

SPRAWOZDANIE

Zajęcia: Grafika Komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium: 11

Data: 20.05.2024

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

Michał Michalik
Informatyka I stopień,
stacjonarne,
4 semestr,
Gr.3a

Zadanie 1

1. Polecenie:

Plik lab12.html pokazuje mały sześcian, który można obrócić, przeciągając myszą na płótnie. Zadaniem jest zastąpienie sześcianu dużym wiatrakiem siedzącym na prostokątnej podstawie, jak pokazano na rysunku. Łopatki wiatraka powinny obracać się po włączeniu animacji. Każda łopatką 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:

```

135  function draw() {
136      gl.clearColor(0, 0, 0, 1);
137      gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
138
139      mat4.perspective(projection, Math.PI / 4, 1, 1, 50);
140      gl.uniformMatrix4fv(u_projection, false, projection);
141
142      mat4.lookAt(modelview, [0, 0, 25], [0, 0, 0], [0, 1, 0]);
143
144      mat4.rotateX(modelview, modelview, rotateX);
145      mat4.rotateY(modelview, modelview, rotateY);
146
147      //podstawa
148      pushMatrix();
149      currentColor = [0.999, 0.111, 0.111];
150      mat4.translate(modelview, modelview, [0, -5, 0]);
151      mat4.scale(modelview, modelview, [5, 1, 5]);
152      cube.render();
153      popMatrix();
154
155      //slupek
156      pushMatrix();
157      currentColor = [0.999, 0.555, 0.333];
158      mat4.translate(modelview, modelview, [0, 0, 0]);
159      mat4.rotateX(modelview, modelview, Math.PI * 0.5);
160      mat4.scale(modelview, modelview, [0.4, 0.4, 10]);
161      cylinder.render();
162      popMatrix();
163
164      //lopatki
165      pushMatrix();
166      mat4.translate(modelview, modelview, [-2.9, 2, 0.2]);
167      pushMatrix();
168      mat4.translate(modelview, modelview, [2.9, 2, 0]);
169      mat4.rotate(modelview, modelview, rotateEachFrame, [0, 0, 1]);
170      mat4.translate(modelview, modelview, [2.9, -2, 0]);

```

```

172   for (let i = 0; i < 17; i++) {
173       pushMatrix();
174       mat4.translate(modelview, modelview, [-3, 1.95, 0]);
175       mat4.rotateZ(modelview, modelview, i * (360 / 17) * (Math.PI / 180)); // Changed 13 to 17
176       mat4.rotateY(modelview, modelview, Math.PI);
177
178       pushMatrix();
179
180       currentColor = [200 / 255, 1, 252 / 255];
181       mat4.rotateY(modelview, modelview, Math.PI / 2);
182       mat4.translate(modelview, modelview, [0, 0, 2.7]);
183       mat4.scale(modelview, modelview, [0.7, 0.7, 4.1]);
184       cone.render();
185       popMatrix();
186
187       pushMatrix();
188       currentColor = [200 / 255, 1, 252 / 255];
189       mat4.translate(modelview, modelview, [0.3, 0, 0]);
190       mat4.rotateY(modelview, modelview, Math.PI / 2);
191       mat4.rotateX(modelview, modelview, Math.PI);
192       mat4.scale(modelview, modelview, [0.7, 0.7, 0.7]);
193       cone.render();
194       popMatrix();
195
196       popMatrix();
197   }
198 } // end draw();
199
200

```

3. Wykorzystane komendy:

Link git dodac

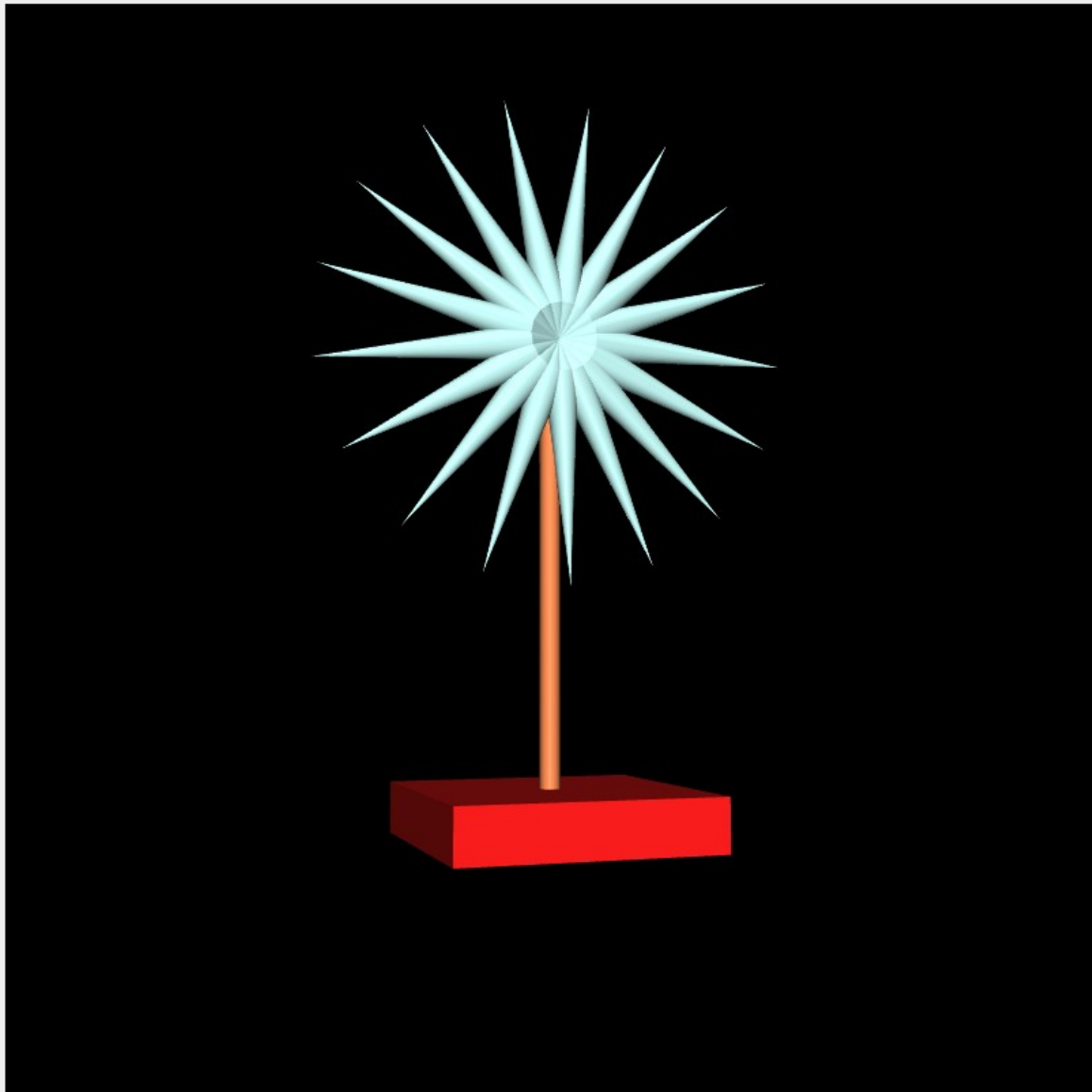
4. Wynik działania:

DiskWorld 2: WebGL Lighting and Hierarchical Modeling

Drag your mouse on the model to rotate it.

☐ Animate

Reset



5. Wnioski:

Biblioteka WebGL/GLSL to narzędzie umożliwiające tworzenie grafiki w przeglądarkach internetowych, bazujące na bibliotece OpenGL. Choć oferuje szerokie możliwości, jej obsługa może być skomplikowana.