

エディタのカメラの位置 について

GC3-14、椎名太陽

1 : 設定するデータ

```
//=====
// 視点座標を計算
//=====
void CCamera::CalcPosV()
{
    // 視点は対象物の後方に
    m_PosV =
    {
        m_Pos.x - (sinf(m_Rot.y) * cosf(m_Rot.x) * (m_fDistance)),
        m_Pos.y - (sinf(m_Rot.x) * (m_fDistance)),
        m_Pos.z - (cosf(m_Rot.y) * cosf(m_Rot.x) * (m_fDistance))
    };
}

//=====
// 注視点座標を計算
//=====
void CCamera::CalcPosR()
{
    // 注視点を対象物の前方に
    m_PosR =
    {
        m_Pos.x + (sinf(m_Rot.y) * cosf(m_Rot.x) * m_fDistance),
        m_Pos.y + (sinf(m_Rot.x) * m_fDistance),
        m_Pos.z + (cosf(m_Rot.y) * cosf(m_Rot.x) * m_fDistance)
    };
}
```

- Pos = (0.0f,0.0f,0.0f);
- Rot = (0.0f,0.0f,0.0f);
- m_fDistance = 400.0f;

2 : ゲームのカメラ

```
// =====  
// 視点座標を計算  
// =====  
void CCamera::CalcPosV()  
{  
    // 視点は対象物の後方に  
    m_PosV =  
    {  
        m_Pos.x - (sinf(m_Rot.y) * cosf(m_Rot.x) * (m_fDistance)),  
        m_Pos.y - (sinf(m_Rot.x) * (m_fDistance)),  
        m_Pos.z - (cosf(m_Rot.y) * cosf(m_Rot.x) * (m_fDistance))  
    };  
}  
  
// =====  
// 注視点座標を計算  
// =====  
void CCamera::CalcPosR()  
{  
    // 注視点を対象物の前方に  
    m_PosR =  
    {  
        m_Pos.x + (sinf(m_Rot.y) * cosf(m_Rot.x) * m_fDistance),  
        m_Pos.y + (sinf(m_Rot.x) * m_fDistance),  
        m_Pos.z + (cosf(m_Rot.y) * cosf(m_Rot.x) * m_fDistance)  
    };  
}
```

- 決めた位置から向きを参照し、向いている前方向に注視点(PosR)、後ろ方向に視点(PosV)です。

3 : 計算式の説明

```
// 視点座標を計算
// =====
void CCamera::CalcPosV()
{
    // 視点は対象物の後方に
    m_PosV =
    {
        m_Pos.x - (sinf(m_Rot.y) * cosf(m_Rot.x) * (m_fDistance)),
        m_Pos.y - (sinf(m_Rot.x) * (m_fDistance)),
        m_Pos.z - (cosf(m_Rot.y) * cosf(m_Rot.x) * (m_fDistance))
    };
}

// 注視点座標を計算
// =====
void CCamera::CalcPosR()
{
    // 注視点を対象物の前方に
    m_PosR =
    {
        m_Pos.x + (sinf(m_Rot.y) * cosf(m_Rot.x) * m_fDistance),
        m_Pos.y + (sinf(m_Rot.x) * m_fDistance),
        m_Pos.z + (cosf(m_Rot.y) * cosf(m_Rot.x) * m_fDistance)
    };
}
```

- Cosは(Yaw0.0f)の時、1 になります。Zがcosなのは前を正としている。
- Sinは(Yaw0.0f)の時、0 になる。向きが0なら完全に前を向きたいのでXはsinにする。
- Yはsinなので、Pitchが0の時前を向く（上や下を向かない。）90°の時1や-1（上や下）を向く。
- XやZにcos(Pitch)を掛けているのは、上や下を向くほどXやZの長さを短くする（カメラを球状に操作するため）

4：補足

- もしカメラを動かさないなら、向きとか関係なく、
m_fDistance = 400.0f;
Pos = (0.0f,0.0f,0.0f);
PosR = (0.0f,0.0f,m_fDistance);
PosV = (0.0f,0.0f,-m_fDistance);

で良いよ！