考虑到多人游戏的情况，将WBP放在PC或HUD中进行创建

额外的，对于一些互斥UI（如战绩板、商城等），也应该有对应的解决方案，如全局性质的CombatManager，在GameMode或GameState中实现

首先创建一个UserWidget类BaseWidget，实现HandleSwitch的功能

并在每次“波动”开关的时候，给出通知

创建蓝图子类，简单实现显示角色数值的逻辑

战斗界面角色UI

首先创建C++CombatUI

我们需要在里面创建一些变量和函数用于初始化UI，特别是BaseCharacter这个变量和函数InitUI

这里的InitUI只需要给Character赋值即可，并不需要干其他的（如绑定函数到进度条Percent）。这些其他的事我们用蓝图来搞，不用C++。

然后考虑要在合适的位置CreateWidget。在多人游戏中，WBP通常在PlayerController中CreateWidget。通过IsLocalController（）确保只在客户端创建

考虑到多人游戏中，玩家先进入地图，然后是选择角色进入地图，所以应当是在OnPossess时CreateWidget而不是BeginPlay时

在BasePlayerController的头文件中重载OnPossess函数，

首先在头文件中

public:

    UPROPERTY(BlueprintReadOnly)

    UCombatUI\* CombatUI;

    UPROPERTY(EditAnywhere,BlueprintReadWrite,Category="InitValues",meta=(ToolTip="用于显示角色主UI的变量，使用之前需要赋值"))

    TSubclassOf<UCombatUI> CombatUIClass;

protected:

    virtual void OnPossess(APawn\* InPawn) override;

然后实现OnPossess函数

void ABasePlayerController::OnPossess(APawn\* InPawn)

{

    Super::OnPossess(InPawn);

    if(InPawn)

    {

        GEngine->AddOnScreenDebugMessage(-1, 5.f, FColor::Red, "Possessed");

    }

    else    GEngine->AddOnScreenDebugMessage(-1, 5.f, FColor::Red, "Error");

    if(IsLocalController())

    {

        CombatUI = CreateWidget<UCombatUI>(this,CombatUIClass);

        if(CombatUI)

        {

            CombatUI->InitUI(Cast<ABaseCharacter>(InPawn));

            CombatUI->AddToPlayerScreen();

        }

    }

}

然后在BP\_BasePlayerController的细节面板中，给CombatUIClass参数赋值。应该赋的值是CombatUI的子类，现在创建这个类，并实现UI排版

然后在蓝图中重载InitUI函数，来将角色的数值绑定到上面

