System design document for Group 13

Table of Contents

Version:

Date

Author

This version overrides all previous versions.

1 Introduction

1.1 Design goals

The game logic and game design must be separated. All modules should be exchangeable and dependency should be kept low. The look and feel of the game is very important as is the responsiveness of the controls.

1.2 Definitions, acronyms and abbreviations

- LWJGL - Light weight java game library.
- Slick2d - A game framework based on LWJGL.
- GUI - Graphical User interface
- Java - Programming language
- Player - The person playing the game
- Tile - A small image containing specific properties
- Class - A file containing code
- Abstract Class - A Class which other classes work towards

2 System design.

2.1 Overview

The application will be implement MVC and run the slick2d game engine. The levels will be created using the TileD while the tiles and all entities will be created in Adobe Photoshop. All interactive objects will implement the abstract entity class. Several interfaces are to be created, such as IMoveable for entities that are supposed to be able to move like the player or enemies. The design focuses on a number of states that are interchangeable, each state holding a different view to the player. These are controlled by the GameController class.

2.1.1 Entities

Every interactable object in the game is an entity. An an entity contains basic information regarding the objects position and how it should look like, that is all. The object itself contains most of the more specific logic on exactly how it interacts with other objects in the game.

2.1.2 Level

Slick2D supports the .tmx (Tilemap) file format that TileD uses. The GameModel loads the specific Level. The level itself iterates the tilemap and interprets the different tiles based on their information. It then creates the specific objects for each tile and places them at the correct position, thus building the level. The model then moves the the entities based on the players movements to create a scrolling effect.

2.1.3 States
There are seven different states, excludeing Slick2D's BasicGameState. AbstractMachoDudeState is the abstract class which all the other states extend. It sets the basics like the background image and the window size. The MenuState class contains the menu where the player navigates to some of the other states. The options are:

- Start Game - Which sends the player to the Gamestate where the gameplay begins.
- Settings - Where the player can select the controls for the game in SettingState
- Stats - StatState displays information about the players statistics.
- Quit - Exits the game and prompts the player to keep playing in QuitState.

The last one is GameOverState which is shown to the player when their health has reached 0 or MachoDude has fallen into a pit.

2.2 Software decomposition

2.2.1 General

Every class in the application works towards an interface, as the UML diagram shows no classes have circular dependencies. The modules used are:

- Game - Holding the main method for starting the application
- Controller - Contains the different states and the controler for the states
- Entities - All the entities for the game are placed here
- Exception - The java exceptions used for error handling
- Level - This is the where the levels and the level abstract class is stored
- Model - The game Logic
- Particles - Code for creating all the particles used in the game
- Util - Utilities such as cameras and constans
- View - Containing code for the different views that are updated by the model

2.2.2 Decomposition into subsystems

The application itself is a subsystem to the Slick2D engine

2.2.3 Layering

The layering can be seen below,, the higher up in the document the higher up in the layering hierarchy.

2.2.4 Dependency analysis

2.3 Concurrency issues

The application is single threaded and thus there are no concurrency issues.

2.4 Persistent data management

The game does not include a save system and so player progress will not be stored. However some statistics will be saved and are placed ???????? The resources, such as graphics, maps and sound are filed accordingly in a project file.

2.5 Access control and security

2.6 Boundary conditions

Application is started as a normal desktop application and can wither be closed as one too or exited from the in-game menu.

3 References

APPENDIX