

WMM

Lab 8: Generowanie grafiki z wykorzystaniem popularnej biblioteki graficznej

Yelsukova Anastasiia, 323719

Zadanie 1

Wynik zadania można zobaczyć z pomocą: `--model_name=sphere.obj`
`--shader_path=./resources/shaders/robot --shader_name=robot`

Po uruchomieniu widać takiego robota z:

zieloną głową:

```
self.color.value = (0.155, 0.50, 0.200) #kolor
move = Matrix44.from_translation((0.0, 0.0, 5.0)) #translacja
self.mvp.write((projection * lookat * move).astype('float32')) #zapisanie macierzy z translacjami do mvp
self.sphere.render() #Rendering głowy
```

pomarańczowe ciało:

```
self.color.value = (0.955, 0.400, 0.200)
move = Matrix44.from_translation((0.0, 0.0, 2.0))
scale = Matrix44.from_scale((1.0, 1.0, 2.0)) #skalowanie
self.mvp.write((projection * lookat * move * scale).astype('float32'))
self.cube.render() #render ciała
```

niebieskie ręce:

```
self.color.value = (0.255, 0.80, 0.800)
move = Matrix44.from_translation((0.0, -3.0, 3.0))
rotate = Matrix44.from_x_rotation(0.25 * np.pi) #rotacja
scale = Matrix44.from_scale(Vector3([0.5, 0.5, 1.25])) #skalowanie
self.mvp.write((projection * lookat * move * rotate * scale).astype('float32'))
self.cube.render() #render lewej reki

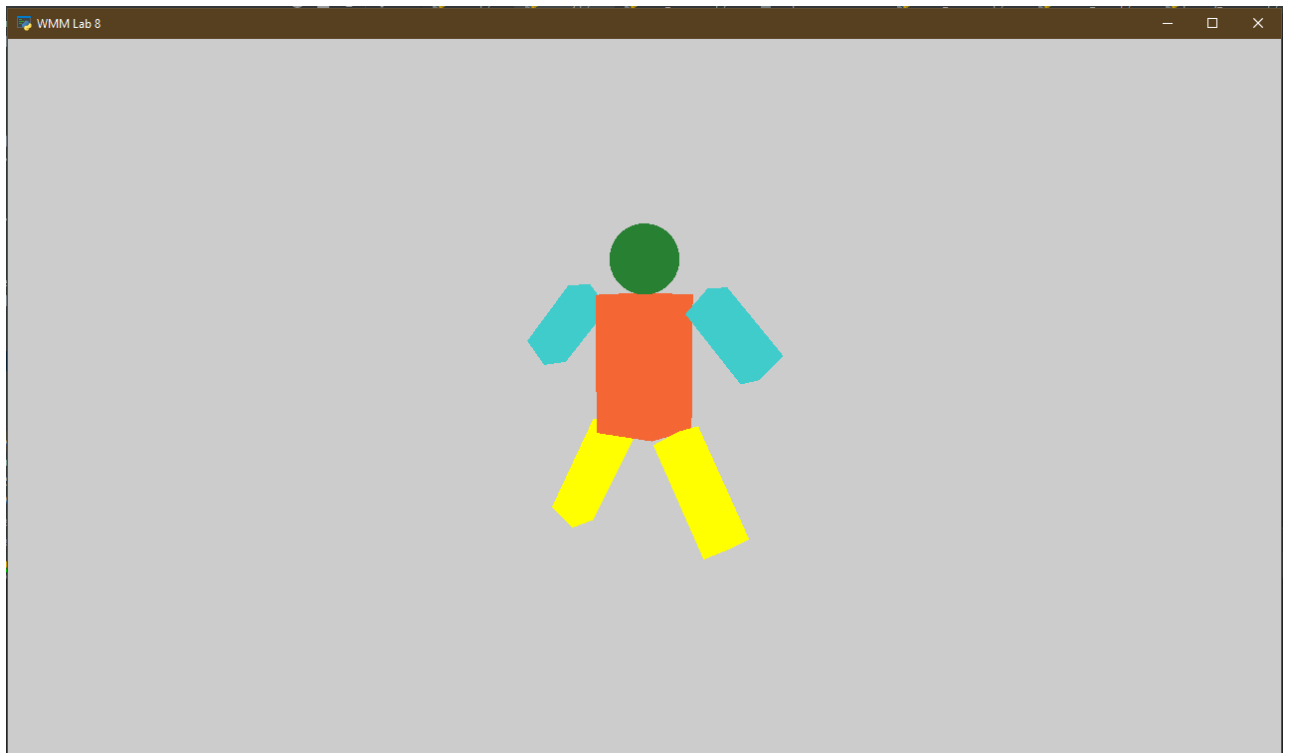
self.color.value = (0.255, 0.80, 0.800)
move = Matrix44.from_translation(Vector3([0.0, 3.0, 3.0]))
rotate = Matrix44.from_x_rotation(-0.25 * np.pi)
scale = Matrix44.from_scale(Vector3([0.5, 0.5, 1.25]))
self.mvp.write((projection * lookat * move * rotate * scale).astype('float32'))
self.cube.render() #render prawej reki
```

żółte nogi:

```
self.color.value = (1.0, 1.0, 0.0)
move = Matrix44.from_translation((0.0, 2.0, -1.5))
scale = Matrix44.from_scale((0.5, 0.5, 1.75))
rotate = Matrix44.from_x_rotation(np.pi / -6)
self.mvp.write((projection * lookat * move * rotate * scale).astype('float32'))
self.cube.render() #render lewej nogi

self.color.value = (1.0, 1.0, 0.0)
move = Matrix44.from_translation((0.0, -2.0, -1.5))
scale = Matrix44.from_scale((0.5, 0.5, 1.75))
rotate = Matrix44.from_x_rotation(np.pi / 6)
self.mvp.write((projection * lookat * move * rotate * scale).astype('float32'))
self.cube.render() #render prawej nogi
```

Wynik:



Został stworzony taki robot z pomocą transformacji obiektów (skalowanie, kolor, rotacja, translacja).