

Министерство образования
Российской Федерации

*ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)*

Контрольная работа
По дисциплине «Компьютерная графика»
Вариант №13

Студент

2007

Цель работы

Написать программу для изображения многогранника, вращающегося вокруг оси ОУ. Ось вращения не должна совпадать с собственной вертикальной осью фигуры

13	Икосаэдр	Перспектива (1 точки схода)
----	----------	-----------------------------

При выполнении этого задания необходимо реализовать алгоритм удаления невидимых линий. Все грани рисовать закрашенными различными цветами.

Алгоритм решения задачи

Грани Икосаэдра задаются тремя точками, которые соединяются между собой линиями.

Для поворота фигуры используются матрицы поворота вокруг оси OX, OZ, OY. Для вращения объекта координатные вектора умножаются на матрицы вида:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\beta) & \sin(\beta) & 0 \\ 0 & -\sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ Матрица поворота относительно оси OX}$$

$$\begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 & 0 \\ \sin(\beta) & \cos(\beta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ Матрица поворота относительно оси OZ}$$

$$\begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ Матрица поворота относительно оси OY}$$

Для каждой грани Икосаэдра задается единичный вектор внешней нормали:

$N(b_1 \cdot c_2 - b_2 \cdot c_1, -(a_1 \cdot c_2 - a_2 \cdot c_1), a_1 \cdot b_2 - a_2 \cdot b_1)$, где $a_1, b_1, c_1, a_2, b_2, c_2$ – координаты векторов (ребер). Векторы нормалей считаются внешними, т.к. скалярные произведения соответствующих векторов нормалей и точек граней отрицательные.

Если вектор нормали к грани составляет с направление проецирования (направлением взгляда на объект), острый угол то такая грань видна (лицевая), если угол тупой, то грань не видна (не лицевая).

Если $\cos(\alpha) < 0$ ($\cos(\alpha) = (a, b) / |a| \cdot |b|$), то такой угол тупой.

Где b – вектор проецирования; a – вектор нормали к грани.

Ребро, которое находится в обоих гранях, и, причем, в одной грани оно видимо, а в другой не видимо, то считается что грань видимая.

Исходный текст программы

```
uses crt, graph;
Type
  Icosaedr=record
    PixMas:array [1..12] of record
      x,y,z:real;
      xB,yB:integer;
    end;
    Grani:array [1..20] of record
      a,b,c:integer;
    end;
  end;
Matrix=array [1..4,1..4] of real;
var
  MyStruct:Icosaedr;
  HMatr:Matrix;
  dr, mode:integer;
  Stor1, Stor2:integer;
  i:integer;

procedure Preobraz(var MyIcos:Icosaedr;Matr:Matrix);
var
  h1,h2,h3:real;
  n:integer;
begin
  for n:=1 to 12 do
    with MyIcos.PixMas[n] do
      begin
        h1:=Matr[1,1]*x+Matr[1,2]*y+Matr[1,3]*z+Matr[1,4];
        h2:=Matr[2,1]*x+Matr[2,2]*y+Matr[2,3]*z+Matr[2,4];
        h3:=Matr[3,1]*x+Matr[3,2]*y+Matr[3,3]*z+Matr[3,4];
        x:=h1;y:=h2;z:=h3;
      end;
    end;

procedure Vrashenie(var Ok:Matrix;k,l:integer);
var
  x1,x2:integer;
  i,j:integer;
begin
  for i:=1 to 4 do
    for j:=1 to 4 do
      if (i<>j) then Ok[i,j]:=0 else Ok[i,j]:=1;
      Ok[k,k]:=cos(0.1);
      Ok[k,l]:=-sin(0.1);
      Ok[l,k]:=sin(0.1);
      Ok[l,l]:=cos(0.1);
    end;

procedure IcosDraw(var MyIcos:Icosaedr);
var
  n:integer;
  R: array [1..12] of integer;
  Norm:real;
begin
  with MyIcos do
    for n:=1 to 12 do with PixMas[n] do begin
      xB:=trunc(x*200/(z-350));
      yB:=trunc(y*200/(z-350));
    end;
  SetColor(15);
  with MyIcos do
    for n:=1 to 20 do begin
```

```

        R[1]:=300+PixMas[Grani[n].a].xB; R[2]:=220-PixMas[Grani[n].a].yB;
        R[3]:=300+PixMas[Grani[n].b].xB; R[4]:=220-PixMas[Grani[n].b].yB;
        R[5]:=300+PixMas[Grani[n].c].xB; R[6]:=220-PixMas[Grani[n].c].yB;
        Norm:=(R[3]-R[1])*(R[6]-R[4])-(R[5]-R[3])*(R[4]-R[2]);
    if Norm<0 then
        begin
            SetFillStyle(1,n);
            FillPoly(3,R);
        end;
    end;
end;

begin
    Stor1:=70;
    Stor2:=100;
    for i:=1 to 12 do
        begin
            MyStruct.PixMas[i].x:=0;
            MyStruct.PixMas[i].y:=0;
            MyStruct.PixMas[i].z:=0;
        end;
        with MyStruct do begin
            PixMas[1].x:=-Stor1;    PixMas[1].z:=Stor2;
            PixMas[2].x:=Stor1;     PixMas[2].z:=Stor2;
            PixMas[3].x:=-Stor1;    PixMas[3].z:=-Stor2;
            PixMas[4].x:=Stor1;     PixMas[4].z:=-Stor2;
            PixMas[5].y:=Stor2;     PixMas[5].z:=Stor1;
            PixMas[6].y:=Stor2;     PixMas[6].z:=-Stor1;
            PixMas[7].y:=-Stor2;    PixMas[7].z:=Stor1;
            PixMas[8].y:=-Stor2;    PixMas[8].z:=-Stor1;
            PixMas[9].x:=Stor2;     PixMas[9].y:=Stor1;
            PixMas[10].x:=-Stor2;   PixMas[10].y:=Stor1;
            PixMas[11].x:=Stor2;    PixMas[11].y:=-Stor1;
            PixMas[12].x:=-Stor2;   PixMas[12].y:=-Stor1;
            with Grani[1] do begin a:=2;b:=5;c:=1;end;
            with Grani[2] do begin a:=5;b:=10;c:=1;end;
            with Grani[3] do begin a:=5;b:=6;c:=10;end;
            with Grani[4] do begin a:=9;b:=6;c:=5;end;
            with Grani[5] do begin a:=2;b:=9;c:=5;end;
            with Grani[6] do begin a:=2;b:=11;c:=9;end;
            with Grani[7] do begin a:=11;b:=4;c:=9;end;
            with Grani[8] do begin a:=9;b:=4;c:=6;end;
            with Grani[9] do begin a:=4;b:=3;c:=6;end;
            with Grani[10] do begin a:=4;b:=8;c:=3;end;
            with Grani[11] do begin a:=4;b:=11;c:=8;end;
            with Grani[12] do begin a:=11;b:=7;c:=8;end;
            with Grani[13] do begin a:=7;b:=12;c:=8;end;
            with Grani[14] do begin a:=7;b:=1;c:=12;end;
            with Grani[15] do begin a:=7;b:=2;c:=1;end;
            with Grani[16] do begin a:=11;b:=2;c:=7;end;
            with Grani[17] do begin a:=12;b:=1;c:=10;end;
            with Grani[18] do begin a:=3;b:=12;c:=10;end;
            with Grani[19] do begin a:=6;b:=3;c:=10;end;
            with Grani[20] do begin a:=12;b:=3;c:=8;end;
        end;
        Vrashenie(HMatr,2,3);
        Preobraz(MyStruct,HMatr);
        Vrashenie(HMatr,3,1);
        InitGraph(dr,mode, 'c:\bp\bgi');
        SetColor(Random(14)+1);
        OutTextXY(100,350,'Press any key');
        while not keypressed do

```

```
begin
SetColor(Random(14)+1);
OutTextXY(100,350,'Press any key');
SetBkColor(0);
ClearDevice;
IcosDraw(MyStruct);
Preobraz(MyStruct,HMatr);
delay(10000);
end;
end.
```

Заключение

В результате на Pascal 7.0 была написана программа вращения Икосаэдра. В программе был реализован алгоритм скрытия невидимых линий. Все грани Икосаэдра закрашиваются разными цветами.