

# Texturing

305890  
2009년 봄학기  
4/8/2009  
박경신

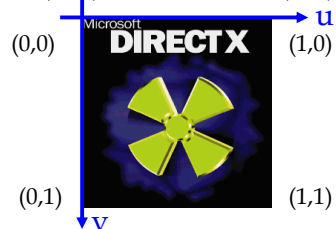
## Overview

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

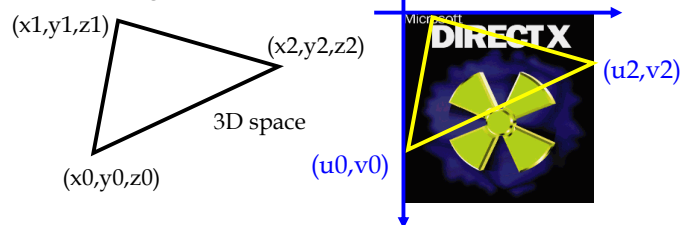
## Texture Coordinates

### □ 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  
    LPCWSTR 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): `D3DTEXTF_POINT`, default 방식, 가장 저품질, 가장 고속
- 선형 필터링 (Linear filtering): `D3DTEXTF_LINEAR`, 높은 품질, 비교적 고속, 가장 권장함.

```
Device->SetSamplerState(0, D3DSAMP_MAGFILTER, D3DTEXTF_LINEAR);  
Device->SetSamplerState(0, D3DSAMP_MINFILTER, D3DTEXTF_LINEAR);
```

- 비등방성 필터링 (Anisotropic filtering): `D3DTEXTF_ANISOTROPIC` 가장 고품질, 가장 느림.
  - 품질을 결정하는 `D3DSAMP_MAXANISOTROPY` 레벨로 지정해야함. 높은 값일 수록 높은 품질임. Default 1

```
Device->SetSamplerState(0, D3DSAMP_MAGFILTER,  
D3DTEXTF_ANISOTROPIC);
```

```
Device->SetSamplerState(0, D3DSAMP_MINFILTER,  
D3DTEXTF_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);
```

- `D3DTEXTF_NONE`: mipmap을 사용하지 않음
  - `D3DTEXTF_POINT`: Direct3D는 스크린 삼각형과 가장 비슷한 크기의 mip맵 레벨을 선택한다. 일단 레벨이 선택되면 지정된 min과 mag 태그에 따라 선택된 레벨을 필터링한다.
  - `D3DTEXTF_LINEAR`: 가장 비슷한 두 개의 mip맵 레벨에 min과 mag 필터에 따른 필터링을 적용한다. 즉, 선형적으로 조합된 두 개의 레벨을 이용해 최종 픽셀 컬러를 계산해낸다.
- Direct3D에서 mipmap 사용은 대부분 자동으로 처리됨
    - `D3DXCreateTextureFromFile`을 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

- `D3DTEXTUREADDRESS_WRAP`: repeat the texture on every integer junction
- `D3DTEXTUREADDRESS_MIRROR`: every other row and column is a mirrors image of the preceding row or column
- `D3DTEXTUREADDRESS_CLAMP`: smear the color of edge pixels
- `D3DTEXTUREADDRESS_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);

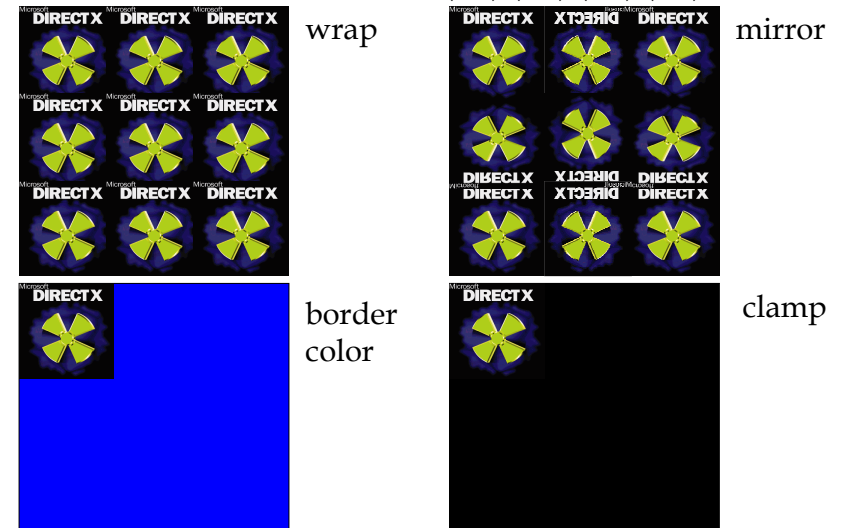
// border color
Device->SetSamplerState(0, D3DSAMP_ADDRESSU, D3DADDRESS_BORDER);
Device->SetSamplerState(0, D3DSAMP_ADDRESSV, D3DADDRESS_BORDER);
Device->SetSamplerState(0, D3DSAMP_BORDERCOLOR, 0x000000ff);

// 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, 1);
    Quad->Unlock();
}
```

## Example: TexQuad

```
D3DXCreateTextureFromFile(Device, "dx5_logo.bmp", &Tex);
Device->SetTexture(0, Tex);
Device->SetSamplerState(0, D3DSAMP_MAGFILTER, D3DTEXTF_LINEAR);
Device->SetSamplerState(0, D3DSAMP_MINFILTER, D3DTEXTF_LINEAR);
Device->SetSamplerState(0, D3DSAMP_MIPFILTER, D3DTEXTF_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.cpp

```
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();
}
```

## 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, -1.0f, 0.0f, 0.0f, 0.0f); // left
v[17] = Vertex(-1.0f, 1.0f, 1.0f, -1.0f, 0.0f, 0.0f, 0.0f, 1.0f);
v[18] = Vertex(-1.0f, 1.0f, -1.0f, -1.0f, 0.0f, 0.0f, 1.0f, 1.0f);
v[19] = Vertex(-1.0f, -1.0f, -1.0f, -1.0f, 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);
```

cube.cpp

```
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();
}
```

## 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