Online Pictionary Game With Flask And SocketIO

Contents:

Analysis

Objectives

Design

Implementation

Testing

Evaluation

**Analysis**

**Introduction**

Pictionary is a popular party game in which players take turns drawing things, and other players try to guess what the person is drawing. The aim of the game is to score points (by either guessing a drawing or having your drawing guessed.) and then have the most points when the game ends.

**Problem**

For this project, I will be creating a website which players can visit to play Pictionary with their friends. I will be using Flask for the back end, along with the Flask-SocketIO library to enable use of web sockets. There will also be a log in system, which will make use of the Flask-Login library and SQL databases to store things like user passwords, usernames, high score, etc. These databases will also be used to populate leaderboards on the site.

**Research**

There are already quite a few Pictionary websites, and I decided to take a look at some for inspiration for my project.

One of the most popular Pictionary games out right now is https://skribbl.io/ .Skribblio lets players choose a username and character, then either join a random public game or make a private room for them and their friends. I liked the fact that players could make custom rooms, so I decided to implement this on my website. I didn’t feel like the avatar creation added much to the website, and I dind’t want to copy too much from it so I decided not to add it. I also decided to improve on the username aspect, as on Skribblio users choose a one-time username to play with, but I decided to add a login system which will let players create permanent accounts they can use to play the game.

Another popular Pictionary website is Drawize, which was an interesting one as I noticed that it advertised as an online, multiplayer game however upon playing a game I realized that all of the other players were bots, with very standardized names and putting unrealistic phrases in chat. I found this cheesy and thought it harmed the user experience to advertise as a multiplayer game then have all the players be bots, so I decided not to implement that into my program, even though it would be quite easy to hard code some drawings given my implementation.

**Objectives**

* Let users log in
* Users can join rooms
* Users Can draw
* Users can chat
* Users can see others’ drawings
* Users can guess drawings
* Users can play full games
* Leaderboards showing user stats
* Have the website look nice

**Design**

The website will be made with Flask on the back-end and will use the Flask-SocketIO library to enable use of websockets. Websockets allow for real-time viewing of other player’s screens, as it would be virtually impossible to have any sort of real time communications without websockets.

Players