# Software Requirements Specification

## for

# Game of Checkers

**Version 1.0 approved**

**Prepared by Natalie Ownby, Anna Cowsar, Cassandra Coyle, Niraj Regmee, and Alexander Rodriguez**

**Texas State University, CS 3398**

**September 10, 2018**

# Table of Contents

# Revision History

| Name | Date | Reason For Changes | Version |
|------|------|--------------------|---------|
| Anna Cowsar | 9/21/18 | Software Quality and Communications | 1 |
| Cassandra Coyle | 9/23/18 | Working on User Interface | 1.1 |
| Alexander R. | 9/25/18 | Hardware Interface Section 1 | 1.2 |
| Natalie Ownby | 9/26/18 | Defining Section 2.4-2.7 | 1.3 |
| Cassandra Coyle | 10/21/18 | Revised based on professor input | 1.4 |
| Natalie Ownby | 10/26/18 | Rewrote requirements and added requirements for extra features | 1.5 |
| Anna Cowsar | 10/26/18 | 3.1.1-3.1.7 and revisions | 1.6 |

# 1. Introduction

## 1.1  Purpose

The purpose of this document is to showcase an in-depth description of "A Game of Checkers" application. The SRS document is a baseline for the test plan and customer acceptance. Being that the SRS is a contract between a customer and developer; therefore, the SRS document is intended for the customers and developers. It will provide a detailed overview of the checkers game and parameters. This document will explain how the product shall interact with the user, how the user can use features, and display the constraints of how the application will operate during the duration of a game.

## 1.2  Document Conventions

The SRS was written in a way to section off topics, so that it is easier to follow along. Each section is sub-sectioned off, so that all of the sub-sections relay important information pertinent to its section. The font is standard Times New Roman font, size 11 with the section descriptions larger in font size and bold.

## 1.3  Intended Audience and Reading Suggestions

The SRS is intended for the customer and developers. The document is used by the project managers to understand the scope of the project. The SRS contains the introduction, an overall description, external interface requirements, system features, other nonfunctional requirements, and other requirements. It is organized in a manner that includes topics and subtopics. For instance, the introduction topic includes the purpose of the problem being solved (implementing a game of checkers), document conventions, the intended audience and reading suggestions, the scope of the project, and any references. The second topic is the overall description which includes the product perspective and functions, the user classes and characteristics, the operating environment for the product, the design and implementation constraints, user documentation, and any assumptions and dependencies. The third topic is about the user interface. This topic includes a description of the user interface and includes images of the intended interface. The fourth topic is the game play section, which involves a more detailed outline of what the user will go through as they play the checkers game. It involves the AI implementation, stimulus and response sequences, rule enforcement, which is how the rules will be enforced. Also, it includes the additional features like night mode, player name entry, music, and timer. The fifth topic is other nonfunctional requirements. This includes requirements such as the performance, safety, and security requirements of the game, software quality attributes, and any business rules. The last topic is a glossary which includes acronyms and their meanings. The SRS document should be read in order of appearance, so the reader can get a full understanding of the software requirements specifications of the game of checkers being implemented. All readers, whether tester, developer, customer, etc should read the SRS in its entirety to gain a full grasp of the product. This will lead to more common understanding of the product and its requirements and clear up potential miscommunication.

## 1.4  Product Scope

This product will allow the user to play an interactive game of checkers against the computer. The objective is to implement a simple AI that the user can play against with a user interface that resembles a checkerboard and pieces.

## 1.5  References

Rules of Checkers

- o   http://www.indepthinfo.com/checkers/

# 2.  Overall Description

## 2.1  Product Perspective

The "Game of Checkers" is a new, self-contained product. This product is not required to interface with other products and is standalone.

## 2.2  Product Functions

This product will perform the following major functions:
1) The product shall provide an AI implementation that is capable of playing checkers against the user
2) The product shall determine the winner and loser of each game of checkers
3) The product shall allow and enforce standard piece movements and positions defined under system requirements

## 2.3  User Classes and Characteristics

The primary user will be any person who wants to play a game of checkers against the computer (AI). The primary user should have a basic understanding of the rules of checkers and have experience using a computer. The user is not required to have any special knowledge, experience, or technical experience other than having the basic know how of playing a common game.

## 2.4  Operating Environment

This game will operate on a computer using a Windows, Linux or iOS operating system.

## 2.5  Design and Implementation Constraints

This software is not limited by any corporate or regulatory policies and has no security constraints. The software must display all text in English and shall not exceed a time of 1 second between the user input and the system response.

## 2.6  User Documentation

A user manual will be provided that describes all requirements for running the software including set up instructions for how to interact with the software and the actual rules of the game will be provided in the user manual. Further instructions for the checker game set up is in the README.md file.

## 2.7  Assumptions and Dependencies

It is assumed that the user has a computer and can run JS. It is assumed that the music API remains functional for the life of the software. It is also assumed that the user is not visually impaired or colorblind and can interact with the game as described in this document.

# 3.  External Interface Requirements

## 3.1  User Interfaces

For the User Interfaces, CSS, HTML, and JavaScript will be used to compose our Checkers game. The game board will have green and white checker pieces and starts out on a red and black board. The board can be changed to Night Mode which is a blue and black board with green and white pieces as shown in section 3.1.6 and 3.1.7. 3.1.1 through 3.1.7 shows each user interface component.

3.1.1 Customizable name interface:
It includes a textbox for entering the name. The "Submit Name" button is below the textbox which is for submitting a name to be displayed.

Submit Name
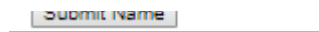
3.1.2 Example:
This is an example of what it would look like, if "Anna" was typed in the textbox. A Welcome, Anna! would appear.

3.1.3 Music player:
The music player plays classical music. Play button is shown displayed by the arrow sign.
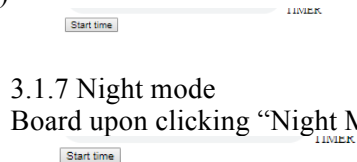
3.1.4 Music player:
The music player can pause the classical music. Pause button is shown displayed by the vertical equal sign—like a typical pause button:

3.1.5 Start time:
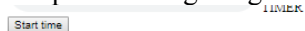Starts the timer for the game. This timer button calculates the game duration.

3.1.6 New Game:
Board upon clicking "New Game" button (The look of the screen of the initial load up is shown in 4.1)

Start time

3.1.7 Night mode
Board upon clicking "Night Mode" button

Start time

o An 8 by 8 board shall be displayed.
o The board shall have black and red squares for new game, or blue and black squares for night mode.
o The board shall have green checker pieces for one player and white checker pieces for another player.
o Green and white checker pieces shall be displayed on the board.
o The user interface layout shall be constrained to fit on a standard window.
o A customizable name feature shall be displayed.
o A timer shall be displayed below the board.
o A night mode button shall be displayed.
o A new game button shall be displayed.
o A toggle-able sound button shall be displayed allowing for the user to select to either pause or play music while they play the game.
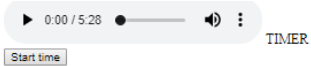
# 4. System Features

## 4.1  Game-Play

Game-Play lays out the game play of the process the user will take in playing the game of checkers.

4.1.1 Description and Priority
• When the checkers game is played the user shall come across:

1. `▶ 0:00 / 5:28 ●———— ◀) ⋮`  TIMER
   `Start time`

   a. Note: this page will not include a checker board yet, only after the new game button is clicked will the board be displayed.

2. The system shall allow the user to type in their name and submit it to the submit name textbox above the submit name button.

   a. Refer to section 3.1.1 and 3.1.2 for the way it looks in the user interface

3. The system shall allow the user to start a new game of checkers by clicking on the new game button

   a. Refer to 3.1.6 for the way it looks in the user interface

4. The system shall display the checker board game with green and white checkers

   a. Refer to section 3.16 for the way it looks in the user interface.

5. The system shall display the resulting board of the users selected checker move.

6. They system shall display the resulting board of the AIs selected checker move.

7. The system shall repeat steps 5 and 6 until the pieces:

   a. Get kinged: meaning a checker piece reached the opposite end of the game board and is now either

      i. Yellow if originally white

      ii. Olive green if originally a basic colored green

   b. Note: once kinged a checker shall move in any direction diagonally

8. The system shall continue to play and update the checker game until there are only checker pieces of one-color left.

9. Once there is only one-color piece left the system shall declare the winner and restart the game.

The product shall provide a user interface that resembles an 8 row by 8 column checkers board with circular pieces (green and white). The user interface is a high priority.

#### 4.1.2    Stimulus/Response Sequences
Once a piece and its target tile is selected, if this is a valid move, the system shall move the piece to that location. If it is not a valid move, the system shall not move the piece.

#### 4.1.3     Functional Requirements
REQ-1: The user shall be able to select a piece by clicking on it : The user shall be able to select a piece by clicking on it

REQ-2:        The piece shall move to the target position selected if it is a valid move

REQ-3:  The piece shall remain in its location if the selected position is invalid

## 4.2  AI Implementation

4.2.1    Description and Priority

The product implements an AI that can play checkers against the user. This AI implementation will move the checkers piece to a valid location and will take turns with the user. This is a high priority. Failure to implement the AI will result in the user not being able to play the game.

4.2.2    Stimulus/Response Sequences

After the user has made a valid move (selected a piece and selected its new location) the AI moves its piece. This is displayed in the user interface.

4.2.3    Functional Requirements

REQ-1: The AI shall choose a piece and a target location for the piece in accordance with the rules of checkers.

REQ-2: The AI shall wait until the user has made a valid move to select and move a piece.

REQ-3: The AI shall follow the rules of checkers.

## 4.3  Rule Enforcement

4.3.1    Description and Priority

The product shall not allow the user or AI to move a piece in a nonlegal way according to the rules of checkers. This is of high importance.

4.3.2    Stimulus/Response Sequences

The user selects the piece they would like to move and the target location to which they would like to move it. The program ensures the move follows the rules of checkers. If it does, the piece is moved, and the UI is updated. If it is not a legal move, then the piece remains in its same location. This idea is the same for the AI implementation. If the AI selects a piece and its target location in order for the piece to move it must be valid, if it is not a valid move then the piece is not moved.

4.3.3    Functional Requirements

REQ-1:        The system shall not move any piece if the user selects an invalid move.

REQ-2: The system shall remove the piece from the board when it is jumped.

REQ-3: The system shall "king" a piece by updating its appearance when it reaches the other side of the board. The "kinged" white piece shall turn yellow and a green "kinged" piece shall turn olive green.

REQ-4: The system shall allow a kinged piece to move in any diagonal direction one space. It shall not alert the user that backwards diagonal is an invalid move.

REQ-5: The program shall declare a winner via a pop up when either the user or the AI has "jumped" all of the opponent's pieces, i.e., all of the opponent's pieces are inactive

## 4.4      Night Mode

4.4.1     Description and Priority

This product shall allow the user to select "Night Mode" and update the UI appropriately. When Night Mode is selected the colors of the board will be changed from the regular alternating black and red to black and blue. Reference section 3.1.7 to look at an image of the Night Mode.

4.4.2     Stimulus/Response Sequences
The user selects "Night Mode", the user interface is updated to show the night mode version of the board, pieces and background. The user selects "Night Mode" again and the user interface returns to the normal, non-night mode colors.

4.4.3 Functional Requirements
REQ-1: The program shall include a button that allows the user to select night mode. (refer to section 3.1.6 and 3.1.7 that show the button on the top left above the board by "New Game").
REQ-2: If the program is already in night mode and the user selects night mode again, the user interface shall return to non-night mode.

## 4.5      Player Name Entry

4.5.1     Description and Priority
This feature allows the user to enter their name. This will allow the system to display the name of the player throughout the game and the name of the winner at the end. This is a low priority as it does not affect the functionality of the actual checkers game. Reference section 3.1.5 to look at an image of the Player Name Entry.

4.5.2     Stimulus / Response Sequences
When the user starts the game, they will be able to enter their name in a textbox at the top of the game.

4.5.3     Functional Requirements
REQ-1: The system shall allow the user to enter their name in the textbox.
REQ-2: The system shall update the player name based on the user input.
REQ-3: The system shall display the user name as the winner if the user wins.

## 4.6      Music

4.6.1     Description and Priority
The music will be playable through the web page by clicking the play button. The music feature is not a priority function since it doesn't affect the functionality of the checkers game. The music uses this link which has one track of playable classical music: https://www.mfiles.co.uk/mp3-downloads/moonlight-movement1.mp3 Reference section 3.1.3 to look at an image of the music feature.

4.6.2     Stimulus / Response Sequences
Once the game is launched the music can be played. The music can be played by clicking the play button (arrow symbol) or paused by clicking on the vertical equal sign.

     4.6.3    Functional Requirements
            REQ-1: The system shall start the music when the play button is pressed.
            REQ-2: The system shall pause the music when the pause button is pressed.

## 4.4  Timer

     4.4.1    Description and Priority
            The timer feature displays the duration of the game. The timer is a low priority feature, as it does not affect the functionality of the game. Reference section 3.1.5 to look at an image of the Timer.

     4.1.2    Stimulus / Response Sequences
            The user starts the game and clicks the timer button in the bottom right corner. The timer keeps a duration count of how long since the timer button has been clicked.

     4.4.3    Functional Requirements
            REQ-1: The system shall start the timer when the "start timer" button is clicked

# 5. Other Nonfunctional Requirements

## 5.1  Performance Requirements

Performance requirements include a quick response time. The checkers game and AI should not lag. The checkers game is not intended to be scalable in that a checkers game is played one game at a time between a user and AI. The platform shall include client-side JavaScript to minimize complicated system overhead. The rationale behind these decisions are so that the user shall be able to play a simple game of checkers, like they would if it were a real checkers board game. The AI should not take longer than a normal 2 player checkers game in real life on a hard-physical board would take. Design decisions should reflect this thought process of making a simple checkers game for a user to play. The timing for this real time system of the user versus the AI should be to demonstrate a real life physical hard board game of checkers, so the AI should not lag in moving pieces. Therefore, the longest time the AI should take to move a piece is 1 minute. The board shall update as to not exceed a time of 1 second between the user input and the system response.

## 5.2  Safety Requirements

There are no safety requirements associated with this product.

## 5.3  Security Requirements

There are no real security or privacy risks. The system shall allow the user to customize their name if they choose to do. There are no user identity authentication requirements because user authentication is not a requirement of the checkers game.

## 5.4  Software Quality Attributes

Quality characteristics for the checkers game are availability, portability, and usability. This game shall be widely available and used on any PC or Mac computer with internet access, and easy to understand and use.

## 5.5  Business Rules

Individuals involved in the game of checkers are the user and the AI. The roles of the user and AI involve following the rules of the game of checkers outlined in the link found in section 1.5.

# Appendix A: Glossary

*Player/User: The user and the AI playing the checkers game.*
*AI: The algorithm that will be playing against the user who is playing the checkers game.*
*JS: Java Script is a high-level interpreted programming language*
*HTML: Hypertext Markup language used for creating web pages and applications*
*CSS: Cascading Style Sheets used to describe the presentation of a document written in HTML*
*REQ: requirement*