Myles Greene-Beaupre 04/06/2022
C12195336
ECE 470

General Architecture/Scenarios

NOTE: I have decided to change my project to be a multiplayer trivia game instead of a file transfer application, so the general architecture and scenarios below are about a trivia game where a number of users can participate simultaneously and they will all be presented with the same trivia question, and the first person to answer the question correctly gains a point.

General Architecture

The trivia game system will use UDP to receive messages from clients. When the server is started, it will go into a UDP listening loop waiting for client messages. At this point, the server will be aware that a game is not in progress through the use of a variable passed to the function that interprets incoming messages, and so it will only be accepting messages containing login requests or player ready confirmations. A login request will simply contain a username that the client typed in, and when the server receives it they will be added as a player to the game. There is no password authentication required, but if you try to login while a game is active you will be put on a waitlist until it ends. Once every player in the pre-game lobby has sent a ready request, the server will flip the variable to "game in progress" and thus will waitlist incoming login requests. Once the game starts, every user will receive the first trivia question in a series of rounds, and the first client to send back an answer message containing the correct answer to the trivia question will be given a number of points equal to the number of players in the game. For every correct answer after this, they will be given a number of points equal to the number of players in the game minus the number of people who got the answer correct before they did. For instance, the second person to answer the trivia question correctly will be awarded (numplayers - 1) points, and the third person (numplayers - 2), and so on. Players can keep submitting as many answers as they want until they get it correct, but if they do not submit a correct answer before the round time is up then they will be awarded zero points. The server will send them the number of points they were awarded if they got it right, or a message indicating that their response was incorrect if they got it wrong, or a message

indicating they got zero points if the time runs out. Once a round ends and everyone's points have been recorded by the server, another round starts and this process repeats until the end of the last round where the server sends out a list of players in the order of their number of points, indicating that the winner is #1, second place is #2, etc. After this, the server goes back into the pre-game lobby state and takes everyone on the waitlist and puts them into a new game, and then waits for everyone to be ready before starting another game. The actual trivia questions that will be displayed will be kept in a file accessible by the server where they will be selected at random and will rotate so repeats do not happen often. The way this rotation will work is when the server starts, it will read every trivia question in the file into a priority queue-like data structure so that high priority questions are used first and then marked as low priority after use, so that we only start repeating questions when we have gone through every possible question.

Scenario 1: Join Game (client → server)

The client sends a join game message containing their username to the server, assuming that the server has been started. If the server determines that a game is in progress, the client will be put on a waitlist until the game ends. Once the game ends, the client will be put into the pre-game lobby for a new game and will be sent a message telling them to enter 'R' when they are ready. If a game was not already in progress, they will be put into this pre-game lobby immediately instead. In both cases, if the user enters a name that is already in use, either in a game currently being played, on the waitlist, or in the pre-game lobby, they will be prompted by the server for a different name and will not be put into the pre-game lobby or waitlist until they enter an unused name.

Scenario 2: Player Ready (client → server)

Once a client has done the join game scenario, the server will send them a message telling them to send 'R' when they are ready. A message containing anything else will be ignored by the server, and the client will be informed of this. The ready message will also contain the user's name, so when the server gets an 'R' from a user it will send to all users in the lobby a list of everyone they are playing with, and whether or not they are ready to play (so everyone in the lobby will be able to see who is delaying the game).

Scenario 3: Answer Question (client → server)

Answer messages will be ignored by the server unless a game is in progress, and the username contained in the anwer message is the name of a user currently playing in the game (not waitlisted). When a server receives an answer message, it will check to see if the answer contained within it matches the answer to the last trivia question that was displayed to the players. When it is checking to see if a player's answer matches the correct answer, capitalization will be ignored but if you spell the answer incorrectly and/or your answer does not match the length of the correct answer, it will be considered incorrect. If the server determines an answer is wrong, the person who sent the message will get a message back indicating it was incorrect, and they will be allowed to enter another answer. If it was correct, the server will calculate the number of points to award based on how many people got the answer right before them, it will increment their points on the server side, and then send a message to the user that submitted the answer indicating that it was correct and how many points they earned that round, as well as their total number of points for the whole game.

Scenario 4: Global Message (server → all clients)

This scenario is more general, but it is any situation in which all of the players in the game must be given the same information to display on their screens. The server will construct the message, and then send it to all clients with an indication that it should be printed to the player on the client side. This scenario will happen in the pre-game lobby, whenever a client indicates that they are ready every user that is currently in the lobby will be sent an updated list of everyone who is ready and everyone who is not. This scenario will also happen for displaying the same trivia question to all players of the game, and for messages such as "a game is starting soon." Finally, this scenario will happen at the end of the final round in a game, in which case every player will receive an ordered list of all players and their score with the winner at the top, then second place, then third, etc. all the way to last place. By abstracting all of these situations to one scenario, the system is made much simpler since this "global message" scenario will be implemented as a function that can be used in many cases.