

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "GameFramework/Character.h"
7 #include "Components/InputComponent.h"
8 #include "Camera/CameraComponent.h"
9 #include "MyCharacter4.generated.h"
10
11 UCLASS()
12 class JETFIGHTERSIM_API AMyCharacter4 : public ACharacter
13 {
14     GENERATED_BODY()
15
16 public:
17     // Sets default values for this character's properties
18     AMyCharacter4();
19     bool grounded;
20
21     UPROPERTY(EditAnywhere, Category = "Movement")
22     float XAxisSpeed;
23     UPROPERTY(EditAnywhere, Category = "Movement")
24     float YAxisSpeed;
25     UPROPERTY(EditAnywhere, Category = "Movement")
26     float ZAxisSpeed;
27
28
29 protected:
30     // Called when the game starts or when spawned
31     virtual void BeginPlay() override;
32
33 public:
34     // Called every frame
35     virtual void Tick(float DeltaTime) override;
36
37     // Called to bind functionality to input
38     virtual void SetupPlayerInputComponent(class UInputComponent*
39         PlayerInputComponent) override;
40
41
42 private:
43     void HorizontalMove(float value);
44     void ForwardThrust(float value);
45     void Yaw(float value);
46     void HorizontalRot(float value);
47     void VerticalRot(float value);
48     void Roll(float value);
49     void JetEnginePush(float value);
50     void Pitch(float value);
51     void ChangeCamView(void);
```

```
52     void ChangeCamViewBack(void);
53
54
55     void CalculateThrust(float value);
56     void CalculateDrag(void);
57     void CalculateLift(void);
58     void CalculateGravity(void);
59     void CalculateAcceleration(float value);
60
61     void CalculateYaw(void);
62     void CalculatePitch(void);
63     void CalculateRoll(void);
64
65
66
67
68     UPROPERTY(EditAnywhere, Category = "Camera")
69         UCameraComponent* cam;
70
71 private:
72
73     float CurrentVelocity;
74     UPROPERTY(EditAnywhere, Category = "Plane Rotation Variable")
75     float YawRotation;    //Y axis rotation
76     UPROPERTY(EditAnywhere, Category = "Plane Rotation Variable")
77     float PitchRotation;  //Z axis rotation
78     UPROPERTY(EditAnywhere, Category = "Plane Rotation Variable")
79     float RollRotation;   //X axis rotation
80     UPROPERTY(EditAnywhere, Category = "Physics Variable")
81     float Thrust;
82     UPROPERTY(EditAnywhere, Category = "Physics Variable")
83     float Drag;
84     UPROPERTY(EditAnywhere, Category = "Physics Variable")
85     float Lift;
86     UPROPERTY(EditAnywhere, Category = "Physics Variable")
87     float Gravity;
88     UPROPERTY(EditAnywhere, Category = "Physics Variable")
89     float Acceleration;
90     UPROPERTY(EditAnywhere, Category = "Physics Variable")
91     float Mass;
92     UPROPERTY(EditAnywhere, Category = "Physics Variable")
93     float Volume;
94     UPROPERTY(EditAnywhere, Category = "Physics Variable")
95     float Density;
96     UPROPERTY(EditAnywhere, Category = "Physics Variable")
97     float Velocity;
98     UPROPERTY(EditAnywhere, Category = "Physics Variable")
99     float JetReferenceArea;
100    UPROPERTY(EditAnywhere, Category = "Physics Variable")
101    float JetWingArea;
102    UPROPERTY(EditAnywhere, Category = "Physics Variable")
103    float JetFighterDragCoeffecient;
104
```

```
105
106     float XRotation;
107     float YRotation;
108     float ZRotation;
109
110     float speed;
111
112     UPROPERTY(EditAnywhere, Category = "Vector Angle")
113     float XLength;
114     UPROPERTY(EditAnywhere, Category = "Vector Angle")
115     float YLength;
116     UPROPERTY(EditAnywhere, Category = "Vector Angle")
117     float ZLength;
118 };
119
```