Software Requirements Specification

for

<Snake Game>

Version 1.1 approved

Prepared by <Zachary Lloyd>

<CSC318 Group A>

<10/13/2019>

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

2. Overall Description 1

2.1 Product Perspective 1

2.2 Product Functions 1

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 2

3.1 User Interfaces 2

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

4. System Features 3

4.1 Feature 1: Menu 3

4.2 Feature 2: Snake 3

4.3 Feature 3: Score 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Software Quality Attributes 4

Diagrams: 5

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Zachary Lloyd | 10/27/19 | Making the document related to the team’s project | 1.0 |
| Zachary Lloyd | 11/6/19 | Making changes to the document given to us from feedback | 1.1 |

# Introduction

## Purpose

The purpose of this project is to recreate the popular Snake Game. The scope of the project is simple. It will contain an interface through the console. It will have a menu, snake, board, food, win/loss criteria, and score.

## Document Conventions

Everything in the project is just about the same priority level. The user needs the menu to get into the game. The user needs the snake to control something in the game. The board is needed to spawn in food to collect score and give the user an area to play on which will give the user win/loss criteria.

## Intended Audience and Reading Suggestions

The intended audience for this document is the team and the stakeholder. That way the stakeholder understands what will be completed along with the team understanding what all needs to be completed for the project to be viable.

## Product Scope

The scope of the project is simple. The team is not doing anything that does not need to be included in the project to add anything extra. The team is on a straight and narrow path to complete the required requirements for the project. The requirements for the project consist of having a menu, snake, board, food, win/loss criteria and score.

# Overall Description

## Product Perspective

The product will be produced for any computer using windows 10 that has access to the console. It’s going to be a very simple product.

## Product Functions

The user will be able to play the product where they will be greeted at a menu and given the option to play the game, look at top scores, or quit the application.

## User Classes and Characteristics

The user will have access to the menu. They will be moving the snake around the game to collect food that is spawned into the game. The food will affect the players score.

## Operating Environment

The product will be played on windows 10 on a computer that has the availability to pull up a console window.

## Design and Implementation Constraints

## User Documentation

The menu will give an overview of how the product will function and how to play. From there when

they go into the game they will be able to press a pause button which will also bring up controls.

# External Interface Requirements

## User Interfaces

## Hardware Interfaces

The intended hardware audience for this product is a basic computer it does not require high level specs considering it will be played on the console.

## Software Interfaces

The required software for the product is the console and the game will have a menu that puts the user into the game. From the menu you can go into the game or you may quit. The user will also be able to see scores if time permits.

# System Features

## Feature 1: Menu

4.1.1 Description and Priority

This feature will be what allows the player to go into the game. It will allow them the choice to see scores, quit, and play. This is a medium level priority.

4.1.2 Stimulus/Response Sequences

There will be numbers listed out (if possible, buttons) that allow the user to select different situations: playing, see the scores, quitting.

4.1.3 Functional Requirements

The menu must have functionality such as quitting for the game, playing the game, and displaying the scores that have been saved.

## Feature 2: Snake

4.2.1 Description and Priority

This feature will be what allows the user to move around the board, eat the food to increase difficulty and create win/loss criteria.

4.2.2 Stimulus/Response Sequences

This feature will be used to allow the user to see the snake move around the board, upon eating the food the player will see the snake grow in length and score rise, and lastly if the player runs into itself or a wall they will lose.

4.2.3 Functional Requirements

The snake needs to be able to move throughout the board, gain length upon eating while also raising the score, and the snake also needs to detect if it runs into itself or a wall causing the user to lose.

## Feature 3: Score

4.3.1 Description and Priority

This feature will be what allows the user to see the amount of food that has been eaten with also creating top scores that can be saved and displayed.

4.3.2 Stimulus/Response Sequences

This feature will be used to allow the user to see the score giving them incentive to do better and get a high score that will be saved and able to be viewed

4.3.3 Functional Requirements

The score needs to be able to detect when food has been eaten to add to the current score. The score also needs to be able to determine whether it is a new top score or that the score can just be deleted.

# Other Nonfunctional Requirements

## Performance Requirements

The product needs to be able to perform with out harming performance and since the project is small it should not harm performance whatsoever.

## Software Quality Attributes

The product needs to be able to be played and fun give the user a quality experience. The whole experience relies on how the users feels about the game while it is being played.