Software Requirements Specifications Document
GameBasedLearning
Project-4

150101027- I.N.Dilip Kumar
150101051- Hareesh Reddi
150101032- L.Sai Shobith

February 19, 2017

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to present a detailed description about the Game Based Learning. This document will explain the purpose and features of the system, Interfaces of the system, Various constraints under which the system has to operate and requirement analysis which includes both the functional and nonfunctional requirements along with external interface requirements and performance requirements.

1.2 Definitions

Term	Definition
Game Engine	A game engine is a system designed for the creation and development of video games
Android	Android (Google product) is a Linux-based operating system
SRS	Software Requirements Specification
UI	User Interface
Gamer	A person who plays a game or games, typically a participant in a computer or role-playing game
System	A system is a set of interacting or interdependent components forming an integrated whole or a set of elements (often called components) and relationships which are different from relationships of the set or its elements to other elements or sets.

1.3 Scope of the Project

The Mobile App will be a platform especially for students who are pursuing their under graduation. This app will help them understand the sorting algorithms like Insertion sort, Quick sort and Bubble sort in a more efficient way which is in a game based learning platform.

The main intention of ours is to provide the users/students with a mobile app through which the user will be able to increase their extent of un-derstanding about the algorithm and therefore can improve their clarity about these concepts. More specifically the student/user will be able to improve his/her performance by looking at their Statistical data which is provided by the app.

1.4 Intended Audience and Reading Suggestions

The SRS document gives project managers a way to ensure the games adherence to our original vision. Although the document may be read from front to back for a complete understanding of the project, it was written in sections and hence can be read as such. For an overview of the document and the project itself, see Overall Description.

1.5 Overview of the Document

The first chapter contains the Introduction part which gives a basic idea about what the project is going to be and provides the definitions which will be used throughout the document and also the basic definitions given by the IEEE Computer society.

The second chapter will give the overall description which contains all the interfaces like system interface, user interface, software and hardware interfaces in a detailed manner followed by Design constraints, Product function, User characteristics and will be ended with the Memory constraints.

The third chapter will have the system requirements which include functional requirements and non-functional requirements like security, maintainability, portability.

1.6 References

1. https://en.wikipedia.org/wiki/Software_requirements_specification#Structure
2. Software Engineering: A Practitioner's Approach, Roger S Pressman - sixth Edition
3. http://www.nptel.ac.in/courses/106105087/
4. https://www.cise.ufl.edu/class/cen3031sp13/SRS_Example_1_2011.pdf
5. IEEE SRS Template

2 Overall Description

2.1 Productive perspective

The app is supposed to be an android based game highly useful for students especially undergraduates. The product is independent and self-contained. The following are the main features:

- This app provides a very efficient way to understand the sorting algorithms.
- This app gives a statistical report at the end of completion of each game, so the user can know whether he is improving or not.
- The student can play and learn during the lecture or outside of the college hours.
- The teacher can also use this as a supplementary for teaching sorting algorithms.

2.1.1 User Interface

- Every game must have a menu so it is easy for the user to see everything at one spot.
- The user on opening the application can use the various utilities of the app like Start Game, Highscores.

2.1.2 Hardware Interfaces

- Touch screen
- Speakers to generate sounds

2.1.3 Software Interfaces

- Product should run on android versions 4.4 Kitkat and after
- Product is preferably made on Unity-3D, Blender using C#.
- C# programming language: C# is a hybrid of C and C++. It is an object oriented programming language used with XML- based web services on .NET platform. It's also used by several type of system for building other applications. Examples that use C# are visual studio, Unity etc.
- Unity-3D: Unity is a cross platform game engine developed by Unity Technologies and used to develop video games. We will use Unity-3D to build the game and then publish it to Android.

- Blender: Blender is an open-source 3D graphic software product used to create animated video, 3D printed model, interactive 3D application and video games.
- SQLite for database

2.1.4 Communication Interface

The communication between the different parts of the system is very important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems. There will not be any communications between this app and the other apps since it is completely independent and self contained.

2.2 User Characteristics

There is only one user at a time in this software and the user interacts with the game/system. Therefore, the user is the only one who communicates with the system through playing the game. In addition, this gamer can be any person. The only primary requirement is that, the player who is playing must read the playing procedure and instructions provided by the developers.

2.3 Assumptions and Dependencies

One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example, the users might have allocated them with other applications; there may be scenarios where the application does not work as intended or even at all. Another assumption is that the user who uses this app is acquainted with enough knowledge of how to use an android mobile phone.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 Player Name

Input: String (Name of the player)
Output: Player name is stored for later usage.
Description: The user who is playing the game enters his/her name. The final score of the game played by the user is saved in the high scores database with this name.

3.1.2 Start Game

Input: Input wil be taken from the touch screen
Output: Game will be started
Description: The system shall allow the player to start a game.

3.1.2.1 Random Number Generator

Input: Number of two digit numbers to be generated
Output: Two digit numbers in unsorted order.
Description: The function gives the requested number of two digit numbers in unsorted order, so that user can sort them.

3.1.2.1.1 User Moves Correctness

Input: The move performed by the player
Output: Score is increased if the move is correct and score is decreased if the move in incorrect.
Description: This function determines whether the player performed a correct or incorrect move and increases/decreases the score in score board accordingly.

3.1.2.1.2 Three Consecutive Wrong Moves

Input: Last three moves
Output: Restarts the game if not all the given moves are correct.
Description: If the player performs three consecutive wrong moves then the game is restarted.

3.1.2.2 Pause Game

Input: Input will be taken from the touch screen
Output: Time counter stops. User cannot make any moves in the game.
Description: A player can pause a game, so that he can resume it afterwards.

3.1.2.2.1 Resume Game

Input: Input will be taken from the touch screen
Output: Time counter starts. User can make moves in the game
Description: A player can resume a paused game.

3.1.2.3 Restart Game

Input: Input will be taken from the touch screen
Output: Starts a new game
Description: The player can restart in the middle of an ongoing game or when there are three consecutive wrong moves, the user will be asked to restart the game.

3.1.2.4 Stop Game

Input: Input will be taken from the touch screen
Output: Stops the current game.
Description: The current game the user is playing is aborted.

3.1.2.5 Time Counter

Input: None
Output: Duration of the game played
Description: The total time taken by the player to complete a game is calculated

3.1.3 Game Over

Input: The array of numbers
Output: Stops the game
Description: The game is over if the numbers are in sorted order.

3.1.3.1 Score Generator

Input: No of wrong moves, number of right moves and time taken.
Output: Total score of the player in the game
Description: This function calculates the final score by considering number of wrong moves, number of right moves and the time taken by the player to finish the game and gives the average time to complete the game.

3.1.3.2 High Scores

Input: Total score of the player in the game.
Output: Score is stored in the game if it is high enough
Description: If the score of the player is higher than at least one of the high score or the high score, database is not full then it is stored in the high scores

3.1.4 Exit

Input: Input will be from the touch screen
Output: The game application ends.
Description: The game app stops running.

3.2 Performance Requirements

- The minimum frame rate must be twenty frames per second. The average frame rate must be greater than thir-ty.
- The game must be able to run with a minimum of 512 MB of RAM.
- The average response time between click and reaction must be less than 0.5 seconds. The maximum response time between click and reaction must not be greater 2 seconds.

3.3 Non-Functional Requirements

3.3.1 Reliability

As the system provide the right tools for discussion and problem solving, it must be made sure that the system is reliable in its operations.

3.3.2 Usability

The app is easy to handle and navigates in a most expected way with no delays. A user help guide is also constructed to give the user a complete documented version of what the app is and how can he/she make it useful to the extreme levels.

3.3.3 Security

The data which is generated in the app is not necessarily need to be secured. because the data generated will be like the Name of the player, number of restart he/she has used in the game, Average time needed to complete the game. So, it is not needed to secure this data and can be viewed by anyone who opens the app. But the data generated in this app cannot be accessed from another apps.

3.3.4 Maintainability

- The app should ask to send the bug reports and crash reports to the developer which would help in maintenance.
- The app should ask for feedbacks so as to improve the app in next updates.

- The app shall be updated frequently.
- The source will be properly commented such that it may be maintained by others in future.

3.3.5 Portability

The app should be easily ported into any other device of android platform.An .apk file will be made available to install the application on that device.

3.3.6 Extensibility

The app should be made such that it may be extended to a larger system which has more number of sorting algorithms.