

OpenGL 在视频游戏中的应用

张大强, 翟素兰, 程家兴

(安徽大学 计算智能与信号处理实验室, 安徽 合肥 230029)

摘要: OpenGL 是一个图形系统的标准软件接口, 使用 OpenGL 可以开发出高性能并能充分发挥硬件机能的图形程序。OpenGL 被广泛使用在视频游戏、三维图像制作、工业设计等多个方面, 它包含有 120 个不同的命令用于描述 3D 物体以及渲染方式。文中简述了在 Windows 平台下 OpenGL 在三维游戏开发中的地位, 及它是如何将游戏画面快速高质地显示在屏幕上的。文中详述了使用 OpenGL 绘图, `gltranslate` 函数实现对坦克的平移, 利用 OpenGL 提供的函数实现坦克的 3d 效果。通过使用 OpenGL, DirectInput 在 Windows 下设计一个 3D 游戏的构架。

关键词: 开放图形库; 直接输入; DirectX

中图分类号: TP317.4

文献标识码: A

文章编号: 1005-3751(2006)02-0073-03

Game Programing with OpenGL

ZHANG Da-qiang, ZHAI Su-lan, CHENG Jia-xing

(Key Lab. of Intelligence Computing and Signal Processing,
Anhui University, Hefei 230039, China)

Abstract: OpenGL is a software interface to graphics hardware. OpenGL can develop high performance graphic program. It is widely used in Video Game, 3D designing and industrial design. It contains 120 commands to specify the objects and operations needed to produce interactive three-dimensional applications. This paper discusses the importance of OpenGL in the development of 3D Video Game, and explains how to rapidly display pictures on the screen. It also presents the ways to design with OpenGL, to move the tank using `gltranslate` function, and to achieve 3D effect of the tank. The paper uses OpenGL to design a basic framework of 3D Video Game.

Key words: OpenGL; direct input; DirectX

0 引言

视频游戏一直是计算机技术不断发展的一个推动力, 视频游戏的设计要求开发人员有更强的性能意识。一款优秀的视频游戏在程序方面必须是使用最优化的算法并且能尽可能发挥硬件的功能。随着个人电脑图形运算机能的加强以及 Playstation2 等 3D 游戏平台的流行, 传统的 2D 游戏有逐渐被取代的趋势。文中将介绍如何使用 OpenGL 开发高性能高画质的 3D 游戏。

1 OpenGL 简介

OpenGL 是 Open Graphics Library 的缩写, 也是计算机 3D 图形学方面的工业标准, 它使用与硬件无关的设计原则, 提供命令方式的图形描述方法。现在 OpenGL 被广泛应用在各个领域, 成为最优秀的 3D 程序设计开发工具。

OpenGL 支持各个计算机平台不论在 Linux 还是 Windows 下都可以得到很好的兼容性。下面将逐步介绍

OpenGL 在 Windows 下的开发。

2 在 Windows 下使用 OpenGL

Windows 的结构并不适合运行高性能的图形程序。Microsoft 将 Windows 设计成与硬件无关的应用平台, Windows 窗口及其界面都使用 Microsoft 提供的 GDI 图形函数库进行绘制。GDI 函数库的特点是与硬件无关, 例如如果你想窗口上写几个字那么你只需要调用 `Textout` 函数即可, 而无需考虑运行程序的显卡是什么。但是这也带来了 GDI 的最大缺点——速度慢。在 Windows 平台上几乎没有使用 GDI 开发的优秀视频游戏^[1]。

在 Windows 推出后不久, Microsoft 为了让 Windows 也成为电脑游戏的理想平台, 设计并推出了 DirectX, DirectX 可以绕过 GDI 函数库直接对硬件进行操作, 因此获得了很好的性能, 也逐渐成为电脑游戏的标准。

Windows 平台也提供了对 OpenGL 的支持, OpenGL 和 DirectX 中的 Direct3D 一样可以使用显卡的硬件加速机能实现快速的光栅化、三角形填充、法线计算等功能。现阶段 PC 平台的 3D 游戏主要采用的是如图 1 所示的构架^[2]。

收稿日期: 2005-05-14

作者简介: 张大强(1980—), 男, 安徽宿州人, 硕士研究生, 研究方向为研究方向软件分析设计; 程家兴, 男, 教授, 博士生导师, 研究方向为图形识别与智能优化等。

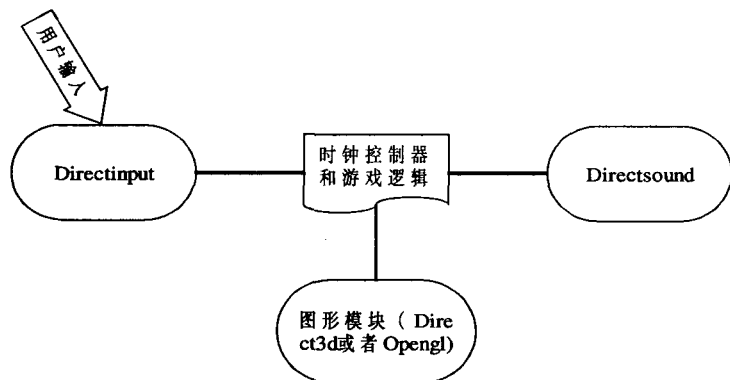


图 1 3D 游戏构架

使用 DirectInput 可以快速处理用户输入并具有缓冲功能,而使用 DirectSound 则可以将同一时间游戏中的各种声音使用软件或者硬件混合成一个声音流发送到声卡,以提高声音系统的性能。

而 Direct3D 或 OpenGL 的功能则是将游戏画面尽可能快速、高画质地显示在屏幕上。

与 Direct3D 不同 OpenGL 不是通过设备的方式控制视频硬件的,OpenGL 通过 HRC 变量将自己的绘图功能与一个 Windows 的 HDC 连接起来,在绘制图像时 OpenGL 将渲染好的图像在这个 HDC 所描述的窗口区域上显示。Windows 提供了 wglMakeCurrent 函数用于将 HRC 与 HDC 相连:

```
hdc = ::GetDC(hwnd);
hrc = ::wglCreateContext(hdc); //建立渲染 dc
wglMakeCurrent(hdc, hrc); //启动 OpenGL
```

其中 hwnd 是绘图窗口的句柄,将 OpenGL 和窗口相连后调用 ::SwapBuffers(hdc); OpenGL 的绘图结果就会显示在窗口上,SwapBuffers 的作用类似与 Direct3D 中的 Present 和 Directdraw 中的 flap。其功能是将后台缓冲区中画好的图像 copy 到位于显存中的前台缓冲区,而前台缓冲区的内容会在屏幕做扫描时显示到屏幕上,当然为了保证画面完整地显示,这个过程是在显示器做垂直同步的时候才进行的。

对于如何构建游戏的核心时钟控制器以及如何使用 DirectInput 这里不在叙述,下面将介绍如何使用 OpenGL 建立一个简单的坦克游戏的示例。

3 使用 OpenGL 绘图

一个使用 OpenGL 的程序首先要建立一个 OpenGL 的 3D 环境。这个环境使用两个矩阵来描述一个是 GL_PROJECTION 投影矩阵,还有一个是 GL_MODELVIEW 矩阵。其中 GL_PROJECTION 在建立后就不在做任何变化了,GL_PROJECTION 所定义的是 OpenGL 的虚拟视角的位置及其观察世界的各种属性。而 GL_MODELVIEW 是不停变化的,通过它的变化实现如视角转换、物体移动、物体转动等功能。

定义 OpenGL 的观察视角信息如下:

视角原点的位置为三维坐标 (0.0f, 0.0f, 0.0f), 视角观察方向为 Z 轴负方向, 投影面为 $z = -8.0f$ 面, 投影窗口大小为 $x = 1.0f, y = 0.75f$ [3]。

定义好视角之后需要定义若干个 displaylist, displaylist 是 OpenGL 提供了一种存储三维坐标顶点和转换矩阵的方式。OpenGL 通过整形数管理 displaylist, 使用 displaylist 可以优化 OpenGL 的性能。坦克使用了 4 个 displaylist, 分别对应坦克的底盘、履带、炮塔、大炮 4 个部分, 由于这 4 个部分在游戏中有相对运动, 所以分别使用不同的矩阵对它们做变换。

由于每一个 displaylist 都在坐标原点建立, 因此必须将它们移到 Z 轴的负方向才能进入视角之内。只需要使用 OpenGL 函数 glTranslate 就可以将坐标平移的矩阵乘到当前矩阵上, 从而实现对坦克的平移。

由于坦克是由 4 个子物体构成的, 按照 3 维坐标的转换原则: 子物体对与父物体的相对变换必须在父物体整体变换之前完成。所以对于构成坦克的 4 个子物体以及大地变换顺序如下:

- 1) 炮管的上下转动, 由于炮管是炮塔的子物体。
- 2) 炮塔的左右转动, 此时将炮塔和炮考虑为一个整体, 它们共同作为坦克的子物体。
- 3) 履带和轮子的转动
- 4) 坦克在地图上的移动和旋转。此时炮塔、履带和底盘都作为坦克的子物体。
- 5) 坦克、地图作为一个物体做视角的旋转。
- 6) 视角旋转后坦克、地图作为一个物体平移到 Z 轴视角之内的指定位置。

如果顺序被打乱就会出现难以预料的后果, 在 OpenGL 编程时由于先出现的矩阵会先和 MODELVIEW 矩阵相乘, 因此以上 6 步的顺序在编程时要倒过来才可以。

对于图中的阴影笔者使用的是矩阵投影的方法, 这种方式的实质是将需要绘制影子的物体(坦克)再画一遍并使用投影矩阵将其映射到大地所在的平面, 并使用 BLEND 的方法让它只显示出黑色, 从而达到影子的效果。

在游戏中坦克不但要做移动转向等运动, 而且在移动转向时也要和真实的坦克一样有左右前后的颠簸, 在开炮时也要有后坐力造成的颠簸。此时会经常造成图形上的错误, 例如坦克的前后颠簸是使用 ::glRotatef(curzdy, 0.0f, 1.0f, 0.0f); 来实现的, curzdy 表示颠簸的幅度, 这个函数会使用一个矩阵来乘以当前 MODELVIEW 矩阵, 实现对坦克以 Y 轴为中心的旋转。而此时坦克履带的一部分就会进入影子中去, 因为影子虽然能表示出颠簸的效果, 但其投影面是一个固定的平面, 履带陷入这个平面之下的部分就会在 z-buffer 检测时被消去, 因此造成坦克陷入影子中的错误图形。为此使用了 stencilbuffer 检测的方法。

在 OpenGL 中一个三角形要被显示出来需要经过多道工序的检测以保证图形的正确性,其中 stencilbuffer 的检测可以实现很多图形功能,在这里使用 stencilbuffer 来控制画面的层次,如效果图所示,当视角在大地之上时,将图中的两个物体——坦克和影子,按照首先绘制坦克,然后绘制影子的顺序绘制。结合对 stencilbuffer 的操作可以保证影子永远也不会遮挡住坦克。这种方法的原理是使用每像素 1 个 bit 的 stencilbuffer,在绘制图像前将 stencilbuffer 所有像素置 0,绘制坦克时如果某个像素被绘制,也就是说有构成坦克的多边形的点投影到这个像素,那么置这个像素点对应的 stencilbuffer 为 1。绘制完坦克绘制影子时对于在 stencilbuffer 中为 1 的像素点不去覆盖已有的值。这样凡是在画面上绘制了坦克的点就不会再绘制影子,从而避免了坦克陷入影子里面去的错误图像出现。

坦克的模型由 61908 个三角形构成,细节处也十分精美。在光影方面笔者使用的是面法线的角度平均法来计算每个顶点的法线^[2]。在 OpenGL 中由于三角形内部的点都是通过插值计算的光影效果的。因此,三角形的 3 个顶点的法线对于图像效果显得非常重要。笔者使用的算法公式如下^[4]:

$$n_v = \frac{\sum_{i=1}^n n_i a_i}{\sum_{i=1}^n a_i}$$

其中 n_i 为顶点 M 的第 i 个邻接面的法线(面的法线很好算,只需要将两边的向量叉乘即可), a_i 为构成这个面的三角形在 M 点处的角度,如图 2 所示。

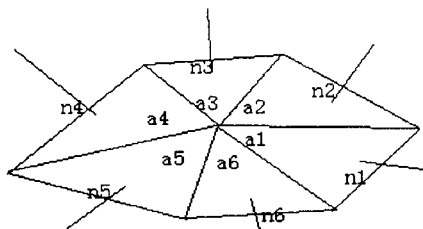


图2 点法线的计算

图中 $a1 \sim a6$ 为面三角形对应于顶点的角度, $n1 \sim n6$ 为 6 个面的法线,这种方式考虑到了每个面对顶点法线的贡献度,从而实现了图形光照平滑的效果。

计算点法线由于要对点所邻接的每个面做计算,而 3D 模型中顶点的数据又非常多,因此文中使用了离线计算的方法,将每个点的点法线计算好保留在文件中。程序只需要在读取顶点坐标的时候读取法线就可以了,而不必在游戏运行的时候去计算。

4 性能测试

游戏使用 DirectInput 控制用户输入,提供 360 度的视角水平转换、8 度到 85 度的垂直视角转换还有视角的缩放功能。

游戏使用 Windows 提供的系统时钟,以计算硬件触发次数的方式控制帧数。使用 QueryPerformanceFrequency 函数得到当前系统每秒的硬件触发数,然后使用 QueryPerformanceCounter 函数得到当前触发的序号^[5]。通过触发序号计算时间从而得到一个稳定的时钟^[6]。图 3 是游戏在 800×600 解析度下打开 smooth 选项渲染出来的图形,使用投影的方式计算阴影。游戏使用 directinput 作为用户输入实现了坦克的移动、炮塔转动、开炮火光等效果。

该游戏在赛扬 1.3G、256M 内存、gforce2mx 32M 显卡,64 位显卡上可以跑 55 帧每秒。游戏中贴图采用线性逼近的过滤方式,构建了 4 个层次的 Minimap。

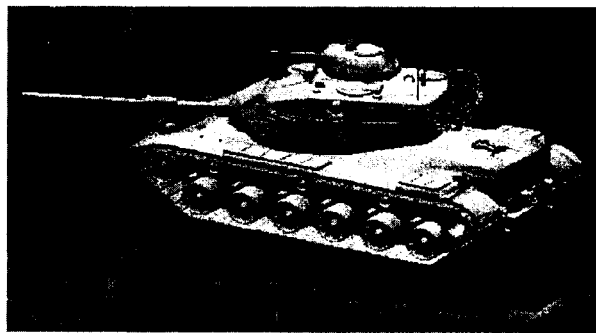


图3 利用 OpenGL 开发的坦克

5 结束语

使用 OpenGL 开发视频游戏可以更加有效地利用显卡提供的硬件加速机能,为游戏提供优秀的性能支持。而编程方面与硬件无关的特性也大大简化了开发的工作量。而且 3Dmax, maya 等三维编辑软件都使用的是与 OpenGL 相同的右手笛卡儿坐标系,便于模型的移植,缩短游戏开发时间。

不同于 Direct3D 只对 Windows 平台提供支持,OpenGL 是计算机图形方面的工业标准。在不同硬件不同平台上 OpenGL 都能得到很好的支持,使用 OpenGL 做图形开发对于跨平台开发游戏有很现实的意义。

参考文献:

- [1] Petzold C. Windows 程序设计[M]. 北京博彦科技发展有限公司译. 北京:北京大学出版社,2002.
- [2] TreGLia D. 游戏编程精粹 3[M]. 张磊译. 北京:人民邮电出版社,2003.
- [3] 张俊霞. 三维地形可视化及其实时显示方法概述[J]. 北京测绘,2001(2):6-9.
- [4] Hearn D. 计算机图形学(第3版) Computer Graphics with OpenGL 英文版[M]. 北京:电子工业出版社,2004.
- [5] Woo M, Neider J, Davis T. OpenGL 编程权威指南[M]. 吴斌,等译. 北京:中国电力出版社,2000.
- [6] OpenGL Programing Guide (Red Book) 英文版[EB/OL]. <http://www.gameres.com/>, 2002-03.