Software Requirements Specification

Space Checkers

Version 1.0

Prepared by:

Alan Tsai

Klaus Nuredini

James Swanick

Deepak Yadav

Table Of Contents

1. Introduction
    a. Purpose
    b. Scope
    c. Overview
2. Description
    a. Product perspective
        i. Operation System Interface
    b. Product Functions
    c. User Description
    d. Assumptions and Dependencies
    e. Requirements Apportioning
3. Specific Requirements
        i. Functional Requirements
    b. Non-functional Requirements
        i. Extensibility
        ii. Maintainability
    c. User Interface
        i. Application Interface
4. Use Cases
    a. Use Cases
    b. Use case diagram
    c. Activity Diagram
5. System evolution

1 Introduction

This section gives both a scope description and overview of everything included in this SRS document, and a purpose for the document.

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Checkers software. It will illustrate the purpose and complete declaration for the development of the system. It will also explain the system constraints, the interface, and the interactions between the server, clients and OS application.

1.2 Scope

This document describes the software requirements for Checkers. This document is intended for the use of the developers, testers, and users of the application.

1.3 Overview

This document will discuss information regarding the application Checkers and its functional requirements and nonfunctional requirements.

2 Description

2.1 Product perspective

Checkers is a software application that allows two people to play a game of checkers interactively, from remote locations. The objective is to develop a software that provides an environment for two people to play a game against each other. The application will have interactive experience as close to the physical playing experience as possible.

Official rules for Checkers can be found here:
http://boardgames.about.com/cs/checkersdraughts/ht/play_checkers.htm

2.1.1 Operation User Interface

The user will start the application from a local machine. The main interface will consist of a window with menu buttons allowing the user to start the checkers game or exit the application. When the user decides to start a game and there is no other player online at the moment, then the interface will display a screen informing the user that the application is waiting for another user to start a game. When there is already someone else who has started the application and decided to start a game, then the interface will consist of a screen, which will have a checkerboard and checkers already laid out and ready for the user to start playing.

2.2 Product Functions

Checkers will provide the following functions:

- Allow the user to get online and ask for starting a checkers game against another player
- Allow the user to move a checker piece to a new position
- Validate each move the user tries to do according to the rules of checkers
- Update configuration of the checkerboard after each valid move
- Displaying the rules of the checkers game on the screen

2.3 User Description

The user can be any person who wants or intends to play a checkers game against some other person from a remote location. Each of these persons is considered a user. A user will utilize the application by asking to start a game and then hold until another user starts the application and decides to start a game as well, in which case, the application will display the standard checkerboard along with the basic rules of checkers on the side of the screen. At that point, the user can start moving the checker pieces according to the rules, until the game ends in a win, lose, or draw.

2.4 Assumptions and Dependencies

For this product, the assumption is that the network connection speed on both remote locations, so for both users, is fast and decent enough in sending data so that the checkerboard configuration updates quickly, at least at a rate acceptable for the user interactive experience to be as close to the physical playing experience as possible.

2.5 Requirements Apportioning

The priority levels for the requirements are:

Priority Level	**Description**
1	This is the highest priority level; requirements of this level are essential for the application’s functionality and must be fully satisfied and verified in order for the software to be released.
2	Requirements of this level are not required, but are highly desirable. Requirements of this level are expected to be satisfied, but not fully verified.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 OS Application

The application will contain three main views: the Menu, and the Game screen.

R1.1 Menu Screen

R.1.1.1 The menu will allow the player to attempt to connect to the server. **Priority 1**

R.1.1.2 The menu will notify the player if they fail to connect to the server. **Priority 1**

R.1.1.3 The menu will notify the player if there is already the maximum number of players. (2, currently) **Priority 1**

R 1.1.4 The menu will allow the player to close the application. **Priority 1**

R1.2 Game Screen

R.1.2.1 The game screen displays the checkers board. **Priority 1**

R.1.2.2 When the user clicks on a checker piece, the valid spots where it can be moved should be highlighted. **Priority 1**

R.1.2.3 When the user clicks on new spot, the checkerboard gets updated if the move is valid, and notify the user if move is invalid. **Priority 1**

R.1.2.2 The game screen will contain a chat box that allows the players to communicate. **Priority 2**

R.1.2.3 The game screen will contain a help button that opens the Help Pop-up window. **Priority 2**

R.1.2.4 A button “Quit” that allows the player to quit the game and return to the menu. **Priority 1**

R1.3 Help Pop-up Window

R1.3.1 The Help Pop-up window will display all the checkers rules so that the user can access it when needed. **Priority 2**

R1.3.2 The window should have a “Back to Game” button to allow the user to get back to the Main Screen when finished looking at the rules. **Priority 2**

R1.4 Chat Box

R.1.4.1 The chat box window will be displayed on the right side of the game screen. **Priority 2**

R.1.4.2 The window will contain a textbox allowing the players to type messages. **Priority 2**

R.1.4.3 Contains a button “Enter” that sends the user’s message. **Priority 2**

R1.5 End Game Screen

R.1.5.1 The end game screen displays once the game ends. **Priority 1**

R1.5.2 The screen displays win, lose or draw depending on the game result. **Priority 1**

R1.5.3 This screen should have a button “Start New Game” to allow the user to connect to server and start a new game. **Priority 1**

R1.5.4 It should also have a button “Quit” to allow the user to exit out of the game. **Priority 1**

3.2 Non-functional Requirements

3.2.1 Extensibility

The checkers program will be coded in such a way that the server maintains an “instance” of a checkers game, allowing for multiple instances of the game to be running at the same time in the future.

3.2.2 Maintainability

A module containing program logic for the Checkers ruleset will be developed separately, to be used by both the the server and the client, in order to ensure rules logic is not modified inconsistently.

The server program will be developed separately from the client, so that both can be maintained separately.

3.3 Application Interface

Figure 1 is the Menu Screen. It allows the players to connect to the server and start a game or to exit the game.

Figure 2 is the Game Screen. It displays the checkers board, chat box, help button and quit button.

Figure 3 is the Game Screen. It displays the help window.

Figure 4 is the End game screen. It displays the result of the game and allows the players to rematch or exit the game.

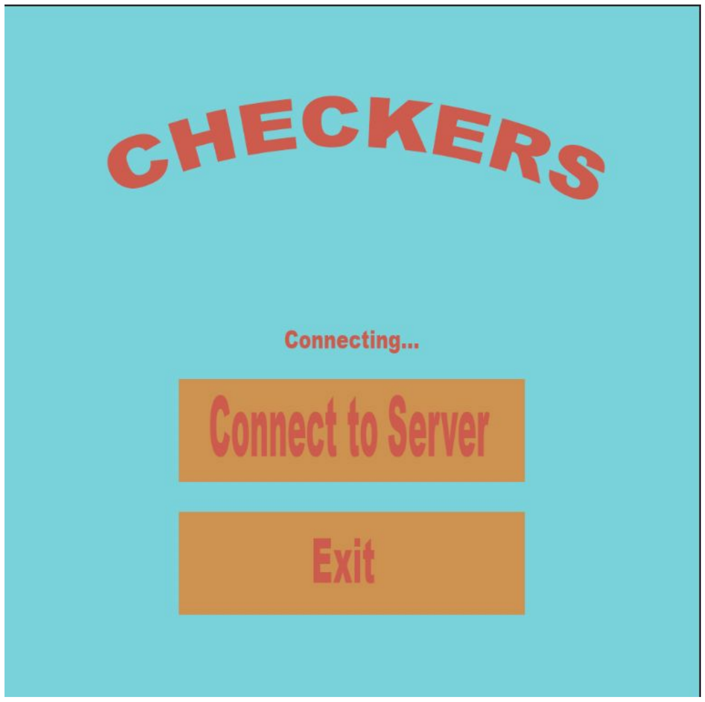


Figure 1: Main Menu

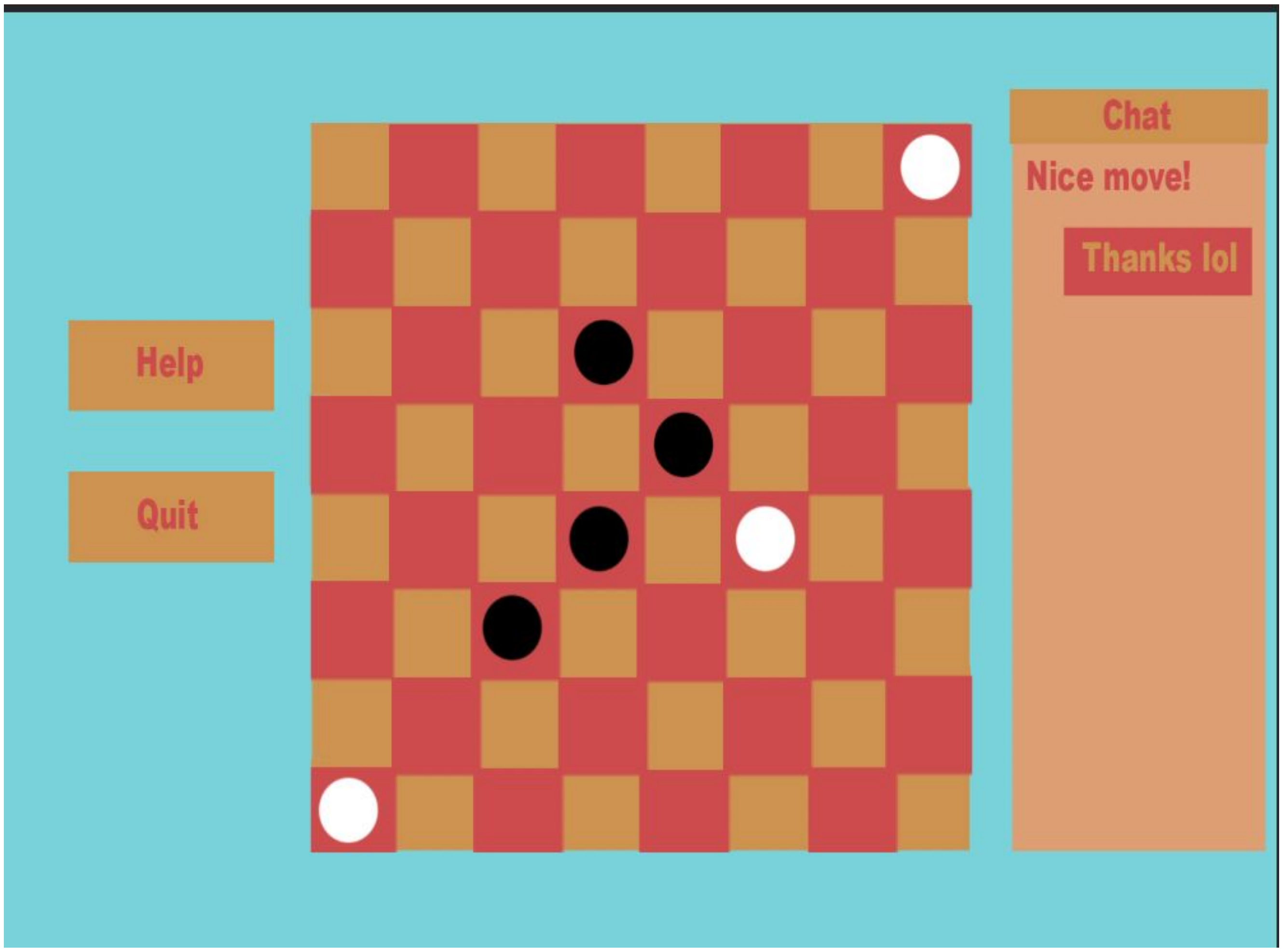
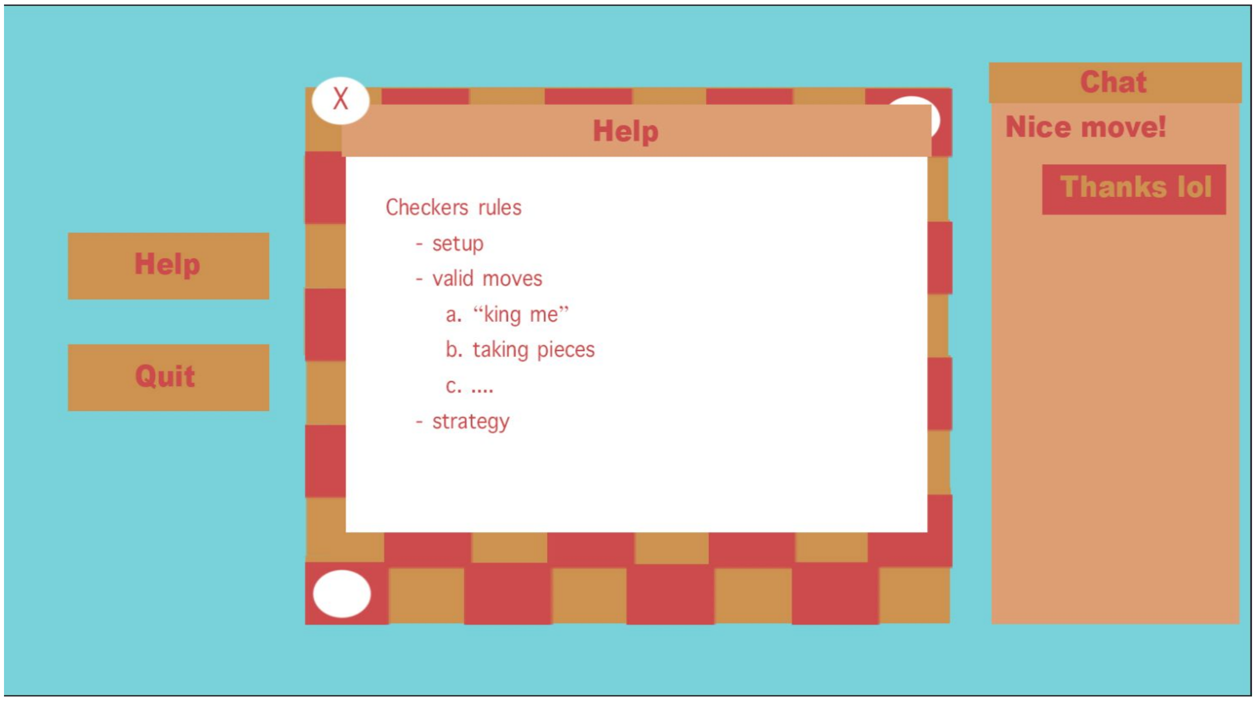


Figure 2: The Game Screen displays the checkers board, chat box, help button and quit button

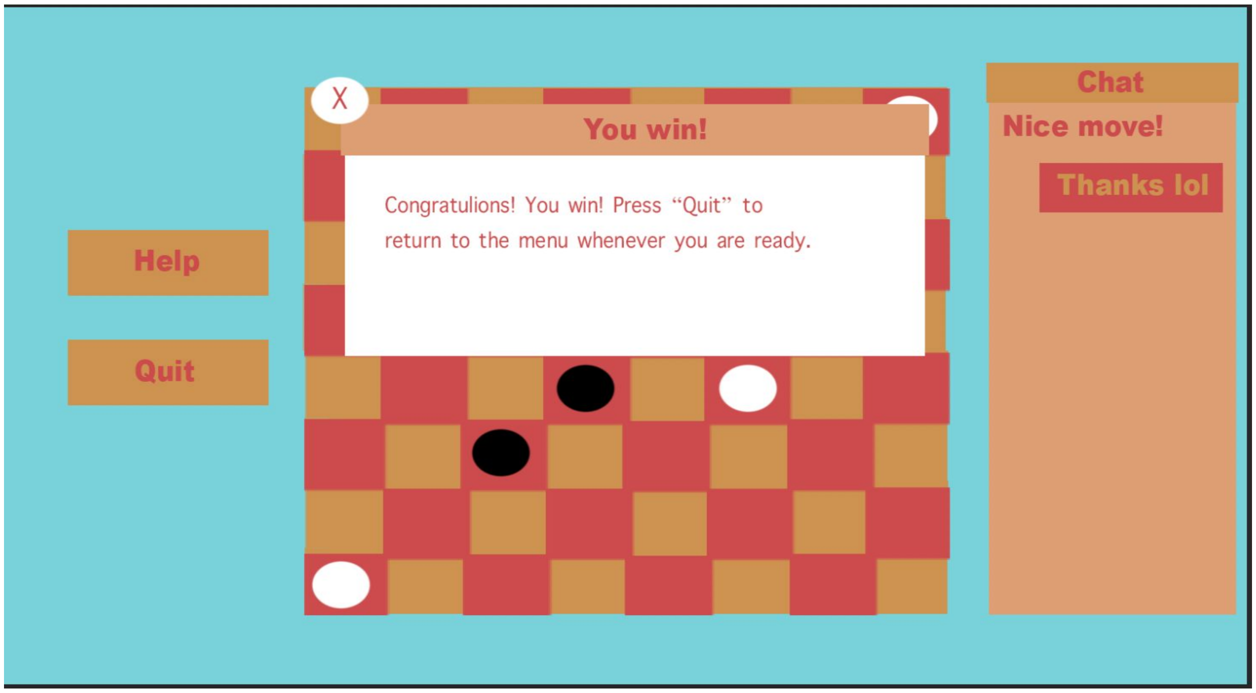


Nice move!

Thanks lol

Checkers rules
- setup
- valid moves
  - a. "king me"
  - b. taking pieces
  - c. ....
- strategy

Figure 3: Game Screen with help Pop-up window displayed



Nice move!

Thanks lol

Congratulions! You win! Press "Quit" to return to the menu whenever you are ready.

Figure 4 is the endgame screen. It displays the result of the game and allows the players to rematch or exit the game.

4. Use Cases

4.1 Use Case Flow

4.1.1 Connect to server

Preconditions

- The user has installed the application to his/her machine.
- User has access to network.

Main Flow

- User/Player will start application and request for a connection to server and server will respond to the request by allowing or declining request.

Post-conditions

- The user will wait for another user to connect to server in order to start game.

4.1.3 User Starts Playing Checkers

Preconditions

- Another user has connected successfully to the server.
- The Game Screen has been launched.

Main Flow

- Player with black pieces (Player1) will move first and then both players will alternate moves until game ends.

Post-conditions

- If move is valid, the checkerboard updates for both players, and the application lets the other user make a move.
- If move is invalid, user is notified by a message on the screen & given the chance to try again.

4.1.4 Game Ends

Preconditions

- Both users are still connected to the server.

- The Game Screen has been launched.

Main Flow

- The checkerboard goes to a state where one user has won or there is a draw.

Post-conditions

- A notification is shown to the screen to the user if he/she won, lost, or game ended in a draw.
- The application returns to Main Menu screen.

4.2.Use case diagram

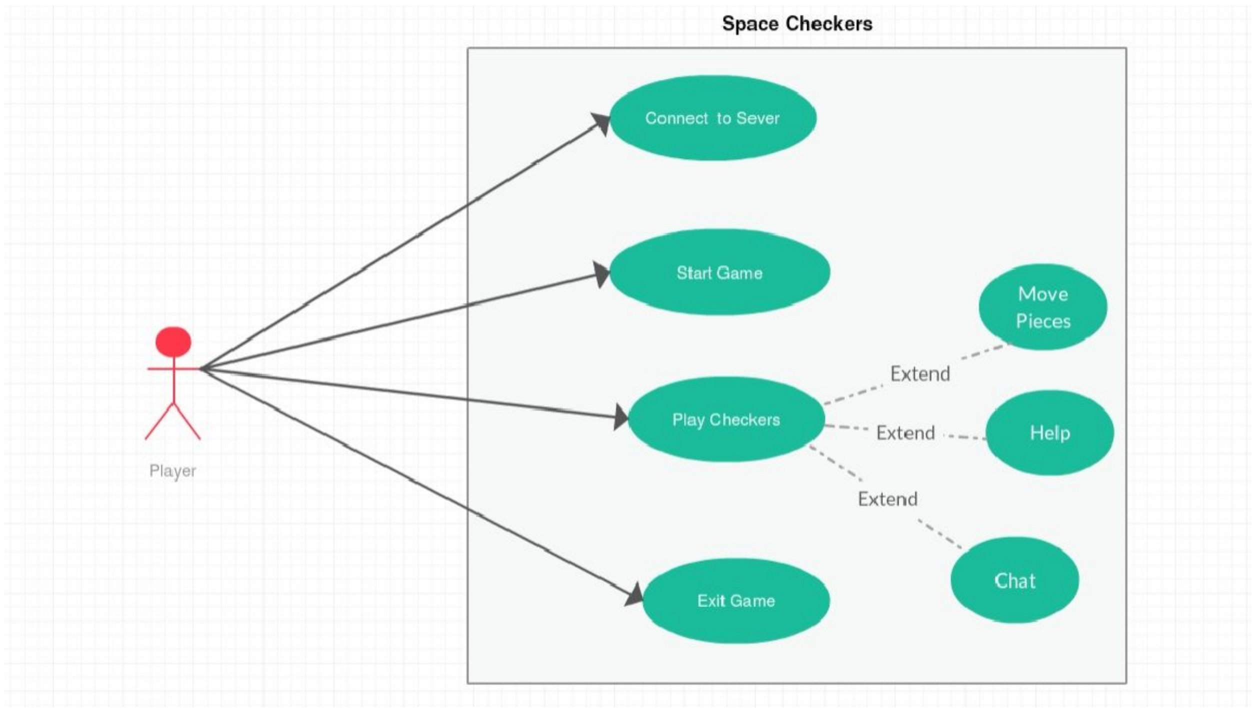


Fig.4.2 Use case diagram for space checkers

4.3 Activity Diagram

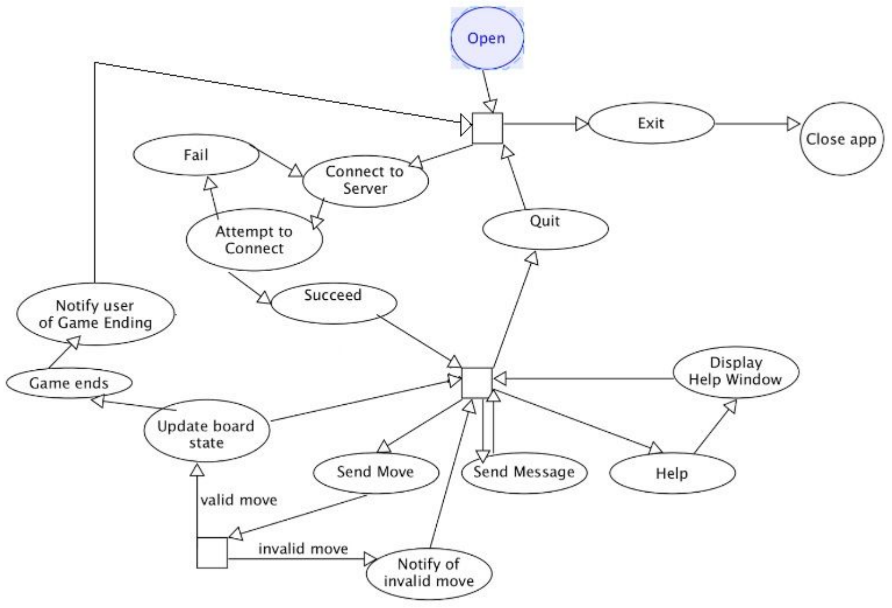


Figure 4.4 Activity Diagram for space checkers

5. System evolution

At this time, “Space Checkers” is being implemented as a basic operating system application allowing players to play checkers through server and client communication. In the future, animation and audio effect may be implemented such as a hand moving the checkers and an audio source being played while the checker is being moved. The expansion of these is to get as physical as possible to playing checkers in real life.