NTNU
Norwegian University of
Science and Technology

The Project Manager Game

A learning game for project management

Eivind Toresen Askestad
Vegard Aas Knutsen

Master of Science in Computer Science
Submission date: June 2016
Supervisor: Alf Inge Wang, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

The Project Manager Game

A learning game for project management

Eivind Askestad

Vegard Aas Knutsen

June 17, 2016

Department of Computer and Information Science
at the Norwegian University of Science and Technology

Supervisor: Alf Inge Wang

Problem Description

The goal of this project is to develop a game where you learn the skills of being a project manager in a software development project. The player will play the role of a project manager and his or her aim is to try to ensure that the project succeeds and do not fail. The goal of the project is first to develop an architecture and models to support the creation of different scenarios that can happen in a project manager life. Further, a prototype must be implemented, then conduct experiments with users to see the effect of the game. The game should be developed on the HTML5/Javascript platform and standardized server technologies.

Assignment given: January 10th, 2016

Supervisor: Alf Inge Wang

Abstract

The goal of this study was to design and develop an enjoyable strategy/simulation game that focuses on teaching project management, and evaluates how this game affects the players. Through a literature study, we found that previous research on the topic of project management learning games lacked a clear focus on the enjoyment that is needed to create motivation. We also found that there was a need for a mean that could combine enjoyment and reflection without interrupting the flow of the game.

The game was developed with an iterative process, while focusing on creating game mechanics that could lead to interesting challenges, and encourage the player to use strategy. In addition, we used theory from the literature study to guide the development. To ensure that the game could be enjoyable, we included playtesting as a part of the process. The main choice of technology includes HTML5, Pixi.js, and TypeScript.

As a result of this process, we created a single player game called "Project Manager", where the player take the role of a project manager in a software company. The main activity in the game is to assign employees to work on tasks, improve their skills, and attend to their mood. Learning in the game is presented through challenges that the player has to deal with, and textual descriptions at different places in the game.

The game was evaluated on 18 participants through a questionnaire, interviews, and observations. This evaluation showed that the game was considered enjoyable and that the participants seemed to use different strategies when playing the game. It also showed that there was a lack of interest in reading the textual descriptions. However, the participants seemed to able to remember how they played the game in detail, meaning what kind of strategy they had used. This suggests that strategy has a potential for effective learning. Furthermore, through testing and developing the game, we found Pixi.js and TypeScript can be suitable options when creating a web game.

Sammendrag

Målet med dette master prosjektet var å designe og utvikle en morsomt strategi/simulator spill som fokuserer på å lære prosjektledelse, og evaluere hvordan dette spillet påvirker spillerne. Gjennom en litteraturstudie fant vi ut at tidligere forskning på dette temaet, ofte manglet et klart fokus på spillerglede som er nødvendig for å skape motivasjon. Vi fant også at det var behov for en måte å kombinere spillerglede og refleksjon uten å avbryte flyten i spillet.

Spillet ble utviklet med en iterativ prosess, med fokus på å skape spillmekanikker som kunne føre til morsomme utfordringer, og oppfordrer spilleren til å bruke strategi. I tillegg ble det brukt teori fra litteraturstudien som en guide under utviklingen. For å sikre at spillet er morsomt, utførte vi spillertesting som en del av prosessen. Det viktigste valget av teknologi inkluderer HTML5, Pixi.js og TypeScript.

Som et resultat av denne prosessen har vi laget et spill som heter "Project Manager", hvor spilleren tar rollen som prosjektleder i et programvareselskap . Hovedaktiviteten i spillet er å tildele ansatte arbeidsoppgaver, utvikle deres ferdigheter, og passe på deres humør. Læring i spillet blir presentert gjennom utfordringer som spilleren må forholde seg til, og gjennom tekstlige beskrivelser presentert forskjellige steder i spillet.

Spillet ble evaluert på 18 deltakerne gjennom et spørreskjema, intervjuer og observasjoner. Evalueringen viste at spillet var underholdende, og at deltakerne syntes det å bruke ulike strategier var morsomt. Evalueringen viste også at det var mangel på interesse for å lese de tekstlige beskrivelsene i spillet. Derimot, kunne deltakerne huske hvordan de spilte spillet i detalj, noe som innebærer hva slags strategi de hadde brukt. Dette tyder på at strategien har et potensial for effektiv læring. Videre gjennom testing og utvikling av spillet, fant vi ut at Pixi.js og TypeScript kan være gode alternativer når man skal lage nettspill.

Preface

This report presents the work done for our master's thesis at the Department of Computer and Information Science at the Norwegian University of Science and Technology. The work was conducted from January to June 2016. We would like to thank the students who participated in the evaluation of the game, and give special thanks to our supervisor Alf Inge Wang for guidance and motivation throughout this project.

Trondheim, June 17, 2016

Eivind Askestad Vegard Aas Knutsen

Contents

List of Figures

List of Tables

Part I

Introduction and Method

This part will first present the problem that we have chosen to focus on. Second, we will outline the structure of this report. Next, we will turn our attention to the method used in this project, and define a set of research questions.

Chapter 1

Introduction

This study is a continuation of the prestudy Askestad and Knutsen (2016) that was conducted from fall 2015 to January 2016. A literature study was conducted as part of that study, and some ideas were generated for a game design. In this chapter, we will present our motivation, and outline the structure of the report.

1.1 Motivation

Digital games are a well-established arena for learning. Learning through games can provide an alternate approach that is fun and engaging. Good games have the ability to motivate players to concentrate for hours at a time, day after day (Prensky, 2003). This motivation is an indispensable part of the gaming industry. The process of learning is also dependent on motivation. Combining the motivating elements of commercial games and educational content is, therefore, a reasonable thought (Prensky, 2003) . But this merge has many complications. A problem is that some attempts have resulted in "boring" games not creating enough motivation, or fun games that do not create the desired educational outcome Van Eck (2006) .

Good computer games often use many proven educational methods (Gee, 2003) . One of this method is conveying information when it is useful in the context where it is used. Gee (2003) compares this to education in classrooms, where the material is often presented then used in different contexts separated by time. He then argues that this makes it harder for people to understand the information being presented. Project management in a software environment is an area where games could be used for learning purposes. Bakken (2013) a former student at Norwegian University of Science and Technology, wrote a master's thesis about the subject during the spring of 2013. He created a game where the player is the project manager in a software project environment. In this study, he investigated educational strategies. Previous studies on the subject of simulating project management in software development is conducted by Navarro (2006); Dantas et al. (2004). Although these studies present good result, one could argue that they lack many of the enjoyable elements often found in commercial entertaining games. This is something we want to explore in this project.

1.2 Report Structure

In this section, we will present the structure of this report.

Part I Introduction and Method

This part will describe the research method, research strategy and present the motivation for this study.

Chapter 1 Introduction - This chapter presents the motivation for doing this research and presents the structure of the report.

Chapter 2 Research Method, Questions and Strategy - In this chapter, we will define the goal and research question for this study, and explain the strategy and the methods that will be used to provide answers to these questions.

Part II Prestudy

In this part, we will present a literature study on topics that are relevant to the goal of this study.

Chapter 3 Technology for Web Games In this chapter, we will present the technology that is available for creating games for the web browser.

Chapter 4 Project Management This chapter, will briefly provide a background to project management.

Chapter 5 Enjoyment and Learning This chapter, will provide research related to enjoyment and present some guidelines on how to create an enjoyable learning experience.

Chapter 6 Game Concepts This chapter, will present theories for designing games, with a focus on game mechanics.

Chapter 7 State of the Art In this chapter, we will present some previous research and game descriptions that are relevant for this study.

Part III Design and Development

In this part, we will present the process of creating the game for this study, a description of the game, the choice of technology and a description of the system,

Chapter 8 Prototype Development This chapter, will describe the design process for creating the game.

Chapter 9 Game Description This chapter will present the game that has been created for this study.

Chapter 10 Technology This chapter will present the technology used when creating the game in this study.

Chapter 11 System Development In this chapter, we will briefly present the architecture and implantation of the game.

Part IV Evaluation

This part presents an evaluation of the game and a discussion of the results.

Chapter 12 Experiment In this chapter, we will present the experiment and the result from a questionnaire, interviews, and observation.

Chapter 13 Discussion In this chapter, we will discuss the results of the experiments.

Part V Summary

In this part, we will present a conclusion and further work.

Chapter 14 Conclusion In this chapter, we will provide answers to the research questions.

Chapter 15 Further Work In this chapter, we will present our recommendations for further research based on our findings.

Chapter 2

Research Method, Questions and Strategy

This chapter aims to explain the methods used during the different stages of this study. First, the "Goal Question Metric" is described, which is used to define goals, research questions and to find means evaluate the result. Second, the chosen research strategy will be presented. Finally, the methods for generating and analysing data are described.

2.1 Goal Question Metric

The Goal Question Metric (GQM) was originally a paradigm created to define and evaluate operational and measurable software goals using quantitative data analysis (Basili, 1992). The goals define the conceptual level and describe what is evaluated. The goals are the basis for the creation of questions (operational level). These questions are formulated to show how the goals will be achieved. Each question is then evaluated by using a metric (quantitative level). The metric should be evaluated quantitatively, answer the questions and be measurable (Basili, 1992).

In this study, GQM acts as an organized guide to define goals, research questions and find means to evaluate these research questions. Because this research explores the impact of a learning game on subjects and not just creation of a system, there is a need for data that can be analysed qualitative and quantitative (Oates, 2005). Qualitative data analysis can help understand what the subjects are actually doing and why (Oates, 2005). This is why this study deviates from GQM which advice to only use measurable research questions that is dependent on quantitative data.

2.2 Research Questions

Using the GQM method as a guide, we created the following goal:

> **Goal -** The goal of this study is to design and develop an enjoyable strategy/simulation game that focuses on teaching project management and evaluate how this game affects the players.

This goal was then decomposed to the following research question:

> **RQ1 -** How does the game affect the player's enjoyment?

This question aims to answer if the player enjoys playing the game and analyse if the chosen approach has the wanted effect on the user. A precondition to this research question is finding and implementing enjoyable elements into the game. To answer this question we will use a questionnaire, observations, interviews and a literature study.

> **RQ2 -** How does the game affect the players learning experience?

This question seeks to understand how the players experience the learning provided in the game. It is also meant to investigate the learning potential in strategy based games. To achieve this we need to create/find a learning process that uses strategy as a mean to create learning. Background information will be found through the literature study. Then we will evaluate the game prototype through the use of a questionnaire, observations, and interviews.

> **RQ3 -** What types of game mechanics can be used to encourage the player to use strategy ?

This question concerns how the game can be design in terms of game mechanics, in order to encourage the player to plan his actions, and use strategy for dealing with the challenges provided by the game. To answer this question, we will need a way to define game mechanics, and experiment to see what kinds of effects these produce. We will also be using a questioner, observation and interviews to see if the players use strategies while playing the game.

> **RQ4 -** What are the technology options for creating web games ?

This question seeks to identify the technology that is available for creating web browser games but considers only those that are most relevant for being used and will therefore not include plugins that can be installed in the browser. The answer to this question will mainly be provided by performing a literature study, but we will also elaborate on the experience gained from the development for this study.

2.3 Research Strategy

Oates (2005) defines a research strategy as the overall approach to answering the research question. For this study, we will use Design and Creation as the main research strategy. In addition, we will also use literature study to get familiar with some of the theories we will use while creating the game, and evaluating the game.

2.3.1 Literature Study

A literature study was conducted as part of the prestudy (Askestad and Knutsen, 2016), which will act as a basis for this study. In the literature study for this report, we will collect the most important theory from the prestudy while expanding one some of the topics. The literature study will presented in Part II Prestudy. When conducting the literature review, peer review academic articles will be prioritized. However, some of the subjects may often be better presented by books. Some theory on game concepts will, therefore, be based on new and frequently cited books. Reviewing literature on technology will also require research outside of articles, since browser games are a fairly new topic, and includes technology that changes on a yearly basis.

2.3.2 Design and Creation

Oates (2005) describes *Design and Creation* as a research strategy which focuses on developing an IT product. The product can either be a result in itself or be used to generate new knowledge. As the goal of this study is to create a game, we will use Design and Creation as our main strategy, with the purpose of creating new knowledge by evaluating the product. The Design and Creation phase will be described in Part III Design and Development. Furthermore, Oates

(2005) describe that central to this strategy is the chosen development methodology. One such methodology is prototyping, which involves iterating over the product, and in each iteration improve the implementation based on what was learned. The benefits of using such an approach are that one does not need to understand the problem before, instead, one has the opportunity to explore different solutions. Oates (2005) also points out that when doing design and creation in research, one should focus on exploring what is not known before. In contrast, normal design and creation will be successful if everything went according to the plan.

Using prototyping as the main development methodology is essential in game development, as the requirements such as enjoyment, is generally known to be difficult to predict and may require some experimentation. Prototyping can also be used as a part of the game design process. The literature on game design often underlines how important it is to create prototypes when creating games, while it is often not necessary to define a strict process (Salen and Zimmerman, 2004). As games often have other requirements than software, that are difficult to predict, for example, enjoyment.

2.4 Data Generation

This section describes the different data generation methods used in this study and explains how these are analysed. Oates (2005) stress that the product that has been created as a part of Design and Creation needs to be evaluated, and includes questionnaire, observations, interviews and documents as methods that are often used.

2.4.1 Questionnaire

A Questionnaire will be the main the main data generation method in this study. The aim of the questionnaire is to produce data for a quantitative analysis, to answer the research questions. The questionnaire will be *method triangulated* by the observations and interviews. Method triangulation is described by Oates (2005), as a way to look at a phenomenon in more than one way, creating more data and potentially improve the quality of the research. In this study, results from all the data generation methods will be compared to look for similarities and differences.

The questionnaire will be answered by all the subjects playing the game, and will mostly consist of predefined multiple choice questions. To remove the influence given by the presence of the researchers, the questionnaire is self-administered and anonymous (Oates, 2005). To make the questionnaire more accessible to the respondents, it will be conducted online. The plan is to send an e-mail to the participants that will include a link to the game and a link to the questionnaire. In the e-mail, we will encourage the participants to answer the questionnaire shortly after playing the game.

2.4.2 Observations

Overt participant observation with the researchers as complete observers is chosen as the method for observation. Participant observation is often used when the researchers want to find out what people do, why they do it, and the meaning they assign to the activity (Oates, 2005). This is why participant observation is chosen as a data generation method. The observations act as a mean to explore what participant are actually doing, discover unpredicted user behavior, and triangulate the data obtained with the questionnaire result. Structured observations are not chosen because it uses a predefined system of observation, and is not usually used to explore for the unknown (Oates, 2005). It was found that actively playing the game together with the participant added little value. To gather as many data as possible, the researchers will be complete observers. Overt observation is the most obvious choice because of ethical and practical reasons. After the observation is completed the participants will be briefed. The researcher will then ask if our interpretations and observations are correct. The data gathered will than be analysed qualitatively.

2.4.3 Interviews

The interviews will be conducted in a semi-structured manner. Allowing the interviewees to speak about the subjects in more detail and introduce issues of their own (Oates, 2005). A list of questions will be created before the interviews. These questions are always asked in the same way, to ensure that the main questions are answered. Follow up questions is allowed to get a deeper understanding of the interviewee's thoughts (Oates, 2005). All the interviews will be recorded, to ensure that nothing is forgotten. The interviewee has to agree to be recorded. The interviews will be conducted within an hour after the game is played and the questionnaire is answered. An hour is chosen because it is preferable that the subjects remembered the game and if they needed a break they have time to summon. All the interviews will be transcribed after completion and analysed qualitatively.

Summary

In this chapter, we began by defining four research questions using the GQM method. Further, we outlined Design and Creation as the main research strategy for this study. To evaluate the game that is going to be created, we include questioner, observation, interviews.

Part II

Prestudy

In this part, we will present the background information that is related to the goal of this study. This information is described in five chapters. Note that this part is heavily based on the prestudy Askestad and Knutsen (2016).

Chapter 3

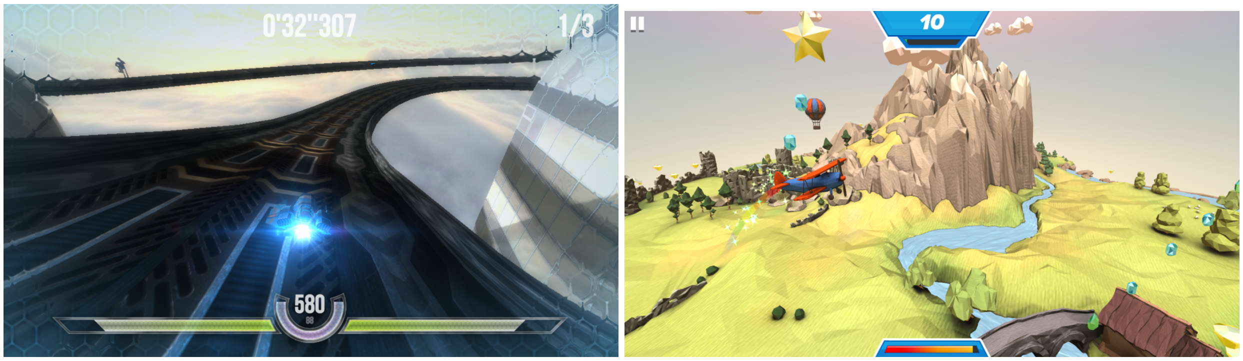
Technology for Web Games

Web games have the advantage of being available for the player without the need for any installation. In this chapter, we will look into technologies that are available for creating web games. Technologies that use plugins will not be included since these receives limited supported by web browsers. The findings from this chapter will contribute to answering *RQ4: What are the technology options for creating web games ?* This chapter is based on Chapter 6 "Technology" from the prestudy Askestad and Knutsen (2016).

3.1 HTML5 Technologies

Web browsers are today capable of displaying advanced 3D graphics without the need of plugins or additional software. HTML5 technologies are what makes this possible, and while a web browser has varying support for these technologies, improvements are made frequently and there exists frameworks to provide fallback mechanisms. Figure 3.1 displays two games that illustrate the graphic capabilities of a modern web game. For some games, it might be sufficient to use technologies like HTML, CSS and JavaScript. However, these technologies do not reach the performance requirements of many games. Instead, one could use some of the technologies provided by HTML5. Curran and George (2012) point out the most important HTML5 technologies for creating games:

Canvas The canvas is an HTML element that can be used to draw graphics in a defined region on a web page and is today supported by all major browsers. To use the canvas, a reference



(a) Hexgl

(b) Swoop

Figure 3.1: Examples of Two Web Browser Games

has to be established using JavaScript. It is then possible to get the context object which contains methods for drawing on the canvas. The 2D Context is one of the several possible contexts that exists, and it acts as a way to draw 2D graphics. It includes methods for drawing shapes, text, and images.

WebGL Another new feature for graphic rendering is WebGL, which is a JavaScript API that can be used for hardware accelerated graphics, which makes it possible to create more graphics intensive games. By using specialized hardware functionality, developers can make games that include visual effects that are not present in web games today without the use of plugins. The current version of WebGL is based OpenGL ES 2.0, which is the version of OpenGL that is currently used on most mobile platforms. This version of OpenGL does not support all of the features that the modern GPU can provide, but great performance gains can be still be achieved.

HTML5 Audio While the canvas element makes it possible to include real-time graphics into the browser, the HTML5 audio element provides a way to play sound without plugins. To use the audio element, a source has to be specified. The source refers to a specific audio format, which again needs to be supported by the browser. However, most browsers today seems to have support for mp3. The audio element has some additional attributes that allow the audio be controlled in different ways, which include operations like repeating the audio, play, stop and seek. While the audio element does provide an important feature that is needed to create web-based games, developers often want more control over audio.

Web Adio API The Web Audio API is meant to extend the purpose of the audio element by introducing a way to process and synthesize audio. It is currently supported by most major browsers.

3.2 Libraries

Using a framework could be beneficial in many ways, especially because they often provide fallback mechanisms where features are not supported by the browser, but more importantly, it speeds up development. There exists a wide range of these JavaScript Frameworks for creating web browser games with HTML5 Technologies, where some are comprehensive and act as full game engine and others are rather tools for the developer. Several books have been written about HTML5 game development, which tends to use different frameworks based on the preferences of the authors. For the purpose of this study, we have identified three different frameworks that are free and focus on 2D game development, which also provides decent documentation.

Pixi.js A small library that focuses on rendering, and encourages additional functionality to be provided by other specialized libraries (PixiJS, 2016). Among its features are multiplatform development, with the possibility of multitouch, which is important for mobile development. Furthermore, it includes simple ways of adding effects such as blurring and tinting, and offers an API that similar to ActionScript. Pixi.js seems to be a popular choice for developers that want a small simple library to work with.

Phaser Is a more comprehensive library than Pixi.js. It provides a wide range of features, including those found in Pixi.js (Phaser, 2016). These features include a physics engine, animation support, preloading, and sound. Phaser seems to provide most of the features one would need in a single library, and act more as a full game engine.

CreateJS Is a collection of multiple libraries that each provides one type of functionality (CreateJS, 2016). One of these libraries is EaselJS which makes it easier for the developer to use the Canvas. Another is TweenJS, which provides an easy way to do animation. CreateJS does also include a library for using sound, called SoundJS. At last, PreloadJS lets the developer load assets efficiently. CreateJs provides access to a lot of functionality and lets the developer include only what is needed.

Lastly, one thing to note is that it is possible to create HTML5 games using other technologies as well. Here are two other technologies that are in the process of transitioning to HTML5 Technologies, and can therefore, be used to create games for the browser without using plugins.

Unity Is a multiplatform game engine for creating both 2D and 3D games (Unity, 2016). It offers a powerful development environment that lets developers program using several languages such as JavaScript, C# and Boo. Unity also has a large community and a store that sells assets that can be used in games. Installing the Unity Web Player allows games to be played in the browser. Although, the recent version provides support for WebGL, which makes it possible to run Unity games in some browsers without the use of plugins.

Flash Adobe Flash has for a long time been the go-to technology when making browser games but has experienced a decline in usage as many browsers and especially mobile browser has stopped supporting it (Adobe, 2015). However, Adobe Flash offers a powerful and unique environment that includes a simple way to create graphics and animations, and program different behavior. Adobe (2015) state that it been decided that Flash changes its name to Adobe Animate, with the goal providing support for HTML5 canvas and WebGL. This decision is certainly interesting for the future of game development for the web .

3.3 JavaScript

HTML5 Technologies rely on JavaScript as the main scripting language. There is a general notion that JavaScript can be confusing for new developers. McAnlis et al. (2014) states that "JavaScript has a number of gotchas that trip up developers coming from other languages" (p. 2). It is pointed out that much of the reason for this is that, while JavaScript is an object-oriented programming language, it uses a different model than traditional OOP, namely prototypal inheritance (p. 10). In addition, JavaScript is under constant development, and features are added to the language specification frequently, were the most recent are called ECMAScript 6. However, the features need to be supported by web browsers. The JavaScript community has created some tools to deal with this issue. These tools let the developer use some of the new features, and will then compile this into code that can run on all web browsers. Two example of such tools are:

TypeScript Lets the developer use strong types and class-based object-oriented programming and in addition provides many of the features from ECMAScript 6 (TypeScript, 2016).

Babel Focus only on providing ECMAScript 6 features to the developer, and does not introduce new concepts like TypeScript does (Babel, 2016).

When developing HTML5 games, one could benefit from using either TypeScript or Babel. However, considerations should be done because the method has to be compiled to JavaScript, thus creating more dependencies and a more complicated workflow.

Summary

In this chapter, we briefly described the potential of browser games. The technologies that make this possible includes Canvas, WebGL, HTML5Audio and WebAudioAPI. Several frameworks for creating games with these technologies exists, and we identify Pixi.js, Phaser and CreateJs as some popular choices. Using HTML5 technologies means developing with JavaScript, which comes with a set of challenges, one of them is that the language under development. To use new language features, one could use a compiler such as TypeScript or Babel. In this study, the technology described will be used to answer - RQ4: What are the technology options for creating web games ? In addition, researching different technologies gives some insight on what technologies should be chosen when developing the game for this study.

Chapter 4

Project Management

This chapter presents an overview of project management. The will guide the overarching concepts used in the simulation of the game. This chapter is based on Chapter 7 "Project management" from the prestudy Askestad and Knutsen (2016).

Kerzner (2013) defines a typical project as a number of activities and tasks that, has a defined start and end, objectives that should be completed within the given specifications, often having a limited amount of resource that can be spent, and that uses humane and non-human resources and fulfills several functions. Kerzner (2013) then describes that the purpose of a project is to create deliverables. In software engineering, the deliverables are typically products, prototypes, reports or services. In a project, the goal of a project manager is to complete an activity within the given constraints (Kerzner, 2013). The constraints traditionally encompass time, cost and performance. Project management is meant as a way to make as good use as possible of the resources. To do this the one needs to do trade-offs in relation to constraints and the resources. An example would be time and quality where the amount of time spent influences how good the quality is and the quality influences how much time is needed.

Kerzner (2013) defines, project initiation, project planning, project execution, project monitoring and control, and project closure as the main processes involved in project management. We chose to focus on the project execution and project monitoring. During the project execution, a project manager should control, give directions and be in charge of the work. He or she should also help the team members improve (Kerzner, 2013). When monitoring and controlling a project the project manager should follow the course of the progress. he or she should also

“Comparing actual outcome to predicted outcome” (Kerzner, 2013) and “Analyse variances and impacts" (Kerzner, 2013) . If something unpredicted happens or the project needs to change in some way, the project manager should do the appropriate adjustments.

Summary

When designing the game, the definition of a project, the project manager’s role, goals, constraints, and purpose creates a set of rules that the game must follow. The two managerial processes, project execution and project monitoring describe the setting in the game and the project managers role in this setting.

Chapter 5

Enjoyment and Learning

This study focuses on creating an enjoyable learning game. It is therefore, essential learn more about formal ways of understanding enjoyment and learning. This chapter presents the GameFlow model, which describes a way to reason about enjoyment. Furthermore, some theories about creating motivating learning games are presented. This chapter is based on Chapter 4 "Enjoyment and Learning" from the prestudy Askestad and Knutsen (2016).

5.1 GameFlow: A Model for Enjoyment

The purpose of this section is to describe the GameFlow model, which is used as guide and building enjoyment into the game for thus study. First, it presents a short description of flow and the GameFlow model, then, it presents all the GameFlow elements.

An established way to describe enjoyment is with the concept of flow (Csikszentmihalyi, 1990) . It is defined as "a feeling so gratifying that people are willing to do it for its own sake, with little concern for what they will get out of it, even when it is difficult or dangerous" (Csikszentmihalyi, 1990). Csikszentmihalyi (1990)'s research showed that people from all over the world described the experience of when their life was at its fullest, the same way. This optimal experience is called Flow.

The GameFlow model combines heuristics from a variety of articles that describes enjoyment in games and uses the characteristics of the flow experience to bind these heuristics together (Sweetser and Wyeth, 2005) . The model consists of eight elements and is created to evaluate and designing enjoyment into games. The eight elements are concentration, challenge, player skills, control, clear Goals, feedback, immersion and social interaction. These elements will be described below, except for social interaction. Social interaction is not a part of this study and is therefore not included. It is also the only GameFlow element that is not directly linked to flow. Player skill and challenge is combined and describe together because they overlap on some points.

Challenge

Sweetser and Wyeth (2005) claim that challenge is the most important thing to consider in good game design. Sweetser and Wyeth state that challenge in games should match the player's skill level, and increase as the player improves. It is emphasized that the players skills and the challenge associated with the activity should be matched and that they both should exceed a certain threshold. This is strongly related to the experience of flow (Johnson and Wiles, 2003; Sharafi et al., 2006). The player may be bored if the skill level is greater than the challenge. However, if the challenge is to demanding, it could result in anxiety. The challenge should result in the ideal experience for the player, see Figure 5.1. This balance should be kept throughout the game, and new challenges should emerge at an appropriate pace. Since players got different level of skill, games should also have different levels of challenge.

Skill and challenge are two closely related elements and are often described together. Often, skills are the tool needed to overcome a challenge. To experience flow, they need to match each other and exceed a certain threshold (Sweetser and Wyeth, 2005). "Player skill" is included as one of the core elements in the GameFlow model. In this context, it is referred to as how the game supports its development and mastery. Learning the game is in itself a skill that the player needs to perceive. Sweetser and Wyeth suggest that this process should be a fun “in game” experience, and feel like actually playing the game. It should also increase throughout the progress in the game. When a skill is mastered, the player should be rewarded appropriately and the challenge adjusted.

Goals

Csikszentmihalyi (1990) states that to a achieve flow the goals needs to be clear. Sweetser and

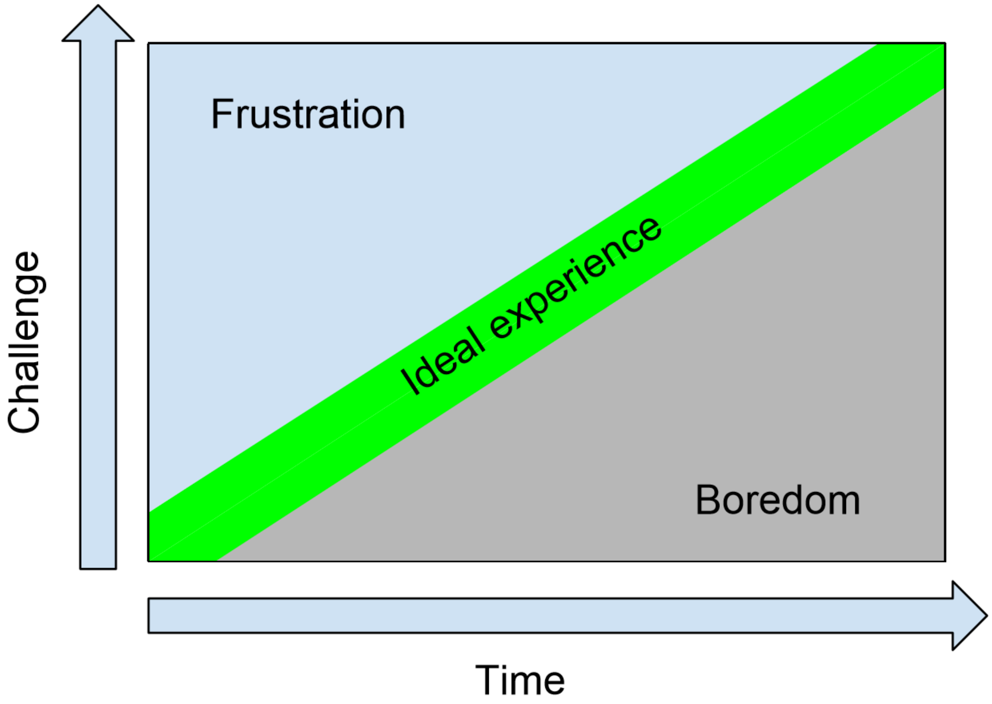


Figure 5.1: Balancing Challenge Level over Time, from Sweetser and Wyeth (2005)

Wyeth (2005) also include "clear goals" in the GameFlow model. To integrate this element Sweetser and Wyeth suggests that games should have clear goals and those overriding goals should be presented early, and intermediate goals should be at appropriate times during the games. Sweetser and Wyeth (2005) argues that integrating the goals into the fantasy is preferable.

Concentration

When describing the flow experience one of the most important criteria is the ability to concentrate on the task (Csikszentmihalyi, 1990). Sweetser and Wyeth (2005) states that “To be enjoyable, a game has to require concentration, and the player must be able to concentrate on the game". Sweetser and Wyeth recommend that the concentration of a player should be kept throughout the game. The ways to achieve this is described through the use of a lot of stimuli that are worth attending to, and a relevant high workload. These methods should not arrive in overwhelming amount, but be adjusted to the player’s cognitive, and memory limits. One of the main pitfalls described is distractions or unimportant tasks, which breaks the state of flow.

Control

(Csikszentmihalyi, 1990) states that control means “The ability to exercise a sense of control over actions”. Sweetser and Wyeth (2005) explains the importance of control over an interface and the game shell, and that games should allow the players to create their own strategies and feel a sense of control over the actions. It is argued that game controls should be easy to learn and the player should be able to use them without extensive practice. Sweetser and Wyeth also explain how errors can reduce the player’s sense of control.

Feedback

Sweetser and Wyeth (2005) formulates that “The players must receive appropriate feedback at appropriate times". It is suggested that players should receive immediate feedback on their actions, feedback on progress towards their goal and always know their status and score. Feedback has the important task to give the player hints and directions towards the goals and achievement of skill. Feedback is also important in relation to concentration. Csikszentmihalyi (1990) states that "concentration is possible because tasks provide immediate feedback".

Immersion

Sweetser and Wyeth (2005) states that “The players should experience deep but effortless involvement in the game”. All of the six previously mentioned elements of flow is in their own way contributing to immersion. In combination with these elements, immersion can be triggered by fantasy and emotional stimuli. Sweeter and Wyeth use first person shooter as an example and suggests that ability to activate emotions through their “violent content” is what makes these games so popular. It is stated that the overall goal is to create a level of immersion so deep that the player becomes less aware of their surroundings, worries, and self that they should experience an altered sense of time and feel emotionally involved and a be subject to a deep inward feeling.

5.2 Guidelines for Educational Games: Challenge, Fantasy and Curiosity

Malone (1980) provides a set of guidelines for designing computer games, and organize these into three categories: *challenge*, *fantasy* and *curiosity*. For a game to be challenging, the game must "provide a goal whose attainment is uncertain". Malone outlines several strategies for making a game challenging, for instance, provide simple goals or multiple levels of goals, variations in the difficulty level and present a score to the player. Furthermore, Malone defines two types of fantasies in relation to the player's skill, called *intrinsic* and *extrinsic fantasies*. Extrinsic fantasy provide the player feedback on "whether or not the skill is used correctly", while in an intrinsic fantasies, the fantasy will be used to indicate how the skill is used and how it differs from correct use. This means that the skill is dependent on the fantasy and the fantasy is dependent on the skill. Lastly, curiosity is described as the "motivation to learn", and can be provided by a game environment that has a sufficient complexity. Malone differentiate between *sensory curiosity* and *cognitive curiosity*. Sensory curiosity is the use of audio and visual effects awake interest in the game. Cognitive curiosity is described as a "desire to bring form to one's knowledge structures". Malone states that motivation can be achieved by presenting only some information so that the game has to be played to "bring completeness" to the knowledge structure.

5.3 A Model for Educational Game Design

In this section, we will describe how flow and enjoyment can be combined to create an educational model. Paras (2005) describes a model for educational game design, through the use of flow, motivation, play and reflection. He starts by comparing characteristics of flow, Motivational theory, good learning environment and heuristics for Game design. The Comparison aims to illustrate commonalities in the context of focus, goals, challenge and feedback. The result shows that they embed many similarities. Paras (2005) uses this to explain that well-designed educational games promote play, which causes a state of flow, that increase motivation, which helps the learning process.

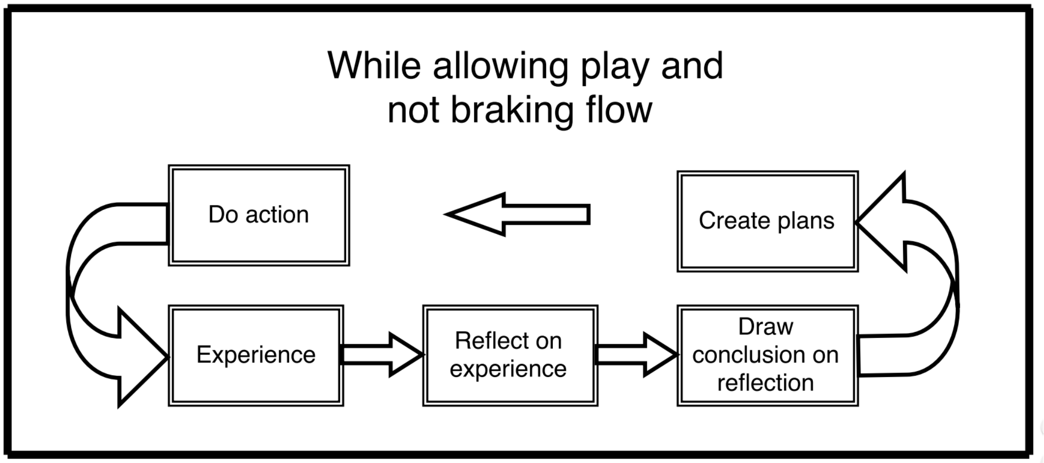


Figure 5.2: Interpretation of Paras (2005)'s Model for Educational Game Design

Paras (2005) states that instructional computer games, in essence, have the same learning capabilities as "active learning". He summarizes active learning as a process where the student engages, and to some extent create their own learning experience. In this process ,the learner participates in the learning and teaching process. Active learning is a cyclic process where the learner first experience, then reflects on those experiences, using the reflections to draw conclusions, use the conclusions to create plans that become new actions that create new experiences and so on (see Figure 5.2). If you remove one of this steps you want to get a conclusion leading to a new action. Paras (2005) then argues that while in a state of flow there is not always room for reflection. Learners experiencing flow may perfect their skill, but does not necessarily need to reflect on their experience. Paras states that this limits the amount of what they can learn. Making the learning process less optimal. Brad Paras argues that if the play and flow are broken up by reflection, many of the benefits of using games will be gone. It is therefore suggested that reflection should be a natural part of the gameplay.

Summary

In this chapter, we presented the GameFlow model, which is based on the concept of flow, and can be used to describe enjoyment in games. This model includes several elements that will guide the design and development of the game for this study. Additionally, it will be used to evaluate the game. Furthermore, we presented some guide for including learning in games in a "fun" way. A taxonomy was presented which includes challenge, fantasy, and Curiosity, Lastly, a model for educational games was described, which underline that reflection should be an essential part of learning games.

Chapter 6

Game Concepts

There are no hard rules on how to make a game, but researchers and game designers have defined many concepts that one could benefit from investigating. This chapter contains descriptions of some concepts that can be used when creating games. The aim is to learn more about game mechanics, and what game mechanics can lead to. This chapter is based on Chapter 5 "Game Mechanics" from the prestudy Askestad and Knutsen (2016).

6.1 Mechanics, Dynamics, and Aesthetics

Games are generally known to be complex. The MDA (Mechanics, Dynamics, and Aesthetics) framework help to understand games by using three components: mechanics, "dynamics" and aesthetics (Hunicke et al., 2004). Hunicke et al. (2004) state that mechanics can be described with data structures and algorithm, the rules, action and control which is available to the player. "Dynamics" are the result of the players actions and the mechanics in the game. A shooting game is used as an example - the mechanics define the weapons, ammunition and movement which can generate dynamics such as sniping and camping. The authors describe aesthetics as the result of the dynamics, and propose a new taxonomy that can be used to better describe words like “fun”. This taxonomy consist of the following components: sensation, fantasy, narrative, challenge, fellowship, discovery, expression and submission. The aesthetics of a shooting game could be described as sensation and challenge, while a game such as "The Sims" would be described as discovery, fantasy, expression, and narrative. Furthermore, it is suggested that the game designer and the player each have a unique perspective on games. Designers start with the mechanics, which create the dynamics. The player considers the aesthetics first, before giving attentions to the dynamics and mechanics.

6.2 Game Mechanic Categories

Having defined games as three components, we can now turn our attention to the term that many game designers find most important, namely game mechanics. There is currently no widely accepted definition for game mechanics. Sicart (2008) highlights this issue by analyzing the major work on this topic, and states that mechanics are often defined out of a specific context. Furthermore, it is stated that game mechanics can often be understood in terms of *verbs*, and the elements that these "verbs" act on.

To gain a more descriptive overview of game mechanics, we will use a taxonomy provided by a book on game design which chooses to view games from many different perspectives - Schell (2014) (Chapter 10, 2014) uses six categories to describe the term:

Space - Defines the different places that can exist in a game when all the visuals are removed, and can be described using a mathematical construct. Space can either be discrete or continuous. For instance - chess would be an example of a discrete space since there are no values in-between the squares on the board. Further, space can have a number of dimensions, most often two or three. Often one has to consider different numbers of dimensions for a single game. For example, a game of pool uses 2-dimensions most of the time, but if the player shoots one ball over another, it has 3-dimensions. At last, spaces can also bee connected to other spaces. Many role-playing games does for example move between an “outdoor space” and an “indoor space” (p. 130 - 134).

Objects, Attributes, and states - Objects is everything that can be manipulated by the player, and can be described as the *nouns* of the game. Attributes are information about an object, for example, the object’s position in the space. Attributes can have states, which can be described as the different values a given attribute can take (p. 136).

Actions - Described as ”what the player can do” and is the *verbs* of the game. One can choose to look at actions in two ways - First, as operative action, which is the most basic actions available to the player. Second, as resultant actions, which are actions that the player uses to achieve a goal over time, and is often associated with strategy (p. 140 - 141).

Rules - Defines as the most fundamental mechanic. Parlett (2005) defines different types. Operational rules are the players understanding of the rules while the foundational rules are the formal mathematical rules in the game. Schell (2014) express that the most important rules are those related to goals, and advice that these should be clearly stated, achievable and rewarding (p. 148 -149).

Skills - Games may require that the player uses some kind of skill. Most skills would fall into one of three categories - physical skills, mental skills and social skills. The list is of skills that may be included in a game is long and one single game might be based on several, for example, memorization, observation, and puzzle solving (p. 150 - 152).

Chance - Described as an essential part of some games since it leads to uncertainty. Chance can be expressed using probability. (p. 153)

It could be argued that some categories should not be taught of as mechanics. For instance, Sicart (2008) ask if mechanics should be separated from the concept of rules. Further, Including skills in the taxonomy one might also question since it changes the perspective of the player, and away from the designer. However, the taxonomy provides an expressive way to defining mechanics the term, while being both precise and flexible.

6.3 Economy

Other attempts at describing categories of mechanics, includes the work of Adams and Dormans (2012), which lists several types of mechanics that one can find in a game: physics, internal economy, progression mechanics, tactical maneuvering and social interaction. Adams and Dormans (2012) goes into depth explaining internal economy, especially in relation to strategy games, where he pick apart popular titles such as *StarCraft* and *Ceasar III*, and describe their inner workings in detail.

An economy is described in terms of *resources*, *entities* and four *functions*. Resources can either exist in a game world as a physical object, or as a value. The game consists of entities which are storing the resources. An entity can either store a single kind of value (simple entity) or multiple kinds of values (compound entity). Four functions are listed which concerns how resources are handled. Sources produce resources at some point in time. It can happen continually, in different increments or be a response to an event. Drains are mechanics that consumes resources. Converters take one resource and convert it into another resource. The Trade function move resources between entities. Adams and Dormans (2012) point out that many games utilizes converters concept when the player has the ability to upgrade to improve for more efficiency.

6.4 Emergence and Progression

While game mechanics act as "building blocks" for creating games, it also interesting to know about the effects that these can lead to. One such effect or "dynamic" is *emergence*. Juul (2002) describes emergent games as games that use simple rules that result in many variations and challenges that the player has to attend to, while often using strategy. Another concept that is often compared to emergence is progression. These are games where the designer has planned the events for the player to meet. Juul (2002) further states that one way to separate these two types of games is that games of progression often lead to "walkthroughs", while emergent games often produce strategy guides or general tips on how the game should be played.

Summary

In this chapter, some concept were outlined that can be used for creating games. The MDA framework provides a way of understanding games in terms of mechanics, "dynamics" and aesthetics, and explains that games can be viewed from both a player's and designer's perspective. In this study, we will choose one or more elements from the aesthetics taxonomy provided by the MDA framework, and use it as a guide when creating the game design. Furthermore, there will be a focus on game mechanics, and the categories defined in Section 6.2, will provide a basis and inspiration for constructing mechanics. It will also be noted that it can be reasonable to refer to game mechanics as player actions. Lastly, game mechanics that can lead to emergence will be prioritized, as this concept associated with strategy games.

Chapter 7

State of the Art

Previous research presents some games that have been used to address the issue of teaching project management. In this chapter, we will describe some of these games, including a previous master thesis with the same assignment as this study. Additionally, a few commercial games will be described. This chapter is based on Chapter 3 "State of the Art" from the prestudy Askestad and Knutsen (2016).

7.1 Incredible Manager: A Simulation-Based Game for Project Management

Dantas et al. (2004) investigates the issue of educating project managers and suggest an experimental learning process for project management. They propose a simulation-based game called *The Incredible Manager*, where students can take the role of a project manager and act in a risk free environment. The game starts by presenting a project to the player, which contains tasks, quality, budget demands and constraints. The player are then asked to create a project plan which includes hiring "appropriate" developers and assign these to work on tasks. The player then needs to get this plan accepted by the stakeholders. After this, the "execution phase" starts, where the player can modify the project plan to meet the cost and time requirements. See Figure 7.1 for an overview of the execution phase. The project ends either as a success or failure depending on choices that the player made. An evaluation of the game were conducted and 7 subjects tested the simulation-game. Some issues with the current game model are pointed out, which includes situations that may occur in a real project management setting such as "Multiple developers working on a single task", "Social interaction" and "Organizational issues".

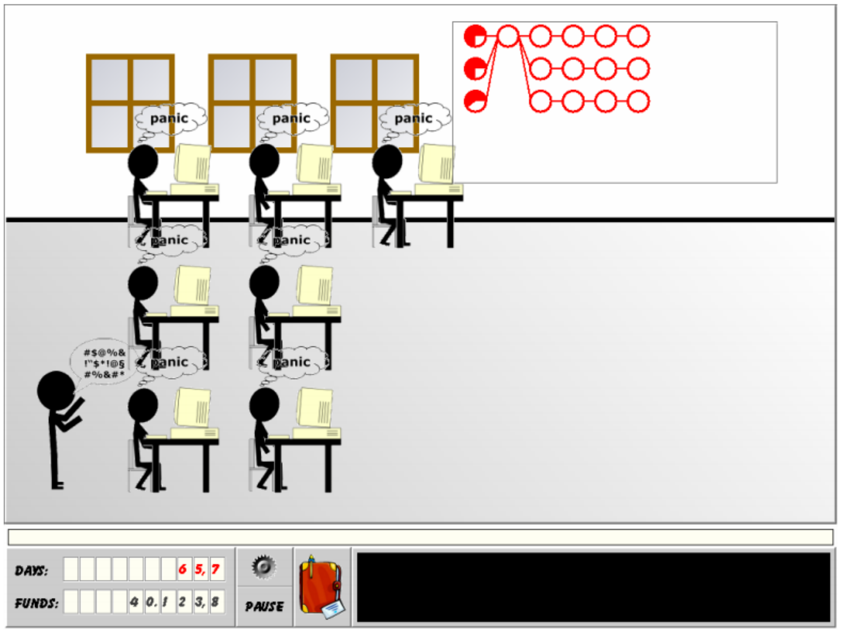


Figure 7.1: The Incredible Manager: The Office Room from Dantas et al. (2004)

7.2 SimSE:A Software Engineering Simulation Environment for Software Process Education

Navarro (2006) points out that students are taught the theory of software engineering process, but there is a lack of ways that these concepts can be put to practice. A game-based simulation environment called *SimSE* is then proposed. In the game, the player take the role as a project manager, with the goal of completing a software engineering task that they are presented with in the beginning of the game. When the game stars, the player can hire employees, assign tasks and "monitor progress". At the end of the game, the player will receive a score that reflects the performance. One important thing to take note of is that there exists multiple versions of the game, each meant to teach a specific software engineering process. The game were evaluated on 29 students which showed potential for using the game as an educational tool, while some issues were pointed out. More recent evaluations of the game seems to support the claim that the game can be used successfully in an educational setting (Navarro and Van Der Hoek, 2009).

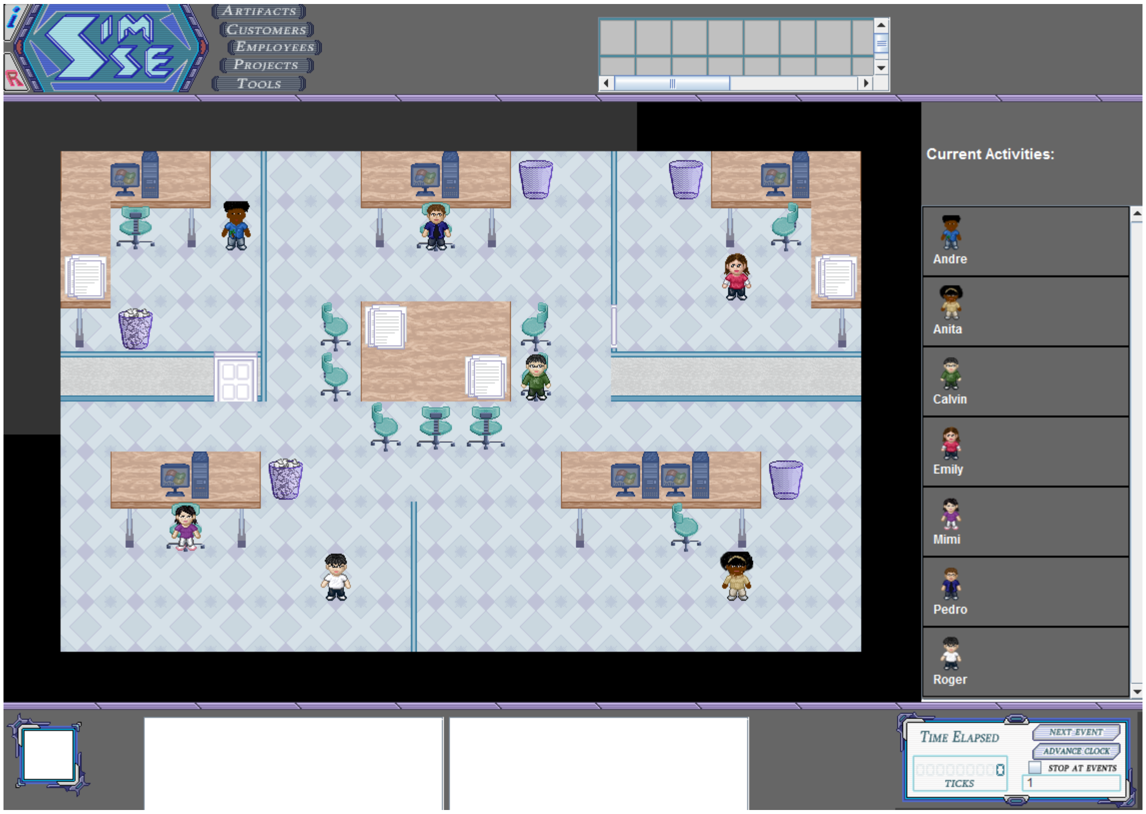
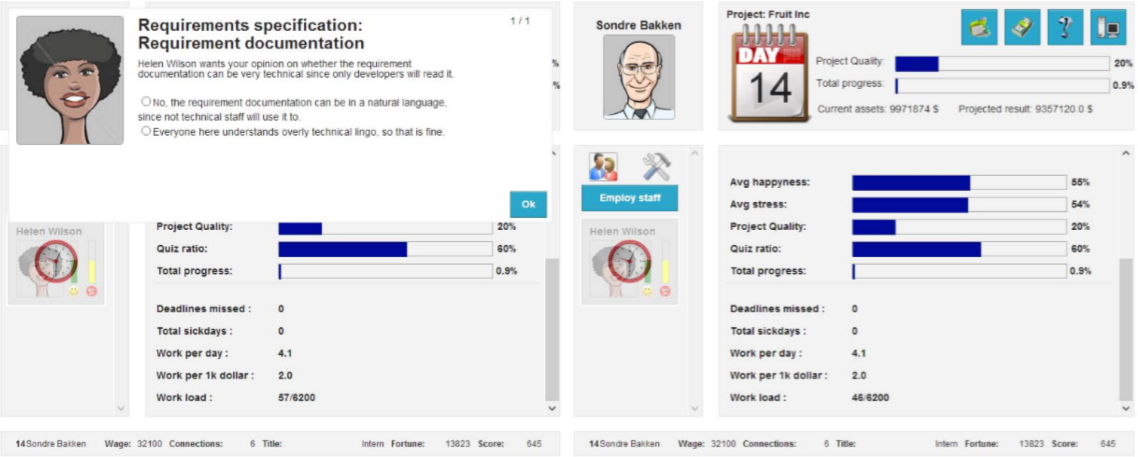


Figure 7.2: SimSE: An Overview of the Office Environment

7.3 The Software Engineering Game: A Previous Master Thesis

There exists one master thesis with the same assignment as this study. Bakken (2013) proposes the game *Software Engineering Game*, with the goal of evaluating different educational strategies. The game puts the player in the role of a project manager, with to goal of creating a web delivery system for a pizza bakery. When playing the game, the player has to assign employees to work on tasks. The player can make the progress faster by matching the employees skills with the skill requirements for the task. Further, the player has to manage the employees stress and happiness values, or else the employee will become sick. To present the educational material, the game includes a "Wikipedia" which contains a collection of articles with content that match some of the objects presented in the game. In addition, the game provides quizzes that are related to the "Wikipedia", and the player receives a bonus for answering correctly. An evaluation shows that there were some issues with the usability of the game, but most of the participants rated positively on several of the educational strategies used in the game.



Requirements specification:
Requirement documentation

1 / 1

Helen Wilson wants your opinion on whether the requirement documentation can be very technical since only developers will read it.

○ No, the requirement documentation can be in a natural language, since not technical staff will use it to.
○ Everyone here understands overly technical lingo, so that is fine.

Project Quality: 20%
Quiz ratio: 60%
Total progress: 0.9%

Deadlines missed : 0
Total sickdays : 0
Work per day : 4.1
Work per 1k dollar : 2.0
Work load : 57/6200

14 Sondre Bakken Wage: 32100 Connections: 6 Title: Intern Fortune: 13823 Score: 645

Project: Fruit Inc
Project Quality: 20%
Total progress: 0.9%
Current assets: 9971874 $ Projected result: 9357120.0 $

Avg happyness: 55%
Avg stress: 54%
Project Quality: 20%
Quiz ratio: 60%
Total progress: 0.9%

Deadlines missed : 0
Total sickdays : 0
Work per day : 4.1
Work per 1k dollar : 2.0
Work load : 46/6200

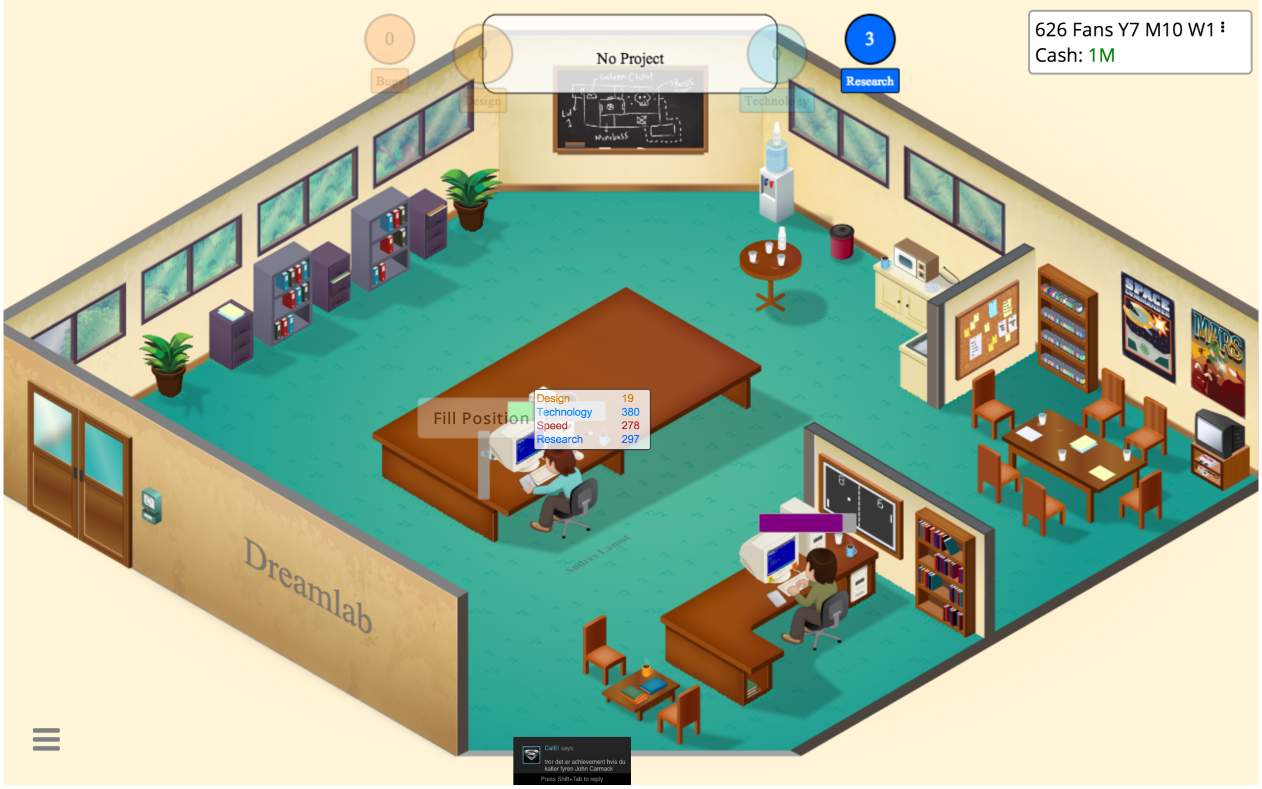
14 Sondre Bakken Wage: 32100 Connections: 6 Title: Intern Fortune: 13823 Score: 645

Figure 7.3: Software Engineering Game, from Bakken (2013)

7.4 Game Dev Tycoon: A "Business Simulation Game"

Several commercial games inspired us to choose this assignment. The most notable are *Game Dev Tycoon* [1], a "business simulation game" that puts the player in the role of a manager for a game development company. Figure 7.4 displays an overview of the game. The main activity in Game Dev Tycoon is to create games. The player is then asked to choose a topic, genre and platform for the game. The development is divided into three phases, which again have three focus areas to choose from. The focus areas should relate to the chosen genre and platform, which will contribute to how the game is going to be received. After creating a game, a summary of the development is displayed, which consists of experience points gained in each of the focus areas. While developing, you will also acquire points in design, technology and research, which can be used throughout the game. After the game is released, you will be presented with reviews which will determine the weekly profit of the game." Similar to the games previously mentioned, Game Dev Tycoon display an office environment, but make more use of typical some features, such as "researching technology".

[1] http://www.greenheartgames.com/app/game-dev-tycoon/



CalEl says:

tror det er achievement hvis du kaller fyren John Carmack

Press Shift+Tab to reply

Figure 7.4: Game Dev Tycoon: "Business Simulation Game"

7.5 The Sims: A "Life Simulator"

One other game that inspired us to take this master assignment was *The Sims*, a "life simulation game". See Figure 7.5 for an overview if the game. In this game, the player are in charge of a household. The player can design their own characters and build a house for them to live in. The players job is then to manage their lives, may which include improving their skills, and monitor their emotions to make sure that they are in a good mood. The user interface in the game, allows the player to easily change between characters, view a character's status and more. The Sims is an example of a highly successfully game, and it is interesting to note that the player is not provided with a concrete goal, but should instead make one up, which is a characteristic of emergent gameplay.

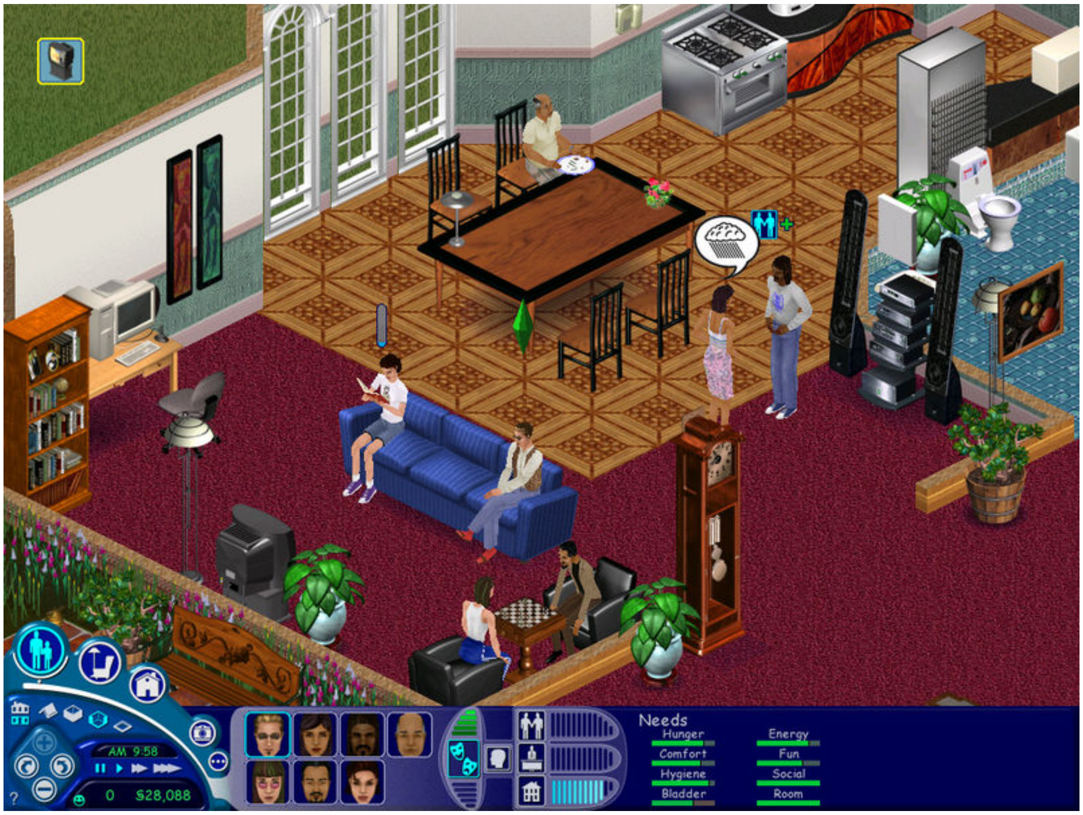


Figure 7.5: The Sims: "A Life Simulator"

Summary

In this chapter, we described some previous research that has resulted in games where the player can act as a project manager. In the Incredible Manager, player goes trough different phases in a project. In SimSE, the player takes control over developers and apply a software engineering process. Further, we described a previous master study "The Software Engineering Game" which tested different learning strategies. At last, we looked at two commercial games which make use of more typical game features. In this study, we will make note of and use as inspiration, some of the features included in these games, especially the actions that are available for the player, and the different objects in the games. The Sims is especially interesting as it provides a more advanced user interface compared to the other games- Additionally, it shows some properties of emergence, for example that the player does not have a predefined goal. Emergent games are interesting because they are often related to the use of strategy.

Part III

Design and Development

This part will first present the design and development process for creating the game. Then, the game design will be described. Lastly, the choice of technology will be presented, and an overview of the game from a system perspective.

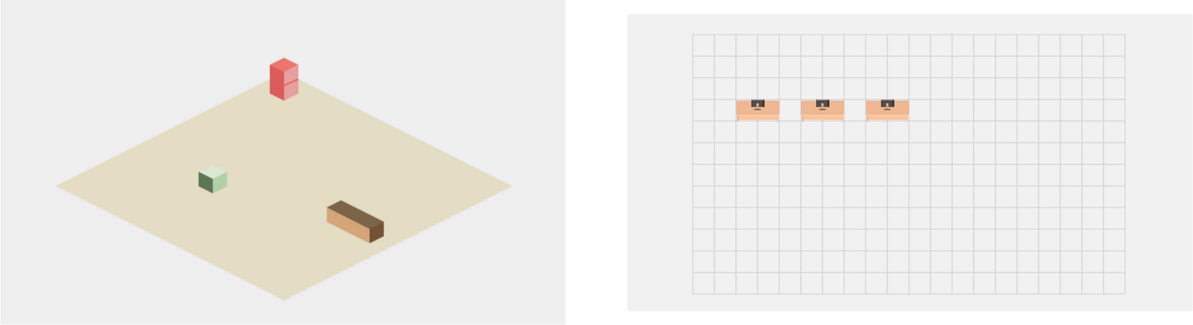
Chapter 8

Prototype Development

The major goal of this study is to create a game that can be used to learn project management. In this chapter, we will explain the process of creating the game design for this study.

8.1 Ideas from the Prestudy

The work conducted in the prestudy Askestad and Knutsen (2016) resulted in some ideas for the game. Most important were the decision to present the available player actions trough an interface element similar to what is used in "The Sims" and "Game Dev Tycoon". When the player clicks on a character in the game, this user interface element will appear over the character, and act as an intuitive way of giving orders. In addition, we tested to kinds of perspectives, isometric and top-down (Figure 8.1). While the isometric-perspective did provide a more realistic view of an office environment, it became too cumbersome to create the artwork that was needed. Therefore, we ended up using a top-down perspective for this project. Lastly, some game model elements were identified such as tasks, skills, mood, and stress, which were inspired by the game model in the "The Software Engineering Game" by Bakken (2013). While the ideas from the prestudy, provided a useful starting point, we were far away from having a complete game design.



(a) Isometric

(b) Top-down

Figure 8.1: Two Types of Perspectives Prototyped, from Askestad and Knutsen (2016)

8.2 Designing the Core Mechanics

At the beginning of this process, we wanted to experiment with how we could create a game with emergence. Juul (2002) explains that strategy games often rely on this concept, which is the main reason for choosing such an approach. Our idea was that the employees should be able to move around in the office and interact with the environment, similar to a game like "The Sims". We hoped that this would create some complexity which we could build our core game mechanics on. To accomplish this, we created a tile-based game world, and we implemented path-finding using the A* Algorithm. The employees could then move around in the office while avoiding objects. Next, we wanted the characters in the game to behave on their own. As a result, we created a scenario where a character would move to the coffee machine, drink a cup of coffee, and then move back to his seat. After implementing this, we did not get any ideas for a core mechanic, and the pathfinding and state machines used in the code had led to the system being complicated, which made it time-consuming to experiment further. We admit that we might have been too ambitious when taking this approach, so we decided to take another direction.

It was decided to shift the focus to experiment with ways to include a software-theme into the game, with the notion that creating a unifying theme can make it easier to decide which elements should be included (Schell, 2014)(p. 49). The idea was that a task would be to create a specific software module. An employee would then need to have some software related skills to be able to work on the tasks. We implemented two types of player actions: "work on task", and "improve skills". To see how these player action relate to game objects, we will use a diagram notation inspired by a video presentation with Ralph Koster "A Theory of Fun, 10 Years Later"

(Koster, 2014), Figure 8.2 displays the player actions "work" and "improve" in relation to the game objects "task" and "skill".

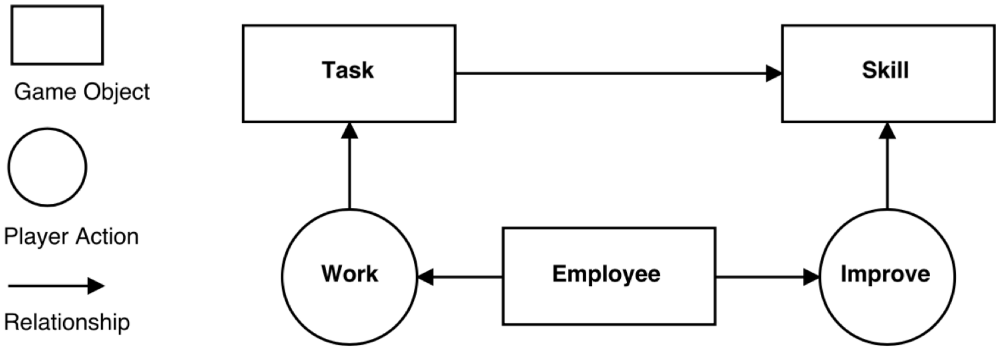


Figure 8.2: The Initial Mechanics

The fact that modules in a software system often depends on other modules meant we could add more depth to the game by creating dependencies between the tasks. By varying a number of tasks, dependencies between the tasks and the required skills allowed us to construct different levels. Furthermore, we included a time limit for every project. Applying time-pressure is a typical challenge found in many games. This meant that the player would have to specialize some of the employees to able to complete the level before the time limit, leading to some potentially interesting gameplay.

To make sure that this concept could be enjoyable, it was essential to develop a working prototype that we could play-test. One of the challenges when creating this prototype was to create an intuitive user interface. In particular, coming up with a way to display a lot of information to the player, without having to display too much at once. We looked for inspiration from popular strategy games and found that many had solved this problem by displaying hints/tooltips when the player hovers over elements with the mouse.

Having created a prototype, we could begin to play-test, on two experienced gamers. We observed them while playing, and noticed that players made use of the hints without the need of telling them. After the play-testing, we had an open discussion about the some of the elements in the game. This led us to keep the current game mechanics as a basis, which meant we could add more features.

8.3 Improving Gameplay

We continued to experiment with additional mechanics and ended up adding some new elements to the game. The main reason for this was to create a higher workload for the player. Sweetser and Wyeth (2005) mention that games need a certain workload, for the player to be able to concentrate. Another reason was to make the simulation of the office environment more complex and somewhat more realistic. We decided that every employee should have a mood, which could be affected by the decisions the players made. Stress was introduced as a way to decrease the mood of the employee, and a new player action "take a break" were introduced, so that the players would have the option to decrease the stress level.

At this point we also experimented with the "economy" described in Section 6.3. We tried to look at the game in terms of "resources", and look at how "resources" could move between "entities". Adams and Dormans (2012) uses the *machination framework* [1], to design and simulate game mechanics. While Adams and Dormans (2012) shows that economy can be used simulate strategy games, we found it more time efficient to only rely on playtesting when testing mechanics. However, the important thing that we took from economy concept, was that as Adams and Dormans (2012) points out, many games often provide an option for the player to invest in "upgrades" to improve for more efficiency. This lead to the idea that there could be two types of skills. One type of skill that was required to work on tasks, and one type of skill that allowed the player to improve the "working" efficiency. Eventually, this leads to a second iteration on the mechanics, using the same diagram notation as before (Koster, 2014). See Figure 8.3 for an illustration of the player actions and the game objects where "mood", "stress", "take a break" and two types of "skills" is included in the game.

When implementing the new elements into a new prototype, the major difficulty was again the user interface and usability. At this point, we were most concerned with giving feedback to the player. Sweetser and Wyeth (2005) formulate that players should receive immediate feedback on their actions. In addition, we may include that the player should receive feedback about the most important state changes in the game. We did several things to deal with this user interface issue. Firstly, and the most important were that we created a lot of custom graphics. Figure

[1] http://www.jorisdormans.nl/machinations/

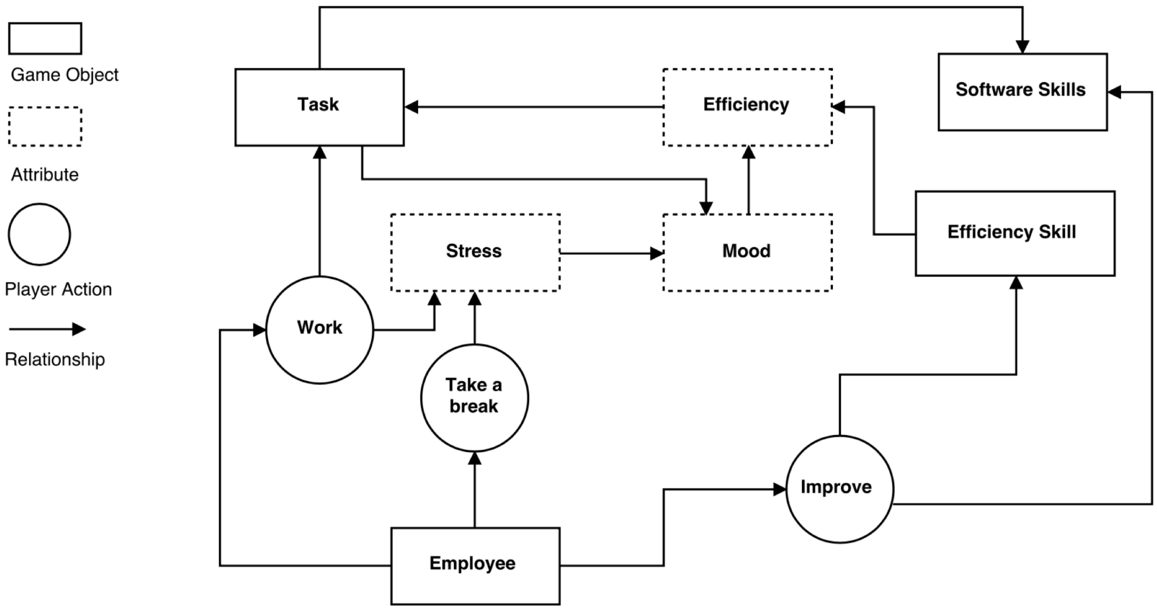


Figure 8.3: Two New Game Mechanics

8.4 shows a collection of the graphics used in the game. This allowed us to create exactly the kind of symbols that we needed. Secondly, we used sound effects to give the player feedback on actions. For example, a sound is played every time the player opens a menu or assign work to an employee. The last thing that should be mentioned is that took inspiration from "The Sims", and included a *speech bubble* to give the player feedback on an action or a state change. An example is when a player assigns a task to an employee that is in the "sad" state. A speech bubble will then appear over the head of the employee saying "I'm not in the mood". Finding intuitive ways to represent state information to the player, were some of the most challenging tasks at this stage of design and development.

This prototype was tested on the supervisor for this study. He said that he liked the game, while pointing out some inconsistencies. He played through all the levels in the game, and while observing him play, it seemed that he remained concentrated. He did also seem to be able to play the game without the need for an explanation of the user interface, suggesting that the usability was sufficient. We decided to keep the existing game design, but we were yet to find a good way of presenting learning material.

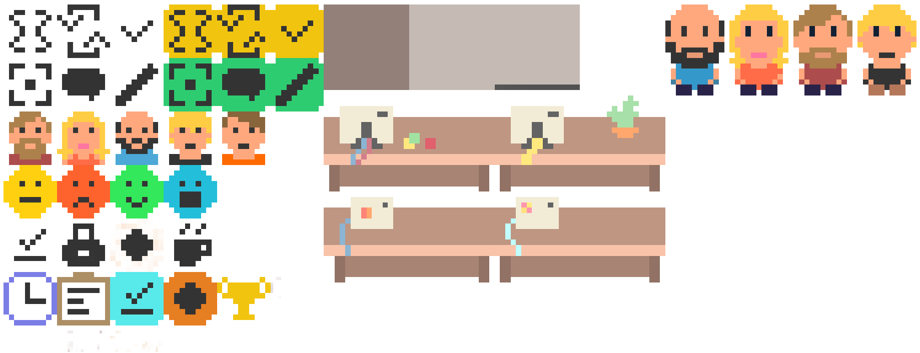


Figure 8.4: The Graphics Used in the Game

8.4 Learning Material and Abilities

One concern was how the player should learn project management in the game. Adams and Dormans (2012) advice that learning games should be designed "around the subject", and that game mechanics should be used to simulate typical challenges on the subject (p. 288). A few challenges that we wanted to simulate were some actions related to "project planning" and "project monitoring", such as assigning tasks and completing a project before the estimated time. In addition, it was decided to present learning material about project management with textual descriptions. Sweetser and Wyeth (2005) states that “players should not be distracted from tasks that they want or need to concentrate on”. Therefore, we attempted to put the textual descriptions in appropriate places in the game, such as after a level has been completed, or before a level starts. We wanted to include the textual descriptions as part of the theme or fantasy of the game. Malone (1980) states that an important part of fantasies closely related to the learning material, is that players can use their knowledge to understand the unknown things presented in the game. Therefore, we included a character “the boss”, which gives the player useful advice on how to play the game, while presenting some of the learning material. Furthermore, we included *abilities*, meaning an aid that the player could use in some situations. If a player would choose an "ability" that increases the employees’ "mood", a textual description illustrating the importance of "team building" would be presented. We hope that the connection between an element in the game and information about some project management concept would interest the player to read the texts.

Figure 8.5 shows the places where the textual descriptions are presented to the player, such

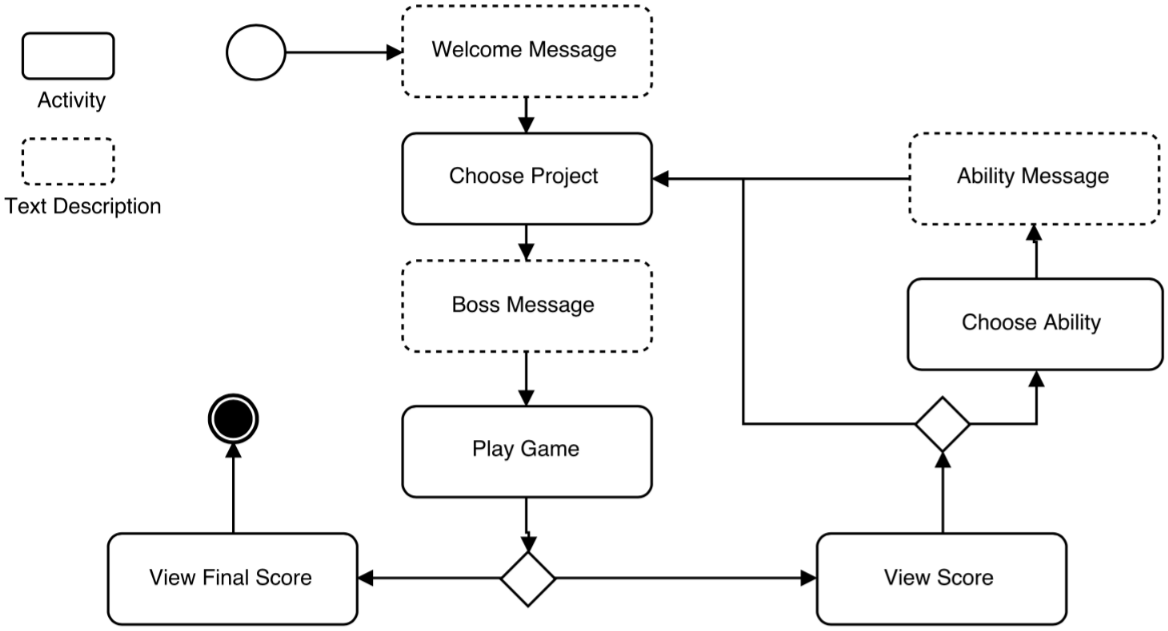


Figure 8.5: Learning Material at Different Places in the Game

as: in the beginning of the game after a project has been chosen, and after an ability has been chosen.

8.5 Balancing the Game

For the last part of the process, we tried to adjust values in the game to create an appropriate level of difficulty, this is known as balancing the game. This is also related to an important element in the GameFlow model. Sweetser and Wyeth (2005) states that "the level of challenge should increase as the player progresses through the game and increases their skill level". When balancing the game, we started with the values that we taught of as the most important - number of tasks per project, and their skill requirement. To get an overview of these values, we used a spreadsheet and used this to select values that would increase the difficulty as the player would progress. In addition to the levels, we also spent some time adjusting the values that had an effect on an employee's efficiency, as this will determine the speed at which the player can finish tasks. At this point, we were unsure about how others would experience the difficulty level. So we decided to test the game on four experienced gamers. After they had playtested the game, we received valuable feedback - for example that one of the abilities could completely remove the time-pressure. We used the feedback from the players to adjust some of the values one last time, trying to make the game a little more challenging.

Summary

In this chapter, we described the process of creating the game "Project Manager". We began be exploring ways of making an emergent game before we decided to focus on ways to include a software-theme into the game. This led us to a core mechanics which were then iteratively improved. We include user interface and usability as one of the most difficult challenges when prototyping the game.

Chapter 9

Game Description

In this chapter, we will describe the game that has been created for this study, called *Project Manager*. We will begin with an overview of the game concept, before going into details on how the game is played. The game is hosted at `http://mohff.github.io/project-manager/`¨. We recommend using Google Chrome when playing the game.

9.1 Concept

"Project Manager" is a single-player browser game where the player takes the role of a project manager in a software company. See Figure 9.1 for an overview of the game. The game is set in an office environment, where there are four employees that the player can control. As the project manager, the player is asked to complete a series of projects. The main activity in the game is to assign employees to work on tasks, and to train the employees so that they gain skills. Some skills allow the employee to work on new tasks, and others make the employee work faster. The challenge in the game is to specialize employees so they can best work together and to complete projects before the time limit.

The player is presented with learning material throughout the game, which includes information about project management, as well as tips on how to play better. The game does also provide some challenges related to project management such as delegating tasks, and completing the tasks before the deadline.

In addition, the player has to monitor the employees' emotions or emotional responses. An

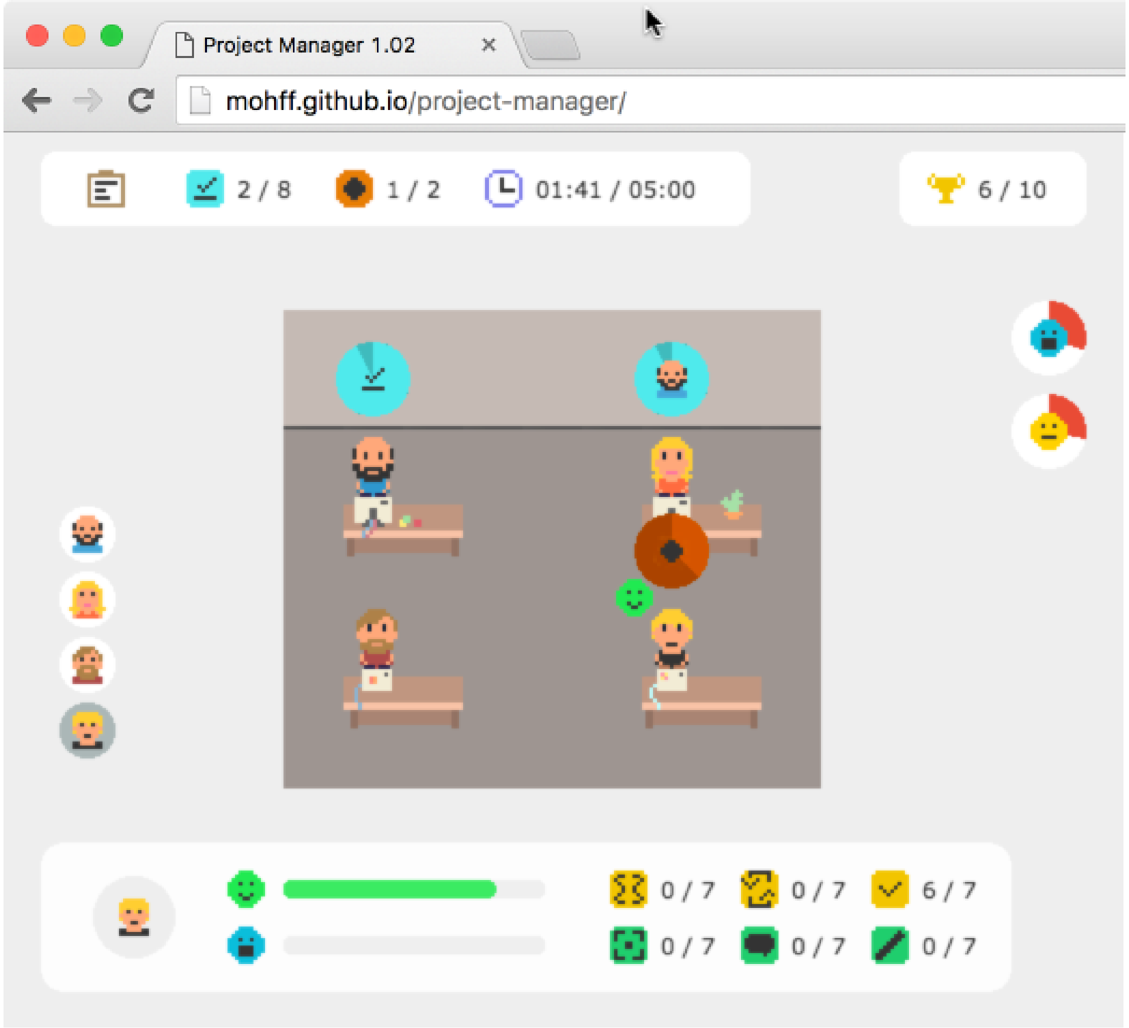
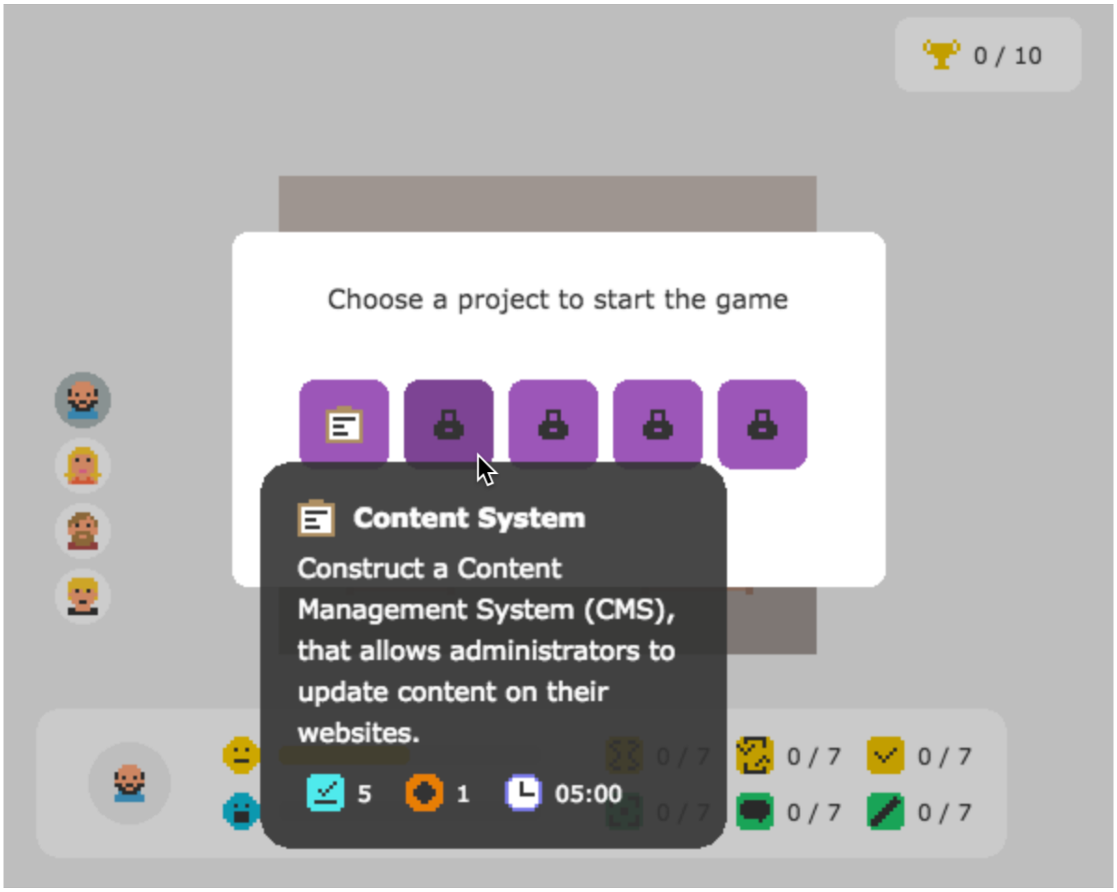


Figure 9.1: Overview of "Project Manager"

employee becomes happy when tasks are completed, and sad if they get to stressed. Stress can be removed if the player orders employees to "take a break". It is important for the player to monitor an employee's emotions because it can greatly affect the efficiency when working on tasks. As the game progresses, the player will also be given special abilities, which may prove helpful in some situations.

9.2 Beginning a Level

In the beginning of the game, the player is presented with projects, which are the levels of the game. There is a total of five levels that the player has to complete. Figure 9.2 displays the menu



Choose a project to start the game

Content System

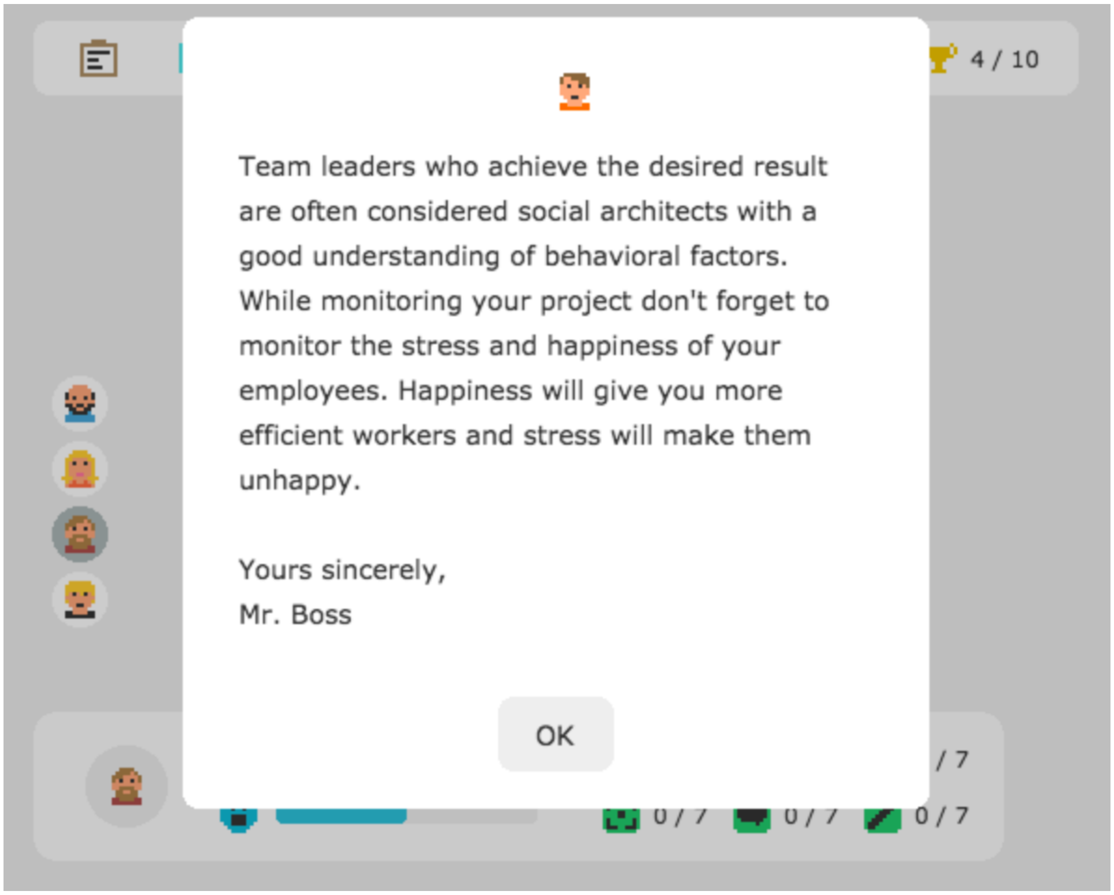
Construct a Content Management System (CMS), that allows administrators to update content on their websites.

Figure 9.2: Project Menu: A Menu Presented at the Beginning of the Game

that the player uses when picking a project. Note that the player cannot really choose a project, as the purpose of the menu is only to show the progress of the game. The player may hover over a button, which will show a description. This description includes information about the software system that the player is going to construct, and is a part of the software-theme in the game. In addition, the information includes a number of project tasks, and the time limit.

The Boss

When a project is selected, the player is presented with a message from a character called *the boss*, see Figure 9.3. This character gives the player information about project management and tips on how to play the game. This is where most of the learning material is presented, and the boss character will continue to appear throughout the game.



Team leaders who achieve the desired result are often considered social architects with a good understanding of behavioral factors. While monitoring your project don't forget to monitor the stress and happiness of your employees. Happiness will give you more efficient workers and stress will make them unhappy.

Yours sincerely,
Mr. Boss

Figure 9.3: "The Boss": A Character in the Game

Action Menu

When the game starts, the player can click on the employees. This opens the *Action Menu*, see Figure 9.4. This menu presents some options to the player, and allows player to give orders to the employees. Three options are displayed - "Work on task", "Improve skills" and "Take a Break".

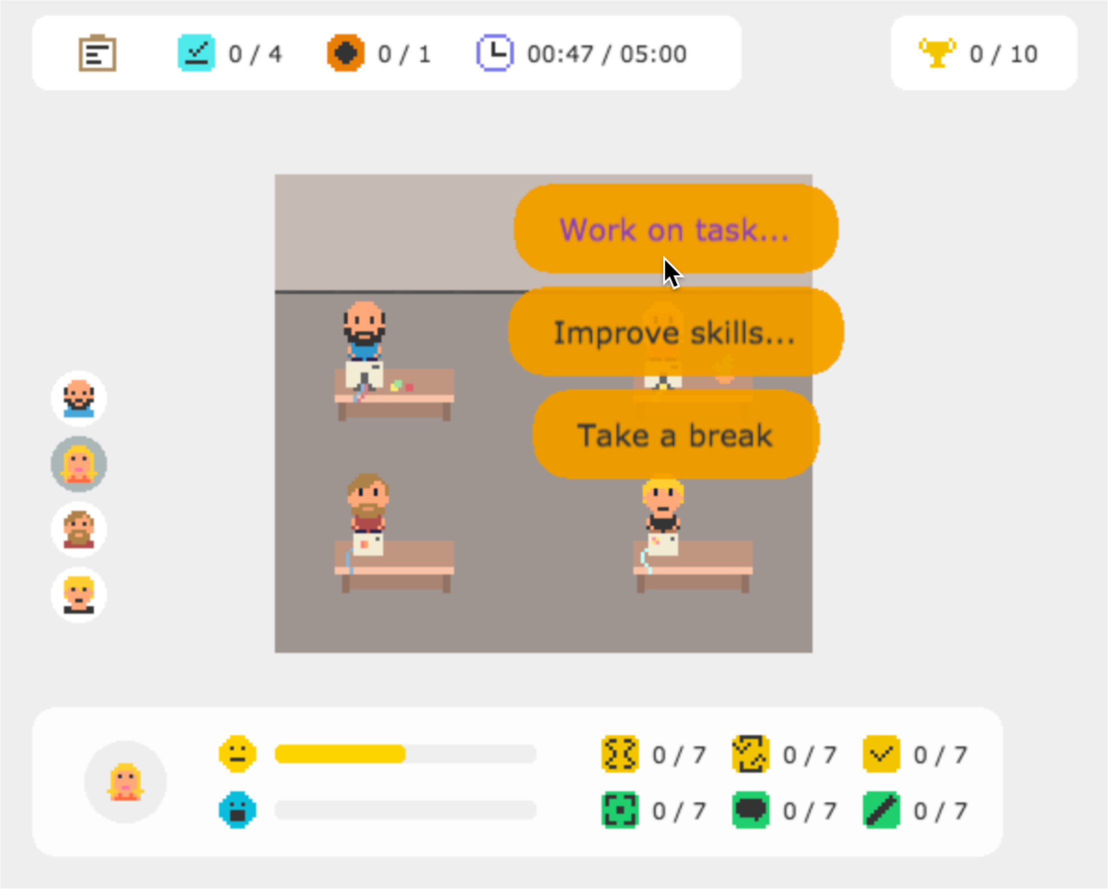
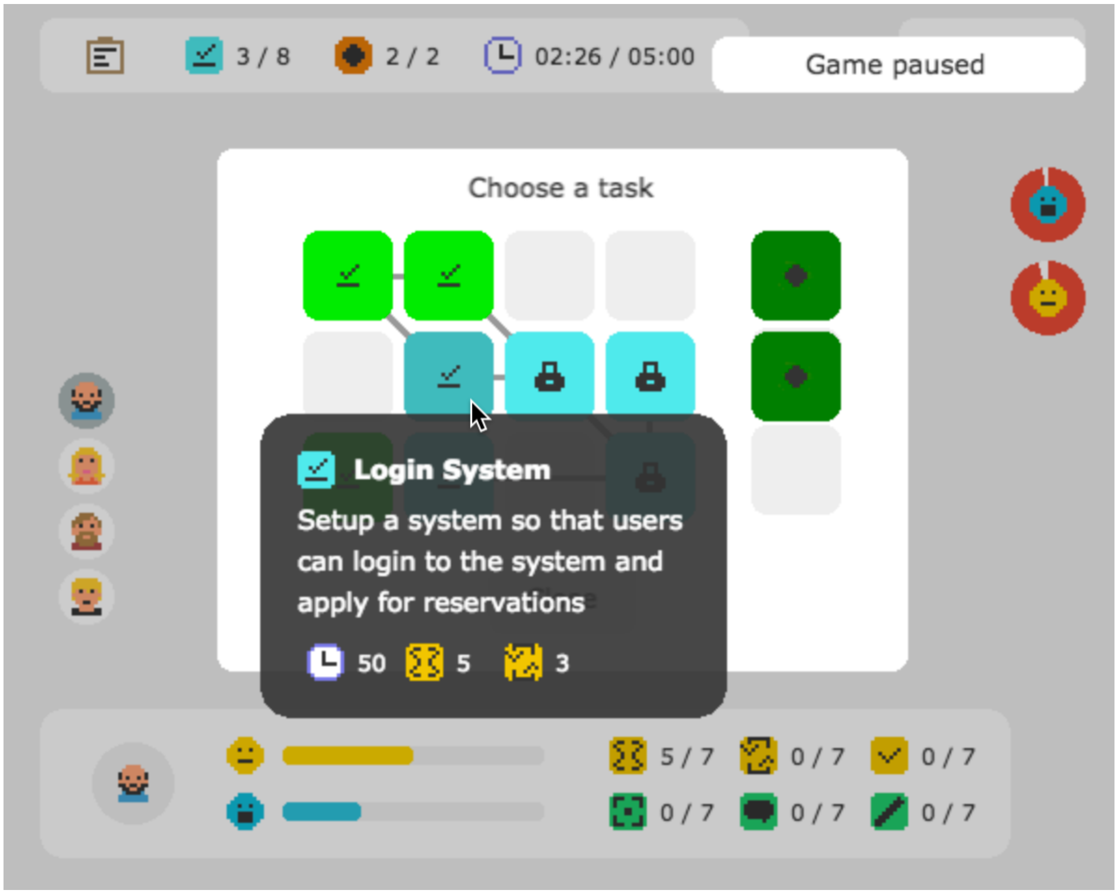


Figure 9.4: Action Menu: A Menu for Giving Orders to the Employees

In this section, we displayed some screenshots and described what happens at the beginning of the game, which includes selecting a project, read tips from "the boss" and give orders to an employee through the action menu. In the following three sections, we will focus on each of the things that the player can order an employee to do.

9.3 Work on Task

Assigning tasks to employees is the most important player action in the game, and is done by picking a task from the *task menu*, see Figure 9.5. This menu is opened by clicking "Work on task..." from the action menu. The task menu displays a collection of buttons, which represent the task that the employee can be assigned. The player can hover over the buttons, leading to information being displayed. This information is a part of the "software theme" in the game and describes the software module that is going to be made, for example, "construct a database that allows the users to store content". In addition, it contains information about the workload and the required skills - the skills that the employee need to have in order to work on the task.

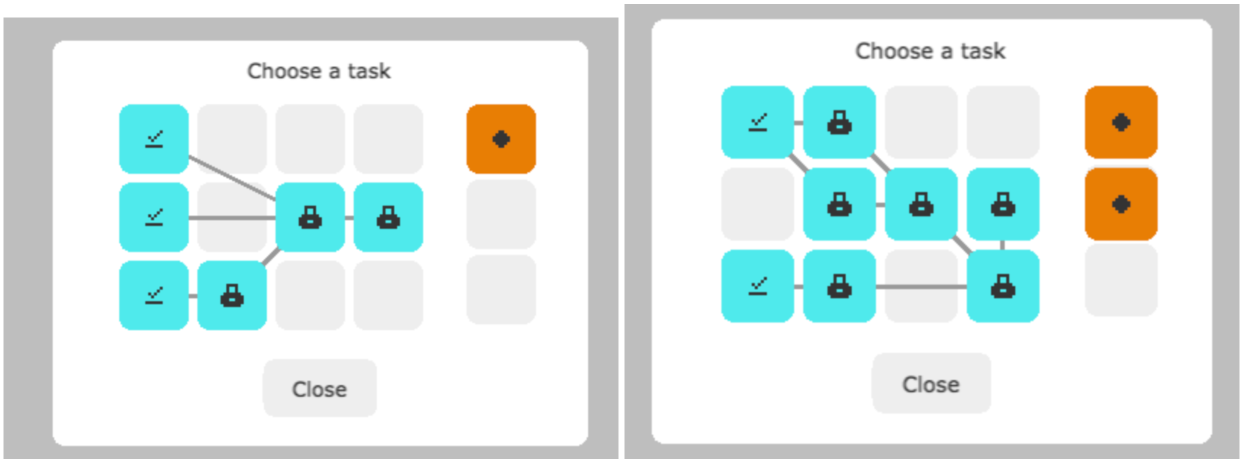


Login System

Setup a system so that users can login to the system and apply for reservations

Figure 9.5: Task Menu: A Menu for Choosing Tasks

The task menu Figure 9.5 displays several other elements that the player needs to consider. First, it displays some gray lines between the buttons. These are important because they represent dependencies between the software modules. A task with unfinished dependencies can



(a) Level 3: "Battlefield Simulator"

(b) Level 4: "Reservation System"

Figure 9.6: Two Levels in the Game

not be assigned to an employee, so the player needs to consider this when planning how the project can be completed as fast a possible. Second, different colors and icons are used to indicate a task's state. A lock icon is used when a task has unfinished dependencies, and a task that is completed has a green color. Using different colors and icons should help the player to get a better overview of the state of the project. Lastly, the column on right side displays another kind of tasks. These are called quality tasks in the game. These are different from normal tasks because only one employee can work on them at a time, and they do not need to be completed, but one score point is gained if they are. Quality tasks display information about a "quality tactic" [1] that is going to be implemented, for example, "Implement active redundancy so the system is protected against". This is a part of the "software theme" in the game. Figure 9.6 shows how a different number of tasks, the arrangement of tasks, dependencies, and quality tasks creates a visual variation in the levels.

[1] http://www.informit.com/articles/article.aspx?p1̄959673&seqNum5̄

Leader and Collaborator

There is a difference between starting a new task, and choosing a task that has been started by another employee. The employee that who starts the task becomes the *leader*. When an employee is assigned to a task that already has a leader, he becomes a *collaborator*. It is only the employee that becomes the leader of the task that needs the skills required by the task, so the player can conveniently assign other employees to collaborate in order to finish a task faster. Figure 9.7 shows one employee that has started a task. This is indicated with a task symbol over his head. Furthermore, it shows two employees that are collaborating on the same tasks, and these have the tasks owner as a symbol over their head. They all contribute to the progress of a task. This is represented by a progress indicator. Starting and collaborating tasks becomes more important when combined with different skills, which is described in detail in the next section.

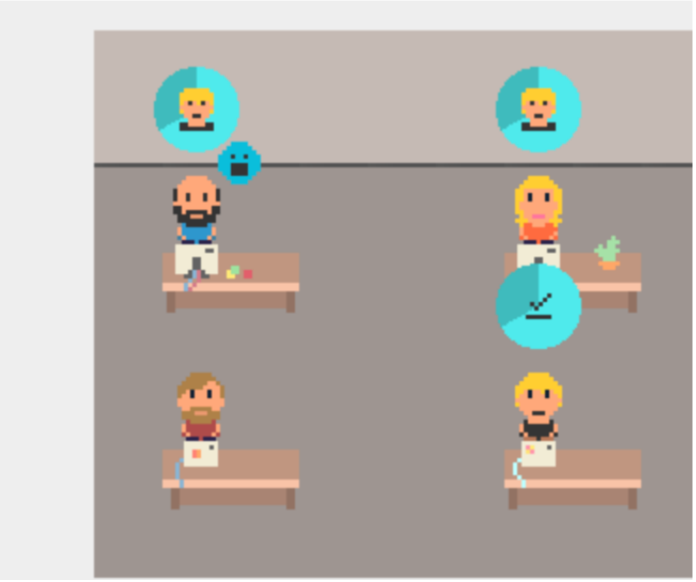
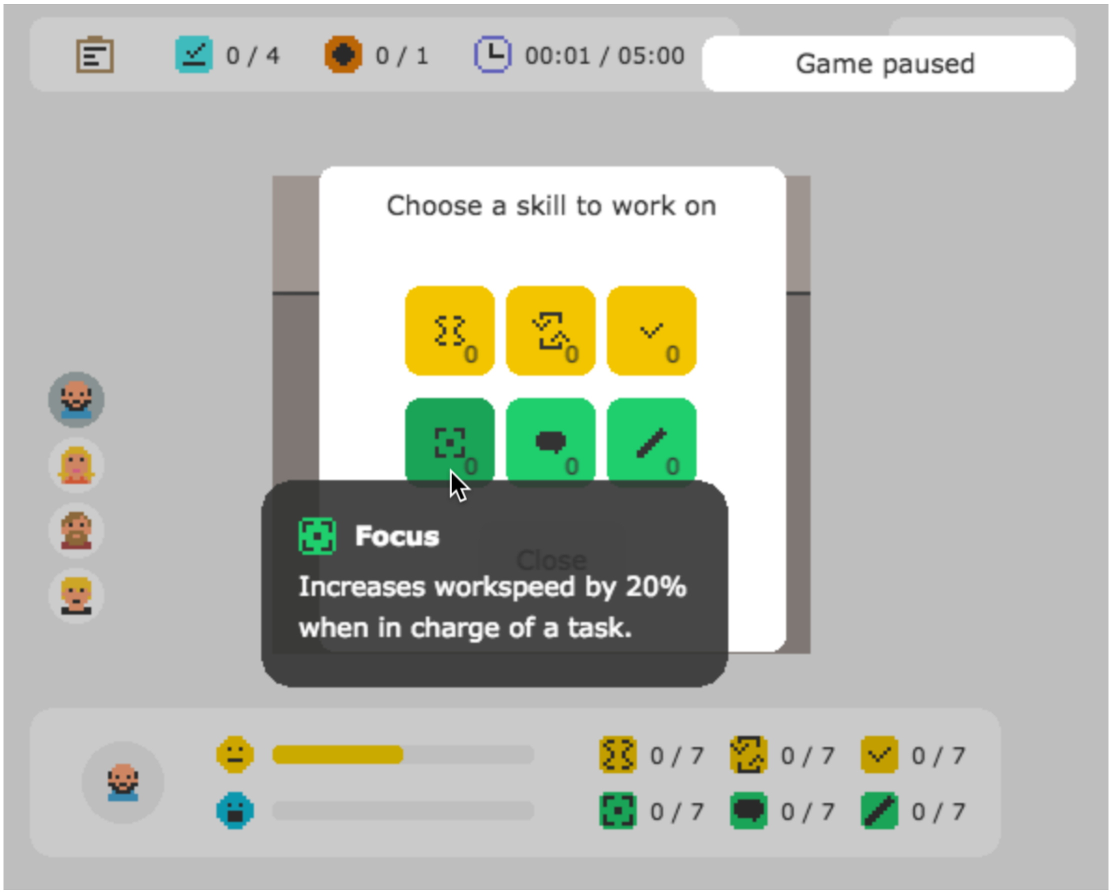


Figure 9.7: A Scenario where Employees Collaborates

In this section, we described how the player can assign tasks to employees using the task menu. Tasks have a workload, required skills, and dependencies, and these attributes are what create variations in the levels in the game. There exist two kinds of tasks, where quality tasks can give the player a better score. Further, we described the difference between starting a task and collaborating on a task.

9.4 Improve Skills

Improving the employees' skills is an essential part of the game. The player can choose "Improve skill..." from the action menu. The *skill menu* will then appear, see Figure 9.8. The menu displays some buttons in different colors which represent different types of skills. An employee can be assigned to work on a skill, by clicking on one of the buttons. This leads to, a progress bar appearing over the head of the employee - similar to working on a task from Figure 9.7. When the skill is finished, the employee will gain a new level, and the time it takes to gain another in the same skill, will increase.


Choose a skill to work on

Focus
Increases workspeed by 20% when in charge of a task.

Figure 9.8: Skill Menu: A Menu for Assigning Skills

The player can train employees to gain levels in six different types of skills, which can be divided into two groups. The first group is skills that are required by an employee to be able to work on tasks. These are named *software skills*. The second group allows the employee to work faster in some situations, and are named *efficiency skills*. The first are the most important because they allow the player to progress in the game, while the second group improves the efficiency of the team, and allows the player complete projects faster.

Software Skills

There are three software skills - *Coding*, *Integration* and *Testing*. Coding is often required by tasks early in the project. Based on the "software theme" in the game, the player is asked to improve coding skills when creating new modules. For instance - a task with the description "Construct a database that can store content", needs coding skills, because the player is asked to create something new. Furthermore, integration is typically required by tasks that have many dependencies. Put in terms of the "software theme" in the game, the player is often asked to combine two or more software modules. This skill is often required by later tasks in the projects. Lastly, testing is always required by the last task in each project. The player is then asked to test each module in the software that has been created.

Efficiency Skills

The second group of skill include *Focus*, *Teamwork* and *Architecture*. Focus increases work-speed by 20 percent when an employee is a leader. Teamwork increases work-speed by 20 percent when collaborating on a task. The last skill, architecture, increases work-speed by 20 percent when working on quality tasks.

In this section, two groups of skills were described. The player needs to improve the employees' software skills in order to progress through the game, while efficiency skills improve the employees' working speed in three different situations.

9.5 Mood, Stress and Take a Break

While playing the game, the player has to frequently monitor the employees' feelings. One such feeling is mood, which affects how efficiently an employee works. Mood have three different states - *sad*, *indifferent* and *happy*. An employee that is "sad" will work at half the speed, and will not accept new assignments, while a "happy" employee will work twice as fast. Furthermore, the mood can take on a range of values. The value can exist in between several intervals, and this is what determines the state of the mood. The mood value will stabilize at the center, which means it will always try to reach the state of indifferent. For example - If an employee reaches the mood state "happy", the value will slowly decrease until it reaches the state indifferent. The same rule is applied to the mood state "sad", but in this case, the value will increase until it stabilizes at the center. In addition to monitoring the mood, the player can actively affect it by having employees complete tasks.

Another employee feeling is stress. Stress is gained when an employee is assigned a new task. When the stress reaches a certain level, it will begin to affect the employee's mood. The mood will slowly begin to decrease, eventually reaching the "sad" state. The player can avoid this by ordering the employees to take a break. This can be done by choosing "Take a break" from the action menu, see Figure 9.4. This will put the employee in a relaxed state, and the stress levels will slowly begin to decrease.

When playing the game, the player has to watch the mood and stress levels. The monitor panel (Figure 9.9) is the user interface element that allows the player to see the state of mood and stress. Note that the monitor panel does also show the employee's skills. In addition to the monitor panel, the player is notified when employees stress level reaches a high level. This is indicated by an icon over the employee's head. Mood is displayed in a similar way but is only shown when the mood is either "sad" or "happy".

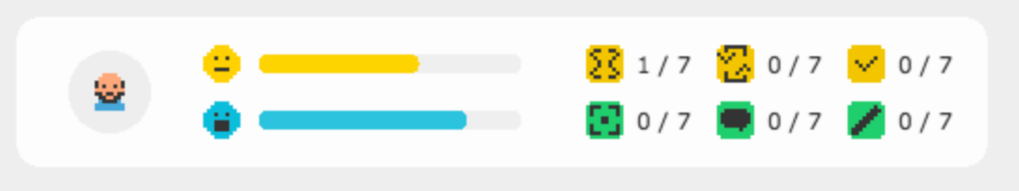
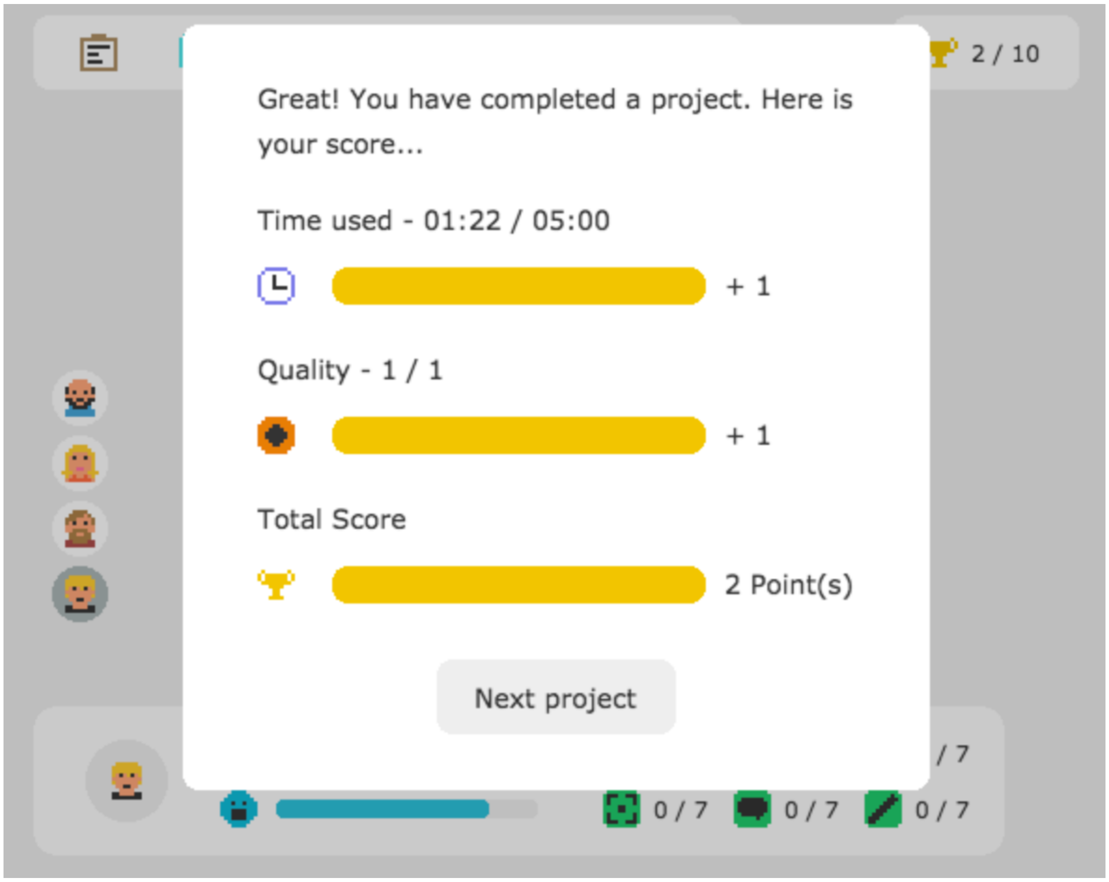


Figure 9.9: Monitor Panel: Displays Information About an Employee

9.6 Completing a Level

After completing all the tasks in the project, but not necessarily the quality tasks, the score screen is displayed, see Figure 9.10. This menu shows how well the player did. One point is received for completing the level before the time limit, and one for completing all the quality tasks. The menu displays the points given to the player, and also a progress bar that gives an indication on how well the player did. The player may get 10 points in total when all the levels are completed. The player is then presented with a final score screen, which displays the total number of points that the player has earned.



Great! You have completed a project. Here is your score...

Time used - 01:22 / 05:00

+ 1

Quality - 1 / 1

+ 1

Total Score

2 Point(s)

Figure 9.10: Score Screen: Presents the Score to the Player

Abilities

Sometimes after a level is completed, the player is asked to choose an ability, which acts as an aid to the player, see Figure 9.11 for the menu that allows the player to choose an ability. The player is asked to choose an ability after completing level 2, level 3 and level 4. After an ability is chosen, a text box is shown, which presents some learning material that at the same time describe how the ability works. Furthermore, an ability can be used while playing the game, and this is done by clicking the ability button. Figure 9.1 shows the abilities at the left side of the screen. When an ability is activated, the player needs to wait for a certain amount of time to be able to use it again. There are three different abilities to choose from. The first is the happiness ability, which increases the happiness of all the employees. Second, the stress ability, which removes the stress from all the employees. Finally, the time ability allows the player to reduce the time spent on a project.

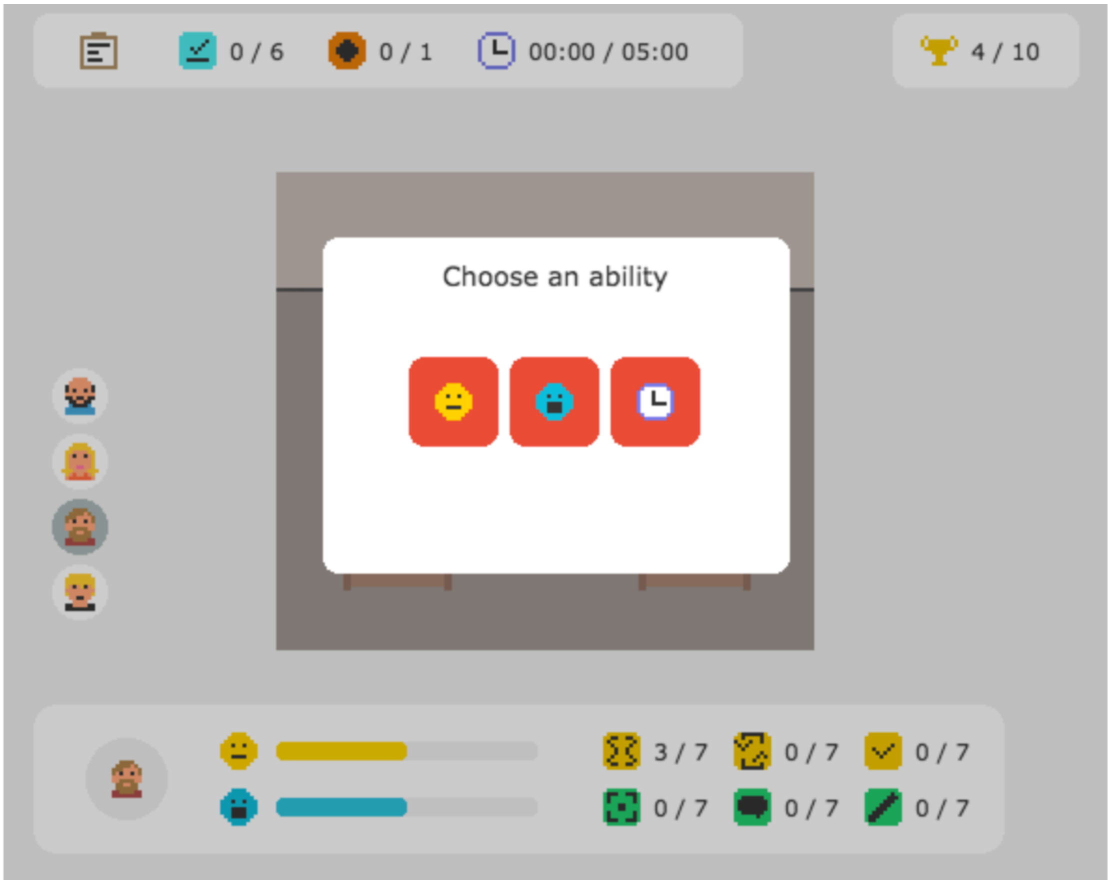


Figure 9.11: Ability Menu: Menu for Choosing Abilities

Summary

In this chapter, we described the game "Project Manager". We began by explaining the overall concept of the game, and some of the rules, before describing the details on how the game is played. The game includes levels, tasks, and skills, where "working on tasks" and "improving skills" are the core mechanics of the game. Furthermore, the player needs to monitor the employees' mood and stress level and make sure that they are not overworked. At the end of every level, the player is shown the score and are sometimes asked to choose an ability that can be helpful in some situations. Lastly, learning material is presented in the game as challenges and textual descriptions, where "the boss" gives useful tips on how to play the game, while sharing information about project management.

Chapter 10

Technology

In this chapter, the main choice of technology will be described, which will contribute to answering *RQ4: What are the available technologies for creating web games ?*

For developing the game, we chose several technologies. One of the major choices where to use HTML5 technologies, and to use Pixi.js as a way to speed up development. HTML5 technologies were chosen over a combination of HTML, CSS and JavaScript for performance reasons. We wanted to create a game more similar to "Game Dev Tycoon" rather than a typical quiz game. Therefore, performance became important, as the game needed to simulate many elements in real-time.

The reason for choosing Pixi.js over a large amount of frameworks available for creating web browser games was that one of the authors had experience with developing games using Flash and ActionScript. Pixi.js is in many ways similar to creating games with ActionScript, which made it easier to learn and use. Furthermore, using Pixi.js provided some features that were both necessary and convenient. For example, Pixi.js includes a convenient way to tint objects. So whenever we wanted to indicate that an object could be clicked on, we could just tint the object or a texture with a color, giving the player feedback. One other feature that should be mentioned is that Pixi.js uses a scene-graph to arrange its objects. This feature is provided in a wide range of game frameworks and is important because it allows the developer to easily position and translate objects in relation to each other.

Two smaller libraries were also used for development. The first, Howler.js was used to include sound in the game (HowlerJs, 2016). This library made it simpler to use the WebAudioAPI and it also provided a fallback mechanism to HTML5 Audio. The second library was Tween.js (TweenJs, 2016). This library provided us with a simple way of creating *tweens* - transformations that are based on mathematical functions that can be used to make objects bounce on the screen or fade in from a predefined position.

One last decision was to use TypeScript rather than plain JavaScript. The main reason for this was that TypeScript allowed us to program with more familiar syntax, having most experience with object-oriented languages such as Java. Another reason is that as TypeScript compiles to JavaScript, it provides new language features that have not yet been supported by the latest browsers. TypeScript is only one such compiler, there exist others which should be considered when choosing the technology. Programming with TypeScript did provide us with many convenient features, such as class-based inheritance. However, we would like to point out one inconvenience. Using additional libraries do require a description file [1], which maps TypeScript to JavaScript. Most libraries provide such a file, but not all. Therefore, this file needs sometimes to be constructed by the developer or else the compiler will produce errors.

Summary

HTML5 technologies, Pixi.js, and Typescript make up the most important choice of technology for this study, while Howler.js and Tween.js provided some convenient features.

[1] https://www.typescriptlang.org/docs/handbook/writing-declaration-files.html

Chapter 11

System Development

In this chapter, we will give a brief description of the game from a system perspective. Oates (2005) states that when using the Design and Creation research strategy one should use an appropriate combination of texts, diagrams, and models to show traceability. Therefore, we will describe our main architectural choice for the game and follow up with an overview and description of the most important classes.

11.1 Software Architecture

It is generally known that there should be a separation of concern between model and view to make the code more modifiable. Creating modifiable code was a priority, as we were going to implement different functionality and change things along the way. The separation of concern between model and view was achieved using the Model-View-Controller (MVC) as the main architectural pattern. We did not follow this pattern in a strict sense, but rather as a way to define the main modules for the system. While developing, we learned that the MVC-pattern can be implemented in different ways, and using it for games and combining it with a library like Pixi.js required some trying out. The implementation chosen for this project was inspired by design patterns head first (Freeman et al., 2004) were Model-View-Controller is defined in terms of three design patterns - observer, composite, and strategy.

Our implementation relied more or less on these three patterns. Figure 11.1 presents our implementation. First, the model implements an observable, so that the view can listen for changes. The model can then notify the view whenever a change occurs. When the user interacts with a view, the view will either handle this interaction itself or delegate the responsibility to a controller. Freeman et al. (2004) uses the strategy pattern when delegating the responsibility to the controller. Both view and controller may update the model. Furthermore, the view contains a list of view components. Freeman et al. (2004) describes that the view should be a composite of components, where the controller only need to update the top view, which updates all its sub-components. As pointed out earlier, our implementation did not follow this definition of MVC in a strict sense. However, it provided us with a way to separate the model and view of the system and made it easier to work two people at a time since we could split some of the responsibility between the modules. Each of these modules will be described further in the following sections.

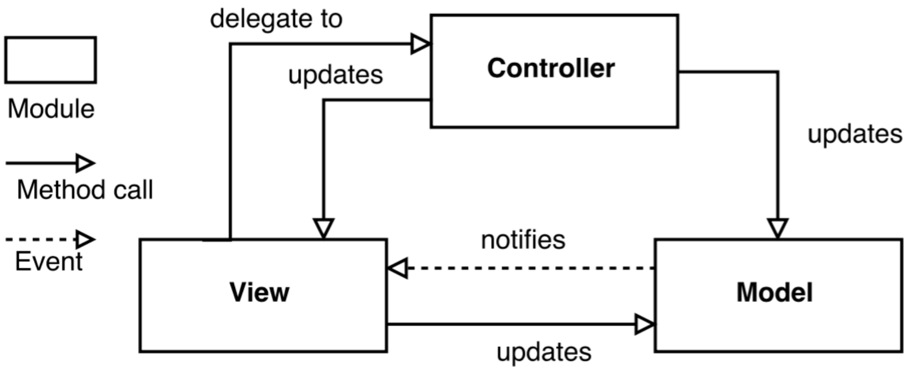


Figure 11.1: Model View Controller

11.2 Model

The model includes the data, game logic, and simulation for the game. Following is a description of the most central classes in the system. Figure 11.2 shows these in a diagram. Note that most of the classes inherit from a class called Observable. Following are descriptions of the most important model classes:

World Is the root of the model. This class is responsible for initiating three important model classes and takes care of communication between them. These classes include Player, Project, and Employee. When the game starts, World will call the service object to get data that is needed to initiate the projects and employees in the game. Furthermore, the world class contains a method update, which is called from the main game loop. World will then forward this call to every object that needs to be updated for every iteration.

Service Is a collection of several small singletons that return data such as levels, employees, and office information. This is data is defined with JavaScript Objects, in separate files. This made it simple to define and change the data used in the game.

Project Defines the levels in the game. This class contains information about a project such as the time limit and a collection of tasks. When a project is chosen, this class becomes responsible for updating the tasks and gives a call back to World if all the tasks are completed.

Employee Contains information about the employee, and is a simple state machine with three different states: "Idle", "Work", "Relax", and "Speak". The state machine is responsible for changing between these states and defines much of the behavior for each.

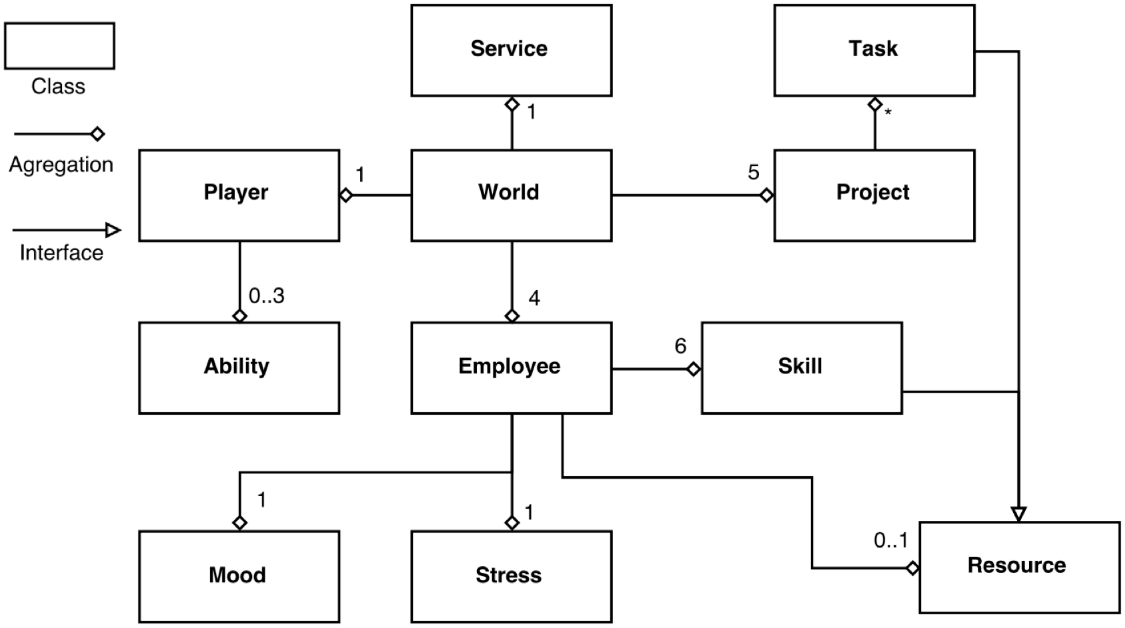


Figure 11.2: System Overview: Class Diagram of the Model

Resource Defines an interface that can be implemented by objects that can be "assigned" and "worked on" by an employee.

Task Implements Resource and contains information such as position, dependencies to other tasks, skill requirements and progress. When an employee is assigned to work on the task, Task becomes responsible for updating its progress, and to notify Project and the employees when the task is completed. Furthermore, a task can be of two different types: "Quality" and "Normal".

Skill Implements Resource and contains information like type of skill, and current level. When assigned to an employee, it updates its progress and changes the level value when progress reaches a threshold.

Mood Takes some of the responsibility from Employee and updates a value representing the employee's mood. As the mood value changes, Mood is responsible for updating its current state to one of three different states: "Sad", "Indifferent" and "Happy". The observers are notified when these state changes occur, signaling the view to update.

Stress Takes some of the responsibility from Employee and updates a value representing the employee's stress. When this value reaches a certain level, the employee becomes stressed, and its observers are notified.

Player Takes some of the responsibility from World, and contains the player's score and initiates and updates the abilities that the player receives throughout the game.

Ability This class is responsible for updating a progress value and change between the states Ready and Loading. An ability change to the loading state when the player uses it. Further, an ability can be of three different types: "HappynessBoost", "StressRelease", "TimeBonus".

11.3 Views and Controllers

The view defines the components that make up the user interface, and the logic to handle user interaction, which is sometimes delegated to a controller. Figure 11.3 shows an overview of

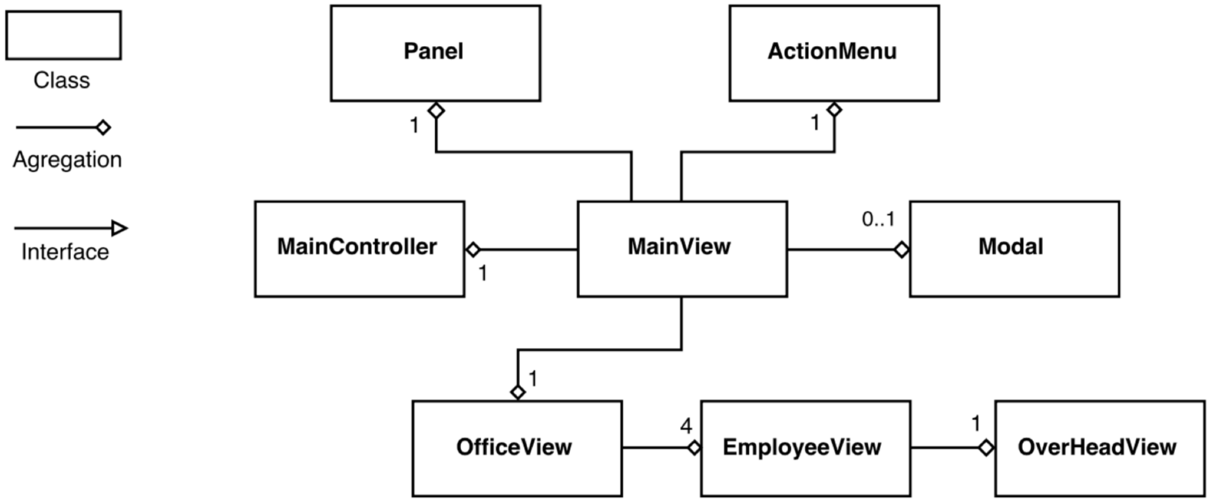


Figure 11.3: System Overview: Class Diagram of the View

the classes found in the view module. The view components inherit from a Pixi.js class called *PIXI.Container*, which makes it possible to add objects to the scenegraph provided by pixi.js. Following are description of the most important view classes:

MainView The root of the view module. This class is responsible for initiating and updating the views. It contains methods for interacting with some of the views.

MainController Takes some of the responsibility of the MainView class. Mainly, it listens for interactions on the ActionMenu and uses the result to update the MainView.

OfficeView Is responsible for initiating the office graphics and the EmplpoyeeView.

EmployeeView Represents the Employee Class from the Model. This class is responsible for displaying graphics, and listens for state updates from the model and uses this information to initiate different types of OverHeadView.

OverHeadView Is a collection of views that appears over an employee's head when it changes state. Different types of OverHeadView includes WorkingView and SpeechBubbleView.

ActionMenu The menu that appears when clicking at an employee. Is responsible initiating the menu elements, and returning the "action" that the player has chosen.

Panel A collection of multiple view classes. A panel is a user interface element that usually contains buttons, and an element to indicate progress, such as a progress bar or simply just text. The different Panel classes in the game are Project Panel, Monitor Panel, AbilityPanel, and Score Panel.

Modal A collection of multiple view classes. Modal is a box with content that appears in the game with a darkened background. This content can either be a menu, or just a simple text description. Modal most often returns a choice that the players have made. The different Modal classes in the game are InformationModal, ProjectModal, TaskModal, SkillModal, AbillityModal.

11.4 Testing

As explained in the "Prototyped Development" Chapter 8, the game was developed over several iterations. One reason for this was to create game mechanics that could lead to interesting gameplay. An equally important reason was that the game's functionally could be tested and improved over iterations. Frequently meetings with the supervisor ensured that new functionality had to be completed and tested regularly. Receiving feedback on the newly added features and changes made, were important for the progress of the game, and allowed to iteratively create and maintain the modules and classes in the system.

Summary

In this chapter, we provided a brief overview of the system, while describing the use of Model-View-Controller as the main choice of architectural pattern. In addition, we gave an outline of the most important classes.

Part IV

Evaluation

This part presents the experiment and the results from the data generation method questionnaire, interview, and observation. The results will also be discussed together with our approach and research questions.

Chapter 12

Experiment

The three data generation methods used in this study were a questionnaire, interviews, and observations. This chapter presents the results and approach used in each method. First, a short overview of the whole experiment will be given. Second, the questionnaire and the questionnaire results will be explained. Then, the interviews and the related results will be described. Lastly, the data from the observations are presented.

12.1 Experiment

A total of 18 people participated in the experiment. All the 18 research subjects responded to the questionnaire, five of them were interviewed and four of them were observed. Two of the subjects interviewed were also observed. The experiment started with the researchers sending an e-mail to the subjects. The e-mail contained a link to the game and to the questionnaire. In the e-mail, the subjects were informed that they were about to play a game that teaches project management. It was deliberately chosen not to include information about how to play the game. This was to ensure that the System Usability Scale produced valid data, and because the GameFlow model encourages games to be understandable and playable by themselves. The observations were done the first time the subjects played the game. The observed subjects were later briefed to confirm or refute our observations. The interviews were conducted after the subject had played the game and answered the questionnaire.

12.2 Questionnaire

This section gives a description of how the questionnaire was designed and how the data from the questionnaire was collected.

Execution

All the subjects who volunteered received an e-mail with a link to the prototype and the questionnaire. The questionnaire was self-administered and anonymous, to remove the influence given by the presence of the researchers (Oates, 2005). To reach more respondents, the questionnaire was accessible online and constructed using Google Forms [1]. The respondents were first encouraged to play the game and then answer the questionnaire shortly thereafter.

Categories

to describe scales used on the on each item and to give background information to the reader, we created 6 categories of items.

Most common scale

The majority of items uses the same scale. Categories two that is related to enjoyment, category three that is related to intrinsically motivating and enjoyable elements, and category five that deals with learning, contains the majority of the items. The scale used in these three categories of items are strongly disagree, disagree, agree and strongly agree. The neutral option was removed to force the respondents to give definite answers. Oates (2005) states that by removing neutral you risk alienating the respondents who actually do not know what to answer. When the results are presented later, these three categories will be presented with disagree and agree.

Explanation of each category

The first category of items produced demographic data , such as gender, if he/she was a student and how many times they played through the game. In addition, we asked them to categorize their gaming habits. The background information is meant to describe the subjects. The

[1] Google Forms - https://www.google.com/intl/no_no/forms/about/

scale, in this case, differs from yes or no answers, male/female, "Does not play games"/casual gamer/experienced gamer/hardcore gamer and a how many times they played the game.

Category number two consists of items that are related to enjoyment. Most of the items are connected to the GameFlow model, which was used as a guide to creating enjoyable elements in the game prototype. Each GameFlow element is connected to flow theory (Csikszentmihalyi, 1990) . Flow is mentioned as an important tool to create motivation in games (Paras, 2005). This section will therefore also be related to flow.

Category three evaluates intrinsically motivating and enjoyable elements. The items in this category are based on several elements such as planning, strategy, fantasy and exploration. Exploration can be related to curiosity, which in addition to graphics, audio and fantasy can be included as intrinsically enjoyable elements (Malone, 1980).

Category four presents one question where the respondents are asked to mark the emotions they felt while playing the game.

Category five encompasses results that concerns learning. Did the participants experience learning? Would this or similar types of games be appropriate in an educational context? It also encompasses items that aim to explore if the game motivated for further learning.

The last category describes the result from the System Usability Scale (SUS) (Brooke et al., 1996). SUS produces a single value when analyzed. Brooke et al. (1996) states that the individual results are not valuable by themselves, all the questions in combination are valuable. SUS uses the Likert scale where each item has the options of strongly disagree, disagree, neutral, agree or Strongly agree. Even though the total score is the most important result when using SUS, we will discuss one of the items in more detail. The reason for this is that the results on this item were evenly distributed on the Likert scale. When analyzing System Usability Scale, the Likert scale is transformed to numerical values ranging from 0 to 4. For question 1,3,5,7 and 9 the score is the row position minus 1. For question 2,4,6,8 and 10 the score is 5 minus the row position. Sum up the 10 transformed results and multiply them by 2.5 to get the SUS score for each respondent. This score is a number between 0 and 100. Higher scores suggest a better usability. The average score of all the respondents is used to rate and compare the usability of the system.

12.3 Result from Questionnaire

This section presents the responses from the questionnaire. All of the items are given an introduction and a short explanation. To make it easier for the reader, most of the results are simplified from strongly disagree, disagree, agree and strongly agree to agree and disagree. This will show the trends more clearly and give a more tidy presentation. The following categories are simplified: "Enjoyment and Flow" Table 12.3 , "Intrinsically Enjoyable Elements" Table 12.4 and "Learning" Table 12.6. The results with the percentages and the whole scale can be found in Appendix A.

Demographics

The population in this study is fairly homogeneous. 17 of the 18 respondents are males. The age of the respondents is in the range of 22 to 28 years old. 16 of the 18 respondents were students.

Table 12.1: Gaming Habits

	Does not	Casual	Experienced	Hardcore
Statement	play games	gamer	gamer	gamer
Which of these categories suites your gaming habits best?	5.6%	33.3%	44.4%	16.7%

Which of these categories suites your gaming habits best?

This item investigates how the player views their own gaming habits. The space in the interval between each option is not equal. The important thing is what the respondents in their own consideration classifies themselves as. The question is meant to investigate if experienced gamers and hardcore gamers respond in a different way than the players who is categorized as “does not play games” and “casually play games”. As we can see in Table 12.1 , 5.6% was classified as “does not play games”, 33.3% was classified as “casually play games”, 44.4% as experienced gamers and 16.7% as hardcore gamers.

Table 12.2: Playthrough

Statement	I did not play through the whole game	1	2	3	4	More
How many times did you play the game?	5.6%	55.6%	22.2%	11.1%	5.6%	0%

How many times did you play the game?

This item is included to investigate if the respondents played through the game. Table 12.2 shows that only 5.6% did not play through the whole game and 55.6% played it one time and 38.9% played it two times or more. Only the respondents who answered that they did not play through the whole game, was asked to explain why. The one person in this category answered that “ it was difficult to do concurrent tasks”.

Table 12.3: Enjoyment and flow

Statement	Disagree	Agree
I enjoyed playing the game	0%	100%
I was concentrated while playing the game	5.6%	94.5%
The challenges presented in the game matched my skill level	27.8%	72.2%
The objectives in the game were easy to understand	16.7%	83.4%
The game gave feedback that helped me reach the goal	33.3%	66.7%
I was in control of my actions while playing the game	5.6%	94.4%
I lost awareness of my surroundings while playing the game	27.8%	72.2%

Enjoyment

I enjoyed playing the game

When responding to "I enjoyed playing the game", 100% of the respondents agreed, see Table 12.3. This shows that the respondents enjoyed playing the game.

I was concentrated while playing the game

This item explores the GameFlow element of a subject being able to concentrate on the task. It is necessary to achieve a sensation of flow (Sweetser and Wyeth, 2005). In Table 12.3 it is presented that 94% agreed and 5.6% disagreed, which suggests that the game successfully managed to create concentration among the respondents.

The challenges presented in the game matched my skill level

The item that addresses if the challenges match the player skill level is presented in Table 12.3. 27% disagreed and 72.2% agreed to this item. The results indicate that the balancing of skills was appropriately generalized to meet different subjects core skill and introducing new challenges at an appropriate pace. Which was one of the focus areas in the design and creation phase of this study. Matching the skill and difficulty level is also important to achieve flow (Sweetser and Wyeth, 2005).

The objectives in the game were easy to understand

The flow theory states that the objectives should be clear and easy to understand (Sweetser and Wyeth, 2005). The results from Table 12.3 shows that 16.7% disagreed and 83.4% agreed to this item. This indicates that the objectives were easy to understand.

The game gave feedback that helped me reach the goal

This item is described in Table 12.3. The GameFlow model states that feedback should guide the user towards the goal (Sweetser and Wyeth, 2005). 33.3% disagreed and 66.7% agreed. The majority of the subjects agreed to this statement.

I was in control of my actions while playing the game

Control is an important factor to achieve flow (Sweetser and Wyeth, 2005). Table 12.3 presents that 5.6% disagreed and 94.4 agreed that they were in control of their actions while playing the game. This indicates that we successfully gave the subjects a sense of control over their actions.

I lost awareness of my surroundings while playing the game

To loose awareness of the surroundings is an indication of immersion, which is described as an important part of experiencing flow (Sweetser and Wyeth, 2005). As shown in Table 12.3 , 27.8% disagreed and 72.2% agreed with the statement. The trend is that players lost awareness of their surroundings.

Except from the statement "I enjoyed playing the game", the items were related to the GameFlow model. The majority of the respondents answered agree or strongly agree to the flow related statements. This is an indication that flow was experienced by some of the players. 100% agreed or strongly agreed to that they enjoyed the game. This shows that all of the users enjoyed the game, even if they did not experience flow.

Table 12.4: Intrinsically Enjoyable Elements

Statement	Disagree	Agree
I enjoyed the fantasy-world provided by the game	33.3%	66.6%
I was motivated to explore the content in the game to perform better	11.1%	88.9%
I could perform better in the game by planning my actions	5.6%	94.5%
Using strategy was an important part of the game	5.6%	94.5%
I was motivated to improve my score	27.8%	72.2%
The audio in the game enhanced my experience positively	22.3%	77.8%
The graphics and animations enhanced my experience positively	11.1%	88.9%

Intrinsically Enjoyable Elements

I enjoyed the fantasy-world provided by the game

This item was included to answer if the fantasy environment in the game appealed to the subjects. Table 12.4 shows that 66.6% agreed and 33.3% disagreed to this item. Malone (1980) suggest creating different fantasies for different players because of the variation in what kind of fantasies people like. Even though we did not include different fantasies in the game, the result indicates that the majority of the respondents liked the fantasy provided by the game.

I was motivated to explore the content in the game to perform better

This item aims to answer if our focus on exploration when creating the prototype, had the effect we wanted. In Table 12.4 it is presented that 11.1% disagreed and 88.9% agreed to this item, indicating that the game managed to interest the player in exploring.

I could perform better in the game by planning my actions

The result from this item is presented in Table 12.4 , and is aimed at investigating if we encouraged the respondents to create plans to perform better. This was a goal when trying to make the players reflect upon their actions and creating plans. 5.6% disagreed and 94.5% agreed to this statement, giving an indication that we positively encourage the use of plans to perform better.

Using strategy was an important part of the game

This item partially overlaps with the previous item, but it is more directly aimed at strategy. Table 12.4 shows that 5.6% disagreed to this statement and 94.5% agreed, indicating that we were successful.

I was motivated to improve my score

The aim of this item was to investigate if the score was a motivating factor in a game where there was no direct way of competing with other players. 27.8% disagreed and 72.2% agreed (see Table 12.4) . This suggests that score is important to players of a game and can be used as a motivating factor in learning games without direct competition.

The audio in the game enhanced my experience positively

22.2% disagreed and 77.8% agreed to this item (see Table 12.4). Indicating that the audio enhanced the respondents experience positively.

The graphics and animations enhanced my experience positively

Table 12.4 shows that 11.1% disagreed and 88.9% agreed that the graphics and animations enhanced their experience positively. Indicating that the graphics and animations had a positive effect on the respondents.

The results show that encouraging the players to create plans and strategies was successful, the graphics and animations had a positive effect on the majority of the subjects and that most of the subjects were motivated to explore the content presented in the game and to improve their score and the two-thirds enjoyed the fantasy world provided by the game.

Table 12.5: Emotions

					A sense of
Statement	Boredom	Frustration	Confusion	Excitement	achievement
Mark the emotions you felt when playing the game	5.6%	11.1%	38.9%	77.8%	77.8%

Emotions

Mark the emotions you felt when playing the game

This item differs from the other items in that it allows the respondents to mark more than one answer. Table 12.5 shows that 5.6% of the respondents answered that they felt boredom, 11.1% answered that they felt frustration, 38.9% answered that they felt confusion. Boredom and frustration are the two extremes of balancing challenge and skill, too little challenge create boredom and too much challenge create frustration (Paras, 2005). Even though challenge can be interpreted this way, the frustration and boredom could be caused by other things, but the lack of these emotions have some meaning. 38.9% answered that they felt confusion. This could have many explanations and no conclusion can be drawn from this result. 77.8% felt excitement, and the same amount answered that they felt a sense of achievement. The excitement is in this study more of a positive emotion. The sense of achievement is related to the experience of mastering something, which Malone (1980) describes as an important part of enjoyable learning games.

Table 12.6: Learning

Statement	Disagree	Agree
I would like to use this or a similar type of game in an educational context	22.3%	77.8%
I got an increased understanding of project management through playing the game	33.3%	66.7%
I learned something while playing the game	44.5%	55.6%
I would like to learn more about the content presented in the game	27.8%	72.2%
I was motivated to click on objects to get information	22.3%	77.7%
I was motivated to learn by playing the game	16.7%	83.4%

Learning

I would like to use this or a similar type of game in an educational context

22.3% disagreed and 77.8% agreed with this statement, see Table 12.6. This indicates that similar type of games or this game could be desirable in an educational context.

I got an increased understanding of project management through playing the game

33.3% disagreed and 66.7% when responding to this item, see Table 12.6. Stating that the majority of the respondents got an increased understanding project management through playing the game.

I learned something while playing the game

44.5% disagreed and 55.6% to this item, see Table 12.6. This shows that half the respondents learned something while playing the game. The achieved learning will be discussed in more detail in the the next chapter, where we will try to find a reason why 44.5% disagreed with this statement.

I would like to learn more about the content presented in the game

27.8% disagreed to this statement and 72.2% agreed, see Table 12.6. This indicates that the game had some effect on the motivation to learn more about the content presented in the game.

I was motivated to click on objects to get information

22.3% disagreed and 77.7% agreed that they were motivated to click on objects to get information, see Table 12.6. This is one of the strategies used to convey the learning material to the user. Were some of the learning material was accessible by clicking on buttons in the game environment. The trend is that the subjects were motivated click on objects to get more information.

I was motivated to learn by playing the game

This item is related to the respondent's motivation to learn and not the learning outcome. 16.7% disagreed and 83.4% agreed to the statement, see Table 12.6. This indicates that the game motivated the respondents to learn.

Table 12.7: Symbols

Statement	Disagree	Agree
I found it easy to understand and remember the meaning of the symbols in the game.	33.3%	66.7%

Usability

I found it easy to understand and remember the meaning of the symbols in the game

This item is important because of the usability in the game. It was not directly covered by System Usability Scale but was important because we used symbols as representations instead of word and sentences, to more efficiently convey information to the user. 33.3% disagreed and 66.7% agreed to this item, see Table 12.7.

Table 12.8: System Usability Scale

Statement	Strongly Disagree	Disagree	neutral	Agree	Strongly Agree
I think that I would like to use this system frequently	11.1 %	27.8 %	22.2 %	33.3 %	5.6
I found the system unnecessarily complex	44.4 %	38.9 %	16.7 %	0 %	0
I thought the system was easy to use	0 %	0 %	22.2 %	50 %	27.8
I think that I would need the support of a technical person to be able to use this system	77.8 %	22.2 %	0 %	0 %	0
I found the various functions in this system were well integrated	0 %	0 %	22.2 %	50 %	27.8
I thought there was too much inconsistency in this system	66.7 %	27.8 %	5.6 %	0 %	0
I would imagine that most people would learn to use this system very quickly	0 %	0 %	27.8 %	44.4 %	27.8
I found the system very cumbersome to use	38.9 %	38.9 %	22.2 %	0 %	0
I felt very confident using the system	0 %	0 %	22.2 %	50 %	27.8
I needed to learn a lot of things before I could get going with this system	33.3 %	50 %	16.7 %	0 %	0

System Usability Scale

The System Usability Scale (SUS) is used as a way to evaluate the usability. The result on each item is presented in Table 12.8. SUS is a broad test for a system usability in general. Even though it is not the most usual evaluation method for games, it creates a score that gives a general impression of the usability of the system, that can be compared to other systems. Bangor et al. (2008) did a study that includes 2324 surveys using SUS with an average score of 70.14. In this study (Bangor et al., 2008) conclude with that "products which are at least passable have SUS scores above 70, with better products scoring in the high 70s to upper 80s". The System Usability Scale gave us an average score of 77.7 with 95% confidence interval between 71.2 and 83.3. Meaning that the score is in the high 70s, a confidence interval within the passable limit and above the mean of that study.

All the items had quite similar trends except from the item “I think that I would like to use this system frequently”. This item diverged from the others in that the responses were spread all over the scale. This was not unexpected, due to this game not being designed for frequent use by the same player over time. To get a better SUS score, the responses to this item would have been in the agree end of the Likert scale. On item 2,4,6,8 and 10 all the responses are placed on strongly disagree, disagree or neutral. On question 3, 5, 7 and 9 all the items are placed neutral, agree or strongly agree.

12.4 Interview

This section presents the results and the approach used when conducting the interviews.

The interviews were carried out shortly after the subjects had completed the game and answered the questionnaire. The chosen subjects were interviewed one at the time with no one else present in the room. The lunchrooms on the second and third floor at the IT-Department at NTNU, was chosen as locations for the interviews. These locations were chosen because of the comfortable sitting opportunities and low noise level.

Before the interviews, we created a list of guidelines and rules for good interview practices that were followed during the interviews. The guidelines and rules were gathered from "Researching Information Systems and Computing"(Oates, 2005). The interviewees were informed that they were recorded during the interview. For ethical reasons, they would have to give their consent. A short presentation of the research and the purpose of the interview was given to prepare them. The predefined questions were asked in the same way each time, to avoid the interviewer imposing their meanings on the subjects. Follow-up questions were allowed, because it was a semi-structured interview, with the purpose of investigating “interesting” topics and get more information about what the subjects really thought about the game. Other guidelines were also used, such as preparing some small talk to make the interviewee feel comfortable before the interview started, testing and preparing all technical equipment, making sure that the interviewee did not have to wait for the researchers, and sitting 90 degrees from the interviewee to appear non-threatening.

It was chosen to record all the interviews to make sure that no information was lost. All of the interviews were transcribed. When transcribing we tried to keep the interview as intact as possible. This was important so that the interviewees' meanings did not change (Oates, 2005). It was allowed to remove redundant sounds and to create complete sentences if it did not alter the meaning of the answer.

Results from the Interviews

This section presents the results from the interviews, where five students were participating. All the interviews were conducted in Norwegian and is translated into English. The content of each interview is shortened and only the relevant information is presented. The answers are also rewritten into a third person point of view, for readability reasons. Only one of the interviewees stated he had no experience in the field of project management before playing the game. The rest had some knowledge about the subject. One of the interviewees had some experience in the subject of creating games and had written a bachelor's thesis about it.

What did you enjoy most about the game ?

When asked what they enjoy most about the game, two of the interviewees answered that they enjoyed creating their own strategies. One liked the intensity of the game and that he always had something to do. Another mentioned the user interface and how easy it was to start playing the game. The last interviewee liked the feeling of flow that is usual in simulation games, and that simple tasks become fun in the context they were presented.

What did you like the least about the game ?

When asked what they liked the least about the game, one answered that he thought the game was too short. Another answered that he thought the game should be more difficult and emphasized that he liked games in that genre to be more difficult. Two of the other interviewees also answered that they thought the game was too easy and that the level of difficulty did not increase enough. Indicating that difficulty could increase more over time. The last one could not remember anything he disliked.

Did you make any strategy while playing the game ?

All of the interviewees used strategies when playing the game. All the strategies were focused on how to structure the project team, and how they delegated skills to tackle this project and the next. As an example, one of the interviewees chose to focus on delegating the skills among the workers to create a team that could tackle the projects. He also focused on one of them having good cooperating skills, to complete the project within the deadline with high quality. Two of the interviewees said they put the skills at random in the first project. But changed their behavior in project number two where they started planning what skills were needed on the tasks and delegating the skills among the workers. The reason one of these interviewees changed strategy, was because he understood that he had a limited amount of time. He then delegated the skills to the employees before he started working on the tasks. Another chose to specialize two employees as leaders and two that had a more general expertise and good cooperation skills.

Did you read the information presented to you in the game?

When asked if they read the information presented to them, all of the interviewees answered that they did. One of them stated that he thought the information was brief and clear and gave him the information he needed. Another answered he read the information presented between the project but did not read the descriptions during the game. He explained that it was because of the time pressure and that he felt he could not use time on reading the texts. The two remaining interviewees answered that they skimmed through the texts. One of them answered he read it in the beginning of the game but read the last texts hastily towards the end of the game. The other one just answered he skimmed through the texts.

Did you learn something while playing the game?

When asked if they learned something while playing the game, one of them said he learned that a team should consist of different people with different skills. He also stated that one employee cannot do the work of four and that a good working environment can increase the productivity. He also mentioned that good cooperation skills were essential. Another interviewee stated he learned to focus on the skill development of the team members, so they could do tasks. He also learned one should not overwork the employees. Two of the interviewees answered that they knew much of the learning content in advance of them playing the game, so they did not learn that much. One of them also said that he got an increased understanding of how difficult it could be to coordinate a project team. The last of them answered that he did not learn much, but that he instead choose to focus mainly on the game mechanics.

What did you think about the graphics and sounds?

When asked what they thought about the graphics and sounds, one of them answered that the graphics were simple, complete and nice. He thought there was good quality on the sounds and that they were comfortable to listen to. Two of the interviewees answered he thought that the graphics were "cool". Three of interviewees mentioned that they liked that the graphics were "old school". One of the interviewees suggested that we should add a mute button and add music. The last one liked the graphic and thought they were very fitting for that genre.

What is your thoughts about the usability in the game?

One of the interviewees said he had no problems with the user interface. A different person answered that it was easy to learn and that it was informative. Another stated that the user interface was intuitive and that "things" were intuitively placed. One mentioned that it was a little cumbersome to click on the workers all the time, but the feedback was good. He also had some problems with getting an overview of the worker's status, and that it should be given more feedback on the stress level of the workers. The last one felt that the resolution on the symbols was a little too low and that the symbols should be more distinguishable from one another. He had problems seeing the difference between the happiness symbol and stress symbol.

Do you have any further comments that you would like to express regarding the game?

When asked if they had any further comments. One interviewee answered that the difficulty level should be raised. Another thought that the concept was "cool" and it was fun to use strategies. He also wanted a little more in depth learning material. One suggested that in further development, we could make the workers seem more personal by letting the user give them names and give them a background story. He also suggested giving more background information on each project to prepare the player.

12.5 Observation

This section presents the result of the observations. The participants played through the whole game one time, while being observed by the researchers. After the observations were completed, the participants were asked questions about unclear observations and the meaning behind some actions chosen by the researchers. Four participants were observed.

Results from the observations

The results from the observations have been arranged in the following topics:

Score Two of the participants got a score of 9 out of 10 points and one got 10 out of 10. When asked if they focused on getting a high score, these three participants all answered that they did. The last participant got 7 points, he later said it was because he did not care about the quality of the projects, and described himself as an achiever, who just got things done.

Unexpected behavior One of the participants resized the game to get it bigger. One of the participants did not use any of the abilities. Later when asked why he did not use them, he answered that he thought that they were permanent abilities and that they were active all the time.

Texts The observations showed that there was a difference in how and if the participants read the texts when playing. The participant who got 10 out of 10 points on the score, read all the information carefully. When asked later if he read all the information carefully, he answered that he did, and that was a part of the way he played games in general, where he investigated his options to perform better. He also stated that he read texts in more detail because it was a learning game. One of the other participants read the texts between the projects carefully. He later confirmed that he did. But when learning material was presented during a project, he immediately removed the information. Later when he was asked about why he removed that information, he answered that he did not prioritize it because of the limited amount of time and that he read the texts between the project because he knew that the game was paused. Two of the participants used a short amount of time when reading the information. One of them explained that he skimmed through the texts and seldom read information presented in games. He also stated that he used to test if things worked or not. The last one just confirmed that he skimmed through the texts.

Time All the participants managed to finish all of the projects before the deadlines. Two of the participants explained that they could use the time the had left to improve the skills of the employees, while two other employees just finished the projects by themselves.

Strategies and plans All of the participants delegated different skills to different workers and all of them had at least one worker that had good cooperation skills, and two with some kind of specialization. In this setting specialization means that one worker had at least two or more points in one skill than the other workers. As an example, one of the participants explained that he used two workers as leaders, were one of them was the best at coding and the other one was best at integration. The two other had the role of being "general support" in all skills. The leaders had a good focus and the "general support" workers had good cooperation skills. All of the participants confirmed that they had a plan on how to delegated skills among the workers. One of the participants confirmed that he tried to delegate skills before he started working on tasks. At the start of each project he seemingly investigated each task two times to see what skills he needed, and then upgraded the workers before starting to work on the tasks. He later confirmed that he began to delegate skills and planned how to tackle each project.

Investigation Three of the participants placed the mouse cursor over all of the buttons during the first project. And seemingly read the tooltip for each button. One of the participants confirmed that he read all the tooltips at the start of the game, the other two did not remember if they did. The last of the participants said he placed the mouse cursor over buttons when he felt it was necessary.

Concentration The player that gained 10 out of 10 points was seemingly focused on the whole game. He seldom looked away from the screen. As he progressed in each project, his head was moving closer and closer to the screen. When each project was finished he leaned back and read the information. for the three other participants, the focus increased as the game progressed. They started off by being a little-unfocused locking away from the screen and sometimes at the observers. They gradually became more focused on project number two and outwards, looking less away from the screen. One of the participants explained that he became very focused when he understood the game and that there was always something to do.

Summary

This chapter presented the results from the data generation methods and the approaches used. The most important results from the questionnaire were related to enjoyment and flow, intrinsically enjoyable elements, learning, and the system usability scale. The enjoyment and flow category showed that all the respondents agreed or more to that the game was enjoyable. The Game Flow elements also got good results. The most important results from the category containing intrinsically enjoyable elements showed that planning actions using strategies were an important part of the game. Most respondents felt that the audio, graphics and animations enhanced their experience positively. Two-thirds enjoyed the fantasy world presented in the game. The category describing learning showed that most of the participants were motivated to learn, to learn more about the content presented and to click on objects to get more information. It also showed that 55.6 % agreed they learned something while playing the game and that got an that two-thirds got an increased understanding of project management. 77.8 % agreed or more that they would like to use our game or a similar type of game in an educational context. Our score on the System Usability scale was 77.7 which according to Bangor et al. (2008), is an above average score, over the passable limit at 70 and in the range of better products. The most important results from the interviews were that all the interviewees used strategies and plans and could describe them in detail. And that some of them felt creating strategies was the most enjoyable thing in the game. We also found that some already knew the learning material before playing the game and that few of the interviewees, thoroughly read the texts presented in the game. Another interesting point was that the most of the interviewees felt the challenge level was too low. The most impotent results from the observations were that the players was concentrated, and become more concentrated as the game progressed. We could also see the participants use strategies in the game. All the participants finished the projects before the time limit, which is related to how we use time pressure to create challenge. The observations also showed that the texts were not thoroughly read, as just one read them in detail.

Chapter 13

Discussion

In this chapter, we will discuss the findings from the results of the evaluation. Based on the goal of this study and the research questions, these findings have been divided into the following topics: *enjoyment and GameFlow, learning, game mechanics, and strategy* and *interface and usability.*

13.1 Enjoyment and GameFlow Elements

This section discusses the result directly related to enjoyment and gives an overview of how enjoyment is designed into the game using the GameFlow model. It also describes how the GameFlow elements affected the players enjoyment through playing the game. Concentration, Challenge and Player Skills, goal, feedback, and immersion will be discussed in more detail.

13.1.1 Enjoyment

According to (Sweetser and Wyeth, 2005), "player enjoyment is the single most important goal of computer games". This inspired us to include the RQ *How does the game affect the player's enjoyment?* When analyzing this question it was interesting to ask if the subjects enjoyed playing the game. In the questionnaire 100 % of the respondents agreed or strongly agreed that they enjoyed the game. All of the interviewees also stated that they felt enjoyment. This shows that the activity of playing the game was enjoyable for players. 38.9 % played the game more than one time, indicating that some of the respondents were motivated to play it more than once.

Malone (1980) focuses on enjoyment when creating intrinsically enjoyable elements to motivate the learner. This study also uses enjoyment as the main tool to create motivation. To support enjoyment in the game, we focus on elements that in itself was enjoyable for the player.

13.1.2 GameFlow Elements

The GameFlow model was used to as a guide to building enjoyment into the game and attempt to give the players an experience of flow. The model consists of 8 elements. This study uses heuristics from seven of these elements. Social interaction was not included and is the only element not directly linked to the experience of flow (Sweetser and Wyeth, 2005). In this study, the challenge and player skill are described under the element of challenge, because the two elements are overlapping and often described together (Paras, 2005). The six elements discussed in this section are, challenge and player skills, goals, concentration, control, feedback and immersion and each of them have a question dedicated to them in the questionnaire. The questions were added to see if each element affected the players enjoyment, to analyze if the subject experienced flow and which of the elements were successfully implemented in the game.

In the questionnaire, the majority of the respondents agreed or strongly agreed to all of the statements related to the six GameFlow elements. This indicates the elements were successfully implemented and affecting the subjects' enjoyment. During the interviews, John mentioned he liked the feeling of flow in the game, that is usual in simulation games. Directly stating that he experienced flow. As we could not assume that the subjects knew what flow was, we could not directly ask if they felt it. We decompose flow through the GameFlow model when presenting it in the questionnaire.

Concentration

With 5.6 % disagreeing or strongly disagreeing, and 94.4 % agreeing or strongly agreeing, the item investigating concentration, had the highest amount of respondents agreeing of the GameFlow elements. Concentration was the most clearly visible element in the observations. All the participants were concentrated and confirmed that later when asked about it. As the game progressed it was noticeable that the participants eyes were looking more and more at the screen and less away. A high workload can absorb the players attention and create concentration Paras

(2005). Through the iterations in the design and development process, the researchers tried to balance the workload and time pressure to create concentration. The workload was increased as the game progressed to challenge the player's cognitive, perceptive and memory capabilities. Paras (2005) explains that user should not be burdened with tasks that do not feel important. To remove irrelevant tasks, the workload was all relevant to accomplishing the goal of the game or to navigate from project to project. Paras (2005) states that the game should give the player stimuli from different sources. To stimulate the player's attention sounds and animations were implemented in the game.

The players enjoyment is affected by the game through the ability to concentrate on the game, the workload being high enough to absorb the players attention over time, the stimuli created by the sounds and animations and the tasks being relevant to accomplish the goal.

Challenge and Player Skills

In the questionnaire 27% disagreed and 44.4% agreed and 27.8% strongly agreed to the item stating, "The challenges presented in the game matched my skill level". Even though 72.2% agreeing or strongly agreeing, three of the interviewees stated that they thought the game was too easy. Paras (2005) states that a game could create frustration when being too difficult, and create boredom if too simple. The result from the questionnaire shows that 5.6% felt boredom and 11.1% felt frustration while playing the game. If the skill level and the challenges were matched very poorly, one could argue that one of these two emotions would be more frequent. The player had to use skills to complete a project before the deadline. These skills could be obtained through playing the game or skills that the player already possessed. Sweetser and Wyeth (2005) states that an enjoyable game must support the player's skill development and skill mastery. 77.8% of the questionnaire respondents answered that they felt a sense of achievement when playing the game. This indicates that respondents mastered some part of the game and felt a sense of achievement. The development of skills started at the first in-game project. In this project, the difficulty was relatively low and information about the core skills was presented. Before each mission texts was introduced guide and helped the user understand how to master the game. To increase the challenge over time, we introduced more tasks into the projects as the game progressed. These tasks also required more skill upgrades from the employees. By increasing the challenge the researchers tried to force the player to perfect their skills and strategy. Some new skills were introduced as some new abilities became accessible to the player during the game. Those abilities had to be used at the right time to be the most effective.

The challenge affected the players enjoyment through balancing the challenge and skill, by learning new skills through playing the game, by introducing more challenge and new skills over time and creating a sense of achievement. From the interviews, we saw that the challenge level was a little too low for the players.

Goal

The overarching goal of the game was to complete each project within the time limit with the given quality. If the player achieves this they could get the maximum score. A total of 10 points was possible. In the questionnaire, we choose to use the word objective instead of the goal for readability reasons. The questionnaire shows that 16.7% disagreed or strongly disagreed and 83.4% agreed that the objective in the game was easy to understand. The overarching goal of the game was presented to the player in the first project. The guidance towards the goal will be described under feedback.

The Goal affected the players enjoyment by helping create immersion through being presented early and being simple.

Feedback

When describing feedback, the GameFlow model emphasis on two things. That the feedback should be immediate and second, and that the feedback should guide the player towards the goal. The immediate feedback is described in section 13.4. The results show 33.3 % disagreed or strongly disagreed and 66.7% agreed or strongly agreed, that the game gave feedback that helped them reach the goal, This shows that most of the respondents felt the feedback guided them towards the goal. The feedback giving guidance towards the goal was presented to the player in a menu that showed the total overview of the project and the tasks related to it. The tasks status was visible through color codes that showed if a task was completed, a padlock symbol on tasks that could not be started because they were dependent on other tasks and the relation between the tasks. In addition, the quality tasks were presented. When all the tasks were completed the project was finished. The progress of each task or quality was visible over the employees, that was assigned to it. The time remaining was always visible, either in the background or in the top of the browser window. In addition the feedback was given on the achievements in the score screen that was visible in between the projects. The score was visible all the time.

The feedback affected the players enjoyment through giving immediate feedback on their actions allowing concentration, giving guidance towards the goal through a menu that gave the most important information on their progress towards the goal, and progression on tasks and score being visible all the time.

Control

In the questionnaire, 5.6% disagreed or strongly disagreed and 94.4% agreed or strongly agreed, that they were in control of their actions while playing the game. This item is directed at the element of GameFlow who states that the players need to be able to feel a sense of control over their actions, to be able to achieve flow (Paras, 2005). Control is built into the game in many ways. The feedback on the player's action described earlier is one of them. Another is the encouragement of using strategy discussed in the strategy. Paras (2005) states that the player should be free to play the game the way he wants to and be able to create their own strategies. Another important statement presented by (Paras, 2005) , is that errors should not be detrimental for the players progress in the game. This was addressed by the researcher testing the game in each iteration. Only one respondent to the questionnaire stated that he had problems with bugs, as the game crashed at the last level in this case.

The players enjoyment is affected by being allowed to feel a sense of control over in-game actions, through immediate feedback, being able to create his own strategies and the amount of bugs being low due to testing in each iteration.

Immersion

This is the ultimate goal of the game flow model. And is often a product of all the other elements described in the experience of flow (Paras, 2005) . The item that states "I lost awareness of my surroundings while playing the game", is directed at immersion. 27.8% disagreed or strongly disagreed and 72.2% agreed or strongly agreed to this item. Lose awareness of the surroundings is often used to describe immersion and is according to Paras (2005), often observed when the players are feeling immersed in a task. An indication of this behavior is that all of the participants in the observation looked more and more at the screen and less away from it. Immersion can also be induced through the use of emotional stimuli. An attempt to evoke some emotions was done through giving the employees emotional states. Stress and happiness level was the chosen emotions.

Immersion affects the players happiness by helping them be less aware of their surroundings giving them a break from reality through, emotional stimuli and all the other GameFlow elements described earlier.

13.2 Game Mechanics and Strategy

As a part of the game design process, we tried to create game mechanics that could be challenging and encourage the player to plan their actions. 94,5% agreed or more that they could perform better by planning their actions, and 94,5% agreed or more to that using strategy was an important part of the game. This shows that the subjects felt that strategy was an important part of the game. When asked what that they thought was most enjoyable two of the interviewees said that they enjoyed creating their own strategies one of the others liked the feeling of flow that is usual in simulation games. When asked about strategies, all of the interviewees could explain in detail how they used strategies and how they planned their projects and how different employees had different responsibilities. This shows that strategy is an important part of the gameplay. To find out what game mechanics that were important in relation to strategy, we asked the participants to explain their what strategy they had used:

- All participants said that they specialized some of the employees with different skills. The "improve skill" action, and the game model that included task requirements and skill levels is what made this possible.
- Most of the participants specialized an employee collaborator. Meaning that some players focused on improving the collaboration skill for one or more employee so that these employees could be used to improve the working efficiency on a task. In addition, some

specialized the employees as leaders. Meaning that some players focused on improving the leader skill. The action "work on task" and the game model that allowed an to either "collaborate on employee to either "collaborate on a task" or "be a leader of a task" made this possible.

- Most of the participants had a dedicated employee to work on quality tasks. The game model included a special kind of task called "quality task". The game model included "architecture" as a skill that could improve the efficiency working on a quality task.
- Some of the participants stated that they planned so that the employees did not get overworked. Meaning that the players would watch the stress level and plan when to take breaks. However, it was also said that this was not necessary when the stress ability was chosen. The action "take a break" and the game model "mood" and "stress" made this possible.

13.3 Learning

This section discusses the players learning and the related results from the experiment. It starts with presenting the motivation felt by the subjects, then the learning outcome is discussed and finally a discussion on the potential of strategy and learning, based on the results.

Motivation

22.3% disagreed or disagreed strongly and 77.8% agreed or agreed strongly to the item stating, “I would like to use this or a similar type of game in an educational context”. This shows that the majority of the subjects thought that this type of game could be used in an educational context. 27.8% strongly disagreed or disagreed and 72.2% agreed or strongly agreed that they would like to learn more about the content presented in the game. 16.7% disagreed or strongly disagreed and 83.4% agreed or strongly agreed to that they were motivated to learn by playing the game. The trend shows that the majority of the subjects was motivated to learn by playing the game. As explained earlier the main mean to create motivation is through the use of enjoyment. These results indicates that we were successful in creating motivation to learn through enjoyment.

Fantasy

The fantasy environment described in the game is software project environment where the player is a newly hired project manager. 33.3 % disagreed or disagreed strongly and 66.6 % agreed or strongly agreed that they enjoyed the fantasy-world provided by the game. The trend is that the players liked the fantasy world. That 33.3% disagreed is not an unexpected result as Malone (1980) states that a consequence of people's emotional needs is that people like different fantasies. He also states that to have a broader appeal, a game should have given different players different fantasies. In a different setting, he describes the advantages of having a fantasy that is closely related to the learning material, and that this can help the player exploit the analogies between their current knowledge and the what they are about to learn. We choose the option where fantasy and learning material matched. In summary, the fantasy influenced the majority of subjects experience.

Learning outcome

5.6% strongly disagreed, 38.8% disagreed, 50% agreed and 5.6% strongly agreed to that they learned something while playing the game. One of the possible explanations that only 55.6% agreed to that they learned something, could be that many of the subjects did not read carefully through the texts presented in the game. During the interviews two of the interviewees answered that they skimmed through the texts, another stated that he read the information between the projects but felt he did not have time to read the rest of texts, another said that he did and that he felt the texts was clear and brief and gave him the information needed. The observations showed that only one of the participants read all the information carefully, two skimmed through the texts and the last one read the texts between projects. The results regarding information are quite diverse. Two of the subjects observed skimmed through the texts and two of the five interviewees skimmed through the texts (four different subjects). This shows that the texts were not read as thoroughly as the researchers had planned. As most of the learning material was presented through the texts this has a negative effect on the learning outcome of this game. The researchers tried to make the texts fun, interesting, short and included them in the fantasy, to encourage the player to read them. This could be an indication that if texts are used to convey information, it would need another approach when being presented. When asked if they learned something while playing the game, one of the interviewees said that most of the learning material was things he already knew and another stated that knew what project management was. This shows that some of the subjects already knew some of the learning material presented. This in combination with the subjects reading hastily through text could be a possible explanation of why 44.4% of the subjects felt that they did not learn from the game. A higher percentage of the respondents agreed or more to that they got an increased understanding of project management through playing the game. 0% strongly disagreed, 33.3% disagreed, 55.6% agreed and 11.1% strongly agreed to this item. This shows a slight tendency, that more of the subjects got a better understanding then actually learned something new.

Strategy and learning

The fact that the participants were able to explain to us their strategy in detail in interesting. If the players strategy is the content learned, this could be an effective teaching approach. The main focus of our simulation was not to create a completely realistic model. In this study, the simulation was used to show the project manager's role and teach the players about structuring their project team and be an arena where the player could test, use strategies and experience what they learned through reading the texts. To incorporate more depth into the simulation and the strategies, one could argue that the learning experience would have had a better effect.

13.4 Interface and Usability

One essential condition for creating an enjoyable game is that the player needs to be able to understand the user interface. While it is not an element by itself, the GameFlow model is based on the literature a review on usability and user experience in game literature. For this reason, we see the importance of discussing the usability of the game.

During the experiment, it was deliberately avoided explaining how the players should play the game. The only information presented, was that the game teaches project management and is played in the browser using a computer. The GameFlow model explains that the player should be able to play the game without reading a manual and that learning the game should be a fun in-game experience (Sweetser and Wyeth, 2005) . Avoiding guiding the user is also beneficial when using the System Usability Scale, as one could argue that the respondents are not familiar with the game already, and the score is more representative for first-time use. The SUS score on the prototype was 77.7. When compared with the results from Bangor et al. (2008)'s evaluation of the systems using SUS, the score was above average and considered a good score.

Positive Feedback

Most of the participants said that they found the user interface intuitive to use. Some of them even underlined this and stated that they had no problem at all using it. While prototyping the game, we had to create an intuitive user interface, to be able to play-test the mechanics of the game. A criterion for the development of player skills in the GameFlow model is - the game interfaces and mechanics should be easy to learn and use (Sweetser and Wyeth, 2005). The result of creating multiple prototypes was that we iteratively could improve the overall usability of the system.

Early in the development process, we discovered that using buttons purely with texts made the user interface user interface cluttered. To deal with this issue we had to find a way to organize and use symbols, to make the user interface tidier. We created our own graphics assets as the most important strategy for good usability. Designing our own symbols meant that we could create graphics that support the theme of the game, and we could use the symbols to help the player understand what to do. One other strategy was to take inspiration from similar games with good usability. Adams and Dormans (2012) advice that one can effectively use an existing game as a model for creating an interface. When prototyping the game, we looked at games such as "The Sims". Viewing similar games gave ideas on interface elements that we could use, that also gave enough feedback to the player.

When the player starts playing the game, it is crucial that he learns what each button does and the meaning of the symbols. This is relevant to understanding how to play the game. We implemented tooltips some buttons to address this issue. When the mouse cursor was pleased over a symbol, a little box with the most important information about it was presented. We did this consequently on all buttons so that the player always knew where to look.

To create a more intuitive user interface, we made sure that all the buttons gave immediate visual feedback when pressed. Most of the buttons also gave auditive feedback when pressed. Actions that was active over a longer period of time, all had some kind of visual representation that showed their progress.

Some Usability Issues

Several participants said that they had to use the first level to try things out, and did not understand much at that time. Sweeter (2004) states that players should be taught to play the game through initial levels. While developing the game, we attempted to make the first level of the game much simpler than the rest, but this was difficult to do since making the first level very simple, would give the player an advantage in the later stages in the game. For instance, the player could use the time to train the employees so that they could get many skills. One way of solving this problem could have been to have a maximum amount of skills that the player could give to the employees, or have the initial level be like a playground, where their action would not have any effect on the later levels.

One of the participants in the observation used the zooming function in chrome to make the game appear bigger. In a discussion afterward, it was said that some of the icons were too small. This was also pointed out by others while play-testing, so we choose to include a question in the questionnaire that asked if the players were able to understand the meaning of the symbols. Most of the participants agreed with this statement, but as the symbols in the game are essential to understanding how to play, we will note this as an improvement that can be made.

Some of the participants said that they found it difficult to see the status of the employees. This was the biggest challenges with the usability while developing the game. Sweetser and Wyeth (2005) states the players should receive feedback on their actions, and the state and score should always be known. The game uses several states and objects, and a question we frequently asked our self while prototyping was, how should the player be aware of all this ? Prototyping and play-testing became the most important method when dealing with this issue.

Sufficient Usability

Overall, the usability of the game seems to be sufficient so the players were able to learn the game reasonably fast, and it seemed they did not meet any major challenges when playing. The fact that we did not provide a user manual or presented any description on how to play the game, makes this result somewhat interesting, and we chose to describe some of the interface elements, and strategies that we used to accomplish this. Some minor issues which include that the first level was too stressful for the player to learn the game, some of the icons were too small and unclear and sometimes it was difficult to know the status of the employees. We will include these in the list of improvements that can be made.

Summary

In this chapter, we discussed the result of this study in four topics. First, we looked at the results related to learning and how these affected the participants. This showed that the all of the GameFlow elements affected the players enjoyment, resulting in that all the participants agreed to that they enjoyed the game. Second, the results showed that the game mechanics in the game allowed the players to use strategy. Third, we discussed how the game affected the learning experience of the player. It was shown that the learning outcome was not as good as expected and this may be a result of players not reading the learning material and knowing parts of the subject already. Lastly, the user interface seemed to be intuitive enough that the participants were able to play the game without any major problems.

Part V

Summary

The aim of this part is to bind the whole thesis together, present the conclusion to each of the research questions and explains the next logical steps for further research.

Chapter 14

Conclusion

The goal of this study is to design and develop an enjoyable strategy/simulation game that focuses on teaching project management, and evaluate how this game affects the players. The game prototype was developed using inspiration from commercial games, the GameFlow model to design enjoyment into the game, and game mechanics encouraging the use of strategy, planning and reflection. Through iterations we tested, implemented and adjusted the game mechanics, usability, learning content and the enjoyable elements of the GameFlow model. The game was tested and evaluated on 18 participants through a questionnaire, interviews, and observations.

RQ1 - How does the game affect the players' enjoyment?

Enjoyment is the main motivating factor for the game and the GameFlow model was used as a guide to incorporate enjoyment into the game. As all the respondents agreed or strongly agreed to that they enjoyed playing the game, it became apparent that enjoyment was successfully implemented. The analysis indicates that all the GameFlow elements had good results and were successfully implemented and that they affected the players' enjoyment. This confirms that the GameFlow is a model that can be used to successfully to build enjoyment into a game and that the GameFlow and the elements implemented in the game affect the player's enjoyment positively. Another important indication was that strategy was an enjoyable part of the game and that strategy can be used as a fun in-game method in a learning process.

RQ2 - How does the game affect the players learning experience?

The learning experience is designed into the game through the use short cycles/missions, in the start of each cycle, new relevant information is given to the learner that directly afterward could be used in a simulation. The simulations were designed to have many possible paths, to make the user create strategies that required reflection and planning. Some issues with the learning experience became clear during the experiment and evaluation. The most apparent issue was that information presented textually was seldom read thoroughly, as this was the most used tool to convey learning material to the learner, this had a negative effect on the learning outcome of the game. This experience shows that being careful when using text in learning games is a sound idea and that there is a need for research on how to use texts in learning games or do not use them at all. Another issue was that some of the participants already knew parts of the learning material before playing the game. This and the text could be an indication of why 55.6% agreed they learned something while playing the game. On the other hand, two-thirds of the participants agreed they got an increased understanding of project management. Our experience and evaluation of the learning outcome also gave unexpected results. It showed that in a game where strategy, planning and enjoyment are used through multiple cycles the players can remember their plans and strategies in detail. And that if the strategies and plans themselves are the content being learned this have potential use in a learning game.

RQ3 - What types of game mechanics can be used in the game to encourage the player to use strategy?

In the literature study, we learned that one way of describing game mechanics, is with "player actions" and the game objects that these actions work on. In the game design process, we focused on experimenting with different mechanics, and our goal was to improve these over several iterations, while doing playtesting to get feedback. The result of this was a game with four main types of actions "work on task", "improve skills", "take a break" and "use abilities". The evaluation showed that the participants was interested in planning their actions, and could clearly tell us what kind of strategy they had used. All of the actions mentioned when explaining their strategy, except for "use ability", which actually made the game less challenging. Most important were the fact that the game allowed the players to specialize employees with different skills and that there was a game model that presented multiple situations where the player could plan for the employees to work with better efficiency.

RQ4 - What are the technology options for creating web games ?

Web games have the advantage of not requiring the player to install anything, and in this study, we chose not to include technologies that require the player to install any plugins. While it is possible to create web games using a combination of HTML, CSS and JavaScript, much better performance can be gained with HTML5 Technologies. The HTML5 technologies that makes it possible to use graphics and audio in the browser is Canvas, WebGL, HTML5Audio, and WebAudioAPI. The conventional way of using these technologies is with a library, which gives the advantage of not having to "reinvent the wheel", and does often provide a fallback mechanism if the browser does not support the technology. In this study, we briefly described Phaser, CreateJs and more thoroughly Pixi.js, which was the main library choice for this study. Pixi.js proved to be simple to learn, using syntax similar to ActionScript, and it provided decent documentation. Furthermore, web games are most often developed using JavaScript. To use the latest features in the language, it may be convenient to use tool that lets developer use these features today. We identified two such tools, namely Babel.js and TypeScript. The later, was chosen for this project and it enabled us to use object oriented syntax while developing.

Chapter 15

Further Work

Given the exploratory nature of this research there is a need to further investigate some of findings and issues. We think that this type of games have potential to create enjoyable and efficient learning experiences. Some of our findings and issues with potential for further exploration and studies will be presented in the list below:

- Further investigations of our prototype with a focus on more participants and a more diverse population, would be a logical next step for this research. The motivation for such a study, would be to create more generalizable results.

- Create a new game that focus on moving the learning content from the texts into the simulations and strategies, to make the learner gain knowledge from planning and creating strategies, instead of using texts as the main method for conveying information.

- Through our study we found that all the interviewees could explain in detail about the plans and strategies they used while playing the game. A study can be conducted on how to create a standardised model for development of strategy based components in learning games. Trying to find techniques to transfer different types of learning material into models of game mechanics that can be used in a standardised way. This would not only be practical for game-based learning but have applications in traditional computer games.

- Another topic that can be further studied is how to use text in a game, for learning purposes. In such a study one could dive into the world of commercial games, and study, compare and explore how good quality games uses texts in a motivating and enjoyable way.

- Investigate how to personalise games and make player connect with the content presented in the game. An interesting point of view could be to change the texts according to the player's in game actions and use analysis of personality to generate or select customized content.

Bibliography

Adams, E. and Dormans, J. (2012). *Game mechanics: advanced game design*. New Riders.

Adobe (2015). Flash. http://blogs.adobe.com/animate/welcome-adobe-animate-cc-a-new-era-for-flash-professional/. Accessed: 2016-06-01.

Askestad, E. and Knutsen, V. (2016). The project manager game: A learning game for project management - prestudy. *NTNU, IDI*.

Babel (2016). Javascript compiler. https://babeljs.io/. Accessed: 2016-05-15.

Bakken, S. (2013). The software engineering game. *NTNU, IDI*.

Bangor, A., Kortum, P. T., and Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594.

Basili, V. R. (1992). Software modeling and measurement: the goal/question/metric paradigm.

Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.

CreateJS (2016). A suite of modular libraries. http://www.http://createjs.com//. Accessed: 2016-05-15.

Csikszentmihalyi, M. (1990). The psychology of optimal experience new york.

Curran, K. and George, C. (2012). The future of web and mobile game development. *International Journal of Cloud Computing and Services Science*, 1(1):25.

Dantas, A. R., de Oliveira Barros, M., and Werner, C. M. L. (2004). A simulation-based game for project management experiential learning. In *SEKE*, volume 19, page 24.

Freeman, E., Robson, E., Bates, B., and Sierra, K. (2004). *Head first design patterns*. " O'Reilly Media, Inc.".

Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1):20–20.

HowlerJs (2016). Modern web audio javascript library. `http://goldfirestudios.com/blog/104/howler.js-Modern-Web-Audio-Javascript-Library`. Accessed: 2016-05-22.

Hunicke, R., LeBlanc, M., and Zubek, R. (2004). Mda: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI*, volume 4, page 1.

Johnson, D. and Wiles, J. (2003). Effective affective user interface design in games. *Ergonomics*, 46(13-14):1332–1345.

Juul, J. (2002). The open and the closed: Games of emergence and games of progression. In *CGDC Conf.*

Kerzner, H. R. (2013). *Project management: a systems approach to planning, scheduling, and controlling*. John Wiley & Sons.

Koster, R. (2014). A theory of fun, 10 years later. `http://www.gdcvault.com/play/1016632/A-Theory-of-Fun-10`. Accessed: 2016-05-27.

Malone, T. W. (1980). What makes things fun to learn? heuristics for designing instructional computer games. In *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*, pages 162–169. ACM.

McAnlis, C., Lubbers, P., Jones, B., Mazur, A., Bennett, S., Garcia, B., Lin, S., Popelyshev, I., Howard, J., Ballantyne, I., et al. (2014). *HTML5 Game Development Insights*. Apress.

Navarro, E. (2006). *SimSE: A Software Engineering Simulation Environment for Software Process Education DISSERTATION*. PhD thesis, Citeseer.

Navarro, E. and Van Der Hoek, A. (2009). Multi-site evaluation of simse. In *ACM SIGCSE Bulletin*, volume 41, pages 326–330. ACM.

Oates, B. J. (2005). *Researching information systems and computing*. Sage.

Paras, B. (2005). Game, motivation, and effective learning: An integrated model for educational game design. ,.

Phaser (2016). Desktop and mobile html5 framework. http://www.phaser.io/. Accessed: 2016-05-15.

PixiJS (2016). Rendering engine. http://www.pixijs.com/. Accessed: 2016-05-15.

Prensky, M. (2003). Digital game-based learning. *Computers in Entertainment (CIE)*, 1(1):21–21.

Salen, K. and Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. MIT press.

Schell, J. (2014). *The Art of Game Design: A book of lenses*. CRC Press.

Sharafi, P., Hedman, L., and Montgomery, H. (2006). Using information technology: engagement modes, flow experience, and personality orientations. *Computers in Human Behavior*, 22(5):899–916.

Sicart, M. (2008). Defining game mechanics. *Game Studies*, 8(2):1–14.

Sweetser, P. and Wyeth, P. (2005). Gameflow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)*, 3(3):3–3.

TweenJs (2016). Javascript library for tweening and animating. http://www.createjs.com/tweenjs. Accessed: 2016-05-22.

TypeScript (2016). Javascript that scales. https://www.typescriptlang.org/. Accessed: 2016-05-15.

Unity (2016). Game engine. https://unity3d.com/. Accessed: 2016-05-15.

Van Eck, R. (2006). Digital game-based learning: It's not just the digital natives who are restless. *EDUCAUSE review*, 41(2):16.

Appendix A

Questionnaire Results

Table A.1: Gender

Statement	Male	Female
What is your gender?	94.4%	5.6%

Table A.2: Student

Statement	Yes	No
Are you currently a student?	88.9%	11.1%

Table A.3: Gaming Habits

	Does not	Casual	Experienced	Hardcore
Statement	play games	gamer	gamer	gamer
Which of these categories suites your gaming habits best?	5.6%	33.3%	44.4%	16.7%

Table A.4: Playthrough

Statement	I did not play through the whole game	1	2	3	4	More
How many times did you played the game?	5.6%	55.6%	22.2%	11.1%	5.6%	0%

Table A.5: Enjoyment, Flow, Enjoyable Elements and Learning

	Strongly			Strongly
Statement	Disagree	Disagree	Agree	Agree
I enjoyed playing the game	0%	0%	66.7%	33.3%
I was concentrated while playing the game	0%	5.6%	38.9%	55.6%
The challenges presented in the game matched my skill level	0%	27.8%	44.4%	27.8%
The objectives in the game were easy to understand	0%	16.7%	66.7%	16.7%
The game gave feedback that helped me reach the goal	0%	33.3%	38.9%	27.8%
I was in control of my actions while playing the game	0%	5.6%	50%	44.4%
I lost awareness of my surroundings while playing the game	0%	27.8%	66.7%	5.6%
I enjoyed the fantasy-world provided by the game	0%	33.3%	44.4%	22.2%
I was motivated to explore the content in the game to perform better	0%	11.1%	38.9%	50%
I could perform better in the game by planning my actions	5.6%	0%	16.7%	77.8%
Using strategy was an important part of the game	0%	5.6%	27.8%	66.7%
I was motivated to improve my score	0%	27.8%	33.3%	38.9%
The audio in the game enhanced my experience positively	16.7%	5.6%	72.2%	5.6%
The graphics and animations enhanced my experience positively	0%	11.1%	50%	38.9%
I would like to use this or a similar type of game in an educational context	5.6%	16.7%	61.1%	16.7%
I got an increased understanding of project management through playing the game	0%	33.3%	55.6%	11.1%
I learned something while playing the game	5.6%	38.9%	50%	5.6%
I would like to learn more about the content presented in the game	0%	27.8%	38.9%	33.3%
I was motivated to click on objects to get information	5.6%	16.7%	44.4%	33.3%
I was motivated to learn by playing the game	0%	16.7%	66.7%	16.7%
I found it easy to understand and remember the meaning of the symbols in the game.	0%	33.3%	38.9%	27.8%

Table A.6: Emotions

Statement	Boredom	Frustration	Confusion	Excitement	A sense of achievement
Mark the emotions you felt when playing the game	5.6%	11.1%	38.9%	77.8%	77.8%

Table A.7: System Usability Scale

Statement	Strongly Disagree	Disagree	neutral	Agree	Strongly Agree
I think that I would like to use this system frequently	11.1 %	27.8 %	22.2 %	33.3 %	5.6
I found the system unnecessarily complex	44.4 %	38.9 %	16.7 %	0 %	0
I thought the system was easy to use	0 %	0 %	22.2 %	50 %	27.8
I think that I would need the support of a technical person to be able to use this system	77.8 %	22.2 %	0 %	0 %	0
I found the various functions in this system were well integrated	0 %	0 %	22.2 %	50 %	27.8
I thought there was too much inconsistency in this system	66.7 %	27.8 %	5.6 %	0 %	0
I would imagine that most people would learn to use this system very quickly	0 %	0 %	27.8 %	44.4 %	27.8
I found the system very cumbersome to use	38.9 %	38.9 %	22.2 %	0 %	0
I felt very confident using the system	0 %	0 %	22.2 %	50 %	27.8
I needed to learn a lot of things before I could get going with this system	33.3 %	50 %	16.7 %	0 %	0