

# 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 program. Both 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 an instance of the 'Comms' class object. This will initiate the variables needed for the socket connections. Within this Comms class there will be:

- Communication variables.
- Send message function
- Receive message function.

Within each Server.py and Client.py programs there are individual '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 inherited by the 'Comms' class and saved as the variable 'Game\_info' within the 'Comms' class. When a game has been won or drawn, and the player wants to play again, the User\_info can be reset.

Within the User\_info class it will include:

- The players move.
- The name of the opponent.
- Whether it's the players turn or not.
- The opponents symbol.
- A function to gather the information needed to send to the other player.

To start the playing of Tic Tac Toe, the 'Game' class will inherit the players 'Comms' object. This allows the 'Game' class to communicate to the other player throughout the game.

This class will also call upon the following classes to preform specific actions.

- 'Tic Tac Toe' class will have the following functions:
  - A function to ask the player to select their move.
  - Check whether the move has won the game.
  - Check whether the game was a draw.
  - Ask if the player would want a rematch.
- 'Board' class will have the following functions:

- Print the board on the user's terminal corresponding to the moves that have been made.

Once the game has been won it will ask whether the players would like to play again, if so, it will reset the User class for the player and start a new game. If not, it will terminate the connection and close the program.

## Flow Chart

