**UNREAL ENGINE C++ NOTES**

**Print on screen debug messages in unreal engine using c++.**

if (GEngine)

GEngine->AddOnScreenDebugMessage(1, 10.f, FColor::Black, FString("HELLO WORLD"))

**For float value:**

if (GEngine)

{

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

GEngine->AddOnScreenDebugMessage(1, 10.f, FColor::Black, Message);

}

**For String value:**

if (GEngine)

{

FString Name = GetName();

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

GEngine->AddOnScreenDebugMessage(1, 10.f, FColor::Black, Message);

}

**Adding Mesh To Actor:**

//mesh for actor in .h file

UPROPERTY(EditAnywhere);

UStaticMeshComponent\* mesh; //mesh is name for the component

//creating a mesh component for myactor in cpp file constructor

mesh = CreateDefaultSubobject<UStaticMeshComponent>("MyMesh");

**DEBUG**

**Drawing Debug Sphere :**

#include "DrawDebugHelpers.h"

UWorld\* World = GetWorld();

if (World)

{

FVector Location = GetActorLocation();

DrawDebugSphere(World,Location,25.f,24,FColor::Red,false,30.f );

}

**Drawing Debug Line:**

#include "DrawDebugHelpers.h"

UWorld\* World = GetWorld();

FVector Location = GetActorLocation();

if (World)

{

FVector Forward = GetActorForwardVector();

FVector Vect = Location + Forward \* 100.f; //adding forward vect with cur\_loc\_vector

DrawDebugLine(World, Location, Vect, FColor::Red, false, 60.f );

}

**Draw Debug Points :**

#include "DrawDebugHelpers.h"

UWorld\* World = GetWorld();

if (World)

{

FVector Location = GetActorLocation();

DrawDebugPoint(World, Location + Forward \* 100.f, 15.f, FColor::Red,true);

}

**Actor World Offset / Actor World Rotation:**

AddActorWorldOffset(FVector(1.f, 0.f, 0.f));

AddActorWorldRotation(FRotator(0.f, 90.f, 0.f));

**(TICK ) but Indepndent Of FrameRate:**

void AItems::Tick(float DeltaTime)

{

Super::Tick(DeltaTime);

//movement rate

float MovementRate = 50.f;

//movement rate (cm/s) \* UNits Of Seconds (s /frame ) = Cm/Frame

AddActorWorldOffset(FVector(MovementRate \* DeltaTime, 0.f, 0.f));

DRAW\_SPHERE\_SingleFrame(GetActorLocation());

**SIN Wave Behivaour**

float RunningTime;

//adding deltatime to Running time

RunningTime += DeltaTime;

//here we Add 0.5 to the sin func to make its amplitude lower &&

//Multiply Running time to 5 to make its speed faster

float DeltaZ = 0.5f \* FMath::Sin(RunningTime\* 5.f);

AddActorWorldOffset(FVector(0.f, 0.f, DeltaZ));