

# Grafika komputerowa

*Autor:*

Jacek Wieczorek (181043)

*Prowadzący:*

Dr inż. Tomasz Kapłon

Wydział Elektroniki

III rok

Pn TP 08.15 - 11.00

13 listopada 2011

# 1 Cel laboratorium

Celem ćwiczenia jest wprowadzenie w zagadnienia modelowania i wizualizacji scen 3D z wykorzystaniem biblioteki OpenGL z rozszerzeniem GLUT. Modelowanym przez nas obiektem jest jajko, opisane następującymi równaniami parametrycznymi :

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

Pętla odpowiedzialna za wyliczenie punktów jajka w przestrzeni 3D :

```
1  float u = 0.0, v=0.0;

    for(int i=0; i< N; i++)
    {
        for(int j=0; j<N; j++)
        {
            u = (float)i/(N-1);
            v = (float)j/(N-1);
            tab[i][j][0] = (-90 * pow(u,5.0f) + 225*pow(u,4.0f) - 270*pow
                (u,3.0f)+180*pow(u,2.0f)-45*u)*cos(PI * v) ;
            tab[i][j][1] = 160*pow(u,4.0f) - 320*pow(u,3.0f)+160*pow(u
                ,2.0f) - 5.0;
11         tab[i][j][2] = (-90 * pow(u,5.0f) + 225*pow(u,4.0f) - 270*
                pow(u,3.0f)+180*pow(u,2.0f)-45*u)*sin(PI * v);

        }

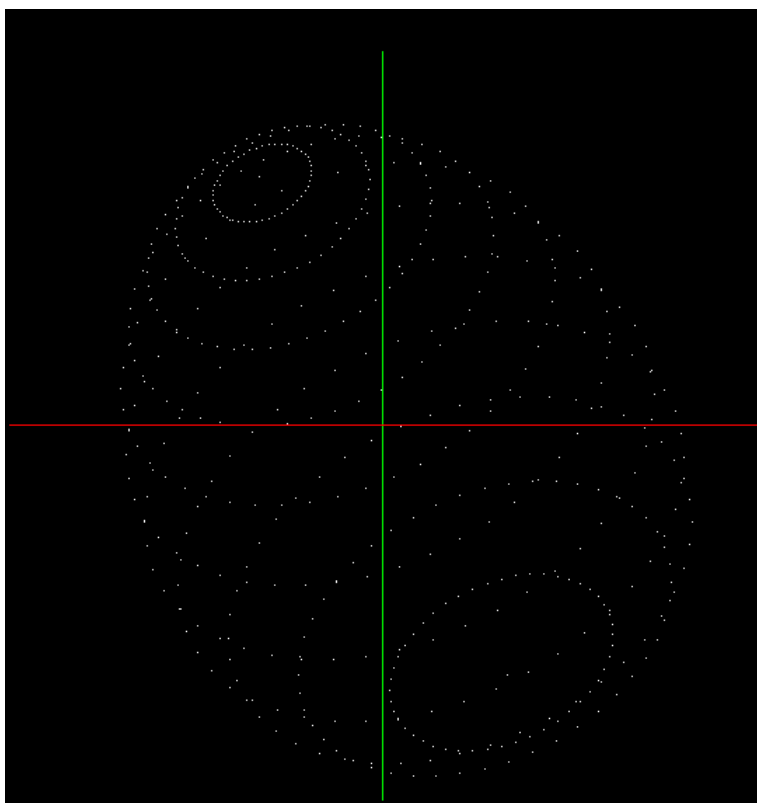
    }
```

## 2 Wyświetlanie jajka jako zbiór punktów

Pierwszym zadaniem było wyświetlenie jajka jako zbioru punktów w przestrzeni  $3D$ .

Funkcja odpowiedzialna za wyświetlanie jajka :

```
void EggsPoints() {  
    glColor3f(1.0f, 1.0f, 1.0f);  
    glBegin(GL_POINTS);  
    for(int k=0; k<N; k++)  
5    {  
        for(int l=0; l<N; l++)  
        {  
            glVertex3fv(tab[k][l]);  
        }  
    }  
    glEnd();  
}
```

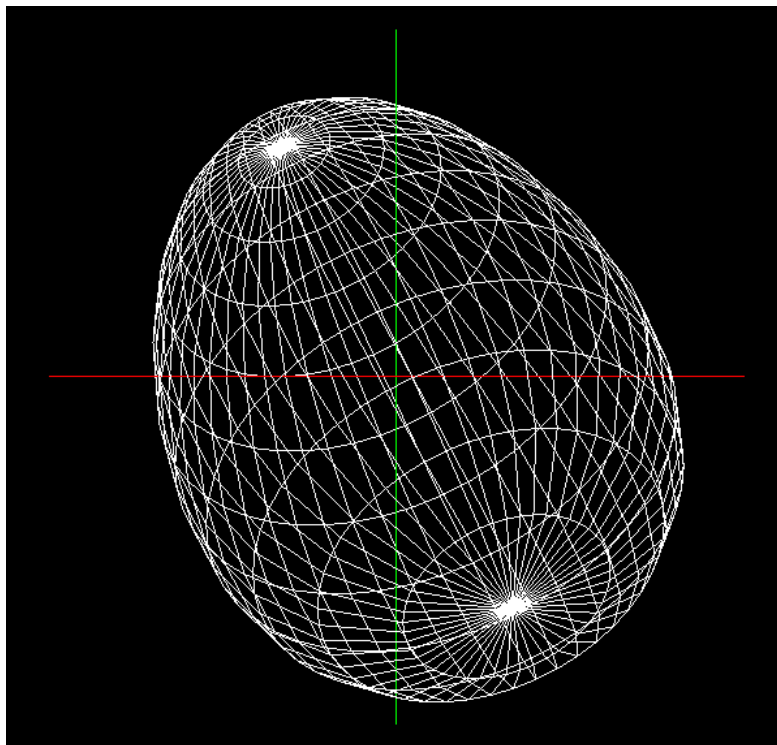


Rysunek 1: Jajko dla  $N=25$

### 3 Jajko jako siatka

Kolejnym zadaniem było połączenie punktów jajka i narysowanie go jako siatki południków i równoleżników. Zadanie wymagało połączenia ze sobą odpowiednich punktów za pomocą konstrukcji *glBegin(GL\_LINES);...glEnd()*. Funkcja odpowiedzialna za rysowanie siatki :

```
void EggsMesh() {  
    glColor3f(1.0f, 1.0f, 1.0f);  
    glBegin(GL_LINES);  
    for(int i=0; i<N-1; i++){  
        for(int j=0; j<N-1; j++){  
            {  
                glVertex3fv(tab[i][j]);  
                glVertex3fv(tab[i+1][j]);  
  
                glVertex3fv(tab[i+1][j+1]);  
                glVertex3fv(tab[i+1][j]);  
            }  
        }  
    }  
    glEnd();  
}
```



Rysunek 2: Jajko dla N=25

## 4 Jajko jako zbiór trójkątów

Ostatnim zadaniem było narysowanie jajka jako zbiór trójkątów o losowo wybranych kolorach. Metoda ta powodowała, że narysowane jajko ma tzw. "szew" biegnący wzdłuż jego połówek. W celu wyeliminowania problemu, należało wartości kolorów skrajnych trójkątów połówek jajka nadać ten sam kolor. Dotakowym problemem okazał się fakt, że dla sąsiadujące ze sobą połówek, skrajne indeksy  $i$  mają inne wartości, a mianowicie  $i \rightarrow (N - i - 1)$ . Dzięki temu otrzymywaliśmy efekt płynnego przejścia kolorów i jednolitą powierzchnię.

Funkcja losująca kolory i eliminująca szew :

```
void randCol() {
    for(int i=0; i<N; i++){
        for(int j=0; j<N; j++){
            for(int k=0; k<3; k++){
2         col[i][j][k] = (float)rand()/(float)RAND_MAX;
            }
        }
    }

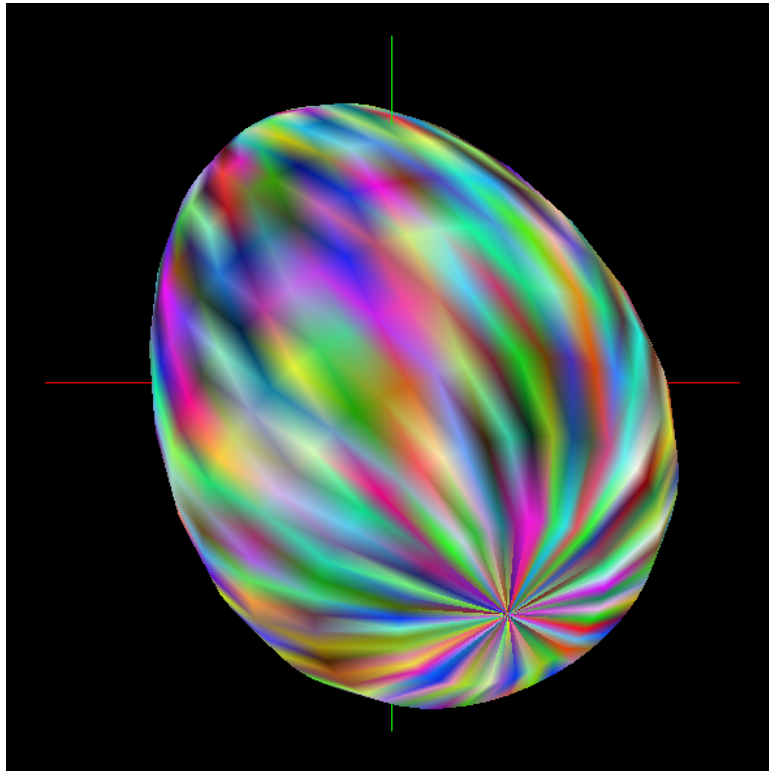
    for(int i=0; i<N; i++){
        for(int k=0; k<3; k++){
            col[i][N-1][k] = col[N-i-1][0][k];
        }
15 }
}
```

Funkcja rysująca jajko :

```
void EggsTriangles() {
    glBegin(GL_TRIANGLES);
    for(int i=0; i<N-1; i++){
        for(int j=0; j<N-1; j++){
5             {
                    glBegin(GL_TRIANGLES);
                    glColor3fv(col[i][j]);
                    glVertex3fv(tab[i][j]);
                    glColor3fv(col[i+1][j]);
                    glVertex3fv(tab[i+1][j]);
                    glColor3fv(col[i][j+1]);
                    glVertex3fv(tab[i][j+1]);
                    glEnd();

15                 glBegin(GL_TRIANGLES);
                    glColor3fv(col[i+1][j+1]);
                    glVertex3fv(tab[i+1][j+1]);
                    glColor3fv(col[i+1][j]);
                    glVertex3fv(tab[i+1][j]);
                    glColor3fv(col[i][j+1]);
                    glVertex3fv(tab[i][j+1]);
                    glEnd();

                }
            }
25         glEnd();
    }
}
```



Rysunek 3: Jajko dla  $N=25$

## 5 Dodatkowe funkcjonalności

Dodatkowe funkcjonalności zaimplementowane w programie :

- Wybór trybu jajka : trójkąty, siatka, punkty jako odpowiednio ławisz klawiatury :  $s$ ,  $w$ ,  $p$
- Ruch jajka

Funkcje odpowiadające za powyższe funkcjonalności :

```

void RenderScene(void)
{
    glClearColor(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
4    glLoadIdentity();
    Axes();
    glRotatef(theta[0], 1.0, 0.0, 0.0);
    glRotatef(theta[1], 0.0, 1.0, 0.0);
    glRotatef(theta[2], 0.0, 0.0, 1.0);
    if(model == 1)
        EggsPoints();
}

```

```

        else if(model == 2)
            EggsMesh();
        else
14         EggsTriangles();

        glFlush();
        glutSwapBuffers();
    }

    void keys(unsigned char key, int x, int y)
    {
        if(key == 'p') model = 1;
        if(key == 'w') model = 2;
24     if(key == 's') model = 3;
        RenderScene();
    }

    void spinEgg()
    {
        theta[0] -= 0.1;
        if( theta[0] > 360.0 ) theta[0] -= 360.0;

34     theta[1] -= 0.1;
        if( theta[1] > 360.0 ) theta[1] -= 360.0;

        theta[2] -= 0.1;
        if( theta[2] > 360.0 ) theta[2] -= 360.0;

        glutPostRedisplay();
    }

```

## 6 Wnioski

Laboratorium pozwoliło zapoznać się z podstawami modelowania obiektów w przestrzeni 3D. Nie jest to łatwe zadanie wymagające dużej wiedzy z zakresu matematyki i grafiki komputerowej.