

Mesh Part II

305890
2009년 봄학기
5/13/2009
박경신

Mesh Part II

- ID3DXBuffer
- XFiles 데이터를 ID3DXMesh 객체로 읽어 들이는 방법
- 프로그레시브 메쉬 (Progressive mesh)를 이용하여 얻을 수 있는 이점에 대한 이해와 메쉬 인터페이스 ID3DXPMesh 이용 방법
- 경계 볼륨 (Bounding volume)의 정의와 용도에 대한 학습과 D3DX 함수를 이용하여 만드는 방법
- 예제

ID3DXBuffer

- ID3DXBuffer
 - D3DX가 연속적인 메모리 블록에 데이터를 저장하기 위해 이용하는 범용 데이터 구조체

```
LPVOID GetBufferPointer(VOID); // 데이터시작 포인터 반환
DWORD GetBufferSize(VOID); // 버퍼 크기를 byte단위로 반환
```

 - 데이터 타입은 프로그래머가 관리해야함

```
DWORD* info = (DWORD*)adjacencyInfo->GetBufferPointer();
D3DXMATERIAL *mtrl =
    (D3DXMATERIAL*)mtrlBuffer->GetBufferPointer();
```

 - 메모리 누출을 막기 위해, 반드시 이용 후에는 객체를 release시킴

```
adjacencyInfo->Release();
mtrlBuffer->Release();
```

ID3DXBuffer

- ID3DXBuffer 생성하기

```
HRESULT D3DXCreateBuffer (
    DWORD NumBytes, // 버퍼 크기 (in bytes)
    LPD3DXBUFFER *ppBuffer); // ID3DXBuffer
```

 - 예제:

```
// 4개의 정수를 보관할 수 있는 buffer의 생성
ID3DXBuffer* buffer = 0;
D3DXCreateBuffer( 4*sizeof(int), &buffer );
```

XFiles

- 3D 모델러를 사용하여 복잡한 물체를 생성
 - 3DS Max (www.discreet.com)
 - LightWave3D (www.newtek.com)
 - Maya (www.aliaswavefront.com)
 - Multigen Creator (www.multigen.com)
 - Soft Image (www.softimage.com)
 - 생성된 메쉬 데이터 (기하정보, 채질, 애니메이션 등과 같은 데이터) 를 XFile 포맷으로 변환 가능
- 3D 모델 Exporter
 - Okino Polytrans (www.okino.com)
- 3D Max object를 .x 파일로 export 해주는 plug-in tool - Pandasoft
 - http://www.andytather.co.uk/Panda/directxmax_downloads.aspx

Converting 3DS MAX to X File

- Converter를 사용하는 방법
 - Run 'conv3ds.exe'
 - Command prompt 상태에서 "conv3ds File.3ds"하면 .x file 생성
 - Conv3ds의 옵션 인수의 지정
 - http://telnet.or.kr/sec_directx/index.html?init_mode=api_contents_read&api_no=86
 - 현재 Microsoft사에서 찾아도 다운로드 위치를 찾을 수 없음. 수업 블로그를 통해서 다운로드 받을 것.

Load Xfiles

- ID3DXMesh 객체를 생성하고 XFile의 기하정보 데이터를 읽어 들이는 함수: D3DXLoadMeshFromX

```
HRESULT D3DXLoadMeshFromX (  
    LPCSTR pFilename,  
    DWORD Options,  
    LPDIRECT3DDEVICE9 pDevice,  
    LPD3DXBUFFER* ppAdjacency,  
    LPD3DXBUFFER* ppMaterials,  
    LPD3DXBUFFER* ppEffectInstances,  
    PDWORD pNumMaterials,  
    LPD3DXMESH* ppMesh);
```

 - pFilename - 읽어 들이고자 하는 Xfile의 파일명
 - Options - 생성 플래그 (D3DXMESH 열거형)
 - ppAdjacency, ppMaterials, ppEffectInstaces, pNumMaterials, ppMesh - 리턴 받을 인자들

Load XFiles

- 예제:

```
HRESULT hr = 0;  
  
ID3DXBuffer* adjBuffer = 0;  
ID3DXBuffer* mtrlBuffer = 0;  
DWORD numMtrls = 0;  
  
hr = D3DXLoadMeshFromX( "bigship1.x",  
    D3DXMESH_MANAGED,  
    Device,  
    &adjBuffer,  
    &mtrlBuffer,  
    0,  
    &numMtrls, // # of D3DXMATERIAL structures  
              // in mtrlBuffer  
    &Mesh ); // loaded ID3DXMesh
```

XFile Materials

- Xfile에 저장된 재질 정보의 구조는 D3DXMATERIAL

```
typedef struct D3DXMATERIAL {
    D3DMATERIAL9 MatD3D;
    LPSTR pTextureFilename;
} D3DXMATERIAL, *LPD3DXMATERIAL;
```

- XFile 은 텍스처 데이터를 직접 포함하지 않음. 파일명만 보관함.
- 리턴된 D3DMATERIAL 배열의 *i*번째 항목은 *i*번째 서브셋과 대응되도록 XFile 정보를 읽어 들임.

Example: XFile

```
ID3DXMesh* Mesh = 0;
std::vector<D3DXMATERIAL> Mtrls(0);
std::vector<IDirect3DTexture9*> Textures(0);
bool Setup() { // Load XFile data
    ID3DXBuffer* adjBuffer = 0; ID3DXBuffer* mtrlBuffer = 0; DWORD numMtrls = 0;
    HRESULT hr = D3DXLoadMeshFromX("bigship1.x", D3DXMESH_MANAGED, Device,
    &adjBuffer, &mtrlBuffer, 0, &numMtrls, &Mesh);
    if (FAILED(hr)) { ::MessageBox(0, "D3DXLoadMeshFromX() - FAILED", 0, 0); return false; }
    if (mtrlBuffer != 0 && numMtrls != 0) { // extract materials, and load textures
        D3DXMATERIAL* mtrls = (D3DXMATERIAL*) mtrlBuffer->GetBufferPointer();
        for (DWORD i = 0; i < numMtrls; i++) {
            // loading시 MatD3d에 ambient값을 가지지 않으므로 지금 지정해줌
            mtrls[i].MatD3D.Ambient = mtrls[i].MatD3D.Diffuse;
            mtrls.push_back(mtrls[i].MatD3D);
            if (mtrls[i].pTextureFilename != 0) { // if it has an associative texture
                IDirect3DTexture9* tex = 0;
                D3DXCreateTextureFromFile(Device, mtrls[i].pTextureFilename, &tex);
                Textures.push_back(tex);
            } else { Textures.push_back(0); } // if it has no texture
        }
    }
    d3d::Release<ID3DXBuffer*>(mtrlBuffer);
}
```

Example: XFile

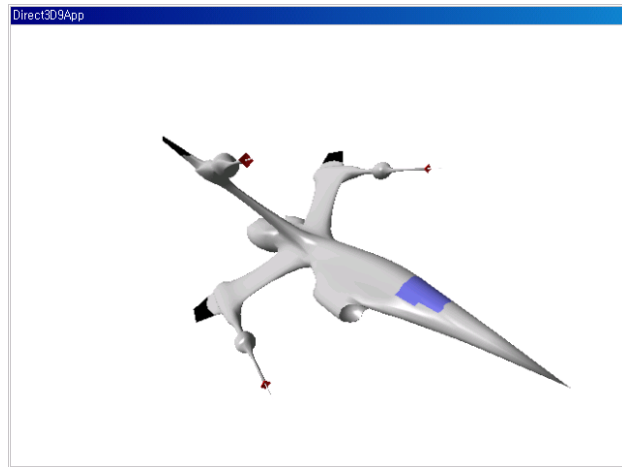
```
// optimize the mesh to generate an attribute table
hr = Mesh->OptimizeInplace(D3DXMESHOPT_ATTRSORT |
    D3DXMESHOPT_COMPACT | D3DXMESHOPT_VERTEXCACHE,
    (DWORD*) adjBuffer->GetBufferPointer(), 0, 0, 0);
d3d::Release<ID3DXBuffer*>(adjBuffer);
if (FAILED(hr)) {
    ::MessageBox(0, "OptimizeInplace() - FAILED", 0, 0);
    return false;
}
// textures
// lights
// camera
// projection matrix
return true;
}

void Cleanup() {
    d3d::Release<ID3DXMesh*>(Mesh);
    for (i=0; i<(int)Textures.size(); i++) d3d::Release<IDirect3DTexture9*>(Textures[i]);
}
```

Example: XFile

```
bool Display(float timeDelta) {
    if (Device) {
        D3DXMATRIX xRot, yRot, World;
        static float y = 0.0f;
        D3DXMatrixRotationX(&xRot, D3DX_PI * 0.2f);
        D3DXMatrixRotationY(&yRot, y);
        y += timeDelta;
        if (y >= 6.28f) y = 0.0f;
        World = xRot * yRot;
        Device->SetTransform(D3DTS_WORLD, &World);
        Device->Clear(0, 0, D3DCLEAR_TARGET | D3DCLEAR_ZBUFFER,
            0x00000000, 1.0, 0);
        Device->BeginScene();
        for (int i=0; i < (int) Mtrls.size(); i++) {
            Device->SetMaterial(&Mtrls[i]);
            Device->SetTexture(0, Textures[i]);
            Mesh->DrawSubset(i);
        }
        Device->EndScene();
        Device->Present(0, 0, 0, 0);
    }
    return true;
}
```

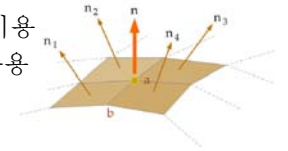
Example: XFile



Creating Vertex Normals

□ ID3DXBaseMesh의 버텍스 법선 계산 함수

- XFile에 버텍스 법선 데이터가 없다면, 조명을 위해 직접 vertex normal을 계산해야 한다.
- 계산 원리 - 이웃하는 면 법선 벡터의 평균을 이용
- 인접 정보 - 중복된 버텍스는 무시하기 위해 사용



HRESULT D3DXComputeNormals (
LPD3DXBASEMESH pMesh,
CONST DWORD *pAdjacency);

- pMesh - 버텍스 포맷은 반드시 D3DFVF_NORMAL 플래그를 포함해야 한다. D3DXComputeNormals 함수 이용 전에 버텍스 포맷 지정
- pAdjacency - 각 면마다 3개의 DWORD를 가지는 배열. 안 쓰면 NULL로 지정한다.

Creating Vertex Normals

□ D3DXComputeNormals 사용법

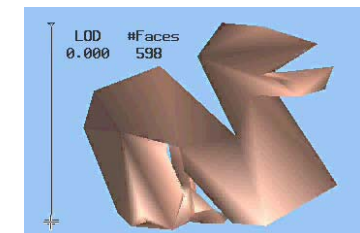
- XFile이 vertex normal 를 가지고 있지 않다면 해당하는 ID3DXMesh 객체의 FVF도 D3DFVF_NORMAL flag가 없음.

```
if( !(pMesh->GetFVF() & D3DFVF_NORMAL) ) {
    ID3DXMesh* pTempMesh = 0;
    pMesh->CloneMeshFVF (
        D3DXMESH_MANAGED,
        pMesh->GetFVF() | D3DFVF_NORMAL,
        Device,
        &pTempMesh );
    D3DXComputeNormals( pTempMesh, 0);
    pMesh->Release();
    pMesh = pTempMesh;
}
```

Progressive Mesh

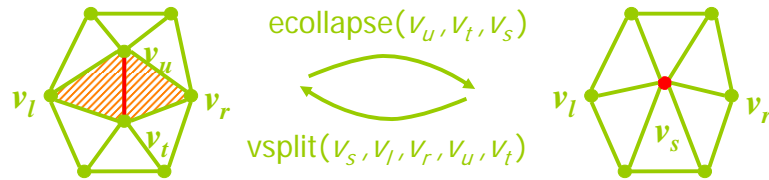
□ ID3DXPMesh 인터페이스에 의해 표현

- 경계 상실 변환 (ECT: edge collapse transformation) sequence를 적용하여 메쉬를 단순화할 수 있도록 해준다
 - 한 ECT는 한 vertex와 1~2개의 face를 제거함
 - 각 ECT는 (vertex split를 통해) reversible함



Progressive Mesh

ECT vs. 버텍스 분할 (vertex split)



- ECT를 이용한 단순화 과정을 되돌려 원래 상태의 메쉬로 돌아갈 수 있음



Progressive Mesh

- 이용방법: 카메라와의 거리에 따라 메쉬의 LOD (Level-Of-Details) 를 조정할 수 있음
 - 작고 먼 메쉬는 크고 가까운 메쉬만큼 많은 삼각형이 필요치 않음
 - 카메라와의 거리가 감소하면 메쉬의 세부(detail)를 높이고 거리가 증가하면 세부를 감소시킴
- 참조논문 “Progressive Meshes” (Hugues Hoppe, SIGGRAPH 96)
 - <http://research.microsoft.com/~hoppe/>

Creating Progressive Mesh

ID3DXPMesh 객체 생성 함수 - D3DXGeneratePMesh

```
HRESULT D3DXGeneratePMesh (
    LPD3DXMESH pMesh,
    CONST DWORD *pAdjacency,
    CONST LPD3DXATTRIBUTEWEIGHTS pVertexAttributeWeights,
    CONST FLOAT *pVertexWeights,
    DWORD MinValues,
    DWORD Options,
    LPD3DXPMESH *ppPMesh);
```

- pVertexAttributeWeights - D3DXATTRIBUTEWEIGHTS 배열 (pMesh->GetNumVertices() 크기)로의 포인터
- pVertexWeights - pMesh->GetNumVertices() 크기를 갖는 float array
- MinValue - 단순화 결과로 얻어질 최소한의 버텍스나 면 개수
- Options - D3DXMESHSIMP_VERTEX, D3DXMESHSIMP_FACE, MinValue가 vertex/face에 적용됨을 지정함

Vertex Attribute Weights

버텍스 속성 영향력 구조체

```
typedef struct _D3DXATTRIBUTEWEIGHTS {
    FLOAT Position;
    FLOAT Boundary; // blend weight
    FLOAT Normal; // normal
    FLOAT Diffuse; // diffuse light value
    FLOAT Specular; // specular light value
    FLOAT Texcoord[8]; // texture coordinates
    FLOAT Tangent;
    FLOAT Binormal;
} D3DXATTRIBUTEWEIGHTS
```

- 각 요소의 영향력을 지정할 수 있도록 함
- 값이 높을수록 버텍스의 영향력이 높아져 단순화 과정에서 제거될 가능성이 낮아짐 (최소값 = 0.0)
- default 영향력 값을 이용하는 것이 좋음

Vertex Attribute Weights

□ Default vertex attribute weights:

```
D3DXATTRIBUTEWEIGHTS AttributeWeights;  
AttributeWeights.Position = 1.0;  
AttributeWeights.Boundary = 1.0;  
AttributeWeights.Normal = 1.0;  
AttributeWeights.Diffuse = 0.0;  
AttributeWeights.Specular = 0.0;  
AttributeWeights.Texcoord[8] = { 0, 0, 0, 0, 0, 0, 0, 0 };
```

ID3DXPMesh Methods

- ID3DXPMesh 인터페이스는 ID3DXBaseMesh 인터페이스를 상속 받음
 - ID3DXMesh의 모든 기능을 가짐



ID3DXPMesh Methods

- ID3DXBaseMesh로부터 상속받은 것 외에 추가된 method
 - DWORD ID3DXPMesh::GetMaxFaces(VOID);
 - 최대 face 수
 - DWORD ID3DXPMesh::GetMaxVertices(VOID);
 - 최대 vertex 수
 - DWORD ID3DXPMesh::GetMinFaces(VOID);
 - 최소 face 수
 - DWORD ID3DXPMesh::GetMinVertices(VOID);
 - 최소 vertex 수
 - HRESULT ID3DXPMesh::SetNumFaces(DWORD Faces);
 - LOD 지정함수. 호출 후의 face의 수는 지정한 faces값보다 1 작을 수 있음. 항상 최대/최소 범위 내에 있도록 함
 - HRESULT ID3DXPMesh::SetNumVertices(DWORD Vertices);
 - LOD 지정함수. 호출 후의 vertex의 수는 지정한 vertices값보다 1 작을 수 있음. 항상 최대/최소 범위 내에 있도록 함

ID3DXPMesh Methods

- ID3DXBaseMesh로부터 상속받은 것 외에 추가된 method
 - HRESULT ID3DXPMesh::TrimByFaces(DWORD NewFacesMin, DWORD NewFacesMax, DWORD *rgiFaceRemap, DWORD *rgiVertRemap)
 - 새로운 최소/최대 face수를 지정함. 새로운 최소/최대 범위는 현재의 최소/최대 범위 즉, GetMinFaces()와 GetMaxFaces() 내에 있어야 함.
 - rgiFaceRemap과 rgiVertRemap은 각 face/vertex당 한 DWORD로 face/vertex의 remap 정보를 가짐 - 즉, i번째 face/vertex가 어디로 이동했는지
 - HRESULT ID3DXPMesh::TrimByVertices(DWORD NewVerticesMin, DWORD NewVerticesMax, DWORD *rgiFaceRemap, DWORD *rgiVertRemap)
 - 새로운 최소/최대 vertex수를 지정함. 새로운 최소/최대 범위는 현재의 최소/최대 범위 즉, GetMinVertices()와 GetMaxVertices() 내에 있어야 함.

Example: Progressive Mesh

```
ID3DXMesh* SourceMesh = 0;
ID3DXPMesh* PMesh = 0; // progressive mesh
bool Setup() { // Load XFile data
    ID3DXBuffer* adjBuffer = 0; ID3DXBuffer* mtrlBuffer = 0; DWORD numMtrls = 0;
    HRESULT hr = D3DXLoadMeshFromX("bigship1.x", D3DXMESH_MANAGED, Device,
    &adjBuffer, &mtrlBuffer, 0, &numMtrls, &SourceMesh);
    if (FAILED(hr)) { ::MessageBox(0, "D3DXLoadMeshFromX() - FAILED", 0, 0); return false; }
    if (mtrlBuffer != 0 && numMtrls != 0) { // extract materials, and load textures
        D3DXMATERIAL* mtrls = (D3DXMATERIAL*) mtrlBuffer->GetBufferPointer();
        for (DWORD i = 0; i < numMtrls; i++) {
            // loading시 MatD3d에 ambient값을 가지지 않으므로 지금 지정해줌
            mtrls[i].MatD3D.Ambient = mtrls[i].MatD3D.Diffuse;
            Mtrls.push_back(mtrls[i].MatD3D);
            if ( mtrls[i].pTextureFilename != 0 ) { // if it has an associative texture
                IDirect3DTexture9* tex = 0;
                D3DXCreateTextureFromFile(Device, mtrls[i].pTextureFilename, &tex);
                Textures.push_back( tex );
            } else { Textures.push_back(0); } // if it has no texture
        }
    }
    d3d::Release<ID3DXBuffer*>( mtrlBuffer);
}
```

Example: Progressive Mesh

```
// optimize the mesh to generate an attribute table
hr = SourceMesh->OptimizeInplace(D3DXMESHOPT_ATTRSORT |
    D3DXMESHOPT_COMPACT | D3DXMESHOPT_VERTEXCACHE,
    (DWORD*) adjBuffer->GetBufferPointer(),
    (DWORD*) adjBuffer->GetBufferPointer(), // new adjacency information
    0, 0);
if (FAILED(hr)) {
    ::MessageBox(0, "OptimizeInplace() - FAILED", 0, 0);
    d3d::Release<ID3DXBuffer*>(adjBuffer);
    return false;
}
// generate the progressive mesh
hr = D3DXGeneratePMesh(SourceMesh,
    (DWORD*) adjBuffer->GetBufferPointer(), 0, 0, 1,
    D3DXMESHSIMP_FACE, &PMesh);
d3d::Release<ID3DXMesh*>(SourceMesh);
d3d::Release<ID3DXBuffer*>(adjBuffer);
if (FAILED(hr)) {
    ::MessageBox(0, "D3DXGeneratePMesh() - FAILED", 0, 0);
    return false;
}
```

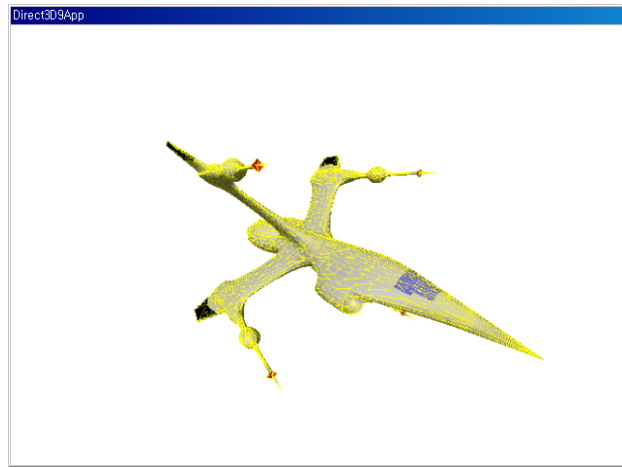
Example: Progressive Mesh

```
// set to original detail - 이렇게 하지 않으면 가장 낮은 해상도로 렌더링함.
DWORD maxFaces = PMesh->GetMaxFaces();
PMesh->SetNumFaces(maxFaces);
// textures
// lights
// camera
// projection matrix
return true;
}
```

Example: Progressive Mesh

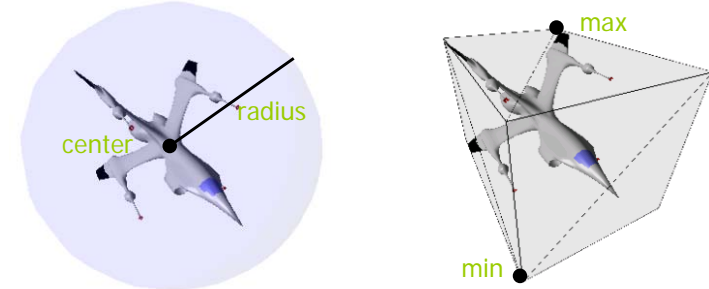
```
bool Display(float timeDelta) {
    if (Device) { //update mesh resolution
        int numFaces = PMesh->GetNumFaces(); // current number of faces
        if (::GetAsyncKeyState('A') & 0x8000f) { // add a face
            PMesh->SetNumFaces(numFaces + 1);
            if (PMesh->GetNumFaces() == numFaces)
                PMesh->SetNumFaces(numFaces + 2); // ECT 역변환이 가능하도록 함
        }
        if (::GetAsyncKeyState('S') & 0x8000f) { // remove a face
            PMesh->SetNumFaces(numFaces - 1);
        }
        Device->Clear(0, 0, D3DCLEAR_TARGET | D3DCLEAR_ZBUFFER,
            0x00000000, 1.0, 0);
        Device->BeginScene();
        for (int i=0; i < (int) Mtrls.size(); i++) {
            // draw pmesh using draw subset
            Device->SetMaterial( &Mtrls[i] ); // draw wireframe outline
            Device->SetRenderState(D3DRS_FILLMODE, D3DFILL_WIREFRAME);
            PMesh->DrawSubset(i);
            Device->SetRenderState(D3DRS_FILLMODE, D3DFILL_SOLID);
        }
        Device->EndScene();
        Device->Present(0, 0, 0, 0);
    }
    return true;
}
```

Example: Progressive Mesh



Bounding Volumes

- 가장 자주 이용되는 경계 볼륨 - 구(sphere), 상자(box)
 - Bounding Sphere: 중심점, 반지름으로 정의됨
 - Bounding Box: 최소점, 최대점으로 정의됨
- 그 외 - 원기둥(cylinder), 타원체(ellipsoid), 마름모꼴(lozenge: 방석모양), 캡슐(capsule)
- 용도 - 가시성 테스트(visibility test)와 충돌 테스트의 가속



Bounding Volumes

- D3DX 라이브러리가 제공하는 경계 구체와 경계 상자를 계산하는 함수

```
HRESULT D3DXComputeBoundingSphere (  
    LPD3DXVECTOR3 pFirstPosition, // 첫번째 vertex  
    DWORD NumVertices,  
    DWORD dwStride, // 바이트 단위의 버텍스 크기  
    D3DXVECTOR3* pCenter, // center of bounding sphere  
    FLOAT* pRadius); // radius of bounding sphere
```

```
HRESULT D3DXComputeBoundingBox (  
    LPD3DXVECTOR3 pFirstPosition,  
    DWORD NumVertices,  
    DWORD dwStride, // 바이트 단위의 버텍스 크기  
    D3DXVECTOR3* pMin, // lower-left corner of bounding box  
    D3DXVECTOR3* pMax); // upper-right corner of bounding box
```

Bounding Volumes - d3dUtility

- 새로운 특수 상수의 추가
const float INFINITY = FLT_MAX; //float에 저장할 수 있는 최대 수
const float EPSILON = 0.001f; // 0의 의미로 사용하는 매우 작은 수

- BoundingSphere/Box 타입 추가
struct BoundingBox {
 BoundingBox();
 bool isPointInside(D3DXVECTOR3& p);
 D3DXVECTOR3 _min;
 D3DXVECTOR3 _max;
};
struct BoundingSphere {
 BoundingSphere();
 D3DXVECTOR3 _center;
 float _radius;
};

Bounding Volumes – d3dUtility

□ BoundingSphere/Box 타입 추가

```
d3d::BoundingBox::BoundingBox() {
    _min.x = _min.y = _min.z = d3d::INFINITY;
    _max.x = _max.y = _max.z = -d3d::INFINITY;
};
d3d::BoundingBox::IsPointInside(D3DXVECTOR3& p) {
    if (p.x >= _min.x && p.y >= _min.y && p.z >= _min.z &&
        p.x <= _max.x && p.y <= _max.y && p.z <= _max.z)
        return true;
};
d3d::BoundingSphere::BoundingSphere() {
    _radius = 0.0f;
};
```

Example: Bounding Volumes

```
ID3DXMesh* Mesh = 0;
ID3DXMesh* SphereMesh = 0; // bounding sphere
ID3DXMesh* BoxMesh = 0; // bounding box
bool RenderBoundingSphere = true;
bool ComputeBoundingSphere(ID3DXMesh* mesh, d3d::BoundingSphere *sphere);
bool ComputeBoundingBox(ID3DXMesh* mesh, d3d::BoundingBox *box);
bool Setup() {
    // load XFile data
    // extract the materials, load textures
    // optimize the mesh
    // compute bounding sphere and bounding box
    d3d::BoundingSphere boundingSphere;
    d3d::BoundingBox boundingBox;
    ComputeBoundingSphere(Mesh, &boundingSphere);
    ComputeBoundingBox(Mesh, &boundingBox);
    D3DXCreateSphere(Device, boundingSphere._radius, 20, 20, &SphereMesh, 0);
    D3DXCreateBox(Device, boundingBox._max.x - boundingBox._min.x,
        boundingBox._max.y - boundingBox._min.y, boundingBox._max.z - boundingBox._min.z,
        &BoxMesh, 0);
    // textures, light, camera, projection matrix
    return true;
}
```

Example: Bounding Volumes

```
bool Display(float timeDelta) {
    if (Device) {
        // update mesh resolution
        // render
        // clear, BeginScene, draw mesh

        // draw bounding volume in blue, and at 10% opacity
        D3DXMATERIAL9 blue = d3d::BLUE_MTRL;
        blue.Diffuse.a = 0.10f;
        Device->SetMaterial(&blue);
        Device->SetTexture(0, 0); // disable texture
        Device->SetRenderState(D3DRS_ALPHABLENDENABLE, true);
        Device->SetRenderState(D3DRS_SRCBLEND, D3DBLEND_SRCALPHA);
        Device->SetRenderState(D3DRS_DESTBLEND, D3DBLEND_INVSRCALPHA);
        if (RenderBoundingSphere) SphereMesh->DrawSubset(0);
        else BoxMesh->DrawSubset(0);
        // EndScene, Present
    }
    return true;
}
```

Example: Bounding Volumes

```
bool ComputeBoundingSphere(ID3DXMesh* mesh, d3d::BoundingSphere *sphere) {
    HRESULT hr = 0;
    BYTE* v = 0;
    mesh->LockVertexBuffer(0, (void**) &v);
    hr = D3DXComputeBoundingSphere((D3DXVECTOR3*) v, mesh->GetNumVertices(),
        D3DXGetFVFVertexSize(mesh->GetFVF()),
        &sphere->_center, &sphere->_radius);

    mesh->UnlockVertexBuffer();
    if (FAILED(hr)) return false;
    return true;
}
bool ComputeBoundingBox(ID3DXMesh* mesh, d3d::BoundingBox *box) {
    HRESULT hr = 0;
    BYTE* v = 0;
    mesh->LockVertexBuffer(0, (void**) &v);
    hr = D3DXComputeBoundingBox((D3DXVECTOR3*) v, mesh->GetNumVertices(),
        D3DXGetFVFVertexSize(mesh->GetFVF()),
        &box->_min, &box->_max);

    mesh->UnlockVertexBuffer();
    if (FAILED(hr)) return false;
    return true;
}
```

Example: Bounding Volumes

