



3D게임프로그래밍

-CHAPTER8-

SOULSEEK



목차

1. Camera



CAMERA

1. CAMERA

카메라 조작에 필요한 기능

- 이동과 회전이 주된 기능이다.
- 좌우, 상하, 전후라는 형태의 이동이 필요하다. – 월드 좌표가 아닌 카메라 좌표계의 축 이동
- 좌방, 우방, 상방, 하방을 바라보는 회전이 필요하다. – 월드 좌표가 아닌 카메라 좌표계의 축 회전

이동(Translation)

- 전진과 후진 : **vView** 벡터의 방향으로 카메라를 이동시키면 된다.

//후진은 **-dist**를 넣으면 된다.

```
D3DXMATRIXA16* Zcamera::MoveLocalZ(float dist)
{
    D3DXVECTOR3 vNewEye = m_vEye;
    D3DXVECTOR3 vNewDst = m_vLookat;

    D3DXVECTOR3 vMove;
    D3DVec3Normalize(&vMove, &m_vView);

    vMove *= dist;
    vNewEye += vMove;
    vNewDst += vMove;

    return SetView(&vNewEye, &vNewDst, &m_vUp);
}
```

이렇게 리턴된 값을 **D3DXMatrixLookAtLH()** 함수를 통해 적용하면 된다.

1. CAMERA

회전(Rotation)

- **vView** 벡터를 회전할 축의 회전 행렬과 곱한뒤 **vEye**값을 더한 위치가 새로운 **vLookAt**의 위치가 된다.
- 새로 변경된 **vLookAt** 지점을 이용해서 **vUp**, **vEye**값으로 카메라 행렬을 유도하면 된다.

// 카메라 좌표계의 Y축으로 **angle**만큼 회전한다.

```
D3DXMATRIXA16* ZCamera::RotateLocalY( float angle )
```

```
{
```

```
    D3DXMATRIXA16 matRot;
```

```
    D3DXMatrixRotationAxis( &matRot, &m_vUp, angle );
```

```
    D3DXVECTOR3 vNewDst;
```

```
    // view * rot로 새로운 dst vector를 구한다.
```

```
    D3DXVec3TransformCoord( &vNewDst, &m_vView, &matRot );
```

```
    // 실제 dst position = eye Position + dst vector
```

```
    vNewDst += m_vEye;
```

```
    return SetView( &m_vEye, &vNewDst, &m_vUp );
```

```
}
```

1. CAMERA

학습과제

- 전진 후진(A, Z)을 이용해 상,하,좌,우 이동을 만들어보자
- 마우스 컨트롤로 좌, 우 회전이 적용되어 있는 것을 이용해서 상, 하 회전을 만들어보자
- 특정키를 누르면 Z축을 기준으로 시계방향 반시계 방향 일정 각도만큼 만 회전하고 놓으면 원래 각도로 돌아오는 상황을 구현해 보자.