**D3D11\_TEXTURE2D\_DESC 쉐도우 맵 정의**

Width = 2048

Height = 2048;

MipLevels = 1;

ArraySize = 1;

Format = DXGI\_FORMAT\_R32G32B32A32\_FLOAT;

SampleDesc.Count = 1;

SampleDesc.Quality = 0;

Usage = D3D11\_USAGE\_DEFAULT;

BindFlags = D3D11\_BIND\_RENDER\_TARGET | D3D11\_BIND\_SHADER\_RESOURCE;

CPUAccessFlags = 0;

MiscFlags = 0;

**LightViewMatrix 생성**

LightPoisition = 빛 위치

Lookat = ( 0.0f, 0.0f, 0.0f );

Up = ( 0.0f, 1.0f, 0.0f );

D3DXMATRIX lightViewMatrix;

D3DXMatrixLookAtLH(&lightViewMatrix, LightPoisition , Lookat , Up );

만들어서 lightViewMatrix를 셰이더에 전달.

**LightProjectiveMatrix 생성**

fieldOfView = (float)D3DX\_PI / 2.0f;

screenAspect = 1.0f;

ScreenNear = 투영행렬 만들었을때 썼던 값

ScreenDepth = 투영행렬 만들었을때 썼던 값

D3DXMATRIX lightProjectionMatrix;

D3DXMatrixPerspectiveFovLH(&lightProjectionMatrix, fieldOfView, screenAspect, screenNear, screenDepth);

만들어서 lightViewMatrix를 셰이더에 전달.

**CreateShadow.hlsl**

--------------------정점셰이더----------------------------------------

cbuffer matrixBufer

{

float4x4 SUNNY\_ModelMatrix

float4x4 lightViewMatrix

Float4x4 lightProjectionMatrix

}

--------------------픽셀셰이더----------------------------------------

Texture2D shadowMaps : register(t0);

cbuffer PSSystemUniforms : register(b0)

{

Light SUNNY\_Light; -> light 구조체에서 빛 위치만 필요

};