
Software Requirements Specification

for

Sir Stan's Well Rounded Adventure

Version 1.0 approved

Prepared by Global Frog Games Team Members

Global Frog Games

2-24-2020

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	2
1.1 Purpose	2
1.2 Document Conventions	2
1.3 Project Scope	2
1.4 References	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 User Classes and Characteristics	2
2.3 Operating Environment	2
2.4 Design and Implementation Constraints	2
2.5 Assumptions and Dependencies	2
3. System Features	2
3.1 System Feature 1	2
3.2 System Feature 2 (and so on)	2
4. Data Requirements	2
4.1 Logical Data Model	2
4.2 Data Dictionary	2
4.3 Reports	2
4.4 Data Acquisition, Integrity, Retention, and Disposal	2
5. External Interface Requirements	2
5.1 User Interfaces	2
5.2 Software Interfaces	2
5.3 Hardware Interfaces	2
5.4 Communications Interfaces	2
6. Quality Attributes	2
6.1 Usability	2
6.2 Performance	2
6.3 Security	2
6.4 Safety	2
6.5 [Others as relevant]	2
7. Internationalization and Localization Requirements	2
8. Other Requirements	2
Appendix A: Glossary	2
Appendix B: Analysis Models	2

Revision History

Name	Date	Reason For Changes	Version
Brooke Smith	10/2/20	Initial writeup	1.0
Team	24/2/20	Group Writeup	1.1

Brooke Smith	8/4/20	Final Edits	1.2

1. Introduction

1.1 Purpose

This SRS describes the functional and nonfunctional requirements for software release 1.0 of *Sir Stan's Well Rounded Adventure*. This document is intended to be used by the members of the project team who will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are committed for release 1.0.

1.2 Document Conventions

No special typographical conventions are used in this SRS.

1.3 Project Scope

Sir Stan's Well Rounded Adventure will serve as a pilot mobile game for Sir Stanley Matthews Coaching Foundation. The mobile game is targeted for children from the ages of 6-11. The mobile game includes minigames that are meant to help teach children the importance of physical activity and proper nutrition. More information is available in the [Vision Document](#).

1.4 References

Team website:

<http://riogrande.cs.tcu.edu/1920GlobalGameApp/index.html>

Glossary

Vision Document

Developers Guide

Software Development Plan

Installation and User Guide

Software Requirements Specification

Testing Plan

Github Repository:

<https://github.com/tcuseriordesigncourse/globalgameapp>

2. Overall Description

2.1 Product Perspective

Sir Stan's Well Rounded Adventure is a pilot mobile game for iOS and android cell phones that will serve as a learning tool for children aged 6-11. This app is a completely new product. The product's main relationships are with the user/player and with the device being used to run the app.

2.2 User Classes and Characteristics

User class	Description
Player	The player is the main user of <i>Sir Stan's Well Rounded Adventure</i> . They are the users that will be playing the game for entertainment and learning purposes.

2.3 Operating Environment

OE-1: The COS shall operate correctly on iOS and android cell phones.

2.4 Design and Implementation Constraints

CO-1: The app should not take up too much of the device's memory to run.

CO-2: The Godot code and scenes must be converted using X-Code to run on iOS systems

2.5 Assumptions and Dependencies

AS-1: The user's phone is updated to the point where they are able to download our App and run it without any difficulty.

3. System Features

3.1 Title Screen

3.1.1 Description

When the player opens the app, the Title Menu is the first thing to be displayed. The player has three navigation options: "Play", "Tips", and "Achievements".

3.1.2 Stimulus/Response Sequences

1. *The player opens the App and the system displays the Title Screen with three menu buttons.*

3.2 Map

■ 3.2.1 Description

When the player selects "Play" from the title screen, the map screen is displayed. The map screen provides navigation buttons to all of the minigames available to the player. The screen will scroll allowing players to see more navigation buttons. Some buttons can be locked, preventing the player from accessing that minigame right away. Player's can unlock these buttons by playing the games. Because this is how the players navigate between minigames, this is a high-priority feature.

■ 3.2.2 Stimulus/Response Sequences

1. The player taps a map node.
2. If the minigame is unlocked, the system will display a tutorial screen.

■ 3.2.3 Functional Requirements

- The user must be able to tap map nodes on the screen
- Map nodes must have a state: unlocked or locked
- The map must be able to scroll, allowing the player to see more of the map

3.3 System Feature 3 (Tips)

■ 3.3.1 Description

The tips screen is a user interface that allows the user to read over the tutorials for each of the minigames as well as more information about the nutritional facts that the minigame is focussing on. This is a low priority feature.

■ 3.3.2 Stimulus/Response Sequences

The user is prompted with three options at the main menu screen. In order to get to the tips screen the user will select the tips button from the main menu. Upon opening of the tips scene background music will begin playing and the user will see buttons that will take the user to another scene that will play the tutorial scene for the game that was selected.

3.4 System Feature 4 (Collections/Achievement)

■ 3.4.1 Description

The Achievements screen is a user interface that allows the user to see what achievements they have completed.

■ 3.4.2 Stimulus/Response Sequences

The user is prompted with three options at the main menu screen. In order to get to the achievements screen the user will select the "Achievements" button from the main menu. Upon opening of the tips scene background music will begin playing and the user will see buttons that will take the user to another scene that will play the tutorial scene for the game that was selected.

■ 3.4.3 Functional Requirements

<Itemize the specific functional requirements associated with this feature. These are the software capabilities that must be implemented for the user to carry out the feature's services or to perform a use case. Describe how the product should respond to anticipated error conditions. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

3.5 Pause Minigame (Pause Screen)

■ 3.2.1 Description

While playing the minigame the user pauses the game. When the game is paused, the minigame stops processing and the Pause Screen GUI appears. The Pause Screen gives options to resume the minigame or quit and return to the map screen. Based on user input the pause menu will either resume the current minigame or return to the map screen. This is a High priority feature.

■ 3.2.2 Stimulus/Response Sequences

1. User presses pause button
2. Minigame stops processing
3. Pause Menu GUI is transitioned into screen view
4. User selects menu option
5. Pause Menu is transitioned from screen view

6. Option Choice
 - 6.1. Resume - Minigame is set to process and resumes
 - 6.2. Return to Map - Minigame is removed from scene, and Map screen is presented.

■ 3.2.3 Functional Requirements

- Touch screen capabilities
- Must be currently playing a Minigame
- Relies on existence of the Map screen
- Pause Menu is a Singleton in global scope (Reduces number of scenes instantiated when a Minigame is loading)

Falling Food

■ 3.4.1 Description

In this minigame food is falling from the sky and the player is tasked with catching it. There are multiple types of food and when the player catches food the score count for it's food group is incremented by one. The minigame is timed and once the timer runs out the minigame ends. After the timer runs out; the score screen is displayed to the player. This is a high priority feature. Also, there's a level two that gives bonuses for catching balanced meals. A balanced meal consists of one food from each food group without any duplicate food groups. So, to catch a balanced meal the player must catch one carb, one protein, one fruit, and one vegetable in any order. They will receive a bonus multiplier that will apply to their score as long as they don't "break the streak".

■ 3.4.2 Stimulus/Response Sequences

1. Minigame begins and food starts falling from the sky
2. The player taps the location on the screen that he/she would like to move the basket. The basket is used for catching the food.
3. Timer runs out
4. Score screen is displayed
5. Player clicks return to map
6. Player is returned to the Map screen

■ 3.4.3 Functional Requirements

- Touch screen capabilities
- Relies on the Map screen for it to be selected
- Relies on game save system for keeping track of high scores

3.6 Goalie Shootout

3.6.1 Description

Goalie Shootout is a mini-game where the objective is to score as many goals as possible by kicking the soccer ball past the goalie and into the net. This Feature priority is high.

3.6.2 Stimulus/Response Sequences

The user is presented with a scene of half a soccer field with a ball towards the player and a goalie moving back and forth between each goal post. The user taps on a location where they want the soccer ball to be “kicked”. If the ball enters the goal, they get a point. Otherwise if the ball bounces off the goalie or misses the net, no point is added.

3.6.3 Functional Requirements

The hierarchy of the Goalie Shootout Game is divided into two canvas layers, one for UI and the other for game objects. The UI layer contains the timer, score tracker, and the pause button. The game layer contains all of the game objects such as the soccer ball, goalie object, and goal object. Along with these things, there is a background image for the soccer field.

3.7 Goalie Defender

3.7.1 Description

The Goalie Defender game is a mini-game where the objective is to act as a goalie and block as many shots from entering the goal as possible. High priority.

3.7.2 Stimulus/Response Sequences

The user is presented with half a soccer field with a ball on the top and a goal net with a goalie towards the player. The user will touch and drag the goalie along a line across the goal net with the intent to “block” a soccer ball that is kicked in a random direction. If the ball is successfully blocked, the game goes on. Otherwise the player will lose a heart. If three hearts are lost, the game is ended. Steps

3.7.3 Functional Requirements

The hierarchy of the Goalie Defender Game is divided into two canvas layers, one for UI and the other for game objects. The UI layer contains the timer, score tracker, and the pause button. The game layer contains all of the game objects such as the soccer ball, goalie object, and goal net object. Along with these things, there is a background image for the soccer field.

3.8 Color Game

3.8.1 Description

Color game is a mini-game where the player is given a grayed-out image of a food item and is shown four different colored buttons. The user must then select the button matching the color of the food.

3.8.2 Stimulus/Response Sequences

The user is presented with four buttons of different colors then is shown a greyed-out food image in the middle of the buttons. The user then selects the color they believe corresponds with the food, is told if they are correct or incorrect, then is shown the next food image until there are no more images left.

3.8.3 Functional Requirements

The hierarchy of Color Game is divided into two canvas layers, one for UI and the other for game objects. The UI layer contains the timer, score tracker, and the pause button. The game layer contains two box containers that hold the four color buttons, as well as a texture rect in the center to hold the food images. Along with these things, there is a child scene on the grayFoods node that holds each question. Each question consists of an image and the corresponding color choice.

3.9 Tutorial

3.9.1 Description

When the user selects a minigame to play, the scene changes to the minigame and the tutorial screen appears. The minigame is paused and all buttons are disabled while the tutorial is present. The user clicks next to iterate through the dialog then, once all dialog has been displayed, the tutorial screen goes away and the minigame starts.

3.9.2 Stimulus/Response Sequences

The tutorial appears when a minigame scene is loaded and immediately displays the first dialog line. There is one button that can be interacted with and it says 'Next' on it. When clicked, it will display the next line of dialog until there is no more dialog left. Upon reaching the final line of dialog, the 'next' button will change to 'finish'. When 'finish' is clicked, the tutorial box will disappear and the minigame will begin.

3.10 Stan's Snacks

3.10.1 Description

The user is prompted to select one of two buttons labeled "Fruit" and "Vegetable" before a countdown timer at the top of the scene reaches zero. The correct selection is based on the food image and associated food name displayed in the center of the scene. If the correct answer is selected, the user will see a solid green background with the text "Correct" displayed across the screen. Then the system will transition to the next question if one is available, and the score text will be incremented by 1. If the incorrect answer is selected, the user will see a solid red background with the text "Incorrect" displayed across the screen, and the system will transition to the next question if one is available.

3.10.2 Stimulus/Response Sequences

1. Upon successful loading of the Stan's Snacks minigame, the system presents the user with a countdown timer that starts at three seconds and counts down to zero, an image of a specific food, a string of text indicating the name associated with that particular image, a score located in the upper left corner of the scene, and two buttons located at the bottom of the scene with the labels "Fruit" and "Vegetable".
2. At the beginning of each question, the system restarts the countdown timer, the user then views the image and selects whether they believe the image is categorized as a fruit or a vegetable by pressing the corresponding button.
3. If the user presses the button attached to the correct answer, then a green transition screen will appear before loading the next question. The system will also increment the score text by one.
4. Contrarily, if the user presses the incorrect button, a red transition screen will appear, indicating the wrong answer choice was selected.
5. If the user presses the pause button, the game should freeze, the timer should pause and the pause menu should appear to the user.
6. After completion of all twelve questions, the system will process the users score and display the score screen to the user.
7. If the user correctly answers between one to four questions, they will receive one star; a score of between five to eleven will result in two stars, and a score of twelve will give the user three stars (the max).

3.11 Score Screen

3.11.1 Description

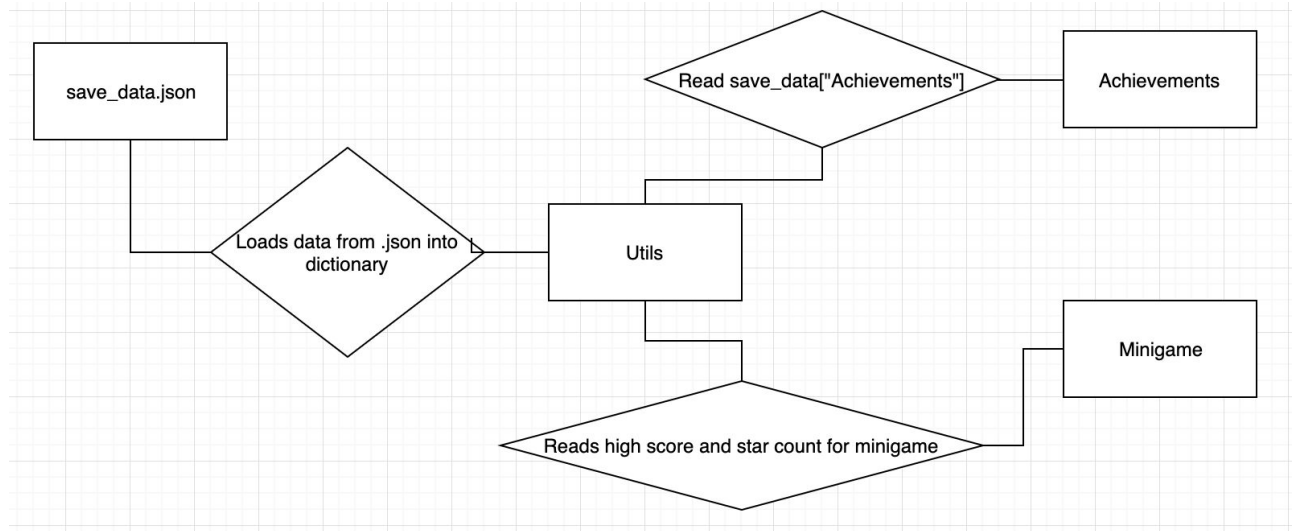
The score scene is displayed upon completion of each mini game. The scene will display the quantity of stars the user achieved for the mini game, the overall score, and any achievements unlocked. In a special instance for the "Falling Food" minigame, the score screen will also display the amount of each type of food collected throughout the game. There is a button displayed at the bottom of the scene with the text "Return to Map"; pressing this button will transition to the map scene.

3.11.2 Stimulus/Response Sequences

1. The player finishes a mini-game.
2. The system displays the player's score.
3. The player can tap the "Back to Map" button when they are ready.

4. Data Requirements

4.1 Logical Data Model



4.2 Data Dictionary

All persistent data was stored in JSON files.

4.3 Data Acquisition, Integrity, Retention, and Disposal

The system shall make a local copy of all saved data at startup, store it in a dictionary, and use it for all data fetches.

5. External Interface Requirements

5.1 User Interfaces

UI-1: *Sir Stan's Well Rounded Adventure* has a common “theme” applied to most of the screens. This allows the buttons to look similar in style across the app.

UI-2: Each minigame includes a tutorial that will explain how the player should interact with their mobile device to play the minigame.

UI-3: Each minigame also includes a score screen that will be displayed at the end of the minigame

5.2 Hardware Interfaces

No hardware interfaces have been identified.

5.3 Communications Interfaces

No communication interfaces have been identified.

6. Quality Attributes

6.1 Usability

USE-1: Well Rounded has a general “theme” applied to the user-interface of most scenes. This allows the user to become comfortable with the “look” of the menus, making navigation easier to learn.

USE-2: Each minigame includes a tutorial that will help the player learn how to play each minigame.

6.2 Performance Needs Completion/Delete

PER-1: Performance is something we definitely wanted to pay attention to during development. We made sure our game would be simple enough that there were no issues with performance.

6.3 Security

No security or privacy issues to note.

6.4 Safety

No safety issues to note.

6.5 Availability

AVL-1: Well Rounded is available to anyone with an android or iOS device.

6.6 Robustness

No issues to note.

7. Internationalization and Localization Requirements

We are using British English formatting and spelling.

● 8. Legal Requirements

LEG-1: Art in Sir Stan's Well Rounded Adventure is created by our team.

LEG-2: Sound effects in Sir Stan's Well Rounded Adventure are under the Creative Commons 0 Licence. This means the creator has dedicated the work to public domain, allowing our team to use the sound effects for commercial purposes without any terms.

LEG-3: Music in Sir Stan's Well Rounded Adventure is under a royalty free licence. This allowed our team to pay a one time fee in order to use the music in our game.

Appendix A: Glossary

Please check our glossary which can be found under "References".