**SPRAWOZDANIE**

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Laboratorium II**

Data

**Temat: Zadanie\_threejs\_k**

**Wariant 8**

Jakub Bąk

Informatyka I stopień,

stacjonarne,

4 semestr,

Gr.3b

# Polecenie

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

# Zdjęcie z Wariantem:

****

# Wykorzystane komendy:

Link do github: <https://github.com/Szeladin/grafika.git>

Kod Programu:

1. <!DOCTYPE html>

2. <html>

3. <head>

4.     <meta charset="UTF-8">

5.  <style>

6.     body {margin: 0;}

7.     canvas {width: 100%; height: 100%;}

8. </style>

9. <script src="three.js"></script>

10. </head>

11. <body>

12. <script>

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

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

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

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

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

18. renderer.setClearColor(0x808080);

19.

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

21.

22. const light = new THREE.PointLight( 0xffffff, 1, 300 );

23. light.position.set(0, 0, 0);

24. camera.add(light);

25. scene.add(camera);

26. const material = new THREE.MeshPhongMaterial(

27.     {

28.         color: 0xffffff,

29.     });

30.

31. let black\_rook = new THREE.Group();

32. const baseGeometry = new THREE.CylinderGeometry(0.4, 0.4, 0.1,100);

33. const base = new THREE.Mesh(baseGeometry, material);

34. black\_rook.add(base);

35. base.position.y = 0.2;

36. const rookGeometry\_1 = new THREE.CylinderGeometry(0.8,0.8, 0.1, 100);

37. const rookPart\_1 = new THREE.Mesh(rookGeometry\_1, material);

38. rookPart\_1.position.y=0.3;

39. black\_rook.add(rookPart\_1);

40. const rookGeometry\_2 = new THREE.CylinderGeometry(0.45, 0.7, 0.8, 100);

41. const rookPart\_2 = new THREE.Mesh(rookGeometry\_2, material);

42. rookPart\_2.position.y=0.75;

43. black\_rook.add(rookPart\_2);

44. const rookGeometry\_3 = new THREE.CylinderGeometry(0.55, 0.55, 0.3, 100);

45. const rookPart\_3 = new THREE.Mesh(rookGeometry\_3, material);

46. rookPart\_3.position.y=1.29;

47. black\_rook.add(rookPart\_3);

48. const rookGeometry\_4 = new THREE.CylinderGeometry(0.6, 0.5, 0.7, 100);

49. const rookPart\_4 = new THREE.Mesh(rookGeometry\_4, material);

50. rookPart\_4.position.y=1.79;

51. black\_rook.add(rookPart\_4);

52. const rookGeometry\_5 = new THREE.CylinderGeometry(0.75, 0.75, 0.1, 100);

53. const rookPart\_5 = new THREE.Mesh(rookGeometry\_5, material);

54. rookPart\_5.position.y=2.1;

55. black\_rook.add(rookPart\_5);

56. const rookGeometry\_6= new THREE.CylinderGeometry(0.8, 0.8, 0.2, 100);

57. const rookPart\_6 = new THREE.Mesh(rookGeometry\_6, material);

58. black\_rook.add(rookPart\_6);

59. rookPart\_6.position.y=2.25;

60. black\_rook.position.set(0,-3,1);

61. scene.add(black\_rook);

62. renderer.render(scene, camera);

63. function animate() {

64.     requestAnimationFrame(animate);

65.     renderer.render(scene, camera);

66. }

67. camera.position.z = 4.5;

68. animate();

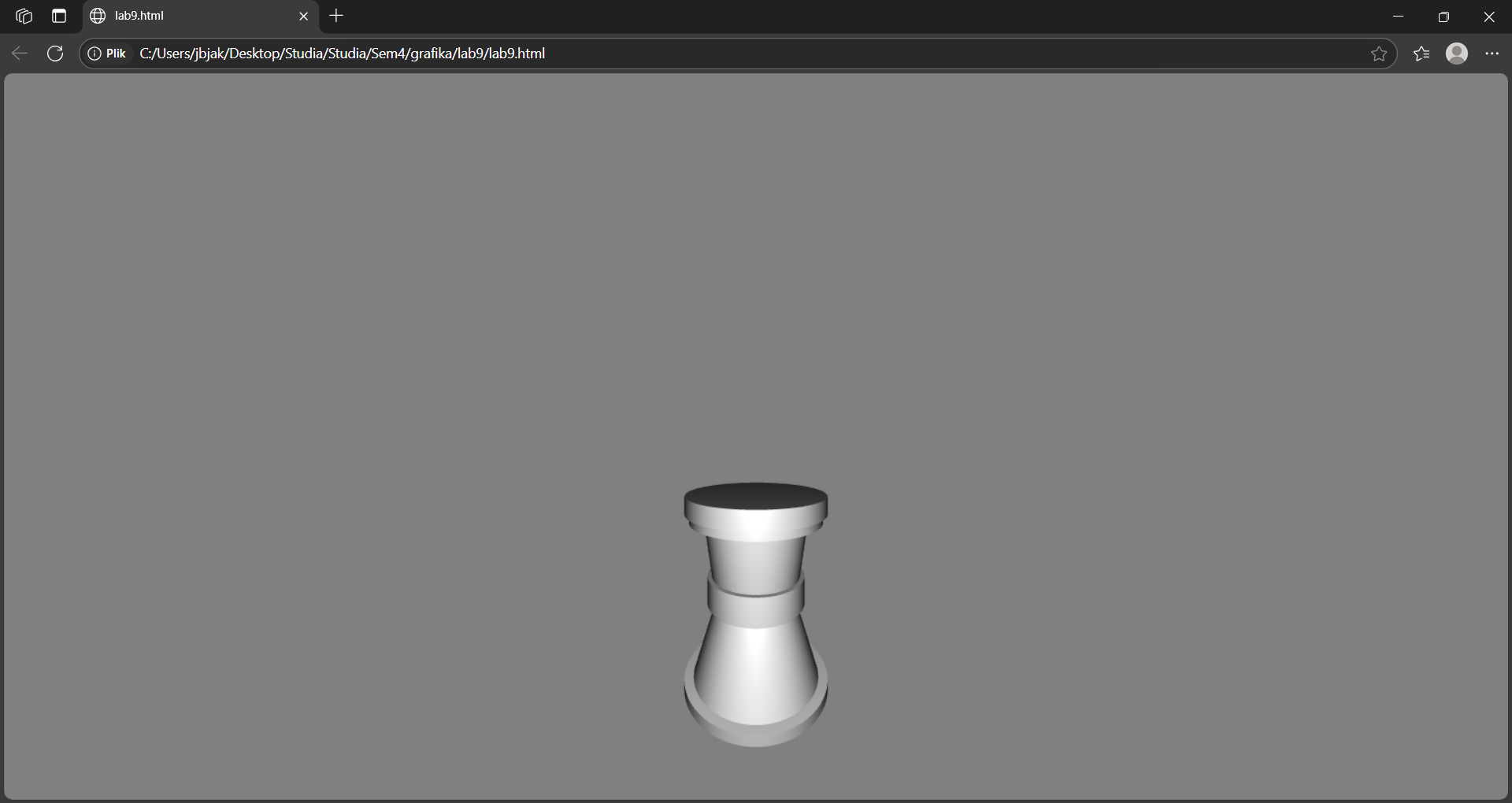
69. </script>

70. </body>

71. </html>

72.

# Wyniki i wnioski:



Three.js jest narzędziem do tworzenia interaktywnych scen 3D w przeglądarce. Dzięki jego szerokim możliwościom, takim jak obsługa geometrii, materiałów, świateł i animacji, można łatwo tworzyć proste modele, jak np. pionek w tym projekcie.