

SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 8

Data 29.05.2022

Temat: Podstawy Three.js

Natalia Pierzchała

Informatyka I stopień,

stacjonarne,

4 semestr,

Gr. 2a

1. Polecenie:

Celem jest konstruowanie złożonego modelu za pomocą three.js - animowanej karuzeli (podstawa karuzeli jest wielokątem odpowiednio z konfiguracją zadania) i co najmniej jednego innego wybranego modelu (patrz Fig.). Pliki do pobrania znajdują się poniżej. Głównym plikiem jest lab9.html. Podfolder zasobów resources zawiera dwa pliki JavaScript używane przez program oraz model konia, którego używamy w karuzeli. Zawiera również kilka plików graficznych, które można wykorzystać jako tekstury

2. Wprowadzane dane i wykorzystane komendy:

```
function createWorld() {  
    renderer.setClearColor("black"); // Background color for scene.  
  
    scene = new THREE.Scene();  
  
    // ----- Make a camera with viewpoint light  
    -----  
    camera = new THREE.PerspectiveCamera(30, canvas.width / canvas.height,  
0.1, 100);  
  
    camera.position.z = 30;  
  
    var light; // A light shining from the direction of the camera; moves  
with the camera.  
  
    light = new THREE.DirectionalLight();  
    light.position.set(0, 0, 1);  
    camera.add(light);  
    scene.add(camera);  
  
    //----- Create the scene's visible objects  
    -----  
    roof = new THREE.Mesh(  
        new THREE.CylinderGeometry(0.1, 11, 3, 10, 25),  
        new THREE.MeshPhongMaterial({  
            color: "violet",  
        })  
    );  
  
    roof.position.y = 6.2;  
    scene.add(roof);  
  
    var geometry = new THREE.SphereGeometry(2.2, 25, 25);  
    var material = new THREE.MeshBasicMaterial({  
        map: new THREE.TextureLoader().load('https://encrypted-  
tbn0.gstatic.com/images?q=tbn%3AAND9GcS9Uv3AxsQE88_-z0Njxqephw8NMTi8odMGRbQPH-m9-  
fsfjde&usqp=CAU')  
    });  
}
```

```

var sphere = new THREE.Mesh(geometry, material);
sphere.position.y = 2.2;
scene.add(sphere);
var floor = new THREE.Mesh(
    new THREE.CylinderGeometry(11, 11, 0.3, 10, 1),
    new THREE.MeshPhongMaterial({
        color: "violet"
    })
);
floor.rotation.y = Math.PI / n;
scene.add(floor);
var newMaterial;
var loader = new THREE.GLTFLoader();
for (let i = 0; i < n; i++) {

horses.add(loader.load('https://threejs.org/examples/models/gltf/Horse.glb',
function(horse) {

    let hScene = horse.scene;
    hScene.scale.multiplyScalar(0.015);
    hScene.position.x = R * Math.cos((2 * Math.PI * i) / n);
    hScene.position.y = 1;
    hScene.position.z = R * Math.sin((2 * Math.PI * i) / n);
    hScene.rotation.y = -(2 * Math.PI * i) / n;
    hScene.traverse((o) => {
        if (o.isMesh) o.material.emissive = new
THREE.Color("#FFFFFF");
    });
    scene.add(hScene);
}));

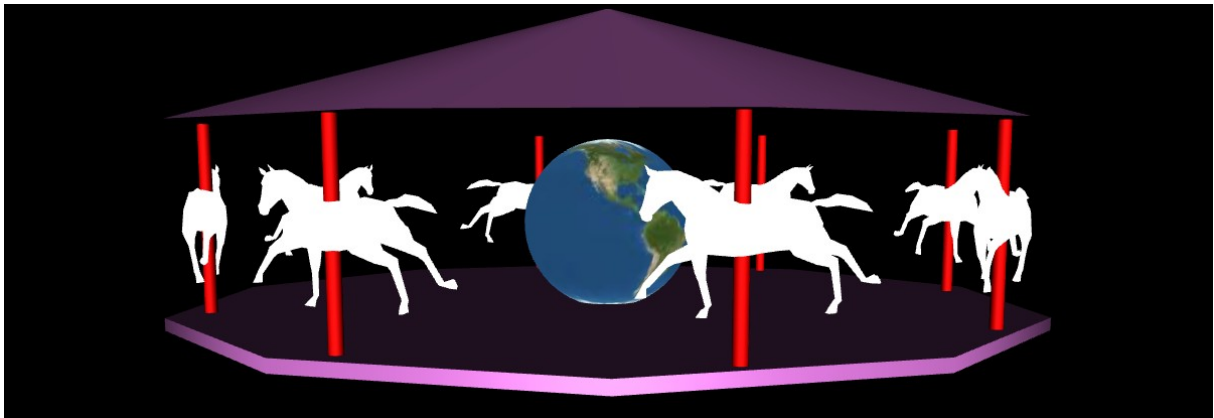
pole = new THREE.Mesh(
    new THREE.CylinderGeometry(0.15, 0.15, 4.9, 30, 1),
    new THREE.MeshPhongMaterial({
        color: "red",
        shininess: 8
    })
);

```

```
pole.position.x = R * Math.cos((2 * Math.PI * i) / n);  
pole.position.y = 2.3;  
pole.position.z = R * Math.sin((2 * Math.PI * i) / n);  
scene.add(pole);  
}  
}
```

<https://github.com/99lucky8/Grafika-komputerowa.git>

3. Wyniki działania:



4. Wnioski:

Ćwiczenie pokazuje nam, że możemy stworzyć złożony model z zaprojektowanych wcześniej elementów.