Cover Page

Team Members:

Aashish Agrawal - aagraw10@uic.edu

Karan Ahuja - kahuja3@uic.edu

Documentation

Instructions:

- The Server picks a port to listen to for clients connecting, and hits start
- The client picks an IP address and port to try to connect to, and hits play
- The client will go to the game screen. The server will make the first move
- All server's moves will also be displayed in the dialog box below the game board
- The client can then pick any of the available boxes to make its move
- When the game is over, the results will be shown, and the client can either quit, or play again
- The top 3 scores will be displayed on the right
- The server will have a log of all moves played by the client, the server, and victories/draws

Changes made to incorporate the Min-Max Algorithm:

- We made a few small changes to AI_MinMax.java
    - We changed its constructor to pass a String Array as a parameter
        - This was done so that we could pass the current state of our game to the AI_MinMax class
    - We also added a function, that returns an Integer ArrayList of the best moves for the server to make given the current state of the board
- In our FindNextMove class, we use some of the changes made above:
    - We create a function, called getBestMove, which takes the current state of the board as a parameter
        - This function creates a brand new instance of AI_MinMax, using our altered constructor for AI_MinMax
        - Using the function we added to AI_MinMax, we get an Integer ArrayList of the best moves for the server
        - Since all of the moves are best ones, we simply pick the first move in the list, clear the list, and return that number
        - This function is synchronized. The server will only instantiate one instance of FindNextMove. So, by synchronizing this function, only one best move can be calculated at a time

Server Logic:

- The server waits for clients to connect
- The server always plays first, since it is always 'X'
- It has a log of all client and server actions, noting against which client the action is made
- The server checks after every move made, if someone has won or drawn, it shows the result.
- Playing again resets the game board for the given client
- It also displays the top 3 scores on the right
- In theory, using the algorithm the way we did, the clients will never win. So, the top 3 scores are always 0, updated every time a game finishes.

Client Logic:

- The client starts the game against the server
- Every time it clicks a button, that move is sent to the server, and that corresponding button is disabled
- Once the game is over, the result is displayed, and all the buttons are disabled until the player chooses to play again
- The play also has the option to quit the game