**武汉纺织大学**

**《Direct3D图形编程》上机实验报告**

**题目:** **D3D光照编程**

**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31日**

1. **实验1**
2. 题目

绘制一个正方体，不指定正方体每个顶点的颜色，通过为场景添加光照效果，使得正方体看起来像是红色的。修改材质的属性和光照的属性，记录属性变化后程序运行的结果，分析结果并给出结论。

1. 实现代码

在上次实验的基础上，修改代码：

1. 修改顶点格式：

struct ColorVertex

{

ColorVertex(){}

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

{

\_x = x; \_y = y; \_z = z;

}

float \_x, \_y, \_z;

float \_nx, \_ny, \_nz;

static const DWORD FVF;

};

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

1. 在d3dUtility.h中的命名空间d3d中加入：

D3DMATERIAL9 InitMtrl(D3DXCOLOR a, D3DXCOLOR d, D3DXCOLOR s, D3DXCOLOR e, float p);

const D3DMATERIAL9 WHITE\_MTRL = InitMtrl(WHITE,WHITE,WHITE,BLACK,8.0f);

const D3DMATERIAL9 RED\_MTRL = InitMtrl(RED,RED,RED,BLACK,8.0f);

const D3DMATERIAL9 GREEN\_MTRL = InitMtrl(GREEN,GREEN,GREEN,BLACK,8.0f);

const D3DMATERIAL9 BLUE\_MTRL = InitMtrl(BLUE,BLUE,BLUE,BLACK,8.0f);

const D3DMATERIAL9 YELLOW\_MTRL = InitMtrl(YELLOW,YELLOW,YELLOW,BLACK,8.0f);

D3DLIGHT9 InitDirectionalLight(D3DXVECTOR3\* direction,D3DXCOLOR\* color);

D3DLIGHT9 InitPointLight(D3DXVECTOR3\* position,D3DXCOLOR\* color);

D3DLIGHT9 InitSpotLight(D3DXVECTOR3\* position,D3DXVECTOR3\* direction,D3DXCOLOR\* color);

1. 在d3dUtility.cpp中实现函数：

D3DLIGHT9 d3d::InitDirectionalLight(D3DXVECTOR3\* direction, D3DXCOLOR\* color)

{

D3DLIGHT9 light;

::ZeroMemory(&light, sizeof(light));

light.Type = D3DLIGHT\_DIRECTIONAL;

light.Ambient = \*color \* 0.6f;

light.Diffuse = \*color;

light.Specular = \*color \* 0.6f;

light.Direction = \*direction;

return light;

}

D3DLIGHT9 d3d::InitPointLight(D3DXVECTOR3\* position, D3DXCOLOR\* color)

{

D3DLIGHT9 light;

::ZeroMemory(&light, sizeof(light));

light.Type = D3DLIGHT\_POINT;

light.Ambient = \*color \* 0.6f;

light.Diffuse = \*color;

light.Specular = \*color \* 0.6f;

light.Position = \*position;

light.Range = 1000.0f;

light.Falloff = 1.0f;

light.Attenuation0 = 1.0f;

light.Attenuation1 = 0.0f;

light.Attenuation2 = 0.0f;

return light;

}

D3DLIGHT9 d3d::InitSpotLight(D3DXVECTOR3\* position, D3DXVECTOR3\* direction, D3DXCOLOR\* color)

{

D3DLIGHT9 light;

::ZeroMemory(&light, sizeof(light));

light.Type = D3DLIGHT\_SPOT;

light.Ambient = \*color \* 0.0f;

light.Diffuse = \*color;

light.Specular = \*color \* 0.6f;

light.Position = \*position;

light.Direction = \*direction;

light.Range = 1000.0f;

light.Falloff = 1.0f;

light.Attenuation0 = 1.0f;

light.Attenuation1 = 0.0f;

light.Attenuation2 = 0.0f;

light.Theta = 0.4f;

light.Phi = 0.9f;

return light;

}

D3DMATERIAL9 d3d::InitMtrl(D3DXCOLOR a, D3DXCOLOR d, D3DXCOLOR s, D3DXCOLOR e, float p)

{

D3DMATERIAL9 mtrl;

mtrl.Ambient = a;

mtrl.Diffuse = d;

mtrl.Specular = s;

mtrl.Emissive = e;

mtrl.Power = p;

return mtrl;

}

1. Setup()函数：

Device -> CreateVertexBuffer(

36 \* sizeof(ColorVertex),

D3DUSAGE\_WRITEONLY,

ColorVertex::FVF,

D3DPOOL\_MANAGED,

&VB,

0);

ColorVertex\* v;

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

//V0(-1,0,-1) V1(-1,0,1) V2(1,0,1) V3(1,0,-1)

//V4(-1,1,-1) V5(-1,1,1) V6(1,1,1) V7(1,1,-1)

v[0] = ColorVertex(-1.0f,0.0f,-1.0f,0.0f,0.0f,-1.0f);//前¡ã面? V0 V4 V7

v[1] = ColorVertex(-1.0f,2.0f,-1.0f,0.0f,0.0f,-1.0f);

v[2] = ColorVertex(1.0f, 2.0f, -1.0f,0.0f,0.0f,-1.0f);

v[3] = ColorVertex(-1.0f,0.0f,-1.0f,0.0f,0.0f,-1.0f);//V0 V7 V3

v[4] = ColorVertex(1.0f, 2.0f, -1.0f,0.0f,0.0f,-1.0f);

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

v[6] = ColorVertex(-1.0f,2.0f,-1.0f,0.0f,1.0f,0.0f);//顶£¤面?V4 V5 V6

v[7] = ColorVertex(-1.0f,2.0f,1.0f,0.0f,1.0f,0.0f);

v[8] = ColorVertex(1.0f,2.0f,1.0f,0.0f,1.0f,0.0f);

v[9] = ColorVertex(-1.0f,2.0f,-1.0f,0.0f,1.0f,0.0f);//V4 V6 V7

v[10] = ColorVertex(1.0f,2.0f,1.0f,0.0f,1.0f,0.0f);

v[11] = ColorVertex(1.0f,2.0f, -1.0f,0.0f,1.0f,0.0f);

v[12] = ColorVertex(1.0f,0.0f,-1.0f,1.0f,0.0f,0.0f);//右®¨°面? V3 V7 V6

v[13] = ColorVertex(1.0f,2.0f, -1.0f,1.0f,0.0f,0.0f);

v[14] = ColorVertex(1.0f,2.0f,1.0f,1.0f,0.0f,0.0f);

v[15] = ColorVertex(1.0f,0.0f,-1.0f,1.0f,0.0f,0.0f);//V3 V6 V2

v[16] = ColorVertex(1.0f,2.0f,1.0f,1.0f,0.0f,0.0f);

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

v[18] = ColorVertex(-1.0f,0.0f,-1.0f,-1.0f,0.0f,0.0f);//左Á¨®面? V0 V5 V4

v[19] = ColorVertex(-1.0f,2.0f,1.0f,-1.0f,0.0f,0.0f);

v[20] = ColorVertex(-1.0f,2.0f,-1.0f,-1.0f,0.0f,0.0f);

v[21] = ColorVertex(-1.0f,0.0f,-1.0f,-1.0f,0.0f,0.0f);//V0 V1 V5

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

v[23] = ColorVertex(-1.0f,2.0f,1.0f,-1.0f,0.0f,0.0f);

v[24] = ColorVertex(-1.0f,0.0f,1.0f,0.0f,0.0f,1.0f);//背À3面? V1 V6 V5

v[25] = ColorVertex(1.0f,2.0f,1.0f,0.0f,0.0f,1.0f);

v[26] = ColorVertex(-1.0f,2.0f,1.0f,0.0f,0.0f,1.0f);

v[27] = ColorVertex(-1.0f,0.0f,1.0f,0.0f,0.0f,1.0f);//V1 V2 V6

v[28] = ColorVertex(1.0f,0.0f,1.0f,0.0f,0.0f,1.0f);

v[29] = ColorVertex(1.0f,2.0f,1.0f,0.0f,0.0f,1.0f);

v[30] = ColorVertex(-1.0f,0.0f,-1.0f,0.0f,-1.0f,0.0f);//底Ì¡Á面? V0 V2 V1

v[31] = ColorVertex(1.0f,0.0f,1.0f,0.0f,-1.0f,0.0f);

v[32] = ColorVertex(-1.0f,0.0f,1.0f,0.0f,-1.0f,0.0f);

v[33] = ColorVertex(-1.0f,0.0f,-1.0f,0.0f,-1.0f,0.0f);//V0 V3 V2

v[34] = ColorVertex(1.0f,0.0f,-1.0f,0.0f,-1.0f,0.0f);

v[35] = ColorVertex(1.0f,0.0f,1.0f,0.0f,-1.0f,0.0f);

VB -> Unlock();

D3DMATERIAL9 mtrl;

mtrl = d3d::WHITE\_MTRL;

Device->SetMaterial(&mtrl);

D3DLIGHT9 dir;

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

::ZeroMemory(&dir, sizeof(dir));

dir.Type = D3DLIGHT\_DIRECTIONAL;

dir.Diffuse = d3d::RED;

dir.Specular = d3d::RED \* 0.3f;

dir.Ambient = d3d::RED \* 0.6f;

dir.Direction = D3DXVECTOR3(0.0f,-1.0f,0.0f);

//D3DXCOLOR c = d3d::RED;

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

Device->SetLight(0,&dir);

Device->LightEnable(0,true);

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

注释掉之前胡线性绘图模式：

//Device -> SetRenderState(D3DRS\_LIGHTING,false);

1. Display()函数：

Device->Clear(0, 0, D3DCLEAR\_TARGET | D3DCLEAR\_ZBUFFER, 0x00000000, 1.0f, 0);

Device->BeginScene();

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

Device->SetFVF(ColorVertex::FVF);

D3DXMATRIX Ts,Rx,Ry;

D3DXMatrixRotationX(&Rx,3.14f/1.57f);

D3DXMatrixTranslation(&Ts,0.0f,-1.0f,.0f);

static float y = 0.0f;

D3DXMatrixRotationY(&Ry,y);

y+=timeDelta;

if(y>=6.28f)

y = 0.0f;

D3DXMATRIX p = Ts \* Rx \* Ry;

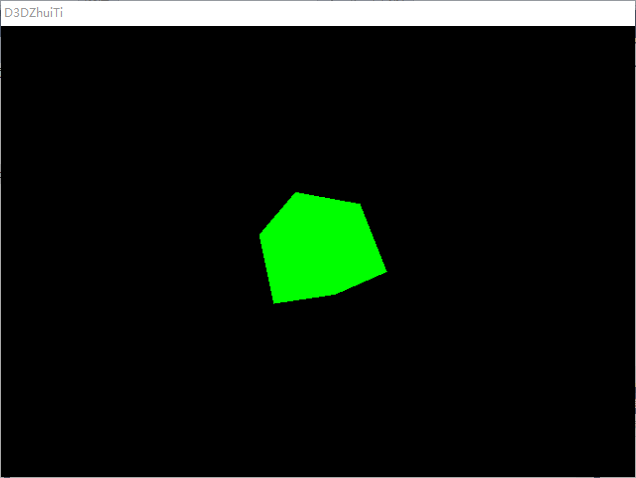
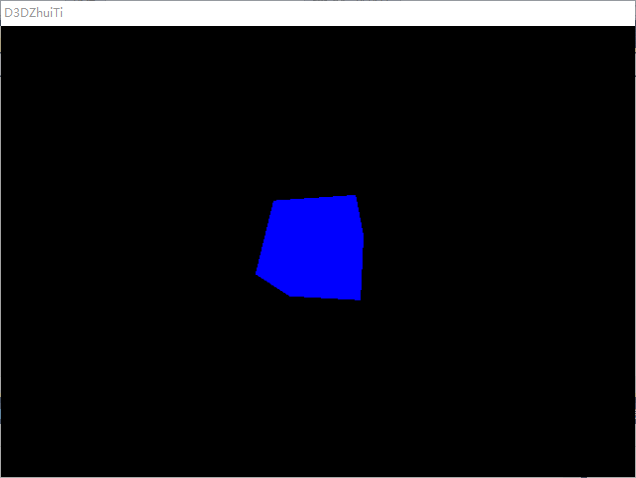
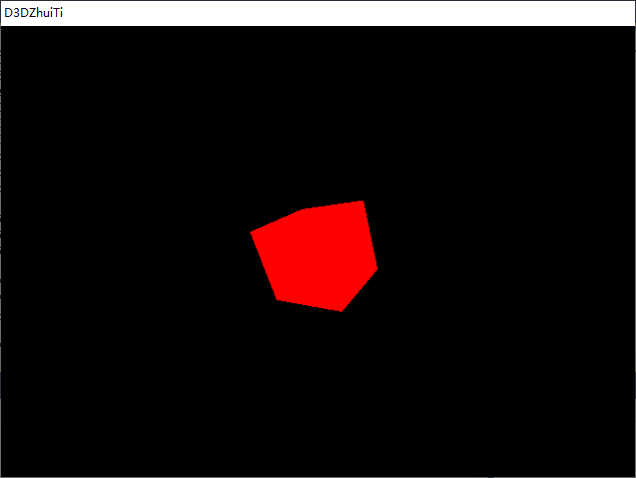
Device->SetTransform(D3DTS\_WORLD,&p);

Device->SetRenderState(D3DRS\_SHADEMODE,D3DSHADE\_FLAT);

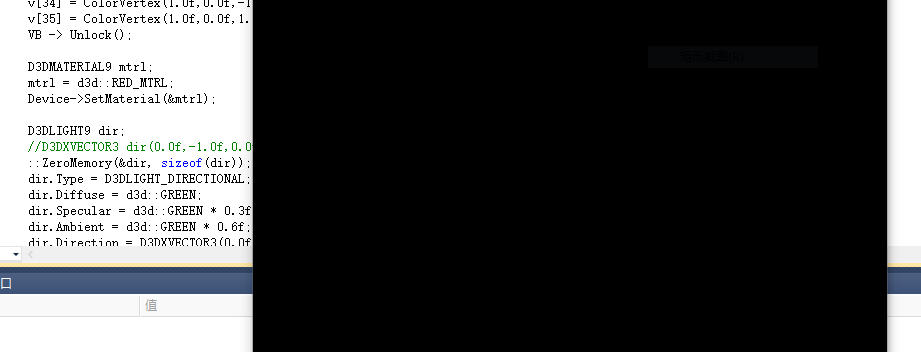
Device->DrawPrimitive(D3DPT\_TRIANGLELIST,0,12);

Device->EndScene();

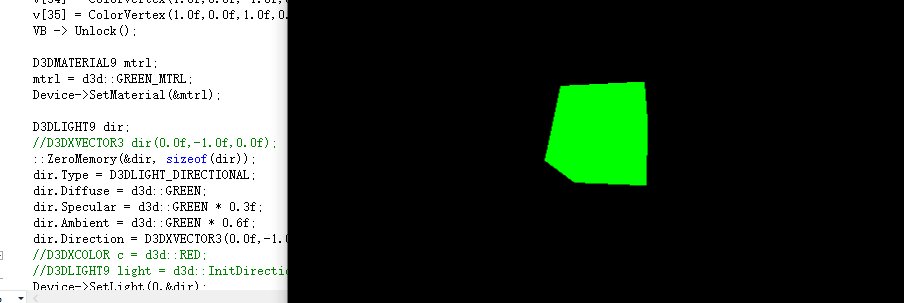
1. 程序运行结果
2. 材质WHITE，光照RED、BLUE、GREEN



1. 材质RED，光照GREEN



1. 材质GREEN，光照GREEN



1. **总结**

白色材质反射所有光，同色材质反射同色系的光，其他颜色胡材质不反射。

1. **实验２**
2. 题目

绘制一个球体，通过键盘控制为场景添加不同的光照效果，具体为：按下’a’键，为场景添加方向光；按下’b’键，为场景添加点光源；按下’c’键，为场景添加点聚光灯。

1. 实现代码

在实验１的基础上，修改代码：

1. 定义全局变量：

ID3DXMesh\* sphere;

D3DXCOLOR c = d3d::RED;

D3DLIGHT9 light;

1. Setup()函数：

D3DXCreateSphere(Device,1.0f,20,20,&sphere,0);

D3DMATERIAL9 mtrl;

mtrl = d3d::WHITE\_MTRL;

Device->SetMaterial(&mtrl);

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

1. Display()函数：

Device->Clear(0, 0, D3DCLEAR\_TARGET | D3DCLEAR\_ZBUFFER, 0xffffffff, 1.0f, 0);

Device->BeginScene();

sphere->DrawSubset(0);

Device->SetLight(0,&light);

Device->LightEnable(0,true);

D3DXMATRIX Ts,Rx,Ry;

D3DXMatrixRotationX(&Rx,3.14f/1.57f);

D3DXMatrixTranslation(&Ts,0.0f,-1.0f,.0f);

static float y = 0.0f;

D3DXMatrixRotationY(&Ry,y);

y+=timeDelta;

if(y>=6.28f)

y = 0.0f;

D3DXMATRIX p = Ts \* Rx \* Ry;

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

Device->EndScene();

1. WndProc()函数：

case WM\_KEYDOWN:

if( wParam == VK\_ESCAPE )

::DestroyWindow(hwnd);

if(wParam == 0x41)

{

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

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

}

if(wParam == 0x42)

{

D3DXVECTOR3 position(0.0f,0.0f,-1.0f);

light = d3d::InitPointLight(&position,&c);

}

if(wParam == 0x43)

{

D3DXVECTOR3 position(0.0f,0.0f,-1.0f);

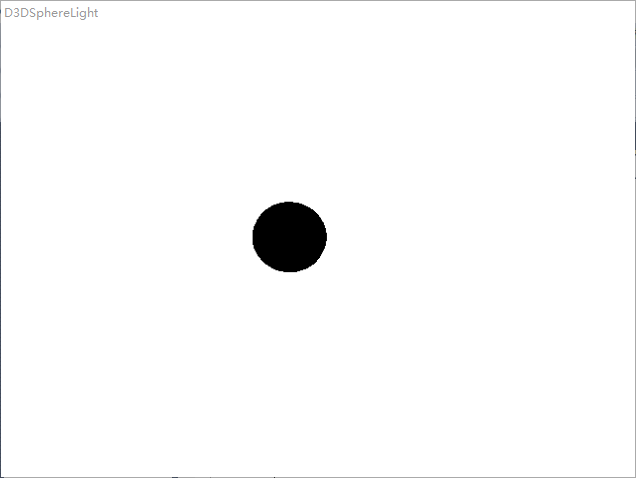
D3DXVECTOR3 dir(0.0f,0.0f,1.0f);

light = d3d::InitSpotLight(&position,&dir,&c);

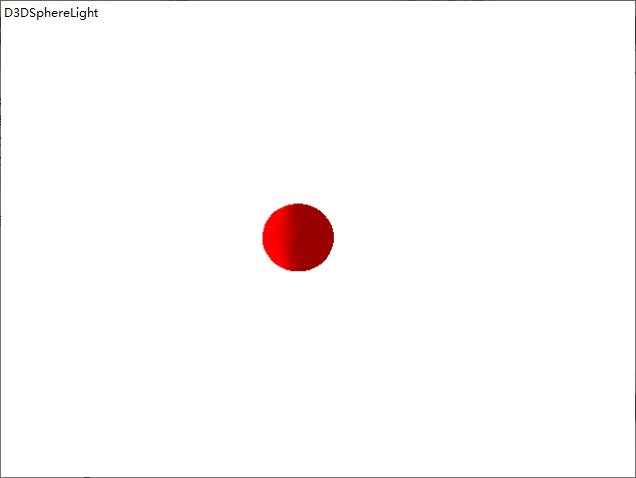
}

break;

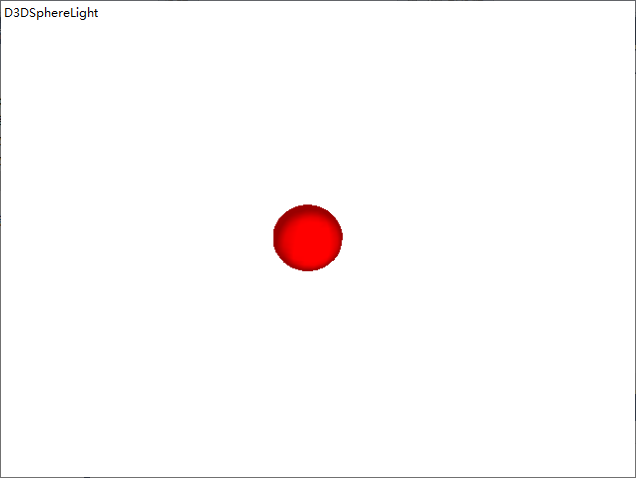
1. 程序运行结果
2. 原图：



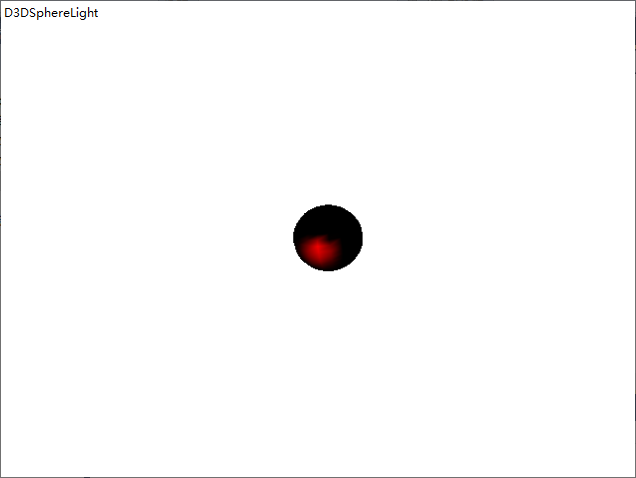
1. 按“a”键：



1. 按“b”键：



1. 按“c”键：



1. **总结**

光源的设置最直接影响视图区物体的可观性。