Final Project: Command Line Chess

John Miller JEMiller@csustudent.net
Alex Irish ADIrish@csustudent.net
Ashley Ward ANWard1@csustudent.net
Milleina Ohara MOhara@csustudent.net

Git location: https://github.com/millerjohneric/csci-325-spring-2017_FinalProject

Abstract

With the advent of technology and graphics, it is easy to lose sight of the joy of simplicity. A simple command line chess game can provide thought provoking entertainment while cultivating creativity and sparking an interest in strategy. This project implements the game of chess between two players. Players will enter their names and will randomly be assigned a color. The player who is on the white team will go first. The basic rules of play will be implemented into the game. A player will select the location on the board to move from and then set the destination. A player can only move his or her own pieces. Each chess piece will be restricted by the standard chess moves.

1. **Program Operation**

   The class Main_PlayChess will call classes and sub-classes as needed to control the actual playing of the game. It will start by initializing a chess board object and implementing it's set board function to place each chess piece in position. The users will be prompted to enter their names and will be randomly assigned a team color. The game will then commence. The player on the white team will be prompted to enter a location for the piece that they wish to move. Locations are given as an alpha column and a row number. After selecting the piece to move, another prompt will ask for the destination. The desired move will be validated inside the chesspiece class using the standard rules of play as the requirements. If the move is a valid move the selected chess piece will move to the new location. If an enemy player was previously in that location, then the enemy piece will be removed from the board. The players will take turns and gameplay will continue until reaching a stale mate, where no more plays are valid, or a check mate when one of the kings is in check and cannot move out of it.

2. User Interface

The interface will be console based with asci characters displayed on the console to represent the board and the players. After each move the board will be refreshed onto the console. The initial display is shown below in figure 1.

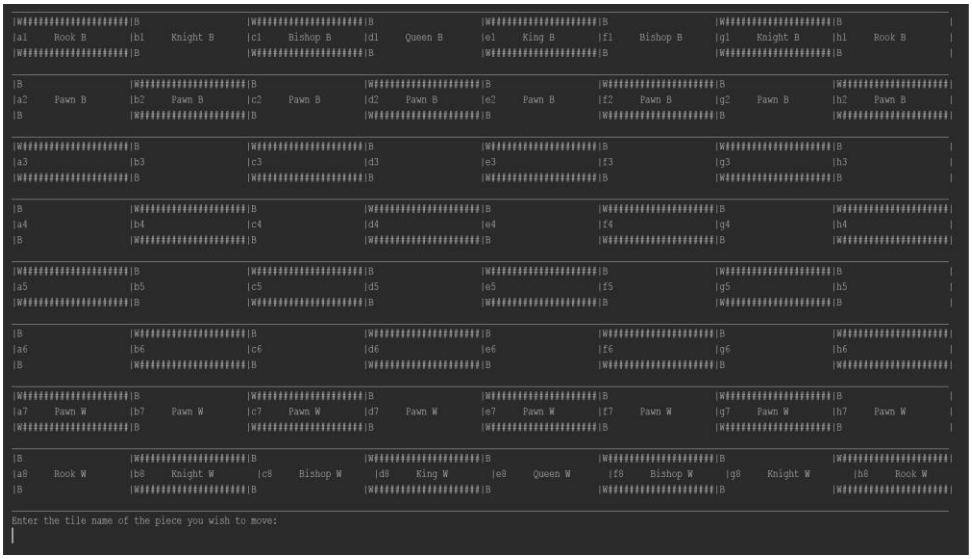

Figure 1

3. Division of Labor

John Miller: Initial design of classes, structure.
Alex Irish: Documentation and implementation of features.
Ashley Ward: Graphics, game sequencing.
Milleina Ohara: Debugging, implementation of features.

4. UML

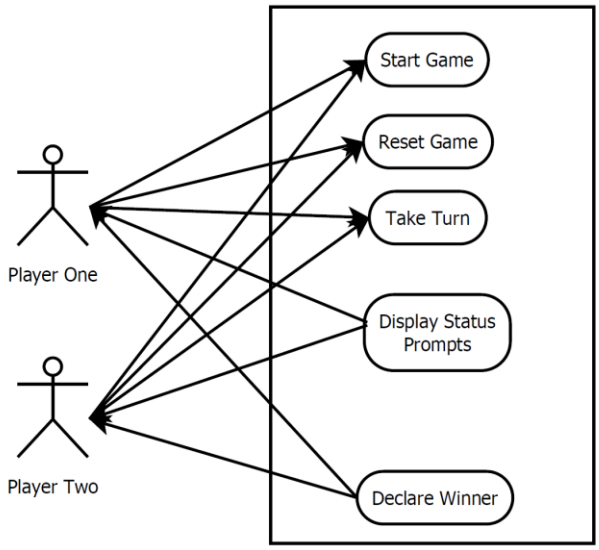

Figure 2