

Requirements and Analysis Document for EGA

Version: 1.0

Date: 2015-03-31

Author: Emil Axelsson, Erik Karlkvist, Hampus Rönström, Rebecka Reitmaier.

1 Introduction

The project is about creating a two dimensional platformer. It is supposed to be a fun game for everyone to enjoy and spend time with when there is nothing else to do. In order to give the game its unique feel the character can grow and shrink. This opens up a dozens of potential puzzle for a user to solve. It should take some effort to complete each map, but you will be rewarded for it.

1.1 Purpose of application

Create a fun platforming game for everyone to enjoy.

1.2 General characteristics of application

It's a two dimensional platformer with a character the user controls. It is based around puzzle solving, jumping and avoiding dangers. The character can collect stars to grow or shrink. This creates a set of challenges the user has to go through to win the game. In order to achieve a higher score the user has to finish each level in a short time, however this is optional.

1.3 Scope of application

It is singleplayer only. There is no networking and it is a standalone desktop app run by Java.

1.4 Objectives and success criteria of the project

There is at least two levels. The character is able to grow and shrink. Time is counted and presented at the end of each level to represent a score. There is at least one dangerous obstacle. The character is able to move in x- and y-axis.

1.5 Definitions, acronyms and abbreviations

- The user - the real life person that plays the game
- Character - the center of the game, a player controlled object
- Level - contains a map and everything else visible to the user

- Map - the layout where the character moves
- Stars - an object the character picks up to grow or shrink
- Exit-door - the end of the map, this is where the user should aim to reach
- Spikes - dangerous objects, if touched resets the character to the start of the map

2 Requirements

2.1 Functional requirements

- Start
- Pause
- Setting
- Levels Menu
- Select level
- Volume Control
- Move
- Jump

2.2 Non-functional requirements

2.2.1 Usability

It should be easy to understand for new users and intuitive. The menus are logical and resemble already known menus. In order to achieve these goals tests will be performed by non computer professionals.

2.2.2 Reliability

The game is as bug free as possible. As in the usability this will be tested by non computer professionals. It will also be tested by testing tools.

2.2.3 Performance

The game is fast and responsive. The goal is to never annoy the user by unnecessary clicks or sluggish movements. The user is always in control and the character does not do actions the user has not asked for.

2.2.4 Supportability

N/A

2.2.5 Implementation

The application will use Java JDK 1.8, Gradle, and be supported by the libGDX library.

2.2.6 Packaging and installation

Every user is required to have Java JRE installed to play. They have to download the game, containing all pictures and music, in order to play it. Apart from this there is no installation required and they will only need to launch the exe file.

2.2.7 Legal

N/A

2.3 Application models

2.3.1 Use case model

See appendix 1.0 Use cases.

Nouns:

- Player
- Menu
- Level
- Block
- Obstacle(spikes)
- Star(big/small)
- Controller
- Exit-door
- Ball

2.3.2 Use cases priority

Highest

- Move
- Jump

Mid

- Start
- Level selector

Low

- Settings
- Sound

2.3.3 Domain model

See appendix 2.0 UML

2.3.4 User interface

Text to motivate a picture.

2.4 References

APPENDIX

1.0 Use case

2.0 UML

GUI

Domain model

Use case texts