James Sabella, Louis Pisicolo, Thomas Herold

Prof. Louis Iacona

CSC 330

5 May 2016

Chess Game

The project our group challenged ourselves with is a take on the classic game of Chess. The game itself we divided into four packages including the board, the pieces, the players, and the GUI. Each of these packages are designed to construct a board with its pieces, include proper piece movement and standard rulings, allow for player interaction and status checks, and effectively designing a proper board along with all its superfluous submenus.

The board package is what checks the board along with its content. The board class is essentially the ‘builder’ of the project. It creates an instance of a board including its tiles and effects the board itself or where each piece is. BoardUtils effects the board more directly taking care of rows and columns, as well as the coordinates of each tile on the x and y planes. The move class is designed to take care of how a piece can move. The type of move, whether it be regular or special, and where it is moved to is designated by the move class. Lastly the Tile class simply creates an instance of a tile, and responds with the tiles occupancy or specific properties.

The next package contains the pieces. Each individual piece is a subclass of the class piece. It is in the piece class that each piece receives its unique properties. It keeps track of the piece’s color, position, and the type of piece it is (determined by a specific value). However, each individual piece is in charge of its own offset (how the specific piece moves around the board). Each piece then checks the move class to see its particular piece exceptions to if it can or cannot make a certain move.

The player package involves player interaction and the rulings that effect the player directly. The player class checks for the status of each player. Are they in check, checkmate, or are there any legal moves remaining. It also checks for which player is currently making a move and each time a player makes a move it transitions into a whole new board of the same components with the updated move.

The project in its entirety contains about 2600 lines of code and 26 classes. Inside the scope of our MVP was to allow the player to start a new game when needed, see the pieces that have been captured and a log of the moves in separate gui’s and create preferences to let the player flip the board to their viewing side and see the possible moves of a piece highlighted on the board. Outside of this we wanted to create a basic AI system and possibly a separate checkers game that works off our existing code.

We used many imports from the java library including external libraries from google guava. The majority of the code is immutable, and contain collections to keep track of moves, lists to add possible scenarios and many others.

Consequently, our groups take on the classic game of chess involves having a class for each individual piece on the board. These pieces determine how they can move and then go through a piece class to receive specific qualities. Then the piece checks the move class for any exceptions before making its move. In the meantime, the player classes are checking for any legal moves, checks, or checkmates. And lastly, the board class is acting as a builder constantly keeping the board updated. Through this course of actions, our take on chess successfully interprets that of the standard game.