Samuel Parkin T2A3

# Develop and describe an algorithmic solution for an application that utilises two way communication over a network?

The application will be a game of TIC TAC TOE between only 2 users.

The users will be communicating to each other over sockets with their messages being sent as a pickle.

Players will either need to open the Server.py or the Client.py programe. Both of these files are dependent on the Classes.py file which contains all the Class objects needed in order to connect to each other and the game to be played.

When either the Server.py or Client.py programs are first started it will create the variable server or clients as an instance of the ‘Comms’ class object. This will initiate the variables needed for the socket connections. Within this Comms class there will also be a send and a receive message function that will be called upon during the game.

Within each Server.py and Client.py programs there are induvial ‘Connect’ functions that are unique to each other. The instances of the ‘Comms’ objects that were previously made will be used as the argument of the ‘Connect’ functions. This will initiate a socket connection.

The ‘User\_info’ class object will store variables for each player and will be used throughout the game. This ‘User\_info’ class object is saved as a variable called ‘Game\_info’ that is within the ‘Comms’ class objects that were created for each player. User\_info can be reset when a game has been won and the player want to play again.

To start the playing of Tic Tac Toe, The Game class will be called using the ‘Comms’ objects that were previously made as its input argument. This allows the ‘Game’ object to use the varibable within the ‘Comms’ class for it communication throughout the game. will have functions within it for flow of playing the game.

These will include:

* A function to ask for the users move,
* A function to checking weather the users move has won the game
* A function that creates a dictionary of information for the recipient to receive.

Once the user has made their move it will check to see weather or not the move has won them the game, call to the ‘Display’ class function to clear the screen and display the updated board and send the dictionary with updated information to the opponent.

If the user has won, it will ask whether they would like to play again

The user will then wait for the recipient to reply with their move.

Once the other user has received the information it will call the ‘Game’ class function to check whether the opponent had won. If the opponent has won it will ask if the user would like to play again, if the opponent hasn’t won, it will ask for the users move and continue with the previous steps until there is a winner or draw.