

Requirements Analysis Document - Dumbit

Version: Alpha 0.2

Date: 13/5 2015

Author: Martin, Oliver, Alexander, Rasmus, Mansoor

Introduction

1.1 Purpose of application

The purpose of this application is to make a game that shall be interesting enough for people to want to pay for it. It shall be distributed online and people shall be able to get it in exchange for a fee.

As such, our application is required to be interesting which puts a high demand on creativity and making sure that the theme, style and music holds a certain high standard.

The target audience is the typical gamer and just about anyone who wants to have fun in his spare time.

1.2 General characteristics of application

This application shall induce an exciting feeling in the player and force her to challenge herself. By having a multitude of enemies and challenges, and different ways of tackling them, the characteristic of our application is that of a great time-killer. It shall require a quick trigger finger, a good sense of the physics of the game, and a good deal of experience in order to master this game.

1.3 Scope of application

The application shall contain:

- a single player campaign with story mode
- fast paced jump shooter
- experience gaining and item collecting
- character customization with different gear and appearance

The application shall not contain:

- multiplayer
- downloadable content (DLC)
- trademarked characters

1.4 Objectives and success criteria of the project

The primary objective is to get a functioning game with basic game mechanics in place so to make it possible to experiment with level design in order to find ways to make the game as enjoyable as possible. These mechanics combined with 5 simple levels, each containing at

least one “boss” are the criteria for this project to be deemed a success. The database for the items and creatures will be big enough for just 5 levels which is around 20 items and 10 creatures.

1.5 Definitions, acronyms and abbreviations

GUI - Graphical User Interface

Java - Programming Language

Boss - a computer controlled enemy that is hard to kill, usually at the end of a level.

Requirements

2.1 Functional Requirements

The application shall have physical simulations

The user shall be able to move the player left and right and jump at varying altitudes depending on how long jump-button is being pressed.

The user shall be able to aim and shoot at enemies using the mouse.

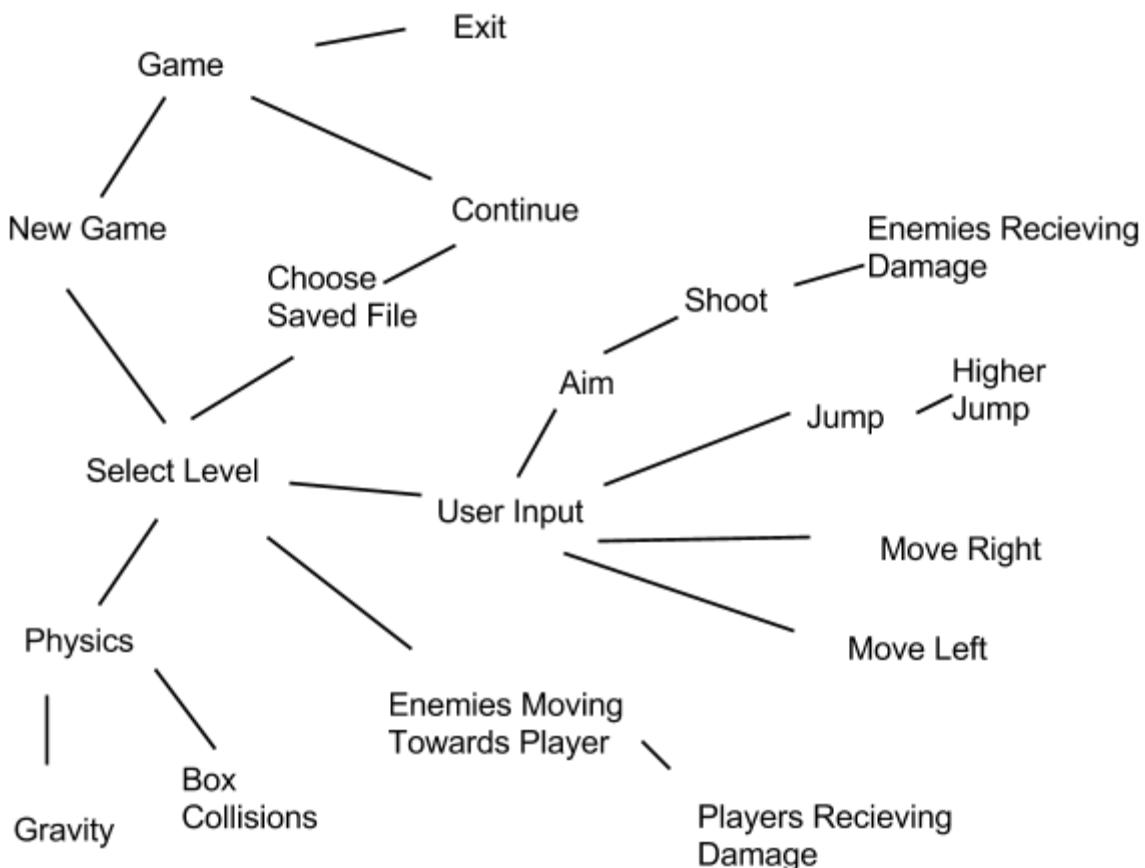
Enemies shall follow player around, damaging him as they get close enough

2.2 Non-Functional Requirements

Not Applicable

2.3 Application Models

2.3.1 Use Case Model



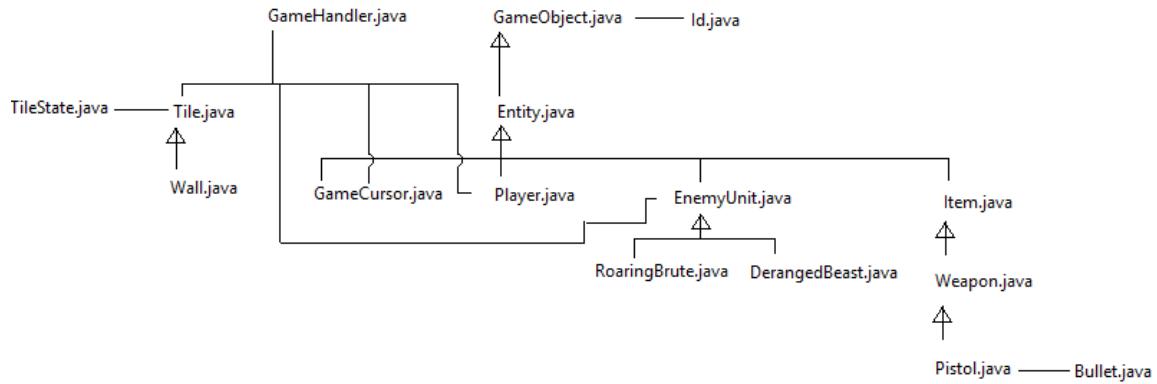
2.3.2 Use cases priority

High: Move left, Move right, Jump, Aim, Shoot

Medium: Higher Jump, Enemies Moving Towards Player, Players Recieving Damage, Enemies Recieving Damage,

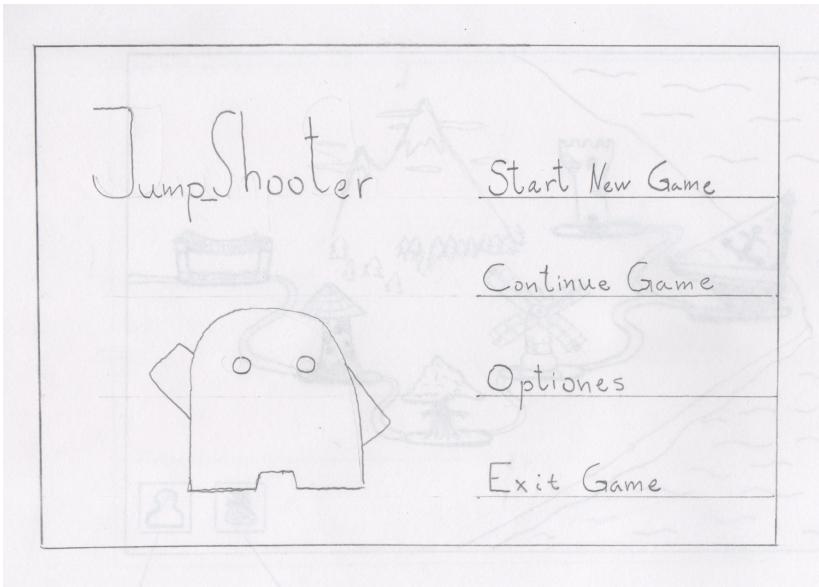
Low: New Game, Continue Game, Choose Saved File, Exit

2.3.3 Domain model



Early diagram showing relations between different parts of the game. The game is built upon two core classes, Handler and Tile. The Handler controls the underlying engine and runs the game. The Tile Class is the building blocks of the visual components of the game, including characters and their functionality such as AI-behaviour or Players Inventory.

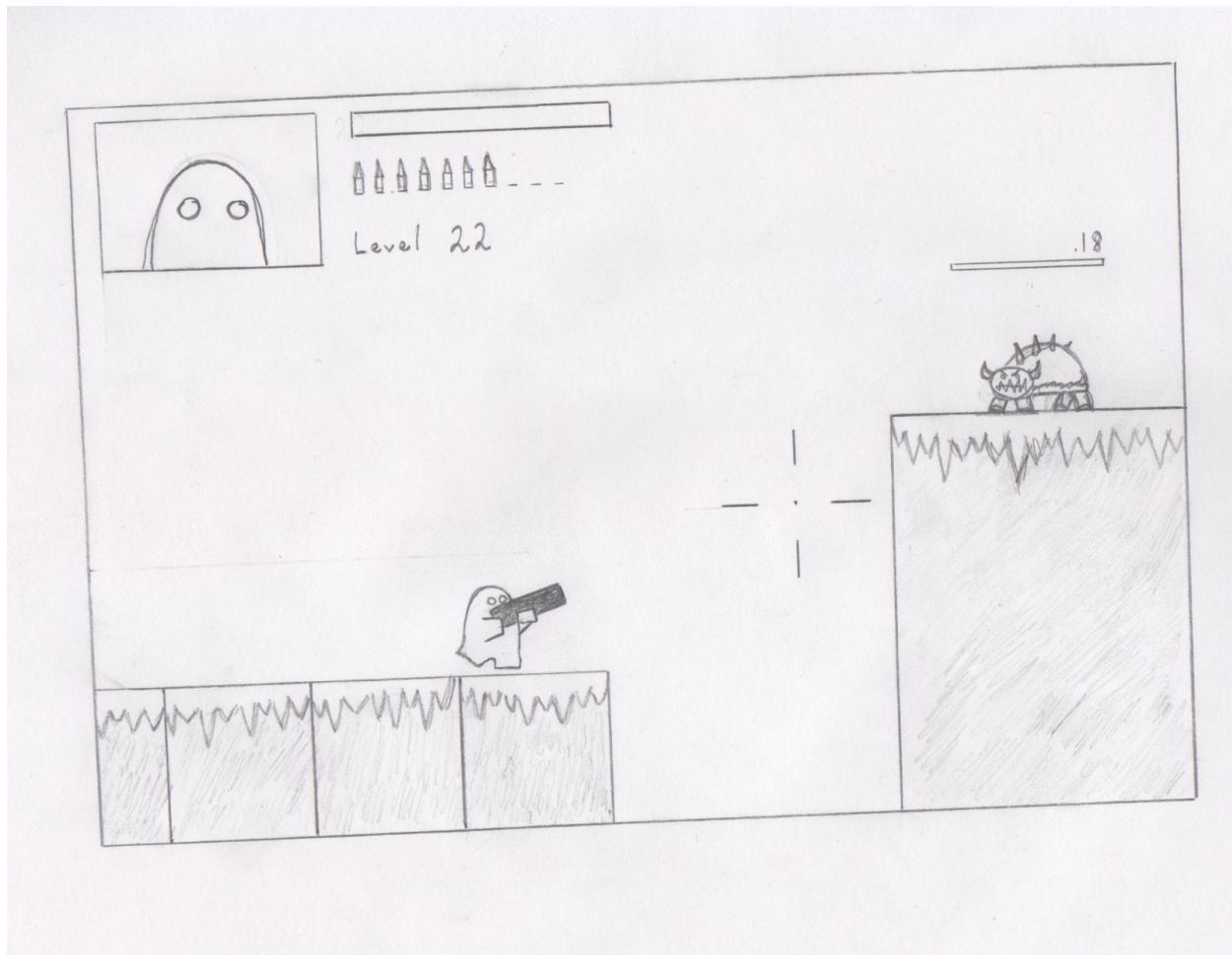
2.3.4 User interface



Start Menu
Providing you with the alternatives to start a new game, continue on a previous saved one, change some settings or exit the application.



World Map
Choose which level to play, or go to the shop to buy items or upgrade your character and view your inventory



First graphical view of actual gameplay. Stages contains of short to medium sequences of platform jumping and shooting action. Once stage is completed a window displaying result statistics in form of experience and currency gain will show. After the stage is completed the player will be returned to the world map.

2.4 References

Not Applicable

2.5 Appendix

For use cases text, please read the document titled “use case” in same folder as this document