**武汉纺织大学**

**《Direct3D图形编程》上机实验报告**

**题目:** **融合技术**

**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4月14日**

1. **实验1**
2. 题目

编程实现如下功能（摄像机位置在Z轴-15位置处）:

先在（0，0，0）处绘制一个正方形并用一幅图像对正方形作纹理映射处理；再在（0，0，-5）处绘制一个茶壶，通过为场景添加光照效果使得茶壶看起来为红色；然后为2个模型添加融合效果，具体要求如下：

[1]原融合因子设置成全零，目标融合因子设置成全1；

[2]原融合因子设置成全1，目标融合因子设置成全0；

[3]使用Alpha的值来实现融合，其中Alpha的值取至茶壶的漫反射分量。并通过键盘的PageUp和PageDown按键增加和减少Alpha分量的大小。

1. 实现代码

在d3dInit程序框架上修改代码：

1. 全局变量：

IDirect3DDevice9\* Device = 0;

const int Width = 640;

const int Height = 480;

IDirect3DVertexBuffer9 \* VB=0;

IDirect3DTexture9 \* Tex=0;

D3DMATERIAL9 mtrl;

ID3DXMesh\* Teapot=0;

D3DMATERIAL9 TeapotMtrl;

1. 定义灵活顶点格式：

struct Vertex

{

Vertex(){}

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

{

\_x = x; \_y = y; \_z = z; \_nx = nx; \_ny = ny; \_nz = nz; \_u = u; \_v = v;

}

float \_x, \_y, \_z;

float \_nx, \_ny, \_nz;

float \_u,\_v;

static const DWORD FVF;

};

const DWORD Vertex::FVF = D3DFVF\_XYZ | D3DFVF\_NORMAL | D3DFVF\_TEX1;

1. Setup()函数：

Device->CreateVertexBuffer(

6 \* sizeof(Vertex),

D3DUSAGE\_WRITEONLY,

Vertex::FVF,

D3DPOOL\_MANAGED,

&VB,

0);

Vertex\* v;

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

v[0] = Vertex(-10.0f,-10.0f,0.0f,0.0f,0.0f,-1.0f,0.0f,1.0f);

v[1] = Vertex(-10.0f,10.0f,0.0f,0.0f,0.0f,-1.0f,0.0f,0.0f);

v[2] = Vertex(10.0f,10.0f,0.0f,0.0f,0.0f,-1.0f,1.0f,0.0f);

v[3] = Vertex(-10.0f,-10.0f,0.0f,0.0f,0.0f,-1.0f,0.0f,1.0f);

v[4] = Vertex(10.0f,10.0f,0.0f,0.0f,0.0f,-1.0f,1.0f,0.0f);

v[5] = Vertex(10.0f,-10.0f,0.0f,0.0f,0.0f,-1.0f,1.0f,1.0f);

VB->Unlock();

mtrl = d3d::WHITE\_MTRL;

TeapotMtrl = d3d::RED\_MTRL;

TeapotMtrl.Diffuse.a = 0.5f;

D3DXCreateTeapot(Device,&Teapot,0);

D3DLIGHT9 dir;

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

dir.Type = D3DLIGHT\_DIRECTIONAL;

dir.Diffuse = d3d::WHITE;

dir.Specular = d3d::WHITE \* 0.2f;

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

dir.Direction = D3DXVECTOR3(0.707f, 0.0f, 0.707f);

Device->SetLight(0, &dir);

Device->LightEnable(0, true);

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

D3DXCreateTextureFromFile(

Device,

"flower.jpg",

&Tex);

Device->SetSamplerState(0,D3DSAMP\_MAGFILTER,D3DTEXF\_LINEAR);

Device->SetSamplerState(0,D3DSAMP\_MINFILTER,D3DTEXF\_LINEAR);

Device->SetSamplerState(0,D3DSAMP\_MIPFILTER,D3DTEXF\_POINT);

Device->SetTextureStageState(0, D3DTSS\_ALPHAARG1, D3DTA\_DIFFUSE);//[3]

Device->SetTextureStageState(0, D3DTSS\_ALPHAOP, D3DTOP\_SELECTARG1);

Device->SetRenderState(D3DRS\_SRCBLEND, D3DBLEND\_SRCALPHA);

Device->SetRenderState(D3DRS\_DESTBLEND, D3DBLEND\_INVSRCALPHA);

/\*Device->SetRenderState(D3DRS\_SRCBLEND,D3DBLEND\_ZERO);//[1]

Device->SetRenderState(D3DRS\_DESTBLEND,D3DBLEND\_ONE);\*/

/\*Device->SetRenderState(D3DRS\_SRCBLEND,D3DBLEND\_ONE);//[2]

Device->SetRenderState(D3DRS\_DESTBLEND,D3DBLEND\_ZERO);\*/

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

D3DXVECTOR3 position(0.0f, 0.0f, -15.0f);

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);

return true;

1. Cleanup()函数：

d3d::Release<IDirect3DVertexBuffer9\*>(VB);

d3d::Release<IDirect3DTexture9\*>(Tex);

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

1. Display()函数：

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

Device->BeginScene();

D3DXMATRIX W;

D3DXMatrixTranslation(&W,0.0f,0.0f,0.0f);

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

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

Device->SetFVF(Vertex::FVF);

Device->SetMaterial(&mtrl);

Device->SetTexture(0, Tex);

Device->DrawPrimitive(D3DPT\_TRIANGLELIST,0,2);

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

D3DXMatrixTranslation(&W,0.0f,0.0f,-5.0f);

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

Device->SetMaterial(&TeapotMtrl);

Device->SetTexture(0, 0);

Teapot->DrawSubset(0);

Device->SetRenderState(D3DRS\_ALPHABLENDENABLE,false);

Device->EndScene();.

Device->Present(0, 0, 0, 0);

1. WndProc()函数：

case WM\_KEYDOWN:

if( wParam == VK\_ESCAPE )

::DestroyWindow(hwnd);

if( wParam == VK\_UP )

{

TeapotMtrl.Diffuse.a += 0.1f;

if(TeapotMtrl.Diffuse.a > 1.0f)

TeapotMtrl.Diffuse.a = 1.0f;

}

if( wParam == VK\_DOWN)

{

TeapotMtrl.Diffuse.a -= 0.1f;

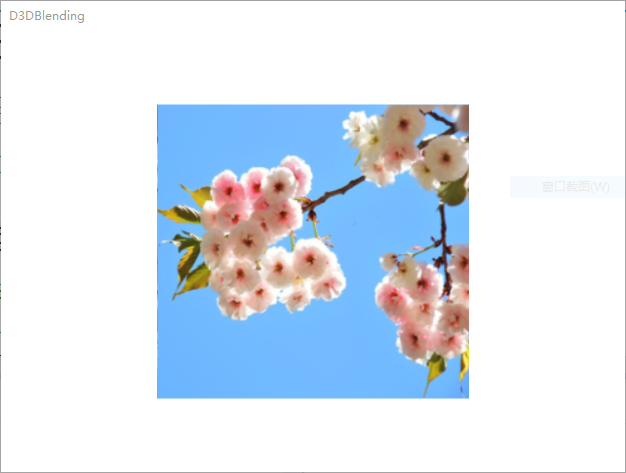
if(TeapotMtrl.Diffuse.a < 0.0f)

TeapotMtrl.Diffuse.a = 0.0f;

}

break;

1. 程序运行结果
   1. 原融合因子为全零，目标融合因子为全一：



* 1. 原融合因子为全一，目标融合因子为全零：



* 1. Alpha漫反射融合：



1. **总结**
2. 问题：第三小问的时候，前一天能运行出预想效果，但第二天的时候，就不能实现Alpha漫反射融合了。

解决方法：把那几条语句删了重新敲就可以了。

1. **实验２**
2. 题目

编程实现如下功能（摄像机位置在Z轴-15位置处）:

先用DirectX Texture Tool创建3幅透明度分别为0%，50%和100%的DDS1，DDS2 和DDS3图像；然后，在(0,0,0)处绘制一个茶壶，通过为场景添加光照效果使得茶壶看起来为蓝色；接着，在(0,0,-2)处绘制一个正方形并分别用DDS1，DDS2 和DDS3图像对正方形作纹理映射处理，通过使用Alpha的值来实现融合，使得我们可以透过图像看到后面的茶壶。

1. 实现代码

在实验一的基础上修改代码：

1. Setup()函数：

TeapotMtrl = d3d::BLUE\_MTRL;

D3DXCreateTextureFromFile(

Device,

"DDS2.dds",

&Tex);

Device->SetTextureStageState(0, D3DTSS\_ALPHAARG1, D3DTA\_TEXTURE);

Device->SetTextureStageState(0, D3DTSS\_ALPHAOP, D3DTOP\_SELECTARG1);

Device->SetRenderState(D3DRS\_SRCBLEND, D3DBLEND\_SRCALPHA);

Device->SetRenderState(D3DRS\_DESTBLEND, D3DBLEND\_INVSRCALPHA);

1. Display()函数：

Device->BeginScene();

D3DXMATRIX W;

D3DXMatrixTranslation(&W,0.0f,0.0f,0.0f);

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

Device->SetMaterial(&TeapotMtrl);

Device->SetTexture(0, 0);

Teapot->DrawSubset(0);

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

D3DXMatrixTranslation(&W,0.0f,0.0f,-2.0f);

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

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

Device->SetFVF(Vertex::FVF);

Device->SetMaterial(&mtrl);

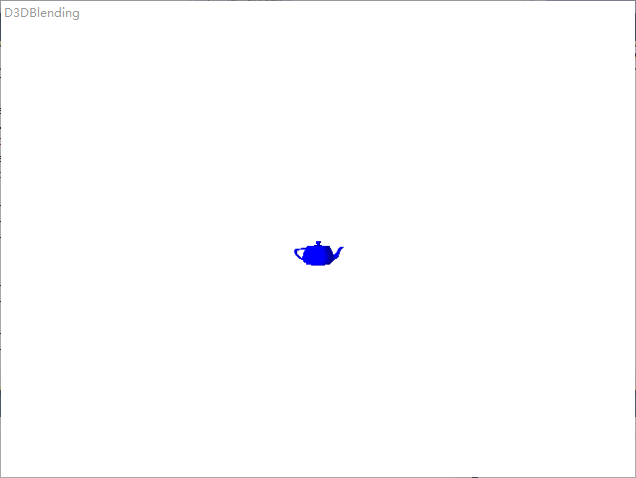
Device->SetTexture(0, Tex);

Device->DrawPrimitive(D3DPT\_TRIANGLELIST,0,2);

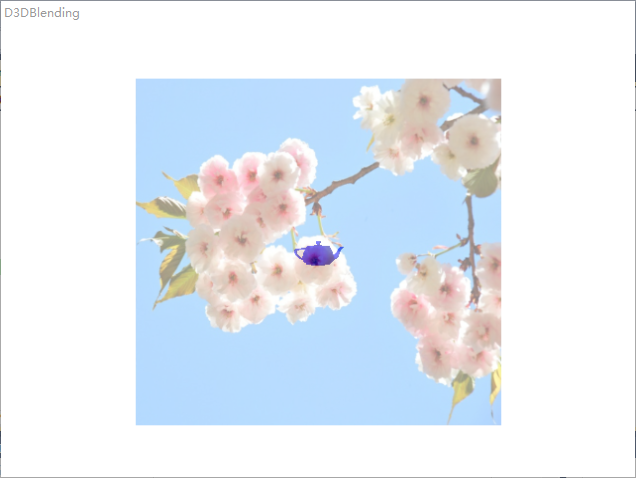
Device->SetRenderState(D3DRS\_ALPHABLENDENABLE,false);

Device->EndScene();

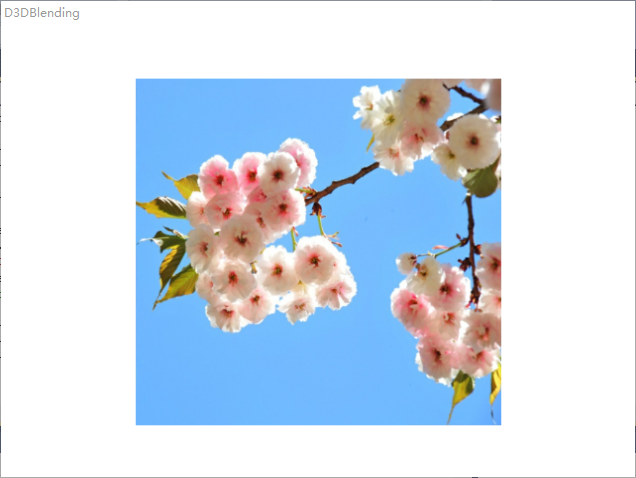
1. 程序运行结果
2. 图片透明度0%：



1. 图片透明度50%：



1. 图片透明度100%：



1. **总结**
2. 创建Alpha通道的方法：
3. 电脑“开始”选项，找到“Microsoft DirectX SDK(June 2010)”，选择该目录下的“DirectX Texture Tool”（每台电脑不一样，有的是在“Utilities”目录下），打开。
4. 然后打开一副图片，右下角查看该图片的像素大小，接着在画图工具里创建一幅相应大小像素的图片，并设置相应的透明度，保存。
5. View->Alpha Channel Only 可以查看图片的透明度。
6. Format->Change Surface Format 将图片格式扩展为A8R8G8B8。接着File->Open Onto Alpha Channel Of This Texxture。选择刚画图工具保存下来的图片。
7. 再重复3)，发现透明度改变了。保存为“.dds”文件。由此，创建Alpha 通道完成。