为编辑器添加可编辑属性

利用虚幻C++的反射机制，声明宏来实现

以变量 float WaveStrength; 为例

1 首先要声明为public才能被编辑器可见

```
 9      *
10      */
11     UCLASS()
12     class MYWARSHIPGAME_API AMyWarship : public AStaticMeshActor
13     {
14         GENERATED_BODY()
15
16
17     public:
18         AMyWarship();
19         virtual void BeginPlay() override;
20         virtual void Tick(float DeltaSeconds) override;
21
22         float WaveStrength;
23
24     private:
25         //float WaveStrength;
26         FVector2D  WaveOffset;
27         FVector2D  WaveFattor;
28     };
```

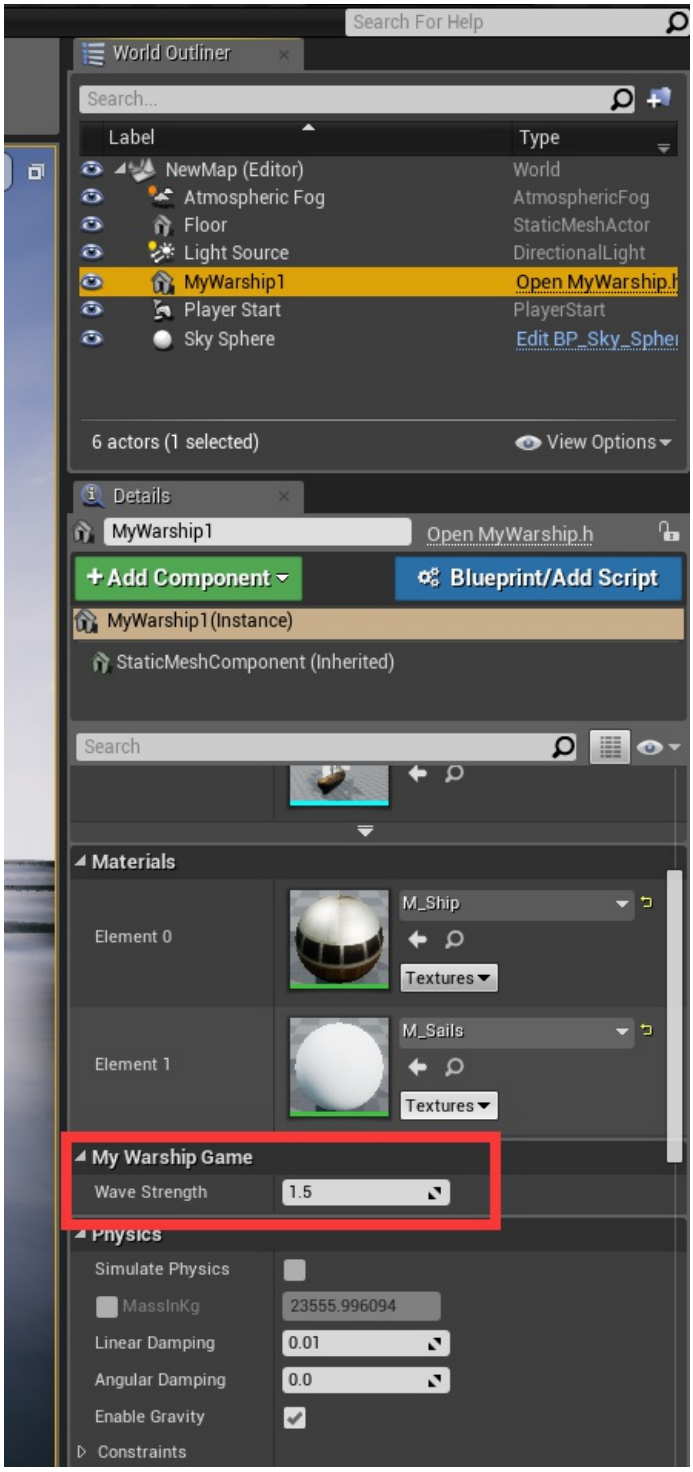
2 添加宏 UPROPERTY

```
6
7      public:
8          AMyWarship();
9          virtual void BeginPlay() override;
0          virtual void Tick(float DeltaSeconds) override;
1
2          UPROPERTY(EditAnywhere, Category = "MyWarshipGame")
3          float WaveStrength;
4
5      private:
6          //float WaveStrength;
7          FVector2D  WaveOffset;
```

第一个参数表示 可编辑 EditAnywhere

第二个参数表示 指定分类和其他属性分开

3 在编辑器里编译 完成后点击对象，观察对象属性



4

如有有其他变量与暴露变量相关联 要重写PostInitProperties方法

```
virtual void PostInitProperties() override;
```

```
void AMyWarship::PostInitProperties()
{
    Super::PostInitProperties();
    WaveOffset.X = FMath::FRandRange(0, WaveStrength * 10);
    WaveOffset.Y = FMath::FRandRange(0, WaveStrength * 10);
}
```

要让暴露的变量立即生效 要重写PostEditChangeProperty方法

因为只是编辑器需要，可以加入宏控制

```
    //只有编辑器才用到的代码，用宏定义处理
#if WITH_EDITOR
    //改变了暴露属性的值可以立即生效
    virtual void PostEditChangeProperty(FPropertyChangedEvent &PropertyChangedEvent) override;
#endif
```

```
#if WITH_EDITOR
void AMyWarship::PostEditChangeProperty()
{
    Super::PostEditChangeProperty();
    WaveOffset.X = FMath::FRandRange(0, WaveStrength * 10);
    WaveOffset.Y = FMath::FRandRange(0, WaveStrength * 10);
}
#endif
```

添加以一个只读的属性VisibleAnywhere

```
    UPROPERTY(VisibleAnywhere, Category = "MyWarshipGame")
    FVector2D WaveOffset;
```

暴露出来后只能看无法编辑