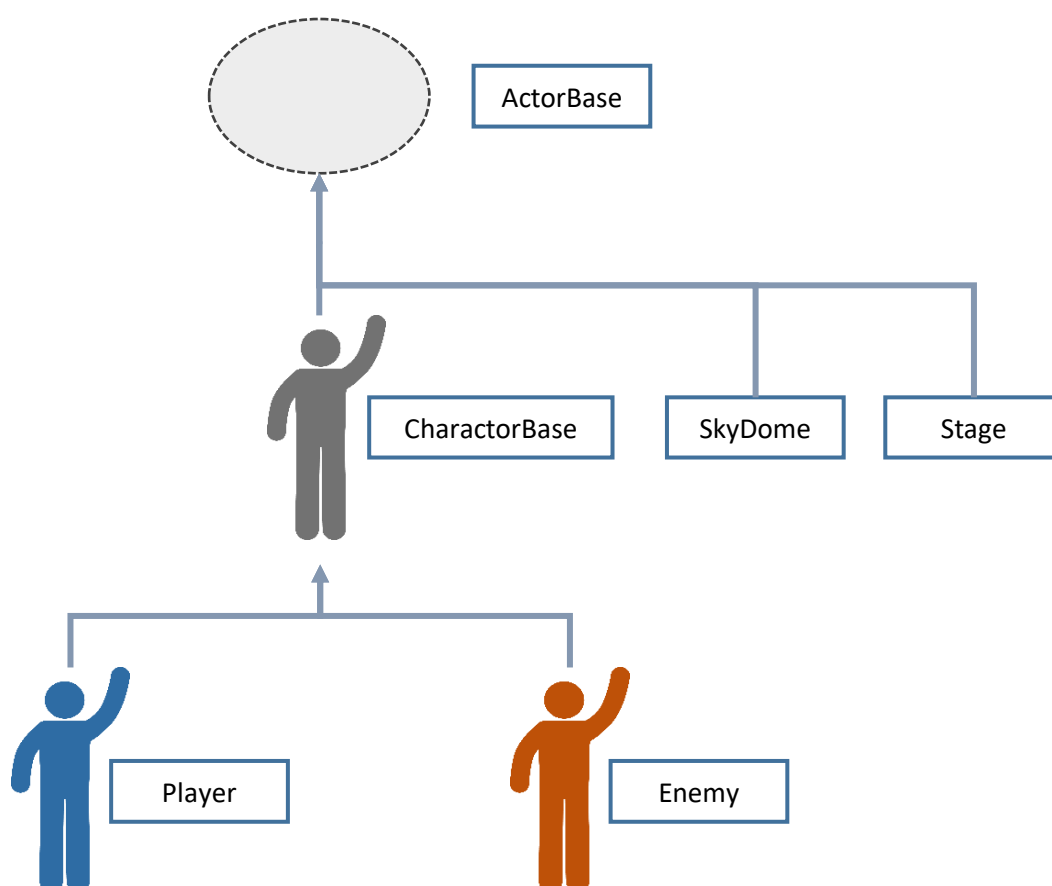


# キャラクター設計

ゲームシーンに操作キャラクターを描画させるため、Playerクラスを作成していきたいと思いますが、行く行くはエネミーも作成していきたいと考えています。

キャラクターも、エネミーも、ステージモデルと衝突して押し戻したり、重力がかかったり、PlayerとEnemyは共通する部分も多いため、ActorBaseとの間に、1つ基底クラスを用意していきます。  
( 3DWorldの教材、継承の復習も合わせて確認しましょう )



設計は、手戻りが無いよう、慎重に行う必要がありますが、学生時代は、あまり考えすぎず、とにかく作ることを優先した方が良いです。完璧な設計というのは世の中には存在せず、実装を進めていくとどうしても、調整が必要になります。

考えすぎて、手が止まらないように、ある程度の大枠を用意しておいて、実装を進めていきながら機能を拡張していきましょう。

ひとまず、超シンプルに。  
無くてよくない？と思われるかもしれませんが、  
最初は、そのくらいの温度感で良いと思います。

```
CharactorBase.h
```

```
#pragma once
#include "../ActorBase.h"

class CharactorBase : public ActorBase
{

public:

    // コンストラクタ
    CharactorBase(void);

    // デストラクタ
    virtual ~CharactorBase(void) override;

};
```

```
CharactorBase.cpp
```

```
#include "CharactorBase.h"

CharactorBase::CharactorBase(void)
:
    ActorBase()
{
}

CharactorBase::~~CharactorBase(void)
{
}
```

### 【要件①】

CharactorBaseを継承した、Playerクラスを作成し、  
ゲームシーン上で、モデルを描画すること。

リソース名	: PLAYER
リソース種別	: MODEL
スケール	: 1.0
ローカル回転	: Y軸180度
座標	: { 0.0f, 0.0f, 0.0f }

### 【目標①】

Tポーズのキャラクターが、Zの正方向を向いて、  
画面中央に描画されること。



## 【要件②】

設計に沿って、キャラクターをアニメーションさせること。

IDLE	:	"Idle.mvl"
RUN	:	"Run.mvl"
FAST_RUN	:	"FastRun.mvl"

## 【目標②】

初期アニメーションで、それぞれのアニメーションを指定し、  
3種のアニメーションが正常に再生されることが確認できること。

下図はIDLEアニメーション。

