Software Requirements Specification

for

Moon Moose

Version 1.0 approved

Prepared by Max O, Sean G, and Billy B

The Moose Fighters

November 2, 2016

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 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 3

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 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to present a detailed description of the moon moose iOS game. It will explain the purpose and features of the game, the interface under which the game will run, and how the game will run by itself. This document is intended for any person including the developers who wish to use the game.

## Document Conventions

Every requirement statement is to have its own priority. There is no requirement statement that holds a higher status over any other.

## Intended Audience and Reading Suggestions

One of the intended audiences for this game is the users. The users of this game can be person looking to pass time who owns an iOS device and has downloaded iOS 8 or later. The users do not specifically need to know how the game run, so the main reading for the users are the introduction section, overall description section, and the section on safety requirements. The second audience this document is intended for are the developers of this game. The suggested reading for the developers is the whole document. The developers must know the intended requirements and overall description of the moon moose game.

## Product Scope

This product will be written using a sprite kit which is the api for which iOS apps are written. This is the code for which apple uses for the devices. After the code is written the moon moose game will be uploaded to the app store and game center so that any person is able to download this game.

## References

Refer to the use case model and use case document.

# Overall Description

## Product Perspective

This product is a new self-contained iOS game, however, it will be uploaded to the apple app store and game center. Moon moose is not part of any other larger system. The use case model will be attached with this document.

## Product Functions

The primary moon moose functions are:

* The system shall allow to user input to be a lean on the screen. Either left, right, up or down.
* The system shall have multiple difficulties.
* The system shall save the highest score reached in the game.
* The system shall display the current score of the player on the game screen.
* The system shall display a moose as the player icon.
* The system will have a start menu which includes a new game option, continue game option, and options button.
* The system shall have an options button which allows the user to change the game settings.
* The system shall have a pause game option to play and pause whenever the user deems necessary.

## User Classes and Characteristics

The user classes include any person who wishes to play this game, that person could be of any age. As long as that person has an iOS 8 or newer capable device (iPhone, iPad, or iPod Touch). All of these devices will have the same user requirements no matter which device the user is playing the game on.

## Operating Environment

The moon moose game will operate on a sprite kit which is used by apple to created apps for the iOS app store. The game will also be uploaded to the iOS game center so that different users can compare his or her own score to other users. The hardware used will primarily be on a mac laptop, but a dell laptop will be used for sprite using an emulator. This game will run on an apple iOS device as long as iOS 8 has been downloaded onto said device. The different devices include and I-Phone, iPad, or iPad Touch.

## Design and Implementation Constraints (Optional)

Along with this requirements document the use case model and use case document will also be turned in.

## User Documentation

This game does not depend on any other project or previous iOS app, therefore there are no dependencies. As of right now most the coding will be done using a sprite kit and other parts will be done on a dell laptop using an emulator for the sprite kit. If this emulator does not work, then the code written on the dell laptops will not work correctly. If this is the case, then all of the code will be done on a mac laptop.

# External Interface Requirements

## User Interfaces

The user interface will begin at a menu screen that will have three buttons for the user to choose from. They can continue a previous game, start a new game, or go to the options menu. The options menu will have settings to change their saved data settings. If they choose to start a new game the system will erase the data of the current game run and begin a new game. If they continue they can simply pick up where they left off. The screen will show at all times what the character model is doing based on the user’s input. While the game is running the user my pause the game by tapping a pause button that will be on screen at all times. From the pause menu the user my exit the game or continue playing.

## Hardware Interfaces (Optional)

The user input will be tracked by tapping and tilting the device. Tapping will be used to select different options on the home screen and when the game is paused. Tilting will be the main input of the game itself. The user will tilt the phone and the phone’s gyroscopic hardware and software will track those inputs and relay the information to the system.

## Software Interfaces

This product will interact with the iPhone App Store on all iOS 8 enabled Apple devices. The system will communicate with the app store to download the game data to the user’s device. After installation the user’s game center information will be saved and the game will be able to track the user’s progression and milestones throughout the game. The system will continuously update the game center’s information. This will enable to user to compare their progress to that of other users playing the game. All coding will be done in swift code and made available for free download. However, any device that is not using iOS 8 or newer will either need to update their device or will not be able to run the game.

## Communications Interfaces

The only communications interface our product will use is to interact with the game center on iOS devices. Each device will require some form of internet connection for the system to be able to update and communicate to the server what changes to the data are needed.

# System Features

## Start Menu

4.1.1 Description and Priority

This is a high priority feature. It is the first thing the user sees and will allow them to select any and all options they wish. Everything from starting a new game to changing game settings will be available here.

4.1.2 Stimulus/Response Sequences

Once the user starts the game the menu will appear. They can then tap the screen to navigate the different options available to them. There will only be three buttons, two of which will take them into a game. The options button will take them to another menu screen where they can change their personal game settings.

4.1.3 Functional Requirement

REQ-1: The menu shall present all available options as buttons

REQ-2: The menu shall only react to the buttons being pressed

REQ-3: The menu shall take you to the proper screen when asked

## Options

4.2.1 Description and Priority

This is a medium priority feature. This feature will allow the user to alter their personal settings and certain game features.

4.2.2 Stimulus/Response Sequences

Once the user starts the game the menu will appear. They will then select the options menu by tapping on the options button. Once they select the options menu the menu itself will appear and present them will all available options. They can then change these options and exit the menu. Upon exit the options will be automatically saved.

4.2.3 Functional Requirement

REQ-1: All options shall be presented on the screen

REQ-2: The options shall be saved upon exit

REQ-3: The user will be able to reset all settings to default

## Game

4.3.1 Description and Priority

This is a high priority feature. This is the core of the application. The game will allow players to play as a moose that is continuously climbing up the screen. To do so the player must avoid obstacles and not fall off the bottom of the screen. They will be able to pause the game and start a new game or continue an old game at any time.

4.3.2 Stimulus/Response Sequences

Once the user starts the game the menu will appear and they will select either new game or continue game. If they select continue game they will be brought to the game pause menu and can unpause and play. If they select new game they will begin a new game which will start after a timer. They can then interact with the game by tilting the phone. If they tilt it right, the character will drift right, if they tilt it left, the character will tilt left. At any time, they may pause the game and exit, this will save their game data.

4.3.3 Functional Requirement

REQ-1: The game may be paused at any time

REQ-2: A paused game will start from the time it was paused

REQ-3: The game will be saved upon exit

REQ-4: If a new game is started, all previous game data will be overwritten

REQ-5: The game character will react to only left and right tilt input

REQ-6: Only the buttons on the game screen will react to tapping

# Other Nonfunctional Requirements

## Performance Requirements

Performance of this game will depend on the version of iOS and the specific iPhone model the user has. The latest iPhones such as the iPhone 7 that run iOS 10 will see the best overall performance when it comes to the game. This game will be available to iPhones that are running iOS later. This includes the iPhone 5 and later. Earlier iPhones than the iPhone 5 will not be able to run this game.

## Safety Requirements

Requirements such as looking and interacting with the iPhone touch screen may cause eye soreness or other minor issues during extended use. Taking breaks every few minutes from the game may resolve these issues.

## Security Requirements

Signing into Apple’s game center may allow other users to view your game center profile and statistics.

## Software Quality Attributes

It is important that the game is coded well so that it could be maintained easily for any future updates and it can be easily testable. It is also necessary to check for correctness before release so that there are not any bugs that will make the game unplayable.

## Business Rules

No business rules apply with this product.

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

If the user has access to the internet and is logged into game center, high scores and leaderboards will be synced to the game center database where the info will be stored.

Appendix A: Glossary

No external terms required to fully understand the SRS.

Appendix B: Analysis Models

Use Case Diagram Model attached with this document.

Appendix C: To Be Determined List

Currently, there is additional information to be determined.