Requirements and Analysis Document for the Life Of A Goblin project (RAD)

Table of contents

Version: 1.0
Date: 2015-03-30
Author: Jesper Olsson, Fredrik Bengtsson, Ulrika Uddeborg, Anton Strandman
This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

The goal of the project is to create a platform-based sidescroller game, for entertainment purposes, aimed at various types of players.

1.2 General characteristics of application

The application will be a desktop, standalone single player game for the Windows/Mac/Linux platforms. Based on our personas, the game will be live action and story-driven with a lighthearted fantasy feeling.

The application will have a set goal with a linear progression. The player will progress in the game by defeating enemies and solving puzzles.

The game will take place in a 3D environment but will be played in 2D. The game will include a start menu and an in game pause menu

1.3 Scope of application

The game will be a single player game with a limited amount of levels, playable characters and will include several game modes. The game will facilitate high scores, achievements and rewards. It will be possible for the player to save and load their progress. At startup a main menu will provide the player with options to start a game, edit settings and manage saved games.

The player will have a finite amount of health which can be reduced by triggering traps or upon being attacked by hostile entities. When the player runs out of health the player character will reappear at a previous point in the game.

The game will NOT:

- Facilitate any form of local or remote multiplayer.
- Support touch.
- Support any kind of controller apart from a regular mouse and keyboard.

1.4 Objectives and success criteria of the project

The objective of the project is to create a playable game with multiple levels and four characters. The game will store high scores and saved games locally. The game will have a main menu from which the player may start a game, change settings and manage saved games.

The success criteria of the project is to deliver a playable game with at least one level and one character.

1.5 Definitions, acronyms and abbreviations

2D	Two-dimensional, typically up/down, left/right.
3D	Three-dimensional, up/down, left/right, in/out.
GUI	Graphical user interface.
Java	Platform independent programming language.
JRE	The Java Runtime Environment. Additional software needed to run a Java application.
Player	The character directly controlled by the player and the main focus of the game.
NPC	Non playable characters, enemies and possible allies.
Background	The parts of the GUI the player cannot in any way interact with and which does not display any direct information about the state of the character or the level.
Level	A course which the player will have to traverse.
Sidescroller	A 2D game in which the player traverse a linear level, typically from left to right.
Checkpoint	A point in the game where progress is saved.
Suspension of disbelief	When the observer does not reflect upon the fact that what he/she sees is not real.
Player status	Displayed information concerning the state of the player, examples include health, mana or equipment
GUI	Graphical User Interface.
HUD	The part of the GUI displaying player status and possibly information about the level.
POV	Point of view, the part of the level currently visible to the player.
Interface	Everything the user can interact with.

Response time	The time it takes for the application to react to user input.

2 Requirements

In this section we specify all the requirements of the project.

2.1 Functional requirements

The player should be able to:

1. Start a game
2. Move backwards and forwards in a level
    a. Move backwards (left)
    b. Move backwards (right)
3. Move up and down
    a. Fall (move down)
    b. Jump (move up)
4. Die
5. Respawn
6. Fight hostiles
7. Interact with items
8. Change settings
9. Manage saved games

2.2 Non-functional requirements

2.2.1 Usability

The game should have controls that are easy to understand for all new players, even to players who are unused to this type of game. The player can change the controls through a settings menu. The game HUD will provide an overlay text which explains the controls the first time the game is run, to aid new players.

The game will be played with mouse and/or keyboard.

During gameplay the game should clearly show the status of the player (health, mana, other stats).

2.2.2 Reliability

The release version of the game should be a stable version in which the game may be completed without crashing.

2.2.3 Performance

The game is supposed to be a live action game for which a very short response time is required during play. The game should react to any user input within 50 ms.

For loading times, such as loading a new map, longer response time is accepted. However, the loading time should not exceed 1 minute on a “modern laptop”.

2.2.4 Supportability

The developers of this game will not provide any support for this game post release.

2.2.5 Implementation

To achieve platform independence the application will use the Java environment. All hosts must have the JRE installed and configured. The application needs to be downloaded to all computers where it will run.

2.2.6 Packaging and installation

The application will be delivered in a zip-archive containing;

1. A file for the application code (a standard jar-file)
2. All needed resources, internationalization and localization files icons etc.
3. Start programs (scripts) to start the game on the different platforms.
4. A README-file documenting installation and start of the application.

2.2.7 Legal

We (the developers of the game) take no responsibility of the application. All interaction with the game and/or any of it's files are on the user's own risk. We are not liable for any damage which may be caused by the application or received when using the application.

This is an open source application, anyone is free to take and use the code provided.

2.3 Application models

2.3.1 Use case model

See APPENDIX for UML diagram and textual descriptions.

2.3.2 Use cases priority

1. Move (left/right).
2. Jump/fall (up/down).
3. Death.
4. Reincarnate.
5. Fight
    a. Destroy hostile entity
6. Take damage
7. Pause game
8. Pick an item in a menu
9. Interact with object
10. Complete a level

2.3.3 Domain model

2.3.4 User interface

The game will take place in a 3D world, but will be played in 2D with an overlaying lightweight HUD displaying player status. When the game is launched the player will be faced with a start screen displaying a menu. The menu will facilitate the start game options as different buttons and an options menu button containing a list of settings for screen resolution, controls and sounds.

2.4 References

See appendix.