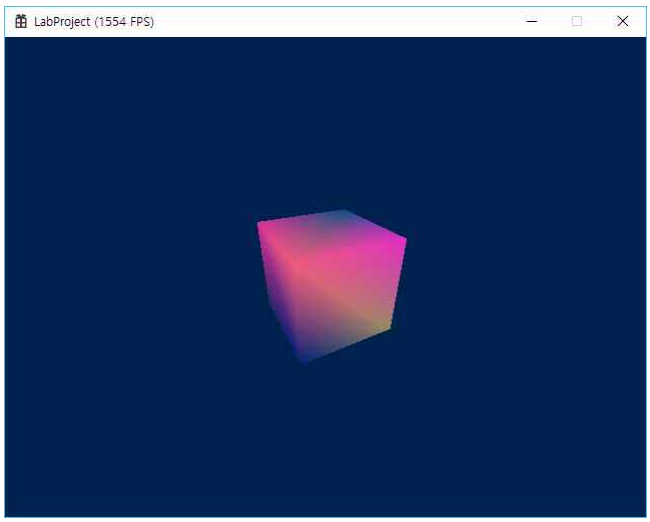
□ 예제 프로그램 10: LabProject09(회전하는 직육면체 그리기)

예제 프로그램 LabProject08을 기반으로 하여 회전하는 직육면체를 그려보도록 하자. 그러나, LabProject08과 다른 점은 직육면체를 표현하기 위하여 인덱스 버퍼를 사용하는 것이다. 그리고 래스터라이져 상태를 변경하여 프리미티브를 선분(와이어프레임: Wireframe)으로 그리도록 한다.



① 새로운 프로젝트의 생성

먼저 새로운 프로젝트 LabProject09를 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 “LabProject09”를 입력하고 『확인』을 선택한다.

❶ 파일 탐색기에서 프로젝트 “LabProject08” 폴더의 다음 파일을 선택하여 프로젝트 “LabProject09” 폴더에 복사한다.

- GameFramework.h
- GameFramework.cpp
- Mesh.h
- Mesh.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 “LabProject09”에 추가한다. 오른쪽 마우스 버튼으로 『LabProject09』을 선택하고 『추가』, 『기존 항목』를 차례로 선택한다. 그러면 “기존 항목 추가” 대화 상자가 나타난다. 다음 파일들을 마우스로 선택(Ctrl+선택)하여 『추가』를 누르면 선택된 파일들이 프로젝트 “LabProject09”에 추가된다.

② LabProject09.cpp 파일 수정하기

이제 “LabProject09.cpp" 파일의 내용을 “LabProject08.cpp" 파일의 내용으로 바꾸도록 하자. “LabProject08.cpp" 파일의 내용 전체를 “LabProject09.cpp" 파일로 복사한다. 이제 “LabProject09.cpp" 파일에서 “LabProject08”을 “LabProject09”로 모두 바꾼다.
그리고 “LABPROJECT08”을 “LABPROJECT09”로 모두 바꾼다.

③ “Mesh.h” 파일 수정하기

“CMesh” 클래스에 인덱스 버퍼를 위한 내용을 다음을 추가한다.

```
protected:
   ID3D12Resource *m_pd3dIndexBuffer = NULL;
   ID3D12Resource *m_pd3dIndexUploadBuffer = NULL;
/*인덱스 버퍼(인덱스의 배열)와 인덱스 버퍼를 위한 업로드 버퍼에 대한 인터페이스 포인터이다. 인덱스 버퍼는 정점 버퍼(배열)에 대한 인덱스를 가진다.*/
   D3D12_INDEX_BUFFER_VIEW m_d3dIndexBufferView;

   UINT m_nIndices = 0;
//인덱스 버퍼에 포함되는 인덱스의 개수이다.
   UINT m_nStartIndex = 0;
//인덱스 버퍼에서 메쉬를 그리기 위해 사용되는 시작 인덱스이다.
   int m_nBaseVertex = 0;
//인덱스 버퍼의 인덱스에 더해질 인덱스이다.
```

④ “Mesh.cpp” 파일 수정하기

❶ “CMesh” 클래스의 소멸자에 다음을 추가한다.

```
   if (m_pd3dIndexBuffer) m_pd3dIndexBuffer->Release();
   if (m_pd3dIndexUploadBuffer) m_pd3dIndexUploadBuffer->Release();
```

❷ “CMesh” 클래스의 ReleaseUploadBuffers() 함수에 다음을 추가한다.

```
if (m_pd3dIndexUploadBuffer) m_pd3dIndexUploadBuffer->Release();
m_pd3dIndexUploadBuffer = NULL;
```

❸ “CMesh” 클래스의 Render() 함수를 다음과 같이 수정한다.

```
void CMesh::Render(ID3D12GraphicsCommandList *pd3dCommandList)
{
   pd3dCommandList->IASetPrimitiveTopology(m_d3dPrimitiveTopology);
   pd3dCommandList->IASetVertexBuffers(m_nSlot, 1, &m_d3dVertexBufferView);
   if (m_pd3dIndexBuffer)
   {
      pd3dCommandList->IASetIndexBuffer(&m_d3dIndexBufferView);
      pd3dCommandList->DrawIndexedInstanced(m_nIndices, 1, 0, 0, 0);
//인덱스 버퍼가 있으면 인덱스 버퍼를 파이프라인(IA: 입력 조립기)에 연결하고 인덱스를 사용하여 렌더링한다.
   }
   else
   {
      pd3dCommandList->DrawInstanced(m_nVertices, 1, m_nOffset, 0);
   }
}
```

❹ “CCubeMeshDiffused” 클래스의 생성자를 다음과 같이 변경한다.

```
CCubeMeshDiffused::CCubeMeshDiffused(ID3D12Device *pd3dDevice, ID3D12GraphicsCommandList
*pd3dCommandList, float fWidth, float fHeight, float fDepth) : CMesh(pd3dDevice,
pd3dCommandList)
{
//직육면체는 꼭지점(정점)이 8개이다.
   m_nVertices = 8;
   m_nStride = sizeof(CDiffusedVertex);
   m_d3dPrimitiveTopology = D3D_PRIMITIVE_TOPOLOGY_TRIANGLELIST;

   float fx = fWidth*0.5f, fy = fHeight*0.5f, fz = fDepth*0.5f;

//정점 버퍼는 직육면체의 꼭지점 8개에 대한 정점 데이터를 가진다.
   CDiffusedVertex pVertices[8];
   pVertices[0] = CDiffusedVertex(XMFLOAT3(-fx, +fy, -fz), RANDOM_COLOR);
   pVertices[1] = CDiffusedVertex(XMFLOAT3(+fx, +fy, -fz), RANDOM_COLOR);
   pVertices[2] = CDiffusedVertex(XMFLOAT3(+fx, +fy, +fz), RANDOM_COLOR);
   pVertices[3] = CDiffusedVertex(XMFLOAT3(-fx, +fy, +fz), RANDOM_COLOR);
   pVertices[4] = CDiffusedVertex(XMFLOAT3(-fx, -fy, -fz), RANDOM_COLOR);
   pVertices[5] = CDiffusedVertex(XMFLOAT3(+fx, -fy, -fz), RANDOM_COLOR);
   pVertices[6] = CDiffusedVertex(XMFLOAT3(+fx, -fy, +fz), RANDOM_COLOR);
   pVertices[7] = CDiffusedVertex(XMFLOAT3(-fx, -fy, +fz), RANDOM_COLOR);

   m_pd3dVertexBuffer = ::CreateBufferResource(pd3dDevice, pd3dCommandList, pVertices,
m_nStride * m_nVertices, D3D12_HEAP_TYPE_DEFAULT,
D3D12_RESOURCE_STATE_VERTEX_AND_CONSTANT_BUFFER, &m_pd3dVertexUploadBuffer);

   m_d3dVertexBufferView.BufferLocation = m_pd3dVertexBuffer->GetGPUVirtualAddress();
```

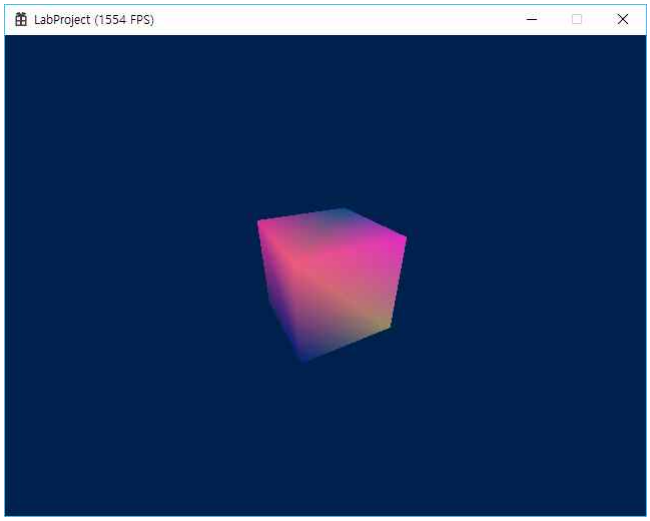
```
    m_d3dVertexBufferView.StrideInBytes = m_nStride;
    m_d3dVertexBufferView.SizeInBytes = m_nStride * m_nVertices;

/*인덱스 버퍼는 직육면체의 6개의 면(사각형)에 대한 기하 정보를 갖는다. 삼각형 리스트로 직육면체를 표현할 것이므로 각 면은 2개의 삼각형을 가지고 각 삼각형은 3개의 정점이 필요하다. 즉, 인덱스 버퍼는 전체 36(=6*2*3)개의 인덱스를 가져야 한다.*/
    m_nIndices = 36;

    UINT pnIndices[36];
//ⓐ 앞면(Front) 사각형의 위쪽 삼각형
    pnIndices[0] = 3; pnIndices[1] = 1; pnIndices[2] = 0;
//ⓑ 앞면(Front) 사각형의 아래쪽 삼각형
    pnIndices[3] = 2; pnIndices[4] = 1; pnIndices[5] = 3;
//ⓒ 윗면(Top) 사각형의 위쪽 삼각형
    pnIndices[6] = 0; pnIndices[7] = 5; pnIndices[8] = 4;
//ⓓ 윗면(Top) 사각형의 아래쪽 삼각형
    pnIndices[9] = 1; pnIndices[10] = 5; pnIndices[11] = 0;
//ⓔ 뒷면(Back) 사각형의 위쪽 삼각형
    pnIndices[12] = 3; pnIndices[13] = 4; pnIndices[14] = 7;
//ⓕ 뒷면(Back) 사각형의 아래쪽 삼각형
    pnIndices[15] = 0; pnIndices[16] = 4; pnIndices[17] = 3;
//ⓖ 아래면(Bottom) 사각형의 위쪽 삼각형
    pnIndices[18] = 1; pnIndices[19] = 6; pnIndices[20] = 5;
//ⓗ 아래면(Bottom) 사각형의 아래쪽 삼각형
    pnIndices[21] = 2; pnIndices[22] = 6; pnIndices[23] = 1;
//ⓘ 옆면(Left) 사각형의 위쪽 삼각형
    pnIndices[24] = 2; pnIndices[25] = 7; pnIndices[26] = 6;
//ⓙ 옆면(Left) 사각형의 아래쪽 삼각형
    pnIndices[27] = 3; pnIndices[28] = 7; pnIndices[29] = 2;
//ⓚ 옆면(Right) 사각형의 위쪽 삼각형
    pnIndices[30] = 6; pnIndices[31] = 4; pnIndices[32] = 5;
//ⓛ 옆면(Right) 사각형의 아래쪽 삼각형
    pnIndices[33] = 7; pnIndices[34] = 4; pnIndices[35] = 6;

//인덱스 버퍼를 생성한다.
    m_pd3dIndexBuffer = ::CreateBufferResource(pd3dDevice, pd3dCommandList, pnIndices,
sizeof(UINT) * m_nIndices, D3D12_HEAP_TYPE_DEFAULT, D3D12_RESOURCE_STATE_INDEX_BUFFER,
&m_pd3dIndexUploadBuffer);

//인덱스 버퍼 뷰를 생성한다.
    m_d3dIndexBufferView.BufferLocation = m_pd3dIndexBuffer->GetGPUVirtualAddress();
    m_d3dIndexBufferView.Format = DXGI_FORMAT_R32_UINT;
    m_d3dIndexBufferView.SizeInBytes = sizeof(UINT) * m_nIndices;
}
```

⑤ 프로젝트 빌드하여 실행하기

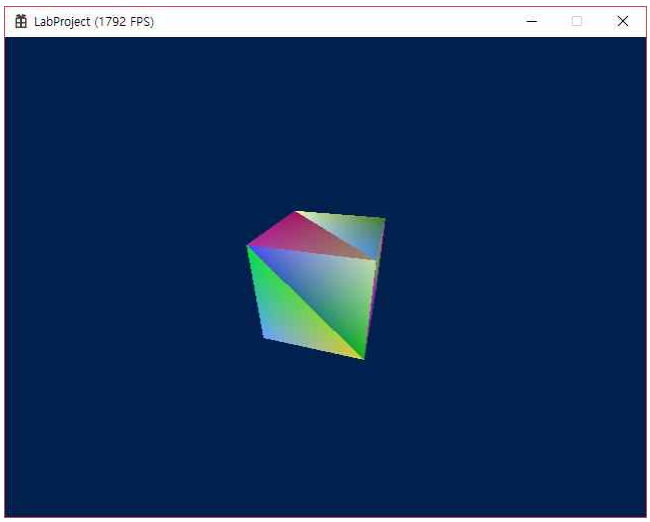
❶ 프로젝트를 빌드하여 실행

프로젝트를 빌드하여 실행하면 다음 그림과 같이 정점 버퍼만을 사용하여 렌더링한 것과 차이가 남을 알

수 있다.



인덱스 버퍼를 사용한 렌더링



정점 버퍼만을 사용한 렌더링

❷ 래스터라이저 상태를 변경하기

“CShader” 클래스의 CreateRasterizerState() 멤버 함수에서 래스터라이저 상태의 채우기 모드를 다음과 같이 변경하여 빌드하고 실행한다.

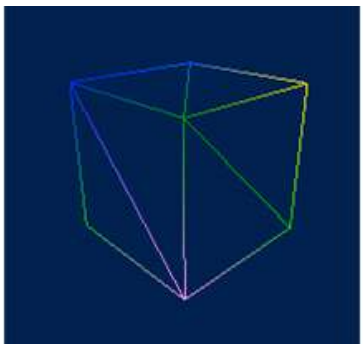
```
//D3D12_FILL_MODE_WIREFRAME은 프리미티브(삼각형)의 내부를 칠하지 않고 변(Edge)만 그린다.
   d3dRasterizerDesc.FillMode = D3D12_FILL_MODE_WIREFRAME;
```

래스터라이저 상태의 컬링 모드를 D3D12_CULL_MODE_NONE 또는 D3D12_CULL_MODE_FRONT로 변경하여 D3D12_CULL_MODE_BACK일 때의 결과와 비교해 본다.

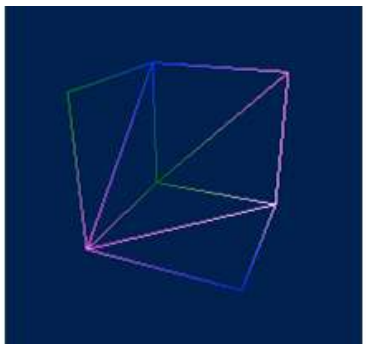
```
   d3dRasterizerDesc.CullMode = D3D12_CULL_MODE_NONE;
```

```
d3dRasterizerDesc.CullMode = D3D12_CULL_MODE_FRONT;
```

_CULL_MODE_NONE



_CULL_MODE_BACK



_CULL_MODE_FRONT

래스터라이저 컬링 모드