**Battery Collector**

Simple Battery Collector game.

A BP version of the character class is used in the level.

Create a class Batter of Actor type

Add following to header file.

UPROPERTY(VisibleAnyWhere, Category = "Component")

UStaticMeshComponent\* MeshComp;

UPROPERTY(VisibleAnyWhere, Category = "Component")

UBoxComponent\* BoxComp;

UPROPERTY(VisibleAnyWhere, Category = "Component")

USceneComponent\* SceneComp;

Forward declaration needed.

class UBoxComponent;

class USceneComponent;

Now to C++ file add following to the constructor

SceneComp = CreateDefaultSubobject<USceneComponent>(TEXT("SceneComp"));

BoxComp = CreateDefaultSubobject<UBoxComponent>(TEXT("SphereComp"));

MeshComp = CreateDefaultSubobject<UStaticMeshComponent>(TEXT("MeshComp"));

RootComponent = SceneComp;

BoxComp->SetupAttachment(SceneComp);

MeshComp->SetupAttachment(SceneComp);

Need the include files

#include "Components/BoxComponent.h"

#include "Components/StaticMeshComponent.h"

Create a BP version of Battery called BatteryBP and add a cylinder and adjust the scale.

Add following to Constructor

MeshComp->SetCollisionEnabled(ECollisionEnabled::NoCollision);

BoxComp->SetCollisionEnabled(ECollisionEnabled::QueryOnly);

BoxComp->SetCollisionResponseToAllChannels(ECR\_Ignore);

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

**Adding Particle Effects**

Create a variable for FX

UPROPERTY(EditDefaultsOnly, Category = "Effects")

UParticleSystem\* PicksupFX;

Create function in header file

void PlayEffects();

Go to implementation in C++ file

void ABattery::PlayEffects()

{

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

}

Need include file

#include "Kismet/GameplayStatics.h"

add following to begin play to test it.

PlayEffects();

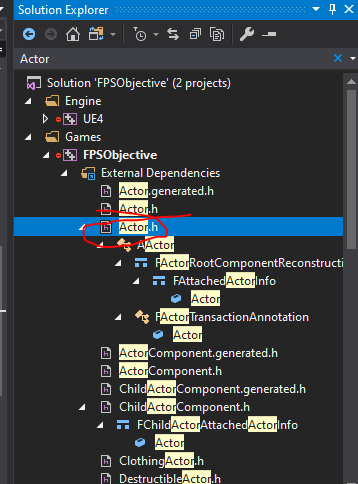
We should be able to get the effects in editor. (pick a particle effect in class defaults)

**Play effects on Overlap**

add following method declaration to header file (can comment out the playeffect() in beginplay)

virtual void NotifyActorBeginOverlap(AActor\* OtherActor);

Method can be found in the following class



then add following definition to C++ file.

void ABattery::NotifyActorBeginOverlap(AActor\* OtherActor)

{

Super::NotifyActorBeginOverlap(OtherActor);

PlayEffects();

}

**Casting & Point incrementing**

Now we need to increment a variable in player character.

Create a variable in PlayerCharacter header file.

int32 BatteryCollected;

We need to check if the pickup actor is the player character.

So do the following in Battery C++ file

#include "BatteryCollectorCharacter.h"

Add the following to the overlap method

ABatteryCollectorCharacter\* OverlapChar = Cast<ABatteryCollectorCharacter>(OtherActor);

Add an if to check if the pointer is not null

if (OverlapChar)

{

OverlapChar->BatteryCollected++;

}

We can move the other statements into the if condition and add a print log as well

Final method will look like this

ABatteryCollectorCharacter\* OverlapChar = Cast<ABatteryCollectorCharacter>(OtherActor);

if (OverlapChar)

{

OverlapChar->BatteryCollected++;

Super::NotifyActorBeginOverlap(OtherActor);

PlayEffects();

Destroy();

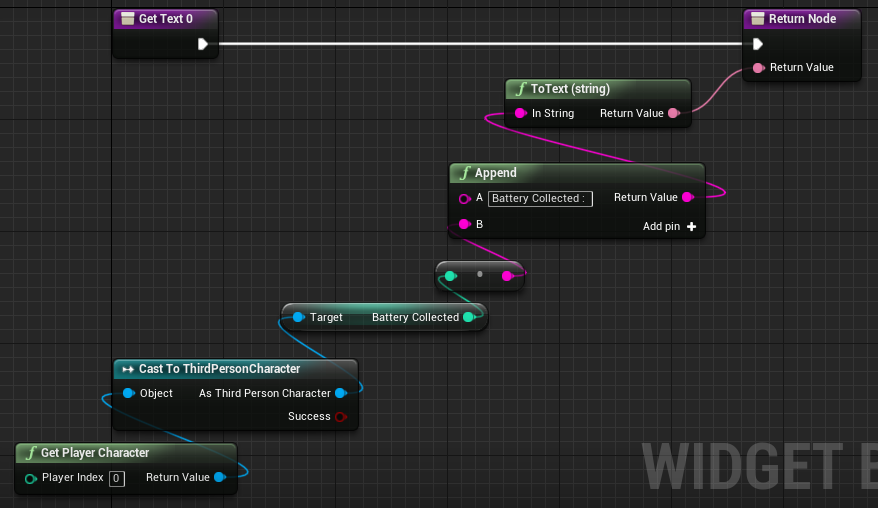
UE\_LOG(LogTemp, Warning, TEXT("overlap %d"), OverlapChar->BatteryCollected);

}

Create widget to display the battery collected.



In the widget add the following BP



Need to expose the variable to BP before this.

So add the following to header file of character

UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "Gameplay")

int32 BatteryCollected;

Create HUD and add following BP



Create a game mode and select HUD and Default Pawn class in the game mode

Add the new game mode in the Maps and Modes in project seetings.