Software Requirements Specification

for

Flash

Version 1.0.1 approved

Prepared by Sahil Sharma
Darshit Patel
Subham Gupta
Kuldeep Singh Palawat

Software Engineering – CSE Dept., IIT Guwahati

19/02/2016

Table of Contents

Table of Contents i
1. Introduction 1
1.1 Purpose 1
1.2 Intended Audience and Reading Suggestions 1
1.3 Product Scope 1
1.4 References 1
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Features 2
2.3 User Classes and Characteristics 2
2.4 Operating Environment 2
2.5 Design and Implementation Constraints 2
2.6 User Documentation 2
2.7 Assumptions and Dependencies 2
3. External Interface Requirements 3
3.1 User Interfaces 3
3.2 Hardware Interfaces 3
3.3 Software Interfaces 3
3.4 Communications Interfaces 3
4. System Features 3
4.1 Game Display 3
4.2 Tracks 4
4.3 Cars 4
4.4 Controller 4
4.5 Keyboard Inputs 5
5. Other Nonfunctional Requirements 6
5.1 Performance Requirements 6
5.2 Safety Requirements 6
5.3 Security Requirements 7
5.4 Software Quality Attributes 7
5.5 Business Rules 7
Appendix A: Glossary 8

1. Introduction

1.1 Purpose

The purpose of this document is to provide comprehensive requirements and goals for the 2d car-racing video game. This document helps the reader to visualize the intent of the developers and how they expect the program to run and react to input from the player. It will provide all requirements for the game. This game is intended to provide entertainment to players through strategic game-play involving intense combat and limited resources.

1.2 Intended Audience and Reading Suggestions

The priority of certain features and requirements will be identified using tags in the beginning of the feature/requirement's description. While some features may be listed earlier than others, this is merely the nature of the document and as such there is no implicit emphasis or priority in the order presented. Any priority a feature may be assigned by the developers shall be made explicit in the feature description.

This document may make references to a "project," "software package," or "game." These and any similar phrases shall be made in reference to the game, the project which this document describes. Any terms the reader may not be familiar with may be defined with footnotes on the page that they first appear.

1.3 Product Scope

This software will be a stand-alone desktop video game that will be used for entertainment purposes. The goal of this project is to provide a user with a leisurely but challenging gaming experience that will grab and hold their attention and interest as well as intellectually stimulate them. The game will be two-dimensional with a simplistic graphics engine and feature a robust turn-based combat system. This software is intended to be delivered as a freeware, open-source product to interested players, with the intent that savvy users could modify the source code to suit their own needs, so long as they provide attribution to the original creators, the authors of this document.

1.4 References

- Unity asset store: http://unity3d.com/unity
- Unity tutorials: http://unity3d.com/learn/tutorials
- The C# Resources Network: http://www.tutorialspoint.com/csharp/
- Unity scripts documentation: http://docs.unity3d.com/ScriptReference/

2. Overall Description

2.1 Product Perspective

Our Game is an open source project done for our CS243 lab. Our Game is 2-D racing game with certain 3-D elements. In this game, the user faces difficulties whose level keeps increasing. User selects his car and a track to race upon. The objective is to unlock all the tracks and cars in the game.

2.2 Product Features

- Graphics
- Character and Class progression system
- Procedurally Generated Environments
- Various difficulties level
- High score storage
- Help system

2.3 User Classes and Characteristics

Expected users are anyone with a desire to play videogame and more specifically those who enjoy playing racing games. Any user with or without experience of a racing game can play our game. Controls are also available in the game for help.

2.4 Operating Environment

Our game is developed on Unity game engine using C# for writing scripts. Our game shall be playable on Windows, Mac and Lenux. Our software requires the use of a keyboard only.

2.5 Design and Implementation Constraints

The game is written in unity and hence many of its components are taken from unity asset store. English is used for all the linguistic work. To make the game light, very few 3-D objects are used. We will design it so that game can run at high fps.

2.6 User Documentation

The game is very basic therefore no manual is available for the game. There will be an option for game controller that the user can select that will explain the basic commands to be used to play the game before the start of the game.

2.7 Assumptions and Dependencies

We assume that the user's computer has the ability to run C# scripts, in addition we assume that user's computer can run and render basic .jpeg, .gif and .png files dynamically to facilitate the playing of our game. It is also assumed that user understands English since all text contained in the game is written in English.

3. External Interface Requirements

3.1 User Interfaces

The user will play the game via their monitor and the standard "windowed" display. The main menu will have options to start a new game, load a previously saved game, change user controls or exit the game.

- The game shall be able to port to the monitor.

3.2 Hardware Interfaces

The keyboard's arrow keys will move the car, and other keys will be used to use power-ups (tentative). However, the user can change keyboard controls at discretion. Also, mouse inputs should be available as the default standard, and Esc key kill be used to pause the game and access the menu at any time. Errors will be displayed to the monitor, and hence, the user is expected to use the keyboard and the mouse keys effectively.

- The game must be able to read inputs from the keyboard.
- The game must be able to read inputs from the mouse.
- The game must be able to effectively interpret the inputs.

3.3 Software Interfaces

The game does not require any specific libraries or packages to be able to run effortlessly, but, in order to load previously stored games, saved game data of previously played games must be accessible.

3.4 Communications Interfaces

This game does not require any active Internet connections or inter-device communication.

4. System Features

In this section, system features will be described so as that user can easily understand, text or expand the program.

4.1 Game Display

4.1.1 Description

Player can view his car and the racing track with other car/cars.

4.1.2 Stimulus Response System

The display will appear when the game is initiated.

4.1.3 Functional Requirement

[Req]- The game shall display player stats, tracks and other cars during game play.

4.2 Tracks

User can select a track from many tracks to race on. By default, only one track is unlocked.

4.2.1 Difficulty

Tracks difficulty level increase as user unlocks them.

4.2.2 Stimulus Response Sequence

The player can unlock these tracks by completing milestones in the game.

4.3.3 Functional **Requirement**

[Req]- The game shall provide the player the mean to unlock these tracks.
[Req]- User should be able to see all the locked and unlocked tracks.

4.3 Cars

User can select his racing car from multiple cars. By default, only one car is unlocked.

4.3.1 Unlock

User can unlock a car by completing a milestone.

4.3.2 Features

Cars with different speed and brakes are available to be unlocked.

4.3.3 System Stimulus Response Sequences

The player upon starting a new game can navigate a series of cars and bu completing the milestones can unlock new cars.

4.3.4 Functional Requirement

[Req]- The game shall provide the player the mean to unlock these tracks.
[Req]- User should be able to see all the locked and unlocked tracks.

4.4 Controller

User can control various aspects of game by this.

4.4.1 New Game

User can start a new game using this.

4.4.2 High Score

User can see the high score made till that time.

4.4.3 Sound

User can change the volume of the game.

4.4.4 Controls

Controls are shown here for the racing.

4.4.5 Stimulus/Response Sequence

Upon launching the game, users shall have the option to launch a new game or to change certain settings or watch the key mappings.

4.4.6 Functional Requirement

[Req]- If the player clicks on any option the action shall execute.
[Req]- This shall provide the user the opportunity to learn the actions associated with each key.

4.5 Keyboard inputs

4.5.1 Description

Many keys including the arrow keys will be utilized to control the interaction between the player and game.

4.5.2 Stimulus/Response sequence

When the user presses a key, a corresponding action will be undertaken by the player's character. The keys and the actions which have been mapped to them is explained in the controller.

4.5.3 Functional Requirement

- When a player presses an action key, the game shall carry out the appropriate action.

- The game shall provide the player with the opportunity to learn the action associated with the key.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1. Frame rate*:* The game must be able to run at 50-60 frames per second on Windows platform OS. At this frame rate, the game will remain constant and playable. It will not be technically demanding and able to run on lower end computers.

5.1.2. Executable Size: The size of Game's executable will be less than one gigabyte. The entire libtcod library is less and the game is programmed in C#, so this is not an unreasonable limitation.

5.1.3. In-game Load Times: Game's load time between levels will be near instantaneous. Game's is graphically minimalistic so loading new levels will not be time consuming. .

5.1.4. Launch Time: The time to reach the main menu upon launching the game will not exceed 10 seconds

5.2 Safety Requirements

5.2.1 External System Integrity: The game will not be able to open, create, edit, or delete any nongame related files and will run in isolation to other unrelated processes that are operating concurrently. This will help to prevent the game from causing any damage to the computer or its internal components.

5.2.2 Save File Integrity: Upon a game crash, user's progress will be reverted to the earliest save prior the crash. This will minimize the loss of user's data that occurs after a crash and the implementation of auto-saves additionally aid to minimize data loss.

5.3.3 Other Safety: The system should warn the use to take a break after every two hours continuous play to prevent eyestrain and repetitive strain injury. The main safety concern that we will be dealing with is the possibility of inducing seizure in the user. Some autistic children are very susceptible to flashing lights and this can be problematic. To safeguard against this we will try to keep flashing lights and rapid screen movements down to a minimum. We will not flash the screen or change colors rapidly when it is unnecessary to do so.

5.3 Security Requirements

Save files used in the game will be compressed so they become illegible to users. To essentially create a black box which prevents users or other parties from manipulating save files to prevent possible cheating. Security will not be a high concern for this project since there is no sensitive data being stored.

5.4 Software Quality Attributes

5.4.1 **Mapping System**: The mapping system will have pre-made maps. This allows for adaptability of the game play and reusability since players will always be able to experience a Better gaming environment.

5.4.2 **Character Variety:** There are options between different choices of races, cars, and maps. Users will have new experiences when replaying the game with these choices, since their choices will influence their playing strategies, which is something desired by the developers.

5.4.3 **Tutorial and Help System:** There shall be a tutorial/FAQ, manual/help system. Usability can be seen in this since this will help teach new user the controls for Game, what each Power-up do, how the items helpful, etc. This would be most helpful to players who have never played this.

5.4.4 **Keyboard Use**: Multiple keys are not pressed at a time, and time is not a factor since nothing in the game happens unless a turn is taken. Key mapping, the ability to change what a key does in the game, will be a feature. This is a feature of most racing-games and will allow for those who prefer one set of keys to play the game user-friendlier. This is something that will allow customizability for the user.

5.5 Business Rules

Roles will be shared among the members of the group. Specific tasks will be delegated throughout the development process. Each will have different roles and parts of the product they will have to work on and maintain. These can include

- Art and Graphics
- Map & Rendering
- Item/Inventory
- User Interface
- Testing

Any major decision in terms of the game development must be discussed and decided among the members of the group. There must be a general/majority consensus for decisions that have resulted in conflict.

Appendix A: Glossary

[Req]- Required

2-D – 2 dimensional

unity – game development engine