Chinese Checkers

# **1. Introduction**

This project is to make a six player game of Chinese checkers. This game will be playable over a network with other human players.

# **2. System-Wide Functional Requirements**

[REQ-0001] This game will allow up to six players to play Chinese Checkers over a network.

[REQ-0005] This game will display a Chinese Checkers board which is shaped like a six pointed star.

[REQ-0010] The board will have ‘holes’ within a hexagonal lattice such that each of the six point shall have ten holes and the hexagon in the middle will have 61 holes and sides of 5 holes long.

[REQ-0015]A player will be able to host a game.

[REQ-0020]A player will be able to join an existing game that has not been started and has less than six players.

[REQ-0025]Upon joining or hosting a game a player shall be able to chose one of six colors that has not been chosen by another player.

[REQ-0030]The hosting player may be able to add computer players.

[REQ-0035]The hosting player will be able to start the game.

[REQ-0040] Each player shall be assigned one of the six points on the board as their starting location

[REQ-0045] The holes in the starting locations for each player will be filled with ten pieces that are that player's color.

[REQ-0050]Upon starting a game a random player will be chosen to move first.

[REQ-0055]Players will take turns depending on the starting locations in a clockwise direction.

[REQ-0060]Players shall only be able to move a piece on their turn.

[REQ-0065]Players shall only be able to move a piece of the same color.

[REQ-0070]Players shall only be able to move one piece on their turn.

[REQ-0075]A players must make a move on their turn if able.

[REQ-0080]A players shall be able to ‘walk’ a peace by moving it to an empty adjacent hole if there is one.

[REQ-0085]A players will only be able to walk a piece once per turn.

[REQ-0090]A players shall be able to ‘hop’ a piece by moving it to an empty hole directly on the other side of an adjacent filled hole.

[REQ-0095]A player shall be able to continue hoping the same piece unlimited times on their turn.

[REQ-0100]A player shall not be able to walk a piece after hoping it or hop a piece after walking it.

[REQ-0105]Play will continue until a player is able to move all of his pieces into the point on the opposite side of the board.

[REQ-0110]The player that is able to move all of his pieces into the point on the opposite side will be declared the winner.

# **3. System Qualities**

## **3.1 Usability**

[USA-001] Movement of pieces shall be able to be controlled using the mouse.

[USA-005] A tutorial shall be provided in a documentation. See [SUP-0015]

[USA-006] Possible moves shall be highlighted when selecting a piece on a player’s turn. See [LOO-0025].

## **3.2 Reliability**

[REL-00] Supports up to 6 players.

[REL-03] The game may save the state so it can be resumed at a later time.

[REL-05] Online server shall be provided by the teacher

## **3.3 Performance**

[PERF-00] Game shall perform on David’s computer.

[PERF-02] Players will be notified when another player exits the game

[PERF-04] A computer player shall act in less than 10 seconds.

## **3.4 Supportability**

[SUP-0000] The application shall be at least compatible with Windows operating systems.

[SUP-0001] The application might be compatible with Mac and Linux systems.

[SUP-0005] The application shall allow the users to choose to play between 2 - 6 players.

[SUP-0010] The user might be able to choose an option to play the game against the computer.

[SUP-0015] The application shall include a help button in the drop down menu to explain the rules of chinese checkers.

# **4. System Interfaces**

## **4.1 User Interfaces**

### **4.1.1 Look & Feel**

[LOO-0000] The user interface might have a Chinese style font.

[LOO-0005] The application shall have the option to play traditional Chinese music in the background.

[LOO-0010] The user interface might have options of different music and themes accessible from an option menu.

[LOO-0015] Each of the 6 triangles in the star of the board shall have a unique color from the rest of the starting locations.

[LOO-0020] The color of the board shall be different from the starting colors.

[LOO-0025] The holes on the board shall light up if it is an eligible place for the selected piece to move.

[LOO-0030] If no piece is selected on a player’s turn, the pieces that can legally move shall light up.

### **4.1.2 Layout and Navigation Requirements**

[NAV-0001] There shall be an exit button to quit the application on the top right corner of the application. This button shall be labeled with an “X”.

[Nav-0002] There shall be a help drop down menu which shall contain the tutorial specified in requirement [USA-005].

### **4.1.3 Consistency**

### **4.1.4 User Personalization & Customization Requirements**

## **4.2 Interfaces to External Systems or Devices**

[IES-0000] The game’s network system/protocol will be determined by the instructor.

### **4.2.1 Software Interfaces**

### **4.2.2 Hardware Interfaces**

### **4.2.3 Communications Interfaces**

[IES-0001] The game must allow for multiplayer between 2 to 6 players over a network.

# **5. System Constraints**

[RES-0001] The game shall be programmed in C#.

[RES-0002] The game shall support Windows 7

[RES-0003] Support for other operating systems may be included.

# **6. System Documentation**

[DOC-0001] Programmers shall not use non-descriptive variable names for anything that will exist for more than one function

\*Note: Internal variables can be foo or bar or i, etc. but class variables shall be named as what they are.

\*Note: non-descriptive is defined as anything similar to foo, bar, x, y, z, i, j, k, etc. These are fine for iterators and similar purposes, but for data storage and class members the names should be what they are.

[DOC-0002] There shall exist a readme file that no one will likely read, but shall contain basic gameplay instructions, and the names of the developers.

# **7. Requirement Verification Plan**

|  |  |  |
| --- | --- | --- |
| Requirement ID | Verification Description | Method |
| [REQ-0001] -  [REQ-0110] | Demonstration that a game is able to be hosted joined and played according to the rules of Chinese Checkers | Demo |
| [RES-0001] | Visual Inspection to check if C# was used in the code. | Inspection |
| [RES-0002] | Demonstrate the game running on Windows 7. | Demo |
| [RES-0003] | Demonstrate on other OS’s if support for them is added. | Demo |
| [DOC-0001] | Visual Inspection of the code to verify that no classes have members known as foo, bar, x, y, i, j, k, etc. | Inspection |
| [DOC-0002] | Visual? I mean if it exists it exists, it’s a readme file, not skynet. | Inspection |
| [IES-0000] | Demonstrate a network exists. | Demo |
| [IES-0001] | Demonstrate 2-6 players can be in one game. | Demo |
| [USA-0005], [NAV-0002] | Show that the tutorial can be opened. | Demo |
| [USA-006] | Play a game showing that legal moves are highlighted. | Demo |
| [PERF-00] | Run a game on David’s laptop. | Demo |
| [PERF-02] | Play a game and have a player leave during the game. | Demo |
| [PERF-04] | Play a game on David’s laptop and time each turn. | Demo |
| [NAV-0001] | Show that the interface has an X in the top corner that closes application. | Demo |
|  |  |  |