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 zaprezentowanie możliwości oświetlania obiektów scen 3-D przy użyciu biblioteki OpenGL.

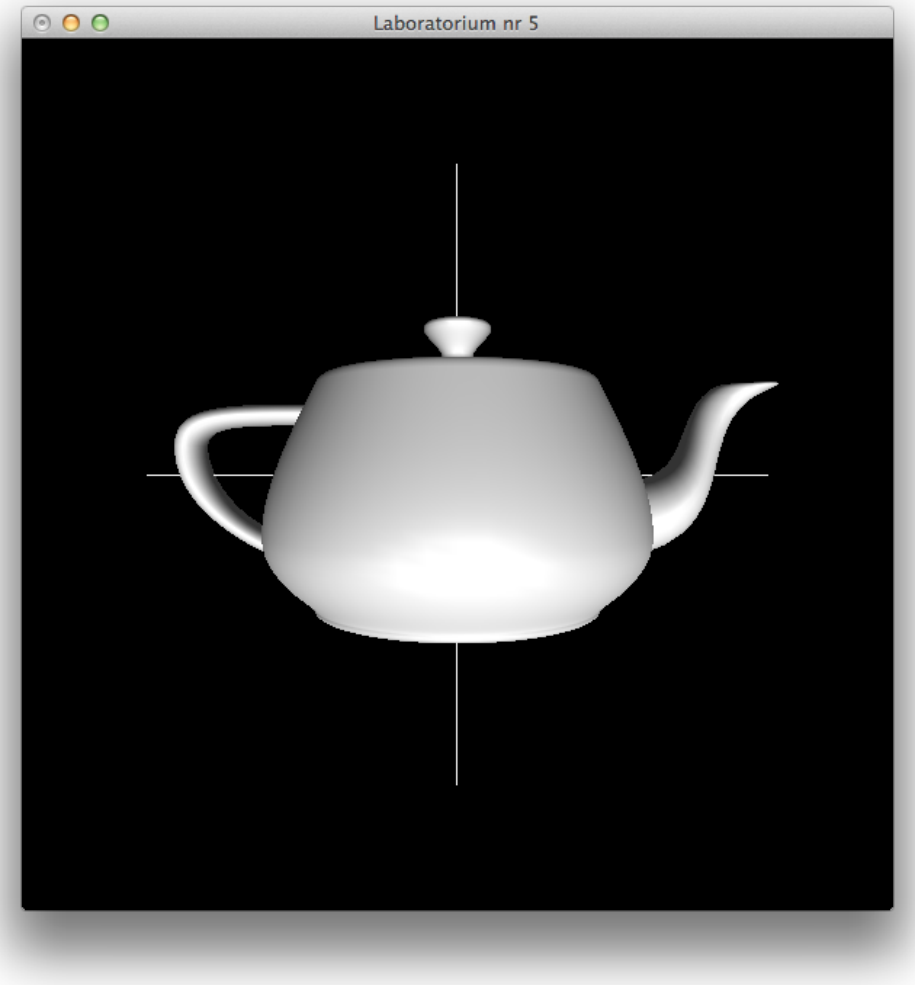
2 Wprowadzenie źródła światła

Pierwsze zadanie polegało na wyświetleniu białego czajnika oświetlonego za pomocą jednego źródła światła.

```
// Obliczenie wspolzednych punktow jajka oraz wektorow normalnych
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);

            GLfloat xu = (-450*pow(u,4) + 900*pow(u,3) - 810*pow(u,2) + 360*
                u - 45)*cos(3.14 * v);
            GLfloat yu = 640*pow(u,3) - 960*pow(u,2) + 320*u;
            GLfloat zu = (-450*pow(u,4) + 900*pow(u,3) - 810*pow(u,2) + 360*
                u - 45)*sin(3.14 * v);

            GLfloat xv = 3.14 * (90*pow(u,5) - 225*pow(u,4) + 270*pow(u,3) -
                 180*pow(u,2) + 45*u)*sin(3.14 * v);
            GLfloat yv = 0;
            GLfloat zv = -3.14 * (90*pow(u,5) - 225*pow(u,4) + 270*pow(u,3)
                - 180*pow(u,2) + 45*u)*cos(3.14 * v);

            normal[i][j][0] = yu*zv - zu*yv;
            normal[i][j][1] = zu*xv - xu*zv;
            normal[i][j][2] = xu*yv - yu*xv;

            GLfloat len = sqrt(pow(normal[i][j][0],2) + pow(normal[i][j
                ][1],2) + pow(normal[i][j][2],2));
            normal[i][j][0] /= len;
            normal[i][j][1] /= len;
            normal[i][j][2] /= len;

            if(i >= N/2) {
              normal[i][j][0] *= -1;
              normal[i][j][1] *= -1;
              normal[i][j][2] *= -1;
            }
        }
    }
}
```

```
void init(){
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

    // Definicja materialu
    GLfloat mat_ambient[]  = {1.0, 1.0, 1.0, 1.0};
    GLfloat mat_diffuse[]  = {0.8, 0.8, 0.8, 1.0};
    GLfloat mat_specular[] = {0.8, 0.8, 0.8, 1.0};
    GLfloat mat_shininess  = {50.0};

    GLfloat light0_ambient[] = {0.0, 0.0, 0.0, 1.0};
    GLfloat light0_diffuse[] = {1.0, 1.0, 1.0, 1.0};
    GLfloat light0_specular[]= {1.0, 1.0, 1.0, 1.0};

    GLfloat light1_ambient[] = {0.0, 0.0, 0.0, 1.0};
    GLfloat light1_diffuse[] = {1.0, 1.0, 0.0, 1.0};
    GLfloat light1_specular[]= {1.0, 1.0, 0.0, 1.0};

    GLfloat att_constant  = {1.0};
    GLfloat att_linear    = {0.05};
    GLfloat att_quadratic  = {0.001};

    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
    glMaterialf(GL_FRONT, GL_SHININESS, mat_shininess);

    // Ustawienie parametrow swiatla
    glLightfv(GL_LIGHT0, GL_AMBIENT, light0_ambient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light0_diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, light0_specular);
    glLightfv(GL_LIGHT0, GL_POSITION, light_pos[0]);

    glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, att_constant);
    glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, att_linear);
    glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, att_quadratic);

    glShadeModel(GL_SMOOTH); // wlaczenie lagodnego cieniowania
    glEnable(GL_LIGHTING);   // wlaczenie systemu oswietlenia sceny
    glEnable(GL_LIGHT0);     // wlaczenie zrodla o numerze 0
    glEnable(GL_LIGHT1);     // wlaczenie zrodla o numerze 1
    glEnable(GL_DEPTH_TEST); // wlaczenie mechanizmu z-bufora
}

// Rysowanie jajka
glColor3f(1.0, 1.0, 1.0);
glBegin(GL_TRIANGLES);

for(int i=0; i<N/2; i++){
    for(int j=0; j<N-1; j++){
        glColor3fv(color[i][j]);
        glNormal3fv(normal[i][j]);
        glVertex3fv(tab[i][j]);

        glColor3fv(color[i+1][j]);
        glNormal3fv(normal[i+1][j]);
        glVertex3fv(tab[i+1][j]);

        glColor3fv(color[i][j+1]);
        glNormal3fv(normal[i][j+1]);
        glVertex3fv(tab[i][j+1]);
```

```
101             glColor3fv(color[i][j+1]);
                glNormal3fv(normal[i][j+1]);
                glVertex3fv(tab[i][j+1]);

                glColor3fv(color[i+1][j+1]);
                glNormal3fv(normal[i+1][j+1]);
                glVertex3fv(tab[i+1][j+1]);

                glColor3fv(color[i+1][j]);
                glNormal3fv(normal[i+1][j]);
111             glVertex3fv(tab[i+1][j]);
            }
        }
```



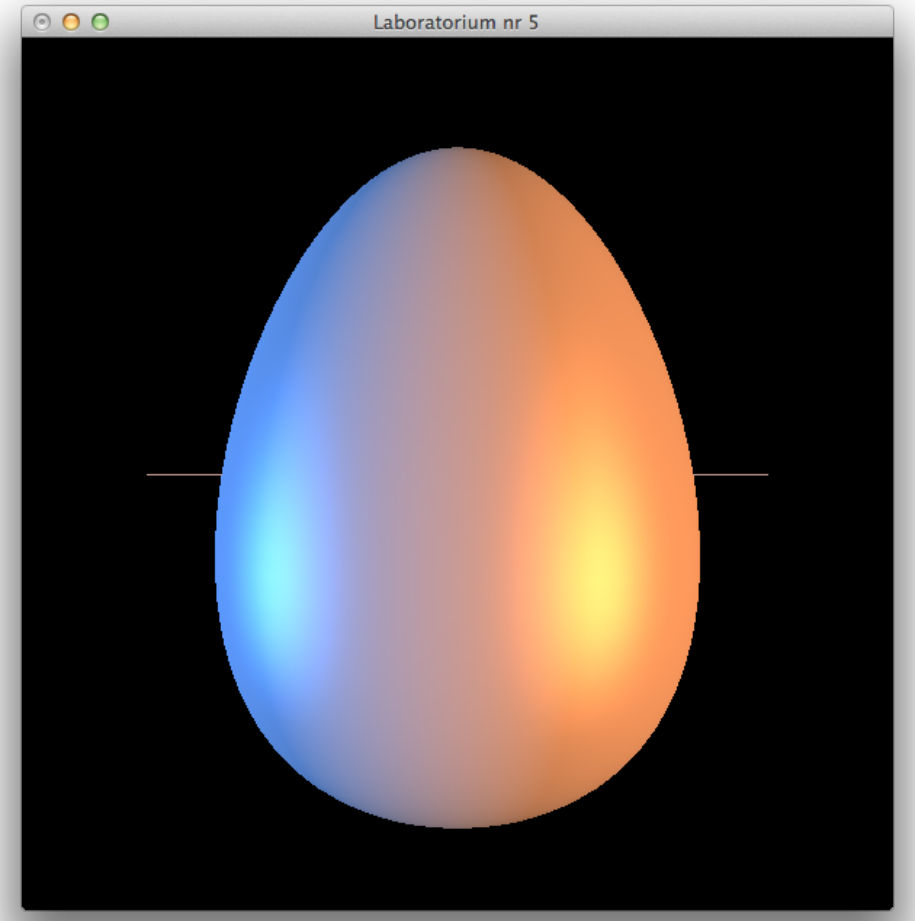
Rysunek 1: Czajnik oświetlony jednym źródłem światła

3 Wprowadzenie drugiego źródła światła

Drugie zadanie polegało na wprowadzenie kolejnego żródła światła o innym kolorze i oświetlenie modelu jajka oraz dodaniu możliwości sterowania położeniem źródeł światła za pomocą myszki.

```
void init(){
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

    GLfloat mat_ambient[]  = {1.0, 1.0, 1.0, 1.0};
    GLfloat mat_diffuse[]  = {0.8, 0.8, 0.8, 1.0};
    GLfloat mat_specular[] = {0.8, 0.8, 0.8, 1.0};
    GLfloat mat_shininess  = {50.0};

    GLfloat light0_ambient[] = {0.0, 0.0, 0.0, 1.0};
    GLfloat light0_diffuse[] = {1.0, 0.5, 0.2, 1.0};
    GLfloat light0_specular[]= {1.0, 0.5, 0.2, 1.0};

    // Ustawienie parametrow drugiego zrodla swiatla
    GLfloat light1_ambient[] = {0.0, 0.0, 0.0, 1.0};
    GLfloat light1_diffuse[] = {0.2, 0.5, 1.0, 1.0};
    GLfloat light1_specular[]= {0.2, 0.5, 1.0, 1.0};

    GLfloat att_constant  = {1.0};
    GLfloat att_linear    = {0.05};
    GLfloat att_quadratic  = {0.001};

    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
    glMaterialf(GL_FRONT, GL_SHININESS, mat_shininess);

    glLightfv(GL_LIGHT0, GL_AMBIENT, light0_ambient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light0_diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, light0_specular);
    glLightfv(GL_LIGHT0, GL_POSITION, light_pos[0]);

    glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, att_constant);
    glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, att_linear);
    glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, att_quadratic);

    // Wprowadzenie drugiego zladla swiatla
    glLightfv(GL_LIGHT1, GL_AMBIENT, light1_ambient);
    glLightfv(GL_LIGHT1, GL_DIFFUSE, light1_diffuse);
    glLightfv(GL_LIGHT1, GL_SPECULAR, light1_specular);
    glLightfv(GL_LIGHT1, GL_POSITION, light_pos[1]);

    glLightf(GL_LIGHT1, GL_CONSTANT_ATTENUATION, att_constant);
    glLightf(GL_LIGHT1, GL_LINEAR_ATTENUATION, att_linear);
    glLightf(GL_LIGHT1, GL_QUADRATIC_ATTENUATION, att_quadratic);

    glShadeModel(GL_SMOOTH); // wlaczenie lagodnego cieniowania
    glEnable(GL_LIGHTING);   // wlaczenie systemu oswietlenia sceny
```

```
        glEnable(GL_LIGHT0);      // wlaczenie zrodla o numerze 0
        glEnable(GL_LIGHT1);      // wlaczenie zrodla o numerze 1
        glEnable(GL_DEPTH_TEST); // wlaczenie mechanizmu z-bufora
    }


    // Sterowanie pozycja zrodla swiatla
    if(status == 1){
56      beta[0][0] += delta_x/40.0;
        beta[0][1] += delta_y/40.0;
    } else if(status == 2){
        beta[1][0] += delta_x/40.0;
        beta[1][1] += delta_y/40.0;
    }

    for(int b=0; b<2; b++){
        if(beta[b][1] >= M_PI) beta[b][1] -= 2*M_PI;
        else if(beta[b][1] <= -M_PI) beta[b][1] += 2*M_PI;
66
        if(beta[b][1] > M_PI/2 || beta[b][1] < -M_PI/2) yp = -1;
        else yp = 1;

        light_pos[b][0] = r*cos(beta[b][0])*cos(beta[b][1]);
        light_pos[b][1] = r*sin(beta[b][1]);
        light_pos[b][2] = r*sin(beta[b][0])*cos(beta[b][1]);
        cout << "b=" << b << " => " << beta[b][0] << "," << beta[b][1] << "," <<
             beta[b][2] << endl;

    }
76
    glLightfv(GL_LIGHT0, GL_POSITION, light_pos[0]);
    glLightfv(GL_LIGHT1, GL_POSITION, light_pos[1]);
```



Rysunek 2: Jajko oświetlone dwoma źródłami światła

4 Wnioski

Dodanie źródła światła do sceny 3-D przy wykorzystaniu biblioteki OpenGL jest stosunkowo prostym zadaniem. Największym problemem podczas realizacji laboratorium okazało się wyliczenie wektorów normalnych do powierzchni jajka.