

# Grafika komputerowa

*Autor:*

Tymon Tobolski (181037)

*Prowadzący:*

Dr inż. Tomasz Kapłon

Wydział Elektroniki

III rok

Pn TP 08.15 - 11.00

14 listopada 2011

# 1 Cel laboratorium

Celem laboratorium było zapoznanie się z modeowaniem obiektów w przestrzeni trójwymiarowej. Głównym zadaniem było narysowanie modelu jajka na podstawie podanych wzorów transformacji.

## 2 Jajko

Program rysuje model jajka w trzech wersjach

- pojedyncze punkty
- siatka
- kolorowe jajko złożone z trójkątów

### 2.1 Generowanie punktów w przestrzeni 3D

```
1 void calc(){
    float u = 0.0, v=0.0;

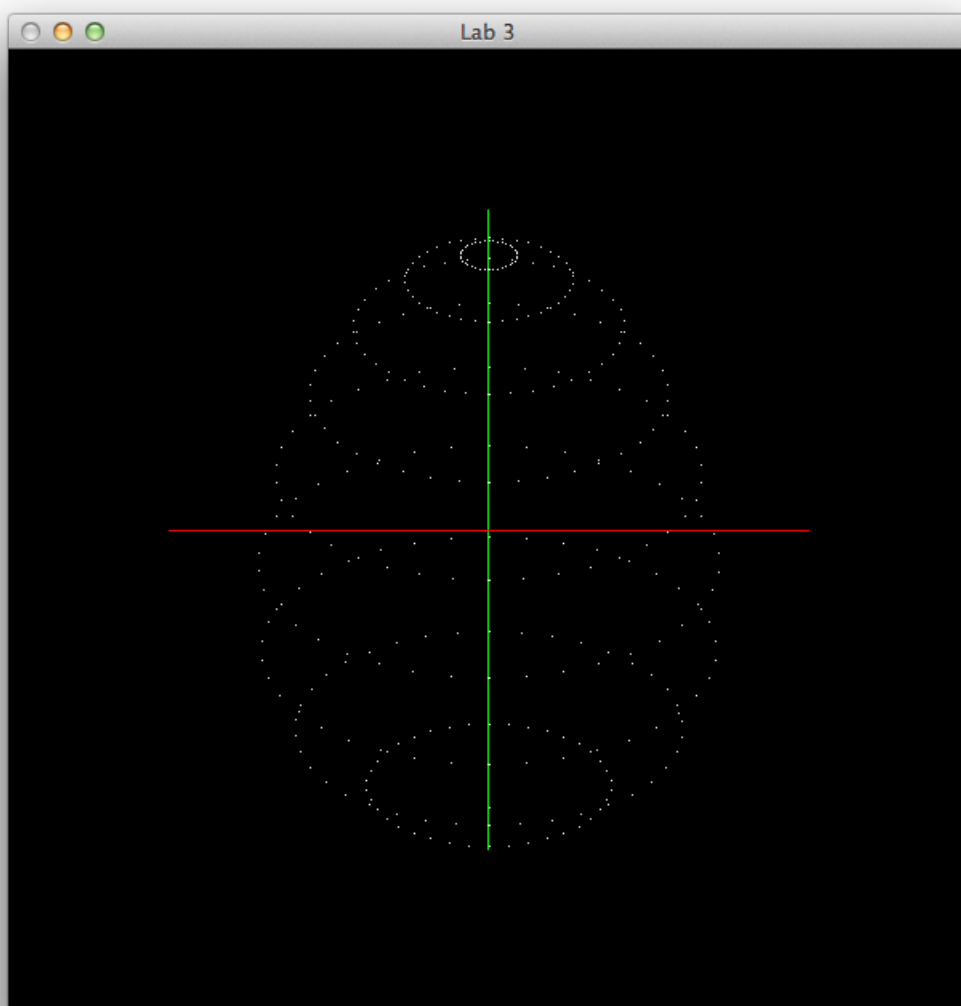
    for(int i=0; i< N; i++){
        u = ((float)i)/(N-1);
        for(int j=0; j<N; j++){
            v = ((float)j)/(N-1);
            tab[i][j][0] = (-90*pow(u, 5.0) + 225*pow(u, 4.0) - 270*pow(u,
                3.0) + 180*pow(u, 2.0) - 45*u) * cos(3.14 * v) ;
            tab[i][j][1] = 160*pow(u, 4.0) - 320*pow(u, 3.0) + 160*pow(u,
                2.0) - 5.0;
            tab[i][j][2] = (-90*pow(u, 5.0) + 225*pow(u, 4.0) - 270*pow(u,
                3.0) + 180*pow(u, 2.0) - 45*u) * sin(3.14 * v);
11         }
    }
}
```

### 2.2 Reprezentacja punktowa

```
glBegin(GL_POINTS);

for(int i=0; i<N; i++){
    for(int j=0; j<N; j++){
        glVertex3fv(tab[i][j]);
6    }
}

glEnd();
```



Rysunek 1: Reprezentacja punktowa

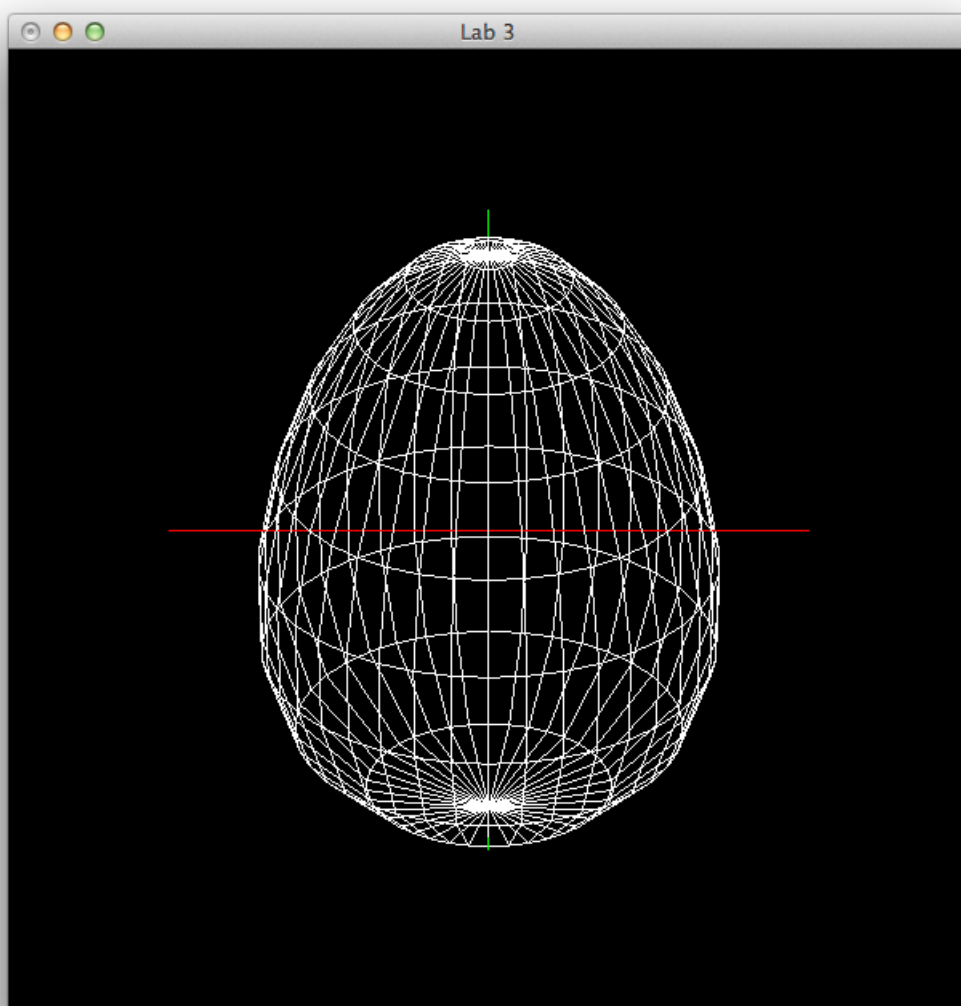
## 2.3 Siatka

```
1  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][j]);
            glVertex3fv(tab[i][j+1]);
        }
11 }

    glEnd();
```



Rysunek 2: Siatka

## 2.4 Pełne, kolorowe jajko

```
glBegin(GL_TRIANGLES);

for(int i=0; i<N/2; i++){
    for(int j=0; j<N-1; j++){
        glColor3fv(color[i][j]);
        glVertex3fv(tab[i][j]);
7
        glColor3fv(color[i+1][j]);
        glVertex3fv(tab[i+1][j]);

        glColor3fv(color[i][j+1]);
        glVertex3fv(tab[i][j+1]);

        glColor3fv(color[i][j+1]);
        glVertex3fv(tab[i][j+1]);
17
        glColor3fv(color[i+1][j+1]);
        glVertex3fv(tab[i+1][j+1]);

        glColor3fv(color[i+1][j]);
        glVertex3fv(tab[i+1][j]);
    }
}

for(int i=N-1; i>N/2; i--){
    for(int j=0; j<N-1; j++){
27
        glColor3fv(color[i][j]);
        glVertex3fv(tab[i][j]);

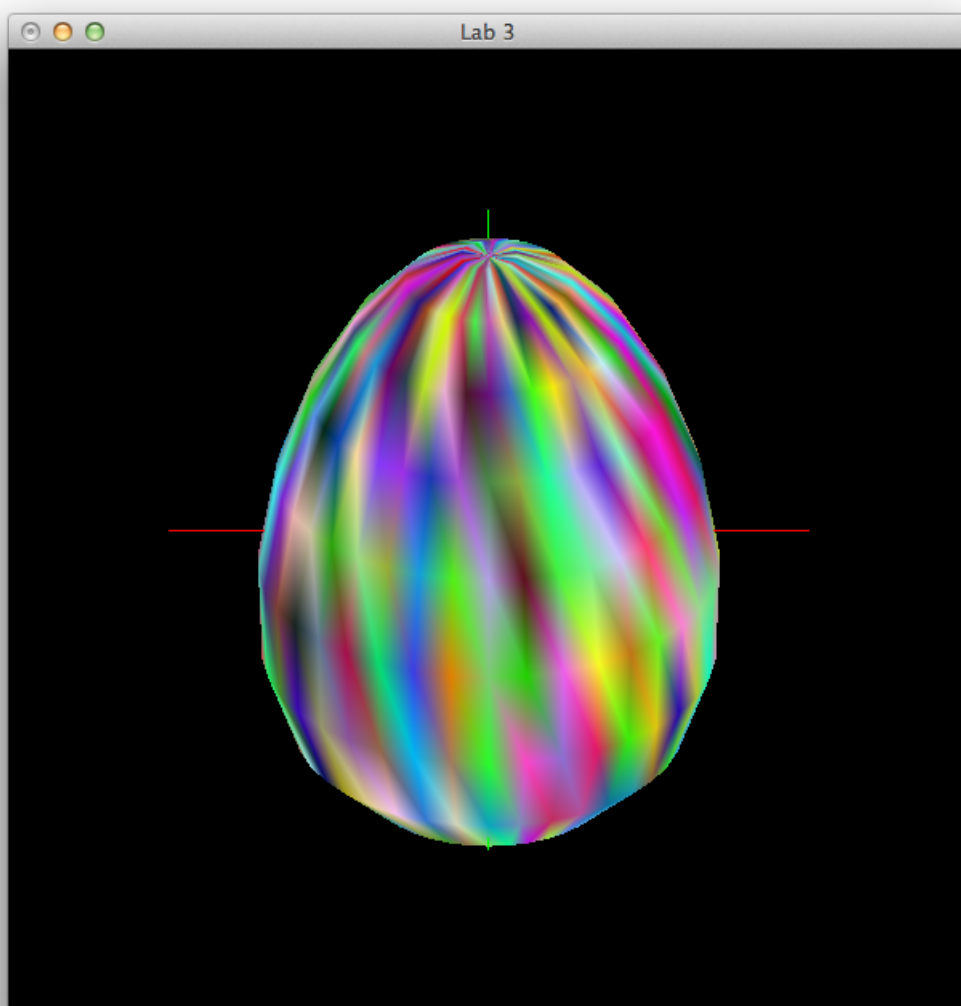
        glColor3fv(color[i-1][j]);
        glVertex3fv(tab[i-1][j]);

        glColor3fv(color[i][j+1]);
        glVertex3fv(tab[i][j+1]);
37
        glColor3fv(color[i][j+1]);
        glVertex3fv(tab[i][j+1]);

        glColor3fv(color[i-1][j+1]);
        glVertex3fv(tab[i-1][j+1]);

        glColor3fv(color[i-1][j]);
        glVertex3fv(tab[i-1][j]);
    }
}

47 glEnd();
```



Rysunek 3: Pełne, kolorowe jajko

## 2.5 Usunięcie "szwu"

Przy kolorowym jajku występował problem tzw. "szwu" - widocznej linii łączącej dwie połowy obiektu. Rozwiązaniem na to było ustawienie takich samych kolorów dla punktów leżących obok siebie.

```
void fillRandomColors(){
    for(int i=0; i<N; i++){
3        for(int j=0; j<N; j++){
            for(int k=0; k<3; k++){
                color[i][j][k] = (rand() % 101)/100.0f;
            }
        }
    }

    // usuniecie "szwu"
    for(int i=0; i<N; i++){
        for(int k=0; k<3; k++){
13        color[i][N-1][k] = color[N-i-1][0][k];
        }
    }
}
```

## 3 Wnioski

Po ustaleniu które punkty należy ze sobą połączyć aby uzyskać pełne, kolorowe jajko jedynym problemem pozostał wspomniany "szew". Udało się go wyeliminować ustalając kolory odpowiednich punktów. Poziom przybliżenia powstałych krawędzi i płaszczyzn do prawdziwego jajka zależy od parametru  $N$  - ilości punktów z których składa się wygenerowany obiekt.