Software Requirements Specification

for

Centipede Army Checkers

**Version 1.0 approved**

**Prepared by Adrienne Bowman,   
Cynthia Cordova, Rogelio Garza,   
Jacob Gibson, Brent Redmon**

**Centipede Army**

**October 29th 2018**

**Table of Contents**

**Table of Contents 1**

**Revision History 2**

**1 Introduction 3**

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Product Scope 3

1.5 References 3

**2 Overall Description 4**

2.1 Product Perspective 4

2.2 Product Functions 5

2.3 User Classes and Characteristics 6

2.4 Operating Environment 7

2.5 Design and Implementation Constraints 7

2.6 User Documentation 7

2.7 Assumptions and Dependencies 7

**3 External Interface Requirements 8**

3.1 User Interfaces 8

3.2 Hardware Interfaces 9

3.3 Software Interfaces 9

3.4 Communications Interfaces 9

**4 System Features 9**

4.1 Single Player Mode 9

4.2 Multiplayer Mode 11

4.3 Spectator Mode 13

4.4 Subscription Service 14

**5 Other Nonfunctional Requirements 15**

5.1 Performance Requirements 15

5.2 Safety Requirements 15

5.3 Security Requirements 15

5.4 Software Quality Attributes 15

5.5 Business Rules 15

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Rogelio G. | 10/24 | Indexed Requirements for Reference in Test Plan | 1.0 |
| Jacob G. | 10/23 | Worked through document to enhance cohesion and standardize format | 0.75 |
| Jacob G. | 09/13 | Wrote Nonfunctional Requirements | 0.5 |
| Rogelio G. | 09/13 | Wrote System Features | 0.4 |
| Brent R. | 09/13 | Wrote External Requirements | 0.3 |
| Cynthia C. | 09/13 | Wrote Overall Description | 0.2 |
| Adrienne B. | 09/13 | Wrote Introduction to Document | 0.1 |

# Introduction

The following document serves as a compilation of requirement specifications for the Centipede Army’s creation of a game of Checkers.

## Purpose

The purpose of this document is to explicitly define the user requirements associated with the Centipede Checkers project, and all the tasks that the project aims to achieve. We will also discuss system requirements and constraints, project scope, data flow control, and the interactions between different aspects of the system and the user. This document will be revised as needed, and will serve as a contract between the client and ourselves.

## Document Conventions

This document has been formatted using Arial style font. The section headers are sized to 18 point font, the subsections to 14 point font, and the content has been sized to 11pt font. The use of bolded text is to aid in distinguishing the titles of sections and subsections from the rest of the content.

## Intended Audience and Reading Suggestions

This document is intended for developers, project managers, and the client to know the scope of this project, establish a common language, and style conventions. This document also serves as a validation tool between each group such that a baseline is established, and changes will be agreed upon as the scope of this project changes over time.

## Product Scope

This Centipede Checkers game has the ability to be played as a single human player with a computer opponent, as a player versus player game on local machines or through on online connection, and to watch a live multiplayer game. This product will be used by and available to end-users, clients, and developers.

## References

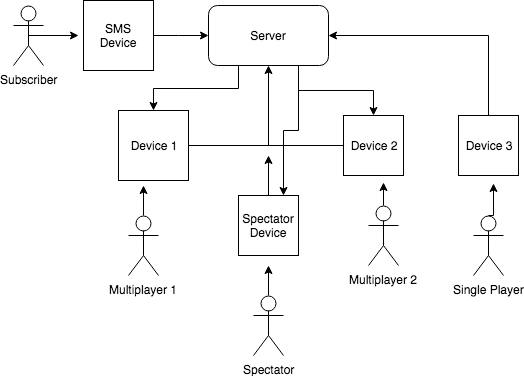
The basic rules for our creation of Checkers which are to be implemented in our project can be found here: *http://centipedecheckers.ngrok.io/rules*

The following standard JavaScript Style Guide will serve as our Style Guide for the code portion of this project: *https://google.github.io/styleguide/jsguide.html*

# Overall Description

## Product Perspective

The game shall be a stand-alone application. This game shall consist of a graphical user interface (GUI) which shall allow the user to maneuver through the web application at startup. The game shall be developed to be simple and user friendly, thereby expanding our user base. The product shall provide a software that allows players entertainment amongst friends over a network or competition against a virtual opponent.

[**](https://www.draw.io/#G1AWc98U7Id5guruQk3Pp8BLIY-_2HRskl)

*Figure 1*

## Product Functions

### User Interface

### The user interface, which shall function on personal computers and mobile handheld devices, shall show the user a board that consists of two sets of checker pieces. Player one shall receive a set of blue checkers and the other player shall receive a set of green checkers.

### AI Implementation

### At the start of the application, the user may choose to play against another player via the network connection or against an AI controlling the other set of pieces. If the user has chosen to play against a computer controlled opponent, the user shall take the first initial move from the normal starting position. If the user has chosen to play against another player, then the first player that connects to the game session shall take the first initial move from the normal starting position.

### Piece Logic

### After the initial move, play shall proceed with each side taking turns moving their pieces according to the logic and rules described in the standard rules listed above in Section 1.5.

### Kings

### Should either player move a piece to the “king position” as described in the standard rules, then, their piece shall be upgraded to a king and shall behave accordingly.

### Normal Starting Position and Win/Loss

### The game shall begin from the normal starting position, as shown in figure 2. The game shall proceed until one side wins, and upon this state the game shall declare the winner accordingly. For multiplayer games, additional options shall exist to allow for the possibility of spectators and subscribers that shall also be informed on the matches progress.

*Figure 2*

## User Classes and Characteristics

This game is designed to be user friendly and simple to play, therefore experience with the product will not be a major factor in determining who is able to use it. Although it is easy to use, players with past game experience may have a strategic advantage due to improved pattern recognition. The classes determining these types of players include:

* **Foresight**: The ability to plan out paths based on the other players moves, or patterns to get to his/her desired location.
* **Intuition**: The ability to make successful guesses about the opponent's next move based on previous moves.

## Operating Environment

Centipede Checkers shall be created to run on the following operating systems:

* Windows (7, 8, 8.1 10)
* MacOS (10.11, 10.12, 10.13, 10.14)
* Android (7, 8, 9)
* IOS (10, 11, 12)

Centipede Checkers shall be created to run on the following web browsers

* Google Chrome version 69.0.4397 or later
* Safari version 11 or later
* Firefox 63 or later
* Microsoft Edge 42.17314 or later

Additional environments may be compatible, but not directly supported or considered.

## Design and Implementation Constraints

* The interstitial viewer only supports a single user at a time.
* A local network or internet connection are required to interface between multiple users
* Difference in network throughput based upon location and speed could affect the overall latency of the application.
* The software will not be maintained by the developer following release, which may result in compatibility issues in the future.

## User Documentation

Intuitive menus and an informative “About” page shall be included inside of our software to help instruct and guide the user in achieving full and effective use of our product.

## Assumptions and Dependencies

In order to effectively use this product, the user must possess a device that is capable of connecting to the internet and interacting with the software through button presses or mouse clicks. Should the user wish to participate in the subscription service, they must possess a mobile device capable of receiving SMS.

# External Interface Requirements

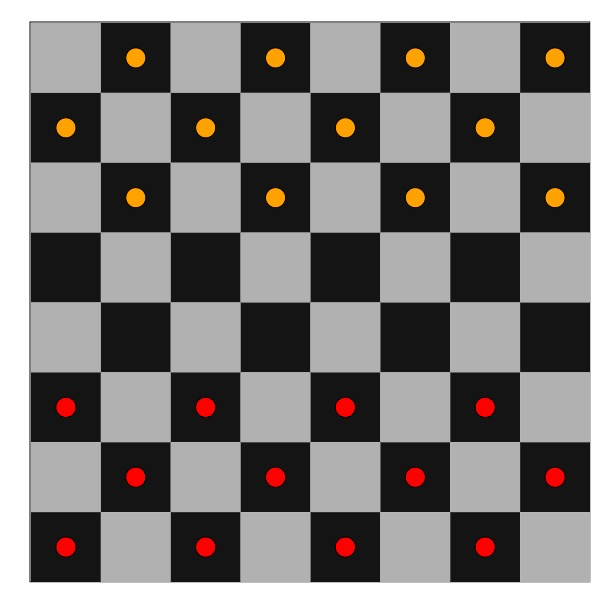
## User Interfaces

This is a web based version of checkers that is fed to the user via a backend server. Therefore, the interface between the software and the user is through a web browser. We want to alleviate architectural and software differences among all our users by delivering this product via a platform that is common amongst all users, a web browser. Furthermore, since this will be web based, even mobile devices can access and play the game.

The opening menu shall contain the following buttons:

* Home - Takes the user to the main menu
* About - Displays the “About” page
* Singleplayer - Begins a single player game against the AI
* Multiplayer - Begins a multiplayer game against a human opponent
* Spectator - Opens a spectating session on an existing match
* Subscribe - Subscribes to an existing match

Figure 3 shows an example of what the product’s game interface looks like once a match has begun. Play will continue by the player selecting their piece and then selecting a position to move it to.

**

*Figure 3*

## Hardware Interfaces

As long as your device can connect to a local area network, deliver input via a mouse or touch screen, and display a web browser through a screen, this application is universally accessible across multiple platforms. To run the server that feeds the game to the player, no constraints are required of the server. As long as the server can execute a binary file, it can be run regardless of what operating system the server has.

## Software Interfaces

The game will be fed through a NodeJS backend server. The computer running the server can also play the game by connecting to port 5000 on its localhost, and other devices can connect to it by connecting to port 5000 on the servers static IP address. Each user that connects to the server will be given a unique ID such that the system can contextually understand how much traffic is being demanded of it, and where each player is during their connection time.

## Communications Interfaces

The Centipede Checkers server will take advantage of the Web Socket communication protocol, which will enable us to host multiplayer checkers matches since sockets establishes bidirectional communication between clients that connect to the server. We will also write API’s that will allow game spectators to receive information about who is currently winning in a multiplayer match. But most importantly, web sockets will used to update the screen of the match so that each player (who will be playing on separate devices) will see the same thing after each turn. There are no security or encryption issues. Regarding synchronicity, we will be taking advantage of Javascript’s asynchronous nature such that we can model our game off a circular Observer/Observable pattern, with each player subscribing to each other “posts” (updating the screen after each move).

# System Features

## Single player mode

**4.1.1 Description and Priority**

The user is able to play against an AI (High priority)

**4.1.2 Stimulus/Response Sequences**

Stimulus: The user selects single player mode

Response: The single player game loads

Stimulus: The user selects their man

Response: The man is selected

Stimulus: The user selects the opponent's man

Response: The opponent’s man is not selected

Stimulus: The user moves a man forward to an unoccupied space

Response: The man moves

Stimulus: The user moves a man backward

Response: The man does not move

Stimulus: The user moves a man to an occupied space

Response: The man does not move

Stimulus: The user jumps over an opponent's man with a man to an unoccupied space

Response: The man jumps over the opponent’s man and takes the opponent's man

Stimulus: The user jumps over an opponent's man with a man to an occupied space

Response: The man does not move

Stimulus: The user moves a man to the other side of the board

Response: The man is turned into a king

Stimulus: The user selects their king

Response: The king is selected

Stimulus: The user selects the opponent’s king

Response: The opponent's king is not selected

Stimulus: The user moves a king forward to an unoccupied space

Response: The king moves forward

Stimulus: The user moves a king backward to an unoccupied space

Response: The king moves backward

Stimulus: The user moves a king forward to an occupied space

Response: The king does not move

Stimulus: The user moves a king backward to an occupied space

Response: The king does not move

Stimulus: The user jumps over an opponent's man with a king to an unoccupied space

Response: The king jumps over the opponent’s man and takes the opponent’s man

Stimulus: The user jumps over an opponent's man with a king to an occupied space

Response: The king does not move

Stimulus: The user jumps over an opponent's king with a king to an unoccupied space

Response: The king jumps over the opponent’s king and takes the opponent’s king

Stimulus: The user jumps over an opponent's king with a king to an occupied space

Response: The king does not move

Stimulus: The user jumps over an opponent's king with a man to an unoccupied space

Response: The king jumps over the opponent’s king and takes the opponent’s king

Stimulus: The user jumps over an opponent's king with a king to an occupied space

Response: The man does not move

Stimulus: The last man is captured

Response: The winner is displayed

Stimulus: The user selects the about button

Response: The about page loads

Stimulus: The user selects the home button

Response: The home page loads

**4.1.3 Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Function** | **Description** |
| FR-SING1 | Select Man | The application shall allow the user to select their man, but not an opponent's man |
| FR-SING2 | Move Man | The application shall allow the user to move their man forward, not backwards, to an adjacent unblocked space |
| FR-SING3 | Jump Man | The application shall allow the user to jump over an opponent's man if the next space is unblocked |
| FR-SING4 | King Me | The application shall turn a man into a king once it reaches the other side of the game board |
| FR-SING5 | A.I. | The system shall control the opposing teams men and play against the user with the same functionality as the user |
| FR-SING6 | Display Winner | The system shall keep track of the men on the board and notify the user who the winner is when the game is over |

## Multiplayer mode

## 4.2.1 Description and Priority

Two users are able to play with each other (High priority)

## 4.2.2 Stimulus/Response Sequences

Stimulus: The user selects multiplayer mode

Response: The please enter a user name pops up

Stimulus: The user selects enter lobby without entering a username

Response: A message stating a username is required pops up

Stimulus: The user selects enter lobby with entering a username

Response: The multiplayer game loads

Stimulus: The user selects their man

Response: The man is selected

Stimulus: The user selects the opponent's man

Response: The opponent’s man is not selected

Stimulus: The user moves a man forward to an unoccupied space

Response: The man moves

Stimulus: The user moves a man backward

Response: The man does not move

Stimulus: The user moves a man to an occupied space

Response: The man does not move

Stimulus: The user jumps over an opponent's man with a man to an unoccupied space

Response: The man jumps over the opponent’s man and takes the opponent's man

Stimulus: The user jumps over an opponent's man with a man to an occupied space

Response: The man does not move

Stimulus: The user moves a man to the other side of the board

Response: The man is turned into a king

Stimulus: The user selects their king

Response: The king is selected

Stimulus: The user selects the opponent’s king

Response: The opponent's king is not selected

Stimulus: The user moves a king forward to an unoccupied space

Response: The king moves forward

Stimulus: The user moves a king backward to an unoccupied space

Response: The king moves backward

Stimulus: The user moves a king forward to an occupied space

Response: The king does not move

Stimulus: The user moves a king backward to an occupied space

Response: The king does not move

Stimulus: The user jumps over an opponent's man with a king to an unoccupied space

Response: The king jumps over the opponent’s man and takes the opponent’s man

Stimulus: The user jumps over an opponent's man with a king to an occupied space

Response: The king does not move

Stimulus: The user jumps over an opponent's king with a king to an unoccupied space

Response: The king jumps over the opponent’s king and takes the opponent’s king

Stimulus: The user jumps over an opponent's king with a king to an occupied space

Response: The king does not move

Stimulus: The user jumps over an opponent's king with a man to an unoccupied space

Response: The king jumps over the opponent’s king and takes the opponent’s king

Stimulus: The user jumps over an opponent's king with a king to an occupied space

Response: The man does not move

Stimulus: The last man is captured

Response: The winner is displayed

Stimulus: The user selects the about button

Response: The about page loads

Stimulus: The user selects the home button

Response: The home page loads

**4.2.3 Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Function** | **Description** |
| FR-MULT1 | User Name | The application shall allow a user to enter a user name to play as |
| FR-MULT2 | Select Man | The application shall allow a user to select their man, but not an opponent's man |
| FR-MULT3 | Move Man | The application shall allow a user to move their man forward, not backwards, to an adjacent unblocked space |
| FR-MULT4 | Jump Man | The application shall allow a user to jump over an opponent's man if the next space is unblocked |
| FR-MULT5 | King Me | The application shall turn a man into a king once it reaches the other side of the game board |
| FR-MULT6 | Display Winner | The system shall keep track of the men on the board and notify the user who the winner is when the game is over |

## Spectator mode

**4.3.1 Description and Priority**

The user is able to watch a live multiplayer game (medium priority)

**4.3.2 Stimulus/Response Sequences**

Stimulus: The user selects spectator mode from the home page

Response: The spectator page loads

Stimulus: The user selects a man on the board

Response: No response

Stimulus: The user selects the about button

Response: The about page loads

Stimulus: The user selects the home button

Response: The home page loads

**4.3.3 Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Function** | **Description** |
| FR-SPEC1 | Restrict Control | The application shall not allow the user to intervene in the game |

## Subscription service

**4.4.1 Description and Priority**

Be able to subscribe and unsubscribe for live multiplayer game alters that are delivered by text message to a phone number provided by the user (Low priority)

**4.4.2 Stimulus/Response Sequences**

Stimulus: The user selects subscribe from the home page

Response: The subscription page loads

Stimulus: The user inputs their phone number and selects subscribe

Response: The users phone number is added to the subscription service and the confirmation page loads

Stimulus: The user inputs their phone number and selects unsubscribe

Response: The users phone number is removed from the subscription server and the confirmation page loads

Stimulus: The user does not input a phone number and selects subscribe

Response: A message stating a phone number is required will pop up

Stimulus: The user does not input a phone number and selects unsubscribe

Response: A message stating a phone number is required will pop up

Stimulus: The user does not input a valid phone number and selects subscribe

Response: A message stating the user has entered an invalid phone number will pop up

Stimulus: The user does not input a valid phone number and selects unsubscribe

Response: A message stating the user has entered an invalid phone number will pop up

Stimulus: The user selects the about button

Response: The about page loads

Stimulus: The user selects the home button

Response: The home page loads

**4.4.3 Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Function** | **Description** |
| FR-SUB1 | Subscribe | The application shall allow a user to subscribe to text message alerts for a live multiplayer game |
| FR-SUB2 | Unsubscribe | The application shall allow a user to unsubscribe to text message alerts for a live multiplayer game |

# Other Nonfunctional Requirements

## Performance Requirements

The user of our product has a reasonable expectation to be able to play Checkers without an undue amount of delay. Therefore, the responsiveness of the product must be swift enough to allow the user to enjoy their experience. During play against the computer, the AI opponent shall take no longer than 5 seconds to select and execute their move. While playing in a user vs. user scenario, communication between individual user’s machines should take no longer than 7 seconds to process after either user has submitted their move. This dedication to responsiveness should be met and prioritized by our team to as much of an extent as lies within our power, though obviously some delays would exist outside of our control (poor user-machine capabilities, ISP server delay, etc.)

## Safety Requirements

The use of our product shall not cause any physical or psychological harm to our users. All colors and effects shall be monitored and moderated so as not to aggravate common triggers of physical disorders, such as epilepsy.

## Security Requirements

During multiplayer sessions, our product shall transmit only that data which is necessary for the two players to conduct a game of Checkers together, and upon termination of our product, those channels of communication shall be ceased.

## Software Quality Attributes

Correctness : The rules of Checkers, and any subset thereof, shall be respected by our product and enforced on the players thereof. Therefore, our product shall maintain consistency of the state of the game and board and the pieces thereon. This consistency shall be maintained in solo and multiplayer matches.

Usability : The product shall have an ease of use that should allow for any user who is proficient in both the rules of Checkers and basic usage of personal computers shall be able to play a game of Checkers without unreasonable difficulty in beginning and playing the game.

## Business Rules

The product shall be freely available for any person(s) affiliated with Texas State University and their associates to use to play a game of Checkers. The source code, resources, and methods used to create the product shall remain trade secret to the Centipede Army and shall be disclosed to no one outside of them team with the exception of their professor and project manager, Jason Diaz.