**武汉纺织大学**

**《Direct3D图形编程》上机实验报告**

**题目:** **网格（一）**

**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5月13 日**

1. **实验1**
2. 题目

使用网格绘制一个立方体，立方体的上下2个面为红色材质，左右2个面为绿色材质，前后2个面为蓝色材质；为场景添加白色方向灯光效果；立方体绕Y轴连续旋转；通过键盘控制使立方体绕不同的轴旋转：X键，绕X轴旋转 ；Y键，绕Y轴旋转；Z键，绕Z轴旋转；并且将按键操作以文字的形式显示在视图区。

1. 实现代码

在d3dInit框架的基础上添加代码：

1. 全局变量

IDirect3DDevice9\* Device = 0;

ID3DXMesh\* cube = 0;

const DWORD NumSubsets = 3;

D3DMATERIAL9 cubeMtrl[3] = {d3d::BLUE\_MTRL,d3d::RED\_MTRL,d3d::GREEN\_MTRL};

ID3DXFont\* Font = 0;

RECT rect = {0, 0, Width, Height};

struct Vertex

{

Vertex(){}

Vertex(float x, float y, float z, float nx, float ny, float nz)

{

\_x = x; \_y = y; \_z = z; \_nx = nx; \_ny = ny; \_nz = nz;

}

float \_x, \_y, \_z;

float \_nx, \_ny, \_nz;

static const DWORD FVF;

};

const DWORD Vertex::FVF = D3DFVF\_XYZ | D3DFVF\_NORMAL;

1. Setup()函数

D3DXFONT\_DESC df;

ZeroMemory(&df,sizeof(D3DXFONT\_DESC));

df.Height = 25;

df.Width = 12;

df.Weight = 500;

df.MipLevels = D3DX\_DEFAULT;

df.Italic = false;

df.CharSet = DEFAULT\_CHARSET;

df.OutputPrecision = 0;

df.Quality = 0;

df.PitchAndFamily = 0;

strcpy(df.FaceName,"Times New Roman");

D3DXCreateFontIndirect(Device,&df,&Font);

HRESULT hr = 0;

hr = D3DXCreateMeshFVF(

12,

24,

D3DXMESH\_MANAGED,

Vertex::FVF,

Device,

&cube);

Vertex\* v = 0;

cube->LockVertexBuffer(0, (void\*\*)&v);

v[0] = Vertex(-1.0f, -1.0f, -1.0f, 0.0f, 0.0f, -1.0f);//前¡ã面?蓝¤?,cubeMtrl[0]

v[1] = Vertex(-1.0f, 1.0f, -1.0f, 0.0f, 0.0f, -1.0f);

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

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

v[4] = Vertex(-1.0f, -1.0f, 1.0f, 0.0f, 0.0f, 1.0f);//后¨®面?

v[5] = Vertex(-1.0f, 1.0f, 1.0f, 0.0f, 0.0f, 1.0f);

v[6] = Vertex( 1.0f, 1.0f, 1.0f, 0.0f, 0.0f, 1.0f);

v[7] = Vertex( 1.0f, -1.0f, 1.0f, 0.0f, 0.0f, 1.0f);

v[8] = Vertex(-1.0f, 1.0f, -1.0f, 0.0f, 1.0f, 0.0f);//顶£¤面?红¨¬,cubeMtrl[1]

v[9] = Vertex(-1.0f, 1.0f, 1.0f, 0.0f, 1.0f, 0.0f);

v[10] = Vertex( 1.0f, 1.0f, 1.0f, 0.0f, 1.0f, 0.0f);

v[11] = Vertex( 1.0f, 1.0f, -1.0f, 0.0f, 1.0f, 0.0f);

v[12] = Vertex(-1.0f, -1.0f, -1.0f, 0.0f, -1.0f, 0.0f);//底Ì¡Á面?,cubeMtrl[1]

v[13] = Vertex(-1.0f, -1.0f, 1.0f, 0.0f, -1.0f, 0.0f);

v[14] = Vertex( 1.0f, -1.0f, 1.0f, 0.0f, -1.0f, 0.0f);

v[15] = Vertex( 1.0f, -1.0f, -1.0f, 0.0f, -1.0f, 0.0f);

v[16] = Vertex(-1.0f, -1.0f, -1.0f, -1.0f, 0.0f, 0.0f);//左Á¨®面?绿¨¬,cubeMtrl[2]

v[17] = Vertex(-1.0f, 1.0f, -1.0f, -1.0f, 0.0f, 0.0f);

v[18] = Vertex( -1.0f, 1.0f, 1.0f, -1.0f, 0.0f, 0.0f);

v[19] = Vertex( -1.0f, -1.0f, 1.0f, -1.0f, 0.0f, 0.0f);

v[20] = Vertex(1.0f, -1.0f, -1.0f, 1.0f, 0.0f, 0.0f);//右®¨°面?

v[21] = Vertex(1.0f, 1.0f, -1.0f, 1.0f, 0.0f, 0.0f);

v[22] = Vertex( 1.0f, 1.0f, 1.0f, 1.0f, 0.0f, 0.0f);

v[23] = Vertex( 1.0f, -1.0f, 1.0f, 1.0f, 0.0f, 0.0f);

cube->UnlockVertexBuffer();

WORD\* i = 0;

cube->LockIndexBuffer(0, (void\*\*)&i);

i[0] = 0; i[1] = 1; i[2] = 2;//前¡ã

i[3] = 0; i[4] = 2; i[5] = 3;

i[6] = 4; i[7] = 6; i[8] = 5;//后¨®

i[9] = 4; i[10] = 7; i[11] = 6;

i[12] = 8; i[13] = 9; i[14] = 10;//上¦?

i[15] = 8; i[16] = 10; i[17] = 11;

i[18] = 12; i[19] = 14; i[20] = 13;//下?

i[21] = 12; i[22] = 15; i[23] = 14;

i[24] = 16; i[25] = 18; i[26] = 17;//左Á¨®

i[27] = 16; i[28] = 19; i[29] = 18;

i[30] = 20; i[31] = 21; i[32] = 22;//右®¨°

i[33] = 20; i[34] = 22; i[35] = 23;

cube->UnlockIndexBuffer();

DWORD\* attributeBuffer = 0;

cube->LockAttributeBuffer(0, &attributeBuffer);

for(int a = 0; a < 4; a++)

attributeBuffer[a] = 0;

for(int b = 4; b < 8; b++)

attributeBuffer[b] = 1;

for(int c = 8; c < 12; c++)

attributeBuffer[c] = 2;

cube->UnlockAttributeBuffer();

D3DXVECTOR3 dir(1.0f,0.0f,0.0f);

D3DXCOLOR c = d3d::WHITE;

D3DLIGHT9 light = d3d::InitDirectionalLight(&dir, &c);

Device->SetLight(0,&light);

Device->LightEnable(0,true);

Device->SetRenderState(D3DRS\_NORMALIZENORMALS,true);

D3DXVECTOR3 position(0.0f, 0.0f, -6.0f);

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

D3DXVECTOR3 up(0.0f, 1.0f, 0.0f);

D3DXMATRIX V;

D3DXMatrixLookAtLH(&V, &position, &target, &up);

Device -> SetTransform(D3DTS\_VIEW, &V);

D3DXMATRIX proj;

D3DXMatrixPerspectiveFovLH(

&proj,

D3DX\_PI \* 0.5f,

(float)Width / (float)Height,

1.0f,

1000.0f);

Device -> SetTransform(D3DTS\_PROJECTION, &proj);

Device -> SetRenderState(D3DRS\_LIGHTING,true);

1. Cleanup()函数

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

d3d::Release<ID3DXFont\*>(Font);

1. Display()函数

Device->BeginScene();

Font->DrawText(NULL," 1、按'X'，绕X轴旋转\n2、按'Y'，绕Y轴旋转\n3、按'Z'，绕Z轴旋转 ",-1,&rect,DT\_TOP|DT\_LEFT,0xff000000);

D3DXMATRIX Rx,Ry,Rz;

static float x = 0.0f;

static float y = 0.0f;

static float z = 0.0f;

D3DXMatrixRotationX(&Rx,x);

D3DXMatrixRotationY(&Ry,y);

D3DXMatrixRotationZ(&Rz,z);

y += timeDelta;

if( ::GetAsyncKeyState('X') )

{

x += timeDelta;

}

else if( ::GetAsyncKeyState('Y') )

{

y += timeDelta;

}

else if( ::GetAsyncKeyState('Z') )

{

z += timeDelta;

}

D3DXMATRIX p = Rx \* Ry \* Rz;

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

for(int i = 0; i < NumSubsets; i++)

{

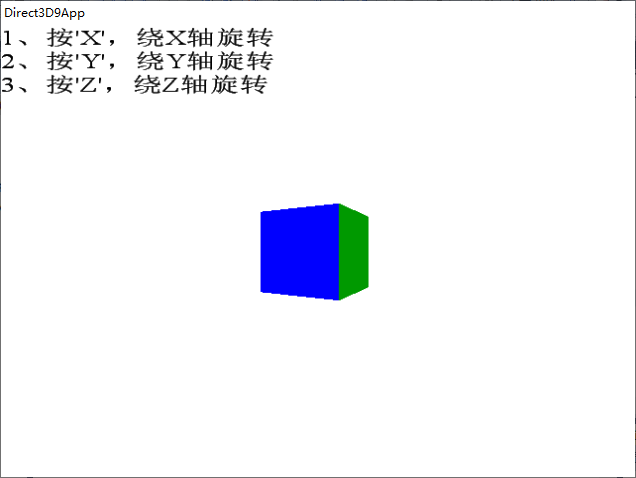
Device->SetMaterial(&cubeMtrl[i]);

cube->DrawSubset( i );

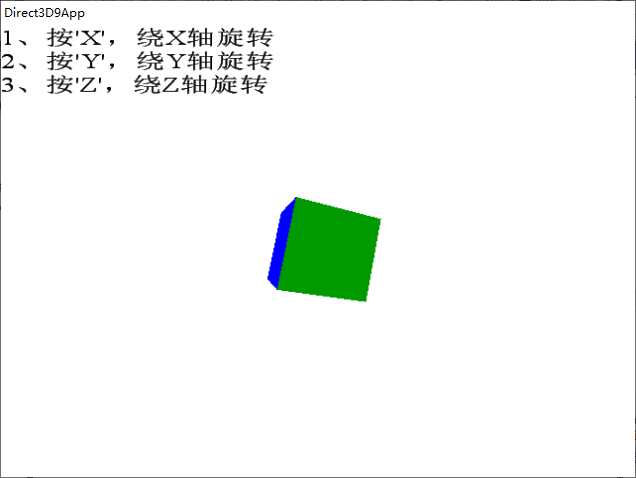
}

Device->EndScene();

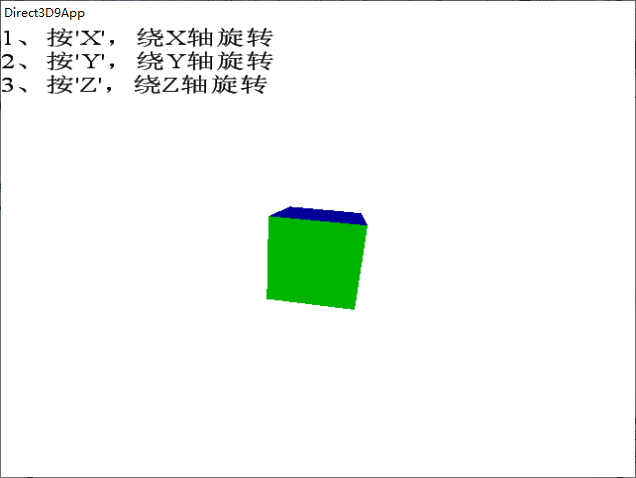
1. 程序运行结果
2. 原本



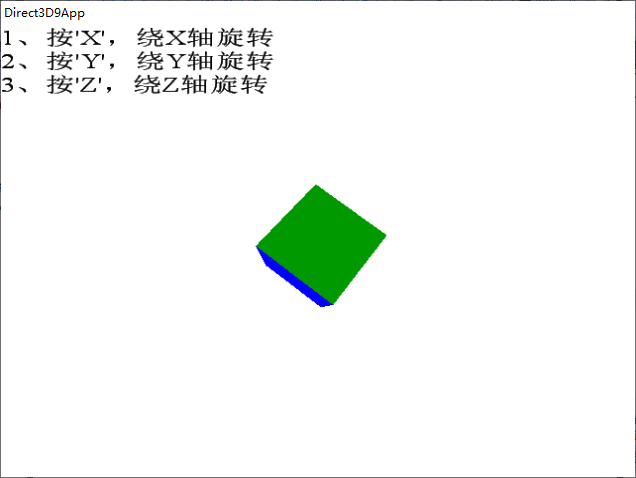
1. 按下“X”键



1. 按下“Y”键



1. 按下“Z”键



1. **总结**
2. 问题：三色交错。

解决方法：经过仔细检查核对顶点缓存和索引缓存确认无错后，检查顶点格式。一开始的思路是加一个color，所以相应的加了D3DFVF\_DIFFUSE。后面发现思路错了，但没有将D3DFVF\_DIFFUSE删除，故出现了正方体显示不出来并三色交错的现象。

1. 按键使球体旋转，在这里只实现了按一下旋转一下，没有实现连续旋转。