Unity HW4 Wander.

Instructor: Dr. Ilmi Yoon

Student: Gangzhaorige Li

Student ID: 918812380

Spring 2020

GITHUB:https://github.com/gangzhaorige/hw4

Group: no group. Teammates are busy until the day before the deadline.

Summary

HW4 is basically the communication between client and the server.

In this case clients are C#(Unity) and server is java.

In this homework I created 4 protocols. Both server and client side.

1. Login
2. Register
3. PlayerLocation
4. SpawnPlayer

Login and Registration are both connected to the Amazon RDS (mysql).

The basics of Client and Server Communication.

First the connection between the client and server must be established. (Address and Port)

Next Server runs in an infinite loop to accept any request that comes from the client.

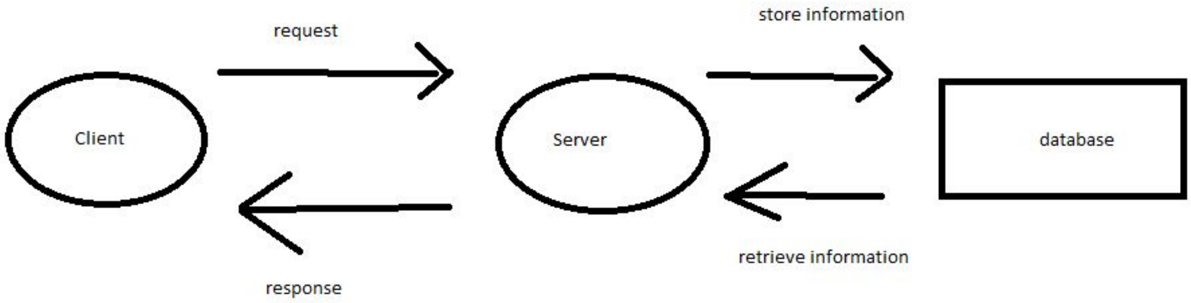
Client creates a request object and sends it to the server.

Server gets the request object. Based on the request server does the job and creates a response object containing information about the status and other information. Server sends the response object back to the client.

Client now receives the response and based on the response updates current users information.

For example, Registration. Client sends username, password, and confirmed password. The server receives the information and checks the database if the username is already in the database. If not it then checks if password and confirmed password are equals. If yes, the server stores the information inside the database. And creates a response object with the status that the user registration was successful. On the other hand if the username was already registered or if the password and confirmed password are not equal based on the situation creates a response that the user already exists or password does not match. The passwords are usually hashed on the client and server side to provide a safe security.

This is how it looks visually.



The same idea goes for the Login, SpawnPlayer, and PlayerLocation.

Reflection.

The communication between server and clients are pretty easy to understand. The hard part about the project is that for one protocol I had to write 2 requests, 2 responses, and main script.cs that does something based on the retrieved responses. Since my partner was busy and I did the hw4 alone, there is nothing I can learn from my partner. However, I was able to teach what I learned about the communication between server and clients to the other groups. I helped them to debug and establish a database (mysql) connection to the server.

The request objects all of them are inheritances. All of them are subclasses of NetworkRequest. Same goes for the response objects all of them are subclasses of Network Responses.

Unity merging is really tricky. Our policy is that everyone that works with a github must have the smart merge setup correctly. Also for this homeworks since servers and clients are separated it is much easier to merge.