CSE 5462: Lab4 (100 points)
TicTacToe - Datagram

Demo in class or by appointment: Feb 23
Electronic Code Submit Deadline: 9pm, Feb 23

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 are to change the socket type from Lab3 from STREAM to DGRAM.
If you write this as a single application, then the command line input will be:

- portnumber that the programs will communicate on,
- player number (1 or 2), where player 1 is the player creating and binding to the socket (not needed if using 2 programs)
- ip address of player 1 (this will only be used by player 2)

This code is the basis for lab 5, 6, etc. You do NOT have to handle lost datagrams in this lab, but you should start thinking about how to handle that. Your application should 'timeout' after 'n' seconds (you choose 'n') if the other side doesn't respond. The application should gracefully handle the timeout.

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 Lab4

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);
- bad return codes from read()
- arrays/structs in C;
- timeouts on socket (setsockopt()).

Concepts:

- how to modify/re-purpose code from connection oriented to connectionless;
- determining when the other application has hung-up on you in connectionless world
- how to gracefully handle errors;
- how to handle garbage sent (intentionally or not);

4.