CS408 - PROJECT STEP 2 – 2018/19 FALL TERM

Our system creates a game where at least two or more players play a quiz game. We created safe-threads through the use of delegate and invoke methods. Server starts a safe-thread of "Accept" and from Accept Thread, we allow clients to connect to server. Accept thread will stop accepting connections after the game starts (or server shuts down). After accepting a client, all the information related to that client is added to a Player Class object which we implemented to store all client information such as score, name, socket, etc. Furthermore, a safe "Receive" thread starts running (in accept thread). In receive thread, if there are at least 2 clients connected to the server, the start button becomes available, obviously game will start when the operator presses the start button.

At the first round, players start accordingly to their names (e.g. alphabetically). After first round, game follows a round-robin fashioned gameplay and lets the next player ask. The player who is going to ask a question is labeled as client, others answers (a 'foreach' block in 'foreach' block). If a player leaves while it's their turn to ask, game continues with the next player by sending him/her "Ask question" command. After each turn (not round), scoreboard is sent to all users so that they'll know their current standings. Once the game finishes (e.g. reaches the round limit), users points are sorted and compared in a special function and winner is announced from the server. If there are tied scores, names of the players with same score are displayed instead of a single player. Throughout the implementation, we've made use of ASCII characters. When the server sends a message to clients, the message is acquired properly to its leading ASCII value. To exemplify;

- `ASCII value of number 2 is used for preventing connections to the server.`
- `ASCII value of number 4 is used for exit situations (when a player exits).`
- `ASCII value of number 5 is used for acquiring answers.`
- `ASCII value of number 6 is used for acquiring questions`

While detecting these values from messages, we made use of "$" symbol. Note that users are not required to enter this symbol in their message (and they probably shouldn't). Whilst sending/receiving messages, this symbol is added to the message with its appropriate ASCII value behind. And once it gets received, the message is parsed with "`Split()`" function from the dollar symbol and the ASCII value is gathered.

These were the things we accomplished. Ultimately, it is noteworthy that the initial question and its answer is separated with "?" as in "question?answer", meaning if a user enters a question without question mark, the program is unable to detect and separate the question and the answer properly, so we expect users to put a "?" after question is ready, and answer afterwards.

Atilla Alpay Nalçacı
Deniz Ademoğlu