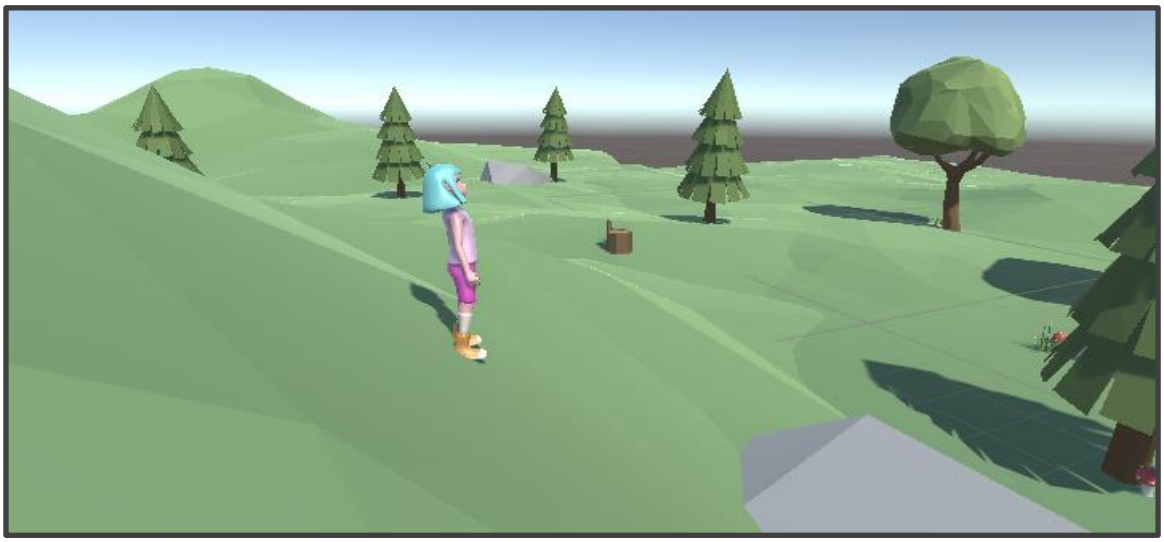
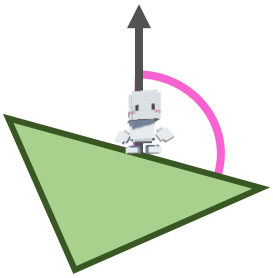
# 傾斜のベクトルと角度

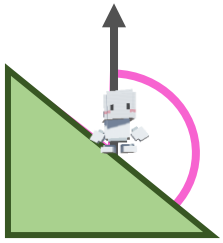
傾斜のベクトルを把握することができれば、
傾斜の角度によって、坂道を滑り落ちる表現であったり、
下り坂だったら移動速度を上げ、坂道だったら上げたりすることができます。



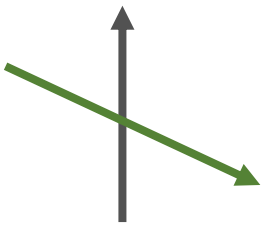
**傾斜の角度とは(ピンクの線)**



角度が小さい
= 緩やかな坂



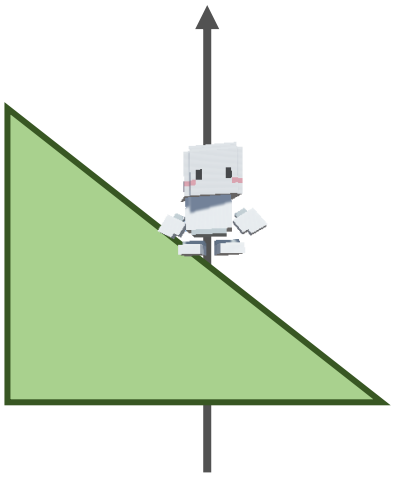
角度が大きい
= 急な坂



２つのベクトルから、
角度を求める式があるので、
重力の反対ベクトルと、
下り坂ベクトルがあれば良い。

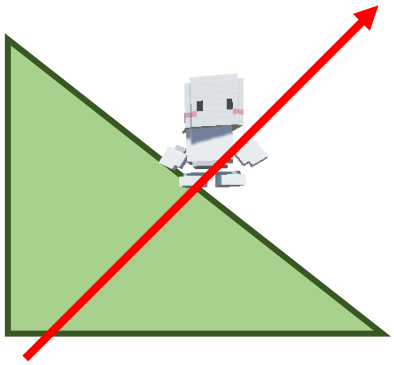
傾斜の角度を求めるにあたり、確実に取得できる情報を整理します。

重力の反対方向

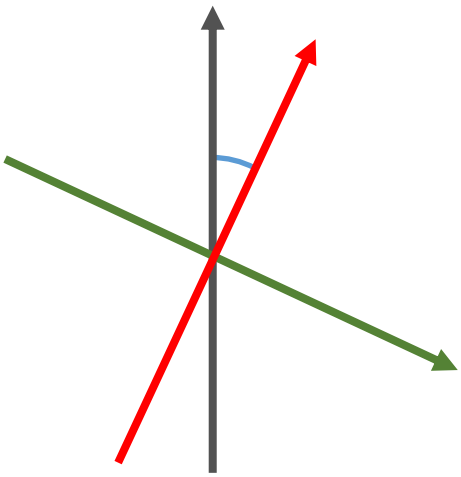


重力ベクトルを
反転して求めます。

地面の法線

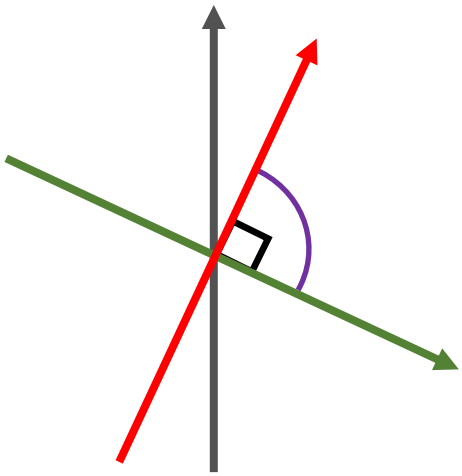


衝突している
地面ポリゴンの法線を取得します。



重力の反対ベクトルと、
法線ベクトル、２つのベクトルを
使用することによって、
クォータニオン計算で、
回転量と回転軸を求めることができます。

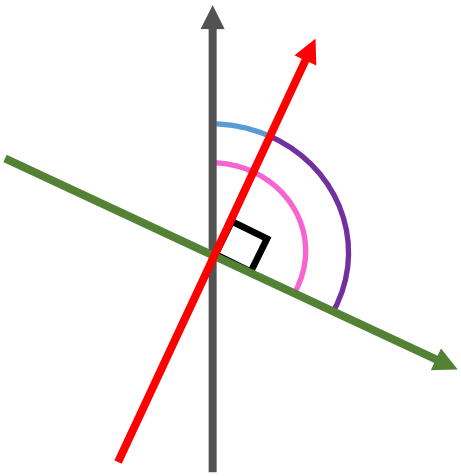
```
VECTOR gravityUp = grvMng_.GetDirUpGravity();

// 重力の反対方向から地面の法線方向に向けた回転量を取得
Quaternion up2GNorQua = Quaternion::FromToRotation(gravityUp, hitNormal_);

// 取得した回転の軸と角度を取得する
float angle = 0.0f;
float* anglePtr = &angle;
VECTOR axis;
up2GNorQua.ToAngleAxis(anglePtr, &axis);
```



ポリゴンの法線は、
ポリゴンの面に対して、垂直です。

今はベクトルが不明ですが、
緑の傾斜ベクトルと
赤の法線ベクトルの角度差は、、

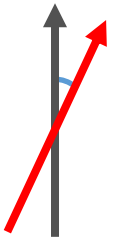
垂直ですので、当然、90度です(紫の角度)。



重力の反対ベクトルと、法線ベクトル
から求めた角度に90度を足した角度分、
重力の反対ベクトルから、法線ベクトルへの
回転軸で回転させると、
傾斜ベクトルを取得することができます。

あとは、重力の反対ベクトルと
傾斜ベクトルを使用して、
角度差を求めると、傾斜角を求めることが
できます。

```
void Player::CalcSlope(void)
{

    VECTOR gravityUp = grvMng_.GetDirUpGravity();

    // 重力の反対方向から地面の法線方向に向けた回転量を取得
    Quaternion up2GNorQua = Quaternion::FromToRotation(gravityUp, hitNormal_);

    // 取得した回転の軸と角度を取得する
    float angle = 0.0f;
    float* anglePtr = &angle;
    VECTOR axis;
    up2GNorQua.ToAngleAxis(anglePtr, &axis);
```



```
    // 90度足して、傾斜ベクトルへの回転を取得する
    Quaternion slopeQ = Quaternion::AngleAxis(
        angle + AsoUtility::Deg2RadD(90.0), axis);

    // 地面の傾斜線(黄色)
    slopeDir_ = slopeQ.PosAxis(gravityUp);

    // 傾斜の角度
    slopeAngleDeg_ = static_cast<float>(
        AsoUtility::AngleDeg(gravityUp, slopeDir_));


    // 傾斜による移動
    if (AsoUtility::SqrMagnitude(jumpPow_) == 0.0f)
    {
        float CHECK_ANGLE = 120.0f;
        if (slopeAngleDeg_ >= CHECK_ANGLE)
        {
            float diff = abs(slopeAngleDeg_ - CHECK_ANGLE);
            slopePow_ = VScale(slopeDir_, diff / 3.0f);
            movePow_ = VAdd(movePow_, slopePow_);
        }
    }

}
```