**Software Design Document**

**Space Checkers**

Alan Tsai

Klaus Nuredini

James Swanick

Deepak Yadav

Feb 16, 2017

**SRS Revisions  
Date Description Revision Editor**

2/16/2017 Created the document 0 Alan Tsai  
2/16/2017 Added Contents & Introduction 1 Alan Tsai  
2/17/2017 Design Overview 2 Klaus Nuredini  
2/17/2017 Added Architecture Diagram 3 Deepak Yadav  
2/18/2017 Added System Operation 4 Klaus & Alan  
2/19/2017 Added Server Manager UML 5 James Swanick

2/20/2017 Added Client Manager UML 6 Deepak & James

**Contents**

1 Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, Abbreviations

2 Design Overview

2.1 Description of Problem

2.2 Technologies Used

2.3 System Architecture

2.4 System Operation

3 Requirements Traceability

4 Server Manager

5 Client Manager

6 References

**1 Introduction**

**1.1 Purpose**

The purpose of this document is to describe the implementation of the Space Checkers software. The Space Checkers is designed to simulate the physical Checkers game with multiplayer server and client communication.

**1.2 Scope**

This document describes the implementation details of Space Checkers Application. The software will consist of two major functions. First is the server side that receives the information from the client side regarding the user’s moves and updates the checkers board accordingly, and the second is the client side which checks for move validation and for the end of the game and sends that information to the server.

**1.3 Definitions, Acronyms, Abbreviations**

**Checkers** Also called, *British,* draughts. ( *used with a singular verb*), a game played by two persons, each with 12 playing pieces, on a [checkerboard](http://www.dictionary.com/browse/checkerboard). [1]

**Client-Server** In[Computer science](https://simple.wikipedia.org/wiki/Computer_science) client-server is a [software architecture](https://simple.wikipedia.org/wiki/Computer_architecture) model consisting of two parts, [client](https://simple.wikipedia.org/wiki/Client) systems and [server](https://simple.wikipedia.org/wiki/Server_(computing)) systems, both communicating over a [computer network](https://simple.wikipedia.org/wiki/Computer_network) or on the same [computer](https://simple.wikipedia.org/wiki/Computer). A client-server application is a [distributed system](https://simple.wikipedia.org/wiki/Distributed_computing) made up of both client and server software. Client server application provide a better way to share the workload.The client [process](https://simple.wikipedia.org/wiki/Process_(computing)) always initiates a connection to the server, while the server [process](https://simple.wikipedia.org/wiki/Process_(computing)) always waits for requests from any client. [2]

**2 Design Overview**

**2.1 Description of Problem**

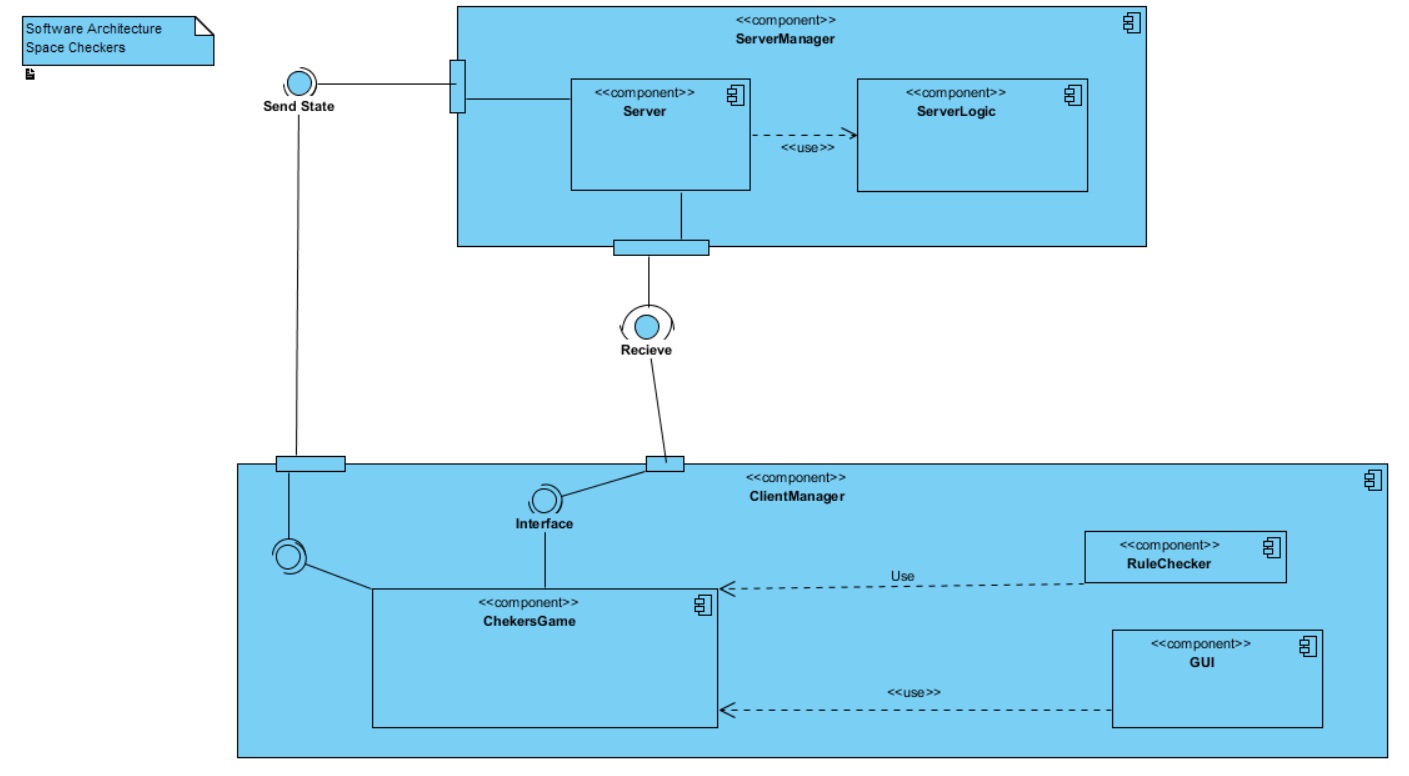
Checkers is a game that is played by two people and is usually played in person using a physical checkers board. Space Checkers will allow a person to be able to play against another player remotely using a computer, disregarding each of the players’ locations, provided that they have an internet connection. This will remove the possibility that any of the players potentially or mistakenly makes an illegal move without the other person noticing. Space Checkers will also update the board for both players in real time after each valid move, making the experience as close to the physical playing experience as possible.

**2.2 Technologies Used**

The Space Checkers Software will run on the user’s local machine and it will communicate with a server, which will be set up and running on one of the developers’ machines (pc running Mac OS). The target platforms will be Microsoft Windows and Mac OS. The development environment is Eclipse and we will program our application in JAVA. For our network we are using client-server model. We will be exchanging moves using TCP protocol and Sockets to exchange information between both clients and server.

**2.3 System Architecture**

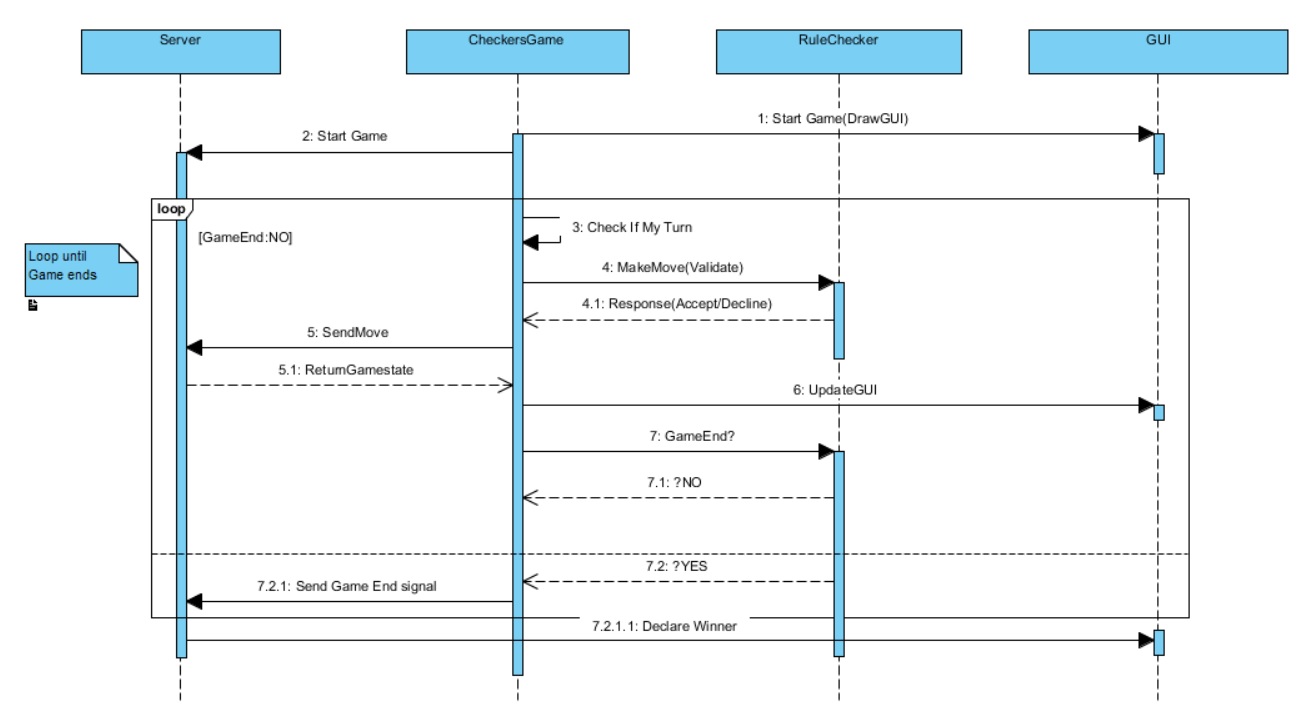
Figure 1 shows the high-level system architecture. The system will be constructed by a few distinct components:



**Figure 1: Space Checkers Architecture**

**2.4 System Operation**

Figure 2 is the typical sequence of events that occur when a user plays Space Checkers Game.

 **Figure 2: Sequence diagram**

**3 Requirements Traceability**

**Requirement Description Design Reference**

[r1.1.\*] Main Menus , Fig. 4  
[r1.2.\*] Game Screen/Gameplay, Fig3,Fig. 4

[r1.5\*] End Game Screen, Fig. 4

[r3.2.1] Extensibility, Fig. 3

[r3.2.2] Maintainability, Fig. 4

**4. Server Manager**

Figure 3 depicts the UML model for the Server Manager Component and its subcomponent. It also define the relation with other classes and the interface used to access server functionalities.

It will handle the connection from both players as well as update the current game state to both clients.

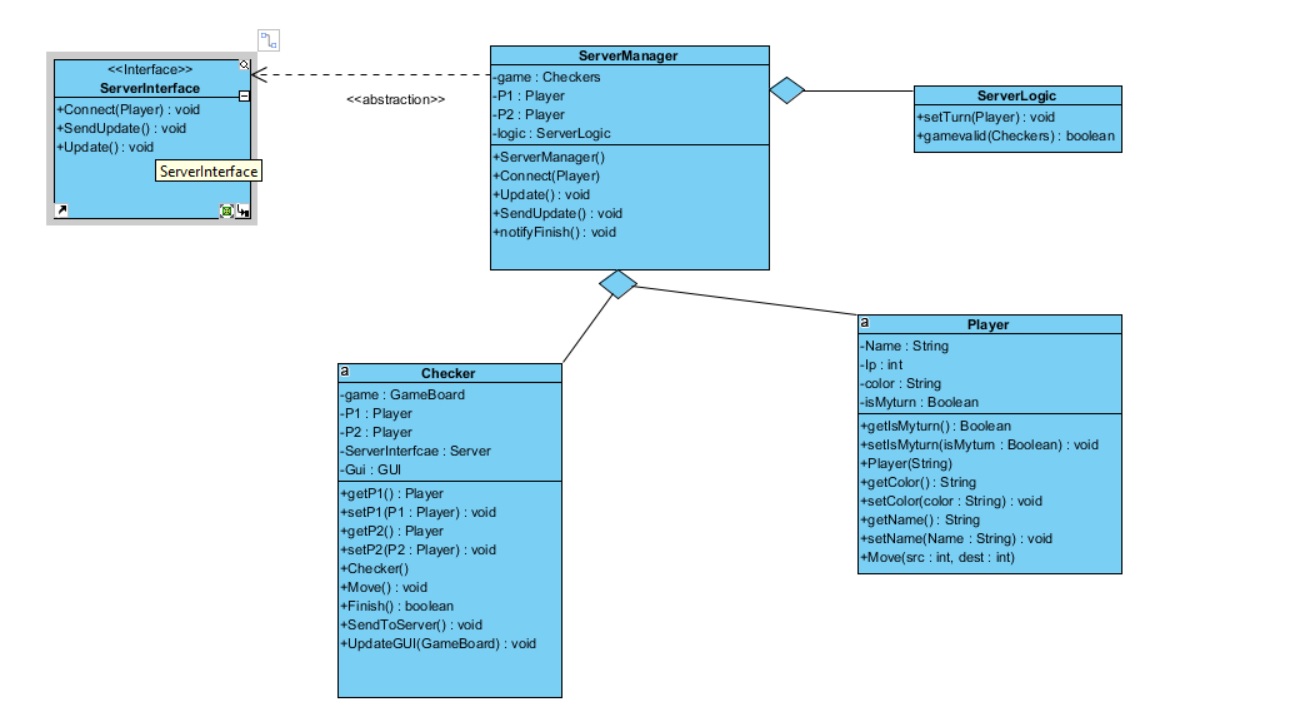
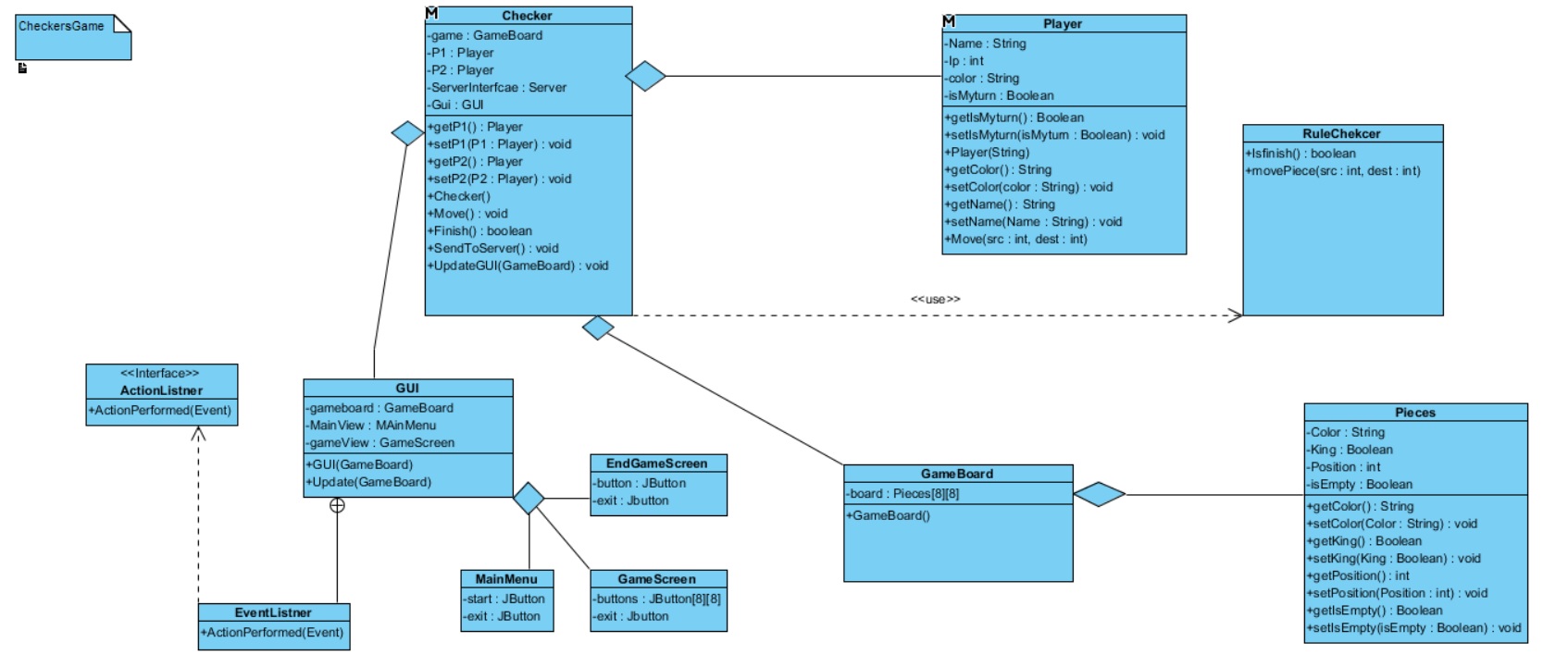


Figure 3: Server Manager component UML diagram

**5. Client Manager**

Figure 4 depicts the UML model for the Client Manager Component and its subcomponent. It also define the relation with other classes and the interface used to access server functionalities.It inlcudes the GUI sub component and the RuleChecker sub component.

This component will handle most of the applications functions(Thick client system) , It will validate move and start the GUI according to board state and send request to server.



**Figure 4. Client Manager Component**

**6 References**

[1] <http://www.dictionary.com/browse/checkers>

[2] https://simple.wikipedia.org/wiki/Client-server