CS494 - INTERNETWORKING PROTOCOL

LAB 01: SOCKET PROGRAMMING

- *Duration: **5 weeks**.*
- *Team: maximum 3 members, minimum 2 members.*
- *Game: Get the sum of the last digits of your StudentIDs and then modulo by 3. The result is your Game you need to implement.*
    - *Ex: Your team 1911203, 1901458, 1911970*
    - *(3+8+0) mod 3 = 2: Game 02*
    - ***CHOOSE WRONG GAME: YOU'LL GET 0 POINT.***
- *Submission: Only one file: **StudentID1_StudentID2_StudentID3.zip** which contains the following folders and files:*
    - *Folder **Source**: your source code, Client and Server*
    - *Folder **Release**: your executable game and the database files.*
    - *File **ReadMe.txt**: contain the youtube link that you demonstrate the game.*
    - *File **Report.pdf**: contain the self evaluation (which is defined in the end of the document), you should define the game story, what does client send to server and what does server send to client? You also have to define the structure of the packets that are sent between server and client and define the steps/conditions to check the packet at server and client. Explain the important snippet codes. If you use the code from the internet, please give the references.*
- ***IF YOU COPY EACH OTHER, BOTH WILL GET 0.***

GAME 00: THE MAGICAL WHEEL

Develops a simple game "***The Magical Wheel***" with a server as a Referee and N clients (players) (N is defined by the server in advance, 2<= N <= 10). The rules of the game are:

1. For each player, the player needs to register to the server to join the game and choose the nickname. You cannot choose the nickname which is already registered. The nickname is composed by the following characters ‘a’...’z’, ‘A’...’Z’, ‘0’...’9’, ‘_’ and the length is not longer than 10 characters. If the player uses the same nickname with other players, the server will ask the player to choose another nickname. For each successful registration player, the server announces “Registration Completed Successfully” and the order of the player.
2. When the server receives enough connections (players), the server gives each player 0 point and once again announces the order of the players to play the game.
3. Game loading:
    a. Server read the data from the file “database.txt” which includes the information:
        i. First line: n - the number of keywords and the descriptions of the keyword.
        ii. The next n pair of lines stores the question, each question has 2 lines:
            1. First line: the keyword, case insensitive and no spaces.
            2. Second line: the description of the keyword to help the player to guess. (You can choose any topic that you want)
        iii. The length of the keyword is not longer than 30 characters.
4. Gameplay:
    a. Server chooses the random keyword in the database, then sends the length of the keyword and its description to all clients.
        i. For example:

            The length of the keyword is 6: ******

            Hints: One of the most popular interpreter programming languages.
    b. Each player sends to the server their guess which is one character in turn. The order is the order of their registrations. The message that is sent to the server includes: the guess character and the full keyword (only accepted after two turns).
        i. For example:

        Client 1 sends:

            The guess character: a

            The keyword: (blank) (not after two turns)

        Client 2 sends:

            The guess character: p

            The keyword: python

ii. The server needs to check whether the turn is greater than 2 or not. If not, the server will ignore the keyword from the client.

c. When the server receives the answer from the player, it will send the correct answer to the players. If the player has the wrong answer, the turn will move to the next player. The point will be granted as following:
   i. Wrong guess: 0 point.
   ii. Correct guess: 1 point.
   iii. Correct keyword: 5 points.

d. The structure of the response from the server:
   i. Case 1: If it receives a wrong answer. Move the turn to the next player. Character 'a' is not in the keyword.
   ii. Case 2: If it receives a correct answer. Continuing guessing. Character 'p' has 1 occurrence. The current keyword is "p*****" (Announces the current keyword for all players).
   iii. Case 3: If it receives a correct keyword. Congratulations to the winner with the correct keyword is "python". End game.
   iv. Case 4: If it receives a wrong keyword. The player who submitted a wrong keyword is disqualified. Move the turn to the next player.

5. After the server has calculated the point, it announces the current point to the players.

6. Notes:
   a. The game ends if one of the conditions is satisfied:
      i. After 5 turns and no one cannot guess the keyword.
      ii. At least 1 player has a correct keyword after 2 turns.
   b. When the game ends, the server summarizes the score and announces the points as well as the ranks to N clients.
   c. When the game ends, the server continues receiving the connection from N clients and starts the new game. The game will end only when the administrator ends the game on the server.
   d. You should set the countdown for each turn.

GAME 01: WHO WANTS TO BE A MILLIONAIRE?

Develops a simple game "**Who Wants to Be a Millionaire?**" with a server as a Referee and N clients (players) for each game (N is defined by the server in advance, 2<= N <= 10). The rules of the each set are:

1. For each player, a player needs to register to the server to join the game and choose the nickname. You cannot choose the nickname which is already registered. The nickname is composed by the following characters 'a'...'z', 'A'...'Z', '0'...'9', '_' and the length is not longer than 10 characters. If the player uses the same nickname with other players, the server will ask the player to choose another nickname. For each successful registration player, the server announces "Registration Completed Successfully" and the order of the player.
2. The server randomly chooses a set of questions from their database.
3. The server sends to all players the information of : the number of players, their orders and the number of questions in this set.
   For example: We have 2 players, a player plays in the second turn, the number of questions in this set is 20. The order of the players as well as the number of questions (the number of questions should be greater than three times the number of the players) which are randomly selected by the server.

4. Each player answers the question in turn from the first player to the last one (N) and then goes back to the first player until the game ends.
   a. The game ends if one of the conditions is satisfied:
      i. There is only one player left. This is the winner.
      ii. The players answer the last question.
   b. Each turn is performed as:
      i. The server announces the player that he/she is in his/her turn and the question.
      ii. The player has 2 options:
         1. Answer the question and send his/her choice to the server.
         2. Do not answer the question and move the turn to the next player.

      Note: Each player can move his/her turn only one time in the game.
   c. When the server receive the choice from the player:
      i. If the player is the last one, he/she is a winner no matter what the result of his/her answer.
      ii. If the player is not the last player, that player chooses to answer the question and has a correct answer, the server will send the next question to that player. If the question which is sent to the player is

the last question, that player is also the winner, no matter what the result of his/her answer.

iii. If the player is not the last player, that player chooses to answer the question and has a wrong answer, that player is disqualified.

iv. If the player chooses to move his/her turn (only if this option is available), the server will move that question to the next player.

5. Server starts another set.
6. You should set the countdown for each turn.

GAME 02: RACING ARENA.

Develops a simple game "**Racing Arena**" with a server as a Referee and N clients (players) for each game (N is defined by the server in advance, 2<= N <= 10). The rules of the each set are:

1. For each player, a player needs to register to the server to join the game and choose the nickname. You cannot choose the nickname which is already registered. The nickname is composed by the following characters 'a'...'z', 'A'...'Z', '0'...'9', '_' and the length is not longer than 10 characters. If the player uses the same nickname with other players, the server will ask the player to choose another nickname. For each successful registration player, the server announces "Registration Completed Successfully".
2. When the server receives enough connections (players), the server sends the length of the race and the start position to the players (the length of the race should be greater than 3 and less than 26). The length of the race is defined by the administrator who manages the server and the start position is 1.
3. For each set:
    a. Server randomly choose 2 integers (from -10,000 to 10,000) and a operator (+,-,*,/,%) and sends them to the players.
    b. The players receive the expression and the players should send the result of that expression to the server.
    c. When the server receives all the answer from the players, the server send the correct answer to the players and the points is granted as follow:
        i. If the player has a wrong answer or loses their turn (out of time), they get -1 point. The duration for each question is defined by the server at the beginning of the game.

ii. If the player has a correct answer and he/she is the fastest player, he/she will get M points, which M is the total points that the other players lose. The other players who have the correct answer get 1 point.

iii. When a player has wrong answers 3 times consecutively, that player is disqualified, the number of the remaining players is updated and is announced to the players.

iv. The points in each question are the steps that players move in the race (move forward if they get positive points and move backward if they get negative points and they cannot move behind the starting points).

d. Server updates the points, the position of each player in the race and announces them to the players.

4. Repeat the game until we have a winner who reaches the finish first.
5. Server starts another set.
6. You should set the countdown for each turn.

CS494 - INTERNETWORKING PROTOCOL

LAB 01: SOCKET PROGRAMMING

SELF EVALUATION

1. Team Information:

No.	Name	Student ID	Contribution (%)
1			
2			
3			

2. Score Sheet:

No.	Requirements	Score	Evaluate
1	Use C/C++, Java, C#	2	
2	Implement whole gameplay properly	3	
3	Socket Non-blocking	2	
4	Have a good GUI (MFC, WPF, Swing, etc.)	3	
	Total	**10**	