

SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 11

Data: 24.05.2022

Temat: "Grafika 3D w bibliotece WebGL/GLSL"

Wariant: 14 skrzydeł

Michał Wielopolski
Informatyka I stopień,
stacjonarne,
4 semestr,
Gr. 4

1. Polecenie:

Plik lab12.html pokazuje mały sześcián, który można obrócić, przeciągając myszą na płótnie. Zadaniem jest zastąpienie sześciánu dużym wiatrakim siedzącym na prostokątnej podstawie, jak pokazano na rysunku. Łopatki wiatraka powinny obracać się po włączeniu animacji. Każda łopatka wiatraka powinna być zbudowana z dwóch stożków. (Dodanie czajniczka, który znajduje się na podstawie, jest konieczne dla uzyskania oceny "5")

Program zawiera trzy zmienne instancji reprezentujące podstawowe obiekty: cube, cone, cylinder. Te zmienne mają metody instancji cube.render(), cone.render(), cylinder.render(), które można wywołać w celu narysowania obiektów. Obiekty nietransformowane mają rozmiar 1 we wszystkich trzech kierunkach i mają swój środek na (0,0,0). Oś stożka i oś cylindra są wyrównane wzdłuż osi Z. Wszystkie obiekty na scenie powinny być przekształconymi wersjami podstawowych obiektów (lub podstawowego obiektu czajnika).

2. Wprowadzane dane:

14 podwójnych stożków

3. Wykorzystane komendy:

a) kod źródłowy

```
function addBlade()
{
    pushMatrix();
    currentColor = [200/255, 1, 252/255];
    mat4.rotateY(modelview, modelview, Math.PI /2);
    mat4.translate(modelview, modelview, [0, 0, 2.4]);
    mat4.scale(modelview, modelview, [0.4, 0.4, 2.5]);
    cone.render();
    popMatrix();

    pushMatrix();
    currentColor = [200/255, 1, 252/255];
    mat4.translate(modelview, modelview, [0.3, 0, 0]);
    mat4.rotateY(modelview, modelview, Math.PI /2);
    mat4.rotateX(modelview, modelview, Math.PI );
    mat4.scale(modelview, modelview,[0.4, 0.4, 1.7]);
    cone.render();
    popMatrix();
}

var rotateEachFrame = 0.5;
let wierzcholki = 14;

function draw() {
```

```

gl.clearColor(0,0,0,1);
gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);

mat4.perspective(projection, Math.PI/4, 1, 1, 50);
gl.uniformMatrix4fv(u_projection, false, projection );

mat4.lookAt(modelview, [0,0,25], [0,0,0], [0,1,0]);

mat4.rotateX(modelview,modelview,rotateX);
mat4.rotateY(modelview,modelview,rotateY);

pushMatrix();
currentColor = [0.999, 0.111, 0.111];
mat4.translate(modelview, modelview, [0, -5, 0]);
mat4.scale(modelview,modelview,[5, 1, 5]);
cube.render();
popMatrix();

pushMatrix();
currentColor = [0.952, 0.784, 0.223];
mat4.translate(modelview, modelview, [0, 0, 0]);
mat4.rotateX(modelview,modelview, Math.PI * 0.5);
mat4.scale(modelview,modelview,[0.4, 0.4, 10]);
cylinder.render();
popMatrix();

pushMatrix();
currentColor = [0.222, 0.888, 0.222];
mat4.translate(modelview, modelview, [1.2, -4.1, 2.0]);
mat4.scale(modelview,modelview,[0.05, 0.05, 0.05]);
teapot.render();
popMatrix();

pushMatrix();
mat4.translate(modelview, modelview, [-2.9, 2, 0.2]);
pushMatrix();
mat4.translate(modelview, modelview, [2.9, 2, 0]);
mat4.rotate(modelview, modelview, rotateEachFrame, [0, 0, 1]);
mat4.translate(modelview, modelview, [2.9, -2, 0]);

for(let i =0; i<wierzcholki;i++){

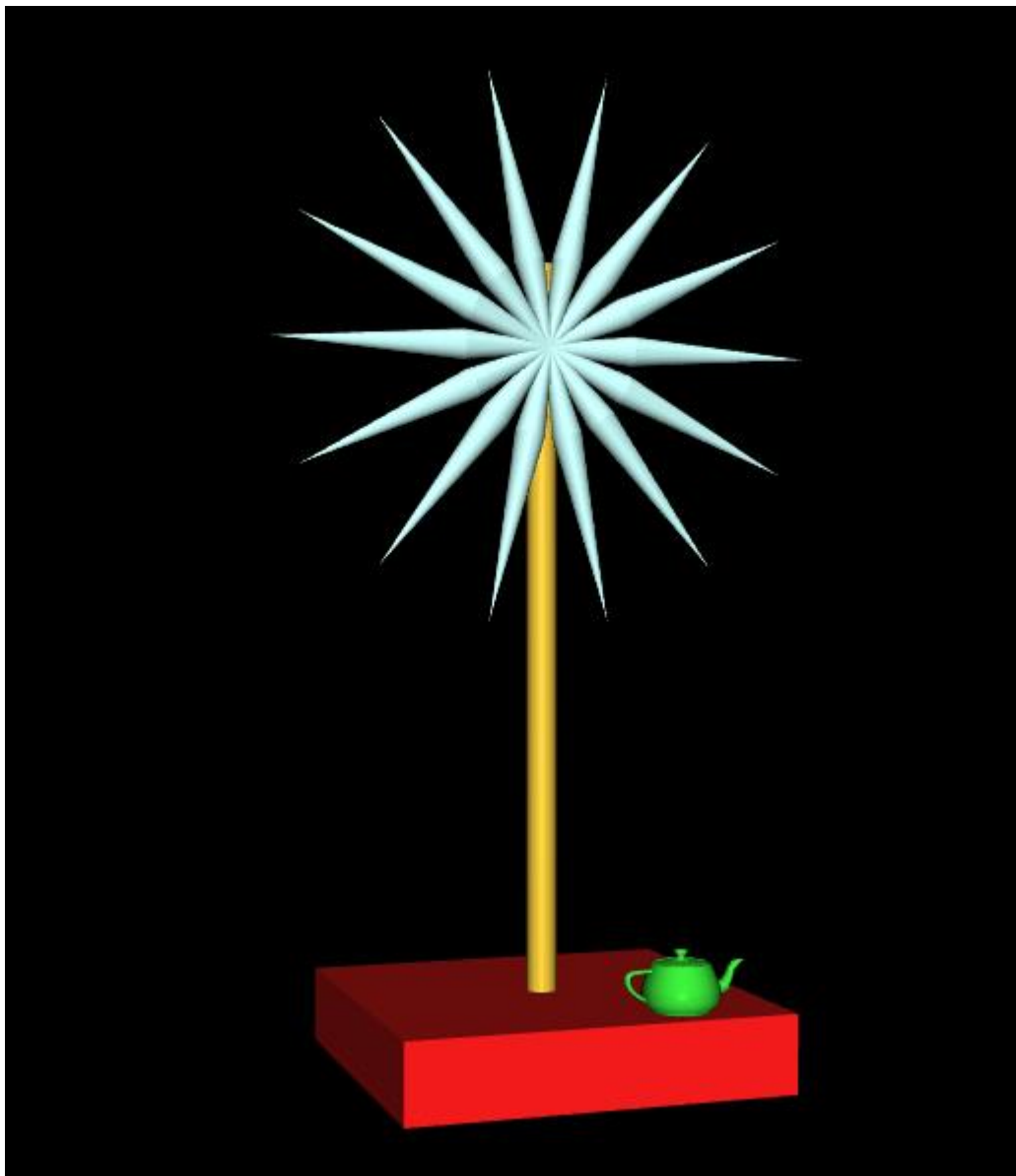
    pushMatrix();
    mat4.translate(modelview, modelview, [-3, 1.95, 0]);
    mat4.rotateZ(modelview,modelview, i*(360/wierzcholki)*(Math.PI/180));
    mat4.rotateY(modelview,modelview, Math.PI);

```

```
    addBlade();  
    popMatrix();  
  
}  
} // end draw();
```

<https://github.com/wielopolski/GrafikKomputerowa>

4. Wynik działania:



5. Wnioski:

Na podstawie otrzymanego wyniku można stwierdzić, że Grafika 3D w bibliotece WebGL/GLSL daje wiele możliwości, stworzenia obiektów i animacji trój-wymiarowych.