

Лабораторная работа №7 «Библиотека GLUT»

Библиотека GLUT содержит набор трехмерных объектов. Для ее использования необходимо поместить файл DGLUT.pas в папку с проектом, и в разделе implementation добавить строку `uses dglut;`

Все объекты библиотеки уже содержат встроенные вектора нормалей к поверхностям, поэтому работа с источником света будет очень простой,

В обработчик `onResize` необходимо добавить две команды, которые позволяют подключить к сцене новые матрицы: матрицу проекций и матрицу моделей. Для этого после команды `glViewport` добавляем команду

```
glMatrixMode(GL_PROJECTION);
```

и перед командой `InvalidateRect` добавить команду

```
glMatrixMode(GL_MODELVIEW);
```

Они позволят отображать объекты на сцене, используя модель освещения и встроенные вектора нормалей библиотеки `glut`.

Библиотека `glut` содержит 8 встроенных объектов: тетраэдр (TETRAHEDRON), икосаэдр (ICOSAHEDRON), додекаэдр (DODECAHEDRON), куб (CUBE), сфера (SHPERE), конус (CONE), тор (TORUS), чайник (TEAPOT).

Для работы с этими объектами объявим переменную `obj` перечислимого типа.

```
Var obj: (TETRAHEDRON, ICOSAHEDRON, DODECAHEDRON, CUBE, SHPERE, CONE, TORUS, TEAPOT) = CUBE;
```

И переменную `mode`, которая будет менять режим отображения объектов.

```
Var mode: (POINT, LINE, FTLL) = FILL;
```

В обработчике `onPaint` будет выбираться режим воспроизведения и сам объект.

```
begin
  glClear(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT);
  case mode of
    POINT: glPolygonMode(GL_FRONT_AND_BACK, GL_POINT);
    LINE: glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    FILL: glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
  end;
  case obj of
    TETRAHEDRON: glutSolidTetrahedron;
    DODECAHEDRON: glutSolidDodecahedron;
    ICOSAHEDRON: glutSolidIcosahedron;
    CUBE: glutSolidCube(1.0);
    TEAPOT: glutSolidTeapot(1.5);
    SPHERE: glutSolidSphere(1.0, 50, 50);
    TORUS: glutSolidTorus(0.5, 1.0, 50, 50);
    CONE: glutSolidCone(0.5, 1.0, 50, 50);
  end;
  SwapBuffers(DC);
End;
```

Для переключения объектов необходимо написать следующий обработчик `onKeyDown`

```
if key = 49 then //кнопка 1 begin
  inc(obj);
  if obj > High(obj) then obj = Low(obj);
  InvalidateRect(Handle, nil, False); End;
if key = 50 then //кнопка 2 begin
  inc(mode);
  if mode > High(mode) then mode = Low(mode);
  InvalidateRect(Handle, nil, False); End;
```

Задание: по нажатию кнопок организовать вращение объектов и их перемещение по сцене.
Несколько трехмерных объектов в Paint

```
glClear(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT);  
glcolor3f(random,random,random);  
glPushMatrix;  
glTranslatef(-2, 0, 0);  
glutSolidSphere(0.4, 50, 50);  
glPopMatrix;  
glPushMatrix;  
glTranslatef(-1, 0, 0);  
glutSolidSphere(0.6, 50, 50);  
glPopMatrix;  
glPushMatrix;  
glTranslatef(0.3, 0, 0);  
glutSolidSphere(0.7, 50, 50);  
glPopMatrix;  
SwapBuffers(dc);  
wglMakeCurrent (dc, hrc); // освободить контекст
```