Zach O’Neill 12/18/19

CSC-200 Writing Assignment

The beginning of the project was rough, as a lot of the information I could grasp but didn’t fully comprehend, so at the start of the project most of my time was spent researching and looking at other work to get a grasp on how the networking and threads appear in a real situation. Once I understood those I came up with an idea for a game, 21 questions. As I played it all the time in iMessage I knew the concept and how the game worked so it would be simple. But a problem arose that if I wanted to have 2 players, they would have different interfaces/guis because one would only be able to give yes and no answers while the other could ask questions. So after thinking of all the ways I could go about solving this I decided to scrap the idea and go for more of a numbers approach with a math game, similar the papers I did as a kid, with a goal of being the first one done.

Again, the begging was very rough as I didn’t know as much information as I would have liked to know. As soon as I knew what I was doing it was simple. I have a client.java that connects to a server.java, and then when 2 players connect to the server the game begins. In the beginning the client class allowed players to type in the answer they wanted and press a button to send that answer to the server, I modified that slightly later on by implementing buttons in a calculator like style. This allowed the class to control what players put into the text field, to some degree, and also made the game a bit more tedious and fun. Once the server receives the answer from the client it checks if it is correct, and tells the user, then prints a new equation, doing this makes it so players get a bigger penalty for a missed question. Once the player has reached 5 correct questions the game prints out to both players the winner.

There was not very many problems but when there was one, it took some time to figure out and solve them. The first problem that arose was allowing both users to proceed to different equations if one got one right and the other was either still behind or wrong. I thought about rewriting almost half my code so that it implemented an equation object that was passed in, or creating 5 equations once the program was loaded up. But in the end the best solution was 2 separate threads that ran at the same time, that way both users wouldn’t be affected in any way by the other user, this also made the program a whole lot easier to code in the end. Another problem was random crashing, the program would sometimes randomly just stop working and proceed to crash with no errors. As it happened more and more I eventually figured out that the cause of these crashes were when the program could not proceed to other parts of the code, such as being stuck in a loop. Once that was figured out, it was a lot easier to pinpoint the crashes and problems.

Looking at my game now, if I were to work on it more in the future I definitely have some more ideas to improve the game overall. I would replace the text area completely so that the game can look a little better overall. I would also configure the text field more to look like an actual calculator screen so then the game looks accurate. Also looking towards the future I would question if it’s possible to allow the game to be played up to as many people as can join, say 5, or 8, not a fixed number such as the game is now with just 2. Assuming it’s possible that would be the hope for the future, to implement a way for more than 2 people to join that so friends can challenge themselves in “simple” math to see who is the fastest.

