Hue Are You by Emerson Thomas and Johana Leaños

Goal

Personality quizzes help us understand ourselves better, giving concrete names and shapes to our abstract characteristics. But more importantly, they can also provide a few moments of distraction and entertainment in a crazy world. Hue Are You takes users on a journey to self-discovery, asking questions and generating their unique inner color.

Our main goal for the project was to make a quiz that would be more fun and visually-appealing than an ordinary personality quiz. In general most online quizzes or surveys are static and do not engage the user. As a result, the task can become boring for users, especially while taking longer quizzes. By animating the experience and allowing the user to interact with the environment as they fill out answers, as if they were going on a journey, we can keep their attention, and then provide a moment of satisfaction at the end.

Our project is a personality quiz meant more for entertainment than for utility, but the design could be applied to digital marketing and education for more real-world impact. Our final implementation poses a series of questions to the user, and these questions are inspired by fantasy stories but also narrow down the bounds of the final color. As the user clicks their response, the camera follows a path to the next question, and finally to their result.

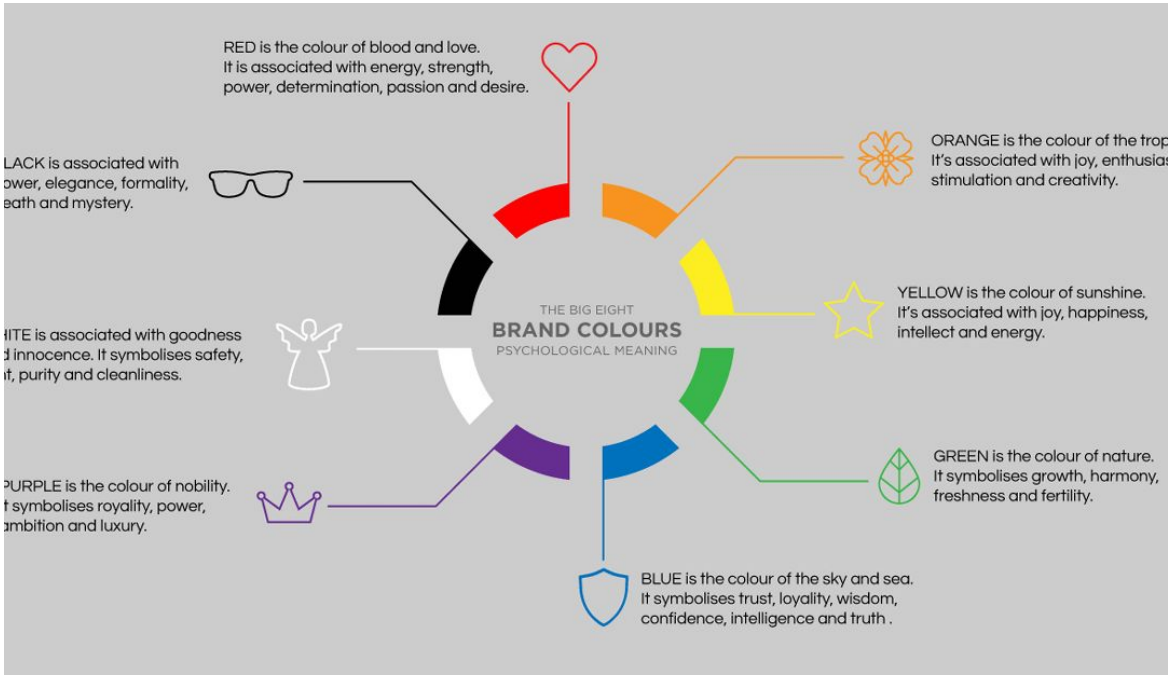
Previous Work

Most online personality quizzes or quizzes in general are composed of a series of questions, where the user selects an answer then either scrolls further on or clicks to view the next question. Oftentimes the quiz maker will focus on a finite set of traits or similarities and produce one of those options as a result for the user at the end. Obviously there are many personality quizzes online, but we were inspired by the tests featured on the Wizarding World of Harry Potter/ Pottermore[1] page. In particular we enjoyed the design of the quiz to discover your patronus, which takes the user on a journey through a dark forest and they must answer questions based on their gut reactions to specific scenarios. This type of design is unique because the user essentially navigates the background scene to encounter questions instead of merely answering them. But even though this quiz featured killer graphics, the user is unable to retake it after their initial result, nor can they go back and appreciate the graphics. Additionally, it's a private project so we had no idea how exactly they rendered the project. Surprisingly we could not find other examples of this type of quiz that we wanted to create, or at least one that was open source.

Once we decided on a style, next we had to come up with a topic for the personality quiz. We chose to focus on generating a single RGB color for the user, since it's simple, visual, and can be determined by a small amount of numeric variables. Many of the example color

[1] https://my.wizardingworld.com/sorting

quizzes we found online were trying to target some other internal internal characteristic merely classified based on color. For example, the Color Code[2] personality quiz distinguishes between people with different motivations by assigning colors Red, Blue, Yellow, or White. The quiz merely uses the colors to code the underlying motivation qualities. We thought that it would make sense to have some sort of theory behind out results, as was also noted from peer feedback, but we wanted the quiz to be more entertaining than informative. But at the same time we didn't want it to be a "what is your favorite color" type of quiz. In the end we created our questions to target traits associated with colors in marketing. Here's the reference image[3]:



As you can see, the color classifications here are also concrete, but we were always planning to have some elements of randomness to keep it interesting.

Approach

Our initial approach for the project was as follows:

- Work with the starter code, get used to loading and creating objects and moving the camera and research potentially useful ways of doing so
- Create an MVP using a single scene and background. Each click would take the user from one question to the next. Add an instructions page and end result (random for now).
- Create the color-finding algorithm and associated questions, and add more elaborate settings for each one to fit with the questions.

[2] https://www.colorcode.com/about/

[3] https://az.design/blog/graphic-design-tips-2-colour-fonts-text/

- Add other desired features such as a dynamic background, more complicated meshes, different question configurations.

Methodology

Text and Questions

Surprisingly, from our research the main way to render text in a javascript project are to create a material or texture with the text or a TextGeometry through Three.js. Alternatively we could have added HTML, but we wanted the text to be more integrated into the scene. Ultimately we found that this was not the easiest mesh to work with for such a simple task. It requires that you load a typeface font using their loader, and each string must have its own geometry. Additionally due to our way of recording answers we couldn't merge the meshes. Overall we spend a lot more time making and moving the text than we anticipated.

Here is the list of questions:

1. You hear howl of a dangerous monster in the distance. What do you do?
    a. Take off before it's too late
    b. Pause and make a plan.
2. Some friends meet you on the road. Who do you bring along for the journey?
    a. Friend with more resources.
    b. Friend with more talent.
3. Along the way you must train. Which course do you select?
    a. Receive sword, study defensive spells.
    b. Receive shield, study offensive spells.
4. As you grow closer, voices scream your worst nightmares. What do you hear?
    a. That you will never do good.
    b. That you will never have power.

Originally we were hoping to add more questions and paths through the scene, but we ended up prioritizing the transitions and lighting. Additionally most of the values had to be hardcoded, so it would have been perhaps more effort than it was worth given that we were content with the color results we had. We did add in each question to appear once the previous question is answered, and for the material of the answer to change upon choosing it.

Background and Lights

We originally had the idea of creating a sunrise, with glowing stars, moon, and a realistically rising sun. The first drawback was the inability to create a background that could support multiple colors at the same time. The second was the creating a mesh moon and stars wouldn't fit the lighting changes we wanted to implement as well as be too expensive. Instead, we decided to play with the lighting and give the illusion of a sun/moon with DirectionalLight from three.js above the mountain and adjust the illumination of the scenery with HemisphereLight and AmbientLight as well as change the background color

accordingly. Using Tween.js[4], 7 sets of background and lighting colors in arrays synchronically transitioned linearly based on the question being answered. For the starting scene, the background is a dark blue with stars[5] rendered using PointGeometry for its light weight. The stars' unrealistic look also fit our scenery aesthetic than mesh stars would have. The DirectionalLight is set to a pale blue to give the illusion of a moon, AmbientLight to dark blue, and Hemisphere to black, creating a night scene with a full moon. The next couple scenes switch the lights and background to an indigo, berry (purple/red), and then light orange. The second to last change switches the background to pale blue and the lighting to a soft yellow, resembling early morning. In the last transition, the stars disappear and the background is sky blue with pale yellow/ white lights to resemble midday. Even if the DirectionalLight itself didn't move to give the illusion of a rising sun, the play on colors with the other lights and the continuous movement of the camera gave a sufficient illusion of a rising sun to create the appropriate atmosphere.

Scenery

We started out wanting to create a scenery that consisted of many small individual settings all wrapped in a forest with many small interactive elements, like fire and water. We soon realized that if we were to build those components, we wouldn't have enough time to adjust other things in the scene, like lighting and text. Instead, we chose to find some prebuilt scenes that matched with our narrative and aesthetic. The mountain scene[6] we settled on gave us a good variety of objects, like the lake and cabin, that we could use as settings for our questions. And it was free. We thought about adding multiple scene islands to encompass the narrative better but the cost of rendering the intricate scenes didn't seem worth the aesthetics. Unfortunately we didn't have enough time to render interactive elements that the user could click on during the exploration stage. As for including the questions, we used Raycasting to enable the user to click on their response and activate the Tween transitions for the camera, lights, and background. The integration of the text with the scene enhanced the Pottermore patronus quiz aesthetics that we wanted to follow.

Camera

We wanted the camera to serve two different purposes, depending on what stage of the project the user was on. For the first part, we wanted the camera to be fixed to specific locations and directions so that it would follow a set path between the questions. After the questions, the user would be able to explore the scene with their arrow keys and mouse. For the first part, the controls were unlocked from PointerLockControls so that the user could use their mouse to click on the responses to the questions. The camera and look at positions are fixed into arrays and transitioned using Tween.js. The animation of the transitions coincides with what type of movement is happening (e.g. "launching" up, "falling" down) and takes advantage of Tween's Easing method. Originally, we had issues with the camera lookAt position and had to face the mountain at all the transitions. Once

[4] https://github.com/tweenjs/
[5] https://redstapler.co/space-warp-background-effect-three-js/
[6] EdwinRC, https://skfb.ly/6CvOW

we crafted the questions to go around the mountain and the OrbitControls were removed, the camera was freed and we decided to create an array from the lookAt vectors, enhancing the variety of positions we were able to create.

Once the quiz portion is complete, the user is launched to the ground level and is able to explore the terrain using their arrow keys and mouse. We initially started to use event listeners and manual camera control but then we found the controls that Three.js provided. This approach was better because we wouldn't have to reinvent the wheel when it came to gaming controls. At first I tried to integrate the controls with OrbitControls but they didn't provide a satisfactory control over the camera. Then we found the PointerLockControls. Our controls are modified from the three.js PointerLockControls example[7] so that there is no movement in the y direction and there is a significant damping to the speed at which the user can move. The only limitation from this example is the creation of boundaries so that users won't explore off the island. When trying to implement boundaries, the user was either too limited in range to return to the island or the camera flew too far from the island to return. Since the scene is one cohesive piece, creating object avoiding logic would be too cumbersome, tedious, and limiting for the users given the small size and tight composition of the scene.

Color Algorithm

First, we decided to use the question answers to find the user's custom color in HSL coordinates. That way three of the questions would focus on narrowing down one single value for Hue, and then one question would establish the Lightness. First we divide the Hue [0-1.0] spectrum evenly into 8 sections ([0 - 0.125], [0.125 - 0.25] ...) In general each section is dominated by a single color or tone of colors. Then to determine the range we use binary search based on the user's responses. Each response is recorded based on the object id of the Text Geometry they click. The first question divides the spectrum between the warmer hues [0.75:1, 0-0.25], and cooler hues [0.25-,0.75], based on presumed temperament.Then the next two questions make it so that the more similar answers also yield similar color categories, like answering B-A-A and B-A-B both yield relative shades of blue. From there we generate a random number in the range. For Lightness a random value between 0.25 - 0.5 or 0.5 - 0.75 is generated based on the last question, which judges morality. Originally we were picking Lightness across the spectrum, but we chose to narrow it down to the more bright colors in the middle. The saturation value is always 1.0. Then we used the functions from Assignment 1 to convert between HSL and RGB, and display a shape (currently a box) and the numeric H value with that color. The Appendix shows the approximate gradients possible.

Results

When evaluating our end product, we measured how much progress we did by how much we learned while implementing it. On the camera and color side, we learned about the wide variety of libraries available

[7] https://github.com/mrdoob/three.js/blob/m

for calculating increments of a value, like Tween and PointerLockOrbit, that allowed for higher level manipulation and higher quality transitions between our colors and camera locations. We experimented with various webpacks implementations in order to render both the text and background adequately. We also experimented with lighting a lot in order to find out which combinations of colors, intensities, and range gave the best results for our desired aesthetics. Based on our results, we believe that we did a successful job at combining a wide variety of elements to create a project that is adaptable to multiple settings, especially that of education for a younger audience.

Discussion

Overall we certainly picked an interesting approach for this project, and we hope to see more similar concepts in the future. Given our limitations in time and experience, we challenged ourselves while also making something we actually enjoyed. At the end of the day it's clearly a lot easier to code a simple test with HTML and CSS or something, but with this project there is something special and magical about the journey to the result. With that said with more questions and perhaps psychology education someone else could add more meaning to the result. We'd be interested to see someone design the animations for this type of project more artistically using Blender, or another Mesh for 3D text which is easier to use and more efficient.

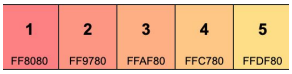
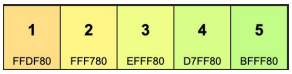
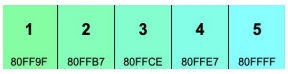
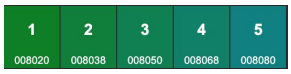
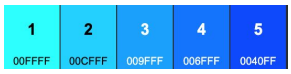
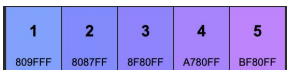
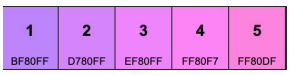
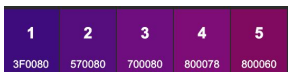
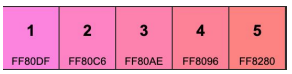
Conclusion

We think that given the time constraints of the project, we create a fun and aesthetically pleasing project that adequately approximates our original intention. While the final product works well, we wish we could have had more time to further develop our ideas and expand on this product. Given another week, we would have implemented more interactive elements at the end and more quiz paths. Yet, for the experience in researching and developing, this project has made us proud.

We pledge our honor this is our own work within university regulation.

Emerson Thomas, Johana Leaños

Appendix

H	L = 0.25	L = 0.5	L = 0.75
[0 - 0.125] [A B A]			
[0.125 - 0.25] [A B B]			
[0.25 - 0.375] [B B B]			
[0.375 - 0.5] [B B A]			
[0.5 - 0.625] [B A A]			
[0.625 - 0.75] [B A B]			
[0.75 - 0.875] [A A B]			
[0.875 - 1.0] [A A A]			

[8]

Licensing for the nature scene:

```
"author": "EdwinRC (https://sketchfab.com/Edwin3D)",
```

[8]Gradients generated from: http://www.perbang.dk/rgbgradient/

```
"license": "CC-BY-4.0 (http://creativecommons.org/licenses/by/4.0/)",

"source": "https://sketchfab.com/models/493b488007f14e3b98502ccc60f78582",

"title": "Low Poly Nature"
```