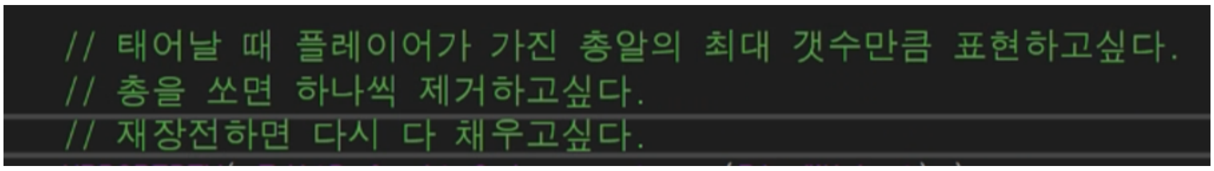
C++ : 총알 UI



```
// 태어날 때 플레이어가 가진 총알의 최대 갯수만큼 표현하고싶다.
// 총을 쏘면 하나씩 제거하고싶다.
// 재장전하면 다시 다 채우고싶다.
```

MainUI에 띄울 WBP_BulletItem 만들어주기

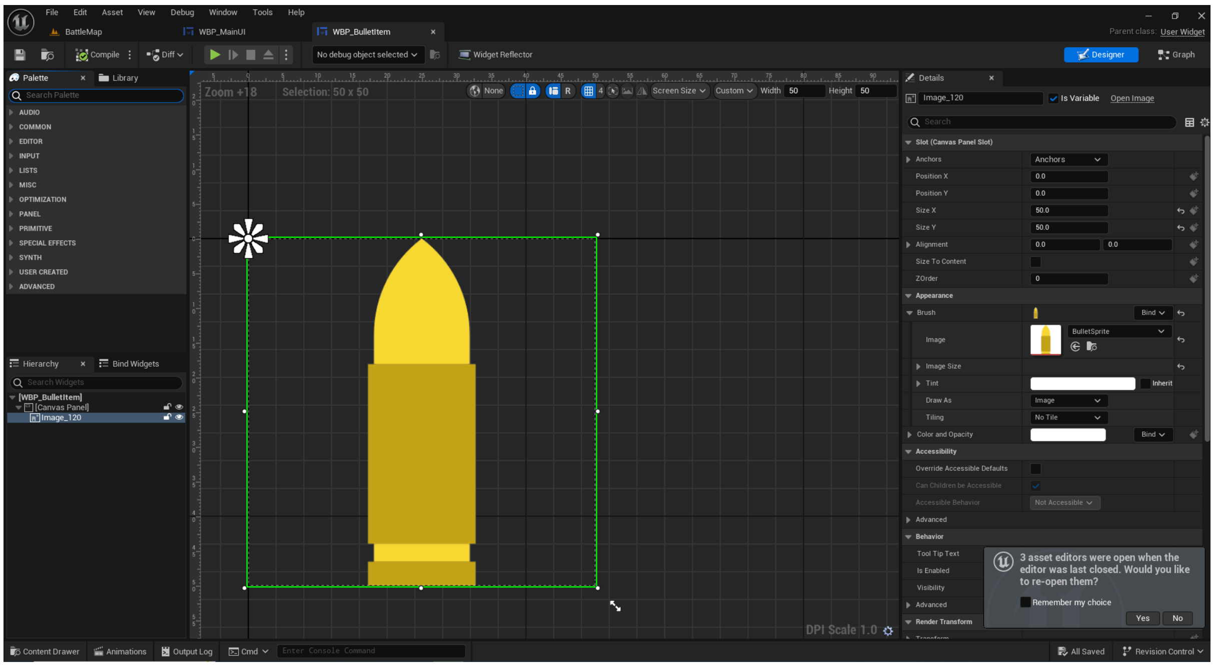


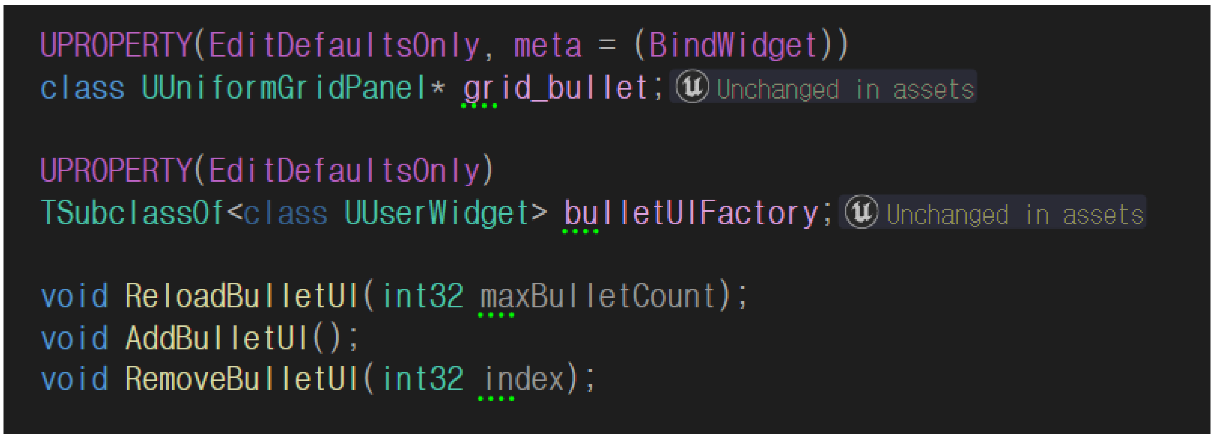
Image_120 ✓ Is Variable

Slot (Canvas Panel Slot)
Position X 0.0
Position Y 0.0
Size X 50.0
Size Y 50.0
Alignment 0.0 0.0
ZOrder 0

Appearance
Brush
Image BulletSprite
Draw As Image
Tiling No Tile

3 asset editors were open when the editor was last closed. Would you like to re-open them?

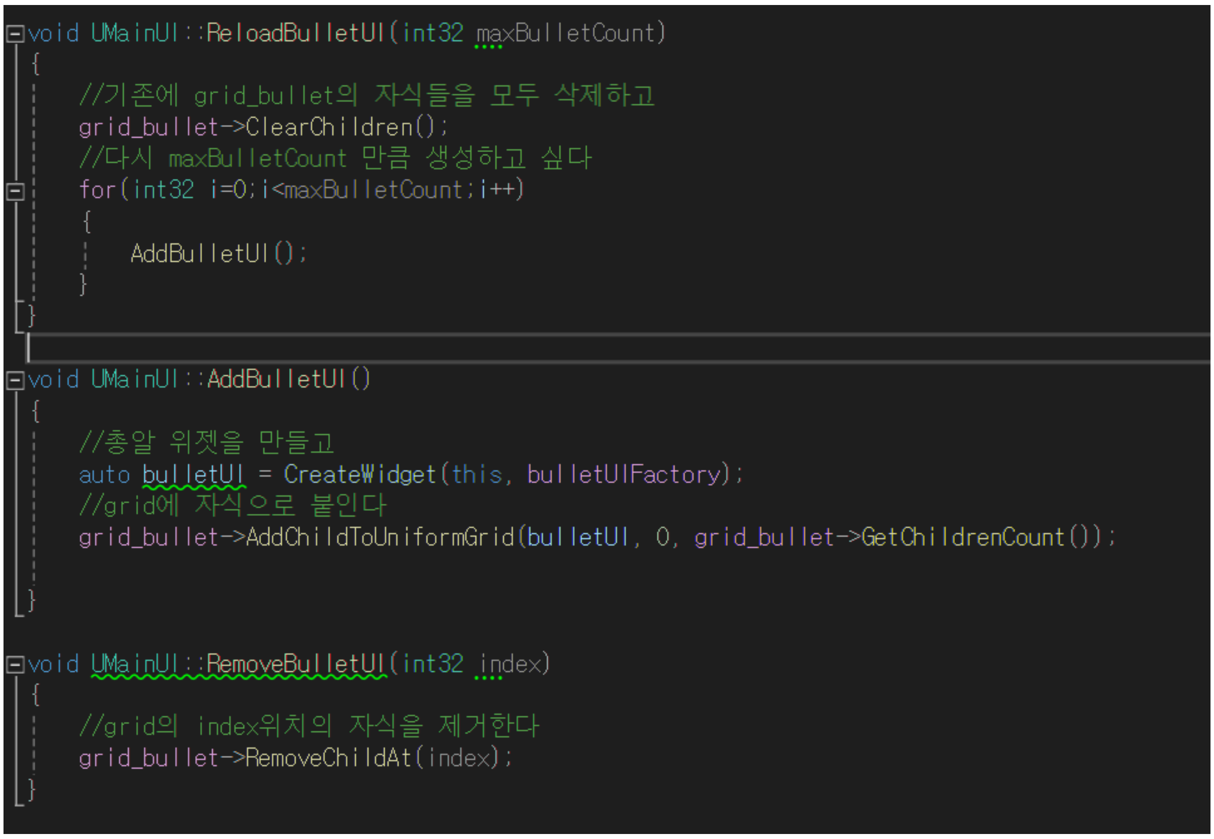
MainUI.h



```cpp
UPROPERTY(EditDefaultsOnly, meta = (BindWidget))
class UUniformGridPanel* grid_bullet; Unchanged in assets

UPROPERTY(EditDefaultsOnly)
TSubclassOf<class UUserWidget> bulletUIFactory; Unchanged in assets

void ReloadBulletUI(int32 maxBulletCount);
void AddBulletUI();
void RemoveBulletUI(int32 index);
```

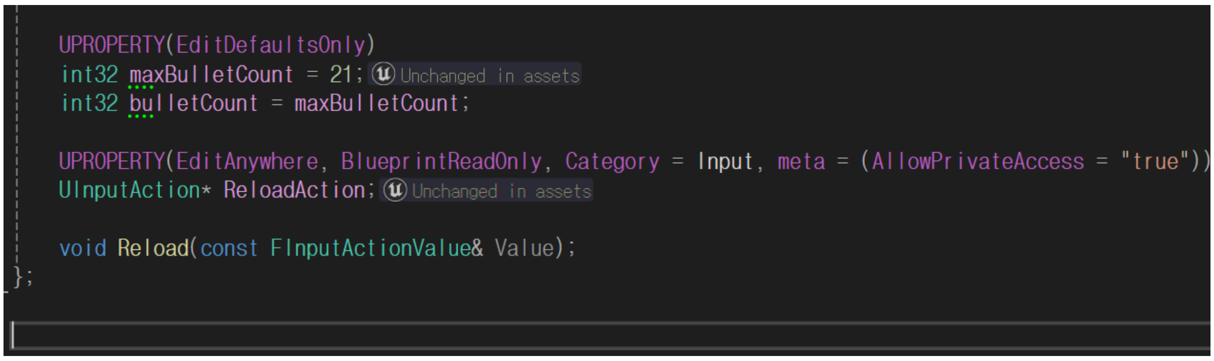
나중에 컴파일하고 bulletUIFactory 채워주기

MainUI.cpp



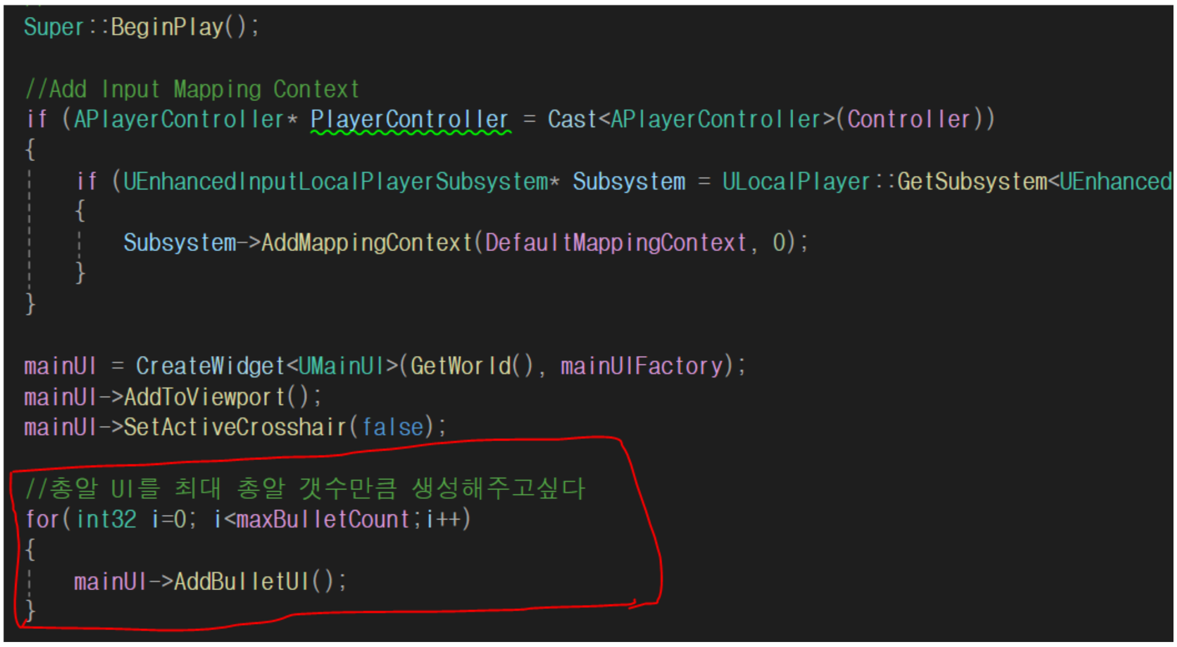
```cpp
void UMainUI::ReloadBulletUI(int32 maxBulletCount)
{
    //기존에 grid_bullet의 자식들을 모두 삭제하고
    grid_bullet->ClearChildren();
    //다시 maxBulletCount 만큼 생성하고 싶다
    for(int32 i=0;i<maxBulletCount;i++)
    {
        AddBulletUI();
    }
}

void UMainUI::AddBulletUI()
{
    //총알 위젯을 만들고
    auto bulletUI = CreateWidget(this, bulletUIFactory);
    //grid에 자식으로 붙인다
    grid_bullet->AddChildToUniformGrid(bulletUI, 0, grid_bullet->GetChildrenCount());

}

void UMainUI::RemoveBulletUI(int32 index)
{
    //grid의 index위치의 자식을 제거한다
    grid_bullet->RemoveChildAt(index);
}
```

player.h

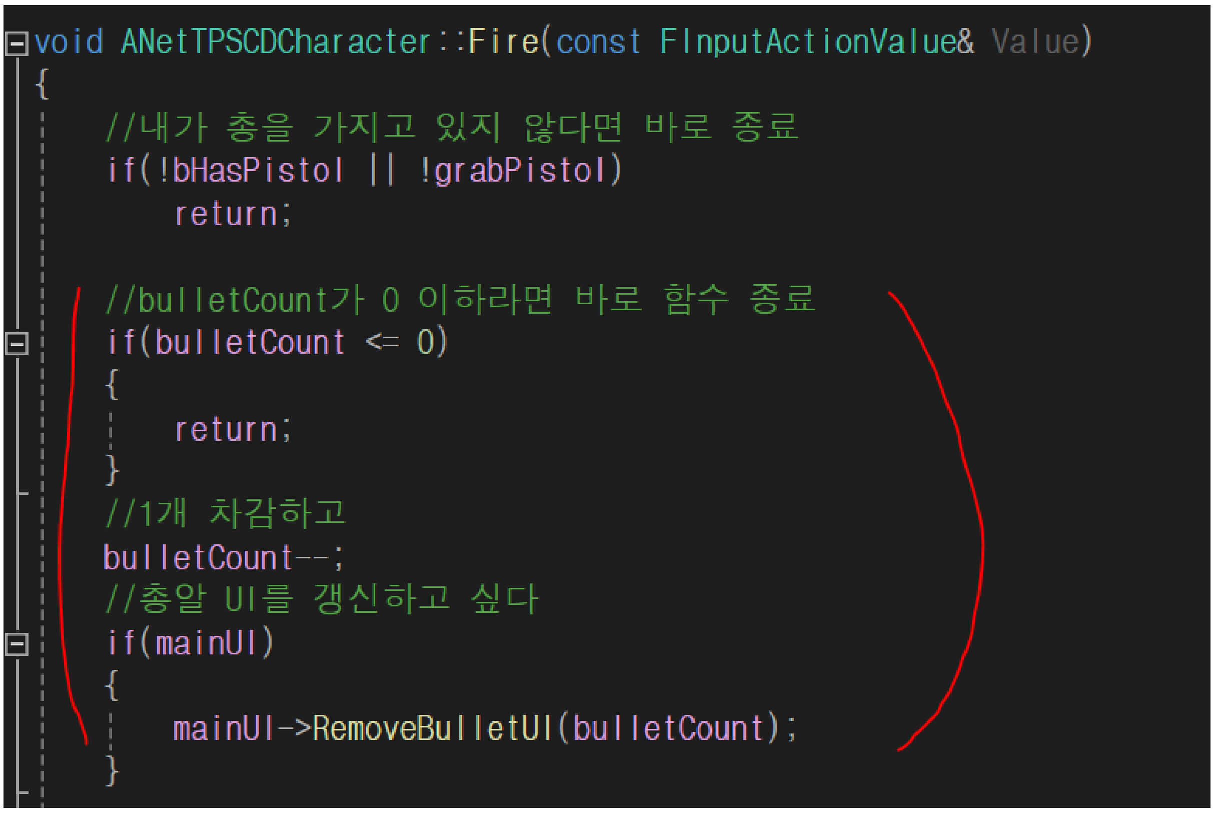


```cpp
	UPROPERTY(EditDefaultsOnly)
	int32 maxBulletCount = 21; Unchanged in assets
	int32 bulletCount = maxBulletCount;

	UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = Input, meta = (AllowPrivateAccess = "true"))
	UInputAction* ReloadAction; Unchanged in assets

	void Reload(const FInputActionValue& Value);
};
```

player.cpp

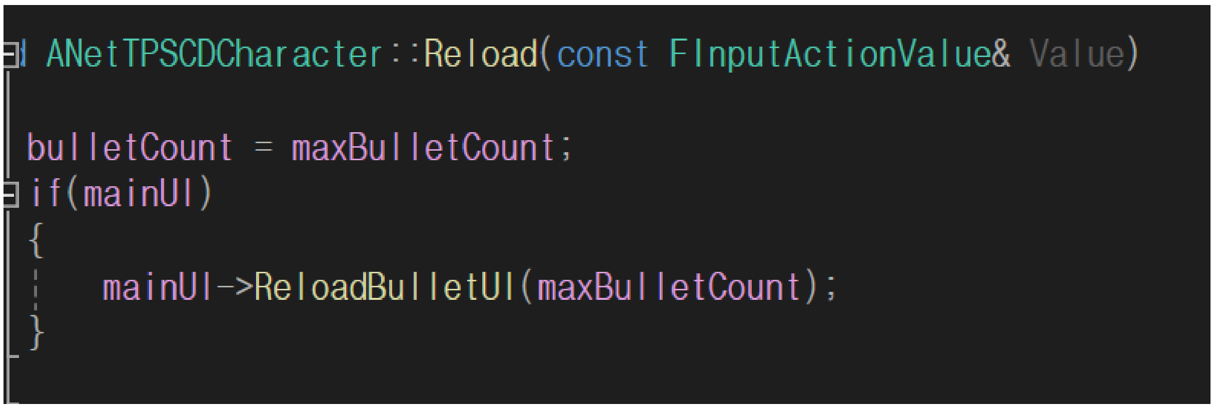


```cpp
Super::BeginPlay();

//Add Input Mapping Context
if (APlayerController* PlayerController = Cast<APlayerController>(Controller))
{
	if (UEnhancedInputLocalPlayerSubsystem* Subsystem = ULocalPlayer::GetSubsystem<UEnhanced
	{
		Subsystem->AddMappingContext(DefaultMappingContext, 0);
	}
}

mainUI = CreateWidget<UMainUI>(GetWorld(), mainUIFactory);
mainUI->AddToViewport();
mainUI->SetActiveCrosshair(false);

//총알 UI를 최대 총알 갯수만큼 생성해주고싶다
for(int32 i=0; i<maxBulletCount;i++)
{
	mainUI->AddBulletUI();
}
```



```
void ANetTPSCDCharacter::Fire(const FInputActionValue& Value)
{
	//내가 총을 가지고 있지 않다면 바로 종료
	if(!bHasPistol || !grabPistol)
		return;

	//bulletCount가 0 이하라면 바로 함수 종료
	if(bulletCount <= 0)
	{
		return;
	}
	//1개 차감하고
	bulletCount--;
	//총알 UI를 갱신하고 싶다
	if(mainUI)
	{
		mainUI->RemoveBulletUI(bulletCount);
	}
```

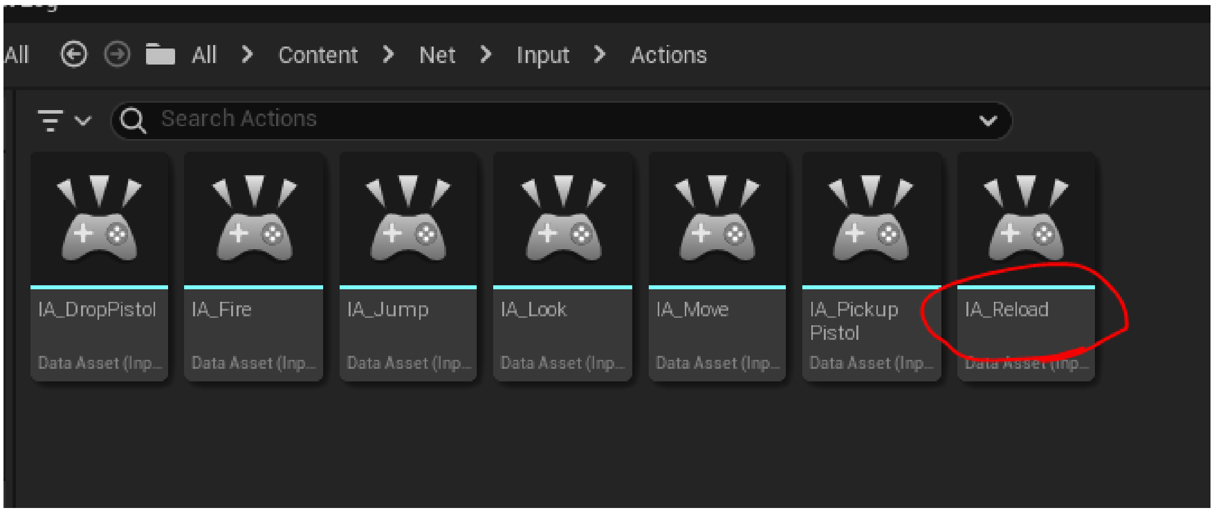
//총알 재장전

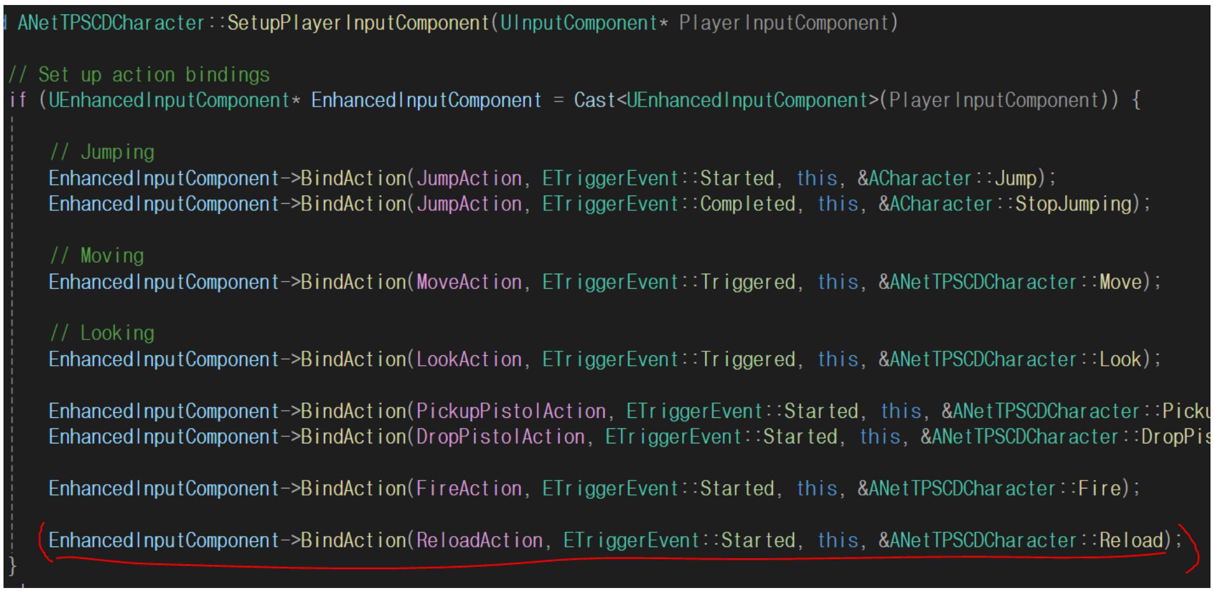


```
 ANetTPSCDCharacter::Reload(const FInputActionValue& Value)

bulletCount = maxBulletCount;
if(mainUI)
{
	mainUI->ReloadBulletUI(maxBulletCount);
}
```

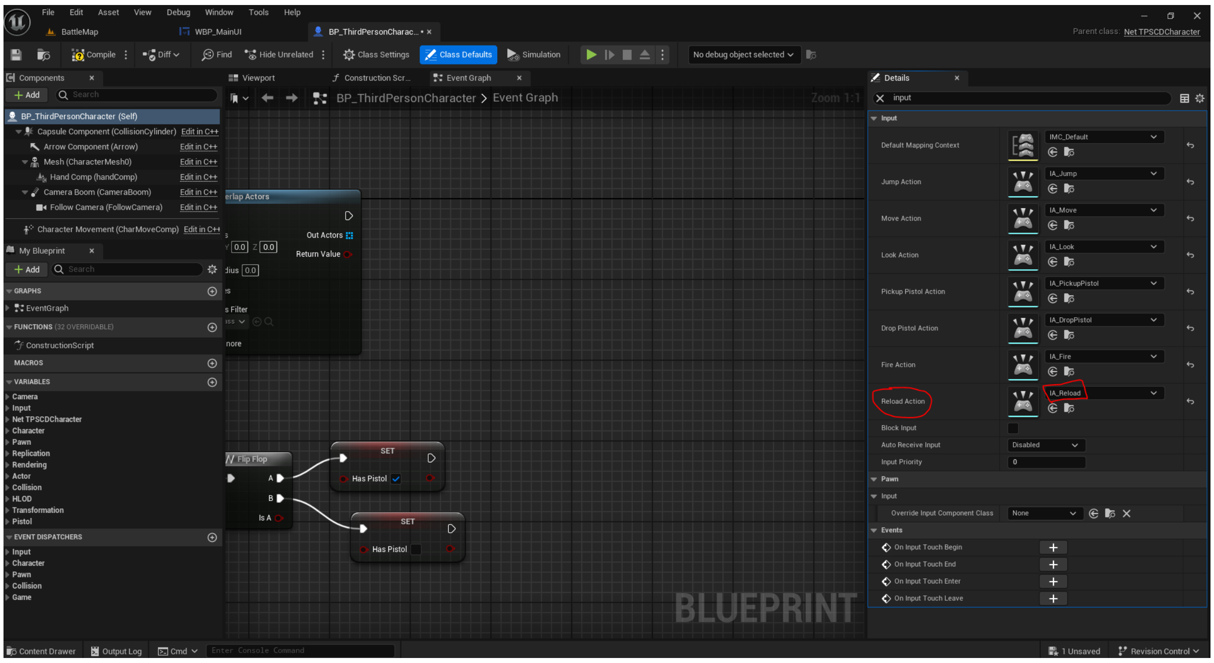
우클릭→ input Action

디폴트 설정에서 키 값 맵핑해주기





```
ANetTPSCDCharacter::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)

// Set up action bindings
if (UEnhancedInputComponent* EnhancedInputComponent = Cast<UEnhancedInputComponent>(PlayerInputComponent)) {

    // Jumping
    EnhancedInputComponent->BindAction(JumpAction, ETriggerEvent::Started, this, &ACharacter::Jump);
    EnhancedInputComponent->BindAction(JumpAction, ETriggerEvent::Completed, this, &ACharacter::StopJumping);

    // Moving
    EnhancedInputComponent->BindAction(MoveAction, ETriggerEvent::Triggered, this, &ANetTPSCDCharacter::Move);

    // Looking
    EnhancedInputComponent->BindAction(LookAction, ETriggerEvent::Triggered, this, &ANetTPSCDCharacter::Look);

    EnhancedInputComponent->BindAction(PickupPistolAction, ETriggerEvent::Started, this, &ANetTPSCDCharacter::Picku
    EnhancedInputComponent->BindAction(DropPistolAction, ETriggerEvent::Started, this, &ANetTPSCDCharacter::DropPis

    EnhancedInputComponent->BindAction(FireAction, ETriggerEvent::Started, this, &ANetTPSCDCharacter::Fire);

    EnhancedInputComponent->BindAction(ReloadAction, ETriggerEvent::Started, this, &ANetTPSCDCharacter::Reload);
}
```



Parent class: Net TPSCDCharacter

Details — input

Input

Default Mapping Context	IMC_Default
Jump Action	IA_Jump
Move Action	IA_Move
Look Action	IA_Look
Pickup Pistol Action	IA_PickupPistol
Drop Pistol Action	IA_DropPistol
Fire Action	IA_Fire
Reload Action	IA_Reload
Block Input	
Auto Receive Input	Disabled
Input Priority	0

Pawn

Input

Override Input Component Class	None

Events

- On Input Touch Begin
- On Input Touch End
- On Input Touch Enter
- On Input Touch Leave