

Лабораторная работа №6 «Источник света»

Для добавления реалистичности в проект добавим на сцену источник света. Для этого в обработчик `onCreate` необходимо дописать следующие команды:

```
glEnable(GL_LIGHTING); // разрешаем работу с освещенностью
glEnable(GL_LIGHT0); // включаем источник света
```

Это минимальные действия для включения источника света. Теперь на сцене присутствует один источник света с номером 0.

При необходимости можно "установить" несколько источников, для этого точно так же используется команда `glEnable`, например:

```
glEnable(GL_LIGHT1); // включаем источник света 1
```

Пока нет смысла использовать дополнительные источники света, это никак не повлияет на получаемые картинки, поскольку все добавляемые источники света используют установки, принятые по умолчанию, и ничем не отличаются друг от друга.

Само по себе включение источника света не приводит к ожидаемым результатам, поскольку в модели освещения для расчета интенсивности каждого пикселя используется вектор нормали к поверхности. Поэтому зададим вектора нормалей для видимых сторон куба. Вектор нормали задается командой `glNormal3f(x, y, z)`. Важны не значения координат вектора, а его направление.

```
glBegin(GL_QUADS);
    glNormal3f(0.0, 0.0, 1.0);
    glVertex3f(1.0, 1.0, 1.0);
    glVertex3f(-1.0, 1.0, 1.0);
    glVertex3f(-1.0, -1.0, 1.0);
    glVertex3f(1.0, -1.0, 1.0);
glEnd;
glBegin(GL_QUADS);
    glNormal3f(-1.0, 0.0, 0.0);
    glVertex3f(-1.0, 1.0, 1.0);
    glVertex3f(-1.0, 1.0, -1.0);
    glVertex3f(-1.0, -1.0, -1.0);
    glVertex3f(-1.0, -1.0, 1.0);
glEnd;
glBegin(GL_QUADS);
    glNormal3f(0.0, 1.0, 0.0);
    glVertex3f(-1.0, 1.0, -1.0);
    glVertex3f(-1.0, 1.0, 1.0);
    glVertex3f(1.0, 1.0, 1.0);
    glVertex3f(1.0, 1.0, -1.0);
glEnd;
```

Вектора нормалей строятся перпендикулярно каждой стороне куба. В силу того, что наш кубик строится вокруг точки (0, 0, 0), аргументы `glNormal3f` в данном случае совпадают с точкой пересечения диагоналей каждой грани куба.

Вектор нормали не обязательно должен исходить именно из середины площадки, достаточно того, чтобы он был параллелен действительному вектору нормали к площадке.

Где бы ни располагалась в пространстве площадка, она освещается единообразно.

По умолчанию источник света располагается где-то в бесконечности, поэтому освещенность площадки не меняется вместе с ее перемещением.

Задание: построить вектора нормалей к остальным сторонам куба.

```

unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms, Dialogs, OpenGL;

type
TForm1 = class(TForm)
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure FormResize(Sender: TObject);
    procedure FormPaint(Sender: TObject);
    procedure FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);

private
    hrc: HGLRC;
    DC:hdc; K:integer;
{ Private declarations } public
{ Public declarations } end;

const
    pos:array[0..3] of GLfloat=(-4,3,0,0);
    pos1:array[0..3] of GLfloat=(3,0,0,1);

procedure TForm1.FormCreate(Sender: TObject);
begin
    dc:=getdc(Handle);
    k:=1;
    SetDCPixelFormat(dc);
    hrc:=wglCreateContext(dc);
    wglMakeCurrent(dc,hrc);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glLightfv(GL_LIGHT0, GL_POSITION, @pos);
    glClearColor(0.5,0.5,0.75,1);
end;

private
    hrc: HGLRC; DC:hdc; K: integer;
procedure TForm1.FormPaint(Sender: TObject);
begin
    glClear(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT);
    glColor3f(random,random,random);
    if k=1 then begin
        glLightfv(GL_LIGHT0, GL_DIFFUSE, @yellow);
        glLightfv(GL_LIGHT0, GL_AMBIENT, @black);
    end;
    if k=2 then begin
        glLightfv(GL_LIGHT0, GL_DIFFUSE, @black);
        glLightfv(GL_LIGHT0, GL_AMBIENT, @yellow);
    end;
    glPushMatrix;
    glTranslatef(-2,0,0);
    glutSolidSphere(0.4, 50, 50);
    glPopMatrix;
    glPushMatrix;
    glTranslatef(-1,0,0);
    glutSolidSphere(0.6, 50, 50);
    glPopMatrix;
    SwapBuffers(dc);
    wglMakeCurrent(dc,hrc); // освободить контекст end;

procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
begin
    if key=49 then begin
        pos[0]:=pos[0]-0.5; // pos in var, not in const
        InvalidateRect(handle,nil,false);
    end;
    if key=50 then begin
        pos[0]:=pos[0]+0.5;
        InvalidateRect(handle,nil,false);
    end;
    if key=65 then begin //A
        k:=1;
        InvalidateRect(handle,nil,false);
    end;
    if key=66 then begin //B
        k:=2;
        InvalidateRect(handle,nil,false);
    end;
end;

```

Задание: реализовать возможность перемещения какого-либо вектора нормали по нажатию КНОПОК.