



POLITECHNIKA WROCŁAWSKA
Instytut Informatyki, Automatyki i Robotyki
Zakład Systemów Komputerowych

Wprowadzenie do grafiki komputerowej

Kurs: INE4234L

Sprawozdanie z ćwiczenia nr 3

Open GL - modelowanie obiektów 3-D

Wykonał:	Wojciech Wójcik, 235621
Termin:	PT/TP 8:00-11:00
Data wykonania ćwiczenia:	29.10.2018
Data oddania sprawozdania:	16.11.2018
Ocena:	

Uwagi prowadzącego:

1. Cel ćwiczenia

Celem ćwiczenia jest wprowadzenie w zagadnienia modelowania i wizualizacji scen 3D z wykorzystaniem biblioteki OpenGL z rozszerzeniem **GLUT**. Zamieszczone w treści opisu ćwiczenia przykłady pokazują jak w układzie współrzędnych trójwymiarowych wykonuje się transformacje obiektów oraz jak na podstawie równań parametrycznych można stworzyć własny model nietrywialnego obiektu. Przy okazji pokazano także sposób sterowania wykonaniem programu przy pomocy klawiatury oraz prosty przykład z zakresu animacji.

2. Zadanie do wykonania

Na laboratorium należało zbudować swój własny model jajka od samych podstaw mając do dyspozycji wzory jak i wyjaśnione podstawowe zagadnienia teoretyczne.

3. Realizacja zadania

3.1. Zbiór punktów

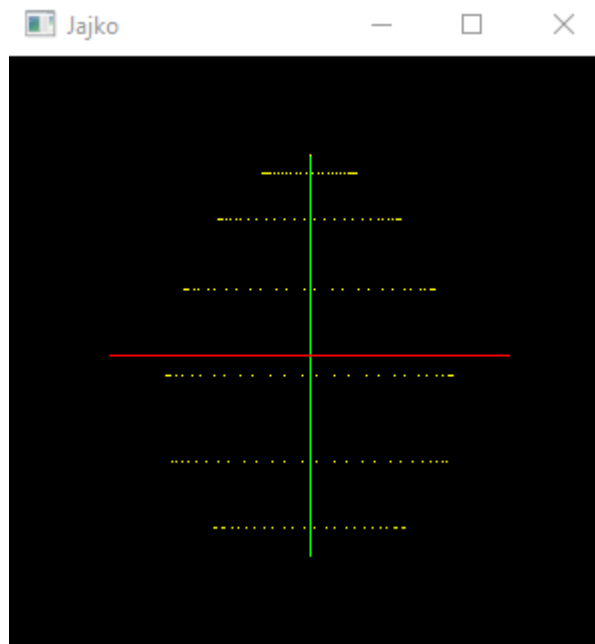
Mając do dyspozycji wzory:

$$\begin{aligned}x(u,v) &= (-90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u) \cos(\pi v) \\ y(u,v) &= 160u^4 - 320u^3 + 160u^2 \\ z(u,v) &= (-90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u) \sin(\pi v)\end{aligned}\quad \begin{aligned}0 \leq u \leq 1 \\ 0 \leq v \leq 1\end{aligned}$$

Zostały wyznaczone pozycje w przestrzeni 3D każdego z wierzchołków, po czym każdy z tych wierzchołków został narysowany w przestrzeni 3D.

```
for (int j = 0; j < N; j++)
{
    for (int i = 0; i < N; i++)
    {
        glVertex3f(x(KwJed[i][j].u, KwJed[i][j].v),
            y(KwJed[i][j].u, KwJed[i][j].v), z(KwJed[i][j].u,
            KwJed[i][j].v));
    }
}
glEnd();
```

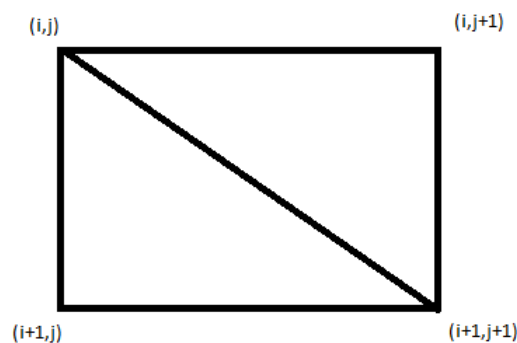
Efektym wyrysowania tych wierzchołków było uzyskanie obrazu 3D z ustawionymi wierzchołkami jajka, co wprowadziło do kolejnych zadań.



3.2. Połączenie punktów w trójkąty

Następnym etapem jest odpowiednie połączenie wierzchołków tak, aby utworzyły siatkę jajka. Do dyspozycji mamy tablicę dwuwymiarową złożoną z punktów z wyliczonymi współrzędnymi 3D. Najłatwiej jest połączyć je w kwadraty, jednak właśnie to trójkąt jest podstawową figurą geometryczną używaną w aplikacjach 3D. Na szczęście kwadrat można złożyć z 2 trójkątów. Aby sprawdzić czy dobrze łączymy wierzchołki najpierw połączyliśmy je liniami w taki sposób by stworzyły trójkąty.

Zamierzony efekt został uzyskany przez ryzowanie trójkątów złożonych z wierzchołków: $(i,j), (i+1,j), (i+1,j+1)$ oraz $(i,j), (i,j+1), (i+1,j+1)$:



```

glBegin(GL_LINES);
for (int j = 0; j < N; j++)
{
    for (int i = 0; i < N; i++)
    {
        if (i - 1 >= 0)
            glVertex3f(z(KwJed[i - 1][j].u, KwJed[i][j].v), y(KwJed[i - 1][j].u,
KwJed[i][j].v), x(KwJed[i - 1][j].u, KwJed[i][j].v));
        else
            glVertex3f(z(KwJed[N - 1][j].u, KwJed[i][j].v), y(KwJed[N - 1][j].u,
KwJed[i][j].v), x(KwJed[N - 1][j].u, KwJed[i][j].v));
        glVertex3f(z(KwJed[i][j].u, KwJed[i][j].v), y(KwJed[i][j].u, KwJed[i][j].v),
x(KwJed[i][j].u, KwJed[i][j].v));

        if (i - 1 >= 0)
            glVertex3f(z(KwJed[i - 1][j].u, KwJed[i][j].v), y(KwJed[i - 1][j].u,
KwJed[i][j].v), x(KwJed[i - 1][j].u, KwJed[i][j].v));
        else
            glVertex3f(z(KwJed[N - 1][j].u, KwJed[i][j].v), y(KwJed[N - 1][j].u,
KwJed[i][j].v), x(KwJed[N - 1][j].u, KwJed[i][j].v));
        if (j - 1 >= 0)
            glVertex3f(z(KwJed[i][j].u, KwJed[i][j - 1].v), y(KwJed[i][j].u,
KwJed[i][j - 1].v), x(KwJed[i][j].u, KwJed[i][j - 1].v));
        else
            glVertex3f(z(KwJed[i][N - 1].u, KwJed[i][j].v), y(KwJed[i][N - 1].u,
KwJed[i][j - 1].v), x(KwJed[i][j].u, KwJed[i][j].v));

        if (j - 1 >= 0)
            glVertex3f(z(KwJed[i][j].u, KwJed[i][j - 1].v), y(KwJed[i][j].u,
KwJed[i][j - 1].v), x(KwJed[i][j].u, KwJed[i][j - 1].v));
        else
            glVertex3f(z(KwJed[i][j].u, KwJed[i][0].v), y(KwJed[i][j].u,
KwJed[i][0].v), x(KwJed[i][j].u, KwJed[i][0].v));
        glVertex3f(z(KwJed[i][j].u, KwJed[i][j].v), y(KwJed[i][j].u, KwJed[i][j].v),
x(KwJed[i][j].u, KwJed[i][j].v));
    }
}
glEnd();

```

3.3. Dodanie kolorów i wykorzystanie rysowania trójkątów.

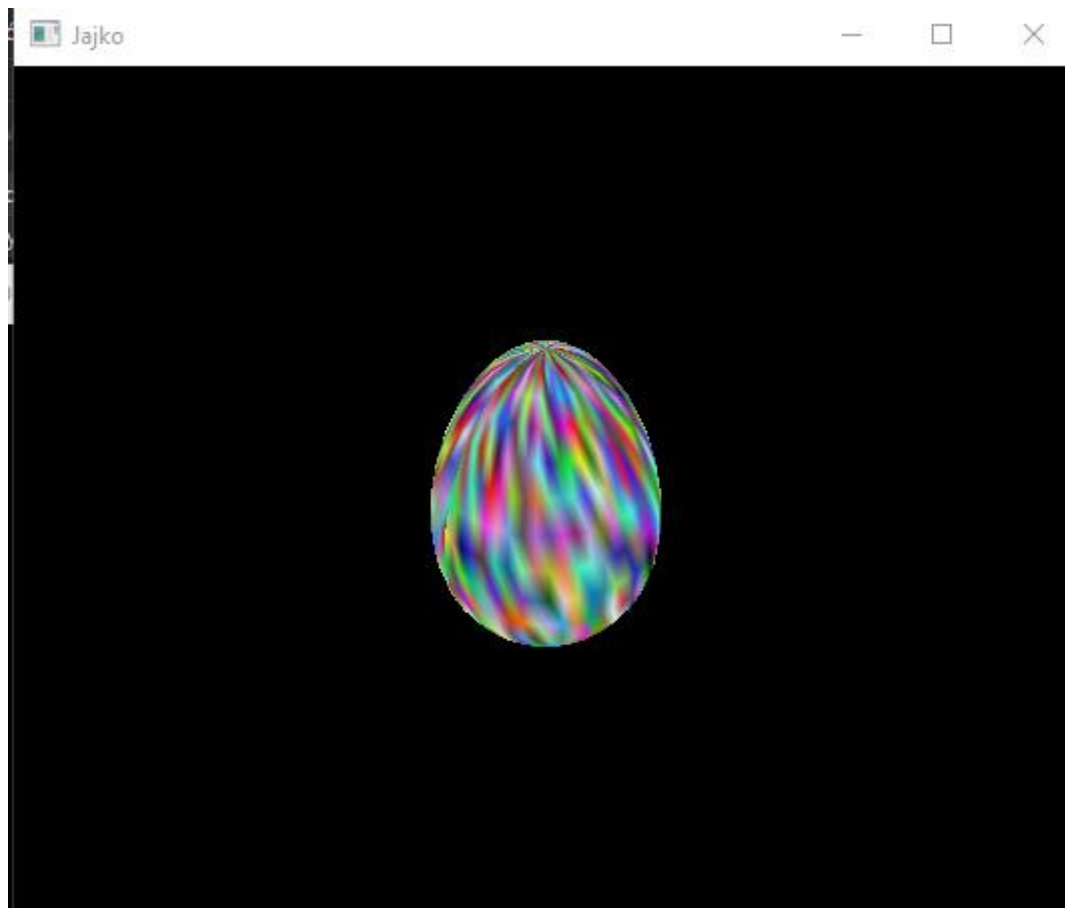
Poprzez wyznaczenie tablicy losowych kolorów, i wywoływanie rysowania wierzchołków w znany już sposób jako zostało pokolorowane. Indeksy tablicy wierzchołków odpowiadały indeksom określającym kolory danego wierzchołka. Efekt i kod prezentuje się następująco:

```

for (int j = 0; j < N; j++)
{
    for (int i = 0; i < N; i++)
    {
        glBegin(GL_TRIANGLES);
        if(i<N-1 && j<N-1)
        {
            glColor3f(colors[i][j].R, colors[i][j].G, colors[i][j].B);
            glVertex3f(x(KwJed[i][j].u, KwJed[i][j].v), y(KwJed[i][j].u,
KwJed[i][j].v), z(KwJed[i][j].u, KwJed[i][j].v));
            glColor3f(colors[i + 1][j].R, colors[i + 1][j].G, colors[i + 1][j].B);
            glVertex3f(x(KwJed[i + 1][j].u, KwJed[i + 1][j].v), y(KwJed[i +
1][j].u, KwJed[i + 1][j].v), z(KwJed[i + 1][j].u, KwJed[i + 1][j].v));
            glColor3f(colors[i + 1][j + 1].R, colors[i + 1][j + 1].G, colors[i +
1][j + 1].B);
            glVertex3f(x(KwJed[i + 1][j + 1].u, KwJed[i + 1][j + 1].v), y(KwJed[i +
1][j + 1].u, KwJed[i + 1][j + 1].v), z(KwJed[i + 1][j + 1].u, KwJed[i + 1][j + 1].v));

            glColor3f(colors[i][j].R, colors[i][j].G, colors[i][j].B);
            glVertex3f(x(KwJed[i][j].u, KwJed[i][j].v), y(KwJed[i][j].u,
KwJed[i][j].v), z(KwJed[i][j].u, KwJed[i][j].v));
            glColor3f(colors[i][j + 1].R, colors[i][j + 1].G, colors[i][j + 1].B);
            glVertex3f(x(KwJed[i][j + 1].u, KwJed[i][j + 1].v), y(KwJed[i][j +
1].u, KwJed[i][j + 1].v), z(KwJed[i][j + 1].u, KwJed[i][j + 1].v));
            glColor3f(colors[i + 1][j + 1].R, colors[i + 1][j + 1].G, colors[i +
1][j + 1].B);
            glVertex3f(x(KwJed[i + 1][j + 1].u, KwJed[i + 1][j + 1].v), y(KwJed[i +
1][j + 1].u, KwJed[i + 1][j + 1].v), z(KwJed[i + 1][j + 1].u, KwJed[i + 1][j + 1].v));
        }
    }
}
glEnd();

```



4. Wnioski

Ćwiczenie pozwoliło zapoznać się z podstawami rysowania obiektów 3D. pozwoliło zrozumieć na czym polega mapowanie UV. Ćwiczenie również wprowadziło w podstawy manipulacją obiektu. Przedstawiło różnicę między wierzchołkami obiektu 3D, a liniami i ścianami. Każde z zadań pozwoliło za stopniowe zrozumienie problematyki.

5. Zadanie domowe – układ odwzorowań iterowanych

Zadanie polegało na pobraniu pozycji myszy przy naciśnięciu klawisza, począwszy od tego punktu trzeba było narysować drzewko zgodnie z dostarczoną tablicą, poprzez losowanie. Wynikiem tych działań jest powstałe drzewko, ta technika może znaleźć zastosowanie na przykład przy generowaniu losowej roślinności.

