

三维虚拟现实控件及其应用

屈景辉 吴晓 廖琪梅 (第四军医大学医工系 710032)

摘要:本文在介绍三维虚拟现实控件 Tego3DVR.VBX 特性的基础上,以具体实例说明如何在 Delphi 环境设计三维虚拟现实程序。

关键词:虚拟现实 Delphi 特性 设计

所谓三维虚拟现实是对三维空间的一种模拟,使观察者好象处于真正的三维空间一样。利用三维虚拟现实 VBX 控件,能够设计三维虚拟现实程序。Delphi 1.0 提供了与 VBX 控件的接口,其元件板上专门设有 VBX 页,用于放置 VBX 控件图标,菜单中设有安装 VBX 控件选项,使 VBX 控件与 Delphi 元件处于同一地位和环境之中,用户可以象使用 Delphi 元件一样使用这些控件,这为 Delphi 利用丰富的 VBX 控件开发各种应用系统提供了方便。我们以 TegoSoft 公司的三维虚拟现实控件 Tego3DVR.VBX 为例,阐述如何在 Delphi 环境下设计三维虚拟现实程序。

使用三维虚拟现实程序需要非常昂贵的设备,如与使用者身体各部分相接触的传感器、耳机和立体墨镜等。我们这里使用计算机显示器、声卡、喇叭、键盘和鼠标代替上述设备,不过效果相对要差一些。

一、三维虚拟现实控件与地图、位图文件

1. Tego3dvr 控件安装

将 Tego3DVR.VBX 文件拷入 Windows 的 System 子目录后,在集成环境选择菜单 Options/Install Components... 项,后续操作按提示进行。安装结束后,元件板的 VBX 页上出现三维虚拟现实控件图标,尔后就可以象使用 Delphi 的元件一样使用它。

2. Tego3dvr 控件特性、方法和事件

Tego3DVR.VBX 安装成功后,在 Windows 的 system 子目录中产生一个名为 tego3dvr.pas 的文件,它将 Tego3DVR.VBX 控件变成一个类对象 TTego3DVR,这个对象实现 Delphi 与 VBX 之间、使用 VBX 的应用程序与 VBX 之间的通信。从这个文件中可以获得有关控件的特性、方法及事件描述,参看表 1。

表 1 Tego3DVR 控件特性列表

特性	声明	意义
FileName	property FileName:TVBString;	二维地图文件名称
hWndDisplay	property hWndDisplay:TVBInteger;	显示三维画面的窗口句柄
NumOfRows	property NumOfRows:TVBInteger;	二维地图的行数
NumOfCols	property NumOfCols:TVBInteger;	二维地图的列数
Command	property Command:TVBString;	执行命令
ErrorCode	property ErrorCode:TVBInteger;	Command 命令执行后返回的错误代码,调用成功返回 0。
CellWidth	property CellWidth:TVBInteger;	地图中一个单元的宽度(象素)
Height3D	property Height3D:TVBInteger;	三维空间的高度
PixelPosX	property PixelPosX:TVBInteger;	观察者位置的 X、Y 坐标,以象素为单位,左下角为(0,0)
PixelPosY	property PixelPosY:TVBInteger;	观察者位置正对的方向(0-360°)
Angle	property Angle:TVBInteger;	垂直墙颜色,取值范围参看表 1-2
WallColorA	property WallColorA:TVBInteger;	水平墙颜色
WallColorB	property WallColorB:TVBInteger;	天花板颜色
CeilingColor	property CeilingColor:TVBInteger;	地板颜色
FloorColor	property FloorColor:TVBInteger;	墙壁垂直隔离条颜色
StripeColor	property StripeColor:TVBInteger;	观察者每次移动的步长(象素)
StepSize	property StepSize:TVBInteger;	对象的图像文件存放路径
SpritePath	property SpritePath:TVBString;	虚拟现实对象的编号
SpriteNumber	property SpriteNumber:TVBInteger;	虚拟现实对象的位图文件名
SpriteFileName	property SpriteFileName:TVBString;	二维地图的单元坐标,右上角(0,0)
CellX	property CellX:TVBInteger;	(1 个字符位置为 1 个单元)
CellY	property CellY:TVBInteger;	观察者所在单元位置的 xy 坐标,右上角(0,0)
CellPosX	property CellPosX:TVBInteger;	property CellPosY:TVBInteger;
CellPosY	property CellPosY:TVBInteger;	property Visible;
Visible	property Visible;	设定三维虚拟现实元件是否可见
Create 方法	Create(AOwner:TComponent);	运行阶段创建 TTego3DVR 对象
OnClick 事件	property OnClick;	执行 Command 时触发该事件

表 2 Tego3DVR 元件颜色表

值	颜色	值	颜色
0	Black	8	Gray
1	Blue	9	Light Blue
2	Green	10	Light Green
3	Cyan	11	Light Gray
4	Red	12	Light Red
5	Magenta	13	Light Magenta
6	Yellow	14	Light Yellow
7	White	15	Bright White

3. 地图文件

地图文件是三维虚拟现实控件必须的标准文本文件,它是对地图的二维描述,由文本处理软件生成。地图文件生成后,通过编程将二维地图变成三维画面,使观察者能够在三维画面中移动。

图1是用Windows记事本(Notepad.exe)生成的2010个单元地图文件。该地图中有三个房间,房间与房间之间有过道,地图的几何形状由数字字符(0-9)围成。

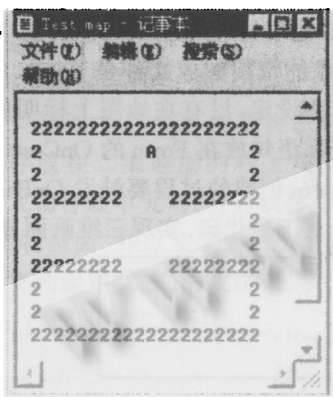


图1 3DVR元件使用的地图文件样板

4. 对象的图像文件

地图文件中的字符A代表三维空间中一个被称为对象的物体,如一张桌子、一盏灯或者一盆花等。对象可在编辑地图文件时设定,也可在程序运行中设定。对象分为软对象和硬对象两种。所谓软对象是指观察者可以穿过的,如房间中吊一盏灯,观察者可以从灯底下走过;硬对象是指观察者不可穿过的,如观察者面前的一张桌子,观察者不能直接穿过而必须绕过去。

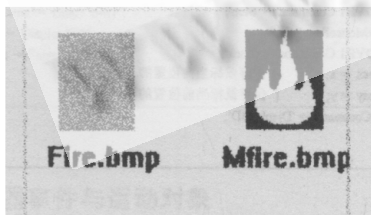


图2 对象的位图文件和掩码位图文件

一个对象有两幅与它相关的BMP格式文件,一幅是对象的位图文件如图2的Fire.bmp,另一幅是对象的掩

码位图(Mask Bitmap)文件如图2的Mfire.bmp。对象的位图文件是对象的实际图像,掩码位图文件确定对象的图像哪些区域是透明的,哪些是不透明的,透明区域用黑色表示,非透明区域用白色表示。这两种位图文件必须是256色的BMP格式。

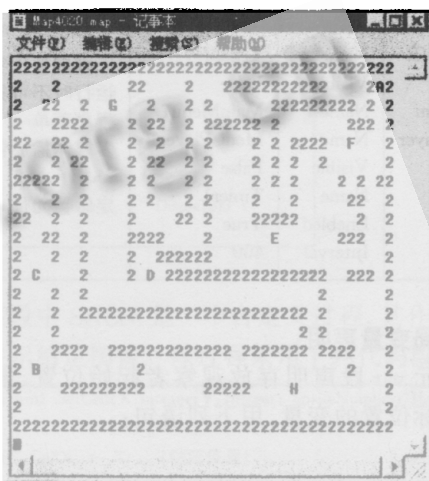


图3 迷宫寻宝二维地图文件MAP4020.MAP

对象的每一幅图像文件须有一个掩码文件与其对应,即Mfire.BMP与Fire.BMP对应,掩码文件的文件名必须以M打头,后跟对象的图像文件名,如Fire.BMP和Mfire.BMP。

二、程序设计方法

现在,我们利用Tego3DVR元件设计一个简单的游戏——迷宫寻宝。用Notepad字处理程序设计迷宫二维地图文件MAP4020.MAP如图3所示。地图由4020单元组成,图中A、B...H代表8幅风景画,A单元是所谓的财宝存放处。

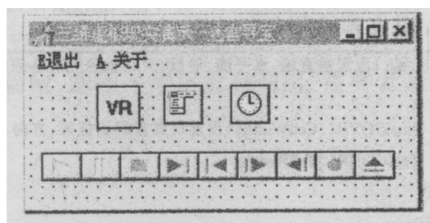


图4 迷宫寻宝设计阶段界面

1. Form 中元件及特性设置

迷宫寻宝游戏设计阶段界面如图4所示(设计阶段Form的大小与三维画面大小无关),按表3设置Form中的元件及特性。

表3 迷宫寻宝游戏Form中元件及特性设置

元件	特性	设定值	说明
TTego3DVR	Name Visible	Tego3DVR1 False	三维虚拟现实元件 运行时不可见
TMenuItem	Name	MainMenu1	程序菜单
TMediaPlayer	Name Visible	MediaPlayer1 False	多媒体播放器元件
Timer	Name Enabled Interval	Timer1 True 400	定时器元件

2. 全局变量声明

在单元var段声明存放观察者起始位置、鼠标状态和上次鼠标位置的变量,用下列语句:

```
var
  StartPosX, StartPosY: Integer;      (观察者起始位置)
  MouseState: Boolean;                (鼠标状态)
  PrevMousePosX, PrevMousePosY: Integer; (上次鼠标位置)
```

3. 初始三维画面设置

在三维虚拟空间出现前,首先要设定待打开的地图文件名称、二维地图的行列数以及显示三维画面的窗口句柄,设置Command特性为Open,用下列语句:

```
Tego3DVR1.FileName := 'MAP4020.MAP';
Tego3DVR1.NumOfRows := 20;
Tego3DVR1.NumOfCols := 40;
Tego3DVR1.hWndDisplay := Form1.Handle;
Tego3DVR1.Command := 'Open';
```

然后设定三维空间的水平、垂直墙和地板、天花板、墙壁隔离条的颜色以及观察者的位置、方向、移动时的步长等参数,用下列语句:

```
WallColorA := 4; (设置垂直墙、水平墙、地板、天花板和隔离条颜色)
.....
PixelPosX := Tego3DVR1.CellWidth; (设置观察者起始位置、方向、移动步长)
PixelPosY := Tego3DVR1.CellWidth;
Angle := 0;
StepSize := 50;
```

PixelPosX、PixelPosY是观察者的位置坐标,以像素为单位计算,坐标(0,0)位于地图的左下角,CellWidth是每个单元的像素宽度。上述语句设定观察者位于左下角第一个单元起始处,每次移动步长为50个像素。

然后,设定地图中对象的位图文件,用下列语句:

```
SpritePath := '';
SpriteNumber := 65;
SpriteFileName := 'Bmp1.BMP';
```

语句中65是字符“A”的ASCII码,意为A对象的编号是65,该对象的位图图像文件是Bmp1.BMP。其他的对象均按此方法设定,没有在地图上标明的对象由程序控制。上述全部语句放在Form的OnCreate事件中。

由于在Form创建的过程要触发OnPaint事件,所以在该事件中加入下列代码,实现三维画面显示:

```
procedure TForm1.FormPaint(Sender: TObject);
begin
  Tego3DVR1.Command := 'Display3D';
end;
```

4. 观察者位置和方向控制

所谓的观察者实际上就是操作键盘或鼠标的用户,观察者通过键盘或鼠标改变自己移动的位置和方向。如果使观察者通过鼠标控制移动,在Form的OnMouseMove事件中加入下列代码:

```
procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
  if (MouseState = False) then exit;
  if (X < PrevMousePosX) then
    Tego3DVR1.Angle := Tego3DVR1.Angle + 2;
  if (X > PrevMousePosX) then
    Tego3DVR1.Angle := Tego3DVR1.Angle - 2;
  if (Y < PrevMousePosY) then
    Tego3DVR1.Command := 'Step';
  if (Y > PrevMousePosY) then
    Tego3DVR1.Command := 'Back';
  PrevMousePosX := X; (保存鼠标当前位置的X坐标)
  PrevMousePosY := Y; (保存鼠标当前位置的Y坐标)
  Tego3DVR1.Command := 'Display3D';
  FindDelPhi;
end;
```

在这个事件处理中,首先检查鼠标状态变量MouseState的值。当MouseState为True时,执行后续处理语句,表示当鼠标键按下时再移动鼠标,移动才是有效的,否则退出该事件处理过程。所以,需根据鼠标键是否按下确定状态变量MouseState的值,在Form的OnMouseMove

Down 事件中设 MouseState 为 True, 而在 OnMouseUp 事件中设 MouseState 为 False。这样, 就完成了对鼠标操作控制。代码段中变量 PrevMousePosX、PrevMousePosY 保存着前一次鼠标所在位置, 将此位置与当前位置比较, 以确定是否前进、后退, 是向左还是向右转。新的位置确定后, 设置 Command 特性为 Display3D 重画三维画面。

观察者的位置还可通过键盘控制, 其处理过程可放在 Form 的 KeyDown 事件中。该事件的代码比较简单, 不再赘述。

5. 目标判断

在 OnMouseMove 事件中调用 FindDelPhi 过程的作用是当观察者位置改变后判断其是否已达到财宝存放区。例中宝物存放在二维地图的右上角(1, 2)单元处。当观察者到达宝物存放单元时, 打开媒体播放器, 播放声音文件, 显示宝物画面, 并以文字方式给予鼓励, 最后将观察者位置设置到起始状态, 游戏结束。

FindDelPhi 过程的完整代码如下:

```

procedure TForm1.FindDelPhi;
begin {判断位置}
if (Form1.Tego3DVR1.CellPosX = 1) and (Form1.Tego3DVR1.CellPosY = 2) then
begin { 如果进入宝物存放区, 指定波形设备和媒体文件, 然后播放}
with MediaPlayer1 do
begin
DeviceType := dtAutoSelect;
FileName := 'Clap.Wav';
Open;
Play;
end;
with Tego3DVR1 do
begin
SpriteNumber := 65; {显示宝物图像}
SpriteFileName := 'Delphi.BMP';
Command := 'Display3D';
ShowMessage('胜利属于坚持不懈的人, 祝贺您获得成功!'); {文字消息框}
PixelPosX := StartPosX; {重回初始位置}
PixelPosY := StartPosY;
Angle := 0;
SpriteNumber := 65;
SpriteFileName := 'Bmp1.BMP';
Command := 'Display3D';
MouseState := False;
end;
end;
end;

```

6. 定时器事件与运动对象

迷宫中放置了一个火苗对象, 这个火苗是活动的。SpriteNumber 73、74 和 75 是火苗燃烧时的三个不同状态, 火苗动画通过定时器 OnTimer 事件实现。用 Timer1

的 Tag 特性作计数器 Timer1.Tag 用, 根据 Timer1.Tag 的值确定对象状态的显示顺序。OnTimer 事件中的代码如下:

```

procedure TForm1.Timer1Timer(Sender: TObject);
begin
case Timer1.Tag of
0: SetCell(37, 17, 73);
1: SetCell(37, 17, 74);
2: SetCell(37, 17, 75);
end;
if Timer1.Tag = 2 then
Timer1.Tag := 0
else
Timer1.Tag := Timer1.Tag + 1;
Tego3DVR1.Command := 'Display3D';
end;

```

程序段中 SetCell 是一个自定义过程, 其作用是在指定单元显示给定图像。该过程采用下列代码段:

```

procedure TForm1.SetCell(X: integer; Y: integer; SpriteNumber: integer);
begin
Tego3DVR1.CellX := X; {设定位置}
Tego3DVR1.CellY := Y;
Tego3DVR1.SpriteNumber := SpriteNumber; {设定对象号}
Tego3DVR1.Command := 'SetCell'; {执行设定操作}
end;

```

7. 程序菜单的事件处理

菜单“关于...”项的 OnClick 事件中利用 Windows API 函数 MessageBox 显示关于 Tego3DVR.VBX 控件的有关信息, “退出”项的 OnClick 事件中添加代码用于关闭播放器和程序窗口, 退出运行状态。

Tego3DVR.VBX 控件采用了 WinG 技术, 程序设计开始前需安装 WinG 的有关文件。WinG 是 Microsoft 公司提供的增强 Windows 图形功能技术, 以高性能方式对图形进行操作。WinG 安装时有两个选项, 仅安装“Install WinG Runtime Libraries”选项就可以。如果没有安装版的 WinG 软件, 只有文件 Wing.dll、Wing32.dll、Wingde.dll、Wingdib.drv 和 Wingpal.wnd 的话, 可直接将它们拷入 Windows 的 System 子目录。

限于篇幅, 源程序清单没有列出, 需要者请与作者联系。

(来稿时间: 1997 年 11 月)