UNIVERSIDAD NACIONAL DE SAN AGUSTÍN

Facultad de Ingeniería de Producción y Servicios

Ciencia de la Computación

Laboratorio 05

ALUMNOS:
Pfuturi Huisa, Oscar David
Quispe Menor, Hermogenes
Quiñonez Lopez, Efrain German
Fernandez Mamani, Brayan Gino
Santos Apaza, Yordy Williams

DOCENTE:
MSc. Vicente Machaca Arceda

CURSO:
Computación Gráfica

6 de mayo de 2021

Índice

1. Github

- https://github.com/oscar-pfuturi-h/Comp-Grafica/tree/main/practica_04

2. Sistema Solar

2.1. Clases Sphere y Planeta

Se creo un una clase sphere y una clase planeta que es una subclase para facilitar la creación del sistema sin mucho código, en su constructor se indican los parámetros de los como radio, posición, texturas, etc.

```
class Sphere {
    constructor(radius, alpha, theta, x, y, z, textures) {
        this.r = radius;
        this.a = alpha;
        this.t = theta;
        this.x = x;
        this.y = y;
        this.z = z;
        this.geometry = new THREE.SphereGeometry(this.r, this.a, this.t);
        this.setTexture(textures);
        this.mesh.position.set(this.x, this.y, this.z);
    }
    set position(newposition) {
        this.mesh.position.x = newposition[0];
        this.mesh.position.y = newposition[1];
        this.mesh.position.z = newposition[2];
    }
    setTexture(textures) {
    // creamos el material con una(s) textura(s)
    }
}

class Planet extends Sphere {
    constructor(radius, alpha, theta, x, y, z, textures, name, orbitRate, rotationRate, nMoons, distance) {
        super(radius, alpha, theta, x, y, z, textures);
        this.name = name;
        this.orbitRate = orbitRate;
        this.rotationRate = rotationRate;
        this.nMoons = nMoons;
        this.distance = distance;
    }
    move(){}
    moveMoon(planetPosition){}
}
```

2.2. Creación de Planetas

Se creo una función para facilitar creación de planetas.

```
function setPlanetData(myOrbitRate, myRotationRate, myXDistance,
   myName, myTexture, myRadius, mySegments) {
    return {
        orbitRate: myOrbitRate,
        rotationRate: myRotationRate,
        distance: myXDistance,
        name: myName,
        texture: myTexture,
        size: myRadius,
        segments: mySegments
    };
}
```

2.3. Movimiento de Planetas y lunas

Se creo una función dentro de la clase planeta para mover los planetas, el movimiento de las lunas es un función de la clase planeta para facilitar el acceso a los datos.

```
move() {
    var time = Date.now();
    this.x = -Math.cos(time * (1.0 / (this.orbitRate * orbitData.value
   )) + 10.0) * this.distance;
    this.z = Math.sin(time * (1.0 / (this.orbitRate * orbitData.value)
   ) + 10.0) * this.distance;
    this.mesh.position.set(this.x, 0, this.z);
    this.mesh.rotation.y += this.rotationRate;
}
moveMoon(planetPosition) {
    this.move();
    this.mesh.position.x = this.mesh.position.x + planetPosition.x;
    this.mesh.position.z = this.mesh.position.z + planetPosition.z;
}
```

2.4. Creación de anillos para planetas

También se tiene una función para agregar anillos a los planetas que los necesiten como Saturno, se usó las formas primitivas de three.js como ring.

```
function getRing(size, innerDiameter, segments, myColor, name, x) {
    var ringGeometry = new THREE.RingGeometry(size, innerDiameter,
   segments);
    var ringMaterial = new THREE.MeshBasicMaterial({color: myColor,
   side: THREE.DoubleSide});
    var myRing = new THREE.Mesh(ringGeometry, ringMaterial);
    myRing.name = name;
    myRing.position.set(x, 0, 0);
```

```
    myRing.rotation.x = Math.PI / 2;
    scene.add(myRing);
    return myRing;
}
```

2.5. Asignación de tiempos de rotación y traslación

Se asigna todos los datos de los planetas para la creación, que fueron definidos en el constructor de la clase planeta.

```
var mercury = new Planet(1.2, 12, 12, sunRadius + 29.0, 0, 0, ['
   textures/mercury.jpg', 'textures/mercurybump.jpg', null],
                         'Mercurio', 58.64, 0.005, 0, sunRadius + 29.0)
   ;
```

2.6. Creación de orbitas

Se crearon orbitas de los planetas para una mejor guía del movimiento de los planetas.

```
function createVisibleOrbits(distance, segments, name) {
    var orbitWidth = 0.05;
    planetOrbit = getRing(
        distance + orbitWidth,
        distance - orbitWidth,
        segments,
        0xffffff,
        name,
        0);
}
```

2.7. Fondo

Se inserto una foto panorámica del universo y se uso como fondo con función de movimiento.

```
const loader = new THREE.TextureLoader();
const texture = loader.load(
  'textures/universo_4k.jpg',
  () => {
    const rt = new THREE.WebGLCubeRenderTarget(texture.image.height);
    rt.fromEquirectangularTexture(renderer, texture);
    scene.background = rt.texture;
  });
```

3. Resultados

Se creo un servidor local y se probo los resultados, siendo todo satisfactorio, desde las texturas, fondo y movimiento de los planetas.

Figura 1: Planetas

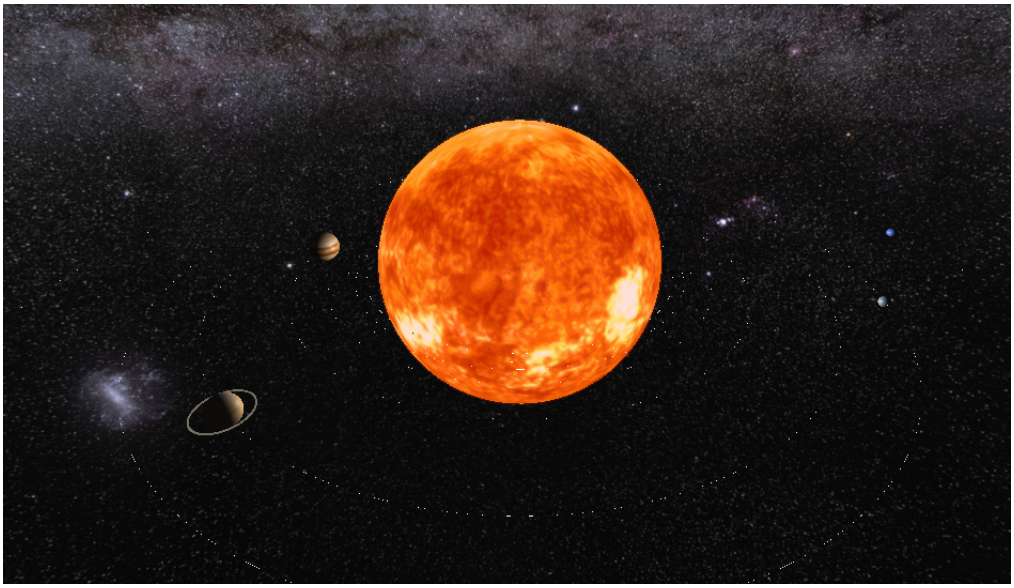

Figura 2: Sistema solar

Referencias

[1] Three.js – JavaScript 3D library. (s. f.). Recuperado 6 de mayo de 2021, de https://threejs.org/