BattleFleet Documentation

Susanna Byun && Alex Carswell

Self Grading Sheet

Working application level game - 5 points
Yes, the game works.

Required network architecture - 5 points
The network protocols are organized in a modular fashion, with a thread handling each client.

No crash in play - 5 points
At no time does our game crash or freeze when playing.

Register - 5 points
We have a protocol implemented that allows a player to register with a username and password. The password (along with all other protocol messages that are being sent to the server) is encrypted using a very simple arithmetic algorithm.

List Games - 5 points
The protocol successfully lists all games. This is done automatically when the player enters the game room. An integer is sent to determine how many game rooms will be sent. Then, each game room information is sent to the client. After each message is passed, the client sends back an acknowledgement indicating that the message was successfully received.

Create Game - 5 points
The protocol to create a game is successfully implemented. The player that creates the game sends the game name and username. This player is automatically set as the host of the game. On the server side, a game room object is created to keep track of all this information and the game room is added to the game room dictionary that keeps track of all game rooms. Then, the server sends the room number to the client, which is determined by a room number counter.

Join Game - 5 points
The join game protocol is successfully implemented. Once a client tries to join a game, it checks if the room is full or not. If it is full, it sends a 0, not allowing the client to change scenes. Else, it sends a success acknowledgement and adds the player as the opponent of the host in the game room.

Exit Game - 5 points

Out exit game implementation allows a player to disconnect gracefully from the game. It is used whenever a user clicks the “Exit game” button in the lobby and when a user times out from a currently running game (this however sends a move protocol message with lose connected to it so that the other player receives a win message by default). In both cases, the TCP connection is properly terminated and the client application closes.

Unregister - 2 points
The unregister protocol simply sends “unregister [username]”, where the server removes the player from the player dictionary. However, due to lack of time, the server side is implemented, but the unregister button has not been implemented on the actual game.

Application Specific protocol
Fire - 7 points
The protocol to make a fire move is successfully implemented.
Move - 7 points
The move protocol is successfully implemented.
Game Ready - 3 points
The game ready protocol allows us to synchronize the start of the game with both players. Once a player enters a game they must first place their ships which means that they will not both necessarily finish placement at the same time. This means that synchronization between the start and end of this phase with both players is crucial which is what this protocol enables us to do.
Make move - 3 points
The make move protocol simply relays the same message to the other player.

Bonus features

Global chat - 1 point
The global chat was implemented and working at one point. It had a queue for all messages that are pushed to other clients. When a globalchat protocol is initiated, it simply adds to the queue and a separate thread pushed that message to all clients that are currently in the lobby. However, while changing protocol logic, it is no longer working.

ScoreBoard - 3 points
We keep track of all players. A function sorts all the players by global score and returns the top 10 players.

Documentation - Done. 10 points