

Programowanie OpenGL z użyciem shader'ów. Proste oświetlenie.

1. Cel ćwiczenia:

Zapoznanie z funkcjami realizującymi oświetlenie sceny. Ustawianie właściwości materiału oraz właściwości źródeł światła.

2. Zastosowanie światła:

W celu określenia sceny wyglądającej maksymalnie realistycznie konieczne jest zastosowanie oświetlenia. Światła oraz materiały posiadają właściwości odpowiedzialne za wygląd oświetlanych obiektów. Dobranie światła i własności materiałów tak, aby uzyskać pożądany efekt, wymaga testów oraz praktyki.

Rozróżniamy 3 rodzaje światel:

1. światła kierunkowe – ich promienie są równoległe, ich natężenie jest jednakowe.
2. światła punktowe – nie posiadają kierunku, oświetlają całą scenę, a natężenie zależy od odległości,
3. reflektory – świecą w danym kierunku z określonej pozycji. Snop światła mieści się w ściśle określonym stożku

Każdy rodzaj światła posiada składowe:

- światło otaczające – nie pochodzi z żadnego kierunku, powoduje równomierne oświetlenie całej sceny,
- światło rozproszone – promienie pochodzą z konkretnego punktu i oświetlają obiekt równomiernie.
- światło odbłyśków – promienie nie są odbijane równomiernie tylko ostro w jednym z kierunków.

Potrzebne informacje:

- a. zaimportować obiekt do VBO wraz z informacjami o wektorach normalnych.
- b. przekazać informacje o wektorach normalnych określając specyfikację formatu danych wierzchołkowych

```
GLint NorAttrib = glGetUniformLocation(shaderProgram, "aNormal");
```

```
glEnableVertexAttribArray(NorAttrib);
```

glVertexAttribPointer(

NorAttrib, 3, *GL_FLOAT*, *GL_FALSE*, 8 * *sizeof(GLfloat)*, (void*)(3 * *sizeof(GLfloat)*));

- c. ustalić położenie światła np:

glm::vec3 lightPos(1.2f, 1.0f, 2.0f);

GLint uniLightPos = glGetUniformLocation(shaderProgram, "lightPos");

glUniform3fv(uniLightPos, 1, &lightPos[0]);

W kodzie vertex shadera:

- a. odebrać informację o wektorach normalnych i przekazać do fragment shadera:

in vec3 aNormal;

out vec3 Normal;

- b. zadeklarować zmienną do przekazania pozycji fragmentów:

out vec3 FragPos;

W kodzie fragment shadera:

- a. odebrać informację o wektorach normalnych z vertex shadera:

in vec3 Normal;

in vec3 FragPos;

- b. dodać moc i kolor światła otoczenia poprzez przemnożenie aktualnego koloru fragmentu przez składową światła:

float ambientStrength = 0.1; // ambient

vec3 ambientlightColor = vec3(1.0,1.0,1.0);

*vec4 ambient = ambientStrength * vec4(ambientlightColor,1.0);*

- c. dodać składową koloru światła rozproszonego:

vec3 diffLightColor = vec3(1.0,1.0,1.0);

- d. Wyznaczymy wektor kierunku między źródłem światła, a pozycją fragmentu i normalizujemy zarówno normalny, jak i wynikowy wektor kierunkowy:

vec3 norm = normalize(Normal);

vec3 lightDir = normalize(lightPos - FragPos);

- e. Ustalamy wpływ składowej rozproszonej światła na bieżący fragment, biorąc iloczyn skalarny normy i wektora lightDir (dla kąta między wektorami >90 stopni wynik jest ujemny dlatego zastosowano funkcję max)

```
float diff = max(dot(norm, lightDir), 0.0);
```

```
vec3 diffuse = diff * diffLightColor;
```

- f. Ostatecznie mając dwie składowe otoczenia i światła rozproszonego dodajemy je do siebie i mnożymy przez bieżący kolor fragmentu

```
outColor = (ambient+vec4(diffuse,1.0)) * texture(texture1, TexCoord);
```

3. Zadanie.

Do utworzonej podczas wcześniejszych zajęć sceny dołożyć oświetlenie. Rodzaj oświetlenia oraz jego moc, włączanie /wyłączenie powinno zostać zdefiniowane i przypisane pod odpowiednie klawisze, tak aby możliwa była zmiana tych składowych.