Brogrammers APCS1 pd 02
Kevin Feng, Mohtasim Howlader, Alan Wang 1/02/18

Final Project Proposal

Matrix Game Simulator

We are going to create a game simulator.

Chess

We are going to use a matrix to represent a chess board and label the sides of the board with letters and numbers. The pieces will be subclasses of the Object gamePiece that correspond with their specific type. The board will be populated with strings that represent each piece. There strings will be colored black or white to indicate possession. This game will be designed for 2 players. The player will then indicate where he wants a piece to move by typing the position he wants to move from to the position he wants to move to, which will only execute if the move is possible. Once the move is executed, the other player is able to take his turn and the game ends when the king is checkmated or when the game ends in a stalemate. We will incorporate the check and checkmate function by having a variable for each object that indicates it's possible moves, and then checking if each square on and around the king are located in this variable using a for loop.

Checkers

We are going to use a matrix to represent a checkerboard. Red team will have red pieces, while black team will have black pieces. This will be very similar to the chess game with similar rules, however we are going to restrict the moves differently and allow for jumps if possible. This game will also be designed as a two player game. Just like in chess, the player will select the piece he/she wants to move and to where he wants to move it. The game will conclude when either player has no more pieces or cannot move any of his pieces.

Card Matching

We will create two matrices, one used as a display for the user and one that contains the actual values pertaining to each coordinate. The matrices will be labeled with letters and numbers on the side. The user will be able to pick their difficulty, which will choose the dimensions of the card game. By picking a pair of coordinates, the game will compare the values behind the scene in the 2nd matrix and if are equal, will be displayed as matched on the display matrix. We will support suits and colors, which could be used for a harder difficulty (matching same color suits only).