How the Game Works

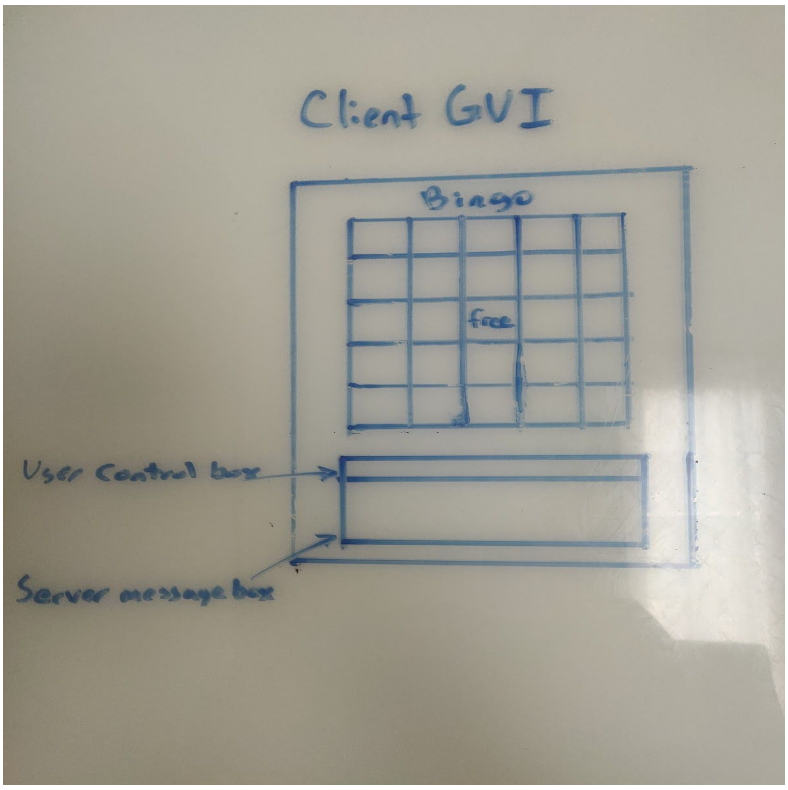
Once at least four clients are connected to the server the game may begin. The server will output each drawn number on a timer. After x amount of time has passed the server will send the next drawn number. The clients must click the drawn number (button) on their bingo sheets in order to activate that number. Each client has the ability to view the drawn number history to check if they had missed a number. The clients are only able to click a number that has been drawn, selecting a number that has not been drawn will do nothing. Once a client believes they have bingo, they will click the Bingo button which will alert the other players that player X has bingo. Then the server will decide and output whether or not player X did in fact have bingo. If they didn't some kind of consequence will happen, or not. If they did, player X will get a point and the game will restart.

New clients may join the game at any time. This means they will also have access to the drawn number history, if they join mid-game. One thing to handle is when the total clients connected to the server falls below for 4. Should we continue the game? Should the game be terminated and wait for additional players?

Both the server and client backends communicate by sending one another a series of instructions. It is through these instructions that the game will be executed. There is a detailed list of instructions below.

Client

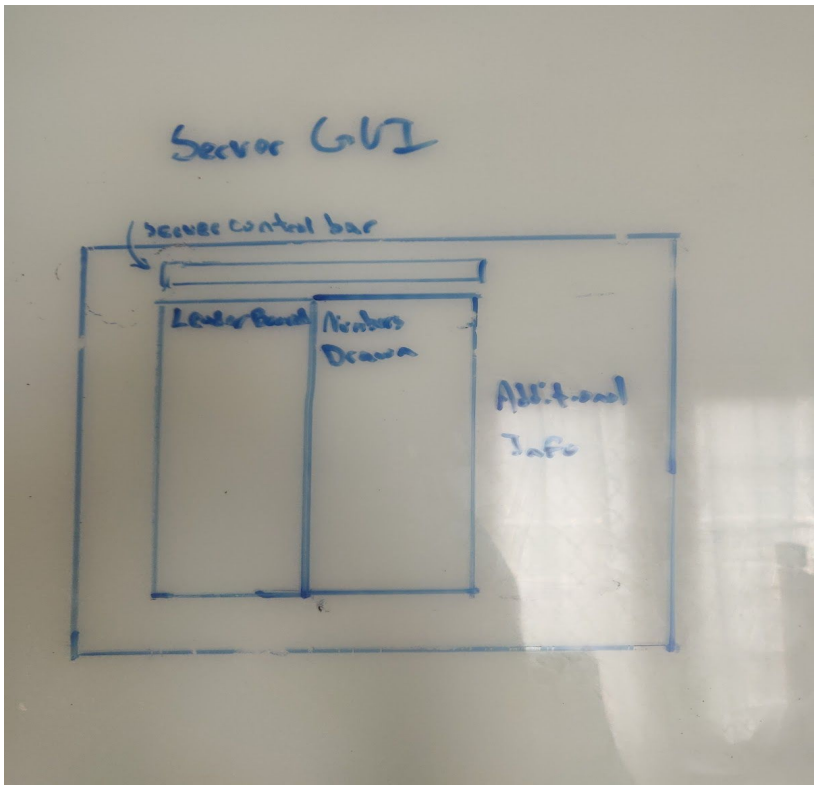
- Receives bingo sheet, as an instruction, from server (5x5).
- Drawn numbers will be sent, as an instruction.
    - It is clients job to select the corresponding number on their bingo sheet.
        - Verification method will not allow user to select number that has not been drawn.
            - Hashmap where number is key, and value is bool.
- Has the ability to see numbers that have already been drawn by pressing history button.
    - Displayed through the server message textbox.
- User Control Box
    - IP input, Port input, Connect to server button, History button, Bingo button, Leaderboard button, Exit button



GUI Idea

Server

- Sends client bingo sheet, as an instruction (5x5).
- Keeps track of numClients
    - If numClients < 4 then wait.
- Keeps track of numbers already drawn.
    - Hashmap where number is key, and value is bool.
    - Used to send history, as an instruction, when requested by client.
        - Displayed in server message box in client GUI.
- Keeps track of player wins.
    - Used to send scoreboard, as instruction, when requested by client.
        - Displayed in server message box in client GUI.
            - Allows for many players to play without having the GUI be cluttered with other players points.
- Each clientThread is listening to see if a client declares that they have bingo.
    - Client will send instruction containing their bingo sheet.
    - Server must verify that the client does in fact have bingo.
        - If client does not have bingo what are the consequences? Nothing? Points lost?
        - If a client does have bingo they will receive a point, and the game will reset.
- When a game is reset each client gets a new bingo sheet.



GUI Idea

Bingo Game

- Makes bingo sheets
- Verifies winner
    - 5 in same column
    - 5 in same row
    - 5 across either diagonal
- Keeps track of numbers already drawn.
    - Hashmap where number is key, and value is bool.
- Keeps track of players
- Keeps track of wins

Instructions list

Client to Server

- bingo row col num … row col num end
    - Allows for the server to know exactly where the players drawn number locations are to verify bingo
    - May only need bingo row col … row col end as client GUI prevents selecting invalid numbers
    - Sending the number allows us to display the drawn numbers on a bingo sheet in the server.

- history
    - Requests drawn numbers history
- leaderboard
    - Requests leaderboard information: playerIDs and points

Server to Client

- playerID ID
    - Sends player their playerID
- bingoSheet num1 num2 … num24
    - Bingo sheet will be filled in row by row sequentially, middle of sheet is free.
- gameWinner playerID points
    - Displays to server message box in client that player X has won and has Y points.
- invalidBingo
    - Alerts player they did not have bingo
- history num … num end
    - Sends all drawn numbers in the round
- leaderboard playerID points … playerID points end
    - Sends all playerIDs and their points