

基于 MFC 和 OpenGL 的喷泉模拟实现

梅章明, 张秀山

(海军工程大学 计算机工程系, 湖北 武汉 430033)

摘要: 粒子系统是模拟喷泉的一种有效方法。首先使用 Visual C++ 的基本类库 MFC 和 OpenGL 图形库建立面向对象的三维图形应用程序的开发环境, 然后再利用各种计算过程生成模型中的各个体素, 利用粒子系统建模的优点, 实现喷泉的模拟。

关键词: 三维图形编程; OpenGL; MFC 类库; 粒子系统

中图分类号: TP391

文献标识码: A

文章编号: 1674-7720(2012)17-0041-03

Simulation of virtual fountain based on MFC and OpenGL

Mei Zhangming, Zhang Xiushan

(Department of Computer Engineering, Naval University of Engineering, Wuhan 430033, China)

Abstract: Particle system is an effective method in fountain simulation. In this article, at first, it uses the base class library of MFC in Visual C++ and Open GL to establish the object-oriented technology developmental environment of three-dimensional graphics and realism programming. Then, it utilizes sorts of process computing system to build the pixels of any kind in the model. Utilizing the merits of modeling technique of particle system to realize the simulation of virtual fountain.

Key words: three-dimensional graphics and realism; Open GL; MFC class library; particle system

由于自然环境中大部分景物(如云彩、火焰、烟雾、瀑布、雪花等特效)具有不规则性、复杂性与随机性,且随着时间变化形状会随之变化,对其进行逼真的实时模拟十分困难,需要大量的计算量和数据量。在虚拟环境中,自然景物的视觉效果直接影响到观察者对周围环境的感知,所以开发一个既能满足逼真度要求,又能实时显示的粒子系统是非常必要的。自从 OpenGL 公布以来,有关图形学方面的书籍、论文等资料就层出不穷,如何利用 Open GL 开发出具有一定水平的计算机图形程序就成为众多学者的追求目标。在 Visual C++ 中,既可利用 Win32 编程,也可利用 MFC 编程,两者各有特点,本文就如何利用 Open GL 在 MFC 中开发出一个简单的喷泉模拟程序作一个简单探讨。

1 OpenGL 绘图环境初始化

OpenGL 是一个跨平台的三维图形库,可在 Windows、Unix 和 Mac 等平台上运行。而 Visual C++ 完善的基本类库 MFC 和应用向导 AppWizard 使得开发一个复杂的应用程序变得轻松自如。如果将两者结合,便可开发出较高水平的 Windows 下三维图形应用程序^[1]。

在 3D 游戏的渲染过程中,传统的建模方法一般只适用于外形比较规则的形体,对于那些像雨、雪、瀑布、喷泉以及火焰等没有固定形状,甚至要随着外部环境或者其他因素的改变而改变的物质建模,传统的方法就显得无能为力了^[2]。1983 年 REEVES W T 提出了一种新的建模方法,称为模糊物体建模,该方法就是粒子系统,它的出现正好解决了上述问题^[3]。

OpenGL 函数库和操作系统无关,它有自己的独特设计,与 Windows 的图像设备接口 GDI 模型以及多数 MFC 应用程序的建立方法不太一致。在 Windows 系统中,这样的一组函数称为 wiggle 函数,每个 wiggle 函数的前缀是“wgl”。

在 Win32 下,首先必须重新设置画图窗口的像素格式,使其符合 OpenGL 对像素格式的需要。为此,声明一个 PIXELFORMATDESCRIPTOR 结构的变量,并适当设置其结构成员的值,使其支持 OpenGL 及其颜色模式。再以此变量为参数调用 ChoosePixelFormat(),分配一个像素格式号,然后调用 SetPixelFormat() 将其设置为当前像素格式。

完成了像素格式的重新设置后,需要为 OpenGL 建立绘制描述表(Render Context)。绘制描述表的作用类似于 Windows 中的设备描述表(Device Context)。只有建立了绘制描述表 RC 后,OpenGL 才能调用绘图原语在窗口中做出图形。Win32API 提供了几个操作绘制描述表的函数,包括建立、复制、使用、删除和查询等,它们都以 wgl 为词头。RC 是以线程为单位的,每一个线程必须使用一个 RC 作为当前 RC 才能执行 OpenGL 绘图原语。

wglCreateContext() 是建立绘制描述表的函数,它以一个指向 GDI 设备描述表的句柄为参数,返回一个与此设备描述表相关联的绘制描述表句柄。在以此 2 句柄为参数调用函数 wglMakeCurrent(),使 RC 成为线程当前使用的 RC,完成 Windows 下 OpenGL 绘图环境的初始化过程^[4]。

2 建立 OpenGL 单文档应用程序框架

使用 Visual C++ 的 AppWizard 和 Class Wizard 可以很容易地生成一个使用 MFC 的 OpenGL 单文档应用程序框架,名称为 MyFountain。

2.1 PreCreateWindow 方法

```

BOOL CMySDOpenGLView::PreCreateWindow
(CREATESTRUCT& cs)
{
    cs.style |= WS_CLIPCHILDREN | WS_CLIPSIBLINGS;
    return CView::PreCreateWindow(cs);
}

```

使视窗口具有 WS_CLIPCHILDREN 和 WS_CLIPSIBLINGS 风格,确保成功地设置像素格式。

2.2 添加消息响应函数

利用 MFC ClassWizard 为 CMySDOpenGLView 类添加消息 WM_CREATE、WM_DESTROY、WM_SIZE 和 WM_TIMER 的响应函数。

首先在 OnCreate 方法中初始化 OpenGL,并设置定时器。

然后在 OnTimer 响应函数中添加定时器响应函数和场景更新命令,使得程序按照定时器设置的时间步长进行中断,并调用 OnDraw 对场景进行更新、渲染。

第三步,添加 OnSize 函数对用户进行窗口调整的消息进行响应,并即时调整窗口的大小^[5]。

最后,当关闭窗口时,将值 NULL (或 0) 赋值给 wglMakeCurrent() 的参数 hRC 后,调用 wglDeleteContext() 删除绘制描述表,并删除调色板和定时器。

3 基于粒子系统的喷泉模拟

构造可视化系统的建模技术大致可以分为两类:几何建模和行为建模。几何建模处理物体的几何和形状,研究图形数据结构等基本问题;行为建模处理物体运动和行为的描述。

一个粒子系统由大量称为粒子的简单体素构成。每

个粒子有一组属性,如位置、速度、颜色和生命期。一个粒子究竟有什么样的属性,主要取决于具体的应用。粒子的初值由随机过程产生。粒子往往由位于空间的某个地方的粒子源产生。

粒子系统也利用了随机过程,并常将物体的几何和行为组合在一个有机模型中。

一个粒子系统是不断进化的。在生命期的每一刻,都要完成以下 4 步工作:

(1) 粒子源产生新粒子。产生任意数目的新粒子,它们的初始属性由随机过程控制。每个粒子都有一个生命期,如果某些粒子不应删除,则可以赋予它无限长的生命期。

(2) 更新现有粒子属性。例如,若粒子有位置和速度属性,在模拟重力场中的运动时,可以如下更新粒子的位置和速度属性:

$$\begin{cases} v=v+gt \\ s=s+vt \end{cases}$$

在该步中,粒子的生命期递减一个时间步。

(3) 删除“死”粒子。检查粒子的生命期,若为 0 则将粒子从系统中删除。

(4) 绘制粒子。显示粒子系统中所有现存的粒子。在一般情况下,粒子的几何特征十分简单,可以采用一个像素或小的多边形来代表^[6]。

3.1 粒子数据结构的定义

粒子数据结构的定义如下:

```

struct particle
{
    float t; //粒子的生命期
    float vel; //粒子运动的速度
    float dir; //粒子运动的方向
    float x, y, z; //粒子的位置坐标
    float xd, zd; //粒子的 X 和 Z 方向增加值
    char type; //粒子类型(运动或淡化)
    float a; //淡化 alpha 值
    struct particle*next, *prev;
};

```

3.2 绘制喷泉

3.2.1 先构造一个场景

由于重点是喷泉,因此简单构造一个模拟的地面能突出喷泉就可以了。实现代码如下:

```

glClear(GL_COLOR_BUFFER_BIT);
glLoadIdentity();
glBindTexture(GL_TEXTURE_2D, texture[1]);
a+=0.2;
gluLookAt(cam.x, cam.y, cam.z, 0, 0, 0, upv.x, upv.y, upv.z);

```

3.2.2 喷泉的渲染处理

喷泉的渲染处理过程主要是利用了 OpenGL 的特征

函数^[7]和方法,主要进行了两方面的处理:(1)将喷泉模型渲染成纹理文件^[8]; (2)采用透明纹理渲染技术^[9]。

3.2.3 喷泉的实现

在构造了简单的地面场景后,取以原点为中心的圆周上的均匀点序列作为喷泉的喷射点,按照上述提到的绘制方法^[10]即完成了喷泉的动态模拟。喷泉系统模拟的主要关键代码在于向内存中添加渲染粒子,即函数 AddParticles(),之后粒子将按照预定的轨道运行,其主要实现代码如下:

```
//添加新的粒子
void CMyFountainView::AddParticles()
{
    struct particle*temp;
    int i, j;

    for (j=0;j<18;j++)
        for (i=0;i<2;i++)
        {
            temp=(struct particle*)malloc(sizeof(struct
            particle));
            if (fn[j])fn[j]->prev=temp;
            temp->next=fn[j];
            fn[j]=temp;

            temp->t=-9.9; //粒子的生命期
            temp->v=(float)(rand()%200000)/100000+1;
            //粒子速度
            temp->d=(float)(rand()%400)/100-2;
            //粒子方向
            temp->x=20*cos((j*3.14159)/180); //开始
            //位置的坐标
            temp->y=0;
            temp->z=20*sin((j*3.14159)/180);
            temp->xd=cos((temp->d*3.14159)/180)*temp->v/4;
            temp->zd=sin((temp->d*3.14159)/180)*temp->v;
            temp->type=0; //粒子状态为运动
            temp->a=1; //粒子淡化
        }
}
```

喷泉的效果显示如图 1 所示。

通过改变程序中 alpha (圆的内接正多边形圆心角)的值,可以改变喷泉粒子流的股数。喷泉的粒子流粗细可通过改变矢量的乘积来实现,通过改变“vectl.x*=5; vectl.y*=5; vectl.z*=5;”等式右边的数值可以控制,图 1 就是改为“5;3;2;”的结果。

通过上述的试验比较可知,喷泉粒子流的股数和每股粒子流的粒子数目都会影响到喷泉模拟效果的真实感^[11]。

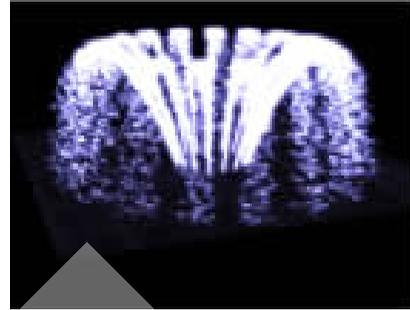


图 1 喷泉的效果显示

越来越多的人注意到使用 Visual C++ 和 OpenGL 开发三维图形动画软件的有利之处,但是有关 OpenGL 的资料大多都是介绍基本的编程指南或者一些基础的原理或方法,却很少有大型的与应用有关的编程案例,而且有也大多都是基于 Win32 的类来实现一些简单的图形功能,介绍 MFC 与 OpenGL 连接的资料却少之又少,本文主要是在 MFC 下实现了一个简单的喷泉模拟程序,主要的创新点是分析了 MFC 下消息响应的内部机制,所以希望本次的探索能对以后利用 MFC 开发出更高效的程序有所帮助。

参考文献

- [1] 和平鹤工作室.OpenGL 高级编程与可视化系统开发(高级编程篇)[M].北京:中国水利水电出版社,2007.
- [2] 徐明亮,卢红星,王琬.OpenGL 游戏编程[M].北京:机械工业出版社,2008.
- [3] REEVES W T. Particle systems—a technique for modeling a class of fuzzy objects [J]. ACM Transactions on Graphics (TOG), 1983, 2(2):359–376.
- [4] 周建龙,肖春.计算机图形学理论与 OpenGL 编程实践[M].广州:华南理工大学出版社,2007.
- [5] 孙鑫,余安萍.VC++深入详解[M].北京:电子工业出版社,2008.
- [6] 雷晓,胡倩.基于 Direct3D 的粒子系统设计[J].微计算机信息,2010(11-1).
- [7] 张芹,吴慧中,谢隽毅.基于粒子系统的火焰模型及其生产方法研究 [J]. 计算机辅助设计与图形学报, 2001, 13(1): 78–82.
- [8] PHONG B T. Illumination for computer generated pictures[J]. Communications of the ACM, 1975, 18(6):311–317.
- [9] 侯阳,迪克.三维图形动画编程实例[M].海洋出版社,1993.
- [10] 杨春雨.基于粒子系统的喷泉动画模拟[J].长春:吉林大学出版社,2008.
- [11] 凌云,储林波.用 Visual C++ 中的 MFC 和 OpenGL 建立三维图形应用环境[J].微型机与应用,1998, 17(4): 8–10.

(收稿日期:2012-05-09)

作者简介:

梅章明,男,1984年生,硕士,主要研究方向:虚拟现实研究。