**SPRAWOZDANIE**

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Laboratorium**

26.04

**Temat:** Konstruowanie obiektów w Three.js

**Wariant: Wieża**

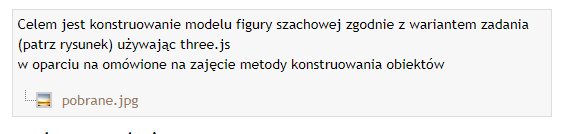
Mateusz Łysoń

Informatyka I stopień,

niestacjonarne,

4 semestr, Gr. 1a

1. **Polecenie**



1. **Wprowadzane dane**

Wieżę wykonano poprzez stworzenie dwóch podstaw, cylindra podłużnego i skopiowanej podstawy mniejszej, na którą odpowiednio naniosłem 5 kwadratów dla imitacji blanek wieży. Kolor uzyskałem poprzez zmianę materiału.

1. **Wykorzystane komendy:**

1. kod źródłowy

**let canvas = document.getElementById("glcanvas");**

**const scene = new THREE.Scene({color: 0xfff});**

**const camera = new THREE.PerspectiveCamera(100, window.innerWidth / window.innerHeight, 1, 1000);**

**const renderer = new THREE.WebGLRenderer({antialias: true, alpha: true});**

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

**document.body.appendChild(renderer.domElement);**

**var light; // A light shining from the direction of the camera; moves with the camera.**

**light = new THREE.DirectionalLight();**

**light.position.set(-10, -2, 1);**

**camera.add(light);**

**scene.add(camera);**

**const material = new THREE.MeshPhongMaterial(**

**{**

**color: 0xFFFFFF,**

**});**

**const GeometriaBazy = new THREE.CylinderGeometry(1, 1, 0.15, 100);**

**const baza = new THREE.Mesh(GeometriaBazy, material);**

**const GeometriaDrugiejBazy = new THREE.CylinderGeometry(0.9, 0.9, 0.25, 100);**

**const DrugaBaza = new THREE.Mesh(GeometriaDrugiejBazy, material);**

**DrugaBaza.position.y = 0.1;**

**let group = new THREE.Group();**

**group.add(baza);**

**group.add(DrugaBaza);**

**group.position.set(0, -4, 0);**

**group.scale.set(2, 2, 1);**

**let points = [];**

**for (var i = 0; i < 7; i++) {**

**points.push(new THREE.Vector2(Math.tan(i \* 0.1)\*3+6 , (i - 4.5) \* 4));**

**}**

**let size = 0.2;**

**let geometry1 = new THREE.LatheGeometry(points);**

**let lathe1 = new THREE.Mesh(geometry1, material);**

**lathe1.position.set(0, -1.7, -0.3);**

**lathe1.scale.set(size, -size, -size);**

**const ringGeometry = new THREE.CylinderGeometry(1.5, 1.5, 0.3, 100);**

**const ring = new THREE.Mesh(ringGeometry, material);**

**ring.position.set(0, 1.3, 0);**

**const ringGeometry2 = new THREE.CylinderGeometry(1.5, 1.5, 0.3, 100);**

**material.side = THREE.DoubleSide;**

**const ball = new THREE.Mesh(ringGeometry2, material);**

**ball.position.set(0, 0.8, 0);**

**ball.scale.set(1,-1,-1);**

**let head = new THREE.Group();**

**head.add( ball);**

**head.position.set(0,1.2,0);**

**let chessPawn = new THREE.Group();**

**chessPawn.add(group, lathe1, head);**

**scene.add(chessPawn);**

**///////////////////**

**const geometry = new THREE.BoxGeometry( 0.8, 0.8, 1 );**

**const cube = new THREE.Mesh( geometry, material );**

**cube.position.set(0,2.5,0.93);**

**scene.add( cube );**

**const geometry2 = new THREE.BoxGeometry( 0.5, 1, 0.3 );**

**const cube2 = new THREE.Mesh( geometry2, material );**

**cube2.position.set(0.8,2.5,0.9);**

**scene.add( cube2 );**

**const geometry3 = new THREE.BoxGeometry( 0.5, 1, 0.3 );**

**const cube3 = new THREE.Mesh( geometry3, material );**

**cube3.position.set(-0.8,2.5,0.9);**

**scene.add( cube3 );**

**const geometry4 = new THREE.BoxGeometry( 0.2, 1, 0.05 );**

**const cube4 = new THREE.Mesh( geometry4, material );**

**cube4.position.set(1.3,2.5,0.50);**

**scene.add( cube4 );**

**const geometry5 = new THREE.BoxGeometry( 0.2, 1, 0.05 );**

**const cube5 = new THREE.Mesh( geometry5, material );**

**cube5.position.set(-1.3,2.5,0.50);**

**scene.add( cube5 );**

**///////////**

**function animate() {**

**requestAnimationFrame(animate);**

**renderer.render(scene, camera);**

**}**

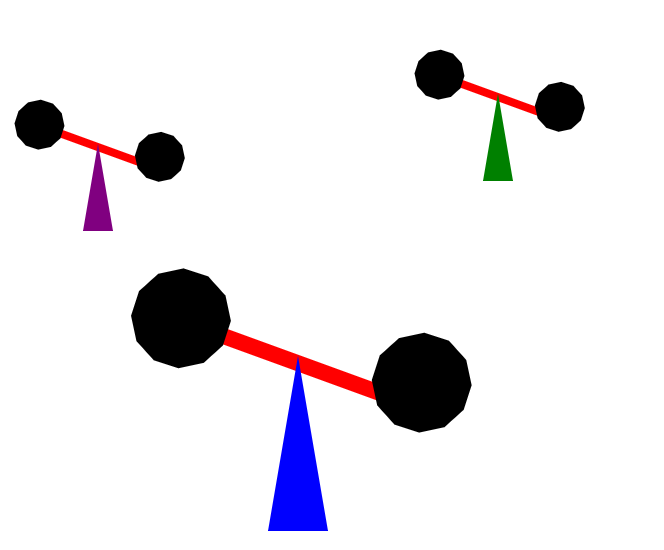
**camera.position.z = 6;**

**camera.position.y = 1.2;**

**animate();**

Link do zdalnego repozytorium (np. GitHub): https://github.com/sensorbtf/GrafikaKomputerowa

1. **Wynik działania:**



1. **Wnioski:**​ Na podstawie otrzymanego wyniku można stwierdzić, że wykorzystując XML możemy stworzyć animacje dowolnego typu i wpływać na ich działanie.