2009, Copyright All rights reserved

3D Game Programming 08

afewhee@gmail.com

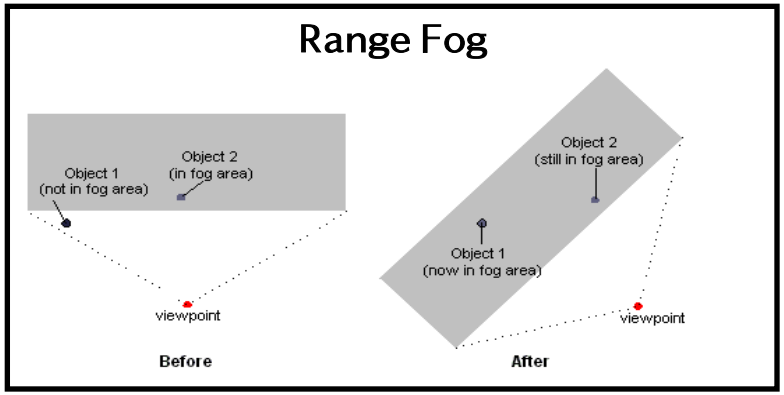
GaLic. Heesung Oh
2009, Copyright All rights reserved

- Fog
- 충돌 상자, 구
- Antialiasing
- X-File Loading
- 정밀한 Frame 계산
- 실습

2009, Copyright All rights reserved

1. Fog (안개 효과)

- 종류
  - Vertex Fog: 정점에 Fog 적용
  - Pixel Fog: Pixel 단위로 Fog 적용



- Vertex Fog
  - Range-Based Fog

- 정점 포그 프로그램 방법
  - 렌더링 머신 포그 활성화
    - pDevice->SetRenderState(D3DRS_FOGENABLE, TRUE);
  - 포그 색상 설정
    - pDevice->SetRenderState(D3DRS_FOGCOLOR, Color);
  - 선형 포그 (Linear Fog) 의 경우 시작, 끝 설정:
    - float Start = 500.f,
    - float End = 2000.f;
    - pDevice->SetRenderState(D3DRS_FOGVERTEXMODE, D3DFOG_LINEAR);
    - pDevice->SetRenderState(D3DRS_FOGSTART, *(DWORD *)(&Start));
    - pDevice->SetRenderState(D3DRS_FOGEND, *(DWORD *)(&End));
  - 선형 포그 이외는 밀도(Density설정)
    - pDevice->SetRenderState(D3DRS_FOGVERTEXMODE, {D3DFOG_EXP| D3DFOG_EXP2});
    - pDevice->SetRenderState(D3DRS_FOGDENSITY, *(DWORD *)(&Density));
  - Range Fog 설정
    - pDev->SetRenderState(D3DRS_RANGEFOGENABLE, {TRUE|FALSE});

- 픽셀 포그 설정 방법
  - 렌더링 머신의 포그를 TABLEMODE로 설정
  - pDevice->SetRenderState(D3DRS_FOGTABLEMODE, {D3DFOG_EXP| D3DFOG_EXP2});

2009, Copyright All rights reserved

2. 충돌 상자, 구

- 충돌 상자(Bounding Box) 구하기

```
D3DXComputeBoundingBox(
                        (D3DXVECTOR3*)(정점 시작 주소)
                        , 정점 수
                        , 한 정점의 크기
                        , 반환 경계 최소값
                        , 반환 경계 최대 값  );
```

- 충돌 구(Bounding Sphere)구하기

```
D3DXComputeBoundingSphere(
                        (D3DXVECTOR3*)(정점 시작 주소)
                        , 정점 수
                        , 한 정점의 크기
                        , 구의 중심
                        , 구의 반지름);
```

- 프로그램 예

```
D3DXVECTOR3         m_vcMax;
D3DXVECTOR3         m_vcMin;
D3DXVECTOR3         m_vcCenter;
FLOAT               m_fRadius;
...

D3DXVECTOR3* pVtx=NULL;
if (FAILED(m_pMsh->LockVertexBuffer(D3DLOCK_READONLY, (void**)&pVtx)))
     return -1;

D3DXComputeBoundingBox((D3DXVECTOR3*)pVtx
                                    , m_pMsh->GetNumVertices()
                                    , D3DXGetFVFVertexSize(m_pMsh->GetFVF())
                                    , &m_vcMin, &m_vcMax);

D3DXComputeBoundingSphere((D3DXVECTOR3*)pVtx
                                    , m_pMsh->GetNumVertices()
                                    , D3DXGetFVFVertexSize(m_pMsh->GetFVF())
                                    , &m_vcCenter, &m_fRadius);

m_pMsh->UnlockVertexBuffer();
```

GaLic. Heesung Oh
2009, Copyright All rights reserved

- Multi-Sampling 지원 확인

```
Ex)
for(int nType=D3DMULTISAMPLE_16_SAMPLES; nType>=0; --nType)
{
    if(SUCCEEDED(m_pD3D->CheckDeviceMultiSampleType(D3DADAPTER_DEFAULT
      , D3DDEVTYPE_HAL
      , Back Buffer Surface Format
      , Windowed?
      , (D3DMULTISAMPLE_TYPE)nType
      , &dQualityLevels)))
    {
      m_d3dpp.MultiSampleType = (D3DMULTISAMPLE_TYPE)nType;
      m_d3dpp.MultiSampleQuality = dQualityLevels-1;
      break;
    }
}
```

- Flags 값이 D3DPRESENTFLAG_LOCKABLE_BACKBUFFER는 안됨
  - (X)m_d3dpp.Flags |= D3DPRESENTFLAG_LOCKABLE_BACKBUFFER;

- 렌더링에서 안티알리아싱 활성화
  - pDevice->SetRenderState (D3DRS_MULTISAMPLEANTIALIAS, TRUE);
  - pDevice->SetRenderState (D3DRS_ANTIALIASEDLINEENABLE, TRUE);

GaLic. Heesung Oh
2009, Copyright All rights reserved

- Load Mesh
  - D3DXLoadMeshFromX()
- Load Material
  - 하나의 Material은 {0,1} 숫자의 텍스처가 존재 → Matrial 수 만큼 돌면서 텍스처 생성
  - Material의 숫자는 Geometry 수와 같음
- Mesh Optimize
  - m_pMsh->OptimizeInplace
- Geometry 생성(옵션)
  - 필요에 따라 Geometry 생성
  - 하나의 Attribute는 Material과 대응
  - Geometry수는 Attribute수와 일치
    m_pMsh->GetAttributeTable(NULL, &m_nGeo);
- 렌더링
  - Material 또는 Geometry수 만큼 돌면서 렌더링

```
for(i = 0; i < m_nGeo; i++)
{
    pDevice->SetMaterial( &m_pMtl[i] );
    pDevice->SetTexture(0, m_pTx[i]);
    m_pMsh->DrawSubset(i);
}
```


2009, Copyright All rights reserved

5. 정밀한 Frame 계산

- 고해상도 타이머
  - 해상도가 1/1,000,000초인 타이머

- Win API 함수
  - QueryPerformanceFrequency(): 진동수 반환
  - QueryPerformanceCounter(): 수행 숫자 반환

  - FPS 구하기(1/1000초 기준)

```
QueryPerformanceCounter(&dCur);
QueryPerformanceFrequency(&Freq);
dTime = DOUBLE(dCur.QuadPart)/DOUBLE(Freq.QuadPart);
dTime *=1000.;
```

- ASM 이용
  - 인텔 MMX이상

  - FPS 구하기(1/1000초)

```
#define cpuid __asm __emit 0fh __asm __emit 0a2h
#define rdtsc __asm __emit 0fh __asm __emit 031h
__asm
{
    cpuid
    rdtsc
    mov dCur.LowPart, eax
    mov dCur.HighPart, edx
}

dTime = DOUBLE(dCur.QuadPart)/DOUBLE(Freq.QuadPart);
dTime *=1000.;
```

GaLic. Heesung Oh
2009, Copyright All rights reserved

- X-File에 대한 3D 모델 클래스를 작성해 보시오.
- 지형 위에 X-File들을 배치해 보시오
- X-File에 대한 피킹(Picking)을 구현해 보시오.