Count-It-First Networked-game Project Proposal

Prepared for:
Dr. Aaron Dingler
CSC 4750/CPE 4750 - Computer Networks

Prepared by:
Hung Nguyen
Tai Doan
Huyen Nguyen

1. Project Overview

The vision of our group upon this project is to provide a CLI-based, client-server game using TCP sockets (related to lab 1) for multiple clients (from 2-3 clients currently to connect to the server to play the game). The game will be implemented in **Python 3**. We all have different IDE preferences so we would share GitHub repository to connect all contributors (using Terminal command line to commit codes and work on it together even though we cannot see each other in person). We make sure that it can work on any CLI-based computer such as MacBook, Linux, or Ubuntu on Window.

2. Project Description

- Network connection: Our game requires a constant connection between client and server so we will use TCP sockets as it ensures a reliable data transmission, and allow a server-client connection setup before the game starts. At this point, we will use localhost (127.0.0.1) on port 43500 to test develop and test the program. We need to communicate with Dr. Dingler more about this matter.

- Game mechanic: The server will be at least accepting two connections but no more than three connections. Once the game starts, the server will then generate a random number based on the number of connected clients. For example: If there are two clients, generate a random number between [20:30], if there are three clients, generate a number between [30:40]. Clients will take a turn to add a number starting from 0, each client will have two options whether to add 1 (+1) or add 2 (+2) to the current total number until it reaches the generated number given by the server. Winner will be stated as whoever reaches the generated number first.

- Functional requirements:
  - Server:
    - After the program is executed, the server will set up a room with an assigned number of players (2 or 3). The server will be always on an assigned IP address and port number. The server allows new connection from the clients (up to 3, any upcoming connection request after that point will be refused).
    - The server will generate a random number and announce it to the clients
    - The server is able to keep track on the current turn and to receive a message (a number input) from the client and broadcast it to other clients

        - The server is able to keep track of the current total number and announce the winner to the clients
    - Client:
        - The client will send a connection request to the server if they both got connected with the assigned socket connection.
        - As client connected, they will be assigned as player 1, 2 or 3 according to their connection order. The game starts when all the clients needed for the room are connected.
        - The client will have two input options (+1) or (+2) and will be instructed with a guide message. Client who does not follow the rule will be prompt to input an appropriate number.
        - During the game, each player can keep track of the current play turn and total number
        - If a specific client tries to skip their turn or try to input value twice when it is not their turn, the server will not allow and give that player a warning notification. (blocking mode)
        - If a client shut off the application in the middle of the game, the game itself will stop working.
        - When a specific player wins by reaching the generated number first, they will get the notification who the winner is by the server.
- Optional/additional features:
    - GUI for client
    - Different difficult levels: More input options are allowed, support other mathematical operations

3. Project Timeline

May 6 - May 13:

- Program the counting algorithm of the game within 2 players on both client.py and server.py files
- Test between two clients if the algorithm is correct and makes sure each player takes only one turn (no players should input twice and input values which does not allow for example +3, +4 or random words)

May 13 - May 20:

- Test with assigned IP Address and Port (ask information from Dr. Dingler)
- Develop and test for allowing 3 players

May 20 - May 27:

- Debug the program
- Develop optional/additional features

May 27 - June 3:

- Integration Testing
- Wrapping up the Project by finishing the project network system design paper, checking for comment style, and final checks the coding
- Add README description as the last step
- Check up with Dr.Dingler if we need to add in anything.