**武汉纺织大学**

**《Direct3D图形编程》上机实验报告**

**题目:** **椎体和柱体的绘制**

**成 绩：**

**学 号： 1604240705**

**姓 名： 胡 莲**

**班 级： 计算机11603**

**指导教师： 李 敏**

**报告日期：2019 年3月18日**

1. **实验1**
2. 题目

绘制一个底为四边形的锥体，通过键盘上的按键来控制锥体的变化：

* 按下R键(0x52)，锥体右移；按下L键(0x4C)，锥体左移；
* 按下I键(0x 49)，锥体放大（Zoom in）; 按下O键(0x 4F)，锥体缩小（Zoom out）;
* 按下X，Y，Z(0x 58, 0x 59, 0x 5A)，锥体分别绕X，Y，Z轴旋转10度。

1. 实现代码

在上次实验的基础上，修改代码：

1. 全局变量：

IDirect3DVertexBuffer9\* VB = 0;

IDirect3DIndexBuffer9\* IB = 0;

float zmX = 0.0f;

float zm = 1.0f;

float Rx = 0.0f;

float Ry = 0.0f;

float Rz = 0.0f;

1. Setup()函数：

Device -> CreateVertexBuffer(

5 \* sizeof(Vertex),

D3DUSAGE\_WRITEONLY,

Vertex::FVF,

D3DPOOL\_MANAGED,

&VB,

0);

Device -> CreateIndexBuffer(

18 \* sizeof(WORD),

D3DUSAGE\_WRITEONLY,

D3DFMT\_INDEX16,

D3DPOOL\_MANAGED,

&IB,

0);

Vertex\* vertices;

VB -> Lock(0, 0, (void\*\*)&vertices, 0);

vertices[0] = Vertex(-1.0f,0.0f,-1.0f);

vertices[1] = Vertex(0.0f, 2.0f, 0.0f);

vertices[2] = Vertex(-1.0f, 0.0f, 1.0f);

vertices[3] = Vertex(1.0f,0.0f,1.0f);

vertices[4] = Vertex(1.0f, 0.0f, -1.0f);

VB -> Unlock();

WORD\* indices = 0;

IB -> Lock(0, 0, (void\*\*)&indices, 0);

indices[0] = 0; indices[1] = 1; indices[2] = 4;//前

indices[3] = 3; indices[4] = 1; indices[5] = 2; //后

indices[6] = 0; indices[7] = 2; indices[8] = 1; //左

indices[9] = 4; indices[10] = 1;indices[11] = 3; //右

indices[12] = 0;indices[13] = 3;indices[14] = 2;//底

indices[15] = 0;indices[16] = 4;indices[17] = 3;//底

IB -> Unlock();

D3DXVECTOR3 position(0.0f, 5.0f, 0.0f);//修改了值，使椎体更直观显示

D3DXVECTOR3 target(0.0f, 0.0f, 0.0f);

D3DXVECTOR3 up(0.0f, 0.0f, 1.0f);//相应修改

1. Cleanup()函数：

d3d::Release<IDirect3DIndexBuffer9\*>(IB);

1. Display()函数：

Device->BeginScene();

Device->SetStreamSource(0,VB,0,sizeof(Vertex));

Device->SetIndices(IB);

Device->SetFVF(Vertex::FVF);

Device->DrawIndexedPrimitive(D3DPT\_TRIANGLELIST,0,0,5,0,6);

D3DXMATRIX zoom,zoom1,zoom2,zoom3,zoom4;

D3DXMatrixTranslation(&zoom,zmX,0,0);

D3DXMatrixScaling(&zoom1,zm,zm,zm);

D3DXMatrixRotationX(&zoom2, Rx);//1°= PI /180°

D3DXMatrixRotationY(&zoom3, Ry);

D3DXMatrixRotationZ(&zoom4, Rz);

D3DXMATRIX p = zoom \* zoom1 \* zoom2 \* zoom3 \* zoom4;

Device->SetTransform(D3DTS\_WORLD,&p);

Device->EndScene();

1. WndProc()函数：

case WM\_KEYDOWN:

if( wParam == VK\_ESCAPE )

::DestroyWindow(hwnd);

if(wParam == 0x52)

zmX += 0.1f;

if(wParam == 0x4C)

zmX -= 0.1f;

if(wParam == 0x49)

zm += 0.1f;

if(wParam == 0x4F)

zm -= 0.1f;

if(wParam == 0x58)

Rx += 0.1f;

if(wParam == 0x59)

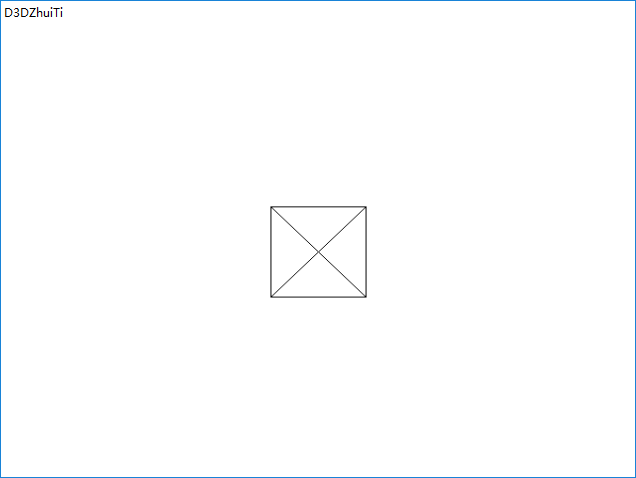
Ry += 0.1f;

if(wParam == 0x5A)

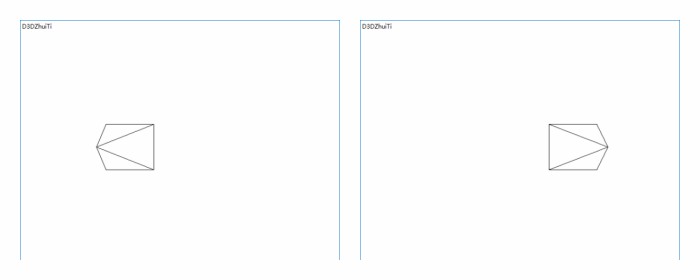
Rz += 0.1f;

break;

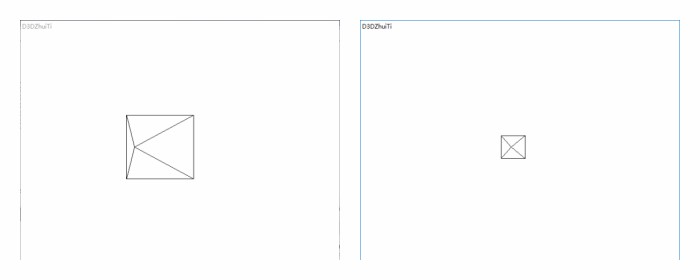
1. 程序运行结果
2. 原图：



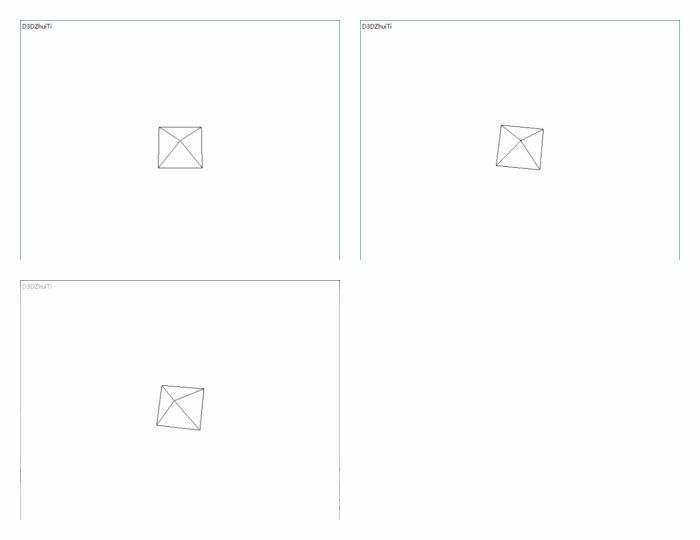
1. 左右移：



1. 放大缩小：



1. X,Y,Z旋转：



1. **总结**
2. 建模后旋转起来，总有一面渐渐消失不见

解决办法：看得见的面索引缓存为顺时针，看不见的面索引缓存为逆时针，左边三角形我认为是看不见的，所以为逆时针，但其结果不能达到预期效果，改为顺时针后，问题迎刃而解。

1. 按键不综合，不能同时左右移及放大缩小

解决办法：D3DXMATRIX p = zoom \* zoom1 \* zoom2 \* zoom3 \* zoom4;

1. **实验2**
2. 题目

调用D3DXCreateCylinder绘制一个圆柱体，并通过旋转变换让圆柱体以更直观的方式显示在屏幕上；通过修改D3DXCreateCylinder函数的参数值来修改圆柱体的外观，例如，增加/减小第二个参数的值，圆柱体的一端半径会变大/变小。

1. 实现代码

在绘制三角形的代码上修改：

1. 全局变量：

ID3DXMesh\* mesh = 0;

1. Setup()函数：

D3DXCreateCylinder(Device,1.0f,1.0f,3.0f,10,10,&mesh,0);

1. Cleanup()函数：

d3d::Release<ID3DXMesh\*>(mesh);

1. Display()函数：

D3DXMATRIX Rx,Ry;

D3DXMatrixRotationX(&Rx,3.14f/3.0f);

static float y = 0.0f;

D3DXMatrixRotationY(&Ry,y);

y += timeDelta;//0.002f;

if(y>=6.28f)

y = 0.0f;

D3DXMATRIX p = Rx \* Ry;

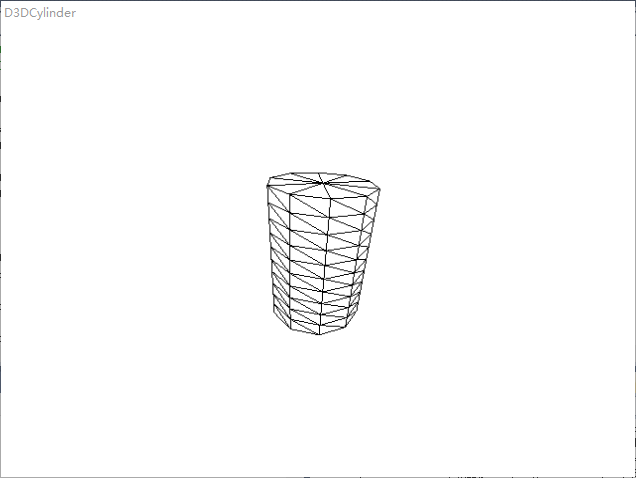
Device -> SetTransform(D3DTS\_WORLD,&p);

Device->BeginScene();

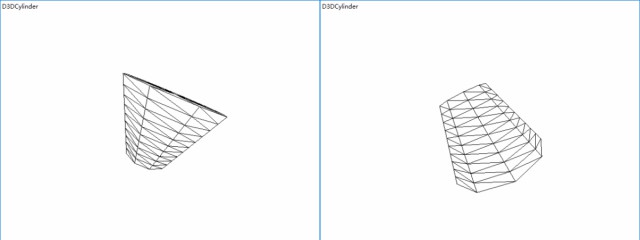
mesh->DrawSubset(0);

Device->EndScene();

1. 程序运行结果
2. 正常柱体：



1. 修改上半径为2和修改下半径为2后：



1. **总结**

调用库里的函数，很简单。