

# 移動処理

操作キャラクターをキーボードやゲームパッドで操作し、  
移動を行う処理を追加していきます。

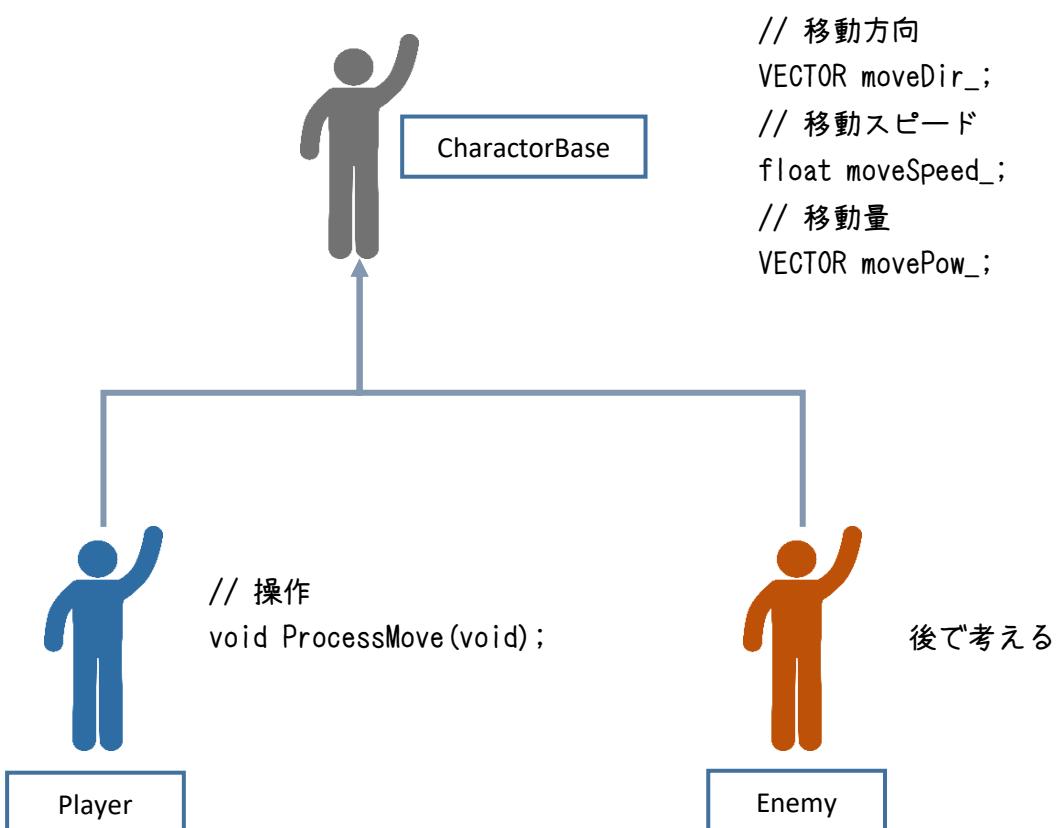
移動するためには、  
移動方向、移動スピード、移動量が必要になります。

通常であれば、Playerクラスにメンバ変数を追加していきますが、  
今回の設計の場合、Playerクラスに何かの実装を行う際に、  
CharacterBaseやEnemyのことも視野に入れて、考えていく必要があります。

Playerに実装する機能が、Enemyにも必要かどうか考え、  
必要だったら、共通機能になりますので、CharacterBaseに実装します。

移動に必要な上記3つのパラメータは、  
移動する際に必要な情報なので、エネミーにも必要。

一方、キーボードやゲームパッドによる操作は、  
エネミーには必要ありませんので、Player独自の移動処理となります。



## Player.h

～ 省略 ～

private:

～ 省略 ～

// 移動速度(通常)

static constexpr float SPEED\_MOVE = 5.0f;

// 移動速度(ダッシュ)

static constexpr float SPEED\_DASH = 10.0f;

// 操作

void ProcessMove(void);

## Player.cpp

void Player::Update(void)

{

// 移動操作

ProcessMove();

// 移動処理

transform\_.pos = VAdd(transform\_.pos, movePow\_);

// 3Dの基礎情報更新

transform\_.Update();

// アニメーション更新

animationController\_->Update();

}

## 【要件①】

3DWorldを参考に、キャラクターの移動処理をProcessMove関数内に実装すること。

キーボード操作は、WASD。

ゲームパッドは、左アナログスティック。  
両方に対応すること。

## 【目標①】

現時点では、カメラは移動しなくてよい。



## 【要件②】

移動時は、RUNアニメーションを行い、  
停止時は、IDLEアニメーションを行うこと。

## 【目標②】



### 【要件③】

ダッシュボタンを押下しながら移動を行うと、  
FAST\_RUNアニメーションを行い、移動速度も速くなるようにすること。

ダッシュボタンは、  
ゲームパッドはRT(右トリガー)、キーボードは右Shift。



### 【目標③】

