

# ***Draw It App***

by Jamie Sasson

## **What is It?**

*Draw It* is a pictictionary style app where users draw and guess at words they have submitted to the server. At first, the user must sign in with a Username, then they must submit 5 words to the server. After which, they will either wait for at least 1 other player or join the current game. There is only one room and it can hold any number of players that the server can handle. (Don't have an estimate, but probably at least 8). After 60 seconds of drawing, where players can try to guess the word, the drawer switches and the round starts over. This goes on infinitely. If there are ever 0 players in the room, the list of words is wiped. If a user joins in the middle of a drawing round, they will be updated on the current draw. If a drawer leaves the round they are currently drawing in, the server will start a new round.

## **What are Websockets Used For?**

Is everything an answer? Websockets handle the communication between players and the update of the players drawstacks. They also handle connection and user submitted words. The full breakdown of what every call does is commented in the client.html and server.js files. A brief rundown is that the websockets and server will handle the new words, timer ticks, turn changing, player disconnection and connection, and updating the player's drawstacks.

## **What Went Right?**

The app works! I was able to get players to understand the rules easily and the most of the functions worked the first time with a few tweaks here and there. The players seem to enjoy submitting their own words as well. The project is light on communications with the server, only needing to update everyone when input is made, rather than every frame. There is a single 'every frame' update, but it is simply a timer tick and that is only every second.

## **What Went Wrong?**

I initially had the players drawing circles instead of lines in the drawstack, as I thought it made the game more entertaining and carefree. However, once I started testing on Heroku, the app started to massively slow down and even made one of my friend's computer's shut down chrome automatically because of the massive updates. I changed this to be lines and I made the draw method only call stroke() once by building a continuous line using lineTo() and moveTo() methods. Now the game doesn't slow down unless it is currently changing players, which only slows down occasionally. The other major issue was player joining and disconnecting. If they joined, they needed the current drawstack, but sometimes caused a completely new round to start. Similarly, if the drawer disconnected, no new round would start and the timer would stop going down.

## **How could I improve?**

I would like to keep track of the player's words and make sure they never received their own words as a draw or would be able to sit out of a round that was using their word. I also considered having players submit pictures along with the words, but that would be more intense. I also thought about having the server never dump the words it received and just grow infinitely with words, but this leaves the potential for duplicate words and players might never see the words they submitted.

## **Above and Beyond?**

This project could have been a pictictionary app where I pull from a database or create a list of 50 words that I thought of myself, but I wanted players to feel like they were participating more in the game. With this in mind, I let the players submit their own words for the game. It was a

simple way to make each game of *Draw It* unique while also allowing players choice in what kind of game they want to play. They could theme it to how they like. I also think my drawing method is pretty well structured, rather than making a large amount of `stroke()` calls on the canvas, I instead create multiple points and then piece them together in a for loop. The other key part here is the release component. When the player releases the mouse after a stroke, they create a 'Release' point. This point tracks when the drawstack needs to make a `moveTo()` call instead of a `lineTo()`. This lets the players "pick up the pencil", so to speak.