

Project3 Post-Mortem

Paul Plew

March 30, 2021

Post-Mortem

The goal of this project was to create a trivia game with at least 10 questions. The game had to have animations throughout and look clean, structured, and organized. In addition to those visual requirements, on the technical side of things, I was required to implement inheritance from a parent class to at least two sub-classes. And the questions had to be loaded in from an external file (either `JSON` or `XML` formatted).

My final code that I submitted included all of these things. In 3 different javascript files using `p5.js` I implemented the trivia game and the environment around it. I first began by creating a `Shape` class that was the parent class to 5 different subclasses. The `Shape` class was extended by different classes that represented 4 different shapes and a `Button` class. Inside of the super class there was a method for updating the shapes (size and rotation), as well as a simple collision detection method. Although the collision detection only detected a rectangle around the shape it was enough to achieve the functionality required. The 4 different shape classes were used to draw a background with interactive shapes floating in space, and the `Button` class was used to create the start button, and different visual elements.

In another `p5.js` file I created a `World` class. The purpose of this class was to deal with the current state of the world and all the transitions between the questions and states. The `World` class handled all the button clicks and loading the questions from the `JSON` file. There were 4 different states in the game.

The start state of the game held the start button as well as the title. When the button was clicked the game moved into the next state, the play state. The play state printed the questions one by one, if the question was answered correctly a boolean representing the correctness of the previous question was set to `true`. This boolean was used to congratulate the user if they answered a question correctly. After the question was answered the state was switched to an answered state. In this state the user was congratulated based on their performance on the previously answered question, and some additional information was shown so the user could learn a little bit more. When the mouse was clicked if there were additional questions then the game would show the next question, otherwise the final ending state would be shown. In the end state the game showed the user how many points they got out of the total possible points. Each correctly answered question awarded the user 100 points.

Issues

Thinking back using the Class name **Button** was not the best choice because in reality the class was used for more than just buttons. Originally the class was created to draw the title screen start button, but over time it grew into a class that represented the question background and other items as well. In a future version I would change the name of the class to something more descriptive to better reveal its purpose.

Another problem that arose during my process was the issue of fading each screen in and out. In the final iteration of the game I used two booleans globally to determine if the game was fading in or fading out and manipulated the variables accordingly. However, this led to a long **if**, **if else**, and **else** statement in my **draw** function. To improve, and make the code more human readable, I should have stored and manipulated these variables in the World class.

Creative Process

My creative process for this project was relatively linear. I started out by creating the background shapes, and setting the visual mood of the game. After I was happy with the background and the color selection I got to work on the game itself. Creating the different elements of the game was difficult. The most difficult part was setting the next question without going to the end state. When I was satisfied with the gameplay I updated my template **JSON** file to show the final questions. After that all that was left was for me to make the final touches. My favorite part of this project was the function I wrote to convert a base 10 number to a base 16 number to manipulate the transparencies of the different game elements.