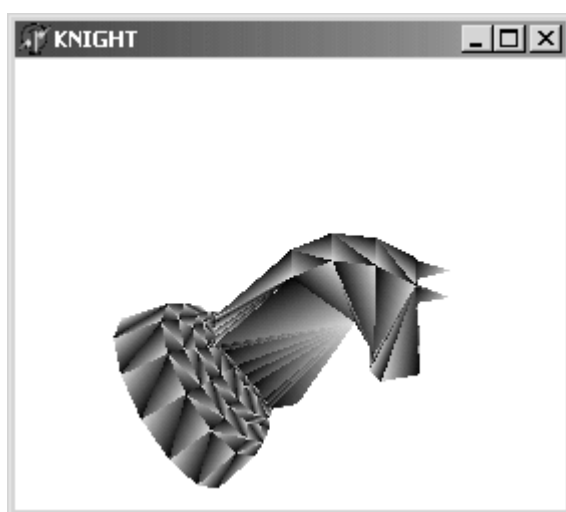


فصل سیزدهم

بارگذاری چند قالب بندی سه بعدی



مقدمه :

ذخیره سازی داده های اشیاء سه بعدی درون برنامه ، برنامه را به شدت ایستا می نماید. بارگذاری فایلی که اطلاعات اشیاء سه بعدی را درون خودش ذخیره کرده است ، به برنامه انعطاف بیشتری می بخشد . برای مثال می توان بدون کامپایل کردن مجدد برنامه ، اشیاء مختلفی را امتحان و اصلاح کرد . در ضمن گاهی اوقات لازم می باشد که اطلاعات اشیاء ترسیم شده را درون فایل ذخیره کنیم ؛ فرمت های بررسی شده در این فصل می توانند ایده های خوبی را پیش روی شما گذارند .

نوع فایل های داده ای که ما در این فصل به آن خواهیم پرداخت ، از نوع متنی بوده و به سادگی در انواع ویرایشگرهای متنی ، قابل مشاهده و اصلاح می باشند .

بررسی فرمت RAW:

در این قالب بندی ، اطلاعات رئوس مثلث ها که قسمتی از وجوه شیء ما را تشکیل می دهند در سطور متوالی ارائه می شوند . داده های این مثلث ها در این فرمت به سه صورت قابل ارائه هستند :

۱- حالت پیش فرض :

نه عدد که سه راس مثلث را در فضای سه بعدی مشخص می کنند :

Ax Ay Az Bx By Bz Cx Cy Cz

e.g. -3.5 -4.1 -5.0 -3.9 -6.2 -5.0 -3.4 -6.2 -7.0

۲- مثلث رنگی :

۱۲ عدد به ازای هر مثلث ، که سه عدد اول اجزای رنگ RGB می باشند :

R G B Ax Ay Az Bx By Bz Cx Cy Cz

e.g. 1.0 1.0 0.0 -3.5 -4.1 -5.0 -3.9 -6.2 -5.0 -3.4 -6.2 -7.0

۳- نام بافتی :

از ۹ عدد و یک نام بافت به ازای هر مثلث تشکیل شده است :

Ax Ay Az Bx By Bz Cx Cy Cz TEXTURE

e.g. -3.5 -4.1 -5.0 -3.9 -6.2 -5.0 -3.4 -6.2 -7.0 RedPlastic

در حالات فوق Ax ، Ay و غیره مختصات x ، y و z رئوس A ، B و ... مثلث ها هستند . هر عدد یا نام با یک فاصله از دیگری جدا می شود و رئوس هر مثلث تنها در یک خط وارد می شوند . در هر سه حالت ، سطر اول فایل ، از یک رشته که نام دلخواه شیء می باشد ، تشکیل شده است.

تذکر:

بدلیل اینکه در اغلب حالات ، از فرمت پیش فرض استفاده می شود و در هر فایل فقط اطلاعات یک شیء ذخیره می گردد ؛ ما در این فصل تنها به قالب بندی پیش فرض RAW ، خواهیم پرداخت و سایر حالت ها را ، خودتان بعنوان تمرین می توانید توسعه دهید .

مثالی از یک فایل RAW :

برنامه NotePad ویندوز را باز کنید و اطلاعات رئوس مثلث ها را همانند آنچه که در جدول زیر ارائه شده است ، در آن وارد کنید . سپس فایل را با پسوند raw ذخیره نمایید .

								
محتویات فایل : Obj.RAW								
my3D_obj								
0	1.1547	0	1	-0.5773	0	0	0	1.6329
0	1.1547	0	-1	-0.5773	0	0	0	1.6329
0	1.1547	0	-1	-0.5773	0	1	-0.5773	0
1	-0.5773	0	-1	-0.5773	0	0	0	1.6329

تابعی که فرمت raw را نمایش می دهد در برنامه فصل ارائه شده است .

بررسی فرمت GEM :

در این فایل متنی ، اطلاعات مربوط به رئوس و وجوه شیء ذخیره می شوند . در سطر اول آن ابتدا تعداد رئوس و سپس تعداد وجوه ، با یک فاصله (Space) از هم ، ذکر می گردند . از سطر دوم به بعد - به تعدادی که در سطر اول معین گردیده است - داده های رئوس شیء ، در هر سطر بصورت جداگانه آورده می شوند .

پس از پایان یافتن داده های رئوس ، اطلاعات مربوط به وجوه ارائه می گردد. هر وجه از به هم پیوستن رئوس داده شده ، تشکیل می شود . برای مثال :

3 1 6 4

عددی که در ابتدا می آید ، بیانگر تعداد نقاطی است که یک وجه را تشکیل می دهند . اعداد بعدی سطر ، شماره نقاطی هستند که از سطر دوم فایل به بعد ، تا ابتدای اطلاعات مربوط به وجوه ارائه شده اند و از به هم پیوستن آنها وجه مربوطه بوجود خواهد آمد.
(۱- شماره سطر = شماره نقطه)

مثالی از یک فایل GEM :

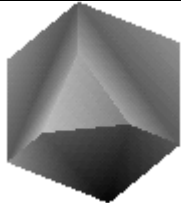
در فایل زیر :

۱- سطر اول معین می کند که شیء از ۱۰ نقطه و ۷ وجه تشکیل شده است .

۲- در سطور ۲ تا ۱۱ ، داده های مختصات رئوس ارائه شده اند .

۳- سپس اطلاعات مربوط به رئوسی که از به هم پیوستن آنها جوه شیء تشکیل خواهند شد ، آورده شده است .

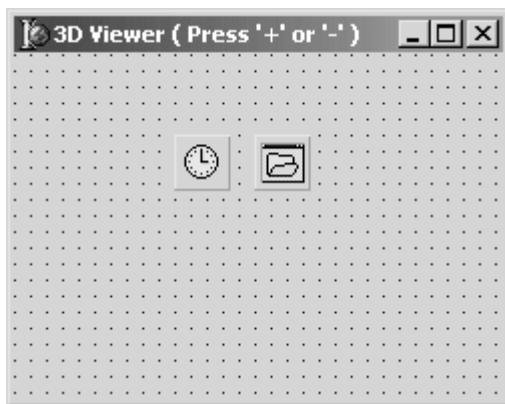
این فرمت از فرمت raw کارآتر و کم حجم تر می باشد ، زیرا داده های نقاط مشترک رئوس شیء ، در آن حذف شده اند .

					
محتویات فایل : cutCube.GEM					
10	7				
1	-1	-1			
-1	-1	-1			
-1	-1	0			
0	-1	1			
1	-1	1			
-1	0	1			
-1	1	-1			
1	1	-1			
1	1	1			
-1	1	1			
5	1	2	3	4	5
4	8	1	5	9	
4	7	8	9	10	
5	2	7	10	6	3
4	8	7	2	1	
5	5	4	6	10	9
3	3	6	4		

فرمت های مشابه بسیاری با فرمت GEM ، وجود دارند که ذکر آنها صرفنظر می شود .

برنامه فصل :

برای اجرای برنامه زیر ، کنترل های Timer و OpenFileDialog را به فرم برنامه خود اضافه کنید :
خاصیت Interval کنترل Timer را مساوی ۶۰ قرار دهید .



```

unit Ch13;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs ,OpenGL , SPF, ExtCtrls , StrUtils ;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    OpenFileDialog1: TOpenDialog;
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure FormResize(Sender: TObject);
    procedure FormPaint(Sender: TObject);
    procedure FormKeyPress(Sender: TObject; var Key: Char);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

  f_Hdc : LongInt;

implementation

{$R *.dfm}
var
  xrot : GLfloat;           // Rotates Cube On The X Axis
  yrot : GLfloat;           // Rotates Cube On The Y Axis
  fileName : string;
  Obj1  : GLuint;           // Storage For The Display List
  x_Scale : single = 1 ;
  y_Scale : single = 1 ;
  z_Scale : single = 1 ;

procedure ReadRawDefaultFormat( fileName : string );
var
  f : text ;
  Ax,Ay,Az ,Bx,By,Bz, Cx,Cy,Cz : single;
  objectName : string;

```

```

begin
  assignfile(f,fileName);
  reset(f);

  readln(f,objectName);
  form1.Caption :=  objectName  ;

  while not eof(f) do
    begin
      readln(f,Ax,Ay,Az ,Bx,By,Bz, Cx,Cy,Cz);
      glBegin(GL_TRIANGLES);
      glColor3f(0,1,0);
      glVertex3f(ax,ay,az);
      glColor3f(1,1,0);
      glVertex3f(bx,by,bz);
      glColor3f(0,0,0);
      glVertex3f(cx,cy,cz);
      glEnd();
    end;
    glFlush();
    closefile(f);
  end;

  procedure ReadGemFormat( fileName : string );
  var
    f : text ;
    Ax,Ay,Az : array of single; // dynamic arrays
    i,j , NoOfPoints , NoOfFaces : LongInt;
    VertexesToLink, NoOfPolygonVertexes : array of LongInt; //dynamic arrays
  begin
    assignfile(f,fileName);
    reset(f);

    readln(f,NoOfPoints,NoOfFaces);

    setlength(Ax,NoOfPoints);
    setlength(Ay,NoOfPoints);
    setlength(Az,NoOfPoints);
    setLength(NoOfPolygonVertexes,NoOfFaces);

    for i:=0 to High(Ax) do
      readln(f,Ax[i],Ay[i],Az[i]);

    for i:=0 to NoOfFaces-1 do
      begin
        read(f,NoOfPolygonVertexes[i]);
        setLength(VertexesToLink,NoOfPolygonVertexes[i]);

        glBegin(GL_POLYGON);
        for j:=0 to NoOfPolygonVertexes[i]-1 do
          begin
            read(f,VertexesToLink[j]);
            glColor3f(Random,Random,0);
            glVertex3f(Ax[VertexesToLink[j]-1],
                      Ay[VertexesToLink[j]-1],
                      Az[VertexesToLink[j]-1]);
          end;
        glEnd();
      end;
    end;
  end;

```

```

end;

closefile(f);
end;

procedure initGL();
begin
  glClearColor(1.0, 1.0, 1.0, 0.0);
  glEnable(GL_DEPTH_TEST);
  glDepthFunc (GL_LESS);
  glClearDepth (1);
end;

procedure DrawGLScene();
begin
  glClear(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT);

  glLoadIdentity();           // Reset The View
  glRotatef(xrot,1.0,0.0,0.0);
  glRotatef(yrot,0.0,1.0,0.0);
  glScalef(x_Scale,y_Scale,z_Scale);
  glCallList (Obj1); //efficient way !

  swapBuffers(f_Hdc);
end;

procedure TForm1.FormPaint(Sender: TObject);
begin
  wglMakeCurrent(f_Hdc,hrc); //activate the RC
  DrawGLScene();
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  OpenFileDialog1.Filter :=
    'Raw Files (*.raw)|*.raw|'+
    'Geometery Files (*.Gem)|*.Gem|'+
    'All Files (*.*)|*.*';
  if OpenFileDialog1.Execute then { Display Open dialog box }
    fileName := OpenFileDialog1.FileName { File selected in dialog }
  else
    halt(1);

  f_Hdc := GetDC(handle);
  SetDCPixelFormat(f_Hdc,16,16); // Create a rendering context.
  InitGL();

  Obj1 := glGenLists(1);
  glNewList (Obj1, GL_COMPILE);
  if LowerCase(RightStr(fileName,3))='gem' then
    ReadGemFormat(fileName);
  if LowerCase(RightStr(fileName,3))='raw' then
    ReadRawDefaultFormat(fileName);
  glEndList;
end;

procedure TForm1.FormDestroy(Sender: TObject);
begin

```

```

glDeleteLists(Obj1,1);
CleanUp(f_Hdc);// Clean up and terminate.
end;

procedure TForm1.FormResize(Sender: TObject);
begin
  wglMakeCurrent(f_Hdc,hrc); //activate the RC
  // Prevent A Divide By Zero If The Window Is Too Small
  if (Height=0)
    then Height:=1;
  glViewport(0, 0, width, Height );
  gluPerspective (30, width/ Height, 1, 100);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  if (width <= Height) then
    glOrtho(-3.0, 3.0, -3.0*Height/width,
      3.0*Height/width, -3.0, 3.0)
  else
    glOrtho(-3.0*width/Height,
      3.0*width/Height, -3.0, 3.0, -3.0, 3.0);
  glMatrixMode(GL_MODELVIEW);
  glLoadIdentity();
  InvalidateRect(Handle, nil, False);// DrawGLScene; Draw the scene.
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  yrot := yrot - 5;
  xrot := xrot + 5;
  Application.ProcessMessages; //DoEvents
  InvalidateRect(Handle, nil, False);// DrawGLScene();
end;

procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if key = '+' then
    begin
      x_Scale := x_Scale + 0.05 ;
      y_Scale := y_Scale + 0.05 ;
      z_Scale := z_Scale + 0.05 ;
    end;

  if key = '-' then
    begin
      x_Scale := x_Scale - 0.05 ;
      y_Scale := y_Scale - 0.05 ;
      z_Scale := z_Scale - 0.05 ;
    end;
end;

end.

```