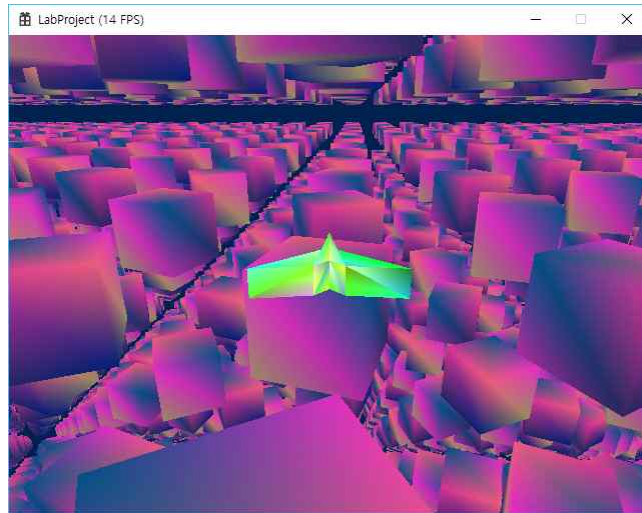


□ 예제 프로그램 16: LabProject15(절두체 컬링: Frustum Culling)

예제 프로그램 LabProject11을 기반으로 여러 개의 직육면체들을 절두체 컬링을 하여 렌더링하도록 구현한다.



① 새로운 프로젝트의 생성

먼저 새로운 프로젝트 LabProject15를 생성한다. “LabProjects” 솔루션을 열고 솔루션 탐색기에서 마우스 오른쪽 버튼으로 『솔루션 LabProjects』를 선택하고 메뉴에서 『추가』, 『새 프로젝트』를 차례로 선택한다. 그러면 『새 프로젝트 대화상자』가 나타난다. 그러면 프로젝트 이름 “LabProject15”을 입력하고 『확인』을 선택한다.

❶ 파일 탐색기에서 프로젝트 “LabProject11” 폴더의 다음 파일을 선택하여 프로젝트 “LabProject15” 폴더에 복사한다.

- GameFramework.h
- GameFramework.cpp
- Mesh.h
- Mesh.cpp
- Camera.h
- Camera.cpp
- Player.h
- Player.cpp
- Object.h
- Object.cpp
- Scene.h

- Scene.cpp
- Shader.h
- Shader.cpp
- stdafx.h
- Timer.h
- Timer.cpp
- Shaders.hlsl

❷ 위에서 복사한 파일을 Visual Studio 솔루션 탐색기에서 프로젝트 “LabProject15”에 추가한다. 오른쪽 마우스 버튼으로 『LabProject15』를 선택하고 『추가』, 『기존 항목』를 차례로 선택한다. 그러면 “기존 항목 추가” 대화 상자가 나타난다. 위의 파일들을 마우스로 선택(Ctrl+선택)하여 『추가』를 누르면 선택된 파일들이 프로젝트 “LabProject15”에 추가된다.

② LabProject15.cpp 파일 수정하기

이제 “LabProject15.cpp” 파일의 내용을 “LabProject11.cpp” 파일의 내용으로 바꾸도록 하자. “LabProject11.cpp” 파일의 내용 전체를 “LabProject15.cpp” 파일로 복사한다. 이제 “LabProject11.cpp” 파일에서 “LabProject11”을 “LabProject15”로 모두 바꾼다. 그리고 “LABPROJECT11”을 “LABPROJECT15”로 모두 바꾼다.

③ “Mesh.h” 파일 수정하기

“Mesh.h” 파일을 다음과 같이 수정한다.

❶ “CMesh” 클래스에 다음 멤버 변수를 선언한다.

```
protected:
//모델 좌표계의 OOB 바운딩 박스이다.
    BoundingBox m_xmBoundingBox;
```

❷ “CMesh” 클래스에 다음 멤버 함수를 선언한다.

```
BoundingBox GetBoundingBox() { return(m_xmBoundingBox); }
```

④ “Mesh.cpp” 파일 수정하기

“Mesh.cpp” 파일을 다음과 같이 수정한다.

❶ “CCubeMeshDiffused” 클래스의 생성자 마지막 부분에 다음을 추가한다.

//메쉬의 바운딩 박스(모델 좌표계)를 생성한다.

```
m_xmBoundingBox = BoundingBox(XMFLOAT3(0.0f, 0.0f, 0.0f), XMFLOAT3(fx, fy, fz), XMFLOAT4(0.0f, 0.0f, 0.0f, 1.0f));
```

② “CAirplaneMeshDiffused” 클래스의 생성자 마지막 부분에 다음을 추가한다.

//메쉬의 바운딩 박스(모델 좌표계)를 생성한다.

```
m_xmBoundingBox = BoundingBox(XMFLOAT3(0.0f, 0.0f, 0.0f), XMFLOAT3(fx, fy, fz), XMFLOAT4(0.0f, 0.0f, 0.0f, 1.0f));
```

⑤ “Camera.h” 파일 수정하기

“Camera.h” 파일을 다음과 같이 수정한다.

❶ “CCamera” 클래스에 다음 멤버 변수를 선언한다.

//절두체(월드 좌표계)

protected:

```
BoundingBox m_xmFrustum;
```

❷ “CCamera” 클래스에 다음 멤버 함수를 선언한다.

public:

//절두체(월드 좌표계)를 생성한다.

```
void GenerateFrustum();
```

//바운딩 박스(OOBB, 월드 좌표계)가 절두체에 포함되는 가를 검사한다.

```
bool IsInFrustum(BoundingBox& xmBoundingBox);
```

⑥ “Camera.cpp” 파일 수정하기

“Camera.cpp” 파일을 다음과 같이 수정한다.

❶ “CCamera” 클래스에 GenerateFrustum() 함수를 다음과 같이 정의한다.

```
void CCamera::GenerateFrustum()
```

```
{
```

//원근 투영 변환 행렬에서 절두체를 생성한다(절두체는 카메라 좌표계로 표현된다).

```
m_xmFrustum.CreateFromMatrix(m_xmFrustum, XMLoadFloat4x4(&m_xmf4x4Projection));
```

//카메라 변환 행렬의 역행렬을 구한다.

```
XMATRIX xmmtxInversView = XMMatrixInverse(NULL, XMLoadFloat4x4(&m_xmf4x4View));
```

//절두체를 카메라 변환 행렬의 역행렬로 변환한다(이제 절두체는 월드 좌표계로 표현된다).

```
m_xmFrustum.Transform(m_xmFrustum, xmmtxInversView);
```

```
}
```

❷ “CCamera” 클래스의 RegenerateViewMatrix() 함수 마지막에 다음을 추가한다.

```

void CCamera::RegenerateViewMatrix()
{
    ...
    //카메라 변환 행렬이 바뀔 때마다 절두체를 다시 생성한다(절두체는 월드 좌표계로 생성한다).
    GenerateFrustum();
}

```

③ “CCamera” 클래스에 IsInFrustum() 함수를 다음과 같이 정의한다.

```

bool CCamera::IsInFrustum(BoundingOrientedBox& xmBoundingBox)
{
    return(m_xmFrustum.Intersects(xmBoundingBox));
}

```

⑦ “GameObject.h” 파일 변경하기

❶ “CGameObject” 클래스에 다음 멤버 함수를 추가한다.

```

public:
    //게임 객체가 카메라에 보이는 가를 검사한다.
    bool IsVisible(CCamera *pCamera=NULL);

```

⑧ “GameObject.cpp” 파일 변경하기

❶ “CGameObject” 클래스의 생성자와 소멸자를 다음과 같이 수정한다.

```

bool CGameObject::IsVisible(CCamera *pCamera)
{
    OnPrepareRender();

    bool bIsVisible = false;
    BoundingOrientedBox xmBoundingBox = m_pMesh->GetBoundingBox();
    //모델 좌표계의 바운딩 박스를 월드 좌표계로 변환한다.
    xmBoundingBox.Transform(xmBoundingBox, XMLoadFloat4x4(&m_xmf4x4World));
    if (pCamera) bIsVisible = pCamera->IsInFrustum(xmBoundingBox);
    return(bIsVisible);
}

```

❷ “CGameObject” 클래스의 Render() 함수를 다음과 같이 수정한다.

```

void CGameObject::Render(ID3D12GraphicsCommandList *pd3dCommandList, CCamera *pCamera)
{
    //게임 객체가 카메라에 보이면 렌더링한다.
    if (IsVisible(pCamera))
    {
        UpdateShaderVariables(pd3dCommandList);

        if (m_pShader) m_pShader->Render(pd3dCommandList, pCamera);
    }
}

```

```
        if (m_pMesh) m_pMesh->Render(pd3dCommandList);  
    }  
}
```