Software Requirements Specification

for the Guess What Game

Revision A

25th April 2021

Table of Contents

1. Introduction

1.1 Scope

This specification establishes functional, performance, and development requirements for the first iteration of the WordRacer game app.

1.2 Purpose

The purpose of this project is to build an interactive computer game with a graphical user interface that is both entertaining and educational for children. The game is a way for children to have fun while learning about new things using image recognition and hints in an attempt to guess the word that is paired with an image.

1.3 Intended Audience & Use

This project is intended for use by younger children (especially ages 4 - 8). Aiming to provide both a source of education and entertainment for children.

2. Overall Description

2.1 Product Perspective

Play has always been a central and integral part to growing up. These days tablets and computers are in nearly every student's hand at home and in the classroom. The gamification of education has been on the steady rise in the 21st century but has especially boomed in the last decade. The whole education system has basically become a game as schools, teachers and students compete with their peers and themselves to be the highest ranked. This virtual gamification of education is increasingly used as children get older and have their own devices at younger ages. We see games now that even infants are able to use on parents tablets and computers.

Because of this trend we set out to make a game that young children can use to learn while having fun. The specifications are a re-implementation of other word guessing games available for younger children while giving more incentive for competition and freedom in the game. Many educational games available to children are heavily guided and come across more often as a lesson than a game. Our goal is to make a game that children want to play and learn while doing so without pushing a lesson.

2.2 Product Functions

These are the main functional areas of the software.

Gameplay: Each game is 4 rounds where the player can select a category at the beginning of every round. The game will then randomly select a word from that category and display images of all the possible words the player can guess from. Along with the images the game will give a short hint of the word it randomly selected. The player is then tasked with either guessing or asking for another hint. The player is only able to ask for 2 additional hints per round and for every hint that is asked for points are deducted from the potential score for the round. If the player guesses wrong 0 points will be awarded for the round.

Category Selection Management: The game uses a graphical user interface where the user can input different types of information. When the game is started the user is prompted for a username which thereafter the game will begin. The rest of the player interaction through player inputs that the game reacts to. Category selection being the first input of every round, where the player selects which category their word pools will take on.

Score Management: The score for each round will be added to a stored score for the game, building up the games score round by round and then reporting it to the player and comparing it to the highest score for the game. If a new high score is achieved the username and the score is displayed on the main menu.

Feedback Management: This will depend on the input. When a username is inputted the game gives feedback to begin the game. After selecting a category the player is prompted with making a guess or asking for another hint. These forms of feedback help the player through the game while giving updates on wrong/right answers, and on points scored per round/game.

The game stores the following information:

Player Description: Player username, and highscore.

Current Game Stats: Score and username of the current game to compare to highscore.

2.3 User Characteristics

Any user should be capable of using the app, as its graphical user interface is simple to use. No special skill should be assumed on part of the users. The difficulty of the game is not variable and so some people will find the game to be difficult and others will say it's easy. It depends on the person's knowledge of the words/things in the categories.

2.4 Constraints

No special constraints have been identified

2.5 Assumptions and Dependencies

No special assumptions or dependencies have been identified.

3. Overview

3.1 External Interface Requirements

3.1.1 User Interfaces

All interaction with the user will be via the graphical user interface. Once the game has started the user is prompted for a username and then told to pick a category. If the input for the category or for a guess is invalid the system will report that the input was not understood.

3.1.2 Hardware Interfaces

None

3.1.3 Software Interfaces

The game is capable of running on any version of the UNIX system, including Linux.

3.1.4 Communication Interfaces

None

3.2 Performance Requirements

The game should respond to user input within a couple of seconds. No other performance requirements are present.

3.3 Design Constraints

3.3.1 Standards Compliance

All language used in the game should comply with Boston University guidelines In the Academic Conduct Code.

3.4 Software System Attributes

3.4.1 Reliability

The game should not crash or hang, unless it is a result of an operating system error.

3.4.2 Security and Privacy

No specific security or privacy requirements are specified nor needed.

3.4.3 Maintainability

All code will be fully documented. The functions will have comments with pre-conditions. All program files will include comments with authorship and date of last change. The code will be flexible to allow for further improvements.

3.4.4 Further Improvements