Daphne Lai / CS106 Final Project: Utinam

What is your project?

My project, Utinam, is a fast-paced game which tests your ability to maintain concentration. After generating a username, the user is then allowed to begin the game. Within 30 seconds, a single polygon will be displayed in the centered with a random amount of sides. The player's role is the correctly determine the number of sides each polygon has. The objective of the game is to successfully go through as many shapes as possible before the timer runs out.

What was your inspiration?

I am not good at many games. However, fast-paced games such as this one is one of the only types I am good at. I have had previous experience in programming with Java and Python so this time around I really wanted to explore concepts that are new to me since learning JavaScript. Therefore, I strived to make this program utilise concepts that I'm not too familiar with.

Instructions

Home

When you run the program, you will first be introduced by a home screen. The "Let's start" button allows you to proceed to the next page. The "Information" button will send you to a vague explanation of what's to come along with a "Back" button on the bottom right. The "Back" button will return you to the home page.

Username Generator

After clicking "Let's start", you will be sent to a username generator. There'll be a "Back" button to return to home and a "PLAY" button that will not be active until you've picked a name for yourself. At the center of the page, you can type in whatever combination of characters you want and then hit the "Generate!" button.

The username will be made base on the following:

1. The name is only five characters long.
2. The first and last character you typed will be first and last in the new name respectively.
3. Any letters in between the first and last will be converted into designated numbers.
4. If there are less than five letters, there will be zero's that will make sure the username is five characters.
5. Symbols (ie. !@#$ etc.) are exceptions to rules 3 and 4.

Now you can "PLAY".

Game Play

You'll see a toggling polygon in the middle with a random amount of sides/points. At the top there's a purple bar decreasing in length. This is a timer that will run out in 30 seconds. On the top left, your username and score will be displayed. At the top right, there'll be a button "HOME" that will send you back to the home page. Below the shape, there's an input box for you to place your answer in. In order to submit the answer, you must hit the enter key on your keyboard. If you guessed the correct amount of sides/points, a different polygon will appear. Otherwise, you're stuck staring at the same shape for the remainder of the round. Try your best not to press enter in the last second or two. The reason will be explained in the next part.

After Game Play

After 30 seconds is up, you'll be prompted to save a file named "top_scores.json" this is a file that externally saves the scores from previous games. Make sure not to spontaneously hit enter at the last second of the round or else you'll be saving a file in who-knows-where. Save the file in FinalProject/Utinam/data and replace the file "top_scores.json" that's already in the data folder. You'll be shown the top five scores from previous games shown from oldest to newest. Then you can hit the "HOME" button to play again.

Topics

Module 2: Containers

When generating the username, I had to parse, manipulate and generate new arrays/strings.

Module 4: Advanced Shapes

In order to draw the polygons, I had to use vertex(), beginShape(), endShape(CLOSE) and the polar() function.

Module 5: User Interfaces and the DOM

The whole program interface is using DOM elements.

Module 8: Randomness and Noise

I used random() to determine how many vertices the next polygon will have and its' color. The noise() function was used to toggle the shape in relation to time.

Module 3: Input and Output / Module 11: JSON

Input and Output was used to load the Information page text and the previous top high scores. The high scores were then processed at the end of the game and are exported as a json file.