# CHAPTER 05

* **To Show Some Message in Output Log :-**

UE\_LOG(LogTemp, Warning, TEXT("Begin Play Called!")) ;

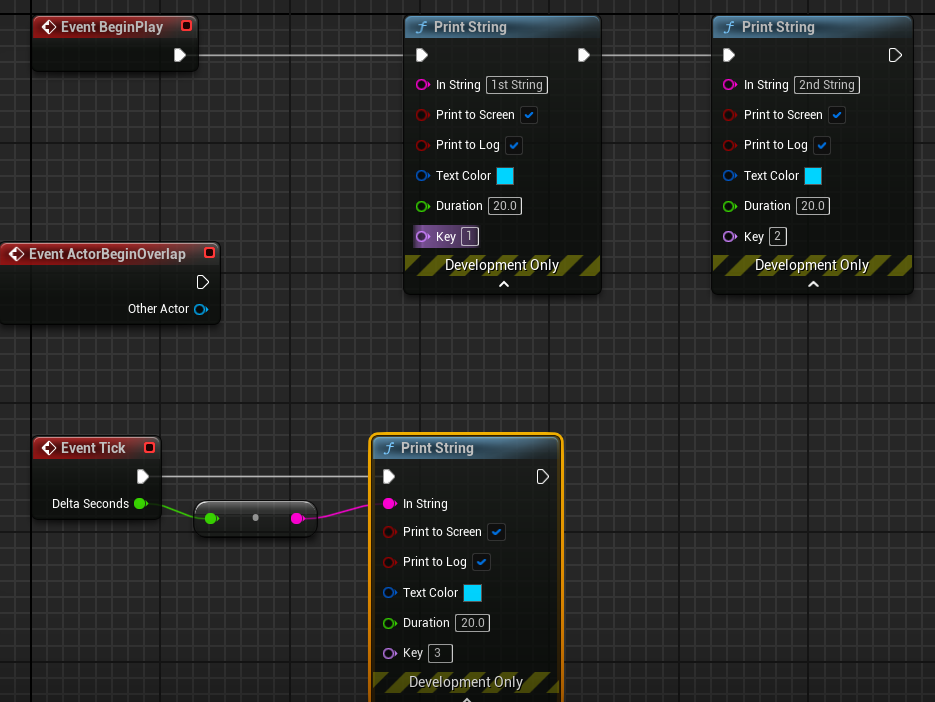
* **To Show Debug Message of Actor On Screen :-**

if (GEngine) // If it is 0 means nullptr and will not run so check pointer 1st.

{

GEngine -> AddOnScreenDebugMessage(1, 60.0f, FColor::Cyan, FString("Testing Item On Screen!"));

}



* **To Debug some format thing like frame seconds OR Actor\_Name :-**

FString Name = GetName() ;

FString Message = FString :: Printf(TEXT("Item Name = %s") , \*Name ) ;

FString Message = FString :: Printf(TEXT("DeltaTime = %f") , DeltaTime ) ;

GEngine -> AddOnScreenDebugMessage(1, 60.0f, FColor::Cyan, Message);

UE\_LOG(LogTemp, Warning, TEXT("Item Name : %s"), \*Name); // OR By Using Log also.

* **To Draw A Debug Sphere Over Actor while gameplay :-**

🡪 Important to include this header file #include<DrawDebugHelpers.h>

UWorld\* World = GetWorld() ;

if ( World )

{

FVector Location = GetActorLocation() ;

DrawDebugSphere( World , Location , 25.0f , 24 , FColor :: Blue , false , 30.f ) ;

}

// OR by using Macro

#define MY\_DEBUG\_SPHERE( Location ) { if( GetWorld( ) ) DrawDebugSphere( GetWorld( ) , Location , 25.0f , 12 , FColor :: Red , true ) ; } // If true so debug sphere will never go away

* **To Draw A Debug Line Along Actor :-**

UWorld\* World = GetWorld();

FVector Location = GetActorLocation();

FVector Forward = GetActorForwardVector() ; // It gets the forward (x-axis as we need to line to scale horizontally) unit vector (which means of 1 unit of UE i.e. 1cm ).

if( World )

{

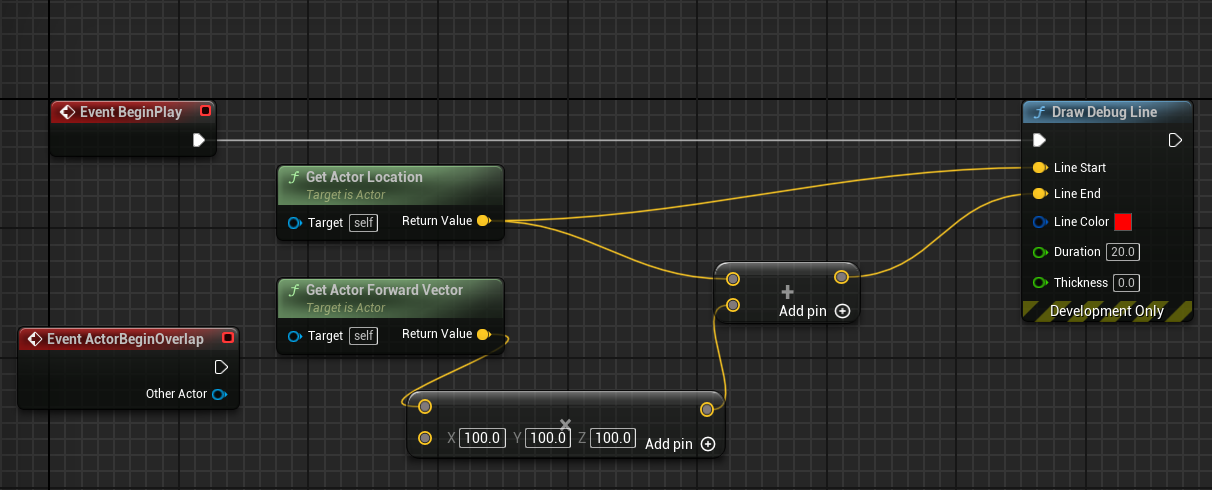
DrawDebugLine(World, Location, Location + 100.0f \* Forward, FColor :: Red, true , -1.0f , 0 , 1.0f ) ;

}

// OR

#define DRAW\_LINE( Start\_Location , End\_Location ) { if( GetWorld( ) ) DrawDebugLine( GetWorld( ) , Start\_Location , End\_Location , FColor :: Red , true , -1.0f , 0 , 1.0f ) ; }

🡪 Macro\_Call DRAW\_LINE(Location, Location + Forward \* 100.0f )



* **To Draw A Debug Point :-**

UWorld\* World = GetWorld();

if (World)

{

DrawDebugPoint(World, Location + Forward \* 100.0f, 15.0f, FColor :: Red, true);

}

// OR

#define DRAW\_POINT( Location ) { if( GetWorld( ) ) DrawDebugPoint( GetWorld() , Location , 15.0f, FColor :: Red, true ); }

DRAW\_POINT( Location + Forward \* 100.0f )

* **To Draw Both Debug Point and Line at the Same Place :-**

// BackSlash allows macro definition to be entered in new line also.

#define DRAW\_VECTOR( Start\_Location , End\_Location ) if ( GetWorld( ) ) \

{ DrawDebugLine(GetWorld(), Start\_Location, End\_Location, FColor :: Red, true, -1.0f, 0, 1.0f); \

DrawDebugPoint(GetWorld(), End\_Location, 15.0f, FColor :: Red, true); \

}

🡪 Macro\_Call DRAW\_VECTOR(Location, Location + Forward \* 100.0f)