

Jukka Haltsonen

GUIDE TO WRITING A GAME DESIGN DOCUMENT

GUIDE TO WRITING A GAME DESIGN DOCUMENT

Jukka Haltsonen
Bachelor's thesis
Fall 2015
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Software Development

Author: Jukka Haltsonen

Title of thesis: Guide to Writing a Game Design Document

Supervisor: Kari Laitinen

Term and year of completion: Fall 2015

Pages: 23 + 1 appendix

An important part of game design is writing a Game Design Document. Game Designers, who are just starting their careers, and are new to the game industry, are usually not familiar with the documentation process. The purpose of this thesis was to create an example Game Design Document, and a guide about writing one, for Oulu Game Lab to use as a part of their coaching.

When this work was begun, it was already known that the game industry does not have any standards for the Game Design Documentation. Several game design related books and web articles were studied, and compared with each other and with personal experiences, to find the most uniform information about writing a Game Design Document. The found information was then used to create the guide. Concurrently with the research, a design of a game was begun and a Game Design Document was written.

The work led to a conclusion that a perfect template for the Game Design Document cannot be created, but the guide works as a good basis for a designer to create a template that suits his needs.

The example Game Design Document was not finished during the timeframe of this thesis. For it was found during the research that the processes of designing a game, and writing a Game Design Document are iterative in nature, and need the effort of the whole development team to be completed.

Keywords: Game Design, Game Industry, Documentation

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikka, ohjelmistokehitys

Tekijä: Jukka Haltsonen
Opinnäytetyön nimi: Guide to writing a Game Design Document
Työn ohjaaja: Kari Laitinen
Työn valmistumislukukausi ja -vuosi: Syksy 2015
Sivumäärä: 23 + 1 liite

Dokumentointi on olennainen osa pelinkehitystä, mutta nuorilla pelisuunnittelijoilla ei useimmiten ole tietämystä dokumentoinnin tarpeellisuudesta, taikka siitä miten se tulisi toteuttaa. Tämän opinnäytetyön tarkoituksena oli tuottaa esimerkki pelisuunnitteludokumentista, sekä opas pelisuunnitteludokumentin laatimisesta, Oulu Game Labin käytettäväksi osana valmennustaan.

Tätä työtä aloitettaessa oli jo tiedossa, ettei pelialalla ole olemassa standardeja pelisuunnitteludokumentaatiolle. Työssä tutkittiin useita pelisuunnitteluun liittyviä kirjoja sekä nettiartikkeleita, joita vertailtiin keskenään sekä henkilökohtaisten kokemusten kanssa. Näin löydettiin yhdenmukaisin tieto pelisuunnitteludokumentaation kirjoittamiselle. Tämän tiedon perusteella laadittiin opas. Tutkimuksen kanssa samanaikaisesti, aloitettiin pelin suunnittelu sekä pelisuunnitteludokumentin kirjoitus.

Työssä tultiin siihen johtopäätökseen, ettei pelisuunnitteludokumentille voi luoda täydellistä mallipohjaa, mutta tehty opas toimii hyvänä alkuna suunnittelijalle, hänen omiin tarpeisiin sopivan pohjan laatimiselle.

Esimerkkiä pelisuunnitteludokumentista, ei saatu valmiiksi asti opinnäytetyön puitteissa. Sillä tutkimuksen aikana todettiin, että pelin suunnittelu sekä pelisuunnitteludokumentin kirjoittaminen, ovat luonteeltaan iteratiivisia prosesseja. Näin ollen, ne tarvitsevat koko kehitystiimin työpanoksen valmistuakseen.

Asiasanat: Pelisuunnittelu, Peliala, Dokumentaatio

CONTENTS

ABSTRACT	3
TIIVISTELMÄ	4
TABLE OF CONTENTS	5
1 INTRODUCTION	6
2 WHY DO WE NEED DOCUMENTS?	7
3 TYPES OF DESIGN DOCUMENTS	8
3.1 High Concept Document	8
3.2 Character Design Document	11
3.3 World Design Document	11
3.4 Flowboard	12
3.5 Story and Level Progression Document	12
3.6 The Game Script	13
4 QUALITIES OF A GOOD DESIGN DOCUMENT	14
5 CASE STUDY: ANGRY MOBS	19
5.1 Overview	19
5.2 Gameplay and Mechanics	19
5.3 Characters	20
5.4 Game World	20
5.5 User Interface	20
6 SUMMARY	21
REFERENCES	22
APPENDICES	24

1 INTRODUCTION

Writing a Game Design Document (GDD) is a relevant part of game design, but still many designers do not write one at all, or write it badly. Often the reason for this is that the designers do not have the knowledge of how this kind of document should be written. In many cases, this leads to disorderly and incoherent design, which unfortunately often leads to the failure of the whole project. The idea for this thesis came from Oulu Game Lab's need to guide new game designers in writing a GDD.

There does not exist any kind of standard for the Game Design Document. The structure of the document is affected by the nature of the game being designed, the designer's personal writing style and the game studio's preferences. However, all GDDs have some similarities and generally well-trying ways.

The aim of this thesis was to create a guide how a GDD should be written, based on the aforementioned generally valid rules. The guide also needed to include an example GDD. For that purpose, a game was designed and the game's design document was written following the guide.

2 WHY DO WE NEED DOCUMENTS?

It is common knowledge around game industry that no one ever reads design documents. Then why should such documents exist then?

“The documents record decisions made and agreed upon orally; they create a paper trail. More important, the process of writing a document turns a vague idea into an explicit plan. Even if no one reads it at all, an idea written down is a decision made, a conclusion reached. If a feature of a game is not described in writing, there’s a good chance that it has been overlooked and that someone will have to make it up on the fly—or, worse, that each part of the team will have a different idea of what they intended to do.” (1, p. 55.)

Schell (2, p. 382) gives two reasons for documents to exist: memory and communication. The design of a game is full of decisions that define the game. How it works and why? It is very likely that the designer will not remember them all. If the designer records his design decisions, he will be spared from solving the same issues again at a later time.

The design decisions have to be communicated to the development team, and with documents this can be done effectively. When the design is in written form, the other members of the team can review it and will be likely to find problems with it, or have ideas for improvements. (2, p. 383.)

3 TYPES OF DESIGN DOCUMENTS

It is a common misconception that the Game Design Document has tens or even hundreds of pages, all combined in one giant tome, and the more pages there are, the better the document. This used to be the way back in the day, but nowadays it is more common to split the design into several smaller documents. (1, p. 58). Of course, there may still be hundreds of pages in total, but quantity does not automatically mean quality. More about the subject can be found in Chapter 4. Qualities of a Good Design Document.

Several kinds of design documents exist. Which ones should be used and what kind of information they should contain is mostly up to the designer to determine. For a rookie designer, this task may be too overwhelming, though. To help in this endeavor, this section introduces the types of design documents Adams (1, p. 55) considers to be some of the most common ones.

3.1 High Concept Document

“A game concept is a description of a game detailed enough to begin discussing it as a potential commercial product—a piece of software that the public might want to buy” (1, p. 67).

According to Adams (1, p. 56), the high concept document is like a résumé. Its purpose is to get a hearing from someone, usually from an investor. It explains a game concept in a short manner; not more than two to four pages. The document should include at least the following nine key points (1, p. 67):

1) High Concept Statement

High concept statement describes in just a few sentences what the game is about. References to other media that utilize similar ideas, can be used. (1, p. 67, p. 83.)

2) Player role

The description of the player's role in the game. Is the player pretending to be someone or something? Is there more than one role? How does the player's role help to define gameplay?

If the player has an avatar character, it should be described briefly. (1, p. 67, p. 83.)

3) Gameplay

The primary gameplay mode is the one that should be described in this section, along with any competition modes the game will support (single- or multiplayer; competitive or co-op). (1, p. 68.)

Adams (3) describes a gameplay mode as "a particular way of playing a game", which consists of three elements: Perspective, interaction model and gameplay.

- **Perspective**

Means of perceiving the game world. Defined by a camera looking at 2D or 3D space from a particular point of view.

- **Interaction model**

Means of influencing the game world. The most common ones are the *avatar-based* model (used in most action games and platformers) and the *omnipresent* model (used in most board and war games).

- **Gameplay**

The challenges the player faces and the actions he/she can take to overcome those challenges.

If one of these elements changes significantly during the play, then the player moves into another gameplay mode. The primary gameplay mode is the one in which the player is going to spend most of the time. (3.)

4) Genre

In which genre the game belongs or, if the game is a mix of genres, which features it contains from the different genres to which it belongs? If the game does not fit into any existing genre, an explanation of why not, is required. (1, p. 68.)

5) Platform

On which machine(s) and operating system(s) will the game run on? This includes details of any special equipment needed/supported (e.g. a camera or VR glasses), and any licenses that the game will utilize. (1, p. 68.) Also, in case of a PC game, the system requirements or, an estimate of them, can be told (4).

6) Target Audience

Who would want to play the game? How do players of this game differ from the mass of players in general? (1, p. 83.) The game's ESRB and/or PEGI ratings, or a prediction of them, can be added here (4).

7) Game World

A brief description of the game world. The look and feel of the environment where the gameplay takes place. (1, p. 68, p. 83.)

8) Game Flow

A general outline of the game's progress from beginning to end. Some ideas for levels or missions should be included. If the game has a story, there should be a synopsis of the storyline. (1, p. 68.)

9) Marketing

As most games are commercial products, this section should include, especially if the game is offered to a publisher, the following points: Game's potential competition, the unique selling points (USPs) that make the game stand out in the marketplace, marketing strategies and possible merchandising opportunities. (1,

p. 68.) In case of a free-to-play game, the monetization model needs to be explained. If the game is not created for financial gain, this section should explain why.

3.2 Character Design Document

The purpose of a character design document is to record the design of the characters who appear in a game. It primarily shows the characters' appearance and animations, so there should be plenty of concept art of the character in different poses and with different facial expressions (Figure 1). It should also include background information about the characters. (1, p. 56.)

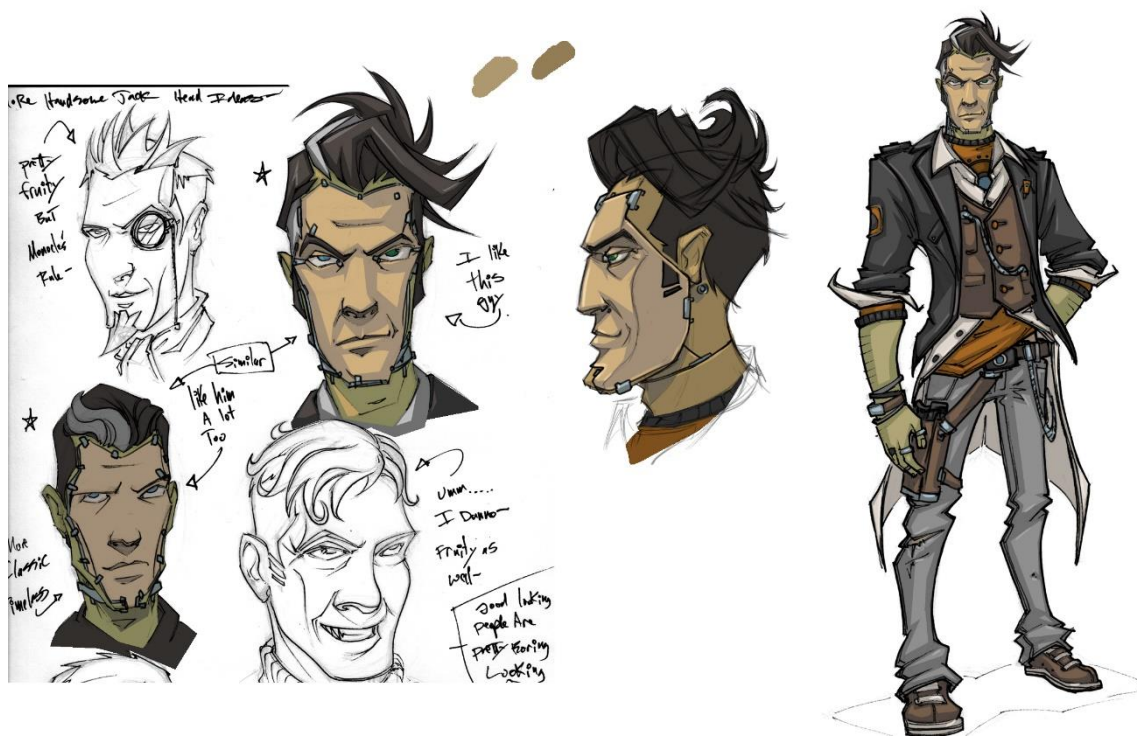


FIGURE 1. The concept art of Handsome Jack, *Borderlands 2* (5).

3.3 World Design Document

The world design document is the base for creating the visual and aural style, and emotional tone of the world. It does not have to record every detail but, rather a general overview. The actual content of the game world is created by level designers, artists and audio designers, based on this information. (1, p. 56-57.)

3.4 Flowboard

A flowboard is the best way to document a game's structure. The term is derived from a flowchart and a storyboard. Storyboards are a linear series of pictures used by filmmakers to plan a set of shots. Flowcharts are diagrams used by programmers for documenting an algorithm. A flowboard combines these two ideas. Each picture is a sketch or a mockup of the screen, in one specific gameplay mode or menu. The pictures are connected via arrows that indicate under what circumstances the transition takes place. (1, p. 57; 3.) The flowboard does not necessarily need pictures in it to work. A simpler approach (Figure 2) might even be better in some cases.

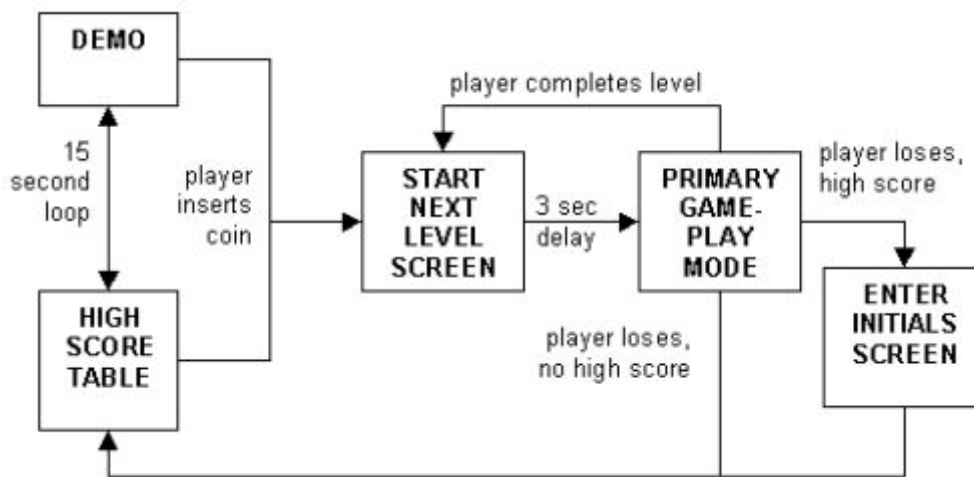


FIGURE 2. A simplified flowboard example, using boxes without pictures (3).

Adams (1, p. 57) suggests that it is better to create the flowboard on several sheets of paper and stick them on a wall, rather than by using a software like Microsoft Visio. Schubert (6; 7) on the other hand, says that Visio is probably the best, single game design tool available. So in the end, it is a matter of personal preference.

3.5 Story and Level Progression Document

Depending on the game being designed, this document can be very large and needs to be split into several smaller documents, or it may not be needed at all.

This document records the story of the game and the way the levels progress from one to another. If the game has a branching story, or a level progression based on the player's actions, this document is the place to indicate which decisions cause the game to take one path over another. How the player experiences the story is also explained here, i.e. if it is told via cut-scenes, dialogue or other narrative elements. (1, p. 57.)

3.6 The Game Script

This document is the core of the game's design, and it is the basis for the programmers to build the game. While the high concept document gives a general overview of what the game is about, this section gives a detailed explanation of how the game is played and all the mechanics that affect the gameplay. (1, p. 58.)

“As a good rule of thumb, the game script should enable you to play the game. That is, it should specify the rules of play in enough detail that you could, in theory, play the game without the use of a computer - -“ (1, p. 58.)

While the game script describes specific game mechanics, like e.g. which button to press to make the player character jump or how much damage a certain enemy can withstand, it should not include any technical design of how to build or implement the game software. This information belongs to a technical design document, if there is one, and is nothing for the game designer to worry about. (1, p. 58.)

4 QUALITIES OF A GOOD DESIGN DOCUMENT

In Chapter 2 it was stated that no one reads design documents. No one reads them because they are considered to be a waste of time. Probably, the most common reason for this is that most GDDs are very poorly written. (6; 7.)

Schubert (6; 7) gives the following reasons for **bad design documentation**:

- Design documents deal with a lot of interconnected systems, which makes it hard to track everything that they are supposed to track.
- Designers tend to overdesign. They design a whole lot of huge features that will never end up in the game, so the rest of the team will not even bother using the document as a reference.
- Design documents do not embrace iteration. Game development is, like all software development, an iterative process. Design documentation often ignores the iterative process of finding fun. Designers do not find the fun on the first time they write a document, but everyone expects the design to be accurate when it comes out of the designer's head. That is not how fun is made. Fun is something that needs to be iterated on a few times, before it is found. Design documentation and the process of creating design documentation need to support that. They need to be revisited as the design changes.
- Documents are not kept up to date as the project progresses, and that makes them useless at some point.

The following eight guidelines are a good basis for the design document, but one should remember that there is always room for improvement (6; 7).

1) Targeted

Documentation means different things to different team members. To producers, it is a tool for scheduling and managing the team, for marketing and getting financing. To designers, it is a way to give details on how the game will work for achieving a design consensus. To programmers and artists, documents are the

instructions for creating the game. To testers it is a tool for building a test plan. (6; 8.)

Ultimately, programmers are the most important audience, so the structure of the GDD should be made to serve the programmers, before all (6; 7).

2) Short

Short documents are easier to read, write, maintain and keep up-to-date. If any one document should become more than ten pages long, it should be split into smaller ones. (6; 7.)

3) Prioritized

Which features and assets are the most important at any given time of the development, need to be pointed out. (6; 9).

4) Illustrated

Visual material should be used as much as possible. This is possible even without actual game art via screens from other games with similar features or with UI mockups and diagrams. Even if no visual material is available, features can be demonstrated by example texts (Figure 3). (6; 7.)

- Players can remove a skill in their skill tree by going to a special NPC (the 'mindwiper') and selecting that skill.
 - Removing a skill has a monetary cost in credits.
 - The player cannot remove a skill that is a prerequisite for another skill in his skill tree.

Joe Bob decides that he wants to unlearn Basic Psionics and Advanced Psionics, so he goes to a mindwiper. He tries to remove the Basic Psionics skill tree, but the transaction fails, as it is a prerequisite for Advanced Psionics. So Joe Bob unlearns Advanced Psionics and then Basic Psionics. In this case, both boxes are successfully removed.

FIGURE 3. A feature and an example text (7).

5) Does not tell others how to do their jobs

The GDD should not try to explain how to implement a feature. This is a problem that is not meant for the designer to solve and will only annoy the programmers. Even if the designer is also a programmer, the GDD is not a place for technical design. If the game designer is not also the UI artist, there should not be any artsy mockup pictures (Figure 4) in the GDD either. That will just make it harder for the artist to do his job, so a more abstract picture (Figure 5) is better. (6; 7.)

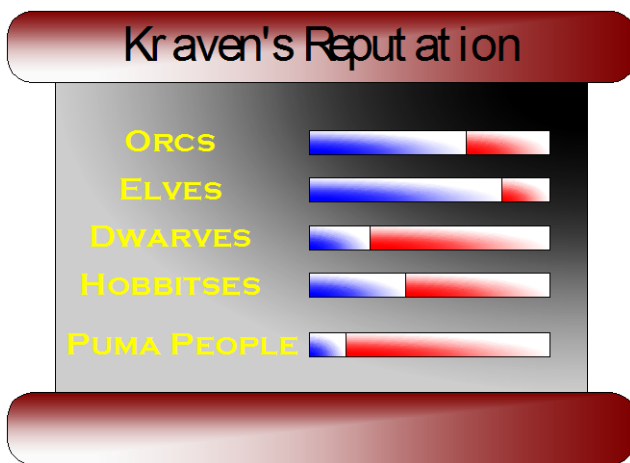


FIGURE 4. The artsy UI mockup (7).

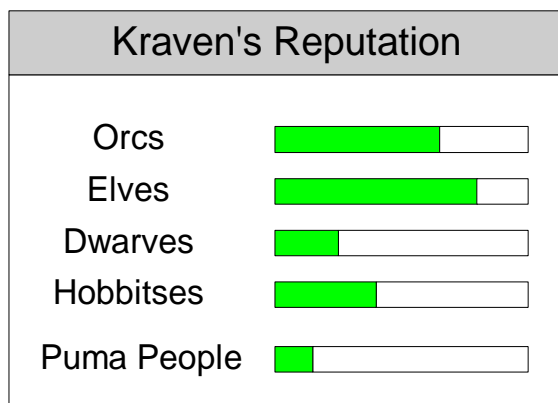


FIGURE 5. The more abstract UI mockup (7).

6) Readable

According to Freeman (9), a good page layout follows these guidelines:

- Plenty of white space
- Serif font for a body text
- Bold headers
- Spaces between paragraphs
- Short lines of text
- Directing the eye towards an important material
- Using a hierarchical, "2D" format

In the end, the most important thing is that the format works for the team.

7) No Redundancy

If a document needs a piece of information that is already described in another document, instead of explaining the same thing again (Figure 6), the text should point to that other document (Figure 7). This helps in keeping the GDD short and avoiding contradictions in the design. (6; 7.)

"CombatStats.doc"	"Items.doc"
<ul style="list-style-type: none">• Strength increases the player's damage by $STRENGTH/2$.• Dexterity increases the player's accuracy by $DEXTERITY/3$• Body odor reduces the player's chance to seduce NPCs by $BODYODOR/2$	<ul style="list-style-type: none">• Strength increases the player's damage by $STRENGTH/2$.• Dexterity increases the player's accuracy by $DEXTERITY/3$• Body odor reduces the player's chance to seduce NPCs by $BODYODOR/2$

FIGURE 6. Two documents explaining the thing (7).

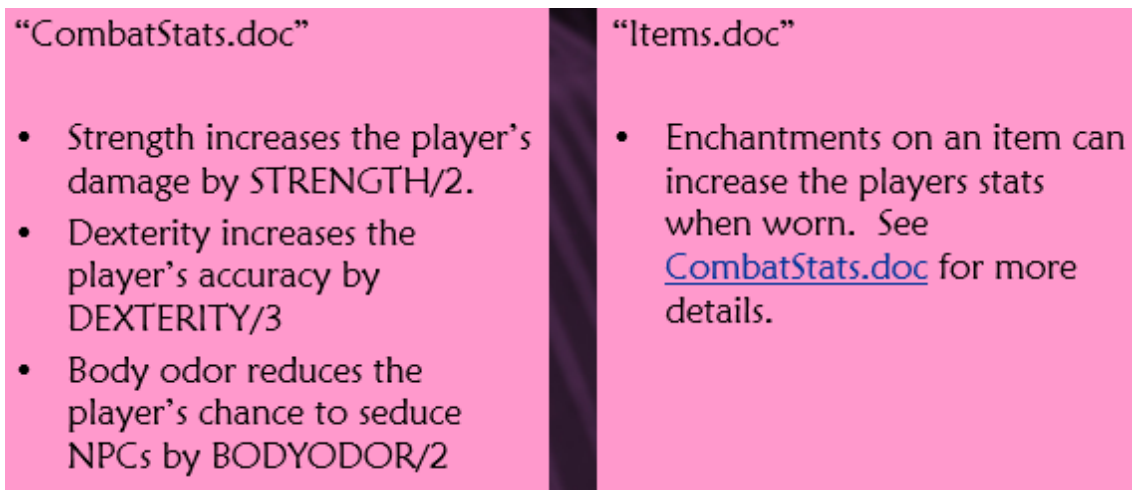


FIGURE 7. A document pointing to another to reduce redundancy (7).

8) No Ambiguous or Weak Language

GDD has to be written in a strong and declarative language. It must not contain words like *maybe*, *could* and *might* because they create a sense of uncertainty about a feature. Surely, there can be some legitimate ‘maybes’ in the design but that is what the prioritizing is for. (6; 7.)

GDD must be unambiguous. There must not be vague expressions like *a number of...* or *affects damage...* in the GDD. For they imply ambiguity. What number? By how much? (10.)

5 CASE STUDY: ANGRY MOBS

In this chapter, some of the topics of the Angry Mobs GDD (appendix 1) are taken under a closer observation. Especially, subjects that are in contradiction with the guide are tried to be addressed.

For the sake of this thesis, the GDD was not split into several documents as it is supposed to. Also, during the design process it was realized that when beginning from scratch, it is easier to first write everything on the same document. After the concept begins to gain some flesh around the bones, it is easier to determine which sections need to be split into their respective documents.

5.1 Overview

This is the High Concept Document but since it has not been separated into its own document, naming this section as Overview seemed more appropriate. The subsection titles here differ from the ones introduced in Section 3.1, but the Overview still contains the information.

Since there is no concept art, music or sound effect samples nor gameplay prototype for the game, many sections contain references to other games, so that the readers can get some kind of image of the proposed game concept in their heads.

There is a core loop -diagram in section 1.2 of the GDD. While it may not serve any purpose for the game's development, the diagram can be helpful when presenting the game concept to a person outside of the development team. Also, in Oulu Game Lab, during Gates, the judges tend to ask for it.

5.2 Gameplay and Mechanics

This is the Game Script, but the name sounds a bit misleading considering the contents of the section. For that reason the section was renamed as Gameplay and Mechanics.

This section is the core of the game. Everything written here needs a thorough testing before one can tell if the design works, so this section will likely be changed several times during the development.

5.3 Characters

This section contains some visual reference and written descriptions of the game's characters' appearance and animations for the artists. Angry Mobs has basically only one character, so in this case the Character Design Document is quite short.

5.4 Game World

As explained in section 2.7 of the GDD, Angry Mobs has only one kind of map that comes in three different sizes, and the map's content will be procedurally generated. So there is no need for level design details in the World Design Document. In this case it only contains a list of art and sound assets needed.

5.5 User Interface

Figure 7 in the GDD is a flowboard of the game's structure. It is a very simplified version and will be updated later on, if necessary. While the guide suggests that the flowboard should be in its own document, it was found to be more convenient to integrate a small diagram like this into an appropriate existing document.

6 SUMMARY

The purpose of this thesis was to create a guide for new game designers about how to write a game design document. The game industry has no standards for the GDD because every game studio and designer have their own preferences for it. Also, the preferences of the audience who will read the GDD need to be considered. The content is also highly dependent on the game being designed. All this considered, it is virtually impossible to create a perfect template for the GDD. The guide still gives a good basis for a designer to create his own template.

Another aim for this work was to create an example GDD to support the guide. What was not taken into consideration in the beginning was the fact that game design and writing the game design document are iterative processes. Trying to create the design and design documentation without iteration is doomed to fail. Before learning this fact, the design process felt hard and frustrating, and the example game design seemed like a total failure. Later on, when the need for iteration was found, the design process came to a halt; the work was being done by only one person, who did not possess all the required skills to make the project move forward.

Since no development other than game design was involved in this work, it is safe to assume that the example GDD is not complete, and it is hard to determine how helpful the incomplete GDD is as an educational material. On the other hand, even the usefulness of a complete GDD would be questionable, since every game and designer is different.

The guide covers solely the writing of a Game Design Document but it is only a part of the whole game design process. Some parts of the guide may be difficult to make use of, and were difficult to write, without getting deeper into game design. This was not possible within the timeframe of the thesis. The guide could be expanded in the future to cover some basics of game design.

REFERENCES

1. Adams, Ernest 2010. Fundamentals of Game Design. 2nd edition. Berkeley, CA: New Riders.
2. Schell, Jesse 2008. The Art of Game Design: A Book of Lenses. Burlington, MA: Elsevier. Date of retrieval 31.10.2015. Available: <http://www.sg4adults.eu/files/art-game-design.pdf>.
3. Adams, Ernest 2004. Designing With Gameplay Modes and Flowboards. Gamasutra. Date of retrieval 6.11.2015. Available: http://www.designersnotebook.com/Columns/061_Designing_With_Gameplay_Mo/061_designing_with_gameplay_modes.htm.
4. Username Gamux 2011. Effectively Organize Your Game's Development With a Game Design Document. Tuts+. Date of retrieval 31.10.2015. Available: <http://code.tutsplus.com/articles/effectively-organize-your-games-development-with-a-game-design-document--active-10140>.
5. Burch, Anthony 2013. Inside the Box: You (Still) Don't Know Jack. Gearbox Software. Date of retrieval 8.11.2015. Available: https://gearboxsoftwarecom.s3.amazonaws.com/images/_/20131025/25f53817ac159a1636589223d02798b33a5d58db.jpg.
6. Schubert, Damion 2007. How To Write Great Design Documents. GDCVault. Audio track. Date of retrieval 1.3.2015. Available: <http://www.gdcvault.com/play/581/Writing-Great-Design>.
7. Schubert, Damion 2007. How To Write Great Design Documents. GDCVault. PowerPoint slideshow. Date of retrieval 1.3.2015. Available: <http://www.gdcvault.com/play/580/Writing-Great-Design>.

8. Ryan, Tim 1999. The Anatomy of a Design Document, Part 1: Documentation Guidelines for the Game Concept and Proposal. Gamasutra. Date of retrieval 4.11.2015. Available: http://www.gamasutra.com/view/feature/131791/the_anatomy_of_a_design_document_.php.
9. Freeman, Tzvi 1997. Creating A Great Design Document. Gamasutra. Date of retrieval 4.11.2015. Available: http://www.gamasutra.com/view/feature/131632/creating_a_great_design_document.php.
10. Romero, Brenda 2008. Creating a Game Design Document. WordPress. Date of retrieval 5.11.2015. Available: <https://bbrathwaite.wordpress.com/2008/11/30/creating-a-game-design-document/>.

ANGRY MOBS

Designed by Jukka Haltsonen

Version 1.0

Created 16.3.2015

Last modified 9.11.2015

CHANGELOG

Date	Changes
16.3.2015	Began to form the idea. Version 0.1
9.11.2015	First pass ready. Version 1.0

CONTENTS

1 OVERVIEW	4
1.1 High Concept Statement	4
1.2 Gameplay	4
1.3 Game World	5
1.4 Genre	5
1.5 Platform and Licenses	5
1.6 Target Audience	5
1.7 Competition	5
1.8 Unique Selling Points	6
2 GAMEPLAY AND MECHANICS	7
2.1 Camera	7
2.2 Controls	7
2.3 Movement	7
2.4 Combat	7
2.5 Gathering the Mob	8
2.6 Buildings	8
2.6.1 Control Points	8
2.6.2 Power-Up Buildings (PUB)	9
2.7 Maps	9
2.8 Leader Attributes	11
2.9 Experience Points (exp)	11
2.10 Level Up	12
3 CHARACTERS	14
3.1 Appearance	14
3.2 Animations	15
4 GAME WORLD	16
4.1 Art	16
4.2 Sound	16
5 USER INTERFACE	17
5.1 Menus	17
5.2 HUD	17
5.3 Messaging	18

1 OVERVIEW

1.1 High Concept Statement

Angry Mobs is a top-down 2D online multiplayer game for PC, where players gather an angry mob and combat against each other with them, while also trying to capture and hold control points in the game zone.

1.2 Gameplay

Angry Mobs is an online multiplayer game, viewed from a top-down camera angle (*reference: Don't Starve*), where players may compete as individuals or as teams.

Players' objective is to gather up peasants from around the game map to form a mob, with which they need to capture control points, and protect them from being captured by another player. Holding a control point gradually gives points to the player. The player having the most points when time runs out, or the first player to reach the point goal wins.

Players also get experience points from capturing control points and fighting with other players. This leads to level ups, which enhance their mob's performance.

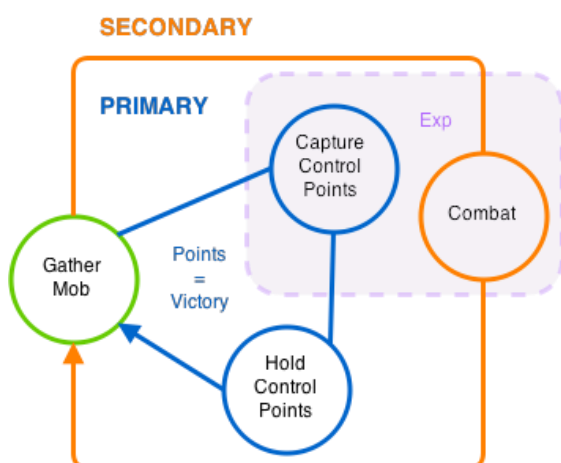


FIGURE 1. Core loop

1.3 Game World

The game has three different sized, procedurally generated medieval styled maps with cartoony 2D graphics (*reference: Prison Architect, Don't Starve*), as well as medieval style music and cartoony sound effects (*reference: The Bard's Tale, Castle Storm*). For more humorous experience, the game will have some features, which were nonexistent in the medieval era (or any other era for that matter).

1.4 Genre

MOBA

1.5 Platform and Licenses

Windows PCs, with regular mouse and keyboard control system.

Game will be created with the Unity3D game engine.

1.6 Target Audience

Angry Mobs will appeal to male players aged 16-30 who typically play MOBA, FPS or RTS games. Also, fans of medieval themed entertainment and humorous cartoons will be attracted to Angry Mobs' theme.

1.7 Competition

Angry Mobs does not have any direct competition because there is no other game that would be exactly alike. With the indirect competitors, the game does not compete for the same audiences.

1.8 Unique Selling Points

- Visual- and aural style uncommon for a MOBA game
- Players' avatars consist of multiple units
- Procedurally generated game map in a MOBA game
- Partially random, partially player chosen upgrades on level up

2 GAMEPLAY AND MECHANICS

2.1 Camera

The game's view will be from top-down perspective angle, in a way that the ground is in perspective but everything else is facing the camera (*reference: Don't Starve*).

The camera follows the player so that the player is always in the center of the screen.

2.2 Controls

Left Mouse Button - Move/Attack/Menu selections

Right Mouse Button - Use Ultimate Power-Up

ESC Button - Open/close in-game menu

2.3 Movement

Mob moves to a point by clicking the left mouse button.

2.4 Combat

Combat is initiated by repeatedly left clicking the opponent. Each click initiates a damage roll which determines how much damage is done to the opponent.

Right clicking the enemy will use the Ultimate Power-Up, if the player has one.

Table 1. Damage roll Formula

$$(aD10 + aMob + aLatt) \times aCrit - (dD10 + dMob + dLdef)$$

Variable	Explanation
<i>aD10</i>	Attacker's Attack roll. Random number between 1-10.
<i>dD10</i>	Defender's Defense roll. Random number between 1-10.
<i>aMob</i>	Attacking mob size
<i>dMob</i>	Defending mob size
<i>aLatt</i>	Attacking Leader's Attack Value
<i>dLdef</i>	Defending Leader's Defense Value
<i>Crit</i>	Attacker's Critical damage multiplier

2.5 Gathering the Mob

The player must move within two (2) unit distance from idle peasants to get them to follow.

2.6 Buildings

2.6.1 Control Points

Mob needs to be within one (1) unit distance to a Control Point (CP) to begin capture.

While a CP is contested, it will not give points.

Table 2. Capturing and Points

Points and Control Point capturing times		
Building	Seconds to get 1 point	Capture time (seconds)
Castle	1	15
Keep	2	10
Tower	3	5

2.6.2 Power-Up Buildings (PUB)

- The types of PUBs are randomized for each match.
- There is a 20% chance for a PUB to spawn an Ultimate Power-Up for the player.
- Multiple different buffs may be active simultaneously, but players may possess only one Ultimate Power-Up at any given time. A new one will override the existing one
- Receiving the same buff that is already active resets the timer, but effects will not stack

Table 3. Power-Up Buildings

PUB	Buff	Ultimate Power-Up
Blacksmith	Increases player's Attack , or Critical Damage by 1 point for 2 minutes	Epic Gear Increases ALL stats by 5 for 3 seconds
Stables	Increases player's Speed by 1 point for 2 minutes	Stampede Horse herd runs over the enemy for 10 damage.
Tavern	Increases player's Defense by 1 point for 2 minutes	Blackout Spins the opponents screen really fast and stops it in a random position
Training Field	Increases player's Attack Rate or Critical Chance by 1 point for 2 minutes	Ninja Moves Mob receives no damage for 3 seconds

2.7 Maps

There are three different sized maps. The dimensions for each size are shown in Figure 2. The places of the control points and buff buildings are randomized for every match. Though, there are limits to where they are allowed to spawn (Table 4. and Figure 2.)

Decorative items will be completely randomized around the map.

Table 4. Map layout color explanations

	Player spawn area. No buildings.
	Tower spawn area. Each section needs to have at least 1 building
	Keep spawn area.
	Castle spawn area.
	Buff buildings may spawn everywhere else but the player spawn areas.
	Section sizes in each map type, shown in Unity3D units and from top to bottom order: small, medium, large

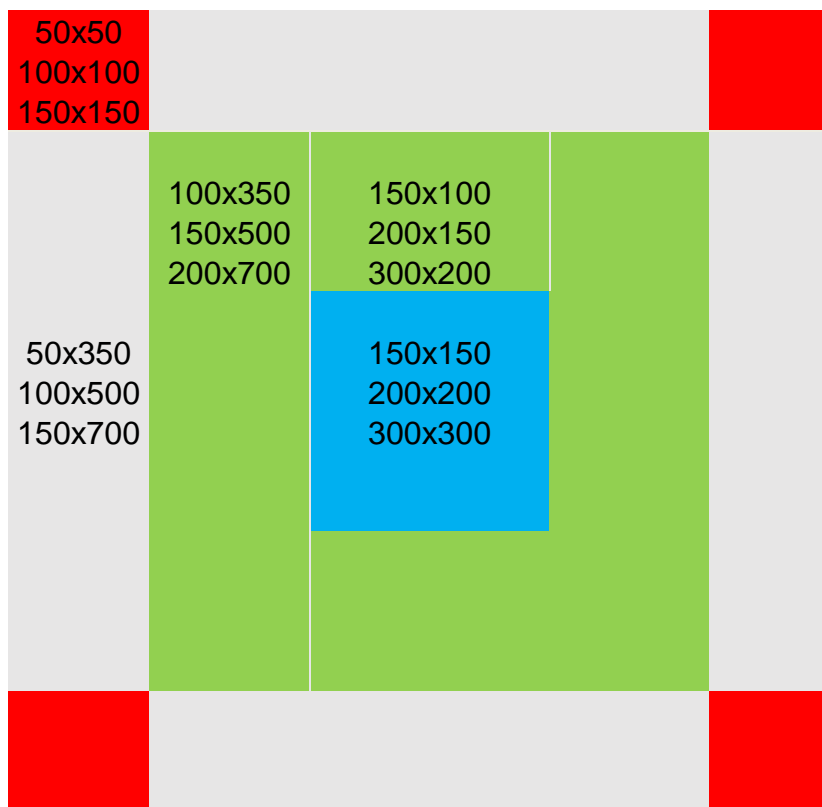


FIGURE 2. Map layout and section sizes

Table 5. Map sizes and amount of buildings

Map	Size (units)	No.# of buildings			
		Tower	Keep	Castle	Buff buildings
Small	450 x 450	4	2	1	4
Medium	700 x 700	6	3	1	6
Large	1000 x 1000	8	4	2	8

2.8 Leader Attributes

Leadership - Increases maximum mob size by 10/point.

Attack - Increases mob's damage by 1/point.

Defense - Decreases mob's damage received by 1/point.

Critical Damage - Increases mob's damage by 5%/point.

Critical Chance - Increases the chance for a critical strike to occur by 5%/point.

Attack Rate - Increases mob's attack speed by 0,125 seconds/point.

Speed - Increases mob's movement speed by 5%/point.

All attributes are at level 1 in the beginning of a match. The maximum level of each attribute is 5, i.e. 4 upgrades/attribute are allowed.

2.9 Experience Points (exp)

Several actions in the game will give experience points to the players.

All exp is split between team members involved in the action.

Table 6. Experience Points

Experience Points	
Action	Exp
Defeat Peasant from opposing mob	1
Defeat Leader of opposing mob	10
Capture Tower	5
Capture Keep	10
Capture Castle	15

2.10 Level Up

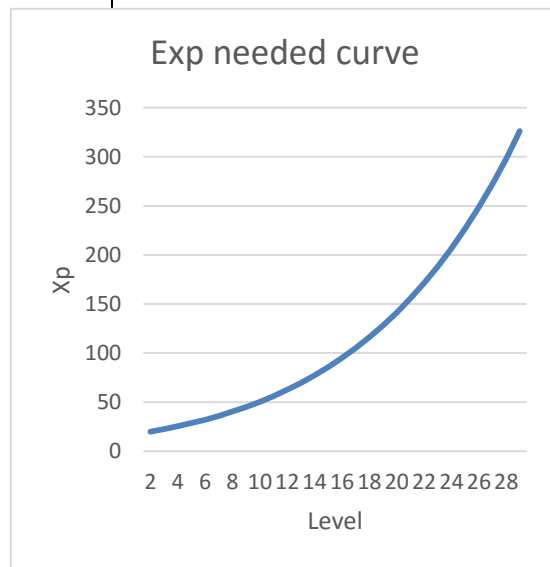
Level up gives the player one (1) attribute point (AP). AP is used for upgrading Leader Attributes.

On level up, a Level Up -notification will be displayed on the screen. Clicking the notification will bring up a screen, displaying 3 random Leader Attributes. The player is prompted to choose one of them to be upgraded. If the player does not choose the attribute upgrades before he gets another level up, the level ups accumulate. Choices are randomized after an upgrade has been made (*reference: Borderlands 2 - Badass rank and -tokens*).

There is no actual level cap, but the theoretical maximum level is 29, since by then each Leader Attribute will be at its maximum level.

Table 7. Level Up formula

Level Up v1			
Total Exp: $a * cl ^ (sqrt cl / b) - a$			
Current Level <i>cl</i>	Total Exp to current lvl	Exp needed from previous to current lvl	Diff
1	0	0	<i>a</i> = 91
2	20	20	20
3	42	22	3
4	67	25	3
5	96	28	3
6	128	32	4
7	164	36	4
8	204	40	4
9	249	45	5
10	299	50	5
11	356	56	6
12	418	63	6
13	488	70	7
14	565	77	8
15	650	86	8
16	745	95	9
17	850	105	10
18	966	116	11
19	1094	128	12
20	1236	141	13
21	1391	156	14
22	1562	171	16
23	1750	188	17
24	1957	207	19
25	2184	227	20
26	2433	249	22
27	2705	272	24
28	3003	298	26
29	3330	326	28



3 CHARACTERS

3.1 Appearance

Game's characters will have a simple 2D appearance resembling common board game pawns (*Figures 3. and 4.*) All characters will look alike, so colors will be used to distinct them from one another: Idle peasants will have gray bodies and the players/teams will have their own colors. Which colors the players/teams have, will be decided later.



FIGURE 3. Character art style of Prison Architect (1).

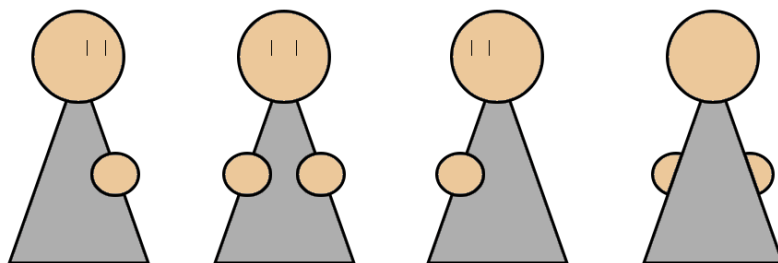


FIGURE 4. Angry Mobs character mockup.

3.2 Animations

Movement - Characters will move by hopping forward (like pawns in a board game).

Combat - When mobs engage in a combat, they will disappear into a “fight cloud” for the duration of the combat. Small details will constantly pop out from the cloud during the fight, like e.g. stars, skulls, small clouds, (*Figure 6.*) and occasionally something more random like a chicken (*Figure 5.*)



FIGURE 5. *Cartoon fight cloud (2).* FIGURE 6. *Cartoon fight cloud 2 (3).*

4 GAME WORLD

4.1 Art

The look of the world is intended to be similar to *Don't Starve* with the emphasis on buildings, having unnatural shapes and proportions.

Art assets needed:

Ground, tower, keep, castle, blacksmith, stables, tavern, training field.

Also, decorative objects like e.g. trees, tree stumps, bushes, rocks, flowers.

4.2 Sound

Background music is intended to be medieval-like to fit the general theme of the game.

Sound effects will be designed later on.

5 USER INTERFACE

5.1 Menus

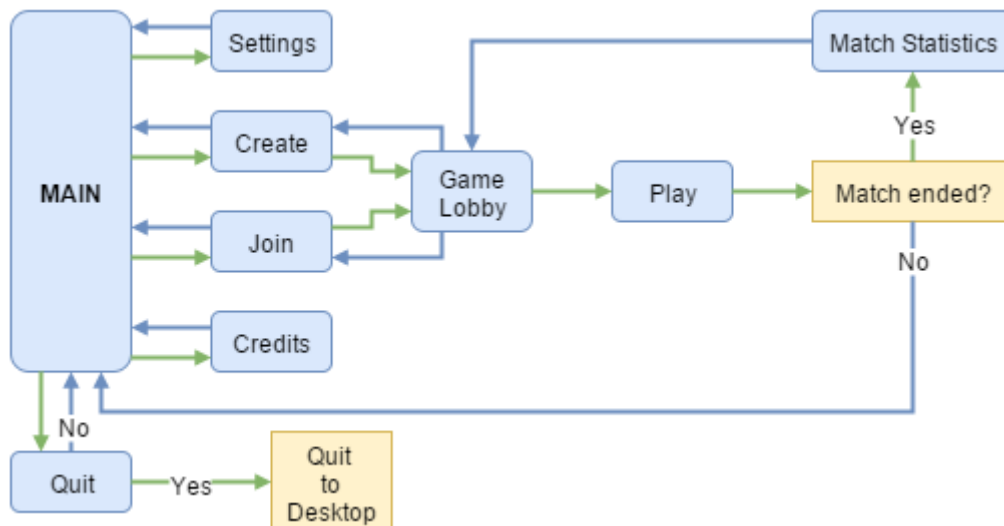


FIGURE 7. Game structure

5.2 HUD

Level Up button on the lower left corner of the screen. Will open the level up - menu, to the center of the screen. Looks like a square with a plus-icon in the middle, which will be lit when a level up is available.

Menu button on the lower right corner of the screen. Will open the in-game - menu to the center of the screen. Looks like a cogwheel.

Mob size is visible on top of the mob in numbers. The Leader does not count to this number.

Leader's health is visible below the mob as a health bar. The health bar will change color according to the amount of health: over 70% = green, 70% - 30% = yellow, less than 30% = red.

5.3 Messaging

There is no chat in the game but players can send simple predefined messages to their teammates. The messages appear as talk-bubbles on the screen (*reference: Full Mojo Rampage*).

The sender will see their message on top of their mob. Other players see the message also on top of the sender, if the sender is visible on their screen. Otherwise, the message will pop up to the side of the screen pointing to the direction of the sender.

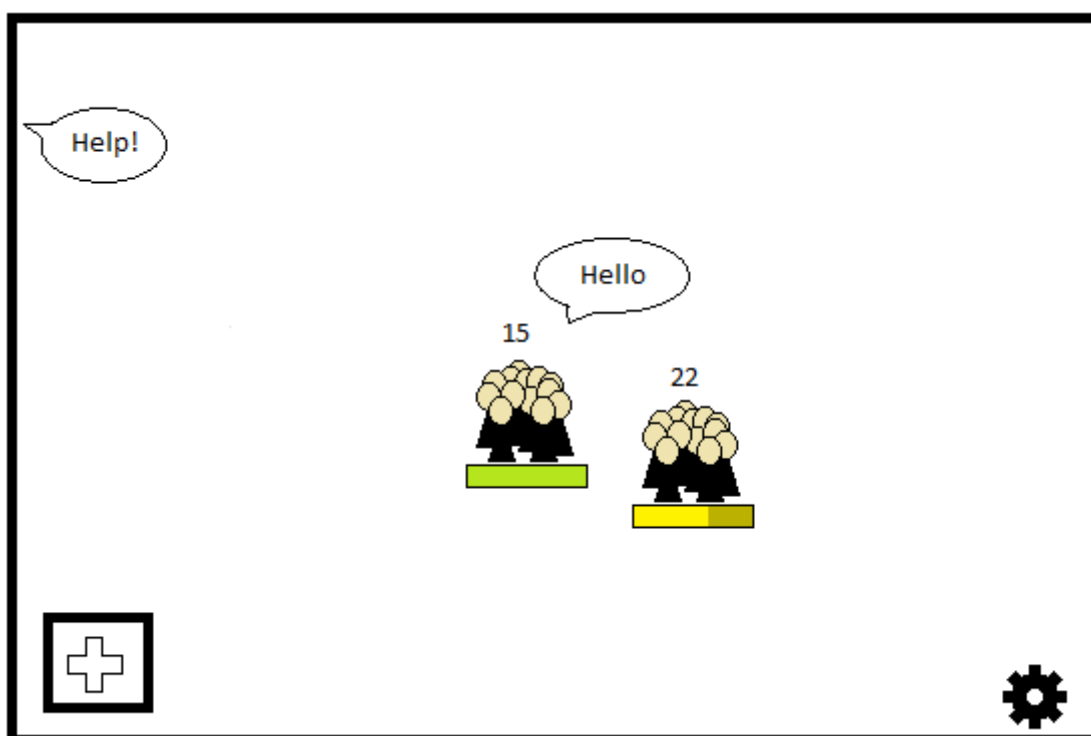


FIGURE 8. HUD mockup

REFERENCES

1. Introversion Software 2013. Prison Architect. Screenshot. Date of retrieval 9.11.2015. Available: <http://cdn.pcgamesn.com/sites/default/files/prison%20architect%20flash%20mob%20alskdn.png>.
2. Gender of the Day 2014. Cartoon cloud of dust with fists flying out. Gender of the Day. Date of retrieval 9.11.2015. Available: http://41.media.tumblr.com/6bd1b4df413edf317cc0dcbebcd19c6b/tumblr_nc7x180WOU1td3k3fo1_500.jpg.
3. Username Stratadrake 2011. Forum post. Tv Tropes. Date of retrieval 9.11.2015. Available: http://static.tvtropes.org/pmwiki/pub/images/resourcegraphics_ball-of-violence_6010.jpg.