

Texturing, Blending

305890
2008년 봄학기
4/16/2007
박경신

Overview

- Texture
- Mipmap
- Blending
- Alpha Channel

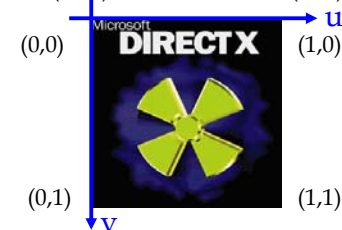
Texturing

- Texture Coordinates
- 텍스처맵핑 생성과 활성화
- 텍스처에 필터 (Filters) 적용
- 밍맵 (Mipmaps)
- Address Modes
- 예제: TexQuad, TexCube

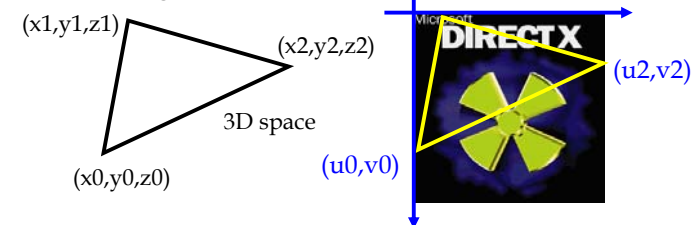
Texture Coordinates

- Texture Coordinates

- (u, v) : normalized to $(0, 1)$



- Mapping



Texture Coordinates

- Vertex 구조체에 텍스처 좌표 추가

```
struct Vertex {
    float _x, _y, _z;
    float _nx, _ny, _nz;
    float _u, _v;    // texture coordinates
    static const DWORD FVF;
};

Const DWORD Vertex::FVF =
    D3DFVF_XYZ | D3DFVF_NORMAL | D3DFVF_TEX1
```

Creating and Enabling Textures

- 파일 이미지를 IDirect3DTexture9 객체로 로드한다.
 - D3DXCreateTextureFromFile
 - Return: D3D_OK or D3DERR_xxx
 - BMP, DDS, DIB, HDR, JPG, PFM, PNG, PPM, TGA 이미지 포맷을 지원한다. (D3DXIMAGE_FILEFORMAT 참고)

```
HRESULT WINAPI D3DXCreateTextureFromFile(
    LPDIRECT3DDEVICE9 pDevice,    // point to IDirect3DDevice9
    LPCTSTR pSrcFile,           // filename
    LPDIRECT3DTEXTURE9 **ppTexture);    // out
```

Creating and Enabling Textures

- 예를 들어, "stonewall.bmp"를 texture로 만들고자 한다면 아래와 같이 작성한다.

```
IDirect3DTexture9* _stonewall;
D3DXCreateTextureFromFile(_device, "stonewall.bmp",
    &_stonewall);
```

- 현재 texture를 지정하는 데는 SetTexture를 사용한다.

```
HRESULT IDirect3DDevice9::SetTexture(
    DWORD Stage, // multi-texture를 위한 0~7범위 값. 흔히 0
    IDirect3DBaseTexture9 *pTexture);    // 지정할 texture
Device->SetTexture(0, _stonewall);
```

- NOTE: multi-texturing은 8개까지의 텍스처를 결합하여 좀더 자세한 이미지를 만들어내는 기법. 나중에 다룬다.

Creating and Enabling Textures

- 특정 텍스처 상태에서 텍스처를 이용하지 않고자 할 때 Texture를 0으로 지정

```
Device->SetTexture(0, 0);
renderObjectWithoutTexture();
```

- 서로 다른 texture를 이용하는 삼각형을 그리는 예:

```
Device->SetTexture(0, _tex0);
drawTrisUsingTex0();

Device->SetTexture(0, _tex1);
drawTrisUsingTex1();
```

Filter

- Texture 이미지에서의 삼각형 영역과 screen space에서의 삼각형 영역과의 맵핑 (mapping) 시의 크기 차이로 인한 왜곡 처리를 위한 필터 제공

- 확대를 위한 필터와 축소를 위한 필터가 필요함
- 확대를 위한 필터: D3DSAMP_MAGFILTER
- 축소를 위한 필터: D3DSAMP_MINFILTER

```
Device->SetSamplerState(0, D3DSAMP_MAGFILTER, D3DTEXF_POINT);  
Device->SetSamplerState(0, D3DSAMP_MINFILTER, D3DTEXF_POINT);
```

근접점 샘플링 필터

Filters

- 3가지 필터 제공

- 근접점 샘플링 (Nearest point sampling): D3DTEXF_POINT, default 방식, 가장 저품질, 가장 고속
- 선형 필터링 (Linear filtering): D3DTEXF_LINEAR, 높은 품질, 비교적 고속, 가장 권장함.

```
Device->SetSamplerState(0, D3DSAMP_MAGFILTER, D3DTEXF_LINEAR);  
Device->SetSamplerState(0, D3DSAMP_MINFILTER, D3DTEXF_LINEAR);
```

- 비등방성 필터링 (Anisotropic filtering): D3DTEXF_ANISOTROPIC 가장 고품질, 가장 느림.

- 품질을 결정하는 D3DSAMP_MAXANISOTROPY 레벨로 지정해야함. 높은 값일 수록 높은 품질임. Default 1

```
Device->SetSamplerState(0, D3DSAMP_MAXFILTER, D3DTEXF_ANISOTROPIC);
```

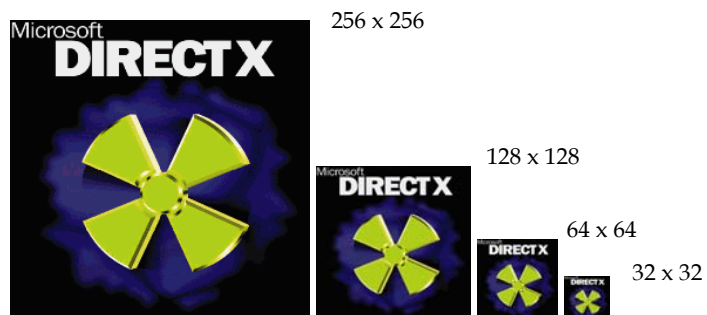
```
Device->SetSamplerState(0, D3DSAMP_MINFILTER, D3DTEXF_ANISOTROPIC);
```

```
Device->SetSamplerState(0, D3DSAMP_MAXANISOTROPY, 4);
```

Mipmap

- 텍스처의 급격한 크기 변화 방지를 위해 원래의 텍스처로부터 여러 개의 작은 해상도로 텍스처 (mipmap) 들을 생성함.

- 각 mipmap은 이전 mipmap 너비와 폭의 절반 크기임.



Mipmap

- Mipmap filter (D3DSAMP_MIPFILTER): Direct3D의 mipmap 이용 방법을 지정해줌.

```
Device->SetSamplerState(0, D3DSAMP_MIPFILTER, Filter);
```

- D3DTEXF_NONE: mipmap을 사용하지 않음
- D3DTEXF_POINT: Direct3D는 스크린 삼각형과 가장 비슷한 크기의 mipmap 레벨을 선택한다. 일단 레벨이 선택되면 지정된 min과 mag 태그에 따라 선택된 레벨을 필터링한다.
- D3DTEXF_LINEAR: 가장 비슷한 두 개의 mipmap 레벨에 min과 mag 필터에 따른 필터링을 적용한다. 즉, 선형적으로 조합된 두 개의 레벨을 이용해 최종 픽셀 컬러를 계산해낸다.

- Direct3D에서 mipmap 사용은 대부분 자동으로 처리됨

- D3DXCreateTextureFromFile을 이용하면 자동으로 mipmap 체인을 생성해 준다. 또한 스크린 삼각형에 가장 잘 맞는 mipmap을 자동으로 선택해 준다.

Address Modes

- 텍스처의 좌표가 $[0, 1]$ 을 벗어날 경우의 처리 방법: address mode로 정의함
 - Address mode: wrap, border, color, clamp, mirror


```
IDirect3DDevice9::SetSamplerState(samplerStageIndex,
                                    D3DSAMP_ADDRESSU/V,
                                    D3DTEXTUREADDRESS addressMode);
```
 - D3DTEXTUREADDRESS enum type의 요소
 - D3DTADDRESS_WRAP: repeat the texture on every integer junction
 - D3DTADDRESS_MIRROR: every other row and column is a mirrors image of the preceding row or column
 - D3DTADDRESS_CLAMP: smear the color of edge pixels
 - D3DADDRESS_BORDER: use the border color, for any texture coordinates outside the range

Address Modes

- Address Mode 예제:

```
// wrap
Device->SetSamplerState(0, D3DSAMP_ADDRESSU, D3DADDRESS_WRAP);
Device->SetSamplerState(0, D3DSAMP_ADDRESSV, D3DADDRESS_WRAP);

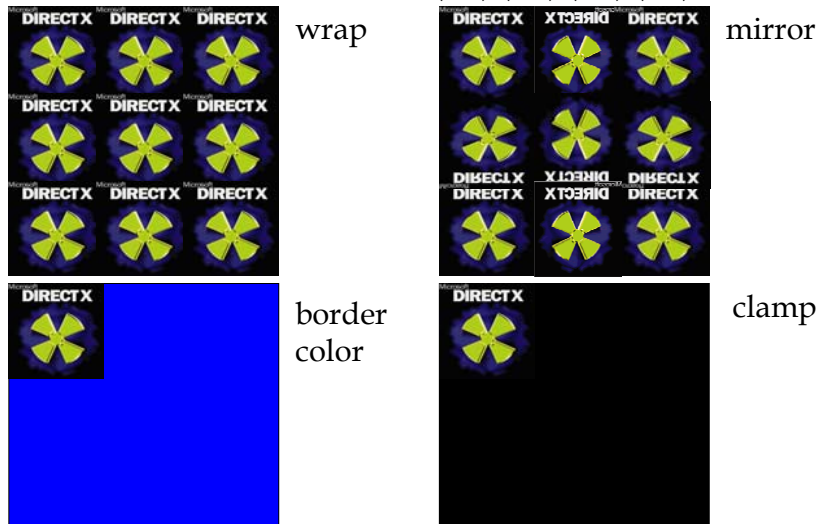
// border color
Device->SetSamplerState(0, D3DSAMP_ADDRESSU, D3DADDRESS_BORDER);
Device->SetSamplerState(0, D3DSAMP_ADDRESSV, D3DADDRESS_BORDER);
Device->SetSamplerState(0, D3DSAMP_BORDERCOLOR, 0x000000ff);

// clamp
Device->SetSamplerState(0, D3DSAMP_ADDRESSU, D3DADDRESS_CLAMP);
Device->SetSamplerState(0, D3DSAMP_ADDRESSV, D3DADDRESS_CLAMP);

// mirror
Device->SetSamplerState(0, D3DSAMP_ADDRESSU, D3DADDRESS_MIRROR);
Device->SetSamplerState(0, D3DSAMP_ADDRESSV, D3DADDRESS_MIRROR);
```

Address Modes

- 사각형에 4개의 텍스처좌표를 $(0,0),(0,3),(3,0),(3,3)$ 로 정의한 예:



Example: TexQuad

- 예제 TexQuad에서는 사각형에 텍스처를 입히고 텍스처 필터를 지정하는 방법을 알아본다.
 1. 텍스처 좌표를 지정하고 물체의 버텍스를 구성한다.
 2. D3DXCreateTextureFromFile 함수를 이용해 IDirect3DTexture9 인터페이스에 텍스처를 읽어들인다.
 3. 축소, 확대, mip맵 필터를 지정한다.
 4. 물체를 그리기 전에 IDirect3DDevice9::SetTexture 함수를 이용해 물체와 연결할 텍스처를 지정한다.



Example: TexQuad

```
#include "d3dUtility.h"
IDirect3DDevice9* Device = 0;
const int Width = 640;
Const int Height = 480;
IDirect3DVertexBuffer9* Quad = 0;
IDirect3DTexture9* Tex= 0;
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;
```

Example: TexQuad

```
void Cleanup() {
    d3d::Release<IDirect3DVertexBuffer9*>(Quad);
    d3d::Release<IDirect3DTexture9*>(Tex);
}

bool Setup() {
    Device->CreateVertexBuffer(6*sizeof(Vertex), D3DUSAGE_WRITEONLY,
    Vertex::FVF, D3DPOOL_MANAGED, &Quad, 0);
    Vertex* v;
    Quad->Lock(0, 0, (void**)&v, 0);
    v[0] = Vertex(-1, -1, 1.25, 0, 0, -1, 0, 1);
    v[1] = Vertex(-1, 1, 1.25, 0, 0, -1, 0, 0);
    v[2] = Vertex(1, 1, 1.25, 0, 0, -1, 1, 0);
    v[3] = Vertex(-1, -1, 1.25, 0, 0, -1, 0, 1);
    v[4] = Vertex(1, 1, 1.25, 0, 0, -1, 1, 0);
    v[5] = Vertex(1, -1, 1.25, 0, 0, -1, 1, 0);
    Quad->Unlock();
}
```

Example: TexQuad

```
D3DXCreateTextureFromFile(Device, "dx5_logo.bmp", &Tex);
Device->SetTexture(0, Tex);
Device->SetSamplerState(0, D3DSAMP_MAGFILTER, D3DTEXF_LINEAR);
Device->SetSamplerState(0, D3DSAMP_MINFILTER, D3DTEXF_LINEAR);
Device->SetSamplerState(0, D3DSAMP_MIPFILTER, D3DTEXF_POINT);
// don't use lighting for this example
Device->SetRenderState(D3DRS_LIGHTING, false);

// set the projection matrix
D3DXMATRIX proj;
D3DXMatrixPerspectiveFovLH(&proj, D3DX_PI * 0.5f,
    (float)Width/(float)Height, 1.0f, 1000.0f);
Device->SetTransform(D3DTS_PROJECTION, &proj);
return true;
}
```

Example: TexQuad

```
void Display(float timeDelta) {
    if (Device) {
        Device->Clear(0, 0, D3DCLEAR_TARGET | D3DCLEAR_ZBUFFER,
            0xffffffff, 1.0f, 0);

        Device->BeginScene();
        Device->SetStreamSource(0, Quad, 0, sizeof(Vertex));
        Device->SetFVF(Vertex::FVF);
        Device->DrawPrimitive(D3DPT_TRIANGLELIST, 0, 2);
        Device->EndScene();
        Device->Present(0, 0, 0, 0);
    }
    return true;
}
```

Example: TexCube

```
#ifndef __cubeH__
#define __cubeH__
#include <d3dx9.h>
#include <string>
class Cube
{
public:
    Cube(IDirect3DDevice9* device);
    ~Cube();

    bool draw(D3DXMATRIX* world, D3DMATERIAL9* mtrl, IDirect3DTexture9*
    tex);
private:
    IDirect3DDevice9* _device;
    IDirect3DVertexBuffer9* _vb;
    IDirect3DIndexBuffer9* _ib;
};
#endif // __cubeH__
```

cube.h

Example: TexCube

```
#include "cube.h"
#include "vertex.h"

Cube::Cube(IDirect3DDevice9* device)
{
    // save a ptr to the device
    _device = device;

    _device->CreateVertexBuffer(24 * sizeof(Vertex),
    D3DUSAGE_WRITEONLY, FVF_VERTEX,
    D3DPOOL_MANAGED, &_vb, 0);

    Vertex* v;
    _vb->Lock(0, 0, (void*)&v, 0);

    // build the box
    .....
    _vb->Unlock();
}
```

cube.cpp

Example: TexCube

```
// build the box
v[0] = Vertex(-1.0f, -1.0f, -1.0f, 0.0f, 0.0f, -1.0f, 0.0f, 0.0f); // front face
v[1] = Vertex(-1.0f, 1.0f, -1.0f, 0.0f, 0.0f, -1.0f, 0.0f, 1.0f);
v[2] = Vertex(1.0f, 1.0f, -1.0f, 0.0f, 0.0f, -1.0f, 1.0f, 1.0f);
v[3] = Vertex(1.0f, -1.0f, -1.0f, 0.0f, 0.0f, -1.0f, 1.0f, 0.0f);
v[4] = Vertex(-1.0f, -1.0f, 1.0f, 0.0f, 0.0f, 1.0f, 0.0f, 0.0f); // back
v[5] = Vertex(1.0f, -1.0f, 1.0f, 0.0f, 0.0f, 1.0f, 0.0f, 1.0f);
v[6] = Vertex(1.0f, 1.0f, 1.0f, 0.0f, 0.0f, 1.0f, 1.0f, 1.0f);
v[7] = Vertex(-1.0f, 1.0f, 1.0f, 0.0f, 0.0f, 1.0f, 1.0f, 0.0f);
v[8] = Vertex(-1.0f, 1.0f, -1.0f, 0.0f, 1.0f, 0.0f, 0.0f, 0.0f); // top
v[9] = Vertex(-1.0f, 1.0f, 1.0f, 0.0f, 1.0f, 0.0f, 0.0f, 1.0f);
v[10] = Vertex(1.0f, 1.0f, 1.0f, 0.0f, 1.0f, 0.0f, 1.0f, 1.0f);
v[11] = Vertex(1.0f, 1.0f, -1.0f, 0.0f, 1.0f, 0.0f, 1.0f, 0.0f);
v[12] = Vertex(-1.0f, -1.0f, -1.0f, 0.0f, -1.0f, 0.0f, 0.0f, 0.0f); // bottom
v[13] = Vertex(1.0f, -1.0f, -1.0f, 0.0f, -1.0f, 0.0f, 0.0f, 1.0f);
v[14] = Vertex(1.0f, -1.0f, 1.0f, 0.0f, -1.0f, 0.0f, 1.0f, 1.0f);
v[15] = Vertex(-1.0f, -1.0f, 1.0f, 0.0f, -1.0f, 0.0f, 1.0f, 0.0f);
v[16] = Vertex(-1.0f, -1.0f, 1.0f, 0.0f, 0.0f, 0.0f, 0.0f, 0.0f); // left
v[17] = Vertex(-1.0f, 1.0f, 1.0f, 0.0f, 0.0f, 0.0f, 0.0f, 1.0f);
v[18] = Vertex(-1.0f, 1.0f, -1.0f, 0.0f, 0.0f, 0.0f, 1.0f, 1.0f);
v[19] = Vertex(-1.0f, -1.0f, -1.0f, 0.0f, 0.0f, 0.0f, 1.0f, 0.0f);
v[20] = Vertex(1.0f, -1.0f, -1.0f, 1.0f, 0.0f, 0.0f, 0.0f, 0.0f); // right
v[21] = Vertex(1.0f, 1.0f, -1.0f, 1.0f, 0.0f, 0.0f, 0.0f, 1.0f);
v[22] = Vertex(1.0f, 1.0f, 1.0f, 1.0f, 0.0f, 0.0f, 1.0f, 1.0f);
v[23] = Vertex(1.0f, -1.0f, 1.0f, 1.0f, 0.0f, 0.0f, 1.0f, 0.0f);
```

cube.cpp

Example: TexCube

```
_device->CreateIndexBuffer(36 * sizeof(WORD),
D3DUSAGE_WRITEONLY, D3DFMT_INDEX16,
D3DPOOL_MANAGED, &_ib, 0);

WORD* i = 0;
_ib->Lock(0, 0, (void*)&i, 0);
i[0] = 0; i[1] = 1; i[2] = 2; // front face index
i[3] = 0; i[4] = 2; i[5] = 3;
i[6] = 4; i[7] = 5; i[8] = 6; // back face index
i[9] = 4; i[10] = 6; i[11] = 7;
i[12] = 8; i[13] = 9; i[14] = 10; // top face index
i[15] = 8; i[16] = 10; i[17] = 11;
i[18] = 12; i[19] = 13; i[20] = 14; // bottom face index
i[21] = 12; i[22] = 14; i[23] = 15;
i[24] = 16; i[25] = 17; i[26] = 18; // left face index
i[27] = 16; i[28] = 18; i[29] = 19;
i[30] = 20; i[31] = 21; i[32] = 22; // right face index
i[33] = 20; i[34] = 22; i[35] = 23;
_ib->Unlock();
}
```

cube.cpp

Example: TexCube

```
Cube::~Cube()
{
    if(_vb){_vb->Release(); _vb = 0;}
    if(_ib){_ib->Release(); _ib = 0;}
}

bool Cube::draw(D3DXMATRIX* world, D3DMATERIAL9* mtrl,
               IDirect3DTexture9* tex)
{
    if( world ) _device->SetTransform(D3DTS_WORLD, world);
    if( mtrl ) _device->SetMaterial(mtrl);
    if( tex ) _device->SetTexture(0, tex);

    _device->SetStreamSource(0, _vb, 0, sizeof(Vertex));
    _device->SetIndices(_ib);
    _device->SetFVF(FVF_VERTEX);
    _device->DrawIndexedPrimitive(D3DPT_TRIANGLELIST, 0, 0, 24, 0, 12);

    return true;
}
```

cube.cpp

Example: TexCube

```
#include "d3dUtility.h"
#include "cube.h"
#include "vertex.h"
...
Cube*      Box = 0;
IDirect3DTexture9* Tex = 0;

bool Setup()
{
    Box = new Cube(Device);      // create the texture cube
    ...
    // Create texture.
    D3DXCreateTextureFromFile(Device, "crate.jpg", &Tex);
    // Set Texture Filter States.
    Device->SetSamplerState(0, D3DSAMP_MAGFILTER, D3DTEXF_LINEAR);
    Device->SetSamplerState(0, D3DSAMP_MINFILTER, D3DTEXF_LINEAR);
    Device->SetSamplerState(0, D3DSAMP_MIPFILTER, D3DTEXF_LINEAR);
    ...
    return true;
}
```

texCube.cpp

Example: TexCube

```
void Cleanup() {
    d3d::Delete<Cube*>(Box);
    d3d::Release<IDirect3DTexture9*>(Tex);
}

bool Display(float timeDelta) {
    if( Device ) {
        ...
        // Draw the scene:
        Device->Clear(0, 0, D3DCLEAR_TARGET | D3DCLEAR_ZBUFFER,
                    0xffffffff, 1.0f, 0);
        Device->BeginScene();
        Device->SetMaterial(&d3d::WHITE_MTRL);
        Device->SetTexture(0, Tex);
        Box->draw(0, 0, 0);
        Device->EndScene();
        Device->Present(0, 0, 0, 0);
    }
    return true;
}
```

texCube.cpp

Blending

- ❑ Blending Equation
 - 블렌딩의 동작원리와 이용 방법
- ❑ Blend Factors
 - Direct3D가 지원하는 다양한 블렌드 모드
- ❑ Transparency
 - 기본 도형의 투명도를 제어하기 위해 알파성분을 이용하는 방법
- ❑ Creating an Alpha Channel Using DX Tex Tool
- ❑ 예제: MtrlAlpha, texAlpha

Blending Equation

- Blending
 - 현재 계산되고 있는 픽셀 값 (즉, 원본 픽셀)에 앞서 쓰여진 픽셀값 (즉, 목적 픽셀)과 결합하는 방식
- Blending Rule
 1. Blending을 이용하지 않는 물체를 먼저 그린다
 2. Blending을 이용하는 물체들을 카메라와의 거리에 따라 정렬한다. (viewing space에서 물체들을 z 성분을 통해 정렬하면 된다.)
 3. Blending을 이용하는 물체들을 뒤에서 앞으로의 순서로 그려나간다.

Blending Equation

- 두 pixel값을 Blending하는 Equation

$$\text{outputPixel} = \text{srcPixel} * \text{srcBlendFactor} + \text{dstPixel} * \text{dstBlendFactor}$$
 - 여기서 각 변수는 (r, g, b, a)벡터이고, *는 성분 단위의 곱임
 - outputPixel은 Blending한 결과 pixel
 - srcPixel은 현재 pixel
 - dstPixel은 back buffer에서의 pixel
 - srcBlendFactor와 dstBlendFactor는 [0,1] 범위 내의 값

Blending Equation

- Blending활성화

Device->setRenderState(D3DRS_ALPHABLENDENABLE, true);

 - Default는 비활성화
 - 계산량을 위해서 꼭 필요한 경우에서만 활성화하여 사용하고 끝나면 다시 비활성화를 시킬 것.
 - 필요한 삼각형들을 한꺼번에 처리하여 하나의 프레임 내에서 여러 번 켜고 끄는 일을 피할 것.

Blend Factors

- Blend factor의 여러 조합으로 원하는 효과를 만들어 냄

Device->setRenderState(D3DRS_SRCBLEND, Source);

Device->setRenderState(D3DRS_DESTBLEND, Destination);

D3DBLEND_ZERO	(0, 0, 0, 0)
D3DBLEND_ONE	(1, 1, 1, 1)
D3DBLEND_SRCOLOR	(r _s , g _s , b _s , a _s)
D3DBLEND_INVSRCOLOR	(1-r _s , 1-g _s , 1-b _s , 1-a _s)
D3DBLEND_SRCALPHA	(a _s , a _s , a _s , a _s) srcFactor의 default
D3DBLEND_INVSRCALPHA	(1-a _s , 1-a _s , 1-a _s , 1-a _s) dstFactor의 default
D3DBLEND_DESTALPHA	(a _d , a _d , a _d , a _d)
D3DBLEND_INVDESTALPHA	(1-a _d , 1-a _d , 1-a _d , 1-a _d)
D3DBLEND_DESTCOLOR	(r _d , g _d , b _d , a _d)
D3DBLEND_INVDESTCOLOR	(1-r _d , 1-g _d , 1-b _d , 1-a _d)
D3DBLEND_SRCALPHASAT	(f, f, f, 1), f = min(a _s , 1-a _d)
D3DBLEND_BOTHINVSRCALPHA	srcFactor = (1-a _s , 1-a _s , 1-a _s , 1-a _s) dstFactor = (a _s , a _s , a _s , a _s)

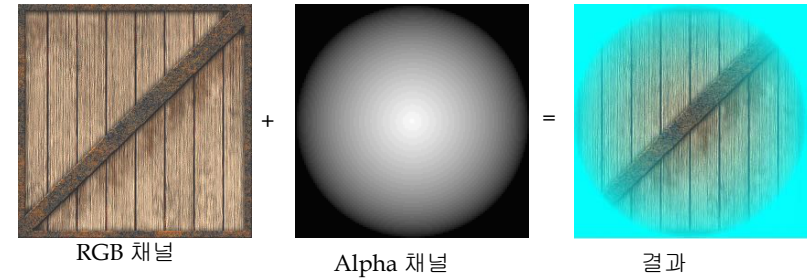
NOTE:D3DRS_SRCBLEND에서만 이용가능

Transparency

- 투명 (Transparency)
 - Pixel의 알파성분 (alpha component)으로 투명도 단계를 지정함.
 - 예를 들어, 각 픽셀마다 8비트씩 알파 성분 할당 [0, 255]:
 - 불투명도: alpha 0 - 0%는 완전투명
 - alpha 128 - 50%는 반투명
 - alpha 255 - 100%는 완전불투명에 해당
 - alpha성분으로 투명도를 지정하려면 srcBlendFactor, dstBlendFactor를 D3DBLEND_SRCALPHA, D3DBLEND_INVSRCALPHA로 지정해야 함 (default로 되어 있음)

Alpha Channel

- 알파채널 (Alpha channel)
 - Alpha 성분: (1) shading시 계산하여 얻거나, (2) texture에서의 alpha channel로부터 얻음.
 - Alpha channel: texture mapping시 alpha channel도 mapping되며 이것이 face에서의 pixel의 alpha성분이 됨.



Set Transparency

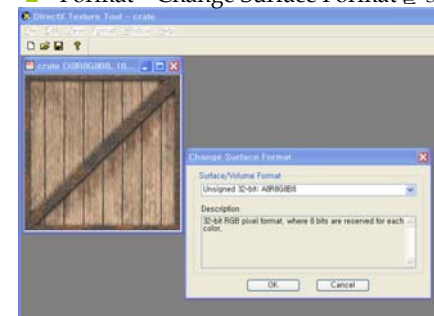
- 현재 texture가 alpha channel을 가지고 있으면 alpha channel에서 가져오고, 없으면 vertex color에서 가져옴.
- 어떤 소스를 이용할 것인지를 지정하는 예:

```
// shading과정에서 diffuse color로부터 alpha를 얻도록 지정함
Device->SetTextureStageState(0, D3DTSS_ALPHAARG1, D3DTA_DIFFUSE);
Device->SetTextureStageState(0, D3DTSS_ALPHAOP, D3DTOP_SELECTARG1);

// alpha channel에서 alpha를 얻도록 지정함
Device->SetTextureStageState(0, D3DTSS_ALPHAARG1, D3DTA_TEXTURE);
Device->SetTextureStageState(0, D3DTSS_ALPHAOP, D3DTOP_SELECTARG1);
```

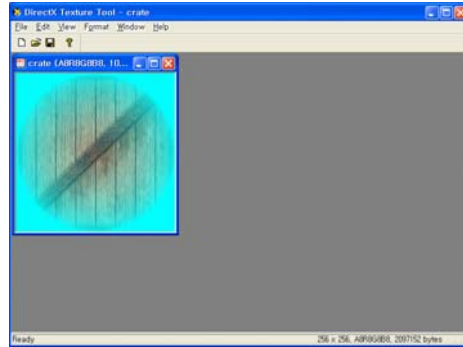
Creating Alpha Channel Using DX Tex Tool

- Alpha channel을 포함하는 DDS파일 제작법
 - DirectX9 SDK Texture Tool을 실행
 - Program->Microsoft DirectX 9.0 SDK Update (February 2005)->DirectX Utilities->DirectX Texture Tool 실행
 - 이미지에 alpha channel을 추가하기
 - File->Open에서 crate.jpg (원래 24-bit RGB)
 - Format->Change Surface Format을 32-bit A8R8G8B8로 변경



Creating Alpha Channel Using DX Tex Tool

- Alpha channel을 포함하는 DDS파일 제작법
 - 이미지의 alpha channel에 데이터를 넣기
 - 8 bit grey-scale image를 준비 (alphachannel.bmp)
 - File->Open Onto Alpha Channel Of This Texture 에서 alphachannel.bmp (8-bit grey-scale)를 읽어들임.
 - File->Save As에서 createalpha.dds로 저장



Example: MtrlAlpha

- 예제 MtrlAlpha에서는 알파블렌딩을 사용하여 Semi-transparent teapot을 그린다. 'A'와 'S'-key로 알파투명도를 조절한다.



Example: MtrlAlpha

```
#include "d3dUtility.h"
IDirect3DDevice9* Device = 0;
const int Width = 640;
const int Height = 480;
ID3DXMesh* Teapot = 0;
D3DMATERIAL9 TeapotMtrl;
IDirect3DVertexBuffer9* BgQuad = 0;
IDirect3DTexture9* BgTex = 0;
D3DMATERIAL9 BgMtrl;
bool Setup() {
    TeapotMtrl = d3d::RED_MTRL;
    TeapotMtrl.Diffuse.a = 0.5f;           // set alpha to 50% opacity
    BgMtrl = d3d::WHITE_MTRL;
    // Create teapot
    D3DXCreateTextureFromFile(Device, "crate.jpg", &BgTex);
    // Set Texture Filter States.
    Device->SetSamplerState(0, D3DSAMP_MAGFILTER, D3DTEXF_LINEAR);
    Device->SetSamplerState(0, D3DSAMP_MINFILTER, D3DTEXF_LINEAR);
    Device->SetSamplerState(0, D3DSAMP_MIPFILTER, D3DTEXF_POINT);
```

Example: MtrlAlpha

```
.....
// set alpha blending
Device->SetTextureStageState(0, D3DTSS_ALPHAARG1, D3DTA_DIFFUSE);
Device->SetTextureStageState(0, D3DTSS_ALPHAOP,
D3DTOP_SELECTARG1);
// set blending factors
Device->SetRenderState(D3DRS_SRCBLEND, D3DBLEND_SRCALPHA);
Device->SetRenderState(D3DRS_DESTBLEND, D3DBLEND_INVSRCALPHA);
...
// set camera
...
// set projection matrix
...
return true;
}
```

Example: MtrlAlpha

```
bool Display(float timeDelta)
{
    if (Device) {
        // increase/decrease alpha via keyboard input
        if (::GetAsyncKeyState('A') & 0x8000f)
            TeapotMtrl.Diffuse.a += 0.01f;
        if (::GetAsyncKeyState('S') & 0x8000f)
            TeapotMtrl.Diffuse.a -= 0.01f;
        // force alpha to [0, 1] interval
        if (TeapotMtrl.Diffuse.a > 1.0f)
            TeapotMtrl.Diffuse.a = 1.0f;
        if (TeapotMtrl.Diffuse.a < 0.0f)
            TeapotMtrl.Diffuse.a = 0.0f;
        ....
    }
}
```

Example: MtrlAlpha

```
.....
// render
Device->Clear(0, 0, D3DCLEAR_TARGET | D3DCLEAR_ZBUFFER, 0xffffffff,
1.0f, 0);
Device->BeginScene();
// draw background
D3DXMATRIX W;
D3DXMatrixIdentity(&W);
Device->SetTransform(D3DTS_WORLD, &W);
Device->SetFVF(Vertex::FVF);
Device->SetStreamSource(0, BgQuad, 0, sizeof(Vertex));
Device->SetMaterial(&BgMtrl);
Device->SetTexture(0, BgTex);
Device->DrawPrimitive(D3DPT_TRIANGLELIST, 0, 2);
.....
```

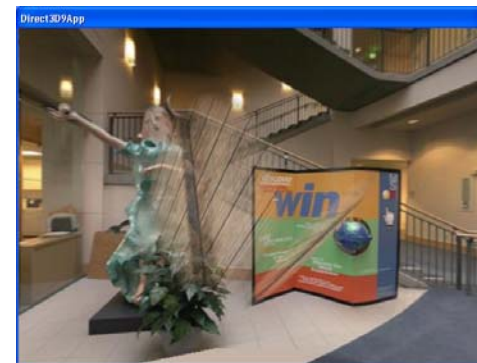
Example: MtrlAlpha

```
.....
// draw teapot
Device->SetRenderState(D3DRS_ALPHABLENDENABLE, true);
D3DXMatrixScaling(&W, 1.5f, 1.5f, 1.5f);
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);
}
return true;
}
```

Example: texAlpha

- 예제 texAlpha에서는 알파블렌딩을 사용하여 Semi-transparent cube를 그린다. 이 예제에서는 텍스처 알파 채널에서 알파값을 가져온다.



Example: texAlpha

```
IDirect3DTexture9*   CrateTex= 0;
Cube*                Box = 0;
D3DXMATRIX           CubeWorldMatrix;
IDirect3DVertexBuffer9* BackDropVB = 0;
IDirect3DTexture9*   BackDropTex = 0;

bool Setup()
{
    // create the backdrop quad
    // create the cube
    // load the textures and set filters
    D3DXCreateTextureFromFile(Device, "cratealpha.dds", &CrateTex);
    D3DXCreateTextureFromFile(Device, "lobbyxpos.jpg", &BackDropTex);
    // Set sampler filter
    // Set alpha blending using alpha channel in texture
    Device->SetTextureStageState(0, D3DTSS_ALPHAARG1, D3DTA_TEXTURE);
    Device->SetTextureStageState(0, D3DTSS_ALPHAOP,
    D3DTOP_SELECTARG1);
}
```

Example: texAlpha

```
// set blending factors
// disable lighting
// set camera
// set projection matrix
return true;
}
bool Display(float timeDelta)
{
    if (Device) {
        D3DXMATRIX xRot;
        D3DXMatrixRotationX(&xRot, D3DX_PI * 0.2f);
        static float y = 0.0f;
        D3DXMATRIX yRot;
        D3DXMatrixRotationY(&yRot, y);
        y += timeDelta;
        if (y >= 6.28f) y = 0.0f;
        CubeWorldMatrix = xRot * yRot;
    }
}
```

Example: texAlpha

```
// render
Device->Clear(0, 0, D3DCLEAR_TARGET | D3DCLEAR_ZBUFFER,
0xff0000ff, 1.0f, 0);
Device->BeginScene();
// draw backdrop
D3DXMATRIX I;      D3DXMatrixIdentity(&I);
Device->SetStreamSource(0, BackDropVB, 0, sizeof(Vertex));
Device->SetFVF(FVF_VERTEX);
Device->SetTexture(0, BackDropTex);
Device->SetTransform(D3DTS_WORLD, &I);
Device->DrawPrimitive(D3DPT_TRIANGLELIST, 0, 2);
// draw cube
Device->SetRenderState(D3DRS_ALPHABLENDENABLE, true);
if (Box) Box->draw(&CubeWorldMatrix, 0, CrateTex);
Device->SetRenderState(D3DRS_ALPHABLENDENABLE, false);
Device->EndScene();
Device->Present(0, 0, 0, 0);
}
return true;
}
```