

计算机图形编程基础

华东师范大学计算机学院

王长波 教授

Win编程

■ MFC

- ☐ 微软基础类库
- ☐ MFC是一个编程框架
- ☐ 让程序员建立Windows下的应用程序
- ☐ 界面

■ Win API

图形编程

■ GDI

- 基于文档视类框架
- OnDraw()
- 调用CDC类的图元绘制函数
- 调用OnTimer制作简单动画(Invalidate)

■ DirectX

■ OpenGL

图形API——OpenGL vs. DirectX

□ OpenGL

- The Industry's Foundation for High Performance Graphics.
- 一项开放性的标准、主攻专业图形应用和3D游戏
- 最初由SGI公司制定与开发，目前由“OpenGL架构委员会(ARB)”掌控

□ DirectX

- 由Microsoft公司制定
 - 专注娱乐应用
 - 仅适用于windows平台
-

DirectX Introduction

What is DirectX

- ❑ DirectX is the Microsoft collection of APIs that are designed to give game developers a low-level interface to the PC hardware that is running Windows.
 - ❑ This interface allows developers to write their games using one set of functions, regardless of the hardware they're being run on.
-

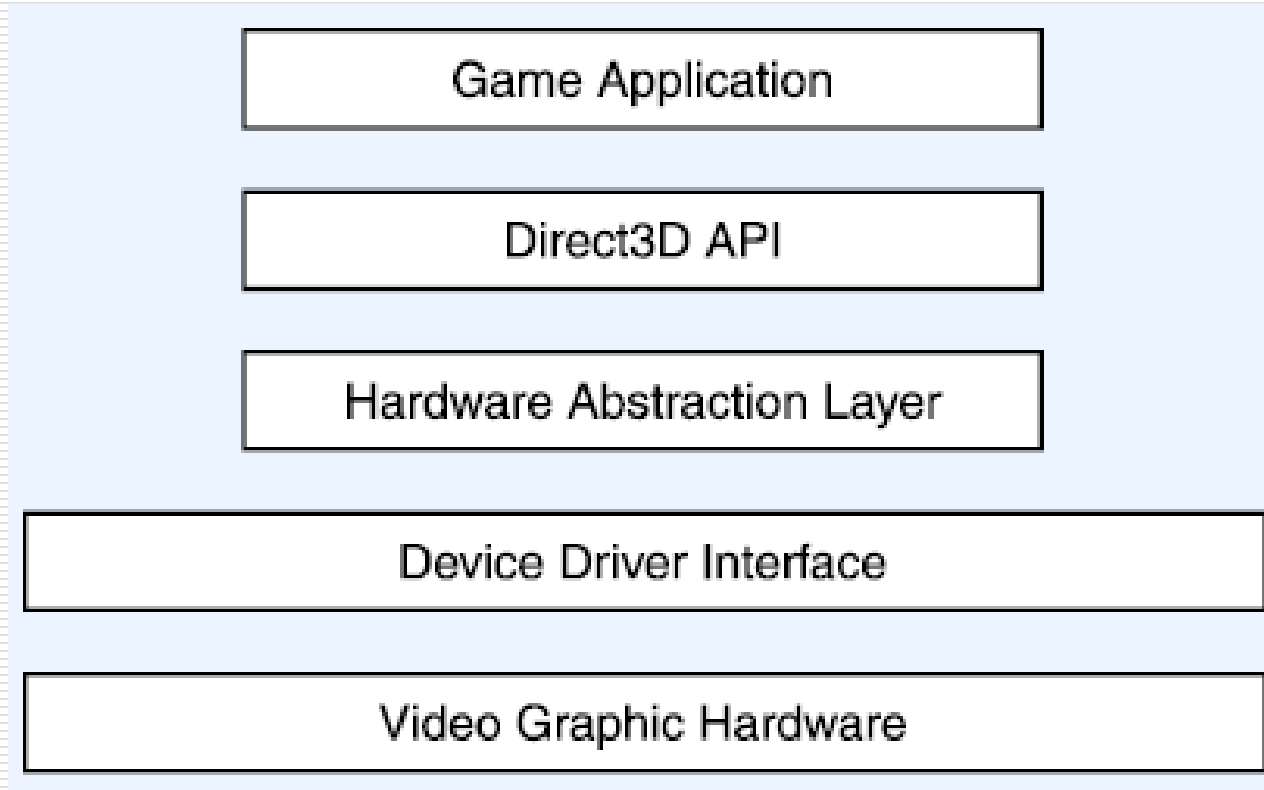
The Components

- ❑ **Direct3D - Graphics**
 - ❑ **DirectInput - Keyboards, Mice, and Joysticks**
 - ❑ **DirectSound & DirectMusic – WAV & MIDI**
 - ❑ **DirectShow - Programmable Media Player**
 - ❑ **DirectPlay - Networking**
-

The COM

- ❑ The DirectX API is based on the *Component Object Model* (COM).
 - ❑ *COM* objects consist of a collection of interfaces that expose methods that developers use to access DirectX.
 - ❑ Must RELEASE.
-

The Architecture



创建一个简单的Direct3D程序

- 创建一个Window窗口。
 - 注册WNDCLASS类: RegisterClass
 - CreateWindow(...)
 - MsgProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)
-

□ 初始化Direct3D程序

- 创建Direct3D对象: Direct3DCreate9()
 - 查询显卡的显示模式:
GetAdapterDisplayMode()
 - 创建Direct3D设备对象: CreateDevice(...)
-

□ 处理消息循环

MSG msg:

```
While(GetMessage(&msg, NULL,0,0))  
{  
    TranslateMessage(&msg)  
    DispatchMessage(&msg)  
}
```

□ 图形显示

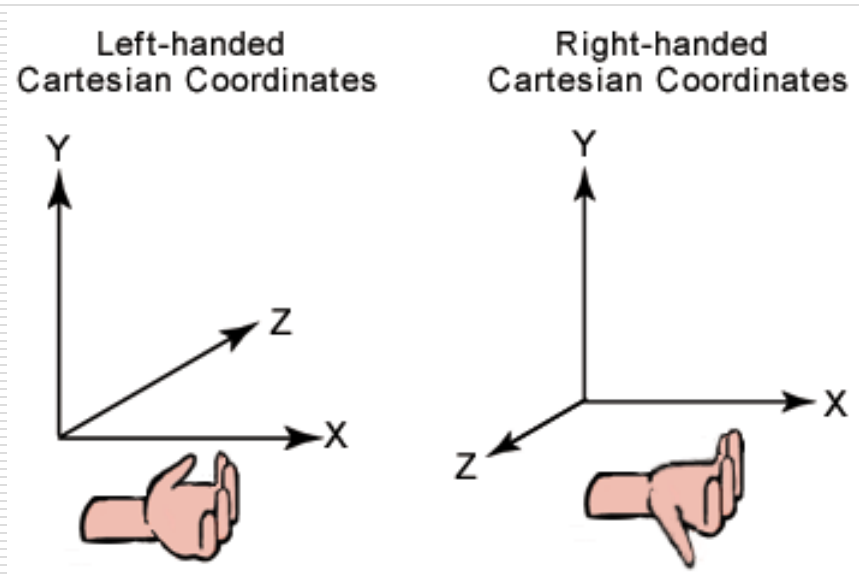
- 清屏: `IDirect3DDevice9::Clear()`
- `BeginScene()`和`EndScene()`: 不嵌套
- 更新缓冲: `IDirect3DDevice9::Present()`

□ 结束Direct3D程序

- `Cleanup()`
 - `UnregisterClass()`
-

基本几何模型显示

□ 3D坐标系



Direct3D uses left-handed, OpenGL (and most math) uses right-handed

□ 绘制图元

- 定义点的集合
 - DrawPrimitive():
 - 六种基本图元: D3DPT_POINTLIST
D3DPT_LINELIST
D3DPT_LINESTRIP
D3DPT_TRIANGLELIST
D3DPT_TRIANGLESTRIP
D3DPT_TRIANGLEFAN
-

□ 使用Vertex Buffer绘制

■ 创建:

```
CreateVertexBuffer( 6*sizeof(VERTEX),  
                    0, D3DFVF_CUSTOMVERTEX  
                    D3DPOOL_DEFAULT, &g_pBuf, NULL ) )
```

■ 加锁解锁:

```
Lock( 0, sizeof(vertices), void**)&pVertices, 0)  
Unlock();
```

□ Vertex Buffer图形显示

- 将顶点缓冲区和渲染数据流链接：
`SetStreamSource(0, g_pBuf, 0,
sizeof(VERTEX));`
 - 设置渲染的灵活的顶点格式：
`SetFVF(D3DFVF_CUSTOMVERTEX);`
-

创建一个简单的OpenGL程序

□ 创建一个OpenGL窗口。

■ 重新设置OpenGL场景的大小

```
GLvoid ReSizeGLScene(GLsizei width, GLsizei height) // 重置OpenGL
窗口大小
{ if (height==0) // 防止被零除
    { height=1; // 将Height设为1 }
    glViewport(0, 0, width, height); // 重置当前的视口
    glMatrixMode(GL_PROJECTION); // 选择投影矩阵
    glLoadIdentity(); // 重置投影矩阵 // 设置视口的大小
    gluPerspective(45.0f,(GLfloat)width/(GLfloat)height,0.1f,100.0f);
    glMatrixMode(GL_MODELVIEW); // 选择模型观察矩阵
    glLoadIdentity(); // 重置模型观察矩阵
}
```

□ 进行OpenGL环境设置

```
int InitGL(GLvoid) {  
    glShadeModel(GL_SMOOTH); // 启用阴影平滑  
    glClearColor(0.0f, 0.0f, 0.0f, 0.0f); // 黑色背景  
    glClearDepth(1.0f); // 设置深度缓存  
    glEnable(GL_DEPTH_TEST); // 启用深度测试  
    glDepthFunc(GL_LEQUAL); // 所作深度测试的类型  
    glHint(GL_PERSPECTIVE_CORRECTION_HINT,  
           GL_NICEST); // 告诉系统对透视进行修正  
    return TRUE; // 初始化 OK  
}
```

□ 开始绘制

```
int DrawGLScene(GLvoid)
// 从这里开始进行所有的绘制 {
    glClear(GL_COLOR_BUFFER_BIT |
           GL_DEPTH_BUFFER_BIT); // 清除屏幕和深度缓存
    glLoadIdentity(); // 重置当前的模型观察矩阵
    return TRUE; //一切 OK
}
```

□ 图元

- GL_TRIANGLES
- GL_QUADS

□ 图元

- glBegin(GL_TRIANGLES); // 绘制三角形
glVertex3f(0.0f, 1.0f, 0.0f); // 上顶点
glVertex3f(-1.0f,-1.0f, 0.0f); // 左下
glVertex3f(1.0f,-1.0f, 0.0f); // 右下
glEnd(); // 三角形绘制结束
-

-
- ❑ glBegin(GL_QUADS); // 绘制正方形 glVertex3f(-1.0f, 1.0f, 0.0f); // 左上 glVertex3f(1.0f, 1.0f, 0.0f); // 右上 glVertex3f(1.0f,-1.0f, 0.0f); // 左下 glVertex3f(-1.0f,-1.0f, 0.0f); // 右下 glEnd();
 - ❑ glTranslatef(3.0f,0.0f,0.0f); // 右移3单位
-

上机练习

- 利用**DirectX**的第二课(三角形)的基础上画一个四边形;
 - 或者利用**OpenGL**的第二个**Course**画一个五角星;
 - 练习利用**GDI**绘制线、圆等。
-