# SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Laboratorium** 9

**Data** 23.03.2022

**Temat**: „Podstawy Three.js”

**Wariant** -

Wojciech Biegun

Informatyka I stopień,

stacjonarne,

4 semestr,

Gr. 2b

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.

3. Wykorzystane komendy:

a) Użyte funkcje:

THREE.Mesh()

THREE.ConeGeometry()

THREE.MeshPhongMaterial()

THREE.CylinderGeometry()

THREE.SphereGeometry()

scene.add()

loader.load()

b) Kod źródłowy:

```
var bottomCylinder,topCylinder,cone,earth,
    horses=new Array(6);
 earth = new THREE.Mesh(
        new THREE.SphereGeometry(3,32,16),
        new THREE.MeshLambertMaterial({
            color: "white",
            map: new THREE.TextureLoader().load("resources/earth.jpg")
        })
    )
    scene.add(earth);

    cone = new THREE.Mesh(
        new THREE.ConeGeometry(7,2,48),
        new THREE.MeshPhongMaterial({
            color: 0x66BB33,
            specular: 0x222222,
            shininess: 16,
            shading: THREE.FlatShading
        })
    );
    cone.position.y = 4.4;
    scene.add(cone);

    topCylinder = new THREE.Mesh(
        new THREE.CylinderGeometry(7,7,0.2,48),
        new THREE.MeshPhongMaterial({
```

```
            color: 0x66BB33,
            specular: 0x222222,
            shininess: 16,
            shading: THREE.FlatShading
        })
    )
    topCylinder.position.y=3.3;
    scene.add(topCylinder);

    bottomCylinder = topCylinder.clone();
    bottomCylinder.position.y = -3.2;
    scene.add(bottomCylinder);
    var stick1 = new THREE.Mesh(
        new THREE.CylinderGeometry(0.2,0.2,6.7,48),
        new THREE.MeshPhongMaterial({
            color: 0xCCBB33,
            specular: 0x222222,
            shininess: 16,
            shading: THREE.FlatShading
        })
    )
    stick1.position.y = 3.3;
    stick1.position.x = -3.5;
    stick1.position.z = 4.5;
    bottomCylinder.add(stick1);
    var stick2 = stick1.clone();
    stick2.position.x = -6;
    stick2.position.z = 0;
    bottomCylinder.add(stick2);
    var stick3 = stick1.clone();
    stick3.position.x = -3.5;
    stick3.position.z = -4.5;
    bottomCylinder.add(stick3);
    var stick4 = stick1.clone();
    stick4.position.x = 6;
    stick4.position.z = 0;
    bottomCylinder.add(stick4);
    var stick5 = stick1.clone();
    stick5.position.x = 3.5;
    stick5.position.z = -4.5;
   bottomCylinder.add(stick5);
    var stick6 = stick1.clone();
    stick6.position.x = 3.5;
    stick6.position.z = 4.5;
    bottomCylinder.add(stick6);
/* load Horse */
const loader = new THREE.OBJLoader();
loader.load(
    'resources/horse.obj',
    function ( object ) {
        /* HORSE 1 */
```

```
        horses[0] = object.clone()
        stick1.add( horses[0] )
        horses[0].rotation.x=-1.5
        horses[0].rotation.z=1
        horses[0].position.y=-2.4
        horses[0].position.z=0.5
        horses[0].position.x=1
        horses[0].scale.set(0.25,0.25,0.25)
        /* HORSE 2 */
        horses[1] = horses[0].clone()
        horses[1].rotation.z=0
        horses[1].position.x=0
        horses[1].position.y+=0.3
        stick2.add(horses[1])
        /* HORSE 3 */
        horses[2] = horses[0].clone()
        stick3.add(horses[2])
        horses[2].rotation.z=-1
        horses[2].position.x=-1;
        horses[2].position.z=0.3
        horses[2].position.y+=0.6
        /* HORSE 4 */
        horses[3] = horses[0].clone()
        stick4.add(horses[3])
        horses[3].rotation.z=-3
        horses[3].position.x=0
        horses[3].position.z=-1
        horses[3].position.y+=0.9
        /* HORSE 5 */
        horses[4] = horses[0].clone()
        stick5.add(horses[4])
        horses[4].rotation.z=-2
        horses[4].position.x=-1
        horses[4].position.z=-0.7
        horses[4].position.y+=1.2
        /* HORSE 6 */
        horses[5] = horses[0].clone()
        stick6.add(horses[5])
        horses[5].rotation.z=-4
        horses[5].position.x=0.5
        horses[5].position.z=-0.4
        horses[5].position.y+=1.5
    },
    // called when loading is in progresses
    function ( xhr ) {console.log( ( xhr.loaded / xhr.total * 100 ) + '% loaded'
);},
    // called when loading has errors
    function ( error ) {console.log( 'An error happened' );});
/* Handling animation and making horses move up and down */
let isHorseGoingUp =[true,true,true,true,true,true];
function updateForFrame() {
```

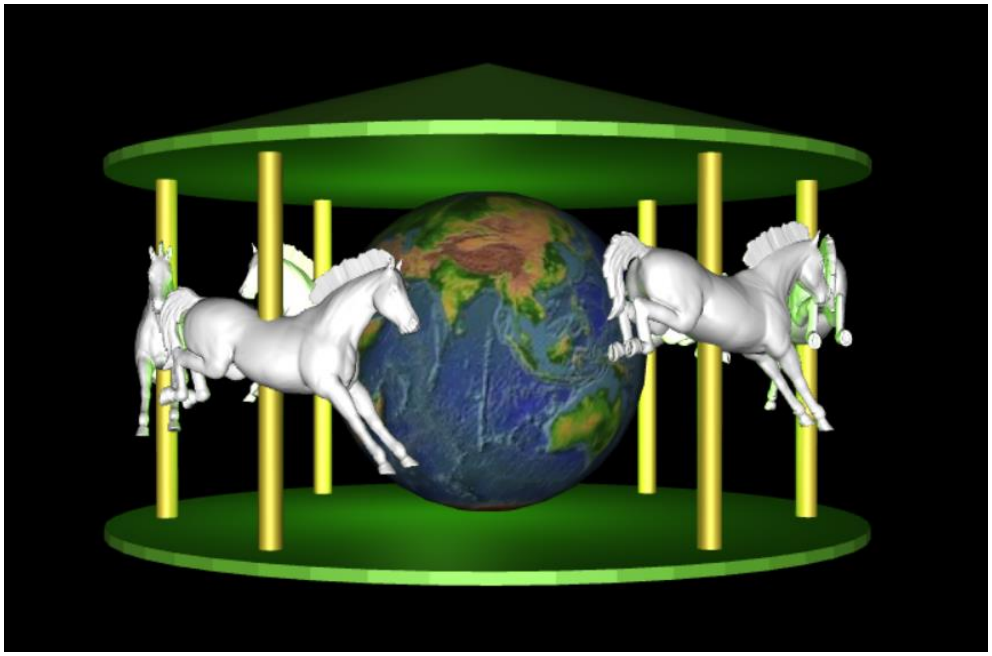
```
    console.log(horses[0].position.y)
    var loopFrame = frameNumber % 240;
    if (loopFrame > 120) {
        loopFrame = 240 - loopFrame;
    }
    var scaleFactor = 1 + loopFrame/120;

    bottomCylinder.rotation.y += 0.01;
    topCylinder.rotation.y += 0.01;
    earth.rotation.y += 0.01;
    let counter = 0;
    for(let horse of horses){
        if(horse.position.y > -1){
            isHorseGoingUp[counter] = false;
        }else if(horse.position.y < -3){
            isHorseGoingUp[counter] = true;
        }
        if(isHorseGoingUp[counter]) horse.position.y += 0.01;
        else horse.position.y -= 0.01;
        counter++;
    }
}
```

Link do zdalnego repozytorium:

https://github.com/WojciechBiegun/GK

Zadanie zostało umieszczone w folderze three.

4. Wynik działania:



Wnioski:

Biblioteka THREE.js jest użytecznym narzędziem pozwalającym tworzyć grafikę 3D. Pozwala ona umieszczać na scenie proste oraz bardziej skomplikowane obiekty 3D i dokonywać ich animacji. Jest ona zdecydowanie przyjemniejsza w użytkowaniu niż omawiana wcześniej biblioteka OpenGL.