

Requirements and Analysis document for the Dimensions project (RAD)

Contents

1. Introduction

- 1.1 Purpose of application
- 1.2 General characteristics of application
- 1.3 Scope of application
- 1.4 Objectives and success criteria of the project
- 1.5 Definitions, acronyms and abbreviations

2. Requirements

- 2.1 Functional requirements
- 2.2 Non-functional requirements
 - 2.2.1 Usability
 - 2.2.2 Reliability
 - 2.2.3 Performance
 - 2.2.4 Supportability
 - 2.2.5 Implementation
 - 2.2.6 Packaging and installation
 - 2.2.7 Legal
- 2.3 Application models
 - 2.3.1 Use case model
 - 2.3.2 Use cases priority
 - 2.3.3 Domain model
 - 2.3.4 User interface
- 2.4 References

APPENDIX

- A. Use Case Model
- B. Domain Model
- C. GUI Sketch

Version: 1.0

Date: 19 may 2013

Authors: Simon Bengtsson, Carl Fredriksson, Kim Kling och Kim Egenvall

1. Introduction

1.1 Purpose of application

The proposed application is Dimensions, a classical 2D “runner game” which in its simplest form is about bringing the player through levels consisting of numerous obstacles. The application is diversified by, and gets its name from the change of dimension that occurs when the player takes a dimension change item. For a better overview of the game, see sketches in references.

1.2 General characteristics of application

The application will be developed primarily for android with focus on simple mobile controls. However there will also be a version of the game runnable on the desktop. The player is moving with a constant speed through the levels if not stuck by a platform. The challenge is to avoid obstacles by either jumping or moving sideways depending on the current dimension. Example of obstacles could be missing ground, spikes or lava. When the dimension changes the player has to quickly reorientate to successfully be able to finish the levels. When a new level starts the dimension shown will most likely be an XY view, similar to Super Mario and other 2D games, but every level could be different.

1.3 Scope of application

It should be easy to understand how to play to make it appealing to as many people as possible. There will be only one way to control the player. The action performed when using that control will change when the dimension changes in the game however. In one dimension it will make the player jump and in the other, make it change direction.

1.4 Objectives and success criteria of the project

It should be possible to play in a story mode where the user can play levels in order and after finishing one, unlocking the next one. The player should also be able to select a level to play among finished levels. One level should include being able to jump and move sideways to avoid obstacles and take items.

1.5 Definitions, acronyms and abbreviations

- GUI, graphical user interface.
- Java, platform independent programming language.
- JRE, the Java Run time Environment.

2. Requirements

2.1 Functional requirements

The player should be able to:

1. Choose game mode
2. Start a new game
 - a. Jump over obstacles or to platforms
 - b. Navigate to avoid obstacles
 - c. Get item-boosts (speed, power etc.)
 - d. Die when hitting obstacle or the player is positioned off screen.
3. Pause the game
4. See the high scores
5. Exit the game

2.2 Non-functional requirements

2.2.1 Usability

The usability is essential. A new user should be able to start a new game and after seeing hints displayed during an intro level, understand how the controls work. Tests will be performed to ensure usability. The language used is English and we have no intentions in providing translations. Primarily because the text is sporadic and controls should be simple enough to understand with a very basic knowledge of English.

2.2.2 Reliability

NA

2.2.3 Performance

The player's actions should be given a response almost instantly.

2.2.4 Supportability

We are going to support android and desktop devices, but the application should be made in such a way that it is easy to port it to other platforms, for example iOS.

There will be automated tests for model classes. The GUI in the game will be tested manually.

2.2.5 Implementation

To be able to make the application available on multiple platforms we will be using the java game library LibGdx. It supports desktop, web, android and iOS.

2.2.6 Packaging and installation

The game will mainly be distributed on the Google Play app store. However, for playing the game on desktop there will also be a an executable jar file available for download that can be run directly.

2.2.7 Legal

No legal issues foreseen.

2.3 Application models

2.3.1 Use case model

See appendix A.

2.3.2 Use cases priority

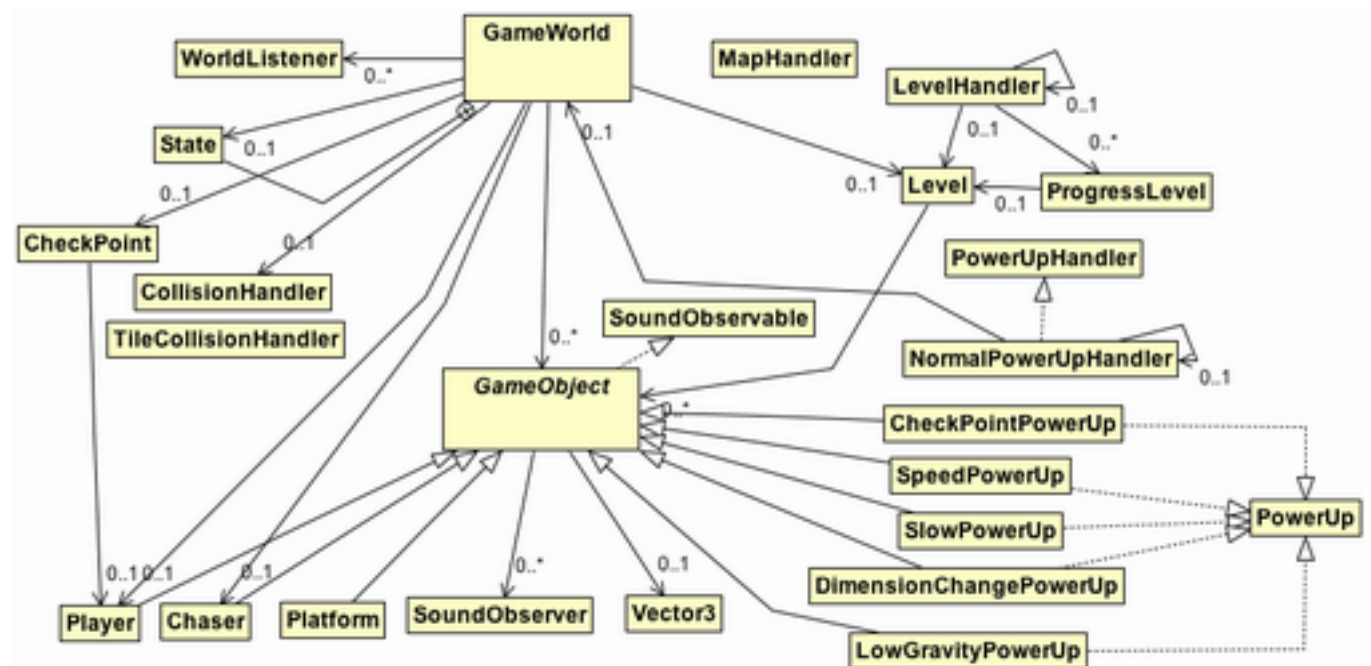
1. Play level
2. Move
3. Jump
4. Die
5. Start new game
6. Take items
7. Pause and resume
8. Play specific level
9. Reach checkpoint

2.3.3 Domain model

See appendix B.

2.3.4 User interface

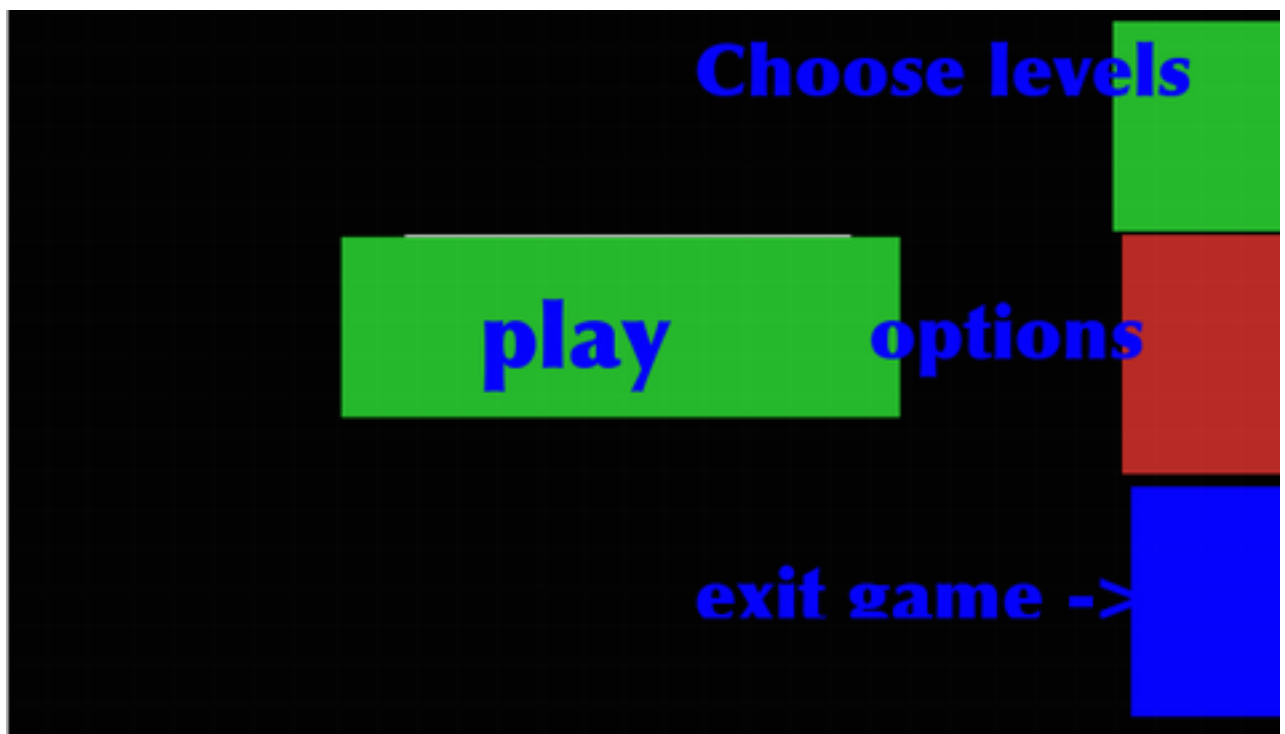
On a android device the game will enforce landscape orientation. When playing the game the entire screen will show the actual level. The entire screen is also a touch area that recieves input when the user touch it - for example a jump in one dimension. For a sketch see appendix C.



C. GUI Sketch



Ingame capture of GUI



Early sketch of main page of application