INFO 6016 Final Project (Final Exam)

100 marks

Due Date: 11:59 EST January 15th, 2017

All work for this project must be done by you, and you alone. This means no other student can help you in any fashion. I will be looking at the code you submit in great detail with compare tools. You may use your buffer class from your previous assignments.

Introduction
You are tasked with designing and developing a game lobby browser. This game lobby browser will be used by players to inform them what game lobbies are open, and which game lobbies they can join. You are responsible for developing the game client, game server, authentication server, and database(s) for the game.

Your code must be done in C++. You must use raw TCP sockets. No websockets, or libraries that handle deserialization for you.

Technical Requirements (50 marks)

Game Client (12 marks)

1. Must use TCP as its primary communication method. **(1 mark)**
2. Must use length-prefixing for message framing. **(1 mark)**
3. Uses Binary for serialization. **(2 marks)**
4. Network requests should be asynchronous, or run in a different thread. **(2 marks)**
5. Deserialize using length-prefixing properly
    1. Handle multiple messages in one receive. **(2 marks)**
    2. Handles partially received messages *(waits for more data based on the length-prefix)*. **(2 marks)**
    3. Checks if there is enough data to read the header first. **(2 marks)**

Game Server (14 marks)

1. Must use TCP. **(1 mark)**
2. Uses the same serialization method as the client. **(1 mark)**
3. Uses Binary for serialization. **(2 marks)**
4. Deserialize using length-prefixing properly
    1. Handle multiple messages in one receive. **(2 marks)**
    2. Handles partially received messages *(waits for more data based on the length-prefix)*. **(2 marks)**
    3. Checks if there is enough data to read the header first. **(2 marks)**
5. Handles concurrent users using non-blocking sockets, or threads. **(2 marks)**
6. Network requests should be asynchronous, or run in a different thread. **(2 marks)**

Authentication Server (22 marks)

1. Must uses TCP. **(1 mark)**
2. Uses the same serialization method as the game server. **(1 mark)**
3. Uses Binary for serialization. **(2 marks)**
4. Deserialize using length-prefixing properly
    1. Handle multiple messages in one receive. **(2 marks)**
    2. Handles partially received messages *(waits for more data based on the length-prefix).* **(2 marks)**
    3. Checks if there is enough data to read the header first. **(2 marks)**
5. Must be able to register an account
    1. Must use SHA-256 as the hash algorithm. **(1 marks)**
    2. Must use a randomized salt for EACH password. **(2 marks)**
    3. Must add this account to the MySQL Database. **(2 marks)**
    4. Must respond with a failure reason on failure. **(1 mark)**
    5. Must respond with a "success" on success. **(1 mark)**
6. Must be able to authenticate an account
    1. Must hash the plaintext with SHA-256. **(1 mark)**
    2. Must compare this hash to the database hash properly **(1 marks)**
    3. Must respond with a failure reason on failure. **(1 mark**)
    4. Must respond with a success on success. **(1 mark)**
    5. Must update the `last_login` column in the `user` table in MySQL. **(1 mark)**

Player Database (2 marks)

1. Must use MySQL. **(1 mark)**
2. Create a table called accounts. **(1 mark)**
    1. id PRIMARY KEY AUTOINCREMENT NOT NULL
    2. username UNIQUE NOT NULL
    3. salt NOT NULL
    4. password NOT NULL
    5. last_login NOT NULL

Project Requirements (50 marks)

Game Client (23 marks)

1. The client must be able to:
    1. Connect to the game server with an ip and port number **(1 mark**)
    2. Register, and authenticate an account. **(2 marks)**
    3. Create a game lobby. **(2 marks)**
    4. View the game lobby list. **(3 mark)**

5. Refresh the game lobby list. This can be done automatically, or by client input. **(2 marks)**
6. Join a game lobby. **(2 marks)**
7. Leave a game lobby. **(2 marks)**

2. If the host leaves the lobby, the game lobby must close and all joined players must be booted from the lobby. A broadcast must be sent to players that the host left the lobby. **(4 marks)**
3. The player must be displayed the possible map details that the player can created using the create command. **(2 marks)**
4. The game lobby browser list must contain: **(3 marks)**
    1. The name of the map
    2. The name of the Lobby
    3. The mode of the game
    4. The number of open spots
    5. The total number of spots
    6. The name of the host

Game Server (22 marks)

1. Either hard code in a separate code file, or load from an external file the game info options, or use a MySQL database (*Map Names*, *Game Modes*, *Max Number of Players*) **(2 marks)**
2. If a player is logged in on one socket, the player is not allowed to log in on another socket. Meaning two clients can not log in using the same username. **(4 marks)**
3. When a player creates a game lobby, store the game lobby information in a data structure to keep track of the game lobby information, and players who join the lobby. **(4 marks)**
4. If a player tries to create a game lobby with a name that already exists, alert the player that the game lobby name already exists. **(2 marks)**
5. When a player joins a game lobby, broadcast a message to all other players in the lobby that the player has joined the lobby. **(1 marks)**
6. When a player leaves a lobby, broadcast a message to all other players in the lobby that the player has left the lobby. **(1 marks)**
7. If a client's connection is dropped, remove the player from the game lobby they are in, and broadcast a message to all other players in the lobby that the player has dropped. **(2 marks)**
8. A player can not join a full game lobby. **(1 mark)**
9. If a player attempts to join a full lobby or the lobby does not exist, alert the player of the reason. **(1 mark)**
10. When a host leaves a game lobby or is unexpectedly disconnected, send a single packet to each of the players in that lobby that will contain a message to return them to the game lobby browser, and a message informing them of the reason they were booted from the game lobby. **(4 marks)**

The project must be done in git, please keep your commits small and sweet. **(5 marks)**

Submission Requirements

1. MUST include a ReadMe that includes:
    1. The project configuration settings
    2. How to build your projects.
    3. How to use your programs.
        1. How to start the authentication server.
        2. How to start the game server.
        3. How to connect a client to the game server.
        4. Detailed instructions on how to use your game client.
2. .zip up your entire project.
3. Must include your .git folder to receive marks for git.
4. Must include a mysql .dump of your database that I can import into my MySQL server. OR credentials to your remote database that I can access. You Must ensure the database is online, and accessible to me when the project is being marked.
5. **You will be deducted <u>20%</u> if any of these requirements are not met.**

Example of a Game Lobby Browser (Id is optional):

<u>Id</u>	**<u>Map Name</u>**	**<u>Lobby Name</u>**	**<u>Game Mode</u>**	**<u>Players</u>**	**<u>Host</u>**
3	New York	Test your might!	Free For All	2|6	sous_chief
8	Toronto	Q(*_*Q)	Team	1|8	F0r3v3rAl0n3
13	Sydney	Need 1 more!!	Capture the Flag	5|8	_popcorn

Example of Client Commands:

Client Connection:

- /connect *ip port*

Account Authentication:

- /register *username password*
- /login *username password*
- /logout

Game Lobby Browser:

- /refresh
- /create *?*
- /create *map mode name*
- /join *id*
- /join *name*

Game Lobby:

- /leave