Floating Gift

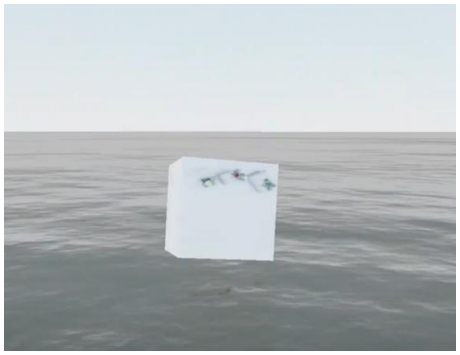
Project name:Floating Gift
Creator:Jiaxin Liang
Student ID:22018177

Introduction

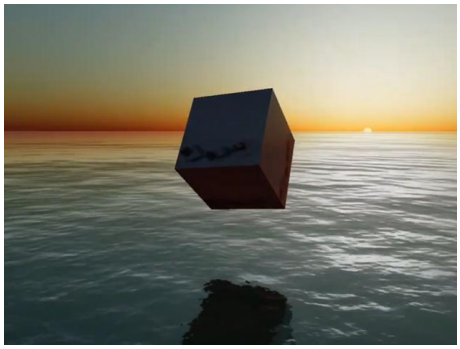
This is my final project for the course Coding one,This project shows a gift block that floats and dives between the sky and the ocean.For a detailed demonstration, you can watch this video:https://youtu.be/pKzm2jIj_ZE
In this project, you can change the color of the sky (elevation), the position of the sun (azimuth), the flow of the water (distortonscale and size) by dragging the buttons in the background music.

Visual

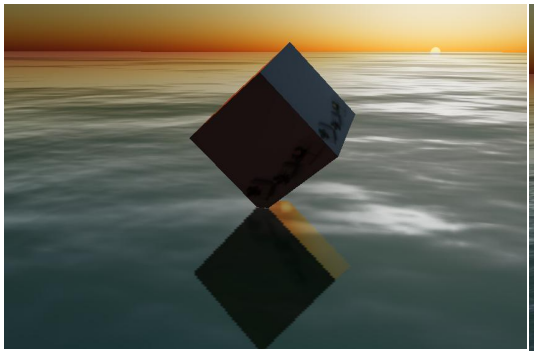
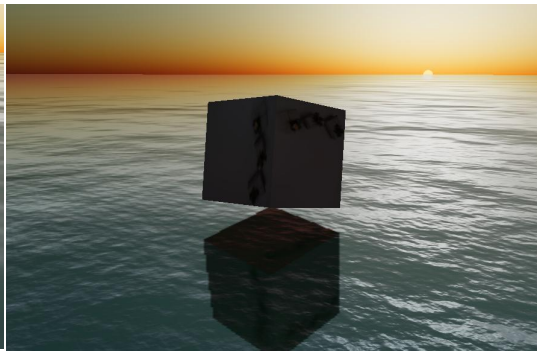
At this time, when the elevation slider is pulled to the far right, there will be a snow-white sky. I want to create a feeling of high-altitude snow.I want to make people feel a cool and slightly lonely feeling.



On the contrary, if the slider is pulled to the far left, the beautiful sky with the setting sun will appear. I want to express the hot temperature of the setting sun.He is completely different from the cube in the previous picture. I want him to be hot and warm, and hope that his enthusiasm can infect everyone who sees him.



The ripples on the water surface can also be changed, pulling it back and forth can give people a sparkling feeling.And the change of the wave will also make the block look big (right picture) and sometimes small (left picture).

Process

The sunflower block of week 7 and the beautiful sunset and ocean scenes in https://threejs.org/ inspired me to make a block that rotates between the sky and the ocean. I refer to a lot of materials to understand how to make a beautiful sky and a beautiful ocean. It is too difficult. I not only use the links mentioned in the course, but also find solutions on bilibili and youtube. I made a lot of mistakes in the middle of the process, tried other software besides mimic, experienced a long time of not working and black screen, I thought I was going crazy, but I persisted. I have to say that the material library is really great, which allows us to get the effect we want more quickly. In my opinion, the overall result is good.

I used three.js to make this project better for me.There are many teachings about three.js on the Internet, and there are a large number of material libraries available for me to use, which is very convenient!
First I created the scene object

【scene = new THREE.Scene();
camera = new THREE.PerspectiveCamera(55, window.innerWidth / window.innerHeight, 1, 20000);

camera.position.set(30, 30, 100);】

Then create a plane, which is the water（I try to make it as big as possible）

【const waterGeometry = new THREE.PlaneGeometry(10000, 10000);
water = new Water(
waterGeometry,
{
textureWidth: 512,
textureHeight: 512,
waterNormals: new THREE.TextureLoader().load('img/waternormals.jpg', function (texture) {
texture.wrapS = texture.wrapT = THREE.RepeatWrapping;
}),
sunDirection: new THREE.Vector3(),
sunColor: 0xffffff,
waterColor: 0x001e0f,
distortionScale: 3.7, }
);
】

and the sky

【const sky = new Sky();
sky.scale.setScalar(10000);
scene.add(sky);
const skyUniforms = sky.material.uniforms;

skyUniforms['turbidity'].value = 10;
skyUniforms['rayleigh'].value = 2;
skyUniforms['mieCoefficient'].value = 0.005;
skyUniforms['mieDirectionalG'].value = 0.8;

const parameters = {
elevation: 2,
azimuth: 180
};

const pmremGenerator = new THREE.PMREMGenerator(renderer);
let renderTarget; 】

a cube

【const textureLoader = new THREE.TextureLoader();】

Find materials and dress them up：

【const texture = textureLoader.load('img/home.jpg', function () {
doRender();
});
const geometry = new THREE.BoxGeometry(20, 20, 20);
const material = new THREE.MeshStandardMaterial({ roughness: 0 ,map: texture});
mesh = new THREE.Mesh(geometry, material);
scene.add(mesh);

controls = new OrbitControls(camera, renderer.domElement);
controls.maxPolarAngle = Math.PI * 0.495;
controls.target.set(0, 10, 0);
controls.minDistance = 40.0;
controls.maxDistance = 200.0;
controls.update();

stats = new Stats();】

To make the project look more beautiful, I added the renderer here

【

renderer = new THREE.WebGLRenderer();

renderer.setPixelRatio(window.devicePixelRatio);

renderer.setSize(window.innerWidth, window.innerHeight);

renderer.toneMapping = THREE.ACESFilmicToneMapping;

container.appendChild(renderer.domElement); 】

and there are many more detailed processes in the files I uploaded.

Conclusion

This course made me know a lot of useful software, and I also realized how difficult it is to create a mimic from the beginning. I am very honored to be able to participate in this course. Although the process is very painful, I saw The final result is really rewarding! Hope to make more beautiful works!