King Me

Requirements Specification

|  |  |
| --- | --- |
| **Group Members** | Vincent Finn, Chris Deslongchamp, Matthew Quaschnick, William Brown |
| **Faculty Advisor** | Dr. Filippos Vokolos, Ph. D. |
| **Project Stakeholder** | Dr. Filippos Vokolos, Ph. D. |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Change** | **Revision** |
| Chris Deslongchamp | 7/17/2018 | First Draft – Sections Outlined | 0.9 |
| Chris Deslongchamp | 7/24/2018 | Second Draft – Content added to all sections | 1.0 |
|  | 7/30/2018 | Revisions | 1.1 |

Table of Contents

[Revision History 2](#_Toc520210095)

[Introduction 4](#_Toc520210096)

[Purpose of Document 4](#_Toc520210097)

[Scope of Document 4](#_Toc520210098)

[Overview of Document 4](#_Toc520210099)

[Description 5](#_Toc520210100)

[Product Perspective 5](#_Toc520210101)

[Player PC Interface 5](#_Toc520210102)

[Product Functions 5](#_Toc520210103)

[Player Functionality 5](#_Toc520210104)

[Server Functionality 6](#_Toc520210105)

[User Description 6](#_Toc520210106)

[Assumptions and Dependencies 6](#_Toc520210107)

[C# 6](#_Toc520210108)

[.NET GUI 6](#_Toc520210109)

[C# Networking 6](#_Toc520210110)

[Art Assets 6](#_Toc520210111)

[Requirements Apportioning 6](#_Toc520210112)

[Functional Requirements 7](#_Toc520210113)

[Player 7](#_Toc520210114)

[R1.1 Starting a Game 7](#_Toc520210115)

[Running a Game 7](#_Toc520210116)

[R2.1 Loading a Game 7](#_Toc520210117)

[R2.2 On Player’s Turn 8](#_Toc520210118)

[R2.3 Off-Turn Player 8](#_Toc520210119)

[R2.4 Paused Game 8](#_Toc520210120)

[R2.5 Game Over 9](#_Toc520210121)

[Non-Functional Requirements 9](#_Toc520210122)

[Network Connection 9](#_Toc520210123)

[R3.1 Player Connection 9](#_Toc520210124)

[Host Operating System Requirements 9](#_Toc520210125)

[R4.1 OS Support 9](#_Toc520210126)

[Accessibility 9](#_Toc520210127)

[R5.1 Game Download 9](#_Toc520210128)

[R5.2 Small File Size 9](#_Toc520210129)

[R5.3 Server Connection 9](#_Toc520210130)

[Playtesting 10](#_Toc520210131)

[User Interface 10](#_Toc520210132)

# Introduction

## Purpose of Document

This document will provide all of the requirements for the game King Me. It will serve as a reference for the developers and the customer for developing the final version of the game.

## Scope of Document

This document will contain enough information such that a developer will be able to easily translate the requirements to code without ambiguity.

## Overview of Document

This document will contain functional and nonfunctional requirements for the game King Me, as well as use cases, diagrams, and UI mock-ups. The game contains two components that allow two players to play on a PC and a third component that runs connects the first two components. The PC components will be referred to as **Player 1** and **Player 2** for the player that moves first and the player that moves second. The component that connects them will be referred to as the **server**. This document will also contain mock-ups for the graphical user interface and use cases.

# Description

## Product Perspective

King Me is an online two-player checkers game designed to be played on two computers across the internet. One player will host the game while the second player joins the host’s game. Once both players have joined, one player will be randomly assigned as Player 1 and the other will be assigned Player 2. Afterwards, the checkers game will launch, and players will take turns making their moves within a certain time frame with Player 1 making the first move.

INSERT IMAGE OF GAME WITH UI

The game is intended to run on two PCs, Player 1 and Player 2. Players will make their move on their own PC and that move will be translated to the other’s PC through the server connection. Once the move has been applied to both Player’s screens, the player who didn’t make a move will take their turn. This will continue until one player has won.

## Player PC Interface

The PC Interface will contain a start menu with options to host or join a game so players can join the game. The PC interface will then display the game board, pieces, which player will go first, who turn it is, and a pause option.

# Product Functions

## Player Functionality

The Player will have the following functionality:

* Ability to host or join a game
* Ability to accept a move
* Ability to conform a move is legal
* Ability to execute a legal move
* Ability to execute a legal capture
* Ability to pause a game
* Ability to surrender

## Server Functionality

The Server will have the following functionality:

* Ability to connect two players
* Ability to communicate each player’s move
* Ability to close connection after a player has won, surrendered, or tied a game

# User Description

The ideal users for King Me would be two people located across from each other in the same room. However, player’s do not have to be in the same room as long as they have an internet connection.

# Assumptions and Dependencies

## C#

C# is an object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications and more. This program will be developed using C# version 7.3 using the .Net Core Framework. The project fully depends on this Language/Framework. If Microsoft stops development of C# or .Net the project will be halted and will have to be migrated over to a new Language such as Java.

## .NET GUI

The GUI will be developed using Visual Studio’s Windows Form editor found when creating a Visual C# Windows Form Application. The GUI will utilize the art assets created for this project which are discussed later under the “Art Assets” section. The GUI is required as it is vital to the other systems working properly.

## C# Networking

C# contains multiple built in libraries to help with Peer to Peer connections. Specifically the System.Net namespace contains classes that provide support for creating applications that send and receive data. Online play is critical to the user experience, thus getting this system functioning is required.

## Art Assets

The various members of the team will create the art assets. These assets while not required to make the game functional, will make the game significantly easier and more enjoyable to play, thus are required to be completed. They will be created using Adobe Photoshop CC. These assets must be scheduled and completed on time as to not impede development of other systems.

# Requirements Apportioning

|  |  |
| --- | --- |
| Priority Level | Description |
| 1 | Priority 1 requirements are essential to the product and must be in the final build. These requirements must be tested and verified to ensure proper functionality. |
| 2 | Priority 2 requirements are not required for the final build, but will be provided if there is sufficient time. The system will be designed such that it is extendable to easily incorporate these requirements at a later time. |
| 3 | Priority 3 requirements are not required, and will not be considered in the design of the system. If sufficient time remains the requirement will be incorporated. |

# Functional Requirements

## Player

INSERT MOCK UI FOR JOINING GAME

### R1.1 Starting a Game

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Title | Description | Priority |
| R1.1.1 | Host Game | Player will be able to chose to host a game. | 1 |
| R1.1.2 | Join Game | Player will be able to join a game. | 1 |

## Running a Game

INSERT MOCK UI FOR GAME

### R2.1 Loading a Game

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Title | Description | Priority |
| R2.1.1 | Player Selection | Players are randomly assigned either Player 1 or Player 2. | 1 |
| R2.1.2 | Load Game Board | A green and white checkered board is displayed on both players screens. The board will contain 64 squares alternating in color and be arranged in 8 rows and 8 columns. The square in the top left corner of the board will be white followed by a green square and continue the alteration for the rest of the board. | 1 |
| R2.1.3 | Load Player Tokens | 12 Red circle tokens and 12 White circle tokens will be placed on the board. The tokens will only be placed on green board squares. White will be placed in the last three rows while Red will be placed in the top 3 rows. Player 1 will be Red and Player 2 will be White. | 1 |
| R2.1.4 | Load Move Timer | A timer will be placed in the top right corner of the screen and will count down at the start of each players turn and will reset once a player has made their move. | 2 |
| R2.1.5 | Show Player’s Id | On the bottom center of the screen Players will be alerted of which Player they have been assigned, either Player 1 or Player 2. | 1 |
| R2.1.6 | Show Current Move Player | Above the Move Timer, a message will show which Player’s turn it is. | 1 |

### R2.2 On Player’s Turn

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Title | Description | Priority |
| R2.2.1 | Move Selection | On the Player’s turn, they will be able to select a token they want to move and where they want to move it. | 1 |
| R2.2.2 | Move is checked if it is a legal move | Once a Player has submitted a move, the move is checked to make sure it is allowed. | 1 |
| R2.2.3 | Move is legal | If the move is legal, the move will be made on the player’s screen. The timer will stop and reset. The other player will be shown the move and their turn will begin. | 1 |
| R2.2.4 | Move is illegal | If the move is illegal, the move will not be made. The current turn’s player will be alerted to choose a different move. The timer will continue to count down. | 1 |
| R2.2.5 | Pause the game | The player can request a pause. | 3 |
| R2.2.6 | Alert of Start of Turn | The player will be alerted that their turn has begun. | 2 |
| R2.2.7 | Surrender | Player can choose to surrender. | 3 |

### R2.3 Off-Turn Player

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Title | Description | Priority |
| R2.3.1 | Other Player has made a legal move | The board will update with the other player’s move. | 1 |
| R2.3.2 | Accept or Decline Pause Request | The player will be alerted that the other player would like to pause the game. They will be allowed to accept or decline the pause. | 3 |

### R2.4 Paused Game

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Title | Description | Priority |
| R2.4.1 | Pause Message Displayed | A message will display stating that the game is paused and by which player. | 3 |
| R2.4.2 | Request Game Resume | A button will be displayed to allow a player to request a game to resume. Once a request is made the other player can choose to accept the resume. | 3 |

### R2.5 Game Over

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Title | Description | Priority |
| R2.5.1 | Player has won | Once a player has met the win condition, both players will be alerted, and the winning player will be displayed. | 1 |
| R2.5.2 | Game is a tie | Once a tie has occurred, both players will be alerted of the tie. | 1 |
| R2.5.3 | Player has surrendered | Once a player surrenders, both players will be alerted, and the non-surrendering player will be declared the winner. | 3 |
| R2.5.4 | Exit Game | Once one of the end game messages has been displayed, players can choose to exit the game. | 1 |
| R2.5.5 | Rematch | Once one of the end game messages has been displayed, players can choose to start a new game with each other. They can choose to keep the same turn order or switch turn order. Player 1 would become Player 2 and vice versa. | 2 |

# Non-Functional Requirements

## Network Connection

### R3.1 Player Connection

Players should remain connected to each other the entirety of the game. There should be no delay from when a player submits a move, the move is confirmed to be legal, and the legal move is made on both player’s screen. This will be a constant priority for the team. **Priority 1**

## Host Operating System Requirements

### R4.1 OS Support

The game expected to support Windows 7,8, and 10. The game may support Mac OS 10.X and above. Other Operating systems will not be test and Mac OS is not guaranteed to be supported. **Priority 1**

## Accessibility

### R5.1 Game Download

The game will be available for download from a GitHub link that is public. **Priority 1**

### R5.2 Small File Size

The file size for the game will be kept small to decrease download times. **Priority 2**

### R5.3 Server Connection

The server will always allow players to connect to each other as long as they maintain an internet connection. **Priority 1**

## Playtesting

INSERT INFO ABOUT HOW WERE GOING TO PLAYTEST

# User Interface

## Start Screen

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Title | Description | Priority |
|  |  |  |  |

## Game Screen

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Title | Description | Priority |
|  |  |  |  |

## Game Over Screen

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Title | Description | Priority |
|  |  |  |  |

## Rules Screen

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Title | Description | Priority |
|  |  |  |  |

# Use Cases

When the players want to play a game of chess, realize they don’t know how to play chess, and then decide to play checkers. No one owns a checkers set, but everyone has an Internet connection. That’s where we come in.

# Glossary