Drawify Web App

A shared art game with friends

Aniket Tyagi (Product Owner)
Wyatt Harris (Initial Scrum Master)
Wesley Chen (Developer)
Nathan Taylor (Developer)

Drawify Gameplay Overview

The game will alternate between phases:

Phase 1. All players are given a prompt to draw within the time limit

Phase 2. Each drawing is then sent to another random player for them to guess what the original prompt was

Phase 3. This new prompt is passed to a different player to draw

The game loops between phases 2 and 3 until all players have drawn for each thread of prompts

Phase 4. At the end of the game, each thread of prompts will be shown to all players with their corresponding drawings to see how far off the original drawing is from the final one

Drawify

Audience:

People who like drawing and playing games with friends

Who will it help:

Help people who want a simple, easy, and accessible place to hang out with their friends artistically

Why Play it:
The game would allow people to express themselves more easily, all while having a decent amount of fun

Project Scope

- Create a fun and entertaining game to play with friends centered around sharing art.
- Create a space for artistic expression through a virtual whiteboard.
- Create a fun interactable online environment within the web browser with a variety of communication and art tools.
- Create a functional database through which the game data and user communication is held
- Create an easily accessible login page to let users quickly leave and exit games

User Stories

1. Drawing

As a player, I want to use a virtual whiteboard as a medium to share and receive drawing with other players during the game

2. Player Interaction

As a player, I would like to have some form of communication with other players through some implementation of a chatbox, voice channel, and seeing other player's game drawings

3. Player Interaction

As a player I would like to have the ability to create game lobbies and choose who can and can't join, whether it be through an invite code or through a username/password system

4. Tools/UI

As a player I want interaction with the virtual whiteboard and chat box to be intuitive but also varied in features, such as multiple pen-colors, sizes, different colored chat text, etc...

5. Profiles

I want to be able to save previous games and performance and show them off in an online profile, keeping track of my progress as I play the game.

6. Peripherals (Minimum Priority)

As an artist I would like to have access to different drawing peripherals such as a pen/tablet substitution for mouse and keyboard.

7. Music/Sound Effects

As a player I want to hear a variety of engaging and entertaining sound effects that trigger when I take actions in game.

8. Online/Player Interaction

As a player I want to be able to play the game on an online platform.

Sprint 1

Sprint Goals: Learn as much as we can about our tools, set up documentation, set up github, and briefly start work on preliminary modules.

Spikes:

- Research UI and front end design style, how to organize UI modules (whiteboard, chatbox, drawing tools, etc...)
- Research JS frameworks and general usage for web app design
- Researching and understanding Firebase server functionality
- Research server infrastructure and design planning (Firebase)
- Setup GitHub repository and code sharing platform via Glitch

User Story 1:

- Create the virtual whiteboard module and test simple mouse drawing and erasing.

User Story 2:

- Create general systems module for sharing drawings from client to server and back to client

User Story 4:

- Create multitude of whiteboard module tools such as color selector, eraser sizes, etc...

Sprint 2

Sprint Goals: Begin work on setting up web app infrastructure ie: get servers running, get general website made, testing client-server communication.

User Story 1:

- Work on creating white board module export function that formats the whiteboard into the style of our server infrastructure (JSON), allowing it to be stored and shared

User Story 2:

- Create general systems module for sharing drawings from client to server and back to client

User Story 3:

- Create Firebase modules for lobby creation and connection
- Create Firebase modules for client-server connection

User Story 5:

- Create user profile module
- Create user profile data module
  - Previously drawn game
  - Previous game data
  - Average scores

Sprint 3

Sprint Goals: Begin connecting front-end and backend module for data upload and download.

User Story 2:

- Finish up UI/UX module server connections, and begin creating the general game object which encompasses all the independent modules

User Story 3:

- Add newly create game/lobby module to lobby creation and server connection module

User Story 4:

- Create chat box module
- Connect chat box module to client-server communication module
- Create user profile icons
- Create scoreboard, current turn, and current prompt icons

User Story 8:

- Connect lobby creation module to server
- Connect whiteboard module to server
- Connect server profile module to server
- Connect chatbox module to server
- Connect score, profile, and other UI display elements to server

Sprint 4

Sprint Goals: Put all independent modules together into the game object, and finish incorporating the game object to the lobby object to finish connecting to server.

User Story 2:

- Finish up UI/UX module server connections, and begin creating the general game object which encompasses all the independent modules

User Story 3:

- Add newly create game/lobby module to lobby creation and server connection module

User Story 6:

- Create peripheral connection modules to substitute mouse and keyboard input options

User Story 7:

- Implement howler.js audio library for sound effects
- Sync sound effects with game modules

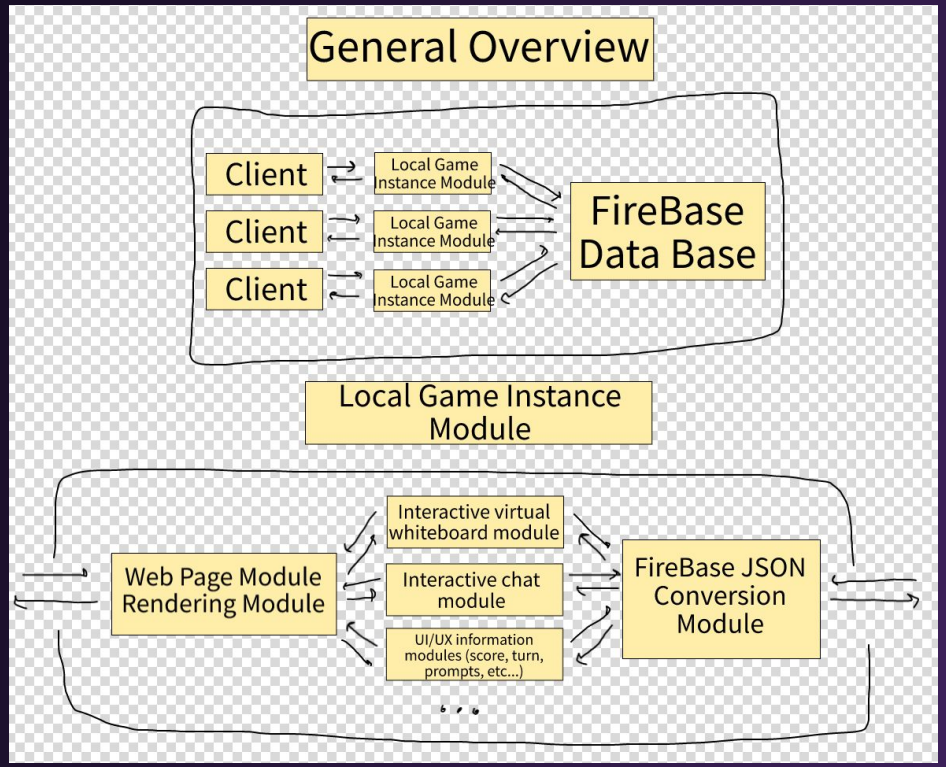
User Story 8:

- Connect lobby creation module to server
- Connect whiteboard module to server
- Connect server profile module to server
- Connect chatbox module to server
- Connect score, profile, and other UI display elements to server

Spikes:

- Research howler.js for sound effect integration

Architecture (Preliminary May Change)



Technologies

- **Javascript (including p5.js) :** the main library we'll be using to build the game's interactive modules such as the white board, chatbox, score and other dynamic UI elements.
- **React:** Javascript library that will be helpful for designing UI/UX.
- **Firebase:** A hosting service. We'll use it as our database.
- **Glitch:** A website management tool. We'll use it to build our website.
- **Howler.js:** An a library for implementing sound files and effects

Challenges / Risks

- **Learning new technology:**

  Many of us are new to many of the technologies we wish to use

- **Connecting frontend UI with backend database:**

  Lack of experience with web development on this scale

- **Communication and collaboration:**

  Using git to merge and commit changes, being sure we are all on the same page

- **Time management:**

  Many of us have other classes and/or jobs, finding times that work for all of us can be difficult

Minimum Viable Product (MVP)

Play the game: As a person who likes to have fun with my friends, I want to be able to play a complete game of Drawify from start to finish.

In order to accomplish this:

- **Drawing.** Players should be able to interact with a virtual white board that has the capability of drawing and erasing art.
- **Player interaction.** Players should be able to play online, talk with friends, and share their art.
- **Player Interaction.** Players should be able to share their drawings, profiles, previous games, and communicate online.
- **Tools/UI.** Players should be able to have access to a variety of white board drawing tools such as multi-colored pens, different pen sizes, etc... along with having the ability to interact with the chatbox dynamically.

THANKS!

That’s our general project plan!