

Sprawozdanie: Ćwiczenie nr. 5

Andrzej Borecki, indeks: 226205

28 stycznia 2019

1 Omówienie tematu

Na zajęciach została zrealizowana tematyka oświetlania obiektów 3D. W jej ramach wykonaliśmy oświetlenia dla imbryczka (w wersji solid) oraz jajka z poprzednich ćwiczeń. Do sprawdzania wyników pracy posłużyła interaktywna zmiana perspektywy z ostatnich laboratoriów.

2 Omówienie kodu

W moim kodzie nie pojawia się nic co nie było ujęte w instrukcji lub omówione w poprzednich sprawozdaniach. Mamy tu kolejno obracanie i zmiana odległości kamery, dzięki czemu możemy oświetlić imbrzynek i obejrzeć go dokładnie. Następnie zostało wykonane jajko, najpierw jako chmura punktów, potem siatka trójkątów z której stworzono bryłę, nadającą się do oświetlania (naiwnego).

```
void Egg()
{
    const int N = 20;
    #define PI 3.1415926535897932384626433832795

    float u = 0;
    float v = 0;
    for (int i = 0; i < N; i++)
    {
        u = i / (float)(N - 1);
        for (int j = 0; j < N; j++)
        {
            v = j / (float)(N - 1);
            vertex[i][j][0] = (-90 * pow(u, 5) + 225 * pow(u, 4) - 270 * pow(u, 3)
                + 180 * pow(u, 2) - 45 * u) * cos(PI*v);
            vertex[i][j][1] = 160 * pow(u, 4) - 320 * pow(u, 3) + 160 * pow(u, 2);
            vertex[i][j][2] = (-90 * pow(u, 5) + 225 * pow(u, 4) - 270 * pow(u, 3)
                + 180 * pow(u, 2) - 45 * u) * sin(PI*v);
            // obliczanie współrzędnych kolejnego wierzchołka według wzoru z instr

            vertexNormal[i][j][0] = (640 * pow(u, 3) - 960 * pow(u, 2) + 320u)
                * (-PI*(90 * pow(u, 5) - 225 * pow(u, 4) + 270 * pow(u, 3)
                - 180 * pow(u, 2) + 45 * u) * cos(PI * v);
            //wyzliczanie składowej wektora normalnego, w tym wypadku
```

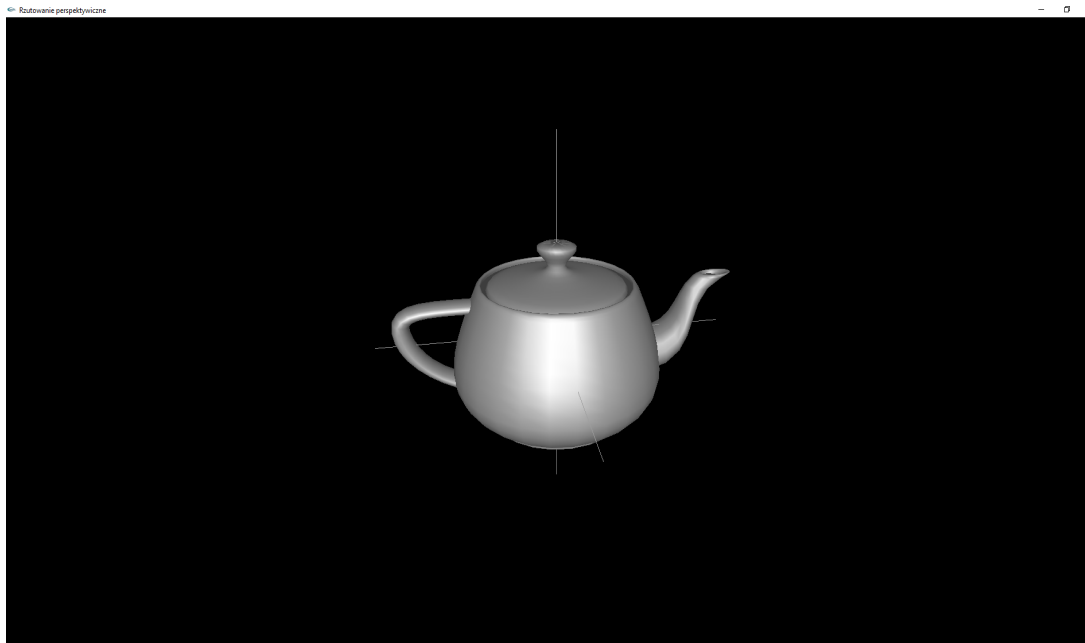
```

        // mozemy skrociac rownanie, gdyz druga czesc zawsze jest rowna 0
vertexNormal[i][j][1] = (-450 * pow(u,4) + 900 * pow(u, 3) - 810 * pow(u, 2)
+ 360 * u - 45) * sin(PI * v) * (PI *(90 * pow(u, 5) - 225 * pow(u, 4)
+ 270 * pow(u, 3) - 180 * pow(u, 2) + 45 * u) * sin(PI * v) - (-450 * pow(u, 4)
+ 900 * pow(u, 3) - 810 * pow(u, 2) + 360 * u - 45) * cos(PI * v)
* (-PI*(90 * pow(u, 5) - 225 * pow(u, 4) + 270 * pow(u, 3) - 180 * pow(u, 2)
+ 45 * u) * cos(PI * v));
vertexNormal[i][j][2] = 0 - (640 * pow(u, 3) - 960 * pow(u, 2) + 320u)
* (PI *(90 * pow(u, 5) - 225 * pow(u, 4) + 270 * pow(u, 3)
- 180 * pow(u, 2) + 45 * u) * sin(PI * v)
    }
}
glColor3f(0.0f, 1.0f, 1.0f); // kolor rysowania - bialy
glTranslated(0, -6, 0); // przesuniecie do srodka osi wsp
glScaled(1.0, 1.0, 1.0); // skalowanie w kazdej osi *1.0

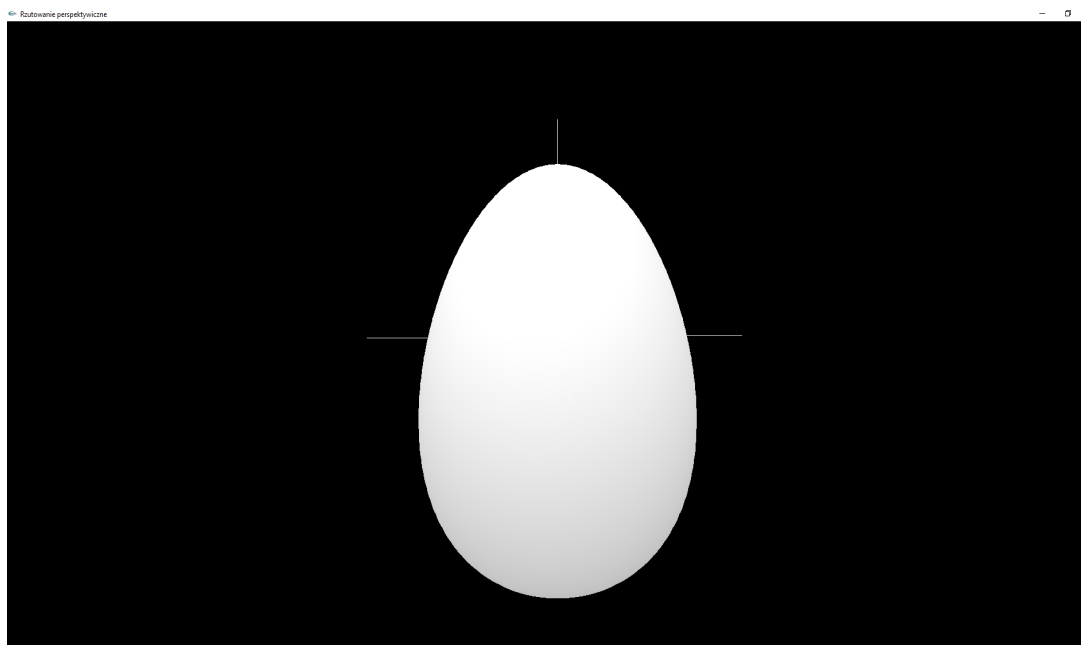
for (int i = 0; i < N; i++) {
    glBegin(GL_TRIANGLES);
    for (int j = 0; j < N ; j++) {
        glVertex3fv(vertex[i][j]);
        glVertex3fv(vertex[i + 1][j]);
    }
}
}

```

3 Rezultat prac



Rysunek 1: Oświetlony imbryczek w postaci bryły



Rysunek 2: Naiwnie oświetlone jajko