



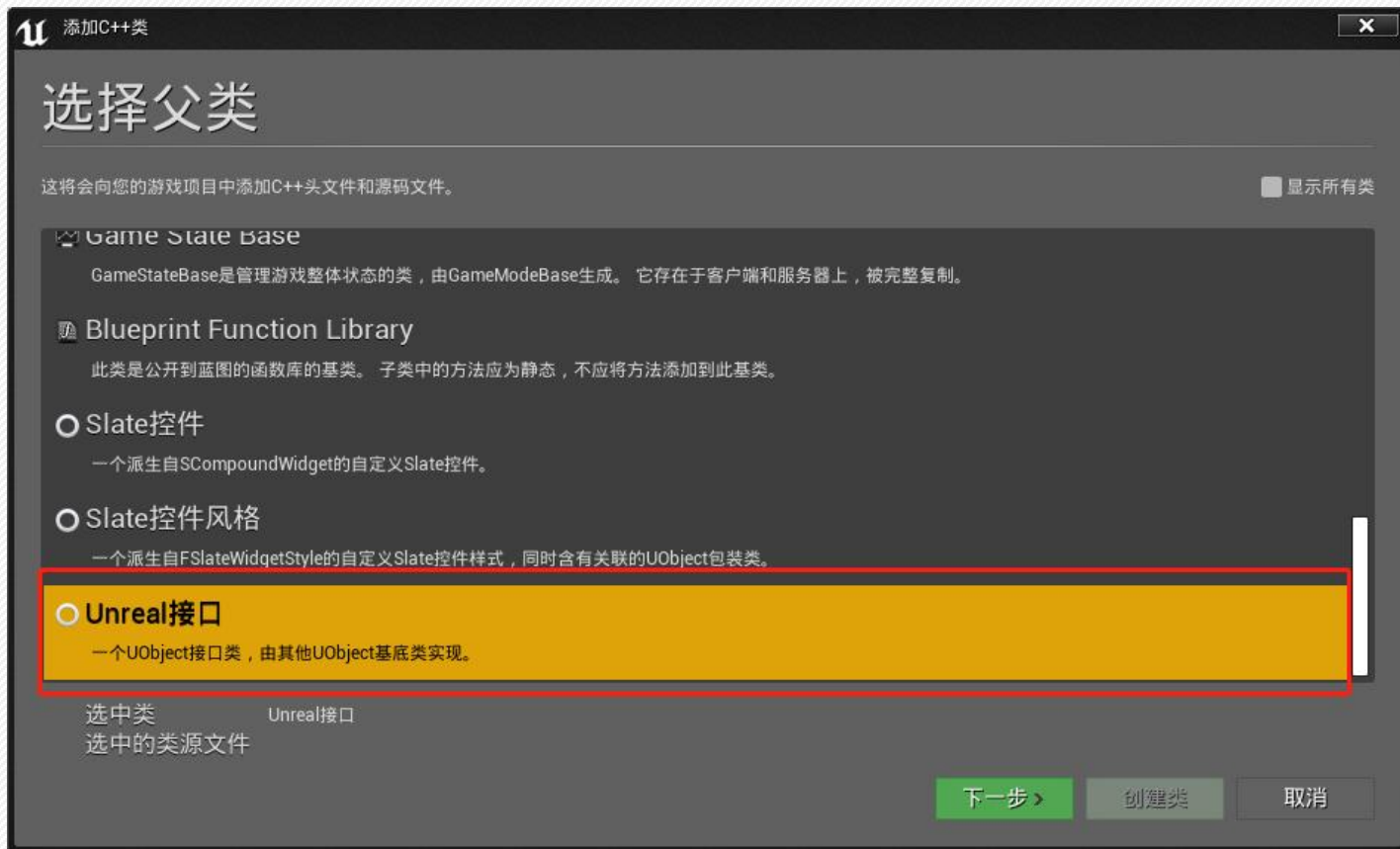
接口

虚幻四高级程序开发工程师



火星时代教育

接口类可以直接在虚幻编辑器中选择继承，然后完成构建



仅在C++中使用接口

当在构建的接口**仅需在C++中使用**时，则可以将接口类中的函数直接定义为**虚函数**。在需要使用接口的类中完成继承，并重写虚函数即可（尽量不要在接口中构建属性）。

如果编写的接口函数是纯虚函数，则无需完成定义，如果不是，则需要在源文件中添加定义函数。

接口调用，需要将对象类型进行转换为接口类（I类）然后调用接口函数即可。

以下是接口类

```
UINTERFACE(MinimalAPI)
class UDamageInterface : public UInterface
{
    GENERATED_BODY()
};

/**
 *
 */
class TANK90CPP_API IDamageInterface
{
    GENERATED_BODY()

    // Add interface functions to this class. This is the class that will be inherited to implement this interface.
public:

    virtual void BulletDamage(class ABulletActor*, class UPrimitiveComponent*);
};
```

继承接口的类

```
UCLASS()
class TANK90CPP_API AMapDirectorActor : public AActor, public IDamageInterface
{
    GENERATED_BODY()

public:
    // Sets default values for this actor's properties
    AMapDirectorActor();

protected:
    // Called when the game starts or when spawned
    virtual void BeginPlay() override;

    void SpawnWallComponent(int32 Code, EWallType Type);

    virtual bool BulletDamage(class ABulletActor* Bullet, class UPrimitiveComponent* Comp) override;
```

调用接口函数

```
35 void ABulletActor::OnComponentBeginOverlapEvent(UPrimitiveComponent* OverlappedComponent, AActor* OtherActor, UPrimitiveComponent* OtherComp, int32 OtherBodyIndex)
36 {
37     IDamageInterface* Di = Cast<IDamageInterface>(OtherActor);
38
39     if (Di)
40     {
41         Di->BulletDamage(this, OtherComp);
42     }
43 }
```

C++中构建与蓝图一起使用的接口

如果希望接口类在C++中定义，在**蓝图中使用**，则需要将接口函数，进行标记，**此种方法不仅可以提供蓝图使用，也可以供C++中使用。**

如果希望编写的接口函数能够在蓝图中**被调用**，则需要增加标记BlueprintCallable。

接口类声明（**此种声明方式不需要在源文件中进行定义**）

```
9 // This class does not need to be modified.
10 UINTERFACE(MinimalAPI)
11 class UDamageInterface : public UInterface
12 {
13     GENERATED_BODY()
14 };
15
16 /**
17  *
18  */
19 class UECPP_API IDamageInterface
20 {
21     GENERATED_BODY()
22
23     // Add interface functions to this class. This is the class that will be inherited to implement this interface.
24 public:
25     UFUNCTION(BlueprintCallable, BlueprintImplementableEvent)
26     void DamagePointer(AActor* Instigator);
27     UFUNCTION(BlueprintCallable, BlueprintNativeEvent)
28     void DamageRadius(AActor* Instigator);
29 };
```

如果在C++中希望获得接口能力，则需要继承接口。需要注意的是，必须继承I开头的接口名称，并且继承修饰为public。

如果接口中的函数使用BlueprintNativeEvent说明，则在继承类（C++）中可以编写同名函数，并用后缀 “_Implementation” 进行标记。

如果接口中的函数使用BlueprintImplementableEvent说明，**则无法在C++的继承类中实现接口函数**

继承接口的C++类（注意，不要省略override，函数的返回值，参数列表需要和接口的一致）

```
UCLASS()
class UECP_API AInterfaceActor : public AActor, public IDamageInterface
{
    GENERATED_BODY()

public:
    // Sets default values for this actor's properties
    AInterfaceActor();

protected:
    // Called when the game starts or when spawned
    virtual void BeginPlay() override;
    //当接口类中带有BlueprintNativeEvent标记的函数，则可以使用下面函数格式进行“重写”
    virtual void DamageRadius_Implementation(AActor* Instigator) override;

public:
    // Called every frame
    virtual void Tick(float DeltaTime) override;
};
```


调用函数，持有继承接口对象指针，第一步先转换到I类指针，调用Execute_接口函数名，参数第一位需要传递原对象指针，后面直接按照原函数参数填入即可。

如有返回值，Execute_函数也将返回数据。

```
6 void AUCCPPGameModeBase::AttackToActor(AActor* Target)
7 {
8     //检查对象是否继承接口
9     IDamageInterface* pI = Cast<IDamageInterface>(Target);
10    if (pI)
11    {
12        //执行调用 参数1是指实现接口的类实例， 参数2接口函数参数
13        pI->Execute_DamagePointer(Target, this);
14    }
```

感谢观看

虚 幻 四 高 级 程 序 开 发

-----•  火星时代教育 •-----