase Player's authority
and to prevent other Players from Harvesting Resources

Only one Creature may occupy single quadrant

Game requires Internet connectivity and Location sensors.
Within DTU this game works best with Eduroam
free Wi-Fi network connection for students.

Game-map:

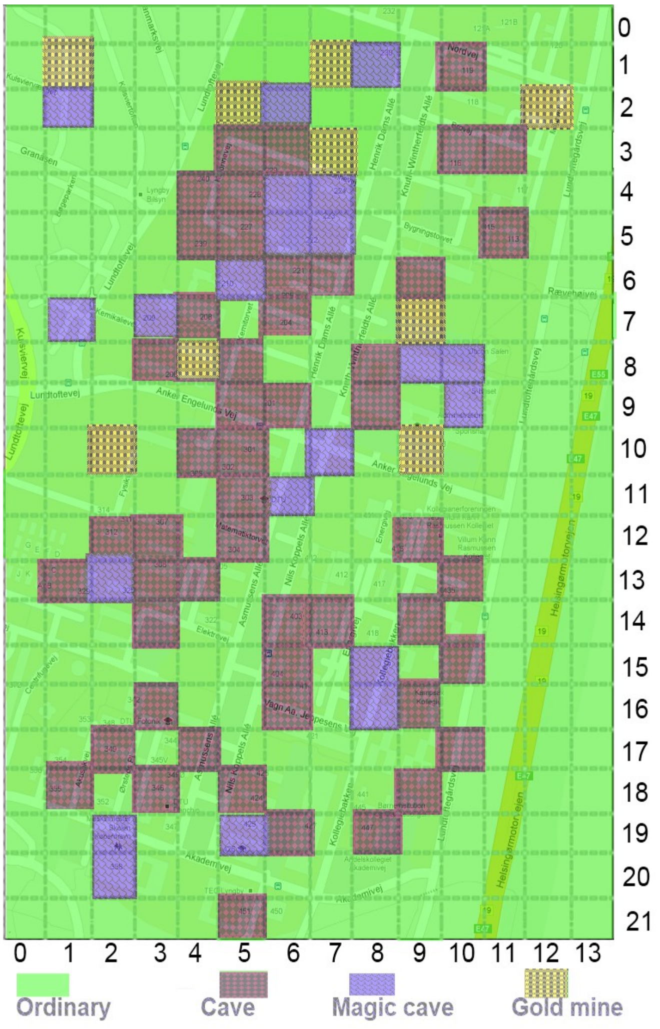
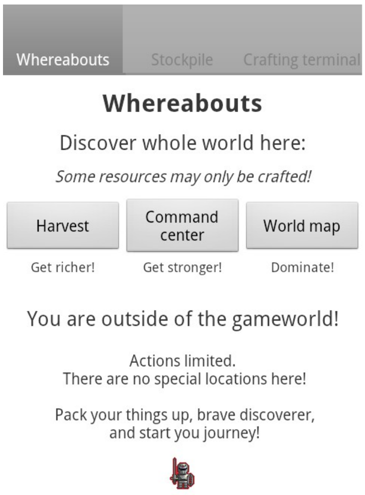


Illustration 27: Game-Map with all special locations marked

User Interface:



Whereabouts

Discover whole world here:

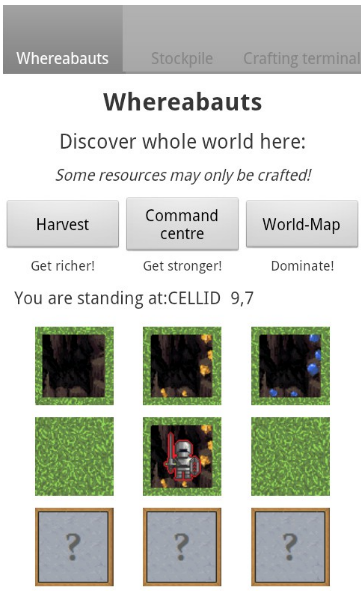
Some resources may only be crafted!

Get richer! Get stronger! Dominate!

You are outside of the gameworld!

Actions limited.
There are no special locations here!

Pack your things up, brave discoverer,
and start you journey!



Whereabauts

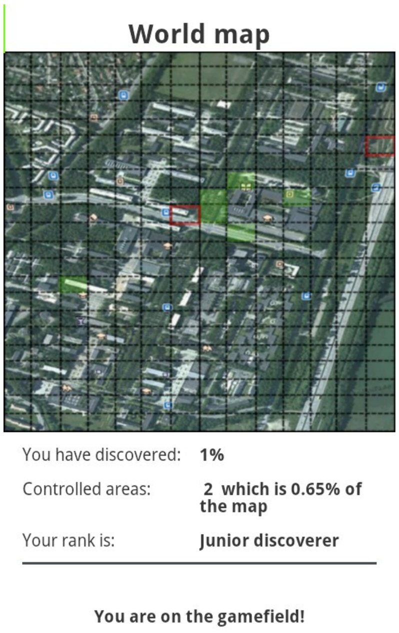
Discover whole world here:

Some resources may only be crafted!

Get richer! Get stronger! Dominate!

You are standing at:CELLID 9,7

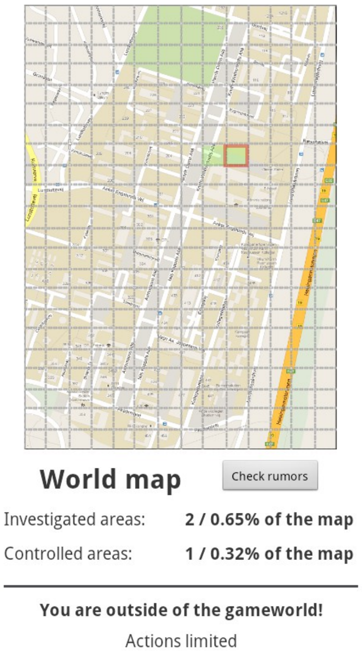
Illustration 28: Alternative for Whereabouts showing Local Map for in-game and out-game locations



World map

You have discovered:	**1%**
Controlled areas:	**2 which is 0.65% of the map**
Your rank is:	**Junior discoverer**

You are on the gamefield!



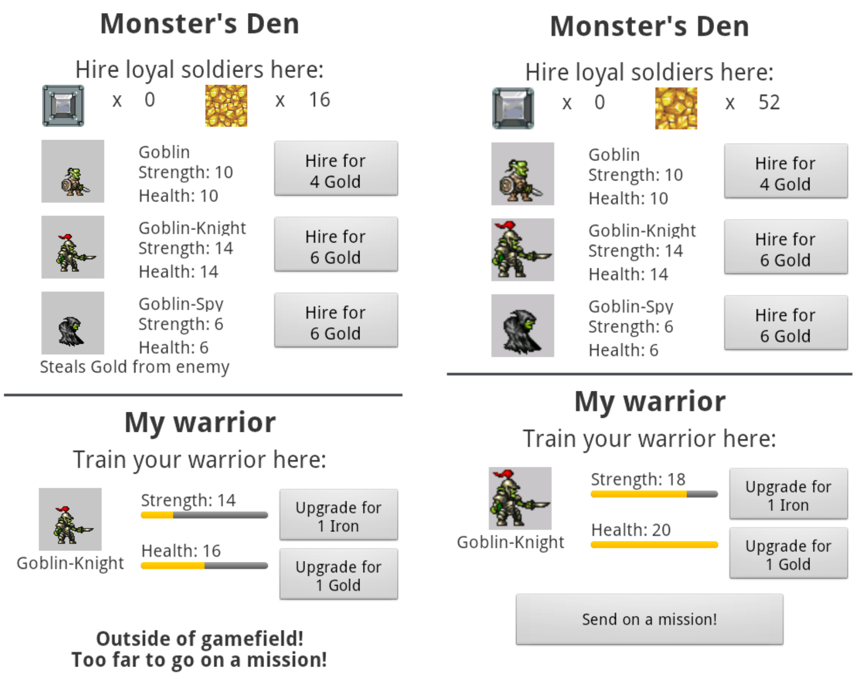
World map

Investigated areas:	**2 / 0.65% of the map**
Controlled areas:	**1 / 0.32% of the map**

You are outside of the gameworld!

Actions limited

Illustration 29: Alternatives for WorldMap (old with satellite view) and new improved



Monster's Den

Hire loyal soldiers here:

x 0 x 16

Goblin Strength: 10 Health: 10	Hire for 4 Gold
Goblin-Knight Strength: 14 Health: 14	Hire for 6 Gold
Goblin-Spy Strength: 6 Health: 6	Hire for 6 Gold

Steals Gold from enemy

My warrior

Train your warrior here:

Goblin-Knight

Strength: 14 — Upgrade for 1 Iron

Health: 16 — Upgrade for 1 Gold

Outside of gamefield!
Too far to go on a mission!

Monster's Den

Hire loyal soldiers here:

x 0 x 52

Goblin Strength: 10 Health: 10	Hire for 4 Gold
Goblin-Knight Strength: 14 Health: 14	Hire for 6 Gold
Goblin-Spy Strength: 6 Health: 6	Hire for 6 Gold

My warrior

Train your warrior here:

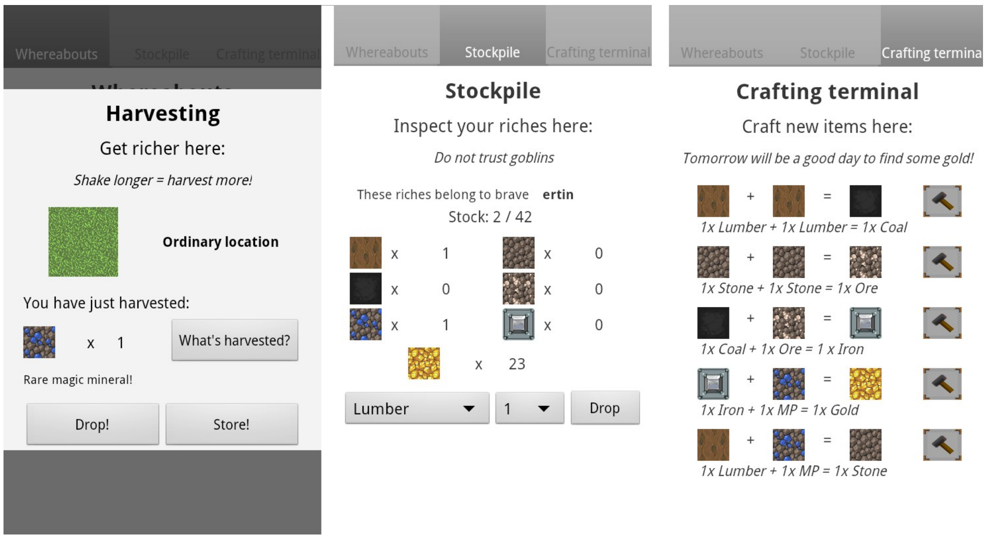
Goblin-Knight

Strength: 18 — Upgrade for 1 Iron

Health: 20 — Upgrade for 1 Gold

Send on a mission!

Illustration 30: Alternative layouts of Command Center for ingame and outgame locations (for different screen quality)



Harvesting

Get richer here:

Shake longer = harvest more!

Ordinary location

You have just harvested:

x 1 — What's harvested?

Rare magic mineral!

Drop! Store!

Stockpile

Inspect your riches here:

Do not trust goblins

These riches belong to brave **ertin**

Stock: 2 / 42

x	1	x	0
x	0	x	0
x	1	x	0

x 23

Lumber ▾ 1 ▾ Drop

Crafting terminal

Craft new items here:

Tomorrow will be a good day to find some gold!

1x Lumber + 1x Lumber = 1x Coal

1x Stone + 1x Stone = 1x Ore

1x Coal + 1x Ore = 1 x Iron

1x Iron + 1x MP = 1x Gold

1x Lumber + 1x MP = 1x Stone

Illustration 31: Resource related UI: harvesting, Storing and Crafting