Matthew Song

Draw Something!

Project Description:

Draw Something! will be a multiplayer drawing game made in python. Similar to skribble.io and the phone app, Draw Something, one player will draw a hidden word on the screen while others try to guess what the word is.

Competitive Analysis:

This project will be based on the popular phone game Draw Something and the online game, Skribble.io. The main elements of the game that I will create will be similar to the actual game in order to preserve the fun features that make up the game itself. However, some features will be modified to make it work with the resources I have and the conditions of the project guidelines.

Similar to both games, the game will feature multiple brush colors, sizes, and eraser tools at the drawer's needs. In addition, there will be a randomized word chosen out of a word bank that the drawer can see beforehand to get an idea of what to draw. However, unlike Draw Something, this version of the game will be live with a chatbox-type answer box. Also, unlike Skribble.io, the points and leaderboard will be round based, instead of individual point system to encourage working as a team.

Note that I have yet to see a similar project made from python Tkinter online, so all parts of the code will be planned and typed originally.

Structural Plan:

Almost all of the code will be organized and separated in the MVC animations format, because it will run CMU graphics Tkinter. Within the model, view, and controller functions, the code will be divided into pages/stages of the game, such as the start page, name choosing page, the actual drawing page, etc.

Algorithmic Plan:

At the moment, the trickiest part of the code will likely be the brush drawing itself. This part is complex because there are many things to consider while making tkinter draw functions, including overlapping, colors, brush size, etc. I will overcome this by taking advantage of app class attributes and dictionaries/2d Lists to store information in an organized matter and draw efficiently. Other complex parts of the project include file storage, file reading and writing for saved leaderboards, custom wordbanks loadable by the user, and the live chatbox for answer input as well as chatting with other users if networking is enabled.

Other than that, the other trickiest part will be networking over a server using sockets and threading if MVP status is reached (ideally). This will be done by watching the sockets mini-lecture and writing the code in a way that has all the code that needs to be sent to the server easily accessible.

Timeline Plan:

- I plan to finish the main parts of the project early before TP1 (during weekends) to ensure that I reach MVP status. This is to ensure that I am able to add networking and local multiplayer by the project due date.

Version Control Plan:

- I plan to back up the code very often on my google drive, after every session that I write code.

Module List:

- CMU 112 Graphics / Tkinter
- random
- os
- time
- *Planning After MVP: socket, threading

TP2 Update:

- After discussing with my mentor, several changes were made to the project. Firstly, to make the experience and complexity of the actual drawing better/smoother, the previous drawing code was changed to draw lines instead of circles to look smoother. In addition, another drawing mode was added that used pixels and recursion similar to the flood fill method.
- Scrolling through the chatbox feature is added
- Overal UI of the drawing board is polished with a more sophisticated size feature
- Round scores were changed to point systems to add complexity using time-based functions
- Word Complexity/difficulty is factored in the game. Word difficulty affects the points given each round and gradient descent of word difficulty per round is added.

TP3 Update:

- Added multiplayer network function for up to 4 people
- Added modules: Queue, sockets, threading