**SPRAWOZDANIE**

Zajęcia: Grafika komputerowa

Prowadzący: mgr Mikołaj Grygiel

**Laboratorium 10**

23.06.2024

**Temat:** Konstruowanie obiektów z użyciem Three.js

**Wariant**: 1

Radosław Skrzypczyński

Informatyka I stopień,

niestacjonarne,

4 semestr,

Gr. 1B

1. **Polecenie:**

Celem jest konstruowanie modelu figury szachowej zgodnie z wariantem zadania (patrz fig. 3) używając three.js w oparciu na omówione na zajęciach metody konstruowania obiektów.

****

**2. Wprowadzane dane:**

Ruchy myszką w celu obrotu figury

**3. Wykorzystane komendy:**

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<meta charset="UTF-8">*

*<title>Three.js Cone with Sphere and Disks</title>*

*<script src="https://cdn.jsdelivr.net/npm/three@0.115/build/three.js"></script>*

*<script src="https://cdn.jsdelivr.net/npm/three@0.115/examples/js/controls/OrbitControls.js"></script>*

*<script>*

*"use strict";*

*let scene, camera, renderer, controls;*

*function init() {*

*// Create scene*

*scene = new THREE.Scene();*

*// Create camera*

*camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);*

*camera.position.set(0, 5, 10);*

*// Create renderer*

*renderer = new THREE.WebGLRenderer();*

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

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

*// Create controls*

*controls = new THREE.OrbitControls(camera, renderer.domElement);*

*// Add light*

*const light = new THREE.PointLight(0xffffff, 1, 100);*

*light.position.set(10, 10, 10);*

*scene.add(light);*

*const ambientLight = new THREE.AmbientLight(0x404040);*

*scene.add(ambientLight);*

*// Create base plane*

*const planeGeometry = new THREE.PlaneGeometry(20, 20);*

*const planeMaterial = new THREE.MeshPhongMaterial({ color: 0x808080, side: THREE.DoubleSide });*

*const plane = new THREE.Mesh(planeGeometry, planeMaterial);*

*plane.rotation.x = Math.PI / 2;*

*scene.add(plane);*

*// Create cone geometry (ostrosłup z okrągłą podstawą)*

*const coneGeometry = new THREE.ConeGeometry(2, 4, 32); // 32 segments for smooth circle*

*const coneMaterial = new THREE.MeshPhongMaterial({ color: 0xFFD700 });*

*const cone = new THREE.Mesh(coneGeometry, coneMaterial);*

*cone.position.y = 2;*

*scene.add(cone);*

*// Create sphere on top of the cone*

*const sphereGeometry = new THREE.SphereGeometry(1, 32, 32);*

*const sphereMaterial = new THREE.MeshPhongMaterial({ color: 0xFFD700 });*

*const sphere = new THREE.Mesh(sphereGeometry, sphereMaterial);*

*sphere.position.y = 4.3;*

*scene.add(sphere);*

*// Create bottom disk*

*const diskGeometry = new THREE.CylinderGeometry(2.2, 2.2, 0.4, 32);*

*const diskMaterial = new THREE.MeshPhongMaterial({ color: 0xFFD700 });*

*const disk = new THREE.Mesh(diskGeometry, diskMaterial);*

*disk.position.y = 0.2;*

*scene.add(disk);*

*// Create middle disk (smaller)*

*const middleDiskGeometry = new THREE.CylinderGeometry(1.4, 1.4, 0.2, 32);*

*const middleDisk = new THREE.Mesh(middleDiskGeometry, diskMaterial);*

*middleDisk.position.y = 1.5;*

*scene.add(middleDisk);*

*// Create top disk (smaller)*

*const topDiskGeometry = new THREE.CylinderGeometry(0.8, 0.8, 0.2, 32);*

*const topDisk = new THREE.Mesh(topDiskGeometry, diskMaterial);*

*topDisk.position.y = 3.3;*

*scene.add(topDisk);*

*// Render loop*

*function animate() {*

*requestAnimationFrame(animate);*

*controls.update();*

*renderer.render(scene, camera);*

*}*

*animate();*

*}*

*window.onload = init;*

*</script>*

*</head>*

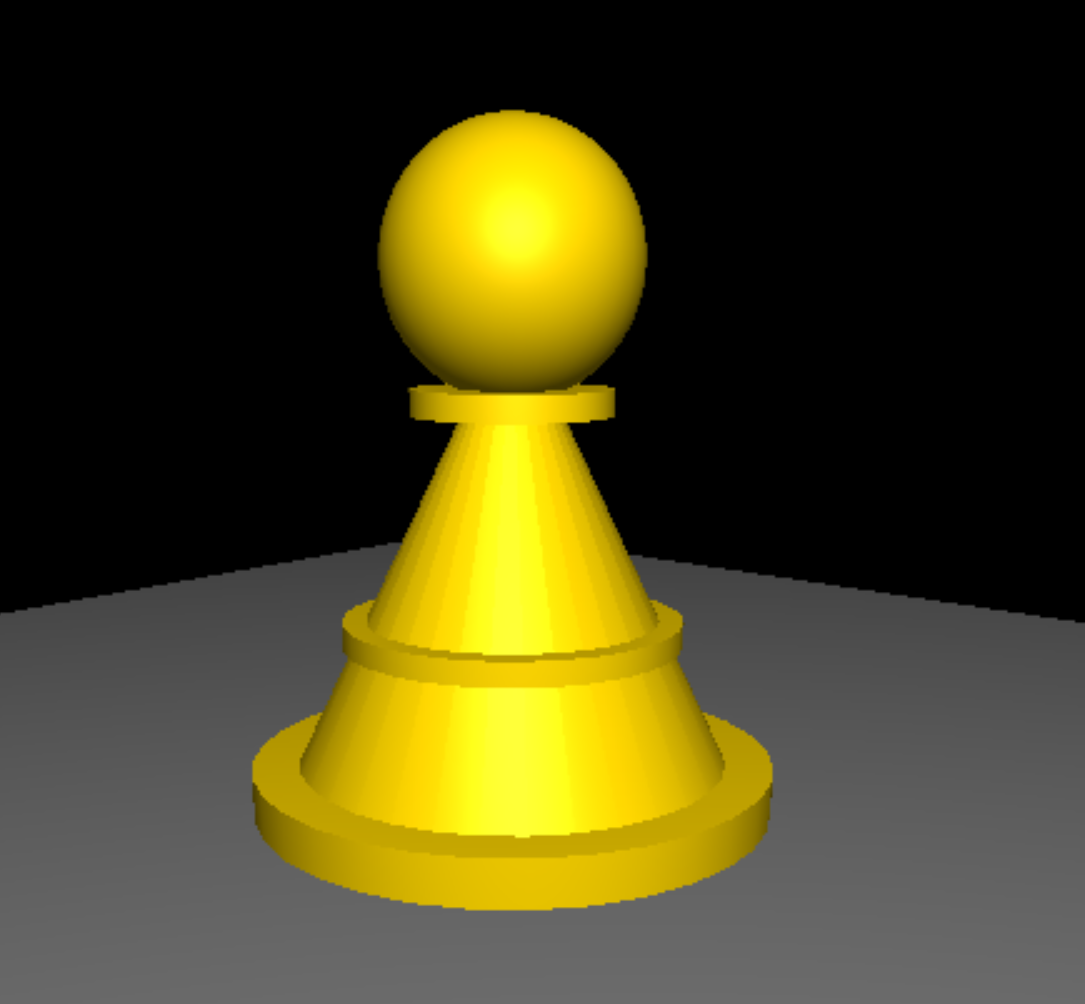
*<body>*

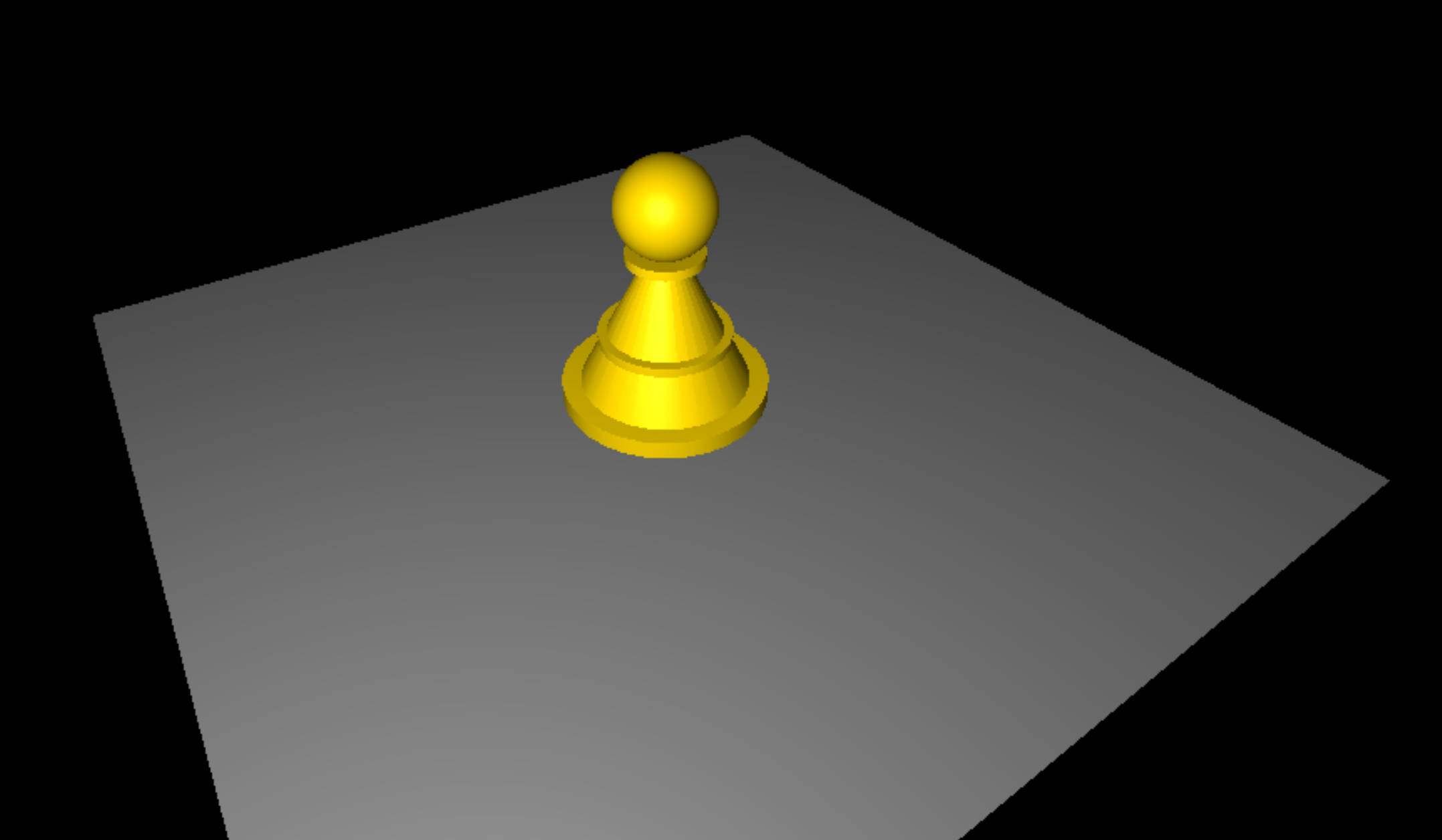
*</body>*

*</html>*

Link do zdalnego repozytorium: https://github.com/Slayzerus/UBB\_GrafikaKomputerowa/tree/main/Lab%2010

**4. Wynik działania:**

****

****

**5. Wnioski:**

Podczas realizacji zadania związanym z konstruowaniem obiektów za pomocą Three.js nauczyliśmy się, jak tworzyć różnorodne geometrie i siatki od podstaw. Proces ten obejmował dodawanie wierzchołków oraz ścian do geometrii, a także obliczanie wektorów normalnych w celu uzyskania odpowiedniego oświetlenia.

Poznaliśmy różne rodzaje materiałów dostępnych w bibliotece, takie jak MeshBasicMaterial, MeshLambertMaterial oraz MeshPhongMaterial, oraz ich zastosowania w renderowaniu scen 3D. Dowiedzieliśmy się również, jak stosować różne kolory i tekstury do ścian obiektów, co pozwoliło na uzyskanie bardziej realistycznych efektów wizualnych.

Kolejnym ważnym aspektem było poznanie metod pracy z krzywymi i powierzchniami parametrycznymi. Dzięki temu mogliśmy definiować skomplikowane powierzchnie i geometrie rurowe oraz obrotowe, co znacząco zwiększyło nasze możliwości tworzenia złożonych modeli 3D.

Ponadto, nauczyliśmy się, jak ładować modele 3D zapisane w formacie JSON, korzystając z klasy THREE.JSONLoader. Umiejętność ta pozwoliła na importowanie i wykorzystywanie gotowych modeli w naszych scenach Three.js, co znacznie ułatwiło proces tworzenia bardziej zaawansowanych aplikacji.

Podsumowując, zadanie to dostarczyło nam solidnych podstaw w zakresie tworzenia i manipulacji obiektami 3D za pomocą Three.js, co otworzyło przed nami szerokie możliwości w dziedzinie grafiki komputerowej i programowania wizualizacji trójwymiarowych.