

Final Game

How The Game Works:

- To say it in the most simple terms the game works in multiple phases
 - 1: The beginning where all players join the lobby
 - 2: the drawing faze where to players draw based on a prompt
 - 3: the guessing faze where the remaining players guess the drawings
 - 4: results are shown to see what drawing was guessed to most accurately
- The game is run via the use of two different programs
 - The Host: the main program displayed on the projector
 - The Player: the program everyone will run on their phones
- The game's logic is done via executing different methods based on what "screen" the player sending the MQTT message is on. These "screens" represent the different fazes that the game will be on

The Screens:

- Host Screens:
 - 1: The join game screen
 - 2: The display drawing screen
 - 3: The guess the drawing screen (will tell the user to guess now + hints)
 - 4: The final results screen
- Player Screens:
 - 1: The entered name + join screen
 - 2: The please wait for the host to start the game screen
 - 3a: The draw screen (the two players chosen to draw)
 - 3b: The look at the drawing screen (the rest of the players)
 - 4: The guess screen (players try to guess the drawing)

MQTT Logic:

- The Host is going to process the messages received via this logic split with a "/"
 - 1: The person who sent it
 - 2: The intended recipient
 - 3: The current "screen" of the sender

- 4: any data that is necessary for said “screen”
- Depending on the screen the player is on the data sent will vary
 - 1: No data will be sent (the host uses the player name for this one)
 - 2: none
 - 3a: the player sends the x and y coordinates for the circle (drawing)
 - 3b: the player sends a string guessing the drawing
 - 4: none

What needs to be added:

- Phase 1:
 - QR code for the player to join with
 - A cleaner GUI for both the user and host screens
 - A List of themes and items to draw (Animals: Cow, Dog, Cat, Dolphin) either in an array or a CSV file
- Phase 2:
 - Drawings need to be displayed and updated on the host’s screen consistently so that the viewers can see what is being drawn
 - Better controls and better data-sending features for the players drawing
 - A Clean GUI for the user and the host
 - Sound for the host to play (music while drawing)
- Phase 3:
 - A method to compare the players' guess and return how accurate it was to them
 - A method to guess how many counts the user has made and a method to count how many right guesses there were for each drawer
 - GUI for said guessing faze
 - A method to determine once all players have finished their 3 guesses for each drawing and start to faze 4
- Phase 4:
 - GUI to display results
 - Small animations and sounds to go with it
 - Method to restart the game with the same players
- Other:

- Finish all the methods to send the data to and from the host
- Finish methods to process data to and from host
- Change players from a string array to an object array
- Test how often the host can get data from the players drawing so that the screen may be updated as much as possible

What I'll be doing:

- Making the methods to add players to the game, sending in their vote, calculating the score, and MQTT data received method for both host and players
- Fix the random player selection method
- Fix logic flaws in code
- Create a player class that will store all player data and methods for player actions
- If I can I'll also make the method to compare how close the guess was and return a string with a hint that will be displayed on the player's phone (complicated!)

What I need you guys to do:

- Make a clean GUI for the game (I will have something basic done probably so just make it decent if I've already made one)
- Sound design for the game (find sounds and music and make them play when appropriate)
- Fix the drawing so that inputs are more reliable and consistent
- Test how often data can be sent to the host so that drawings will be updated on the screen as much as possible
- Make the method that will take the data from the drawers and display it on the screen
- Try to organize code to be in one uniform style and communicate any issues you are running into when needed

Note: if you guys are having any difficulty with the work or feel like the workload is unbalanced please say so in the chat so that we can adjust it

Links:

<https://editor.p5js.org/nikolai.leduc/sketches/UMpl98meE>

<https://editor.p5js.org/nikolai.leduc/sketches/wSKOjJczp>

https://docs.google.com/document/d/1G_hfBFY_8oZwvXCetxOrgwWgf63iLhw7jMun_KbQmkE/edit?usp=sharing