

Software Requirements Specification

War Room Systems

Vito Salerno

Jeff Segall

Ian Yoder

Josh Zenker

March 4, 2009

Revision 1.0

Revision History

Date	Description	Revision
3 Mar 2009	Initial Version	1.0

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	1
1.3	Definitions/Abbreviations	1
2	Architecture Design	2
3	Static Design Structure Modeling	3
3.1	Class Descriptions	3
3.1.1	WarRoomChess.java	3
3.1.2	NetInterface.java	3
3.1.3	NetClient.java	3
3.1.4	NetServer.java	3
3.1.5	SaveGame.java	3
3.1.6	GUI.java	3
3.1.7	ChessBoard.java	3
3.1.8	ChessSquare.java	3
3.1.9	Piece.java	3
3.1.10	Bishop.java	3
3.1.11	King.java	3
3.1.12	Knight.java	3
3.1.13	Pawn.java	4
3.1.14	Queen.java	4
3.1.15	Rook.java	4
3.2	UML Diagram	4
4	Dynamic Behavior Modeling	5
4.1	Game Flow	5
4.2	Start New Game	6
4.3	Save Game	7
4.4	Load Game	8
4.5	Undo Move	9
4.6	Quit Game	10
5	Requirements Traceability Table	11
5.1	Traceability by Requirement Number	11
5.2	Traceability by Design Component	12

1 Introduction

1.1 Purpose

This document will show the UML specifications of War Room Chess. Reading this document will give the user and developer a better understanding of the innerworkings of the software. Readers will gain insight on both the static and dynamic processes of the game software.

1.2 Scope

This document is strictly a UML specification diagram. For other information, consult the War Room Chess software requirements specification. This document is for specification purposes and reflects the software as closely as possible. However, it is subject to change.

1.3 Definitions/Abbreviations

2 Architecture Design

3 Static Design Structure Modeling

3.1 Class Descriptions

3.1.1 WarRoomChess.java

The main class for the application. Handles the main game loop as well as interfacing with the auxiliary classes for the GUI, networking, logic, and game saving.

3.1.2 NetInterface.java

An interface for the networking of games. Provides a class for the application to instantiate to act as either a client or a server for network play.

3.1.3 NetClient.java

Functionality for acting as a client during gameplay.

3.1.4 NetServer.java

Functionality for acting as a server during gameplay.

3.1.5 SaveGame.java

Functionality for saving and loading games. Takes in the current game state and outputs it to a file or does the process in reverse to load.

3.1.6 GUI.java

Handles the drawing of the GUI and user input.

3.1.7 ChessBoard.java

A representation of the game board. Used for keeping track of the game state.

3.1.8 ChessSquare.java

A representation of a single square on a chess board. Used for keeping track of the game state.

3.1.9 Piece.java

An interface for game pieces. All pieces implement this interface. Provides methods for moving and capturing pieces used by the main game loop.

3.1.10 Bishop.java

A representation of a bishop.

3.1.11 King.java

A representation of a king.

3.1.12 Knight.java

A representation of a knight.

3.1.13 Pawn.java

A representation of a pawn.

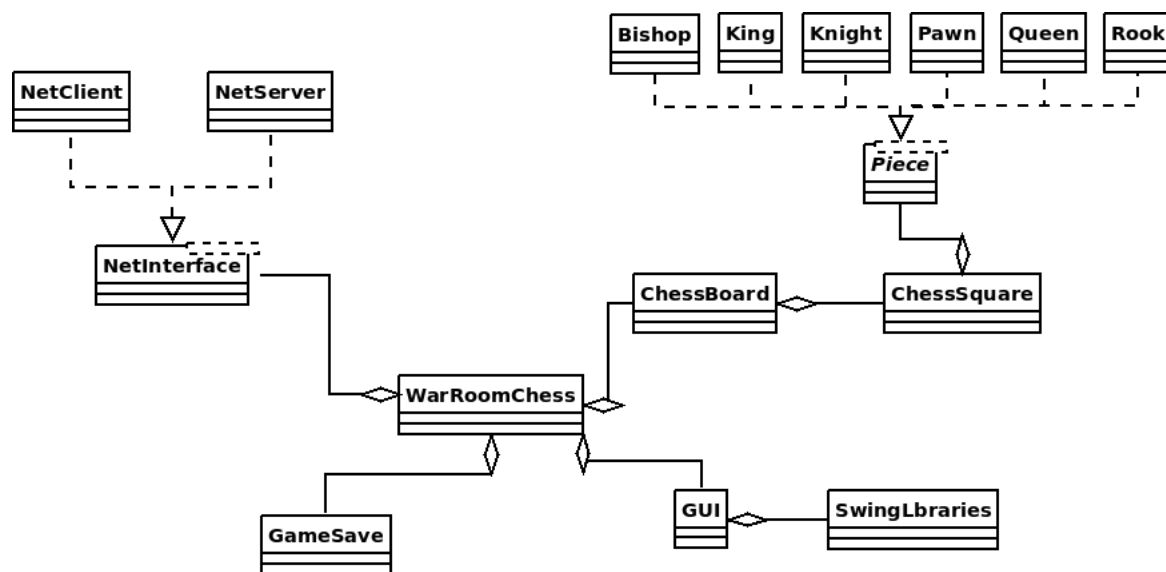
3.1.14 Queen.java

A representation of a queen.

3.1.15 Rook.java

A representation of a rook.

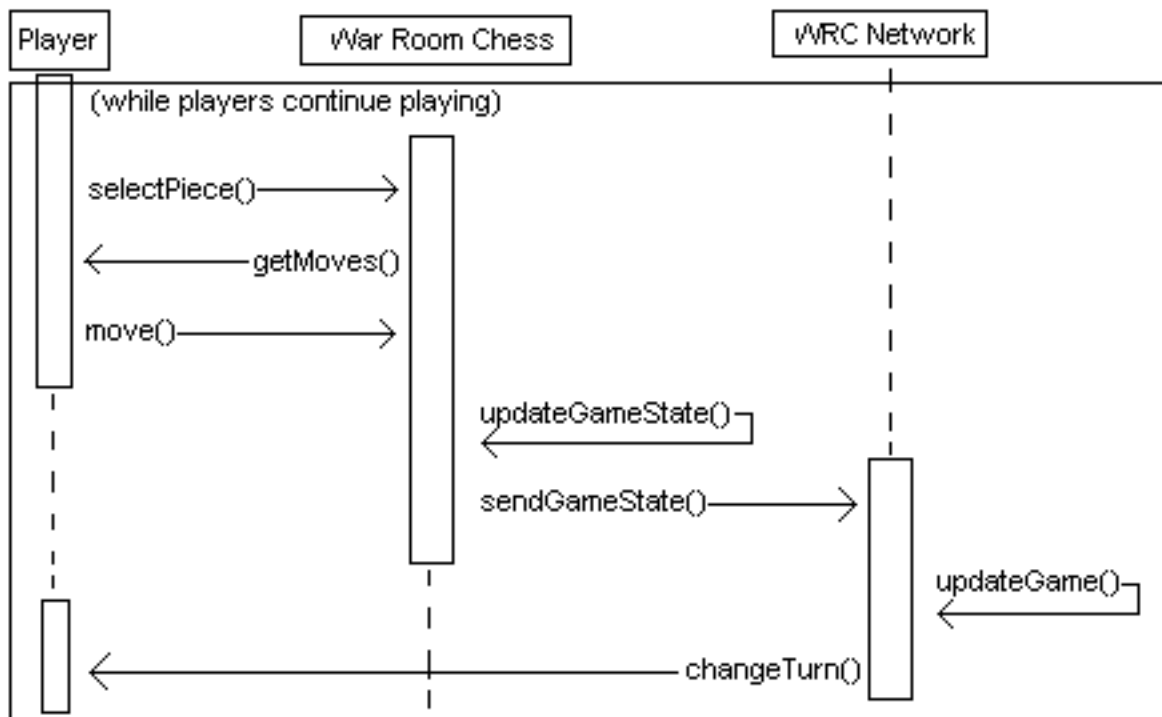
3.2 UML Diagram



4 Dynamic Behavior Modeling

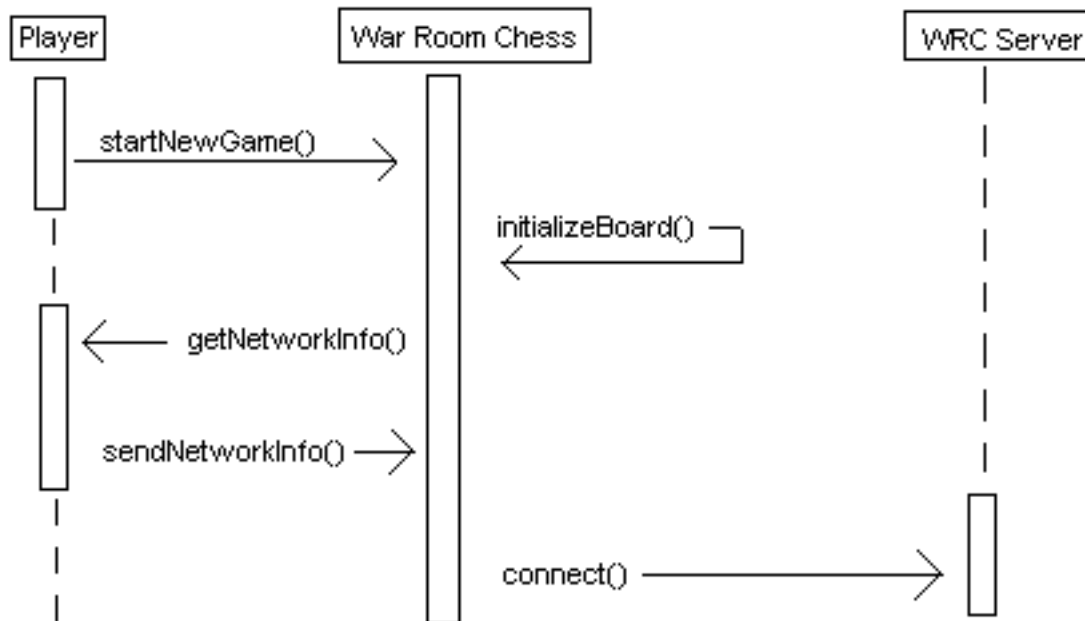
4.1 Game Flow

Game Sequence Diagram



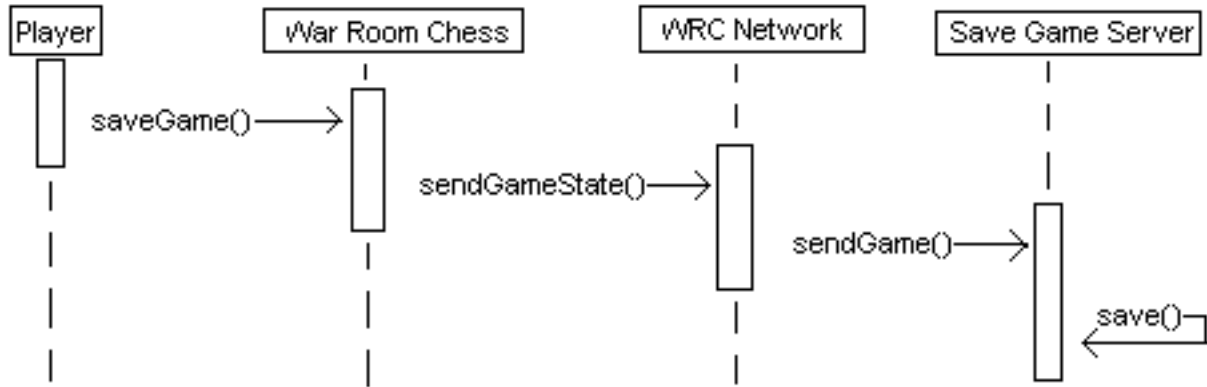
4.2 Start New Game

Start New Game Sequence Diagram



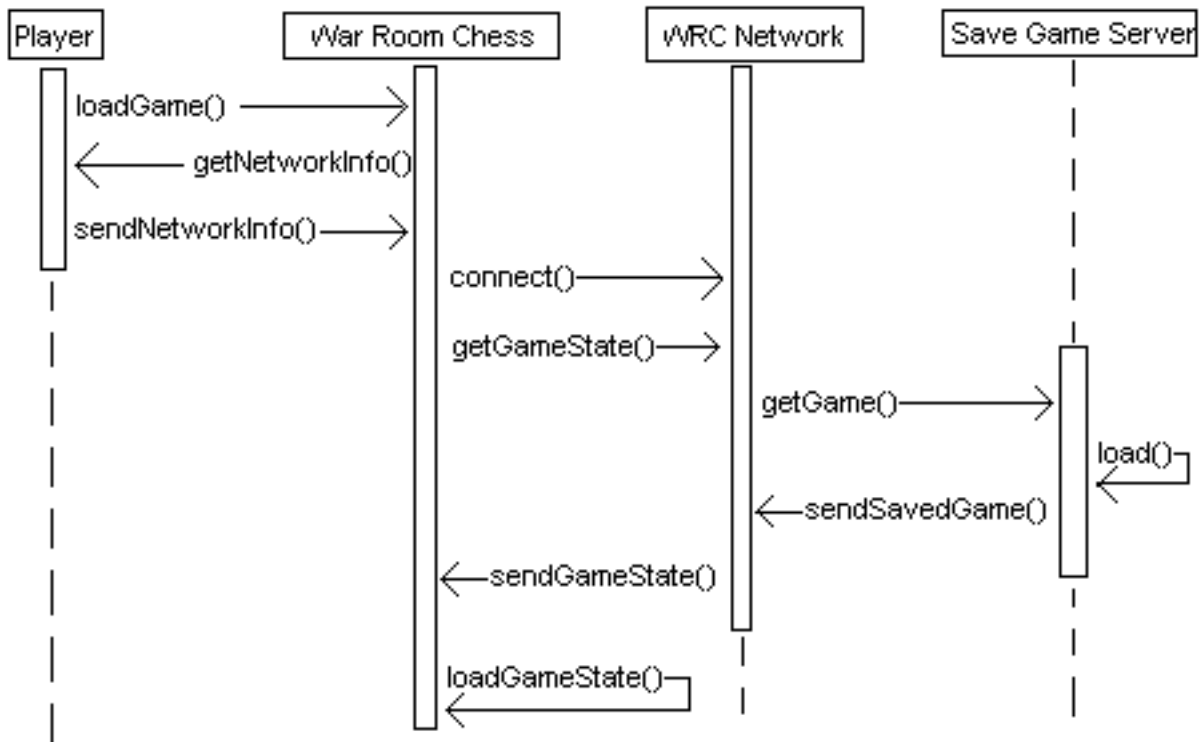
4.3 Save Game

Save Game Sequence Diagram



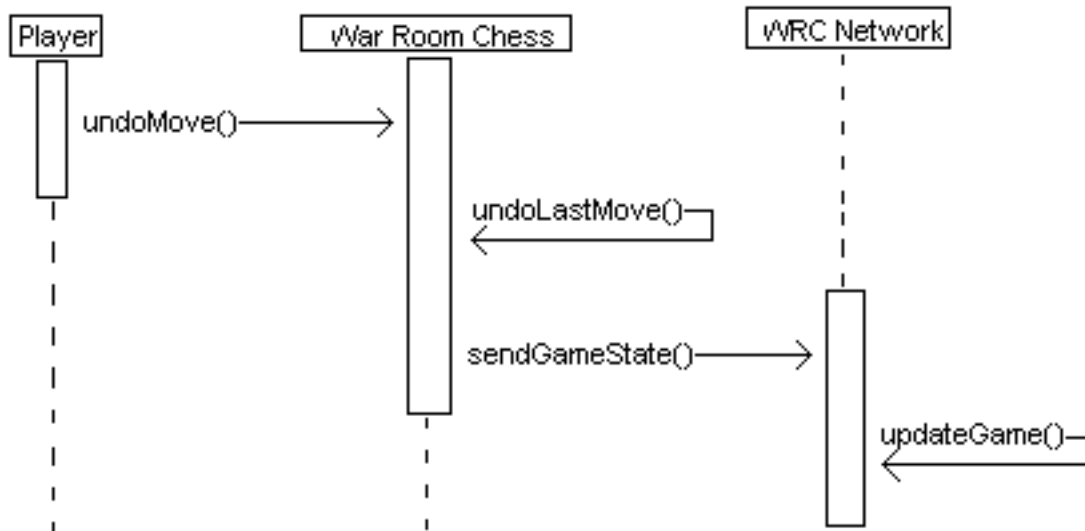
4.4 Load Game

Load Game Sequence Diagram



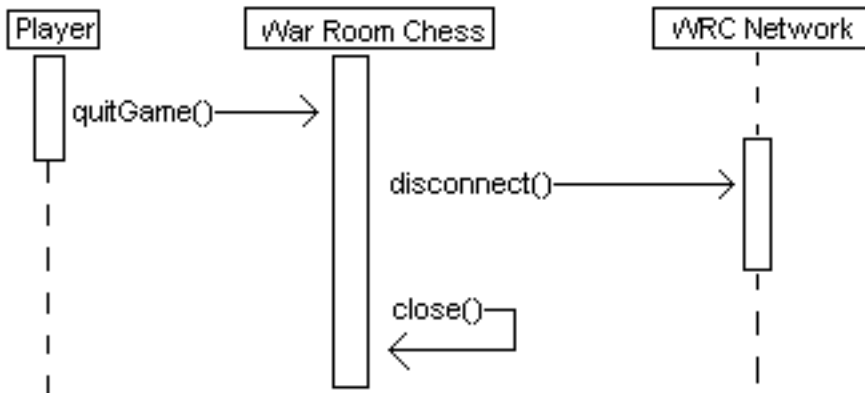
4.5 Undo Move

Undo Move Sequence Diagram



4.6 Quit Game

Quit Game Sequence Diagram



5 Requirements Traceability Table

5.1 Traceability by Requirement Number

Req #	Design Component
0100	3.1.7, 3.1.13, 4.1
0110	3.1.7, 3.1.15, 4.1
0120	3.1.7, 3.1.12, 4.1
0130	3.1.7, 3.1.10, 4.1
0140	3.1.7, 3.1.14, 4.1
0150	3.1.7, 3.1.11, 4.1
0160	3.1.7, 3.1.11, 3.1.15, 4.1
0200	3.1.7, 3.1.10, 3.1.11, 3.1.12, 3.1.14, 3.1.15, 4.1
0210	3.1.7, 3.1.13, 4.1
0220	3.1.7, 3.1.13, 4.1
0300	3.1.7, 3.1.13, 4.1
0400	3.1.7, 3.1.10, 3.1.11, 3.1.12, 3.1.13, 3.1.14, 3.1.15, 4.1
0500	3.1.5, 4.3
0510	3.1.5, 4.4
0520	3.1.5, 4.3, 4.4
0530	3.1.5, 4.3, 4.4
0535	3.1.5, 4.3, 4.4
0540	3.1.5, 4.3, 4.4
0545	3.1.5, 4.3, 4.4
0600	3.1.2, 3.1.3, 3.1.4, 4.1, 4.2
0610	3.1.2, 3.1.3, 3.1.4, 4.1, 4.2
0620	3.1.2, 3.1.3, 3.1.4, 4.1, 4.2
0630	3.1.2, 3.1.3, 3.1.4, 4.1, 4.2
0700	3.1.2, 3.1.3, 3.1.4, 4.1, 4.2
1000	NFR
1010	NFR
2000	3.1.5, 4.4
2100	3.1.2, 3.1.3, 3.1.4, 4.1, 4.2

5.2 Traceability by Design Component

Design Component	Req #
3.1.1	
3.1.2	0600, 0610, 0620, 0630, 0700
3.1.3	0600, 0610, 0620, 0630, 0700
3.1.4	0600, 0610, 0620, 0630, 0700
3.1.5	0500, 0510, 0520, 0530, 0535, 0540, 0545
3.1.6	
3.1.7	0100, 0110, 0120, 0130, 0140, 0150, 0160, 0200, 0210, 0220, 300, 400
3.1.8	
3.1.9	
3.1.10	0130, 0200, 0400
3.1.11	0150, 0160, 0200, 0400
3.1.12	0120, 0200, 0400
3.1.13	0100, 0210, 0220, 0300, 0400
3.1.14	0140, 0200, 0400
3.1.15	0110, 0200, 0400
4.1	0100, 0110, 0120, 0130, 0140, 0150, 0160, 0200, 0210, 0220, 0300, 0400, 0600, 0610, 0620, 0630, 0700, 2100
4.2	0600, 0610, 0620, 0630, 0700
4.3	0500, 0520, 0530, 0535, 0540, 0545
4.4	0510, 0520, 0530, 0535, 0540, 0545
4.5	
4.6	