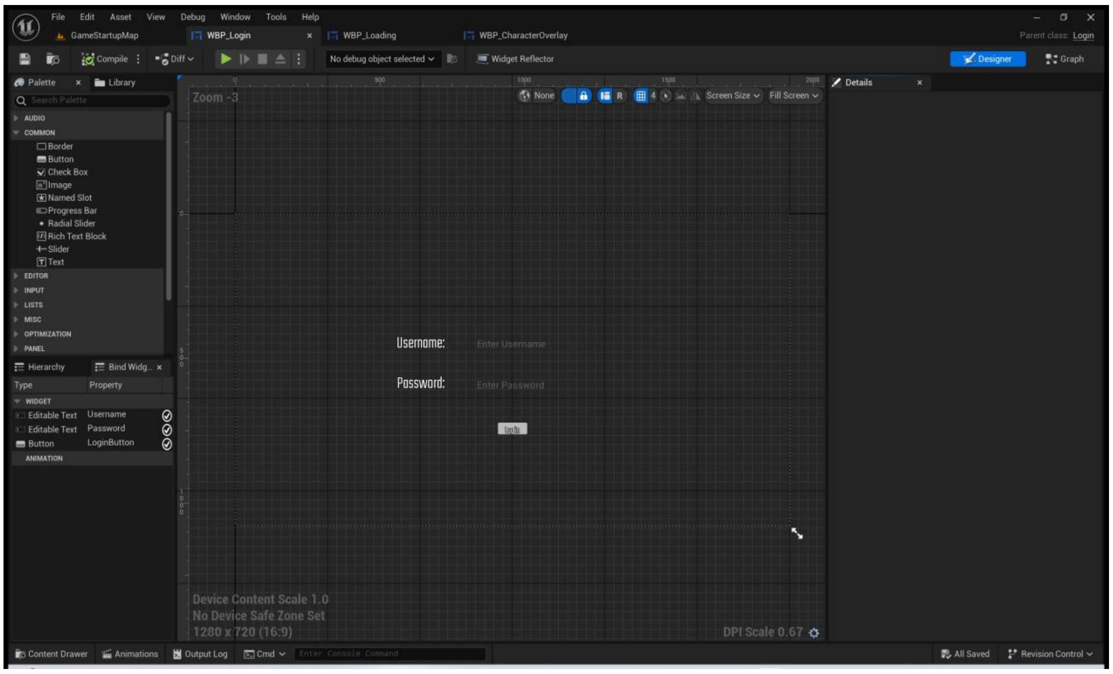
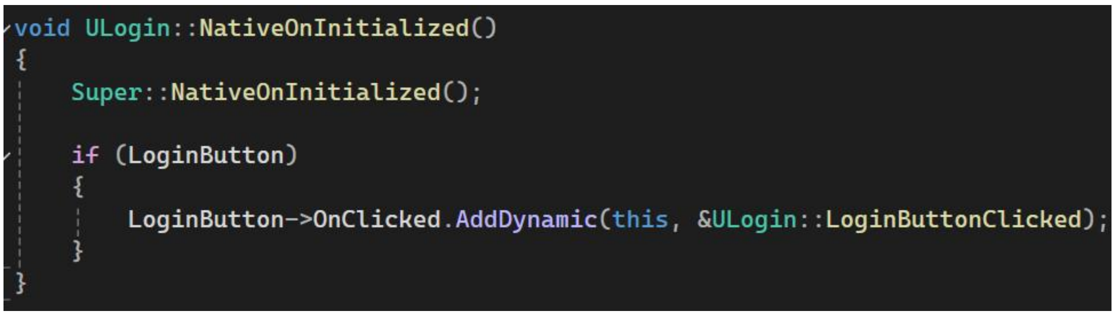
演示视频：

https://www.bilibili.com/video/BV1HCcte3Efs/?spm_id_from=333.1387.homepage.video_card.click&vd_source=f9ab6cfb3a9dcd7822b3ada50a02c2ae

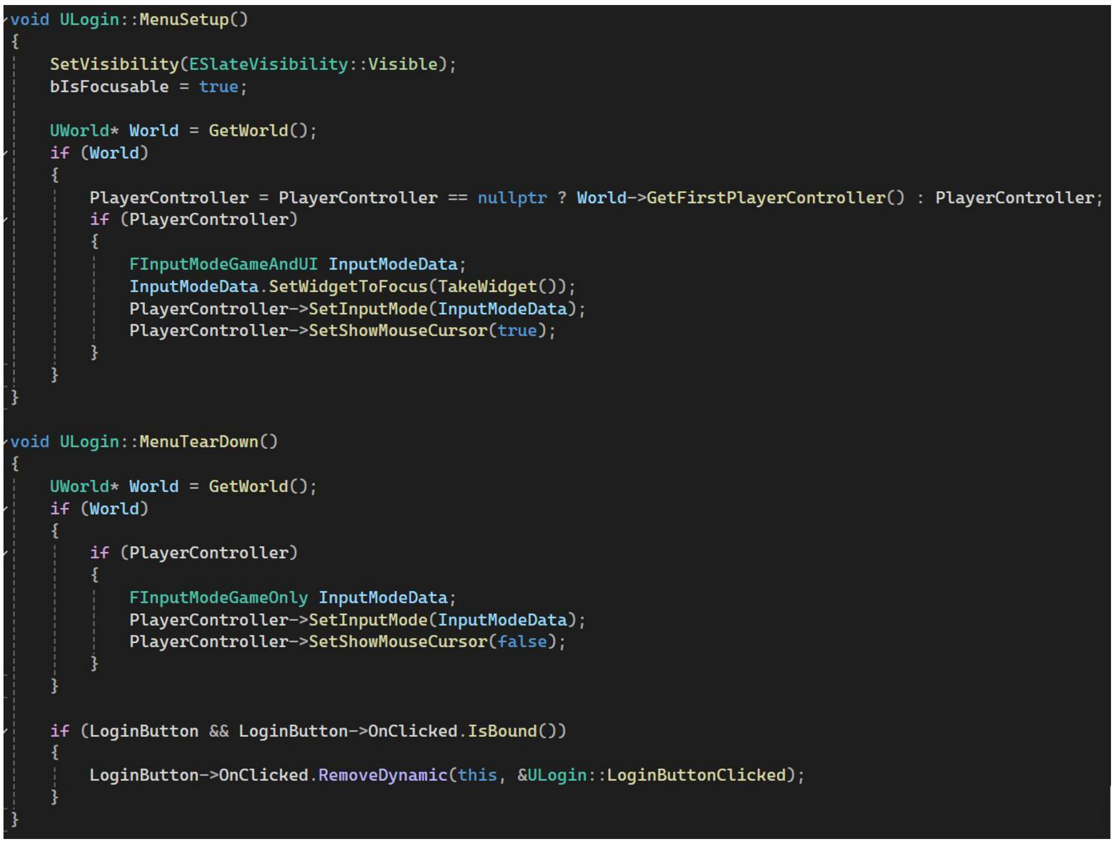
Login：



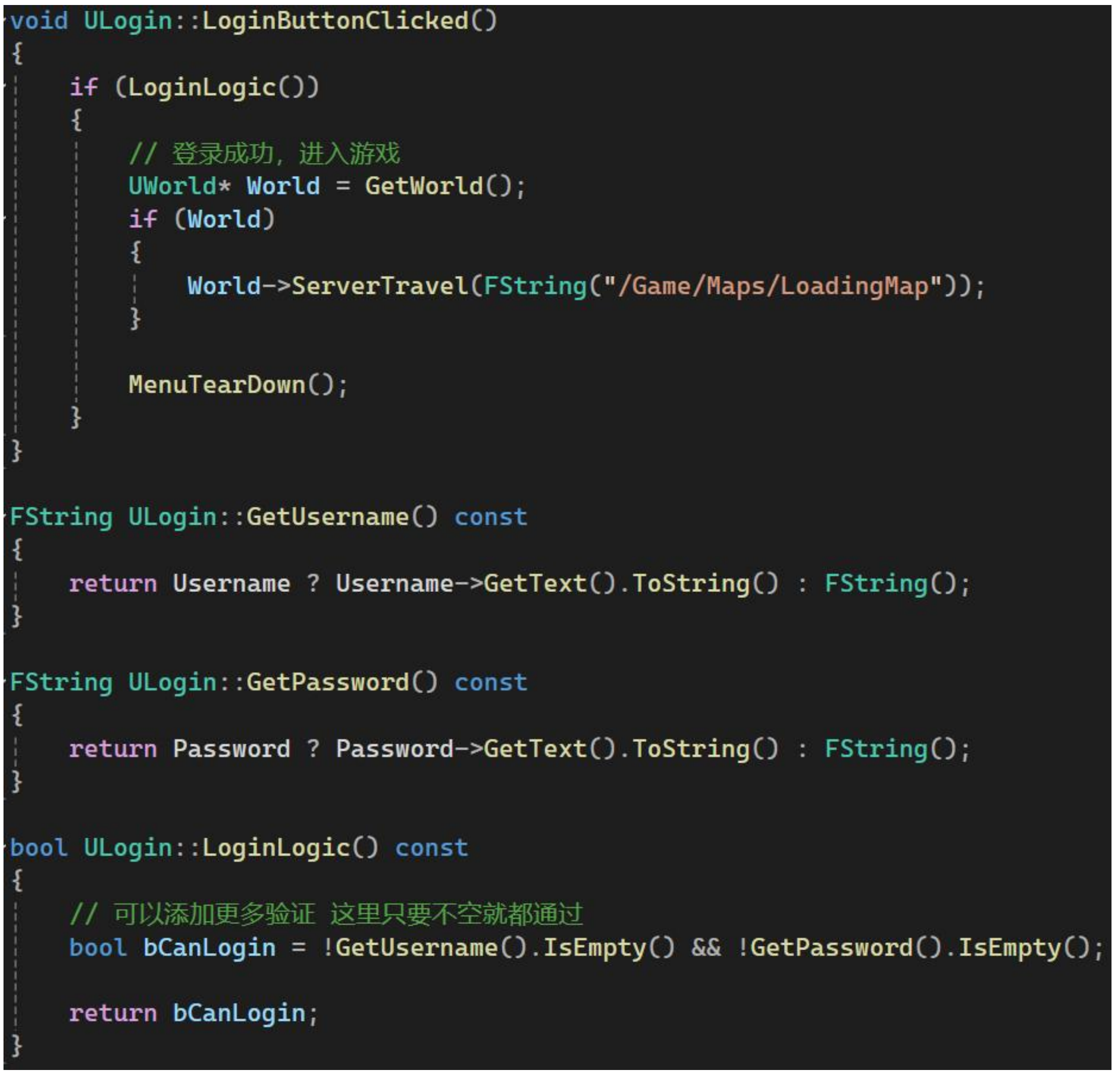
Type	Property
WIDGET	
Editable Text	Username
Editable Text	Password
Button	LoginButton
ANIMATION	



```cpp
void ULogin::NativeOnInitialized()
{
    Super::NativeOnInitialized();

    if (LoginButton)
    {
        LoginButton->OnClicked.AddDynamic(this, &ULogin::LoginButtonClicked);
    }
}
```

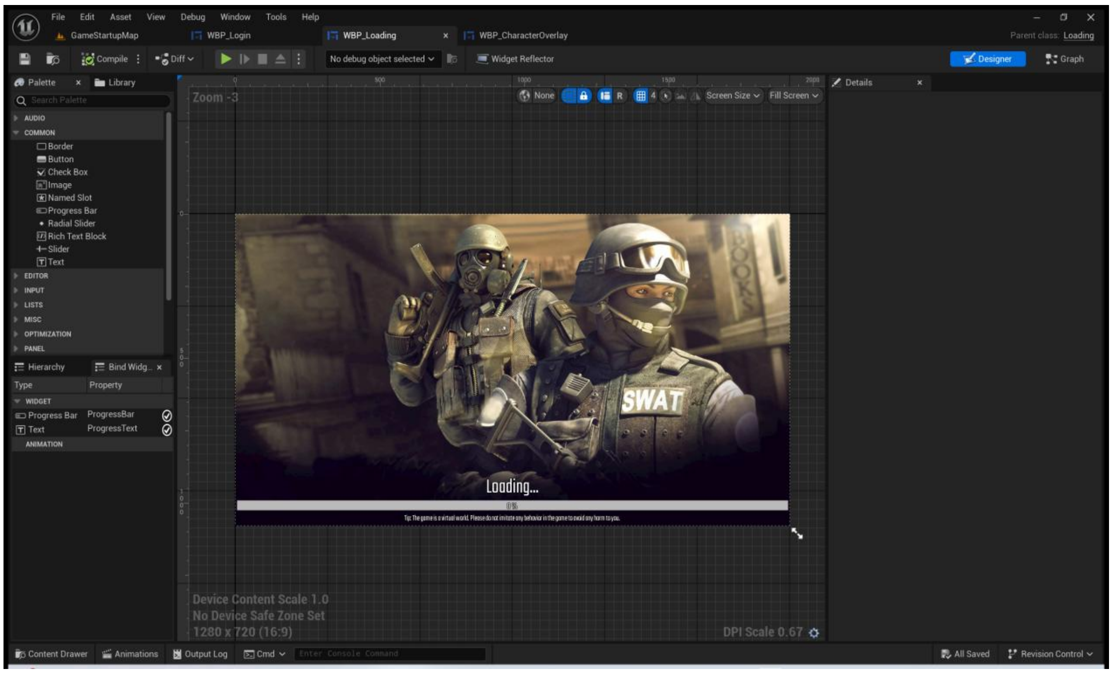


```cpp
void ULogin::MenuSetup()
{
    SetVisibility(ESlateVisibility::Visible);
    bIsFocusable = true;

    UWorld* World = GetWorld();
    if (World)
    {
        PlayerController = PlayerController == nullptr ? World->GetFirstPlayerController() : PlayerController;
        if (PlayerController)
        {
            FInputModeGameAndUI InputModeData;
            InputModeData.SetWidgetToFocus(TakeWidget());
            PlayerController->SetInputMode(InputModeData);
            PlayerController->SetShowMouseCursor(true);
        }
    }
}

void ULogin::MenuTearDown()
{
    UWorld* World = GetWorld();
    if (World)
    {
        if (PlayerController)
        {
            FInputModeGameOnly InputModeData;
            PlayerController->SetInputMode(InputModeData);
            PlayerController->SetShowMouseCursor(false);
        }
    }

    if (LoginButton && LoginButton->OnClicked.IsBound())
    {
        LoginButton->OnClicked.RemoveDynamic(this, &ULogin::LoginButtonClicked);
    }
}
```

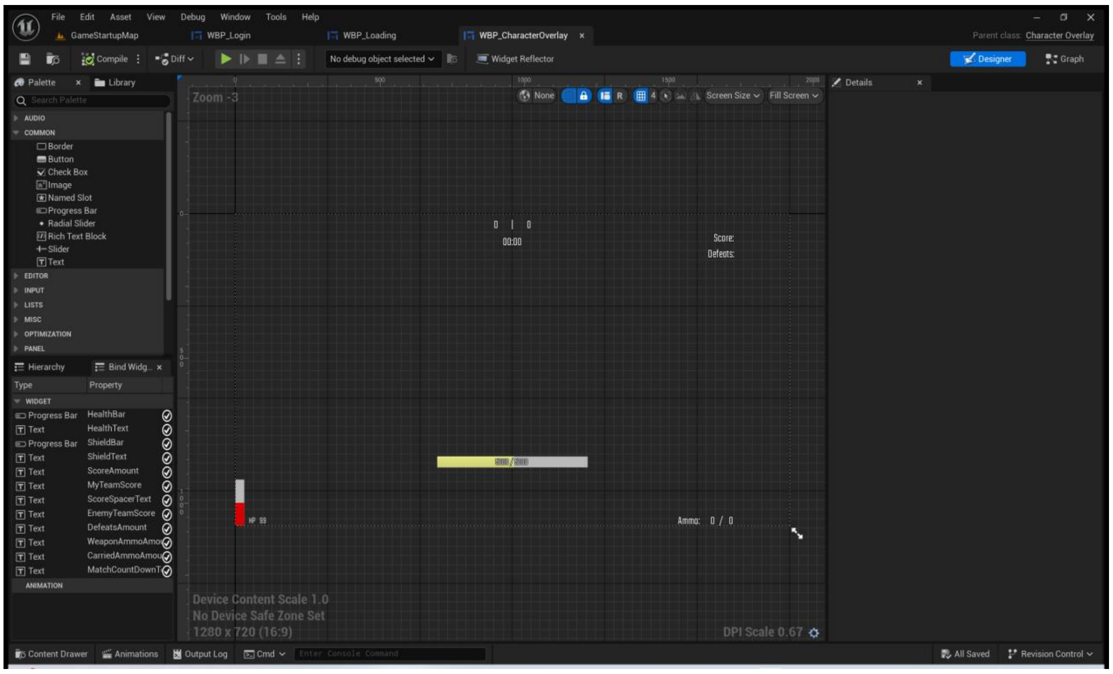


```cpp
void ULogin::LoginButtonClicked()
{
    if (LoginLogic())
    {
        // 登录成功，进入游戏
        UWorld* World = GetWorld();
        if (World)
        {
            World->ServerTravel(FString("/Game/Maps/LoadingMap"));
        }

        MenuTearDown();
    }
}

FString ULogin::GetUsername() const
{
    return Username ? Username->GetText().ToString() : FString();
}

FString ULogin::GetPassword() const
{
    return Password ? Password->GetText().ToString() : FString();
}

bool ULogin::LoginLogic() const
{
    // 可以添加更多验证 这里只要不空就都通过
    bool bCanLogin = !GetUsername().IsEmpty() && !GetPassword().IsEmpty();

    return bCanLogin;
}
```

Loading：



In Game：



在 PlayerController.cpp 中写诸多函数修改 HUD 中血量、子弹、得分等等。在需要修改时通过 PlayerController 调用即可。

关于准心，HUDPackage 中的 CrosshairSpread、bHit、HitScale 分别表示准心扩散多大、是不是击中、击中准心的 Scale。

```cpp
FVector2D WaldSpeedRange(0.f, Character->GetCharacterMovement()->MaxWalkSpeed);
FVector2D CrosshairSpreadRange(0.f, 1.f);
FVector Velocity = Character->GetVelocity();
Velocity.Z = 0.f;

CrosshairVelocityFactor = FMath::GetMappedRangeValueClamped(WaldSpeedRange, CrosshairSpreadRange, Velocity.Size());

if (Character->GetCharacterMovement()->IsFalling())
{
    CrosshairInAirFactor = FMath::FInterpTo(CrosshairInAirFactor, 1.f, DeltaTime, 10.f);
}
else
{
    CrosshairInAirFactor = FMath::FInterpTo(CrosshairInAirFactor, 0.f, DeltaTime, 10.f);
}

if (bCanReduceCrosshairShootFactor)
{
    // 玩家开火后过一会会，该值开始缩小
    CrosshairShootFactor = FMath::FInterpTo(CrosshairShootFactor, 0.f, DeltaTime, 5.f);
}

// 角色速度 是否在空中 是否开枪 共同决定CrosshairSpread
HUDPackage.CrosshairSpread = CrosshairVelocityFactor + CrosshairInAirFactor + CrosshairShootFactor;

// 击中反馈结束了 变回普通准心
if (HitCrosshairScale <= 0.95f)
{
    HUDPackage.bHit = false;
}
else
{
    // 给玩家显示击中反馈 击中反馈逐帧缩小
    HitCrosshairScale = FMath::FInterpTo(HitCrosshairScale, 0.8f, DeltaTime, 10.f);
    HUDPackage.bHit = true;
    HUDPackage.HitScale = HitCrosshairScale;
}

HUD->SetHUDPackage(HUDPackage);
```

会在 ProjectileBullet.cpp 的 OnHit 函数中根据 OtherActor 的类型，决定要不要给出准心击中反馈，如果需要，直接将 CombatComponent 中的 HitCrosshairScale 设置为 2.5 即可。