**Tanks Vs Zombies**

Create a basic C++ project name it Tank

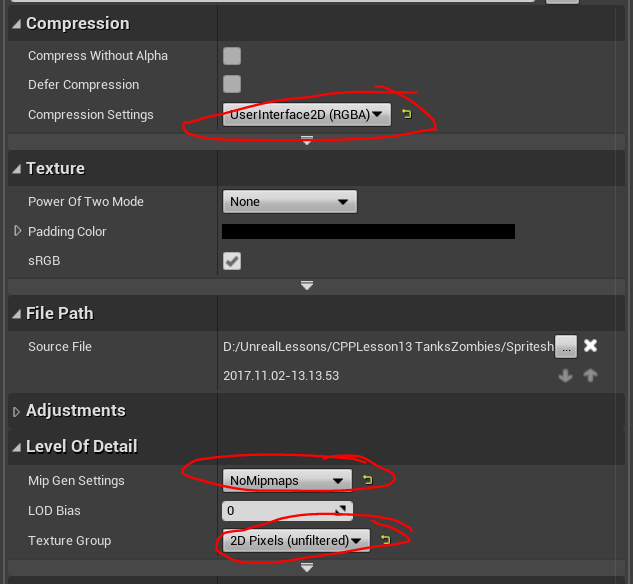
Create a C++ GameMode and name it TankGameMode

Open TankBuild.cs file and add "Paper2D" at the end of PublicDependencyModule as follows

PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject", "Engine", "InputCore", "Paper2D" });

Close VS and right click on uproject file and generate VisualStudioProjectFiles

Import sprite assets Open sprite and change the following.



Rightclick on sprite texture and select SpriteActions/ExtractSprite

Select Grid or Auto and click extract

Name sprites accordingly

Create a Pawn C++ class

Add following code to header file

private:

UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = "Tank", meta = (AllowPrivateAccess = "true"))

UArrowComponent\* TankDirection;

UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = "Tank", meta = (AllowPrivateAccess = "true"))

UPaperSpriteComponent\* TankSprite;

UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = "Turret", meta = (AllowPrivateAccess = "true"))

UChildActorComponent\* ChildTurret;

Need forward decleration

class UArrowComponent;

class UPaperSpriteComponent;

Add the following code to constructor

if (!RootComponent) {

RootComponent = CreateDefaultSubobject<USceneComponent>(TEXT("TankBase"));

}

TankDirection = CreateDefaultSubobject<UArrowComponent>(TEXT("TankDirection"));

TankDirection->AttachTo(RootComponent);

TankSprite = CreateDefaultSubobject<UPaperSpriteComponent>(TEXT("TankSprite"));

TankSprite->AttachTo(TankDirection);

ChildTurret = CreateDefaultSubobject<UChildActorComponent>(TEXT("Turret"));

ChildTurret->AttachTo(TankDirection);

Needs some includes

#include "Tank.h"

#include "Tanks.h"

#include "Components/ArrowComponent.h"

#include "PaperSpriteComponent.h"

Create an Actor class called Turret

Add following to header file

private:

UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = "Turret", meta = (AllowPrivateAccess = "true"))

UArrowComponent\* TurretDirection;

UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = "Turret", meta = (AllowPrivateAccess = "true"))

UPaperSpriteComponent\* TurretSprite;

Add following to C++ file constructor

TurretDirection = CreateDefaultSubobject<UArrowComponent>(TEXT("TurretDirection"));

RootComponent = TurretDirection;

TurretSprite = CreateDefaultSubobject<UPaperSpriteComponent>(TEXT("TurretSprite"));

TurretSprite->AttachTo(TurretDirection);

Create BP version of the tank

Select the tank sprite from the sprite

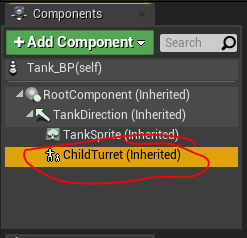
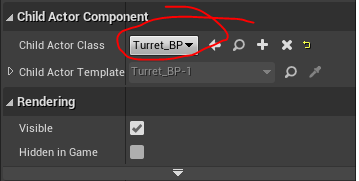
Make rotations to match the level

Create a BP version of the Turret

Select the sprite

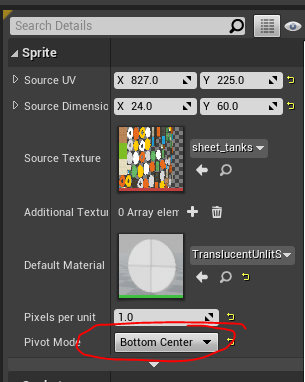
Adjust rotation

Select the Turret\_BP in the child actor in Tank\_BP

To change the pivot of the turret

Open the sprite and select the pivot as follows



We need to add camera. Add the following code to Tank.h

UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = "Tank", meta = (AllowPrivateAccess = "true"))

UCameraComponent\* CameraComponent;

USpringArmComponent\* SpringArm;

Add the following code to Tank.cpp

SpringArm = CreateDefaultSubobject<USpringArmComponent>(TEXT("SpringArm"));

SpringArm->TargetArmLength = 500.0f;

SpringArm->bEnableCameraLag = true;

SpringArm->bEnableCameraRotationLag = false;

SpringArm->CameraLagSpeed = 2.0f;

SpringArm->bDoCollisionTest = false;

SpringArm->bUsePawnControlRotation = false;

SpringArm->AttachTo(RootComponent);

SpringArm->SetWorldRotation(FRotator(-90.0f, 0.0f, 0.0f));

CameraComponent = CreateDefaultSubobject< UCameraComponent>(TEXT("CameraComponent"));

CameraComponent->bUsePawnControlRotation = false;

CameraComponent->AttachTo(SpringArm,USpringArmComponent::SocketName);

CameraComponent->ProjectionMode = ECameraProjectionMode::Orthographic;

CameraComponent->OrthoWidth = 1024.0f;

CameraComponent->AspectRatio = 3.0f / 4.0f;

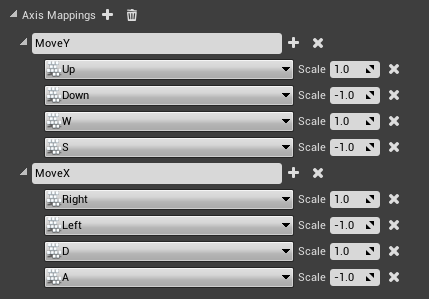
Create a BP version of the TankGameMode

Select TankBP as the default pawn.

Select the TankGameMode\_BP as the game mode

Tank should be the default pawn

Add the following input in project settings



Add following to Tank.h

private:

void MoveX(float AxisValue);

void MoveY(float AxisValue);

Add the following to SetupPlayerInputComponent method in Tank.cpp

PlayerInputComponent->BindAxis("MoveX", this, &ATank::MoveX);

PlayerInputComponent->BindAxis("MoveY", this, &ATank::MoveY);

Add the following Struct to Tank.h file

USTRUCT()

struct FTankInput

{

GENERATED\_BODY()//To recognise by Unreal

public:

UPROPERTY(Visiblenywhere, BlueprintReadOnly, Category = "Tank Input")

FVector2D MovementInput;

void Sanitize();

void MoveX(float AxisValue);

void MoveY(float AxisValue);

private:

FVector2D RawMovementInput;

};

Create a variable of the struct at the end of Tank.h

protected:

UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = "Tank Input")

FTankInput TankInput;

Add the implementation of Move methods as follows

void ATank::MoveX(float AxisValue)

{

TankInput.MoveX(AxisValue);

}

void ATank::MoveY(float AxisValue)

{

TankInput.MoveY(AxisValue);

}

Create implementation of functions in struct in the Tank.cpp file as follows

void FTankInput::Sanitize()

{

MovementInput = RawMovementInput.ClampAxes(-1.0f, 1.0f);

MovementInput = MovementInput.GetSafeNormal();

RawMovementInput.Set(0.0f, 0.0f);

}

void FTankInput::MoveX(float AxisValue)

{

RawMovementInput.X += AxisValue;

}

void FTankInput::MoveY(float AxisValue)

{

RawMovementInput.Y += AxisValue;

}

Add the following in Tick to test

TankInput.Sanitize();

UE\_LOG(LogTemp, Warning, TEXT("Movement : (%f, %f)"), TankInput.MovementInput.X, TankInput.MovementInput.Y);

Output log should give the directions of the key press.

On video 2, 48 minutes

https://youtu.be/eY3SKbVkO2Y?list=PLZlv\_N0\_O1gaz3ydgU5wt6c\_JtJzwXUKW&t=2884

**Reference**

https://www.youtube.com/watch?v=NyXq0Hy9xQs&list=PLZlv\_N0\_O1gaz3ydgU5wt6c\_JtJzwXUKW&ab\_channel=UnrealEngine