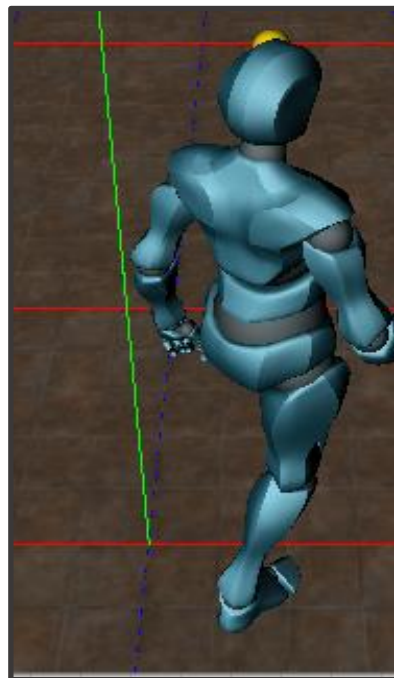


キャラクターの回転動作

3DViewerの時のキャラクターの回転は移動方向に向かって、
回転動作が少ない回り方(時計回りか、反時計回り)で、
『少し』回転していました。

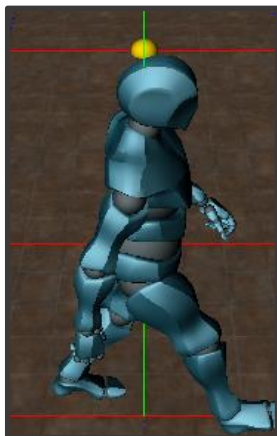


Dキーを押した時



アクションゲームになると、
進みたい方向や、攻撃方向などを細かく操作したい場合、
この回転方法だと、軸がなかなか定まらず、ユーザにストレスを
与えてしまう場合があります。

そこで、例えばDキーを一瞬でも押したならば、



完全に右側に振り向くようにしていきたいと思います。

もちろん、瞬間的に振り向くのではなくて、
ゆっくりと、振り向かせます。
(キーを押していない時も、回転動作が続くように)

【やり方】

まず、振り向かせたい回転(ゴールとなる回転)を求めます。

```
mGoalQuaRot =  
    Quaternion::AngleAxis(cameraRot.y + rotRad, AsoUtility::AXIS_Y);
```

rotRad ... 移動方向への加算回転。
 Dキーだとしたら、右方向を向きたいので、
 90度(deg)。

mGoalQuaRot ... クォータニオン

あとは、

```
mStepRotTime -= mSceneManager->GetDeltaTime();  
mPlayerRotY = Quaternion::Slerp(  
    mPlayerRotY, mGoalQuaRot, (TIME_ROT - mStepRotTime) / TIME_ROT);
```

現在のY回転がゴールとなる回転に、少しずつ近づくように、
Lerp、もしくはSlerpを使って補間していきます。

こうして求められた、Y軸回転をキャラクターの回転情報に
合成していけば良いのですが、
このゲームは、基準を常に重力方向に持つ必要がありますので、
カメラと同じように、
重力制御による正面を取得して、そこからY軸回転させます。

// 重力方向に沿って回転させる

```
mTransform.quaRot = mGravityManager->GetTransform()->quaRot;  
mTransform.quaRot = mTransform.quaRot.Mult(mPlayerRotY);
```



これでバッチリ！