CSE 5462: Lab5 (100 points)
TicTacToe - MultiPlayer

Demo in class or by appointment: March 4
Electronic Code Submit Deadline: 9pm, March 4

Important: You should work in teams of 2 for this lab. See me if you can't find a partner. It is important that each partner clearly identify what parts of the code they write. Do this with comments in your code. Each partner is expected to write ~50% of the code.

We will demo in class or in lab. For in class demos, we will go around the room. Be prepared to demo at the start of class. The submitted code will be used only to verify that you did not copy from others, to compile and re-run your program, to make sure you were indeed demonstrating your own code, and to grade for documentation of your code. You may be asked to connect your client/server to a client/server from another pair of students!

- For socket programming use can either use the CSE linux system (csegrid) or your own machines. For your own machines, demo can be done in classroom.

In this lab, you will take Lab 4, and create 2 separate applications (if you haven't done this already) each one runs on a separate machine:

- tictactoeClient
- tictactoeServer

You will modify the tictactoeServer so that:

- the computer picks the next move (it can be a very simple algorithm, one will be shown in class)
- the server can handle multiple clients sending data to it (but the server will remain single threaded)
- it can handle multiple types of 'requests' from the client (NEWGAME, MOVE)
- when a NEWGAME request is made, it creates a new board for the client/server game

You will modify the tictactoeClient so that:

- It asks the server to create a NEWGAME
- It passes the game number and move on every move

The command line parameter to run the tictactoeServer program are:

- portnumber that the programs will communicate on,

The command line parameters to run the tictactoeClient program are:

- portnumber that the programs will communicate on,
- ip address of player 1

You do NOT have to handle lost datagrams in this lab, but you should start thinking about how to handle that.

Submit well-documented and well indented code along with a README file explaining how to run the program. You must also submit a makefile. Submit it using GitHub, in a subdirectory called Lab5

The format of what is sent between client/server is to be defined in class.

The grading rubric is as follows:

1. Program correctness and robustness: 70% – Application has to work to get these points
2. Coding style (e.g., comments, indentations): 20%
3. Documentation (the README file) and design document: 10%

What you should get from this lab?
Tools:

- basic datagram socket IO (socket(), close(),read(), write() etc);
- select() command
- timeouts on socket (setsockopt()).

Concepts:

- how to restructure code to make it more clear
- handling edge cases – identifying which ones are easy/hard
- how multiplexing works and when you need it