ЛР №7

Планеты

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtCtrls, StdCtrls, XPMan;

const

dest=500;

des=10;

n\_v=10;

type

TVector = record

x,y,z:real;

end;

type

TPlanet = record

pos :TVector;

grad :real;

vec :real;

image :TPicture;

end;

type

TForm1 = class(TForm)

Timer1: TTimer;

Image2: TImage;

Image1: TImage;

Image3: TImage;

Panel1: TPanel;

Button1: TButton;

ComboBox1: TComboBox;

Label1: TLabel;

Button2: TButton;

OpenDialog1: TOpenDialog;

xpmnfst1: TXPManifest;

procedure Timer1Timer(Sender: TObject);

procedure FormDestroy(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

//////////3D////////

Viewpoints : array[1..n\_v] of TVector;

points : array[1..n\_v] of TVector;

newPoints : array[1..n\_v] of TVector;

Planets : array[1..n\_v] of TPlanet;

draw\_posled :array[1..n\_v] of byte;

end;

var

Form1: TForm1;

g:real;

fil:text;

v\_n,k:byte; // кол- во планет

implementation

{$R \*.dfm}

procedure TForm1.Timer1Timer(Sender: TObject);

var i,j:byte;

cs,sn:real;

min:real;

min\_pos:byte;

box,get:TRECT;

// Алгоритм вычисления черзе cos/sin.

begin

g:=0.5;

if g>2\*pi then g:=0;

for i:=1 to v\_n do

begin

planets[i].grad:=planets[i].grad+planets[i].vec;

if planets[i].grad>2\*pi then planets[i].grad:=0;

if planets[i].grad<0 then planets[i].grad:=2\*pi;

cs:=cos(planets[i].grad);

sn:=sin(planets[i].grad);

newPoints[i].x:=points[i].x\*cs+points[i].z\*sn;

newPoints[i].y:=points[i].y;

newPoints[i].z:=-points[i].x\*sn+points[i].z\*cs;

newPoints[i].x:=newPoints[i].x;

newPoints[i].y:=newPoints[i].y\*cos(g)-newPoints[i].z\*sin(g);

newPoints[i].z:=newPoints[i].y\*sin(g)+newPoints[i].z\*cos(g);

Viewpoints[i].x:=((newpoints[i].x\*dest/(dest+newpoints[i].z))\*80)+(image1.Width/2);

Viewpoints[i].y:=((newpoints[i].y\*dest/(dest+newpoints[i].z))\*80)+(image1.Height/2);

Viewpoints[i].z:=newpoints[i].z;

planets[i].pos:=Viewpoints[i];

end;

image1.Canvas.Lock;

image1.canvas.Brush.Color:=clblack;

image1.canvas.CopyRect(image1.Canvas.ClipRect,image3.Canvas,image3.Canvas.ClipRect);

image1.Canvas.Brush.Color:=clred;

image2.Canvas.Brush.Color:=clwhite;

image2.Canvas.Pen.Color:=clwhite;

// Проверка удаленности планет от Солнца

for j:=1 to v\_n do

begin

min:=planets[1].pos.z;

min\_pos:=1;

for i:= 2 to v\_n do

if planets[i].pos.z>min then

begin

min:=planets[i].pos.z;

min\_pos:=i;

end;

planets[min\_pos].pos.z:=-3000; // дистанция от солнца.

draw\_posled[j]:=min\_pos;

end;

for i:= 1 to v\_n do

begin

begin

image2.Canvas.Rectangle(image2.Canvas.ClipRect);

box.Left:=0;

box.Top :=0;

box.Right:=round((planets[draw\_posled[i]].image.Graphic.Width)\*des/(des+Viewpoints[draw\_posled[i]].z));

box.Bottom:=round((planets[draw\_posled[i]].image.Graphic.Height)\*des/(des+Viewpoints[draw\_posled[i]].z));

get.Left:=0;

get.Top:=0;

get.Right:=planets[draw\_posled[i]].image.Bitmap.Width;

get.Bottom:=planets[draw\_posled[i]].image.Bitmap.Height;

image2.Transparent:=true;

image2.Picture.Bitmap.TransparentColor:=$FFFFFF;

image2.Picture.Bitmap.Transparent:=true;

image2.Canvas.copyRect(box,planets[draw\_posled[i]].image.Bitmap.Canvas,planets[draw\_posled[i]].image.Bitmap.Canvas.ClipRect);

// Изменение размера планет - по удаленности

image2.Width:=box.Right-box.Left;

image2.Height:=box.Bottom-box.Top;

// Рисование планеты на следующей точке с Image 2

image1.Canvas.Draw(round(planets[draw\_posled[i]].pos.x-(planets[draw\_posled[i]].image.Graphic.Width div 2)\*des/(des+Viewpoints[draw\_posled[i]].z)),

round(planets[draw\_posled[i]].pos.y-(planets[draw\_posled[i]].image.Graphic.Height div 2)\*des/(des+Viewpoints[draw\_posled[i]].z)),

image2.Picture.Bitmap);

end;

end;

// Разблокируем что бы видеть конечный результат

image1.Canvas.Unlock;

end;

procedure TForm1.FormDestroy(Sender: TObject);

var

i:integer;

begin

for i:=1 to v\_n do

begin

planets[i].image.Destroy; //убиваем из памяти все динамически подгруженые рисунки

end;

end;

procedure TForm1.FormCreate(Sender: TObject);

Var i:integer;

begin

AnimateWindow(Handle, 700, AW\_BLEND); //плавное появление формы

image3.Canvas.Brush.Color:=clblack;

image3.Canvas.Rectangle(Image3.Canvas.ClipRect);

for i:=1 to 1000 do

begin

Image3.Canvas.Pixels[ random(form1.Width),

random(form1.Height)]:=$FFFFFF;

end;

k:=0;

End;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);

begin

AnimateWindow(Handle, 700, AW\_HIDE or AW\_BLEND) //плавное растворение формы

end;

procedure TForm1.Button1Click(Sender: TObject);

Var i:byte;

str:string;

begin

if k<>0 then

Begin

Image1.Canvas.Unlock;

if (Timer1.Enabled=true) then Timer1.Enabled:=false

else Timer1.Enabled:=true;

if k=2 then

Begin

Button1.Caption:='Приостановка';

k:=1;

End

else

if k=1 then

Begin

Button1.Caption:='Пуск';

k:=2;

End;

Exit;

End;

v\_n:=8;

g:=0;

AssignFile(fil,'3d.txt');

reset(fil);

readln(fil,v\_n);

for i:=1 to v\_n do

begin

readln(fil);

readln(fil,points[i].x);

readln(fil,points[i].y);

readln(fil,points[i].z);

readln(fil,planets[i].vec);

planets[i].grad:=0;

planets[i].image:=TPicture.Create;

readln(fil,str);

planets[i].image.LoadFromFile(str);

end;

CloseFile(fil);

for i:=1 to v\_n do

begin

newPoints[i]:=points[i];

Viewpoints[i].x:=round((newpoints[i].x\*100)+(form1.Width/2));

Viewpoints[i].y:=round((newpoints[i].y\*100)+(form1.Height/2));

end;

Button1.Caption:='Приостановка';

k:=1;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

if OpenDialog1.Execute then

planets[ComboBox1.ItemIndex+1].image.LoadFromFile(OpenDialog1.FileName);

end;

end.