# Head:

## 属性声明：

UPROPERTY([specifier, specifier, ...], [meta(key=value, key=value, ...)])

Type VariableName;

### 常用Specifier：

VisibleDefaultsOnly:在蓝图中可见(不可编辑)

VisableAnywhere:公开至编辑器以及蓝图(仅显示)

EditDefaultsOnly:在(仅)蓝图中可编辑

EditAnywhere:公开变量至编辑器以及蓝图(可编辑)

EditInstanceOnly:在蓝图中可见不可编辑,只有实例化蓝图后才可编辑

BlueprintCallable:在蓝图中可调用（仅get）

BlueprintReadWrite:在蓝图中可读可写

BlueprintReadOnly:在蓝图中只读

### Meta:

限制变量大小(范围)

### 自定义分组：

Category:对变量增加标签分组

## 函数声明

UFUNCTION([specifier1=setting1, specifier2, ...], [meta(key1="value1", key2, ...)])

ReturnType FunctionName([Parameter1, Parameter2, ..., ParameterN1=DefaultValueN1, ParameterN2=DefaultValueN2]) [**const**];

### 常用Specifier：

BlueprintCallable:函数可被蓝图调用

BlueprintImplementableEvent:函数可在蓝图或关卡蓝图中实现(重载)

BlueprintNativeEvent:此函数旨在被蓝图覆盖掉，但是也具有默认原生实现。用于声明名称与主函数相同的附加函数，但是末尾添加了`Implementation，是写入代码的位置。如果未找到任何蓝图覆盖，该自动生成的代码将调用 Implementation` 方法

由于meta使用较少，这里不做赘述，官方解释如下：

Out声明由引用传递的参数，使函数对其进行修改。

Optional通过任选关键词可使部分函数参数变为任选，便于调用。任选参数的数值（调用方未指定）取决于函数。例如，SpawnActor 函数使用任选位置和旋转，默认为生成的 Actor 根组件的位置和旋转。添加 = [value] 参数可指定任选参数的默认值。例如：function myFunc(optional int x = -1)。在多数情况下，如无数值被传递到任选参数，将使用变量类型的默认值或零（例如 0、false、""、none）。

### 带有多输出的函数声明方式：

UFUNCTION(BlueprintCallable)

void TraceEvent(USceneComponent\* Actor,bool &IsHit,FHitResult &Hit);

在需要输出的变量名前添加&符号(无需在函数体中return变量)

# CPP:

## 获取常用变量：

Player相关：

添加GameplayStatics头文件，后用UGameplayStatics::即可调用player相关的function，例如GetPlayerPawn、GetPlayerCharacter等等。

Trace相关：

直接使用UKismetSystemLibrary调用其中的Trace函数，射线检测函数的返回值通常为bool

TraceChannel可以使用UEngineTypes::ConvertToTraceType((ECC\_Pawn))来直接调用默认的几种碰撞检测通道(这里使用的是Pawn类型)。

计算用函数：

添加UKismetMathLibrary头引用后可直接调用引擎内的默认计算函数。

## 操作事件绑定：

设定于SetupPlayerInputComponent中

事件绑定需要引用PlayerInputComponent，而后调用BindAction函数



注：绑定的函数不可有默认的输入值 需要在编辑器的项目设定中事先添加好操作事件

绑定轴只需将调用的函数改为BindAxis即可。