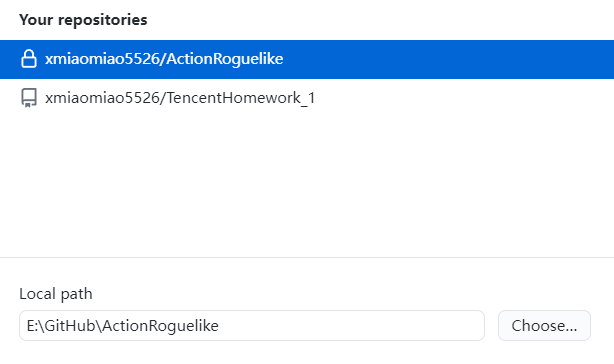
**ActionRoguelikeGame笔记**

# 新建项目与GitHub

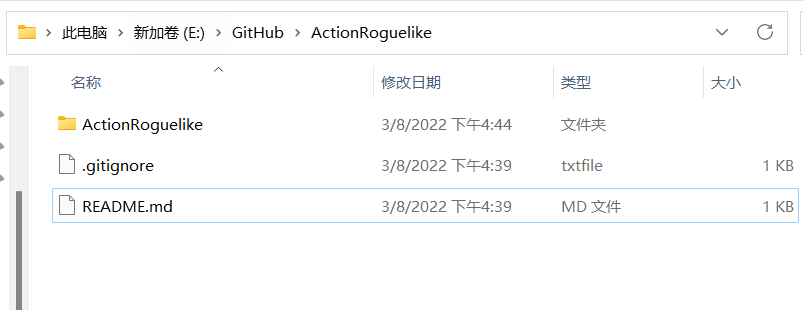
## 新建Github repository

使用Github Desktop，点击File->Clone reposit->建好的仓库->clone。



## 新建UE4项目

在Github的文件夹下新建UE4同名项目，此时文件夹内容：



将该文件夹下同名文件夹中的内容移到外面后，右键.uproject文件重新生成VS文件。

# 人物创建

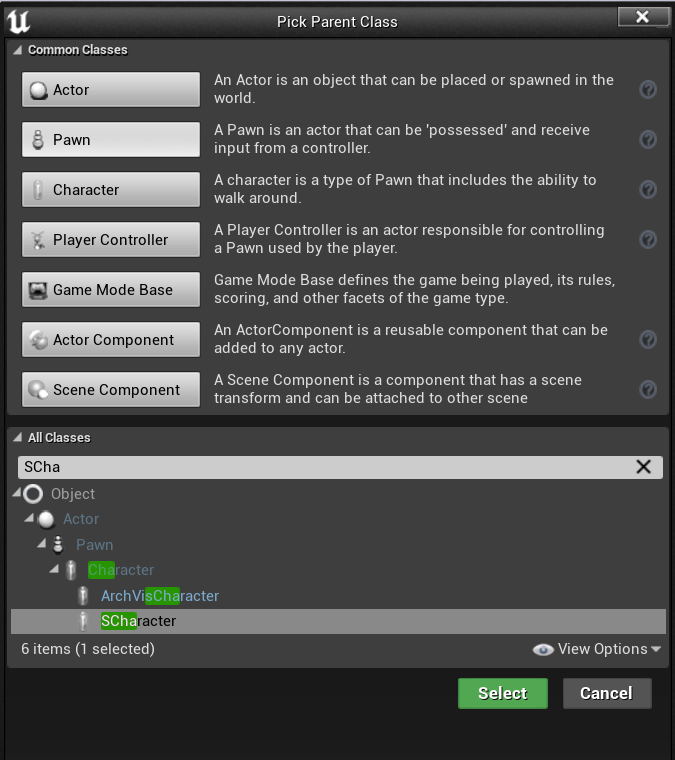
## 新建C++ Character类，名SCharacter

GENERATED\_BODY()宏应保证在类中第一行

CTRL+F5编译并唤醒UE编辑器（推荐使用，而不是热重载）

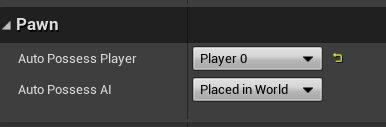
在Content文件夹下新建文件夹ActionRoguelike

## 创建蓝图类继承自SCharacter，命名为



拖拽到场景中，删掉场景中默认的Player Start

在Details面板中设置Auto Possess Player为Player 0

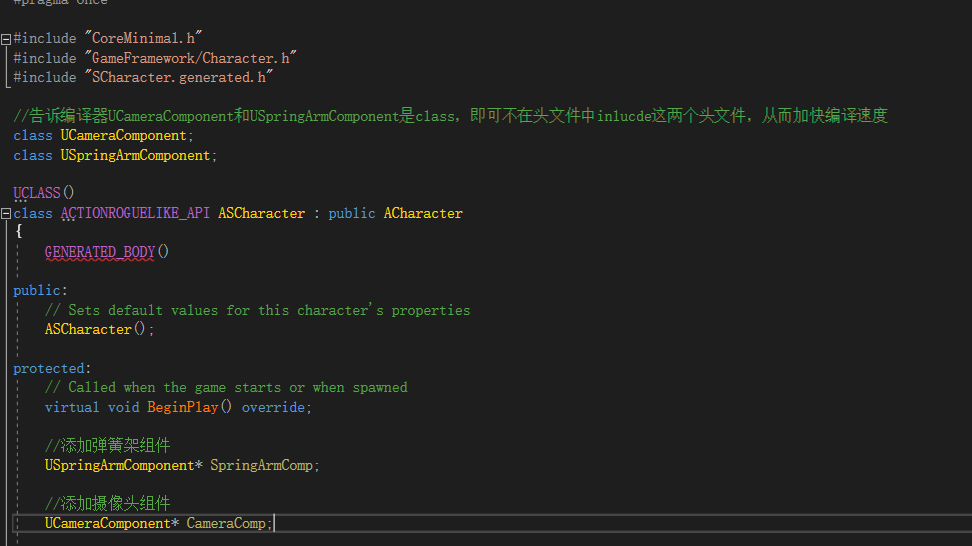


在Content->ActionRoguelike下新建文件夹map

Save all，将level保存到map文件夹下，命名为TestLevel

2 添加摄像头组件

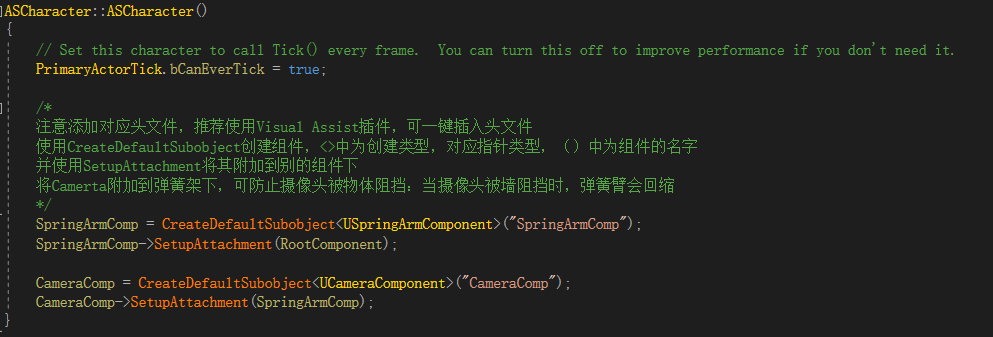
SCharacter.h



也可以把class在类名前面，不用在开头声明，并添加宏：



目前只是空指针，在.cpp文件中添加代码：



将如图所示三个模型文件夹放在Content下：

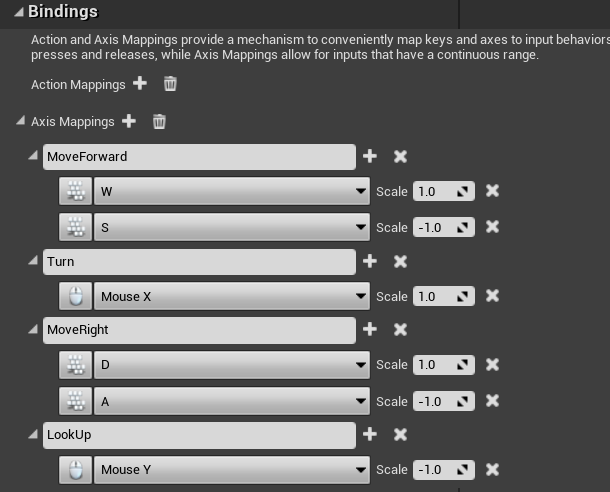


在人物的蓝图中设置Skeletal Mesh，并调整合适的高度与旋转

在人物的蓝图（Mesh）中设置动画

## 绑定运动输入

在编辑器中绑定轴Edit->Project Setting->左边一列input



添加C++代码

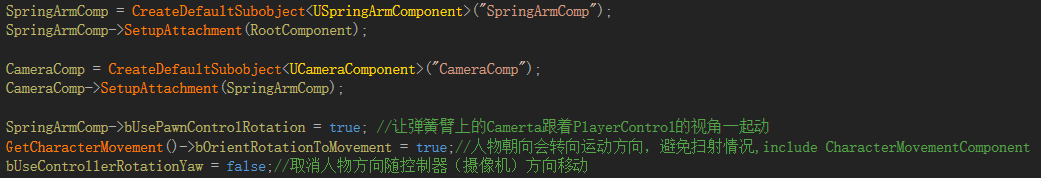
添加绑定



前后、左右运动函数



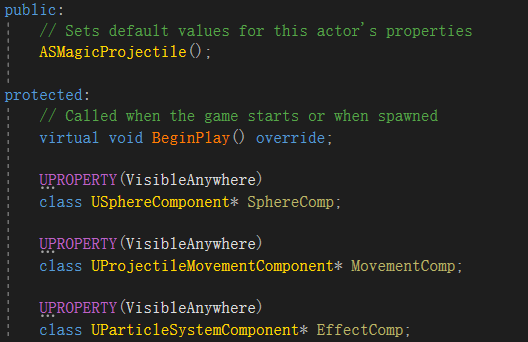
完善角色运动，在构造函数中修改弹簧臂和运动组件相关选项。以下布尔量可以在蓝图中找到并修改。

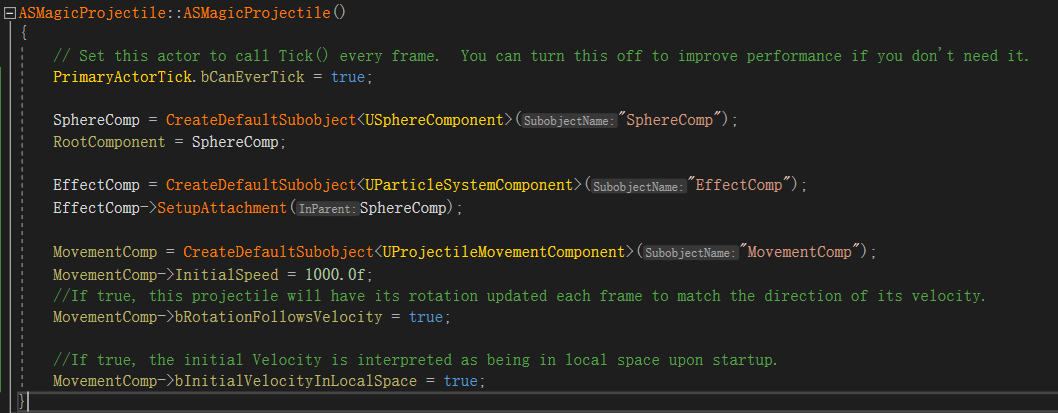


# 魔法攻击

## 创建类

创建如图所示AActor的子类，命名为SMagicProjectile，作为魔法攻击的发射物





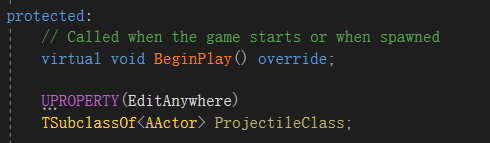
创建对应蓝图类，选择一个酷炫的粒子特效。

## 生成投射物（Spawn Actor）

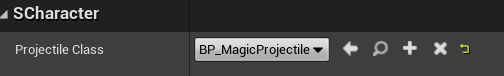
在SCharacter中绑定按键，在编辑器中将PrimaryAttack绑定到鼠标左键



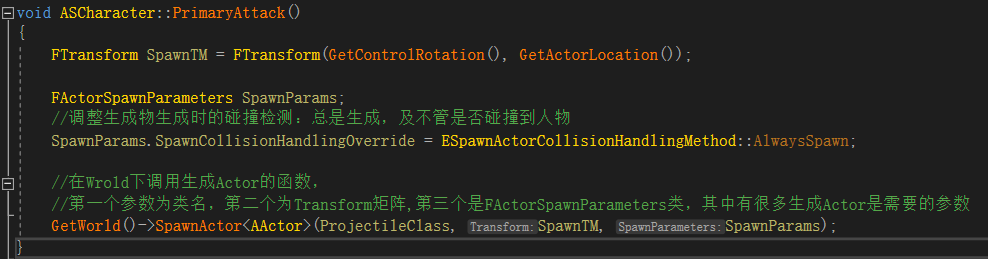
头文件中添加TSubClass



并在蓝图中指定ProjectileClass为魔法攻击粒子的蓝图类：

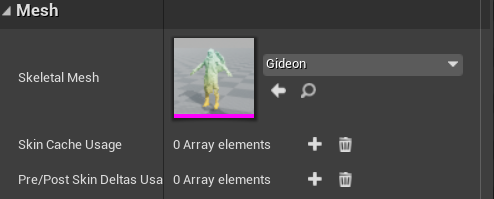


按键绑定触发的函数（绑定Action的函数不需要输入参数）：

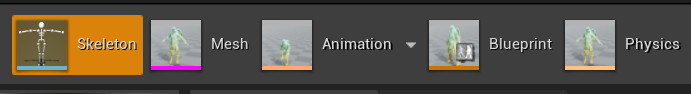


至此人物已经可以发射魔法飞弹了，但发射位置有点奇怪，我们希望从人物手的位置反射飞弹。

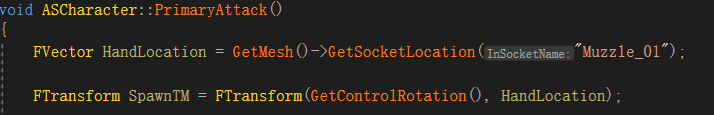
双击skeletal mesh中的图标可以看见骨骼的详情

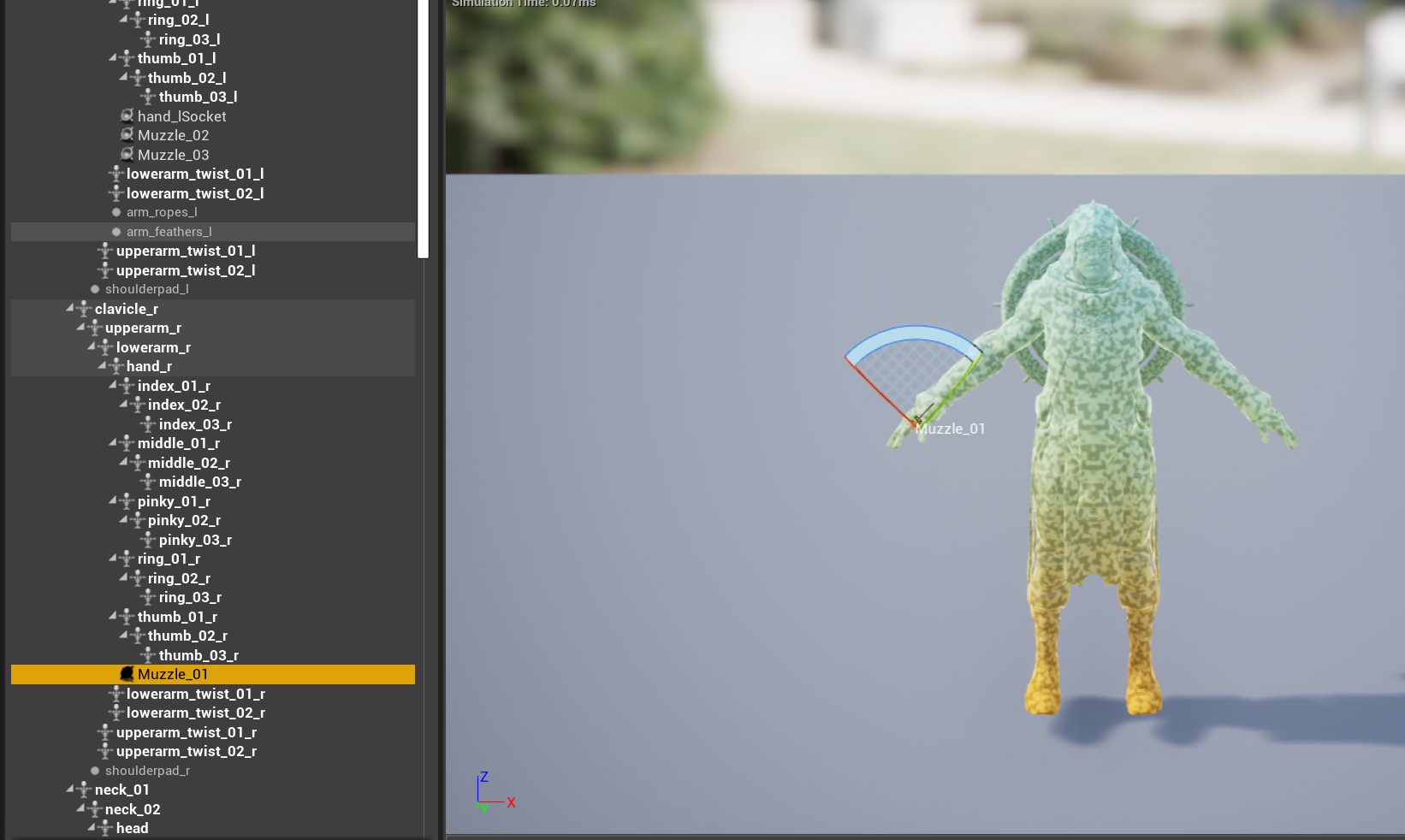


点击右上角Skeleton



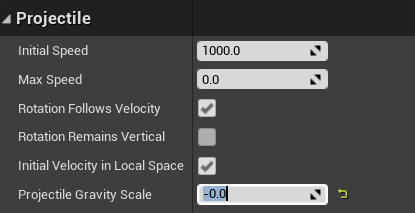
能够在左边一栏看见一堆东西，其中亮白色的是普通骨骼，而暗色的是插槽（Sockets），在UE编辑器中右键骨骼可以add Socket。找到Muzzle\_01，我们进入C++，指定飞弹从这个位置发射。修改部分代码：





## 碰撞设置

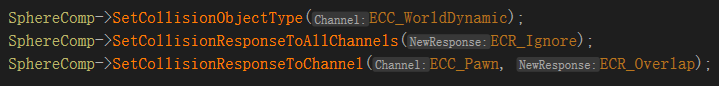
为方便显示，把投射物的重力设为0



摆放一个立方体在场景中，在Detail面板Collision栏中，设置Collision Presets为Custom可以看见碰撞检测相关属性，其全为block（左图）。同理，在投射物蓝图的SphereComp中可以看见全为Overlap（右图）。

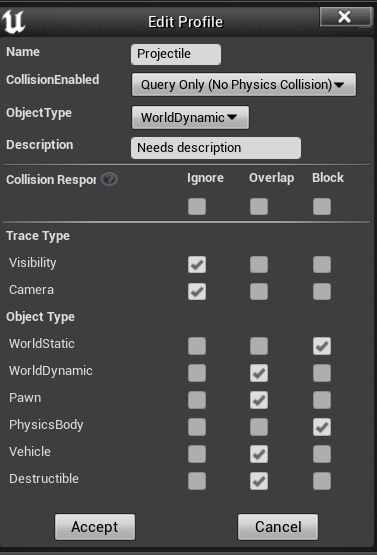
|  |  |
| --- | --- |
|  |  |

当两个物体在游戏场景中相撞时，会触发两个物体勾选的中反应最小的，即按照Ignore、Overlap、Block的顺序检测碰撞反应，例如：若俩物体其中一个为Ignore，则两物碰撞不会有任何反应。因此，投射物会透过Cube，若定义了Overlap事件，会触发该事件。将投射物的WorldStatic勾选为Block，此时投射物会被Cube阻挡。可以才C++中对上面的参数进行修改，避免重复操作：

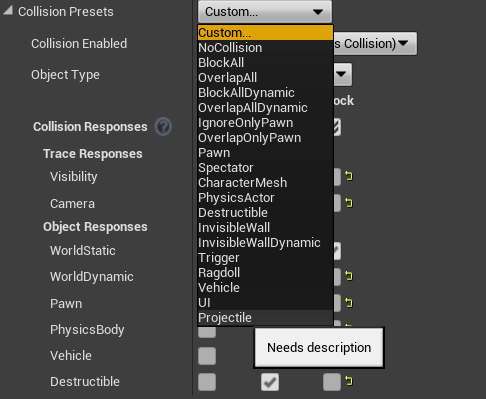


也可以在编辑器中新建配置文件：

Edit->project Settings->Collision->Preset.New



点击Accept，随后即可在蓝图类中Collision Presets中找到Projectile



当然，在C++中修改更方便，可以把之前的注释掉：

