[[中文字幕] Unreal Engine 4 Mastery：Create Multiplayer Games with C++\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1pb41177pn?p=11&spm_id_from=pageDriver)

**第一个项目**

**一、P10-P15（Actor相关）**

讲解Actor

1.P11：为Actor创建碰撞组件和静态网格体组件。

2.P12：处理碰撞事件

API：SetCollisionEnable() //设置碰撞

SetCollisionResponseToAllChannels() //将碰撞设置为响应所有通道

SetCollisionResponseToChannel() //将碰撞设置为响应（）通道

3.P13：拾取物品

角色经过物体，物体消失，表征角色捡起物体的布尔类型 变化。

4.P14：创建UI界面，显示提示信息。

5.P15：创建黑洞，吸入所有的模拟物理Actor。

UPrimitiveComponent:可模拟物体，且拥有变换的组件。(添加径向力)

**二、P16-P22（游戏规则的创建以及处理任务完成和游戏结束状态）**

1.P17：创建通关位置Actor。

2.P18：美化关卡；添加贴花；

UDecalComponent 贴花组件

3.P19：对游戏Mode进行操作，实现角色完成任务，关卡进行反馈。

实现简单的游戏逻辑（禁用玩家输入）。

**禁用玩家输入：Pawn->DisableInput()**

**GetWorld()->GetAuthGameMode() //获取GameMode，对于多人游戏，仅可在服务器调用**

4.P20：显示提示语

5.P21：在游戏完成后调整相机视角，显示其他画面。

将摄像机从第一人称视角移除，改为第三人称视角。（切换控制器可参考）

6.P22：创建发射平台。

通过内置函数：LaunchCharacter()

**三、P23 - P31 (AI守卫)**

1.P24：创建AI角色。

2.P25：创建AI感应组件的视觉模块OnSeePawn；

3.P26：创建AI感应组件的听觉模块OnhearNoise；

4.P27：对AI的听觉模块进行完善，在发射物击中的位置makenoise()；

5.P28：AI发现子弹碰撞发出的声音后，调整转向；

6.P29：游戏结束，根据任务是否完成进行差异性文本UI显示；

7.P30：AI反馈，设置不同的AI状态，根据状态调整AI角色头顶符号；

8.P31：AI在指定位置点间巡逻；

**多人游戏**

**四、P32 - P41（结合第一个项目介绍多人游戏同步的相关内容）**

1.P33：结合第一个项目说明多人游戏与其的差异；

2.P34：子弹发射物的网络复制同步；

3.P35：同步枪械的转动方向、发射物；

4.P36：拾取物的销毁；

5.P37：AI对应界面在客户端上的显示；（AI函数只在服务器执行）

6.P38：复制游戏状态；（一些关键变量的复制同步）

7.P39：GameMode不支持复制，从GameState中实现游戏结束；

8.P40：GameState、PlayController的进一步完善，实现视角切换、HUD显示的同步；

9.P41：打包游戏，开启多人联机；

**第二个项目——多人射击游戏**

**五、P42–P52（开始多人游戏创建）**

1.P43：创建项目；

2.P44：创建Character；

3.P45：为角色添加前后左右移动；

4.P46：设置视角变化；

5.P47：添加相机组件；

6.P48：添加弹簧臂组件；

7.P49：添加角色网格体；

8.P50：添加角色蹲伏；

9.P51：操作角色动画；

10.P52：进一步利用状态机设计角色动画；

**六、P53–P63（创建武器）**

1.P54：创建武器类，并将其附着于角色插槽中；

2.P55：将武器附加于角色上；

3.P56：为武器添加Fire函数；

4.P57：重写GetPawnViewLocation，调整子弹发射的起始位置；

5.P58：应用伤害（ApplyPointDamage()）；

6.P59：测试伤害；

7.P60：增加特效；（枪口火花与出血）

8.P61：增加弹药轨迹特效；（设置轨迹，用到了socket）

9.P62：设置准星；

10.P63：制作手榴弹类型的武器；

**七、P64–P74（丰富武器：开镜、反作用力等）**

1.P65：开镜；（用到了FMath中的FInterpTo()线性插值）；

2.P66：设置控制台变量：用于控制调试时Debug的线条等内容是否显示；

3.P67：调整代码格式，将游戏开始生成武器这一功能用代码方式实现；

4.P68：设置开枪时的枪口抖动；

5.P69：对角色设置不同的材质类型，以实现爆头、击中肉体等的区分；

6.P70：击中不同材质，显示不同效果；

UPhysicalMaterial::DetermineSurfaceType()获取材质类型；

7.P71：调整弹药射线的碰撞通道；

8.P72：增加爆头伤害；(通过区分材质设置不同伤害系数)

9.P73：自动开火；（通过定时器设定自动开火事件）

10.P74：活动课：设计自己的武器；

**八、P75–P82（设置玩家健康状态等相关的组件）**

1.P76：创建对应的组件；

2.P77：绑定受伤害事件：

ApplyDamage等事件会触发对应Actor的OnTakeAnyDamage，在此基础上，通过OnTakeAnyDamage.AddDynamic()可绑定受伤害后的事件。

3.P78：设置事件代理，响应生命值的更改；（\*可在蓝图调用）

4.P79：创建死亡动画；（动画是用角色动画实现的）

5.P80：创建与生命值相关的材质；（生命越大UI越不明显）

6.P81：通过UI界面显示与生命相关的材质；（调整材质的参数，将其与生命值绑定）

7.P82：创建爆炸桶；

**九、P83–P90（在以上基础上扩建构造多人游戏）**

1.P85：复制枪械；

2.P86：开火效果可从服务器RPC实现；

对生命值的更改应置于服务器；

//武器类的调整

**3.P87：通过定义可复制的结构体来调用生成函数；**

**4.P88：设置击中效果的同步；**

课程中是用属性同步做的，我使用了多播形式的RPC；

//健康组件的调整

5.P89：复制健康组件；通过同步角色中的bDied变量实现死亡动画的同步；

GetOwnerRole()：获取组件所属Actor的网络角色；

6.P90：同步复制爆炸桶；

通过SetReplicateMovement(true)同步移动效果，实现爆炸同步；并通过OnRep\_实现视觉效果的同步；

**十、P91–P（创建AI）**

1.P92：创建AI基本的Pawn类，设置其静态网格体不影响寻路系统；

2.P93：设置AI，自动寻找前往玩家的下一个路径点；

3.P94：制作追踪玩家完整的Actor；制作可追踪玩家上一个位置的物体；

4.P95：添加生命组件；

5.P96：击中后物体材质闪烁；

LastTimeDamageTaken：上次受伤害时间；

Mesh->CreateAndSetDynamicMaterialInstanceDynamicFromMaterial()

MaterialInstanceDynamic->SetScalarParameterValue()：设置动态材质的标量参数；

6.P97：Actor血量归零后，产生爆炸伤害，并生成粒子效果；

7.P98：通过碰撞球体，设置定时器，碰到角色后，进行自我伤害，最后爆炸；

8.P99：创建自爆前的声效；

UE4中各种失效类型参考：[UE4 Sound常见类 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/174420322)

9.P100：创建具有衰减效果的音频；音量可体现当前速度；

蓝图中：Map Range Clamped可将一定范围内的数值映射至另一范围。

10.P101：网络同步；

寻路网格体默认只在服务器执行；更改寻路相关函数的执行位置；

11.P102：同步粒子效果等，调整Health组件，调用OnRep\_进行客户端触发；

12.P103：挑战任务：AI附近有同类AI时，闪烁，且造成的伤害翻倍；

OverlapMultiByObjectType（）：使用对象类型在提供的位置测试形状的碰撞，并确定其重叠的组件集

思路：

通过创建FCo llisionShape检查该Actor一定范围内的同类Actor，进而做逻辑处理。