1. 类创建特效

.h

Class UParticleSystem;

UPROPERTY(EditDefaultsOnly,Category = "Effects") //公开声明

UParticleSystem\* PickupFX;

.cpp

UGameplayStatics::SpawnEmitterAtLocation(this, PickupFX, GetActorLocation());

在原处生成 特效 获取物体坐标

1. 创建Actor类

.h

UPROPERTY(VisibleAnywhere,Category=”Components”)

UStaticMeshComponent\* MeshComp;

UPROPERTY(VisibleAnywhere,Category=”Components”)

USphereComponent\* SphereComp

.cpp

MeshComp = CreateDefaultSubobject<UStaticMeshComponent>(TEXT(“MeshComp”));

//创建名为“MeshComp”的静态网格体

RootComponent = MeshComp;

//MeshComp 设为根组件

SphereComp = CreateDefaultSuboboject<USphereComponent>(TEXT(“SphereComp”));

//创建名为“SphereComp“的静态网格体

SphereComp->SetupAttachment(MeshComp);

//SphereComp 设为MeshComp的子组件

1. 设置碰撞 碰撞响应
2. .h文件

UFUNCTION()

void HandleOverlap(UPrimitiveComponent\* OverlappedComponent, AActor\* OtherActor, UPrimitiveComponent\* OtherComp, int32 OtherBodyIndex, bool bFromSweep, const FHitResult & SweepResult);

.cpp文件

OverlapComp->OnComponentBeginOverlap.AddDynamic(this,

&AFPSExtractionZone::HandleOverlap);

void AFPSExtractionZone::HandleOverlap(UPrimitiveComponent\* OverlappedComponent, AActor\* OtherActor, UPrimitiveComponent\* OtherComp, int32 OtherBodyIndex, bool bFromSweep, const FHitResult & SweepResult)

1. .h文件

virtual void NotifyActorBeginOverlap(AActor\* OtherActor) override;

.cpp文件

SphereComp->SetCollisionEnabled(ECollisionEnabled::QueryOnly);

启用碰撞 仅查询

SphereComp->SetCollisionResponseToAllChannels(ECR\_Ignore);

开启碰撞响应 无视

SphereComp->SetCollisionResponseToChannel(ECC\_Pawn, ECR\_Overlap);

设置碰撞响应通道 仅Pawn类 重叠

void AFPSObjectiveActor::NotifyActorBeginOverlap(AActor\* OtherActor)

{

Super::NotifyActorBeginOverlap(OtherActor);

}

1. 转化类型

AFPSCjaracter\* MyCharacter = Cast<AFPSCharacter>(OtherActor);

1. 设置半径

InnerSphereComponet->SetSphereRadius(100);

1. 创建数组

TArray<UPrimitiveComponent\*> OverlappingComps;

数组 原始组件 组名

1. 施加作用力

PrimComp->AddRadialForce(GetActorLocation(),SphereRadius,ForceStength,

半径 力的大小

ERadialImpulseFalloff::RIF\_Constant,ture)

恒速 加速变化

1. 日志

UE\_LOG(LogTemp, Log, TEXT("Overlapped with extraction zone!"));

九、设置响应范围

OverlapComp->SetBoxExtent(Fvector(200.0f))

x,y,z=200个单位

1. 显示隐藏项

OverLapComp->SetHiddenInGame(false);

1. 贴花组件

.h

UPROPERTY(VisibleAnywhere, Category = "Components")

UDecalComponent\* DacalComp;

.cpp

DecalComp = CreateDefaultSubobject<UDecalComponent>()

创建默认子对象 贴花组件

十一、禁用输入

InstigatorPawn->DisableInput(nullptr)

专门禁用对玩家控制器控制着的Pawn 即游戏角色的输入

十二、公开声明函数

.h

UFUNCTION(BlueprintImplementableEvent,Category="GameMode")

设置为蓝图可实现事件 类目 游戏模式

void OnMissionCompleted(APawn\* InstigatorPawn);

.cpp

OnMissionCompleted(InstigatorPawn);

十三、获取游戏模式

GetWorld()->GetAuthGameMode()

客户端多人设置时无效

十四、声音

.h

UPROPERTY(EditDefaultsOnly,Category="Sounds")

USoundBase\* ObjectiveMissingSound;

音基

.cpp

UGameplayStatics::PlaySound2D(this, ObjectiveMissingSound);

十五、设置视图

.h

UPROPERTY(EditDefaultsOnly, Category = "Spectating")

TSubclassOf<AActor> SpectatingViewpointClass;

.cpp

TArray<AActor\*> ReturnnedActors;

UGameplayStatics::GetAllActorsOfClass(this, SpectatingViewpointClass,

ReturnnedActors);

if (ReturnnedActors.Num()>0)

{

AActor\* NewViewTarget = ReturnnedActors[0];

APlayerController\* PC =

Cast<APlayerController>(InstigatorPawn->GetController());

if (PC)

{

PC->SetViewTargetWithBlend(NewViewTarget, 0.5f, EViewTargetBlendFunction::VTBlend\_Cubic);

}

}

十六、创建ACTOR

AActor\* NewViewTarget;

十七、创建数组(actor)

TArray<AActor\*> Actors;