ETEC3702 – Concurrency
Lab 9 – Inter-Process Communication with a Connection (Listeners and Clients)

Due date: 14 April 2020 by the end of the day.

For this lab you are going to write a program that uses multiple processes and connection objects with Listeners / Clients to make a simple multi-player game of tic-tac-toe.

Assignment:

For this program you are going to write a simple game server and game client to play tic-tac-toe.

Server program:

The server program will do the following:

1. Create a listener that listens on your local machine and some port (like 0x1234).
2. Accept two connections from two client programs.
3. Randomly select one of the clients as the active player to go first.
4. Send them a message that consists of the board state and an instruction that let’s them know it’s their turn.
5. Wait for and receive their selected move. If the move is invalid, resend the same board state and command. If the move is valid, update the board state and make the other player the active player.
6. Repeat 4 and 5 until someone wins or the entire board is filled.
7. Send a message that include the board state and the winning player.

Client program:

The client program will do the following:

1. Connect to the server as a client.
2. Wait for a response from the server the will include the board state and a command to make your move.
3. Display the board state.
4. Allow the user to input their move (I’d suggest using cell numbers 1-9)
5. Redraw the board with the new move made.
6. Wait for the next message with a board state with your opponent’s move having been made.
7. Continue until the server sends a message stating that the game is over and that you’ve won or lost.
8. Display the final board state and game result.

Here is some example of what your client output might look like:

```
Connected to server. Waiting for other player...
You are X.  It is your move.

 1 | 2 | 3
---+---+---
 4 | 5 | 6
---+---+---
 7 | 8 | 9

What is your move: 5

 1 | 2 | 3
---+---+---
 4 | X | 6
---+---+---
 7 | 8 | 9

Waiting on opponent...

You are X.  It is your move.

 O | 2 | 3
---+---+---
 4 | X | 6
---+---+---
 7 | 8 | 9
```

(and so on…)

Hints:

- Your clients will need to go through a loop of alternating send / recv messages on the connection.
- You will need to exchange messages from the server to the client that include the board state and a command that indicates something about the state of the game. You will likely want to play out the format of these messages.
- You will need to exchange messages from the client to the server that indicate the move made. You will likely want to plan out the format of these messages.
- To test your program on your local machine you'll want to use localhost address or 127.0.0.1 and you'll need to use multiple python shell instances each running a separate program instance.

Bonus:

1. Write a graphical version of your tic-tac-toe client that allows users to click in boxes to make their moves. (5 points)
2. Modify your program to include the entire board and rules of "ultimate-tic-tac-toe" as described here: https://en.wikipedia.org/wiki/Ultimate_tic-tac-toe (10 points)