



**POLITECHNIKA WROCŁAWSKA**  
**Instytut Informatyki, Automatyki i Robotyki**  
**Zakład Systemów Komputerowych**

**Wprowadzenie do grafiki komputerowej**

**Kurs: INE4234L**

**Sprawozdanie z ćwiczenia nr 6**

**OpenGL - teksturowanie powierzchni obiektów**

<b>Wykonał:</b>	Kamil Kamyszek
<b>Termin:</b>	PT/NP. 11.00-14.00
<b>Data wykonania ćwiczenia:</b>	21.12.18r.
<b>Data oddania sprawozdania:</b>	18.01.10r.
<b>Ocena:</b>	

Uwagi prowadzącego:

# 1 Wstęp

Na szóstym laboratorium z grafiki komputerowej studenci mieli za zadanie zapoznać się z tekstutowaniem powierzchni obiektów używając biblioteki OpenGL. Do zrozumienia tego tematu potrzebne było uważne przestudiowanie instrukcji zamieszczonej przez prowadzącego na stronie ZSK, jak również zaznajomienie się z nowo poznanymi funkcjami bibliotek OpenGL i GLUT.

## 2 Przebieg Laboratorium

Laboratorium rozpoczęło od stworzenia trójkąta oświetlonego światłem białym. Następnie dzięki instrukcji do zajęć nałożono teksturę na figurę. Kolejne zadanie studenci musieli już wykonać sami, a mianowicie było to stworzenie piramidy, którą buduje się poprzez kolejne wciskanie cyfr na klawiaturze, do których przypisane są funkcje wyświetlające kolejne ściany piramidy. Wciskając cyfrę „6” figura miała zniknąć. Budowla miała mieć nałożone tekstury na ściany. Drugie zadanie samodzielne to nałożenie tekstury na jajko z poprzednich zajęć. W przypadku jajka problemem było stworzenie tekstury, która po nałożeniu wyglądałaby estetycznie. Nakładając kwadratową teksturę na obiekt, który nie jest tą figurą musimy pogodzić się niedokładnością tej grafiki na obiekcie.

## 3 Zadania do samodzielnego wykonania

Wykonanie pierwszego zadania z piramidą wymagało zapoznania się z instrukcją na stronie ZSK. Umożliwiło to stworzenie takiej bryły: **Trójkąt z teksturą**

Najważniejsze funkcje:

### Część funkcji RenderScene() rysująca kolejne elementy piramidy

```
if (s[0] != state_1)                                     //Wciśnięta cyfra 1
{
    glBegin(GL_TRIANGLES);
    //pierwszy trójkąt
    glNormal3f(0.0f, 0.0f, 1.0f);                         //Wektory normalne z poprzednich zajęć
    glTexCoord2f(0.0f, 0.0f);                             //-->Funkcja określająca, które wierzchołki
    glVertex3f(0.0f, 10.0f, 5.0f);                       //trójkąta naniesionego na wzorec tekstury
    glNormal3f(0.0f, 0.0f, 1.0f);                         //odpowiadają którym wierzchołkom
    glTexCoord2f(1.0f, 0.0f);
    glVertex3f(-10.0f, 0.0f, 0.0f);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.5f, 1.0f);
    glVertex3f(10.0f, 0.0f, 0.0f);
    glEnd();
}
if (s[1] != state_2)
{
    //drugi trójkąt
    glBegin(GL_TRIANGLES);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.0f, 0.0f);
    glVertex3f(0.0f, 10.0f, 5.0f);
    glNormal3f(10.0f, 0.0f, 1.0f);
    glTexCoord2f(1.0f, 0.0f);
    glVertex3f(10.0f, 0.0f, 0.0f);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.5f, 1.0f);
    glVertex3f(10.0f, 0.0f, 10.0f);
    glEnd();
}
if (s[2] != state_3)
{
    //trzeci trójkąt
    glBegin(GL_TRIANGLES);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.0f, 0.0f);
    glVertex3f(0.0f, 10.0f, 5.0f);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(1.0f, 0.0f);
    glVertex3f(10.0f, 0.0f, 10.0f);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.5f, 1.0f);
    glVertex3f(-10.0f, 0.0f, 10.0f);
    glEnd();
}
```

```

if (s[3] != state_4)
{
    //czwarty trójkąt
    glBegin(GL_TRIANGLES);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.0f, 0.0f);
    glVertex3f(0.0f, 10.0f, 5.0f);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(1.0f, 0.0f);
    glVertex3f(-10.0f, 0.0f, 0.0f);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.5f, 1.0f);
    glVertex3f(-10.0f, 0.0f, 10.0f);
    glEnd();
}
if (s[4] != state_5)
{
    //piąty trójkąt
    glBegin(GL_POLYGON); //poligon pozwala na stworzenie podstawy piramidy
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.0f, 0.0f);
    glVertex3f(-10.0f, 0.0f, 0.0f);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(1.0f, 0.0f);
    glVertex3f(10.0f, 0.0f, 0.0f);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.5f, 1.0f);
    glVertex3f(10.0f, 0.0f, 10.0f);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glTexCoord2f(0.5f, 0.5f);
    glVertex3f(-10.0f, 0.0f, 10.0f);
    glEnd();
}

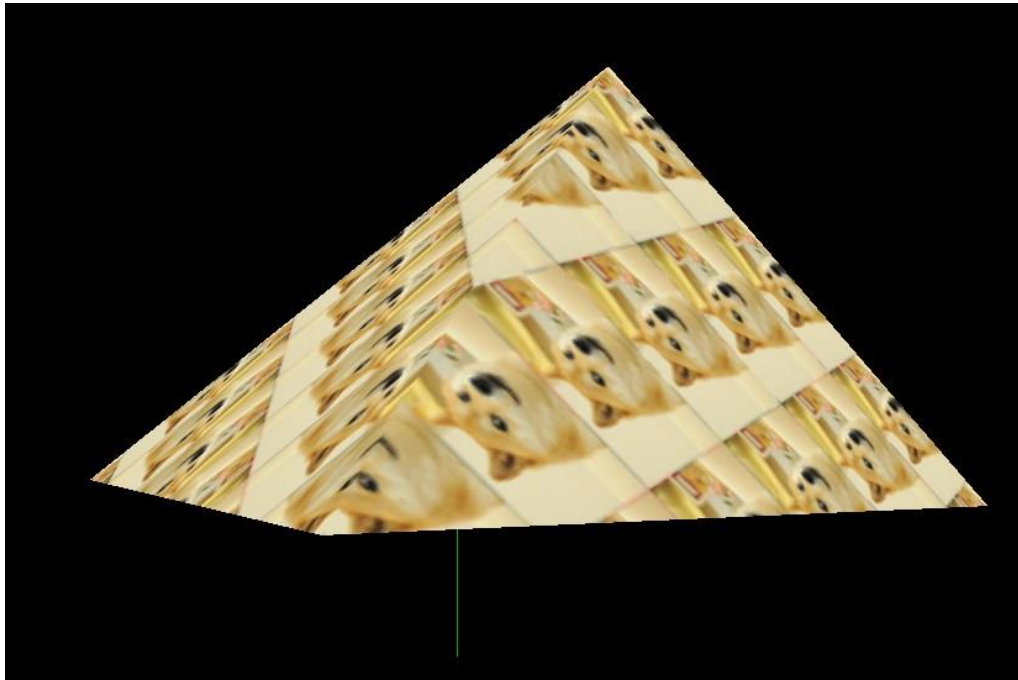
```

### Funkcja odpowiadająca za odczytywanie znaków z klawiatury

```

void keys(unsigned char key, int x, int y)
{
    if (key == '1') state_1 = true;    //→Jeśli klawisz wciśnięty
    if (key == '2') state_2 = true;    //to rysuj
    if (key == '3') state_3 = true;
    if (key == '4') state_4 = true;
    if (key == '5') state_5 = true;
    if (key == '6')
    {
        state_1 = false;                //Wyłączanie widoczności
        state_2 = false;                //piramidy
        state_3 = false;
        state_4 = false;
        state_5 = false;
    }
    RenderScene(); // przerysowanie obrazu sceny
}

```



Rysunek 1 Trójkąt z teksturą

Do drugiego zadania wykonałem w Adobe Photoshop CC 2015 prostą teksturę do nałożenia na jajko: **Jajko z nałożoną teksturą**

Kod nie jest komentowany, zawiera w głównej mierze kod z poprzednich zajęć, gdzie był komentowany. Pozostałe elementy zostały wyjaśnione na stronie ZSK.

```
for (i = 0; i < N - 1; )
{
    for (int j = 0; j < N - 1; j++)
    {
        glColor3f(255.0f, 255.0f, 255.0f);
        glBegin(GL_TRIANGLES);

        glNormal3fv(vectorNorm[i][j + 1]);
        glTexCoord2f(textureCord[i][j + 1][0], textureCord[i][j + 1][1]);
        glVertex3fv(tab[i][j + 1]);

        glNormal3fv(vectorNorm[i + 1][j]);
        glTexCoord2f(textureCord[i + 1][j][0], textureCord[i + 1][j][1]);
        glVertex3fv(tab[i + 1][j]);

        glNormal3fv(vectorNorm[i + 1][j + 1]);
        glTexCoord2f(textureCord[i + 1][j + 1][0], textureCord[i + 1][j + 1][1]);
        glVertex3fv(tab[i + 1][j + 1]);

        glNormal3fv(vectorNorm[i + 1][j]);
        glTexCoord2f(textureCord[i + 1][j][0], textureCord[i + 1][j][1]);
        glVertex3fv(tab[i + 1][j]);

        glNormal3fv(vectorNorm[i][j + 1]);
        glTexCoord2f(textureCord[i][j + 1][0], textureCord[i][j + 1][1]);
        glVertex3fv(tab[i][j + 1]);

        glNormal3fv(vectorNorm[i][j]);
        glTexCoord2f(textureCord[i][j][0], textureCord[i][j][1]);
        glVertex3fv(tab[i][j]);

        glEnd();
    }
    i++;
}
```



**Rysunek 2 Jajko z nałożoną teksturą**

## **4 Wnioski**

Dzięki zamieszczonej instrukcji na stronie zsk, wykonanie zadania nie sprawiło większych trudności i pozwoliło oswoić się z zagadnieniem teksturowania. Biblioteki graficzne OpenGL i GLUT w dużym stopniu ułatwiają użytkownikowi zaprogramowanie nałożenia tekstur, dzięki czemu można w krótkim czasie nadać unikalności naszemu obiektowi. Nauczenie się teksturowania ma bardzo wiele zastosowań w grafice 3D i pozwala na zwiększenie realizmu tworzonych struktur. Pozwala również na stworzenie unikalnego obiektu, który nie będzie składał się tylko z nałożonych na siebie gradientów.