Project X

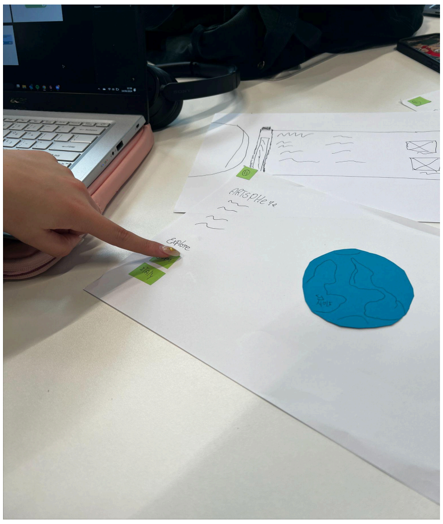
For my last project this semester, I had the freedom to choose any topic within my study path. I chose to develop a webpage using HTML, CSS, and JavaScript because I want to deepen my knowledge in development. I created a website that showcases the diverse art cultures from different countries and regions around the world.

LO:Interactive Media Products

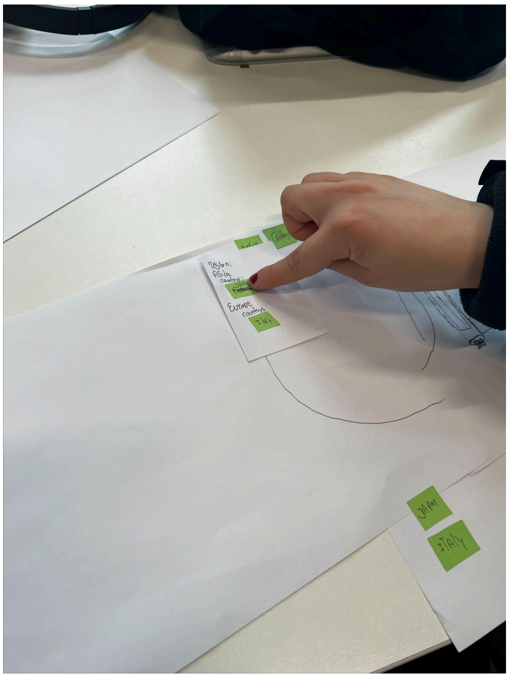
What i did:

For this project, I decided to conduct user testing for my website using paper prototypes. My goal was to evaluate the clarity of navigation and button placements. To achieve this, I tested with three users and assigned each a set of tasks ***(see Figure 1)***. The tasks included navigating to the Japan page, moving from the Japan page to another country's page, and finally returning to the home page. After each task, I asked the participants specific questions to gather detailed feedback. This approach allowed me to understand the users' experiences and identify areas for improvement based on their interactions and responses.

Figure 1: User testing



In this example, one of the participants is clicking the Japan button, which takes them to the page dedicated to Japan.



Here you can see one of the participants going to another country page from the explore tab in the navbar.

What i learned:

I learned how to conduct user testing with paper prototypes to determine optimal navigation and design for the website. This process provided valuable insights into user interactions and preferences, helping me identify areas for improvement. By observing how users navigate and complete tasks, I gained a better understanding of their needs and challenges, which will inform my future design and development decisions. This experience emphasized the importance of iterative testing and feedback in creating user-friendly websites.

How I think it went:

I believe it went well. From the user testing, I received feedback that the navigation and page layout were very clear, so I did not have to make significant changes. This positive feedback confirmed that the initial design effectively met user needs."

LO:Development and vision Control

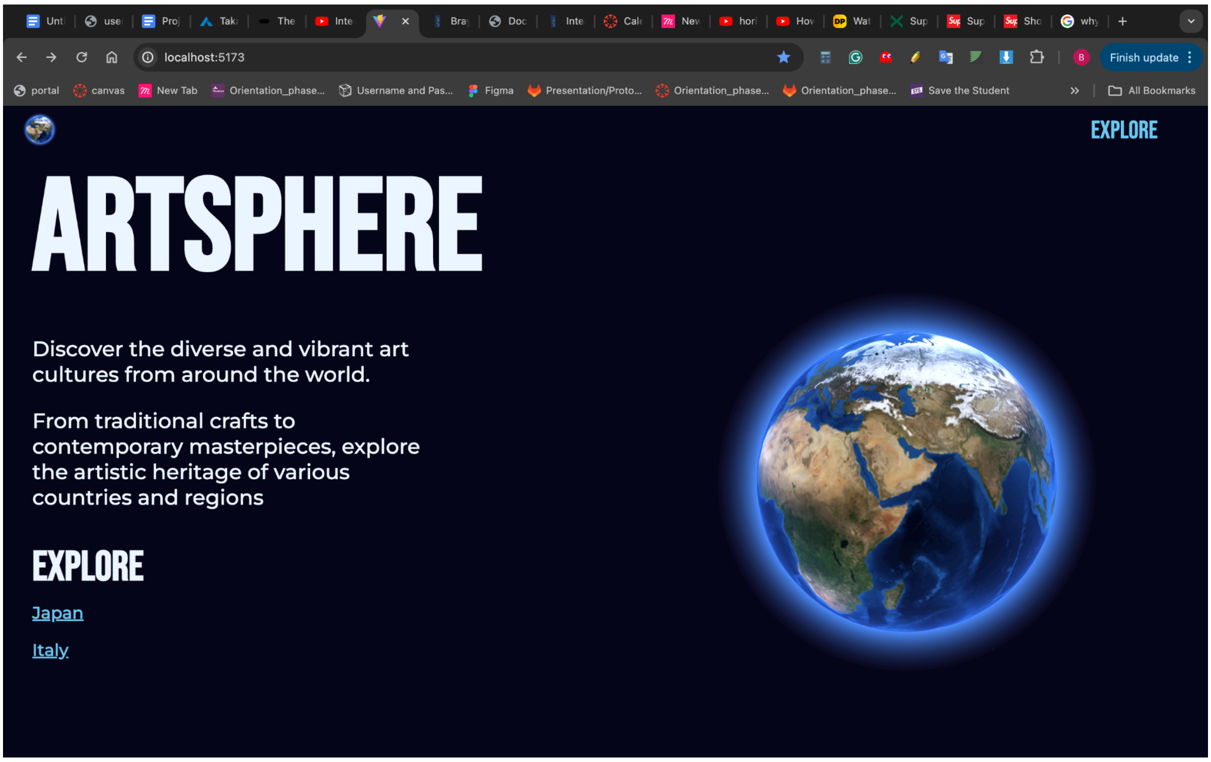
What i did:

I developed a website showcasing different art cultures from around the world. For this website, I decided to add a 3D globe on the homepage that spins and moves with the cursor ***(see Figure 2).*** For the development, I used JavaScript, Node.js, Three.js, GSAP, and Vite.

I installed Node.js to manage dependencies, allowing me to install Three.js, Vite, and GSAP. With Three.js, I created a 3D sphere, setting up a scene and adding a camera to render it on the webpage. I applied shaders and textures to the sphere to make it look smooth and realistic like Earth. The shaders were defined in a GLSL file, which I imported into the JavaScript code for use ***(see Figure 3).***

Using GSAP, I added mouse movement functionality, enabling the 3D globe to follow the cursor and rotate along the X and Y axes. I used Vite as a local server to host the site, facilitating development and testing.

Figure 2: Website



EXPLORE

ARTSPHERE

Discover the diverse and vibrant art cultures from around the world.

From traditional crafts to contemporary masterpieces, explore the artistic heritage of various countries and regions

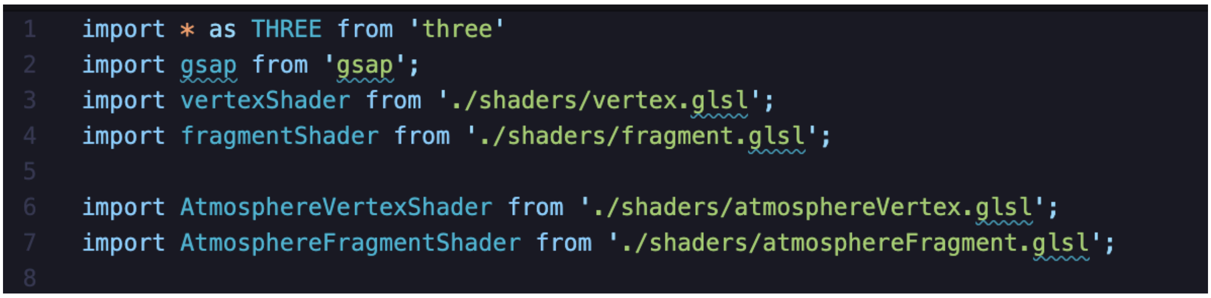
EXPLORE

Japan

Italy

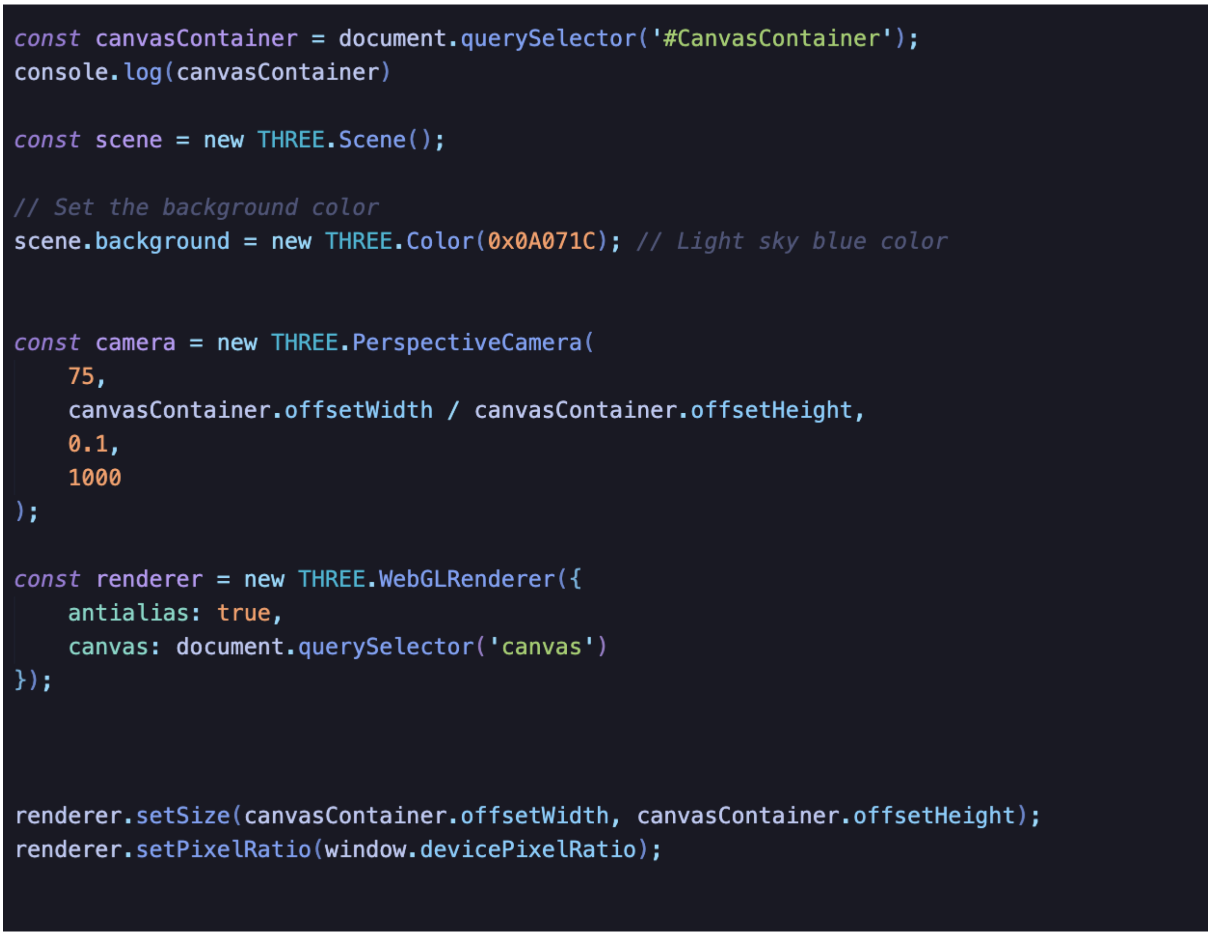
Here you can see the home page of the website with the 3D globe made with three js.

Figure 3: Codes



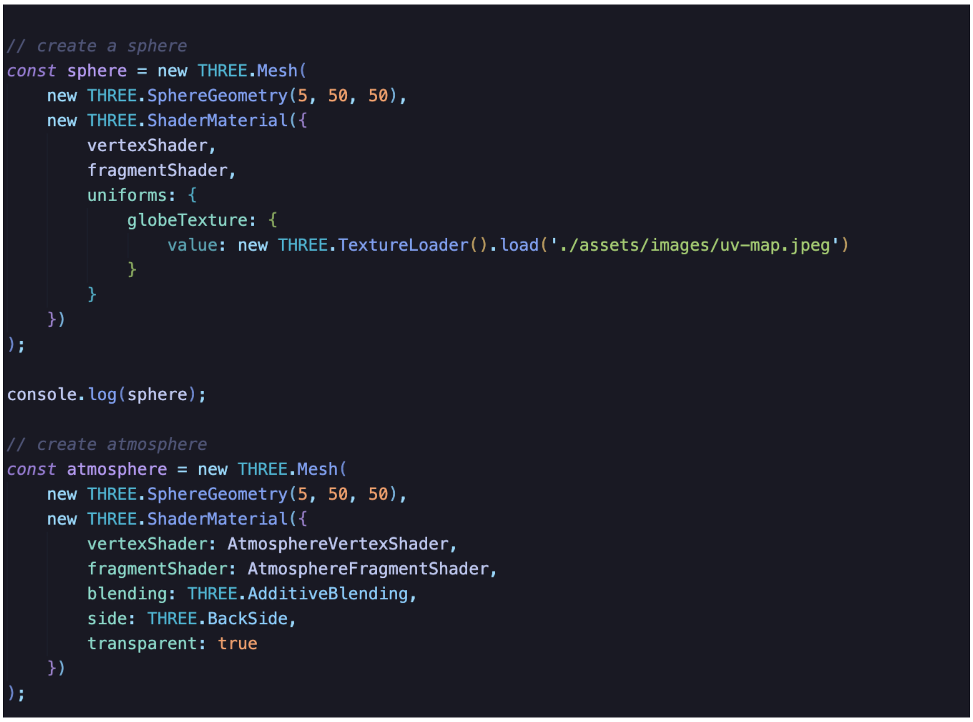
```
import * as THREE from 'three'
import gsap from 'gsap';
import vertexShader from './shaders/vertex.glsl';
import fragmentShader from './shaders/fragment.glsl';

import AtmosphereVertexShader from './shaders/atmosphereVertex.glsl';
import AtmosphereFragmentShader from './shaders/atmosphereFragment.glsl';

```

In Here i am imported Three.js, gsap and the glsl files for the shaders into javascript, so I can work with them.



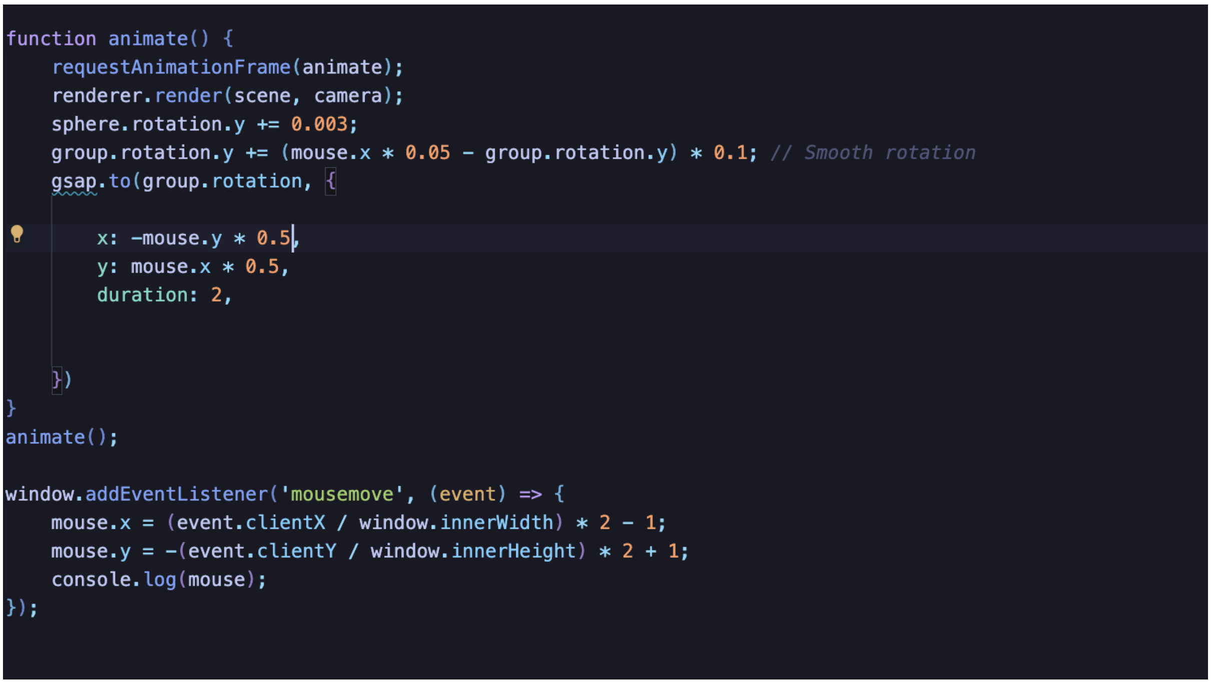
```javascript
const canvasContainer = document.querySelector('#CanvasContainer');
console.log(canvasContainer)

const scene = new THREE.Scene();

// Set the background color
scene.background = new THREE.Color(0x0A071C); // Light sky blue color


const camera = new THREE.PerspectiveCamera(
    75,
    canvasContainer.offsetWidth / canvasContainer.offsetHeight,
    0.1,
    1000
);

const renderer = new THREE.WebGLRenderer({
    antialias: true,
    canvas: document.querySelector('canvas')
});



renderer.setSize(canvasContainer.offsetWidth, canvasContainer.offsetHeight);
renderer.setPixelRatio(window.devicePixelRatio);
```

Here i created a new THREE.js scene where i set the background color and a camara with perspective to render the Globe. I also put the height and with of the canvas where it renders to adjust to the with and height of the canvas in html.



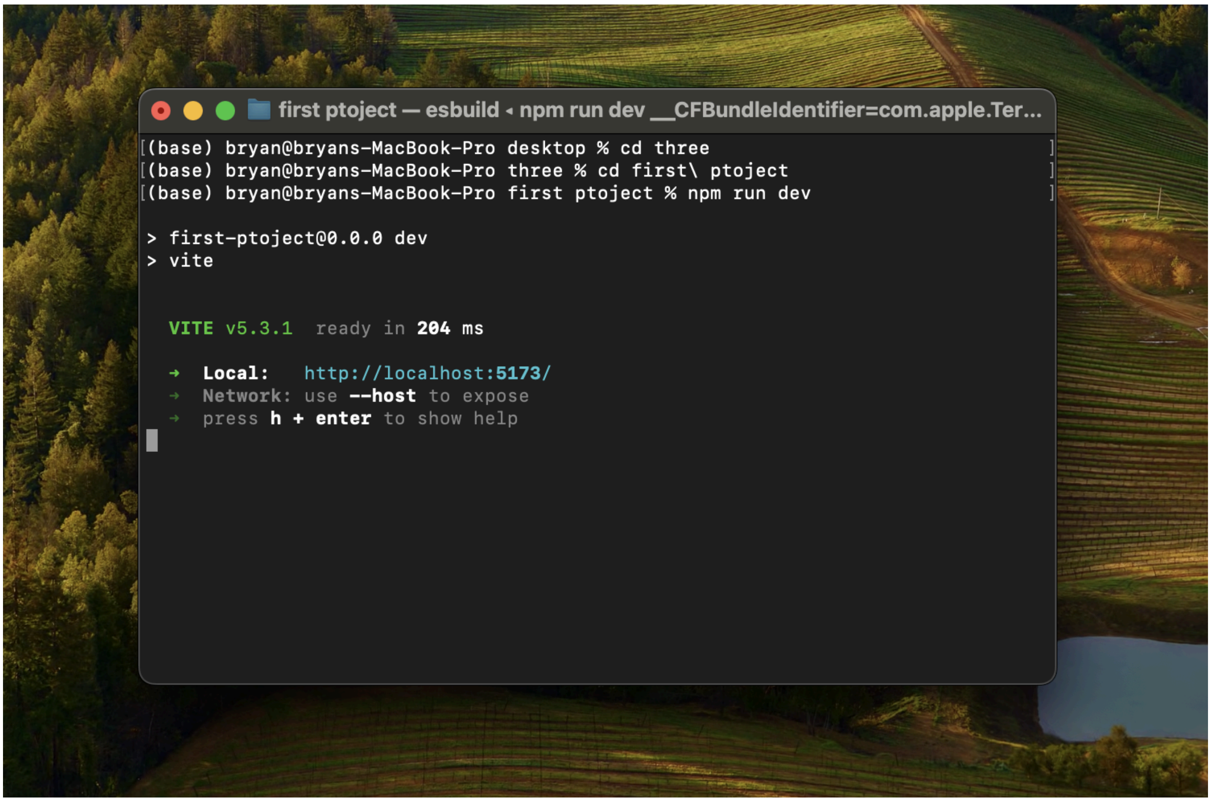
```
// create a sphere
const sphere = new THREE.Mesh(
    new THREE.SphereGeometry(5, 50, 50),
    new THREE.ShaderMaterial({
        vertexShader,
        fragmentShader,
        uniforms: {
            globeTexture: {
                value: new THREE.TextureLoader().load('./assets/images/uv-map.jpeg')
            }
        }
    })
);

console.log(sphere);

// create atmosphere
const atmosphere = new THREE.Mesh(
    new THREE.SphereGeometry(5, 50, 50),
    new THREE.ShaderMaterial({
        vertexShader: AtmosphereVertexShader,
        fragmentShader: AtmosphereFragmentShader,
        blending: THREE.AdditiveBlending,
        side: THREE.BackSide,
        transparent: true
    })
);
```

Here i created the sphere and atmoshopere, and added shaders and textures.



```
function animate() {
    requestAnimationFrame(animate);
    renderer.render(scene, camera);
    sphere.rotation.y += 0.003;
    group.rotation.y += (mouse.x * 0.05 - group.rotation.y) * 0.1; // Smooth rotation
    gsap.to(group.rotation, {

        x: -mouse.y * 0.5,
        y: mouse.x * 0.5,
        duration: 2,


    })
}
animate();

window.addEventListener('mousemove', (event) => {
    mouse.x = (event.clientX / window.innerWidth) * 2 - 1;
    mouse.y = -(event.clientY / window.innerHeight) * 2 + 1;
    console.log(mouse);
});
```

Here i set a function to animate the 3D globe to spin and to follow the mouse cursor with GSAP.



first ptoject — esbuild ◂ npm run dev __CFBundleIdentifier=com.apple.Ter...

```
[(base) bryan@bryans-MacBook-Pro desktop % cd three
[(base) bryan@bryans-MacBook-Pro three % cd first\ ptoject
[(base) bryan@bryans-MacBook-Pro first ptoject % npm run dev

> first-ptoject@0.0.0 dev
> vite


  VITE v5.3.1  ready in 204 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

Here i am running the host with VITE.

What i learned:

I learned how to work with Three.js, including setting up a scene and a camera, rendering, and modifying scene and camera values. I also learned how to add textures, colors, and shaders to objects using Three.js. This involved creating GLSL files for shaders, setting their values, and integrating them into the JavaScript code to enhance the visual quality of the 3D objects.

Additionally, I gained experience with Vite for hosting a website locally, which streamlined the development process with its fast build times and hot module replacement features. This hands-on experience provided valuable insights into using modern development tools and techniques, and it helped me understand how to effectively combine various technologies to create interactive and visually appealing web applications.

How I think it went:

I believe it went pretty well. I successfully created the 3D globe for the website, which I was keen to accomplish. It was my first time working with Three.js, and I found it really interesting. I am now invested in learning more and creating additional projects with Three.js and 3D objects in websites.