TCP2201 Barsoomian Chess Project
Trimester 2, 2016/2017
by ABM

	Name	ID	Phone NO	Email
Group leader	Ahmad Hamdan	1131121467	011-39007471	Ah.w.hamdan@gmail.com
Member 1	Malek Mohammed Abdulqawi Hasan	1132702934	011-37041921	alafifym@gmail.com
Member 2	Daljinder Kaur	1122702693	012-5251315	djawesome93@gmail.com
Member 3	Seham Adam	1132700157	012-6237719	sehamadam93@gmail.com
Member 4	Barat Rashed Mahade	1141328099	01117551599	rashedbarat@gmail.com

February 6, 2017

Contents

Table of Contents **1**

List of Figures **1**

1 Barsoomian Chess **2**
1.1 Introduction 2
1.2 Usecase Diagram 3
1.3 Class Diagram 3
1.4 Sequence Diagram 6

List of Figures

1.1 main Game Board 2
1.2 Usecase Diagram 3
1.3 Class Diagram 4
1.4 Board Class 4
1.5 ChessPlayer Class 5
1.6 Helper Class 5
1.7 Pieces Class 6
1.8 Start game sequence 6
1.9 Save game sequence 7
1.10 Load game sequence 7
1.11 Play game sequence 8
1.12 Get help sequence 8
1.13 Swap turn sequence 9

1 Barsoomian Chess

1.1 Introduction

the Barsoomian Chess is like any chess game and it has the 4 main stones like : Heart, Arrow, Star and cross. the difference for this chess for other chesses is after 4 turns of movement the star will convert to cross and the cross will convert to start, the game will end when the red player kill the heart for the blue player and (vice-versa).
the figure below shows our main game Board that we code it in java swing (played by 2 humans players red and blue) and how you can compile and run it using command line:

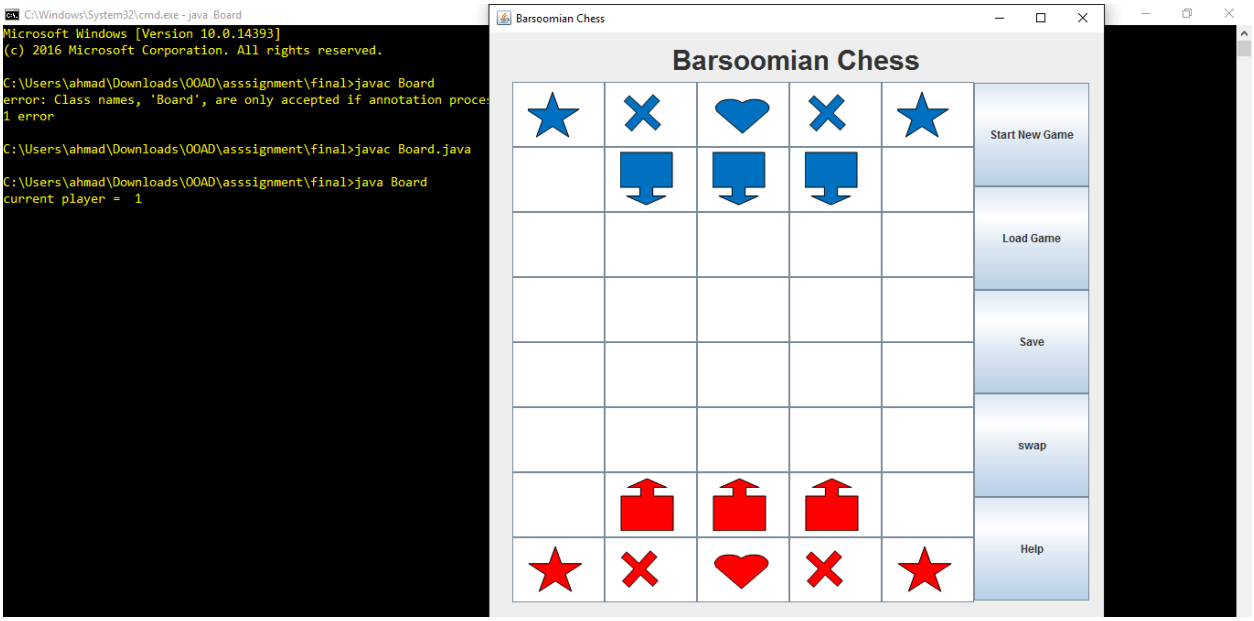

Figure 1.1: main Game Board

documentation on how to use your program.

1.2 Usecase Diagram

the usecase diagram shows the main functions for the player in the game :

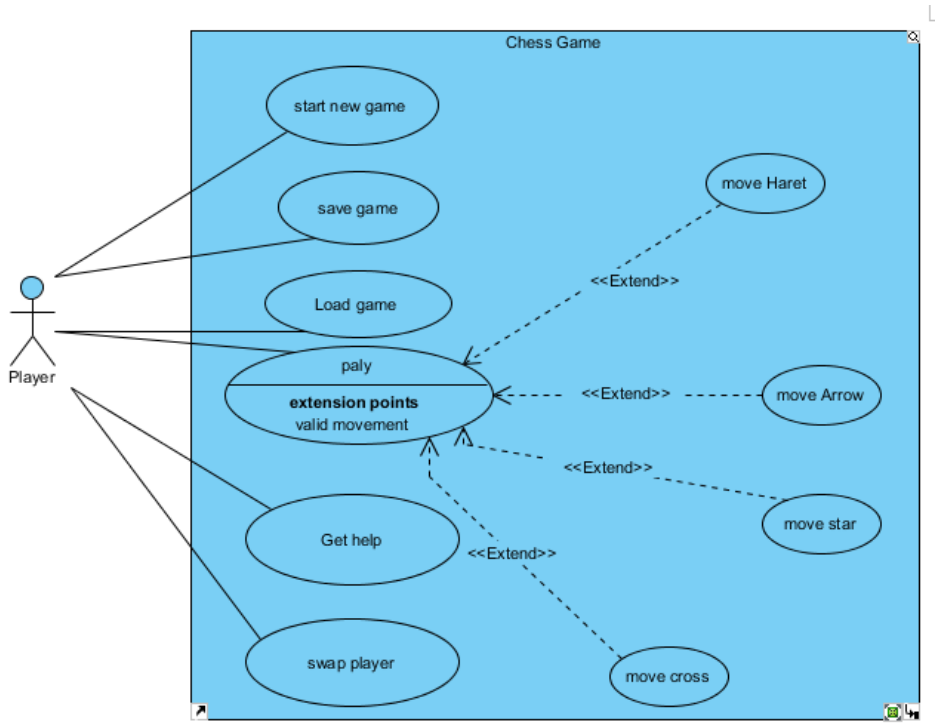

Figure 1.2: Usecase Diagram

The UML Use case diagram includes player actor ,user can save, load, start the game. In addition, the user can get assistance regarding the steps involved to play the game and swap the turn.

1.3 Class Diagram

This is the class diagram, the program consists of 8 classes and they are the Board, ChessPlayer, Helper, Pieces, Arrow, Heart, Star and Cross. The design pattern we used is the composite pattern, the composite pattern describes that a group of objects is to be treated in the same way as a single instance of an object. As shown in the board class all the stones we have are inistansized as a Pieces so that all of them will be treatted in the same way.

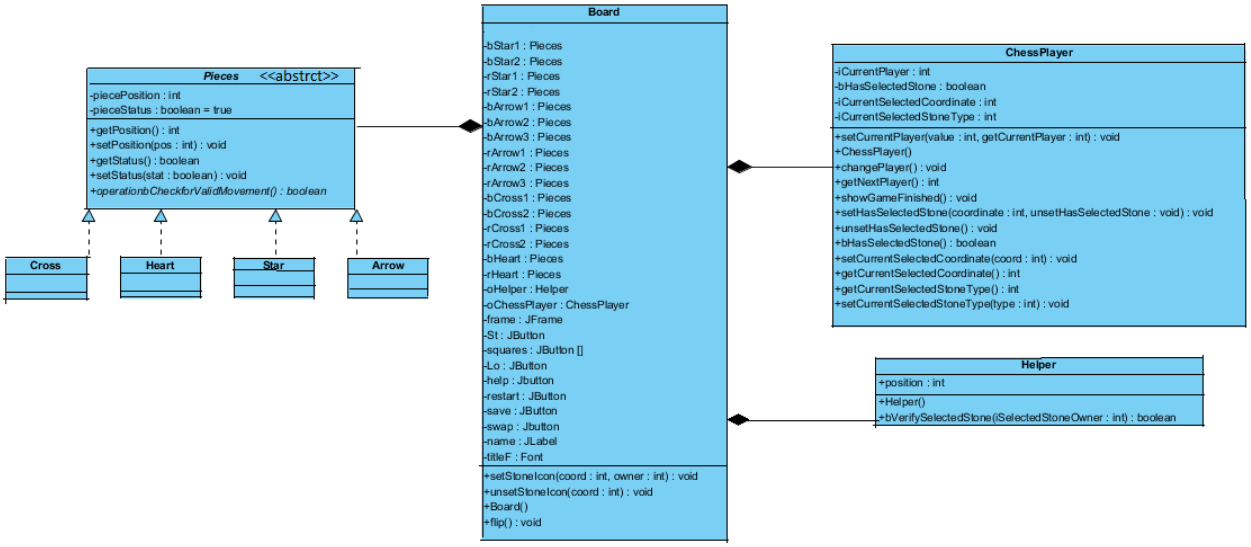

Figure 1.3: Class Diagram

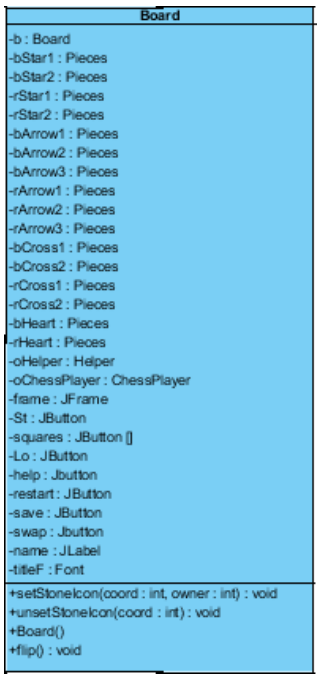

Figure 1.4: Board Class

The Board class is the main class where all the user interactions and the mouseclick events will happen the Board class composites a ChessPlayer so that the players can play and have turns and it also composites all the pieces to check the movements and their positions in the board. And it also consists of other functions and buttons such as the save button to save a game, load button to load an exsisting game, swap button to flip the screen, help button to see the game instructions and a start new game button to restart and reset the board.

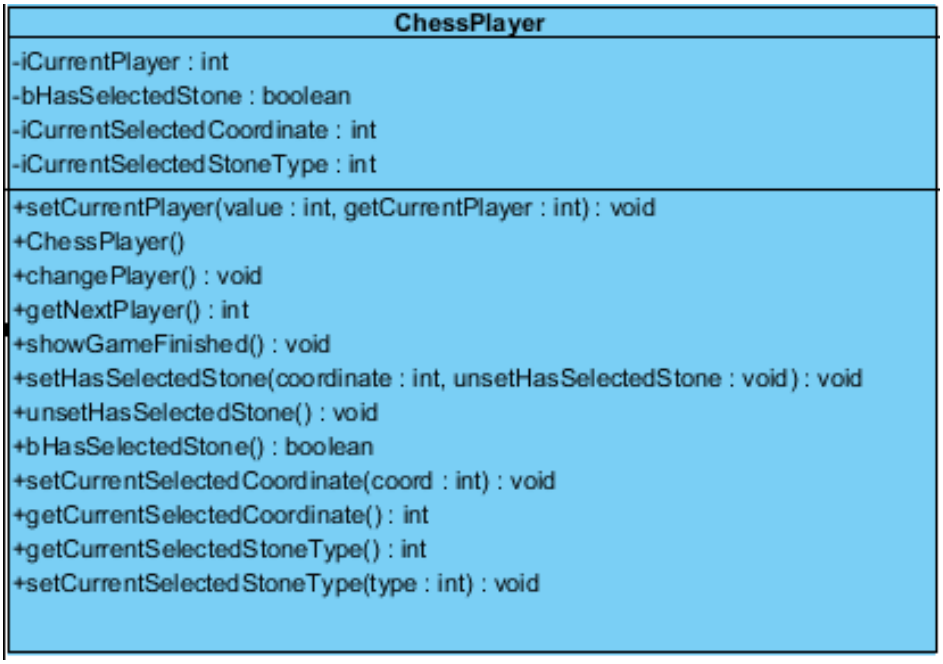

Figure 1.5: ChessPlayer Class

The ChessPlayer class has different function to change players turns and see if the player has selected a coordinate, and if that coordinate he selected belongs to him and to check the type of the stone selected to know the right movement

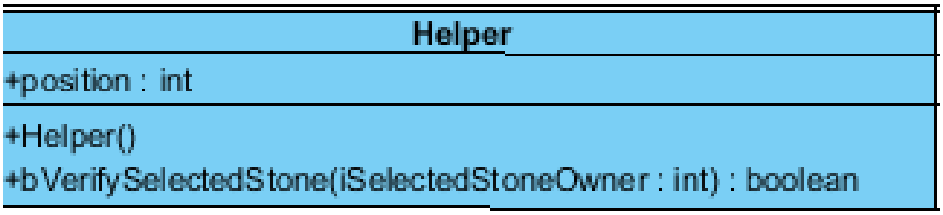

Figure 1.6: Helper Class

The Helper class helps the chess player class to know if the selected type of the stone belongs to the current player or not.

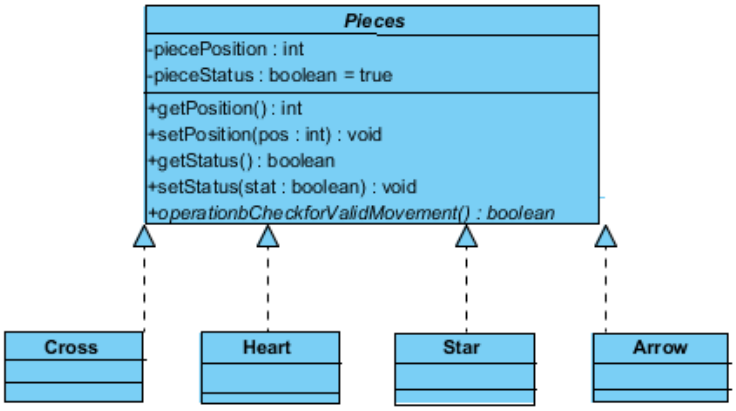

Figure 1.7: Pieces Class

The Pieces class is an abstract class it has an attribute called piecePosition to know the current position of the piece and a pieceStatus attribute to know weather this piece still exists or not and both the attributes com with a getter and setter functions, and it also consists of an abstract function to check weather the move made is valid or not.
The Arrow, Star, Heart, Cross classes extends the pieces class each class will over ride the abstract function in the pieces class with its own algorithm and with every move the position of that piece is changed to the new coordinate that it landed on.

1.4 Sequence Diagram

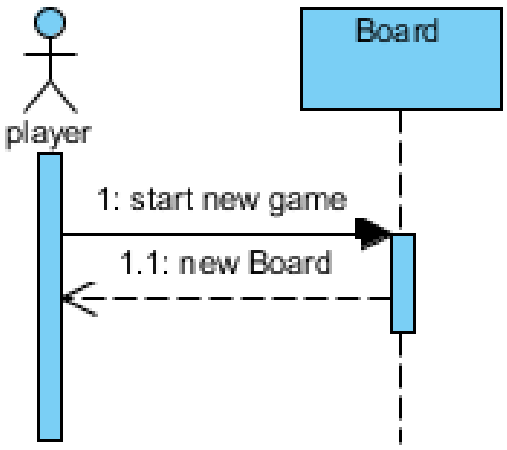

Figure 1.8: Start game sequence

The player clicks on the start button on the right side of the board and a new board will be opened.

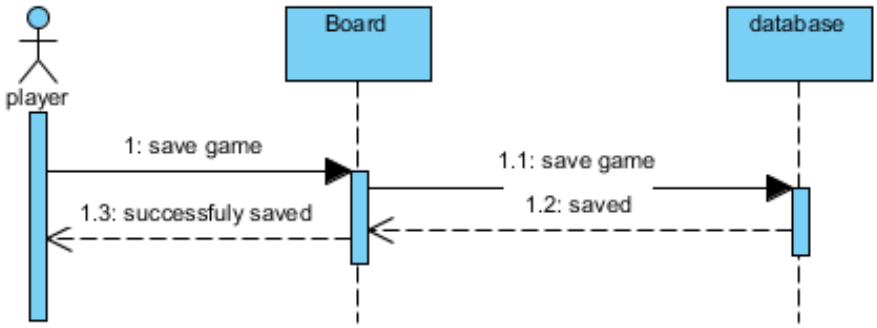

Figure 1.9: Save game sequence

If the player wants to save the game he can click on the save button and the position and the status of each stone will be saved in a txt file.

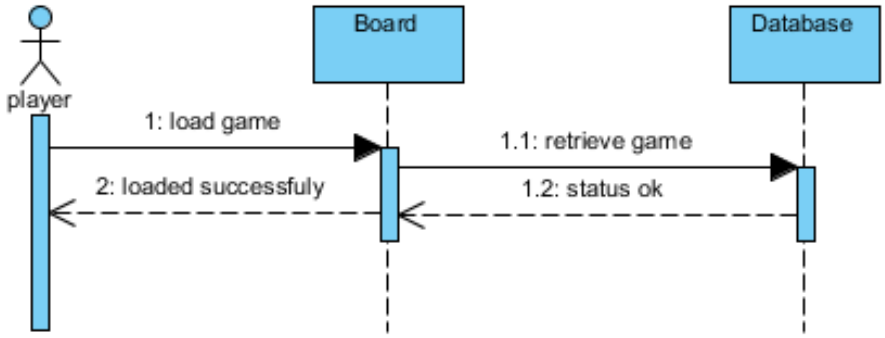

Figure 1.10: Load game sequence

if the player clicked on the load button the last saved game will be loaded and the game can be continued.

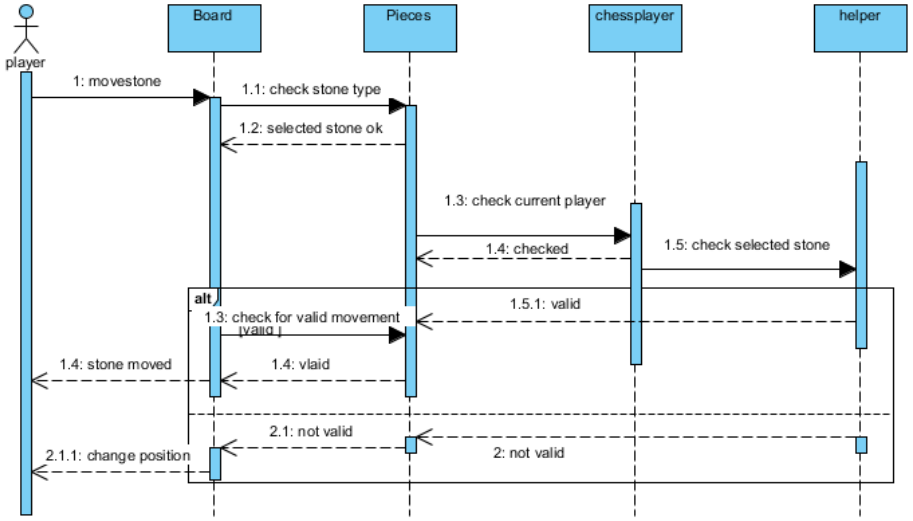

Figure 1.11: Play game sequence

The user will be interacting with the board. The player can do all the functions that are displayed in the board. The movements and will happen here. An array of buttons will be displayed to the user with the pieces placed in their original positions as shown in Figure 1.1. Two players can play the game, one of the players will select a stone the board will check if the current player can select that stone if not the user has to choose a valid stone after selecting a valid stone he can choose a position to move to and the position he chooses will be checked if it is a valid movement for that selected stone or not if its not he have to choose another position to move to.

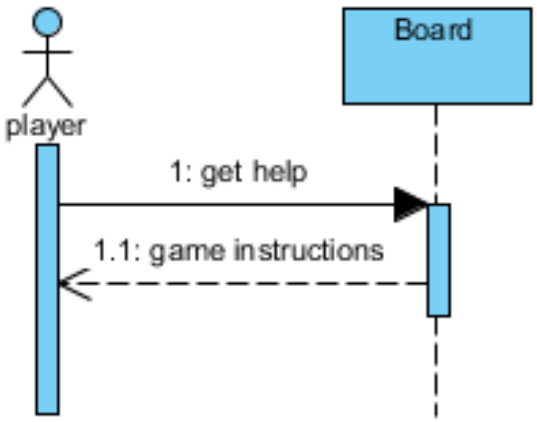

Figure 1.12: Get help sequence

If the user clicked on the Help button a message box will appear with the game instructions. if the user

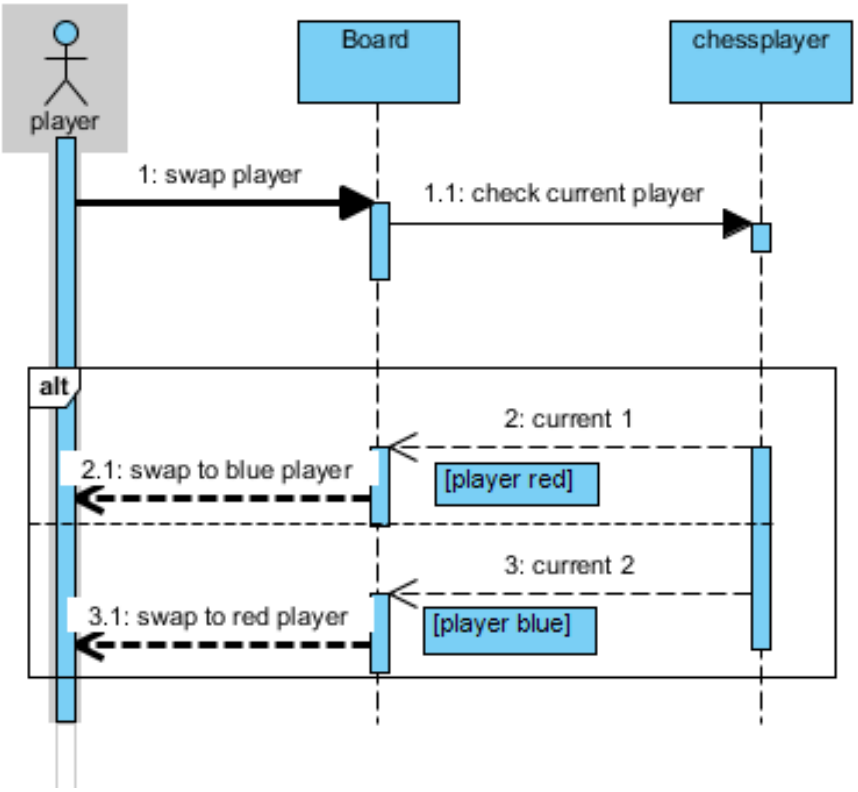

Figure 1.13: Swap turn sequence

clicked on swap the board will be flipped up side down , so the other player can play in his turn.