

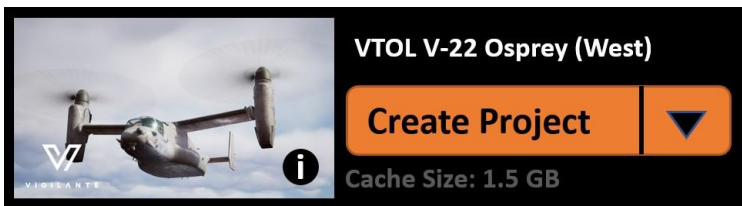
VTOL Bell Boeing V-22 Osprey (West) (with JSBSim support)



Quick Start

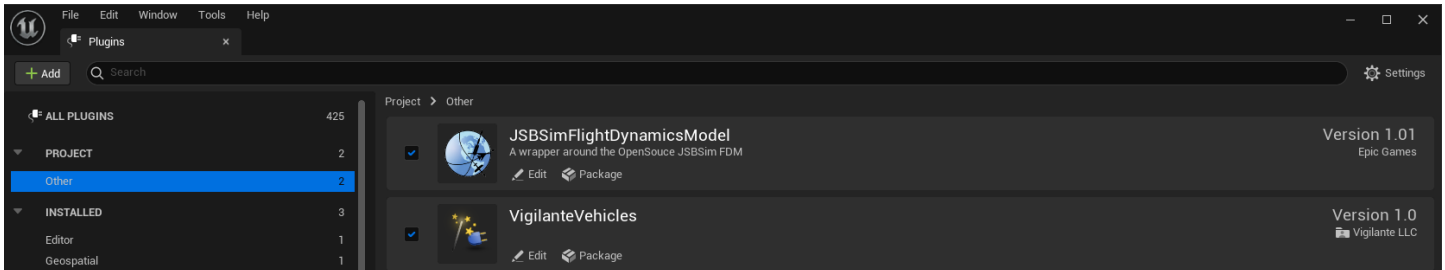
Installation

To install the vehicle from the Epic Games launcher, simply click the **Create Project** button:



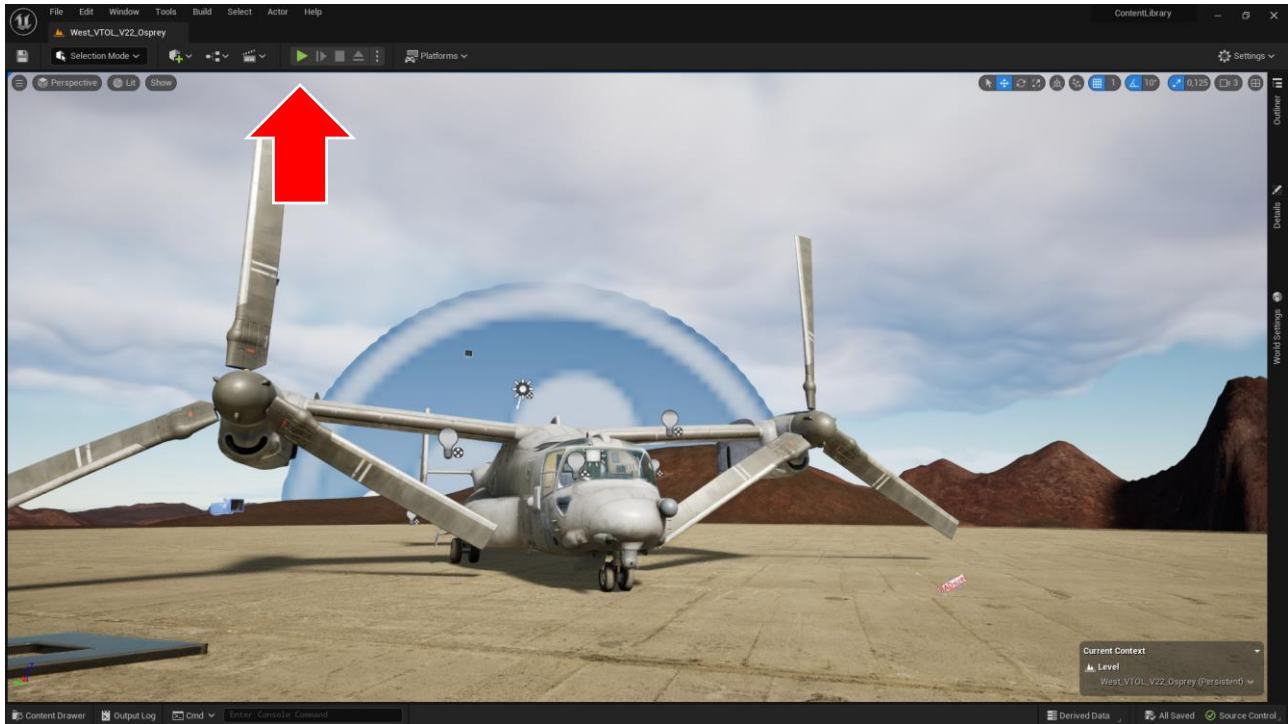
JSBSim requires following Plugins

The following plugins must be installed in the UE5 editor in order for the JSBSim movement to work:



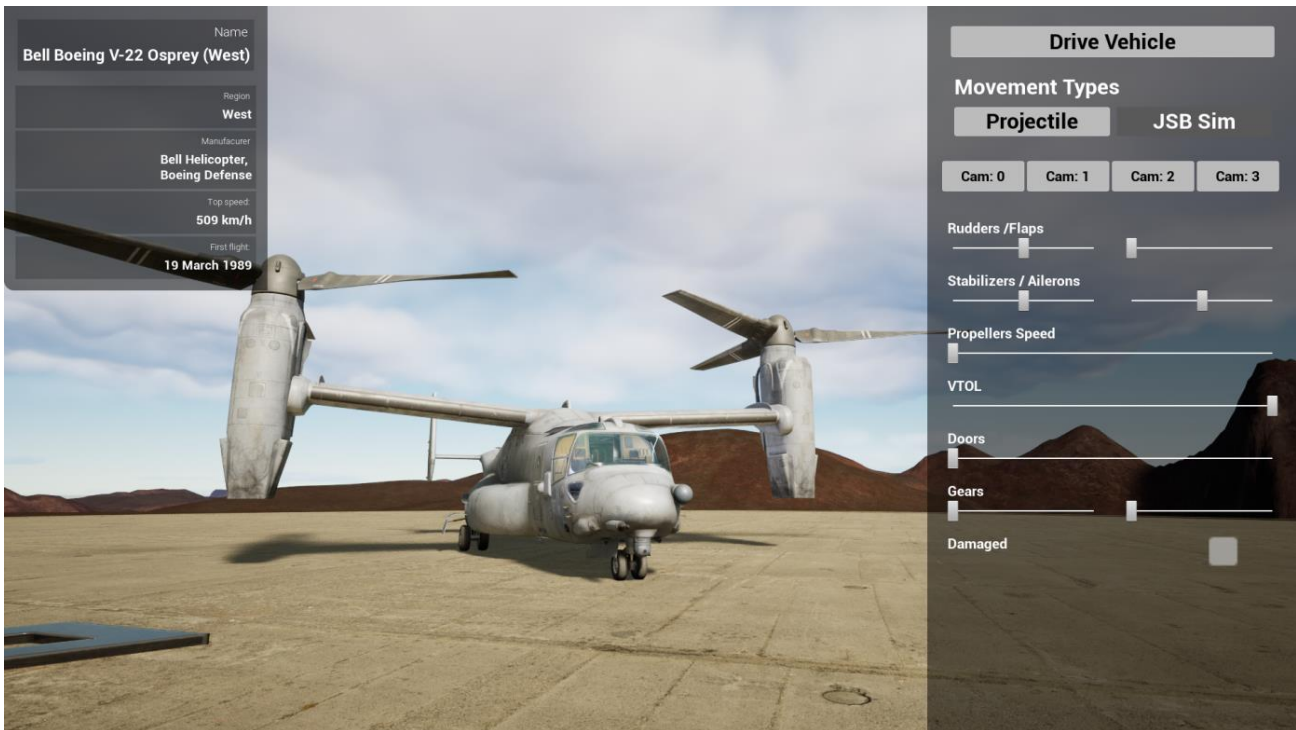
Launch In UE5 Editor

After the project is created, open it. It will load into the map with vehicle in it, as seen below.



Start the Showcase

Press **Play** to start the showcase mode as seen in the previous screenshot. Also, you can use **Alt + P** key shortcut.



Start the Movement

This asset comes with 2 types of flight movement:

1. JSBSim: Movement with JSBSim integrated. Requires the JSBSim plugin to be enabled along with the vehicle's corresponding XML flight model
2. Projectile: Movement based on Unreal Engine's projectile component developed by Vigilante LLC

You can choose between the JSBSim or Projectile flight movement types by selecting either button under the 'Movement Types'.

With the showcase mode active, you can easily switch to the driving mode by pressing the **Drive Vehicle** button in the upper right corner of the screen.



General Information

Asset Description

The Vigilante® West VTOL V22 Osprey asset is rigged and ready to go right from the marketplace! This stand-alone Vehicle is DIS/HLA (RPR FOM) integration ready for import into your network replicated simulation projects. Input configuration is unified across all Vigilante vehicles, so you don't need to worry about corrupting input configuration when importing multiple vehicles.

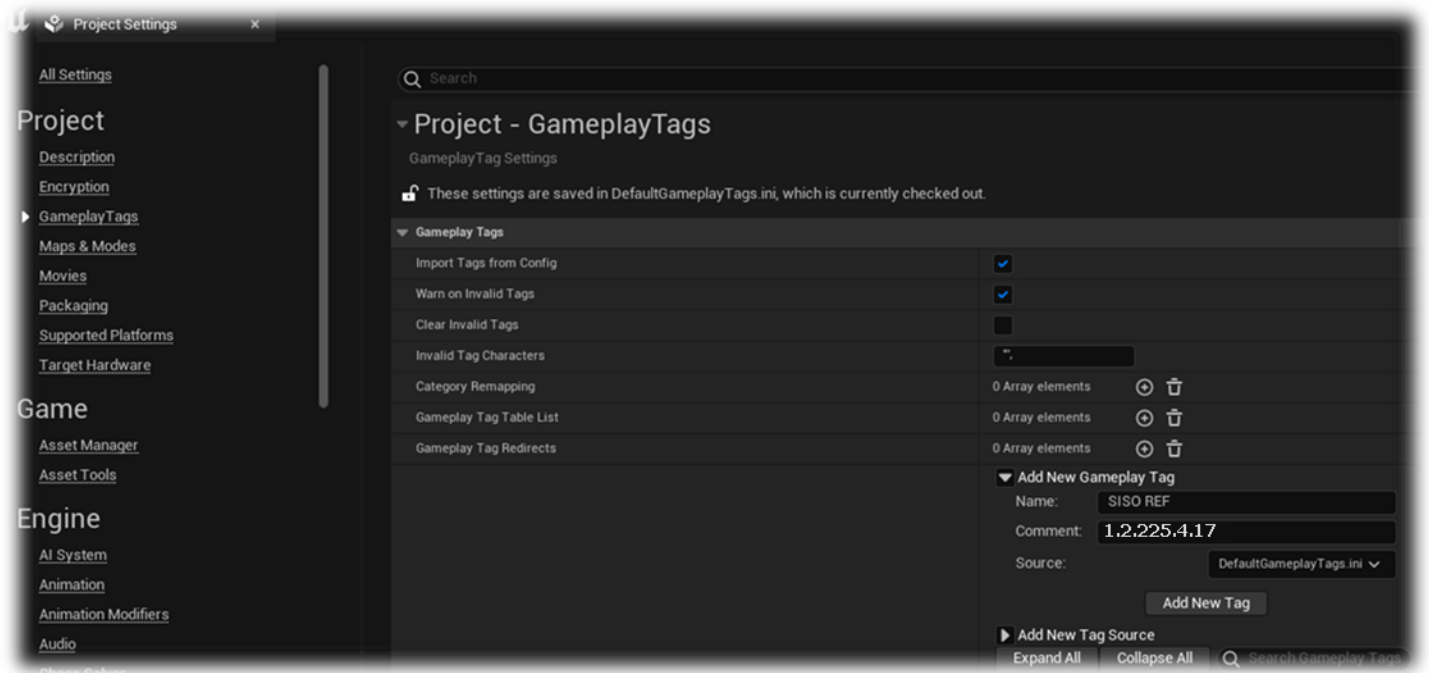
This asset has been cleaned up for import and migration between projects. Instructions explaining how to install the asset along with the setup illustrations are shown below.

This asset has core functionality integrated into **JSBSim**.

SISO Identification

This vehicle is compliant with SISO Entity Reference No.: **1.2.225.4.17**

The SISO REF Number has been added to the Project Settings under **GameplayTags**. The Tag Name is SISO REF, and the Tag Comment includes the SISO Entity Reference No. listed above.



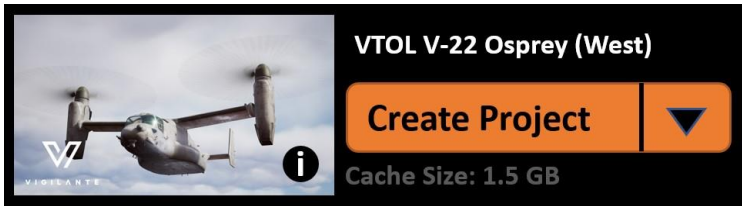
New to Unreal Engine?

Helpful resources for those who may be new to using Unreal Engine:

- [Introduction to working in Unreal Engine 5](#)
- [Programming and Scripting in Unreal Engine](#)

Installation

To install the vehicle from the Epic Games launcher, simply click, Create Project:



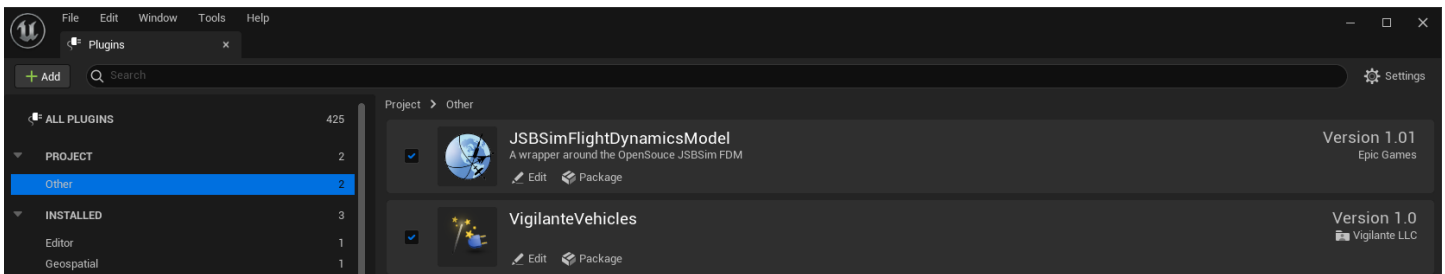
Once opened, the Vigilante created content is in the **Content/VigilanteMovement** folder. Subfolders include folders for the vehicles, along with a shared folder. The shared folder consists of assets to be shared by all vehicles, from the master vehicle parent blueprint to common FX shared by all Vigilante vehicles.

JSBSim

JSBSim is an open-source Flight Dynamics Model (FDM) software library that models the flight dynamics of an aerospace vehicle.

The FDM is essentially the physics/mathematical model that defines the movement of an aircraft, rocket, etc., under the forces and moments applied to it using the various control mechanisms and from the forces of nature.

JSBSim requires the following plugins to be installed in order for it to function properly:



Our asset has a core movement functionality integrated with JSBSim.

A proper JSBSim FDM of the vehicle must exist within JSBSim library as an XML file. Vigilante LLC does not provide actual airplane FDM.

Please choose a FDM from the existing models in the JSBSim library or use an airplane XML template to create one.

Sources:

<http://jsbsim.sourceforge.net/>

<https://github.com/JSBSim-Team/jsbsim>

Vigilante Customer Support

The Vigilante customer support can be reached through our uemarketplace@vigilante.us support team e-mail. Please reach out with any questions regarding our product or asset configuration.

For JSBSim inquires please contact: <https://github.com/JSBSim-Team/jsbsim/discussions>

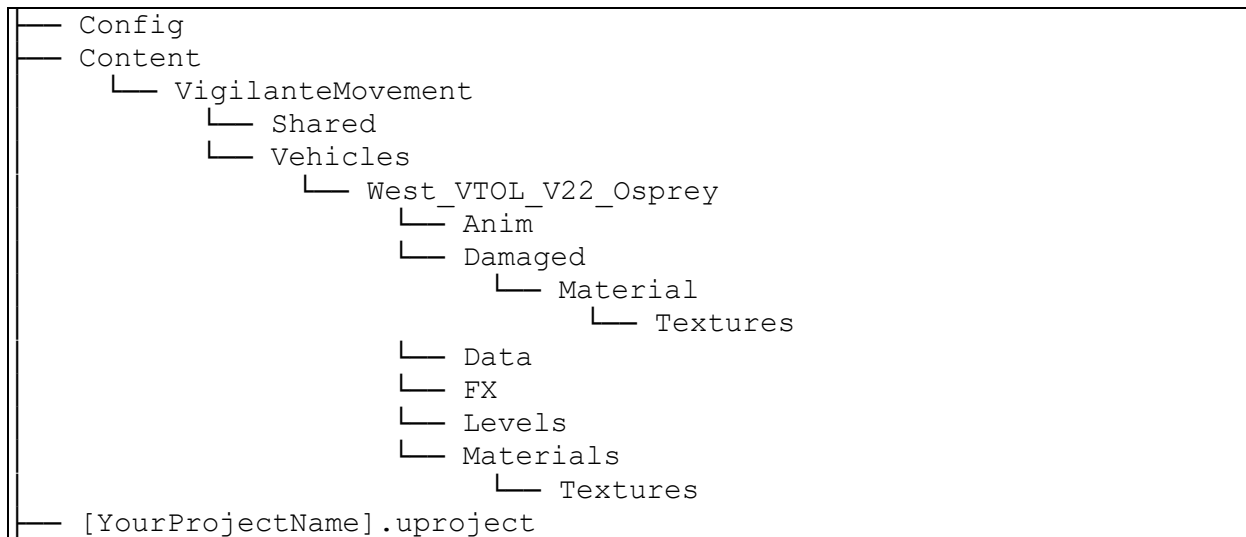
License

Licensed for use only with Unreal Engine-based products.

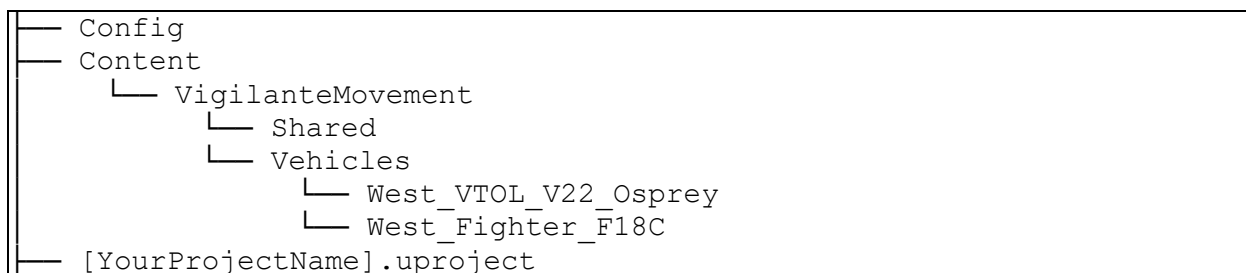
Project Structure

Folder Structure

Once the project is created and you decide to migrate this Vigilante asset into your project, the final folder structure in your project will be as follows:



If you have installed multiple Vigilante assets, your folder structure might look this:



This folder configuration will let you take advantage of resource sharing across assets from the Vigilante Movement Library (e.g., shared particle effects like dust, smoke, fire, etc.).

Asset Configuration Details

When you hit “Play” in the Unreal project, a user interface (UI) will appear on the right-side of your screen that allows you to manipulate the options available for configuration.

This asset comes with three separate Actor blueprints.

The Blueprint with the “**Showcase**” suffix includes a Demo User Interface with sliders and toggles for the various vehicle functions. (e.g., Open / close hatches, fire weapon, etc.). This is designed for you to experiment with how the various components of the asset function and are hooked-up.

To use our vehicle as an example, this project would include the following two Blueprints for you to interact with:

BP_West_VTOL_V22_Osprey – Vehicle with movement functionality.

BP_West_VTOL_V22_Osprey_Wrecked – Vehicle with wrecked (damaged) functionality.

BP_West_VTOL_V22_Osprey_Showcase – Vehicle with showcase/demo functionality.

The terms wrecked and damaged are used interchangeably.

When you feel like you have a good understanding of the asset and are ready to migrate it into your Unreal project, pick the actor blueprint without the “_Showcase” suffix (located in the same folder), and migrate it into your project. This way you will migrate just the essentials needed for your use (no user interface with sliders or toggles, etc.).

Materials

Vehicle Materials

MI_West_VTOL_V22_Osprey_Main_01

Material	File Name	Channel
Base Color	T_West_VTOL_V22_Osprey_ext1_D	RGB
Metallic	T_West_VTOL_V22_Osprey_ext1_ARM	B
Roughness	T_West_VTOL_V22_Osprey_ext1_ARM	G
Emissive	T_West_VTOL_V22_Osprey_ext1_E	RGB
Normal	T_West_VTOL_V22_Osprey_ext1_N	RGB
Ambient Occlusion	T_West_VTOL_V22_Osprey_ext1_ARM	R

MI_West_VTOL_V22_Osprey_Main_02

Material	File Name	Channel
Base Color	T_West_VTOL_V22_Osprey_ext2_D	RGB
Metallic	T_West_VTOL_V22_Osprey_ext2_ARM	B
Roughness	T_West_VTOL_V22_Osprey_ext2_ARM	G
Emissive	T_West_VTOL_V22_Osprey_ext2_E	RGB
Normal	T_West_VTOL_V22_Osprey_ext2_N	RGB
Ambient Occlusion	T_West_VTOL_V22_Osprey_ext2_ARM	R

MI_West_VTOL_V22_Osprey_Glass

Material	File Name	Channel
Base Color	T_West_VTOL_V22_Osprey_glass_D	RGB
Metallic	T_West_VTOL_V22_Osprey_glass_ARM	B
Roughness	T_West_VTOL_V22_Osprey_glass_ARM	G
Ambient Occlusion	T_West_VTOL_V22_Osprey_glass_ARM	R

MI_West_VTOL_V22_Osprey_Main_Cockpit

Material	File Name	Channel
Base Color	T_West_VTOL_V22_Osprey_Cockpit_D	RGB
Metallic	T_West_VTOL_V22_Osprey_Cockpit_ARM	B
Roughness	T_West_VTOL_V22_Osprey_Cockpit_ARM	G
Normal	T_West_VTOL_V22_Osprey_Cockpit_N	RGB
Ambient Occlusion	T_West_VTOL_V22_Osprey_Cockpit_ARM	R

Vehicle Damaged Materials

MI_West_VTOL_V22_Osprey_Damaged_01

Material	File Name	Channel
Base Color	T_West_VTOL_V22_Osprey_ext1_Damaged_D	RGB
Metallic	T_West_VTOL_V22_Osprey_ext1_Damaged_ARM	B
Roughness	T_West_VTOL_V22_Osprey_ext1_Damaged_ARM	G

Normal	T_West_VTOL_V22_Osprey_ext1_Damaged_N	RGB
Ambient Occlusion	T_West_VTOL_V22_Osprey_ext1_Damaged_ARM	R

MI_West_VTOL_V22_Osprey_Damaged_02

Material	File Name	Channel
Base Color	T_West_VTOL_V22_Osprey_ext2_Damaged_D	RGB
Metallic	T_West_VTOL_V22_Osprey_ext2_Damaged_ARM	B
Roughness	T_West_VTOL_V22_Osprey_ext2_Damaged_ARM	G
Normal	T_West_VTOL_V22_Osprey_ext2_Damaged_N	RGB
Ambient Occlusion	T_West_VTOL_V22_Osprey_ext2_Damaged_ARM	R

MI_West_VTOL_V22_Osprey_Cockpit_Damaged

Material	File Name	Channel
Base Color	T_West_VTOL_V22_Osprey_Cockpit_Damaged_D	RGB
Metallic	T_West_VTOL_V22_Osprey_Cockpit_Damaged_ARM	B
Roughness	T_West_VTOL_V22_Osprey_Cockpit_Damaged_ARM	G
Normal	T_West_VTOL_V22_Osprey_Cockpit_Damaged_N	RGB
Ambient Occlusion	T_West_VTOL_V22_Osprey_Cockpit_Damaged_ARM	R

Vehicle Material Graph

In the Vigilante Movement Content Library, we use two master materials to share shader code across all our assets. Diagrams for both master materials are available in the diagram below.

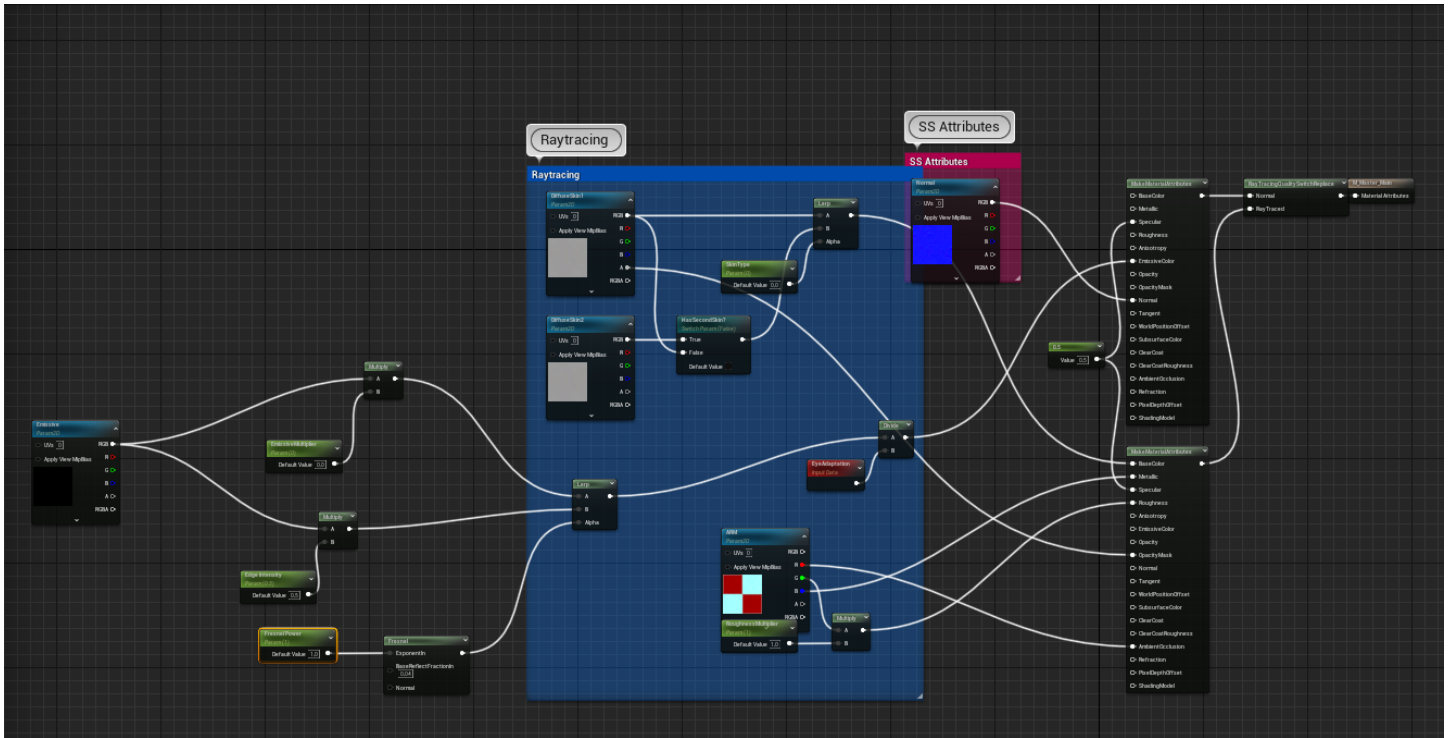
Each vehicle then uses its own Material Instance with vehicle specific textures and settings. This allows for a cleaner and a more convenient asset management.

Notable master material parameters:

- **SkinType** – Toggles between available skin textures if the asset supports different skins.
- **EmissiveMultiplier** – Controls texture emissivity. Most of the vehicles has a mask for the emissive texture.

For the ARM texture (which stands for Ambient Occlusion / Roughness / Metallic), the RGB channels are mapped to the associated channels of the Vigilante Asset Material as shown below (red for Ambient Occlusion, green for Roughness, and blue for Metallic).

The master material for all vehicles in the Vigilante Movement Content Library supports Raytracing by using different paths for higher Raytracing quality settings.



The Vigilante asset has a **'damaged'** (wrecked) actor. A preview of this state is available in the Blueprint actor with the suffix **'_Wrecked'**.

Starting the damaged mode in showcase preview will show damaged model and damaged smoke will start spawning.

Showcase and Movement Mode

Each Vigilante asset comes with a showcase and movement mode.

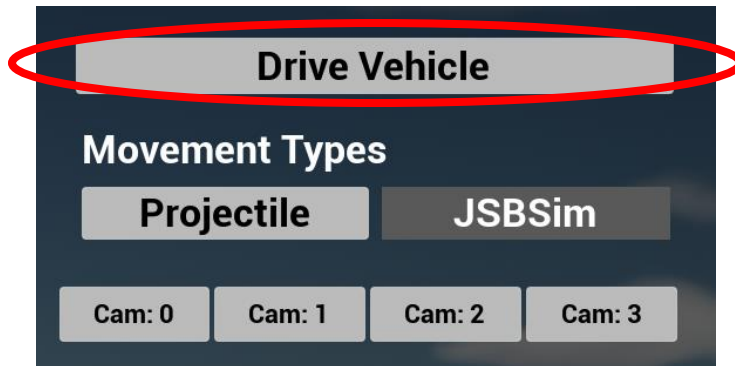
Showcase Mode

The Showcase mode is automatically launched by default when you press **Play** in the editor. The Showcase mode allows you to preview the vehicle's functionalities before taking it for a ride. We understand that some of the functionalities are more visible when the vehicle is standing still on a platform rather than in movement.



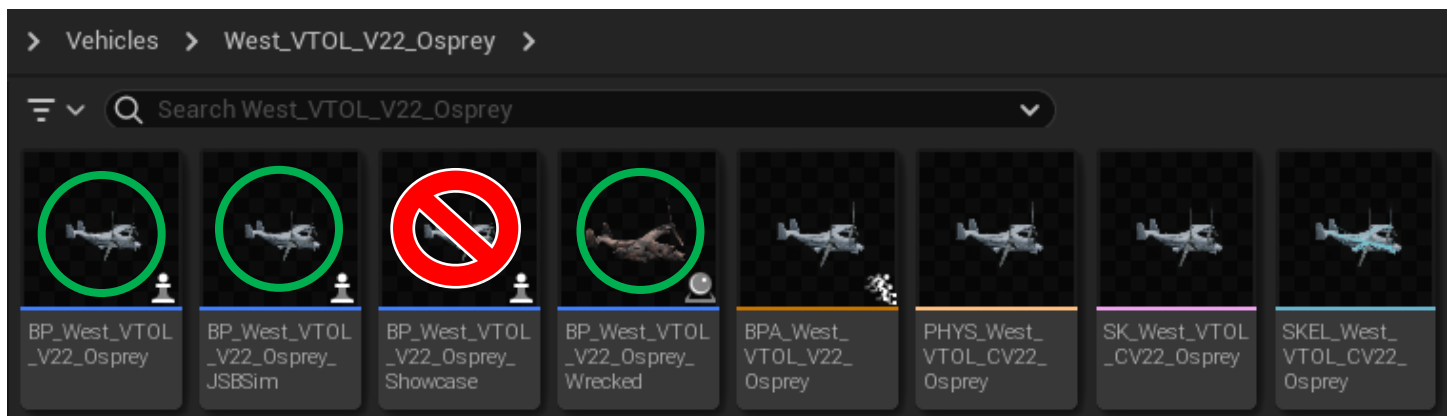
Movement Mode

You can select from various movement types. The Movement mode is by default triggered by pressing the **Drive Vehicle** button in the Showcase.



Switch to Movement Mode

In West_VTOL_V22_Osprey folder you can see three important blueprints. The first is called BP_West_VTOL_V22_Osprey and it contains the projectile movement functionality. The second one is called BP_West_VTOL_V22_Osprey_JSBSim and contains the JSBSim movement. The third one is called BP_West_VTOL_V22_Osprey_Showcase and it stores the showcase settings. If you do not want the Showcase mode to pop up and you want only the Movement mode active, just remove the showcase vehicle from the level and add the Blueprint Actor without any suffix.



Base Vehicle

We have created a **BP_BaseVehicle** blueprint class wherein all the shared functionalities are placed. All Vigilante vehicles use this blueprint as their main parent. This allows us to keep vehicles clean and removes duplication in the vehicle logic.

For example, damaged mode is a functionality supported in all our vehicles and as such it is placed in the parent class.

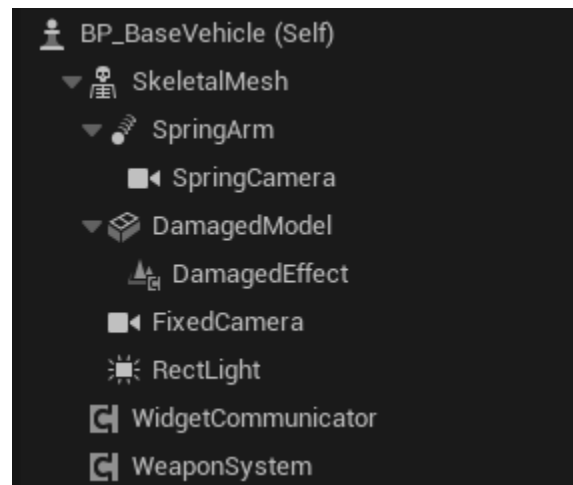
Functionalities

The table below describes the functions in the **BP_BaseVehicle**.

Function Name	Function Description
SetupMaterialInstances	Creates material instances for each material on skeletal mesh
SetSkinType	Sets skin parameter for vehicle materials
SetLightsIntensity	Sets light intensity parameter for vehicle materials
SetDamagedMode	Toggles mesh visibility from wreck to original mesh
SetEngineState	Sets new engine state
GetEngineState	Returns current engine state
Respawn	Respawns vehicle by spawning new pawn and possessing it
FireCountermeasures	Activates counter measures
GetPlayerController	Returns player controller controlling this pawn
SetActiveCamera	Switches between primary and secondary camera
SetInputEnabled	Enables or disables input for this actor
Get Aim	Sets Aim of Turret/Gun based on camera rotation
ToggleInteriorLight	Sets Interior Lights on/off if available

Hierarchy

Each of our vehicles starts with the same component hierarchy which can be modified in children.



BP Hierarchy

- **BP_BaseVehicle** – The parent BP. Both movements share common functionalities within this BP.
- **JSBSim:**
 - **BP_AirVehicle_JSBSim** – BP for core functionalities within JSBSim
 - **BP_Airplane_JSBSim** – BP for airplane-specific functionalities in JSBSim
 - **BP_West_VTOL_V22_Osprey_JSBSim** – BP for specific vehicle asset using JSBSim movement
- **Projectile Movement:**

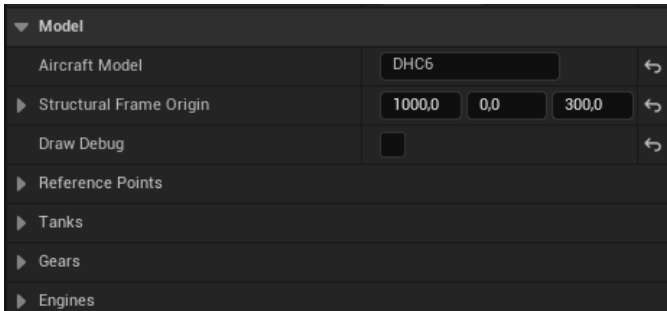
- **BP_BaseAirplaneVehicle** – BP for airplanes’ functionalities
- **BP_West_VTOL_V22_Osprey** – BP for specific vehicle asset
- **Damaged:**
 - **BP_West_VTOL_V22_Osprey_Wrecked** – BP for damaged versions
- **Showcase:**
 - **BP_West_VTOL_V22_Osprey_Showcase** – BP for demonstration purposes only using the Showcase mode

All the Blueprints’ functionalities are described in their own chapters.

JSBSim Movement – BP_West_VTOL_V22_Osprey_JSBSim

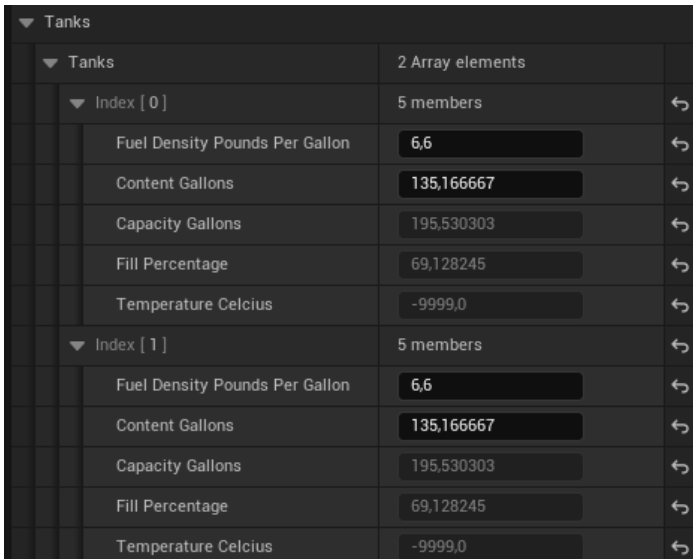
Basic Flight Setup

To setup new JSBSim FMD values, use the following variables from the JSBSimMovementComponent:



- Aircraft Model – used for the name of the XML file from the JSBSim library. You can find the available flight models on JSBSim’s Git Hub or in the project folder: /Plugins/JSBSimFlightDynamicsModel/Resources/JSBSim/aircraft. Alternatively, you can create and use our own FDM XML files based on the aircraft_template.xml file.
- Structural Frame Origin – can adjust the skeletal mesh to reference points of the aircraft.
- Reference Points – will be automatically set from XML flight model.

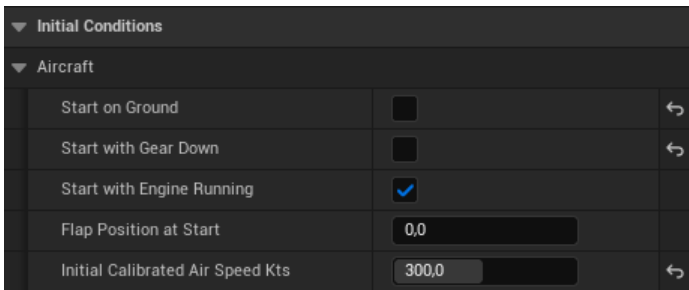
Fuel tanks’ information will be set automatically from the XML file too. Just add values for actual content and density.



The aircraft’s gears will be set automatically from the XML file.



You can set initial conditions here:



For more detailed information, please refer to the [JSBSim and JSBSim Plugin documentation](#).

Events:

BeginPlay	Sets animation instances for plane fuselage and canopy. Sets proper values for exhaust nozzles and afterburner
Event Tick	Updates variables for ABP
Event Set Gear	Sets landing gear position in ABP
RetractFrontWheel	Sets front gear position in ABP
RetractRearLWheel RetractRearRWheel	Sets rear gears position in ABP
Event Set Yaw	Sets yaw on rudders and pilot pedals in ABP
Event Set Roll	Set roll on ailerons and pilot stick in ABP
Event Set Pitch	Sets pitch on stabilizers and pilot stick in ABP
Event Set Throttle	Sets throttle on exhausts and pilot left hand lever in ABP
SetExhaustNozzles	Sets exhaust nozzles size in ABP
EventSetFlaps	Sets flaps position in ABP
EventSetAirBrake	Sets airbrakes position in ABP
SetFrontWheelSteering	Sets front wheel steering rotation in ABP
SetWheelsSpeed	Sers landing gear wheels velocity in ABP
EventSetCanopy	Sets canopy position in ABP (not supported by JSBSim)

Projectile Movement – BP_West_VTOL_V22_Osprey

Basic Flight Setup

To quickly setup new values, use the following variables from the Basic category:



- Top speed: The vehicle’s top speed without an afterburner.
- Top Speed AF: The vehicle’s top speed with an afterburner.
- Has Afterburner: Can be checked if the aircraft has an afterburner.
- Is VTOL: Can be checked if the aircraft supports vertical takeoff and landing (VTOL).

Basic		
Is Flying	<input checked="" type="checkbox"/>	
Top Speed	15230,0	↩
Top Speed AF	15230,0	↩
Has Afterburner	<input type="checkbox"/>	↩
Is VTOL	<input checked="" type="checkbox"/>	
Hide HMDOn Secondary Cam	<input type="checkbox"/>	
Invert Pitch	<input checked="" type="checkbox"/>	↩

Please refer to [BP BaseAirplaneVehicle](#) flight model variables for detailed description

Advanced Flight Setup

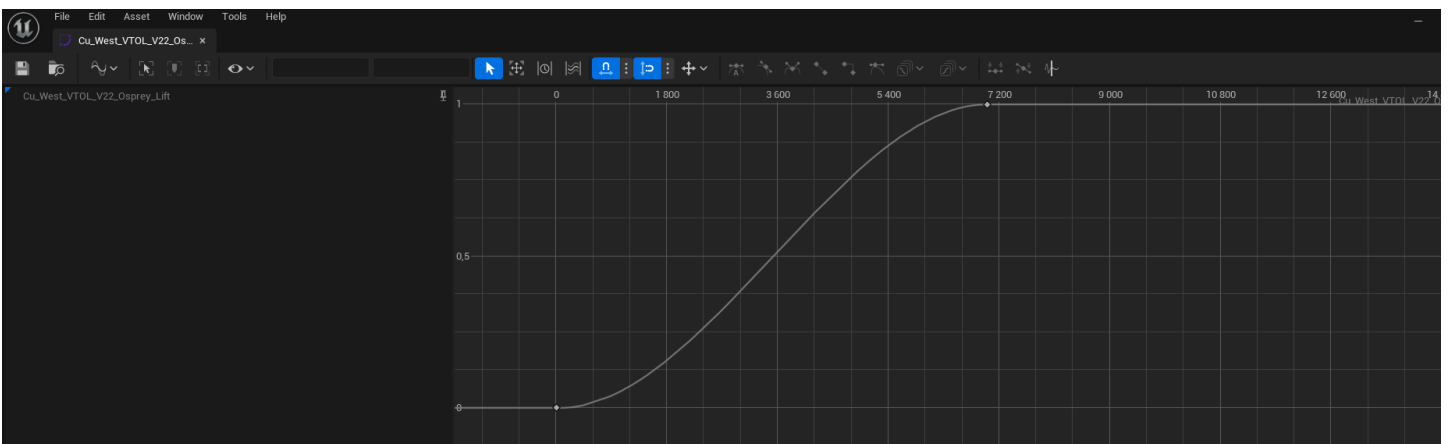
You can setup more advanced behavior by following these variables in the Extended category.

Extended		
Yaw Max Turn Rate	0,5	
Roll Max Turn Rate	2,0	↩
Pitch Max Turn Rate	1,0	
Pitch Turn Rate Curve	 Cu_West_VTOL_	↩
Throttle Rate	0,005	
Acceleration Rate	100,0	
Max VTOLSpeed	2500,0	
Lift Curve	 Cu_West_VTOL_	↩
Flaps Lift Index	0,05	
Gears Drag Index	0,0005	
Air Brake Drag Index	0,001	
Terrain Detection Radius	300,0	
Terrain Detection Offset	0,0 0,0 -2000,0	
Dust Multiplicator	1,0	

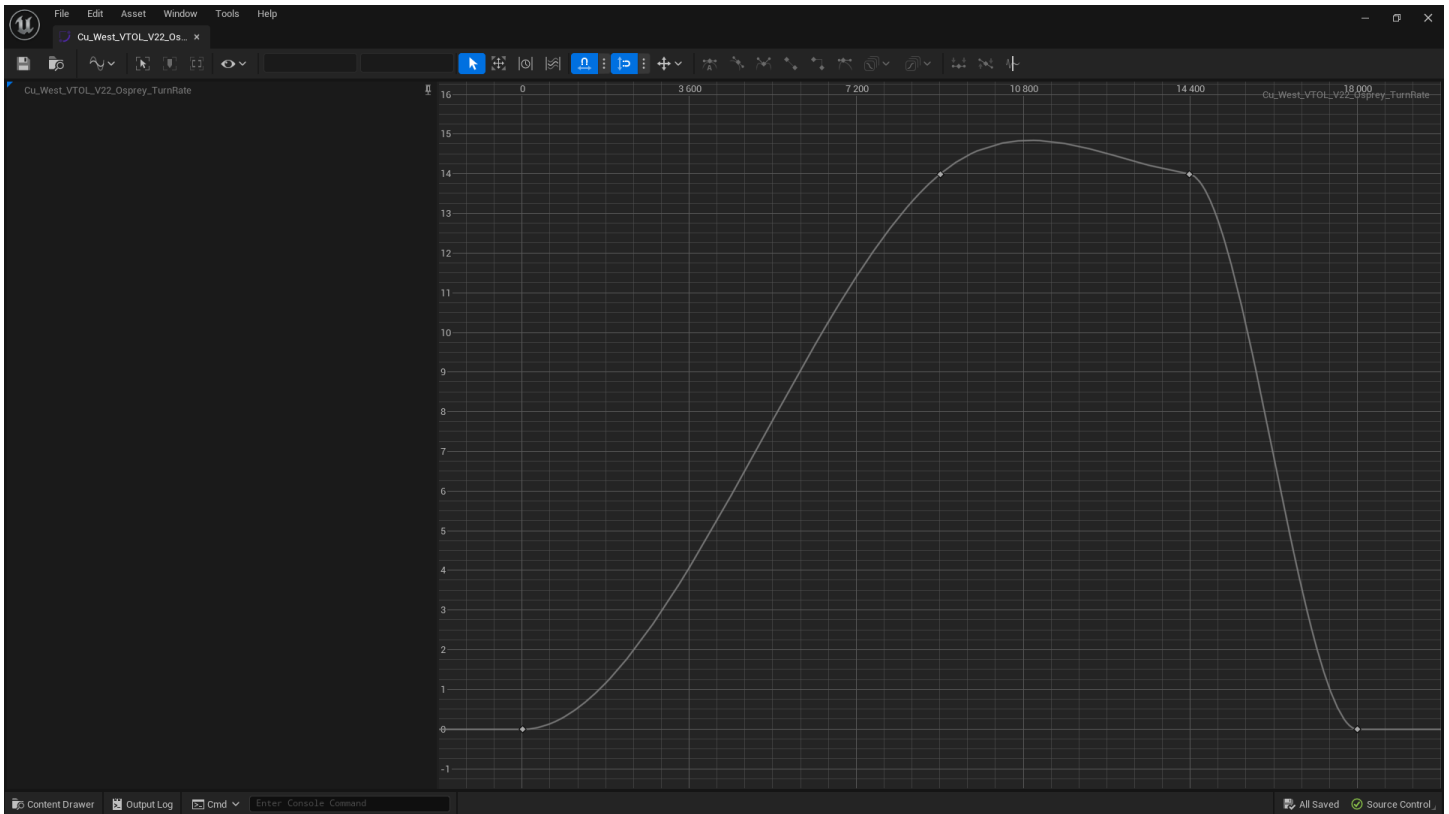
Please refer to [BP BaseAirplaneVehicle](#) flight model variables for a detailed description.

We are using the following curves for the airplane's lift and turn rate.

Cu_West_VTOL_V22_Osprey_Lift:



Cu_West_VTOL_V22_Osprey_TurnRate:



Cockpit Camera Setup:

Use the following settings to set the switch between the pilot and gunner seats using a “fixed” camera for interiors. The Zoomed camera is of the gunner and the un-zoomed camera is for the pilot seat.

Camera Settings			
Cockpit Camera Zoomed Transform			
Location	655,665	-66,0	251,748
Rotation	0,0	-10,0	0,0
Scale	1,0	1,0	1,0
Cockpit Camera Un Zoomed Transform			
Location	655,665	66,0	251,748
Rotation	0,0	-10,0	0,0
Scale	1,0	1,0	1,0
Cockpit Camera Zoomed FOV	130,0		
Cockpit Camera Un Zoomed FOV	130,0		

Countermeasures

Countermeasures	
Countermeasures NS	NS_DistractionF
Optional Actor To Spawn	None

Events:

BeginPlay	Sets animation instances for plane fuselage and canopy. Sets proper values for exhaust nozzles and afterburner
Event Set Gear	Sets landing gear position
RetractFrontWheel	Sets front gear position
RetractRearLWheel RetractRearRWheel	Sets rear gears position
Event Set Yaw	Sets yaw on rudders and pilot pedals
Event Set Roll	Set roll on ailerons and pilot stick
Event Set Pitch	Sets pitch on stabilizers and pilot stick
Event Set Throttle	Sets throttle on exhausts and pilot left hand lever
EventSetFlaps	Sets flaps position
EventSetCanopy	Sets canopy position
EventSetVTOLRatio	Sets VTOL ratio for thrusters

General Setup

Weapons Setup:

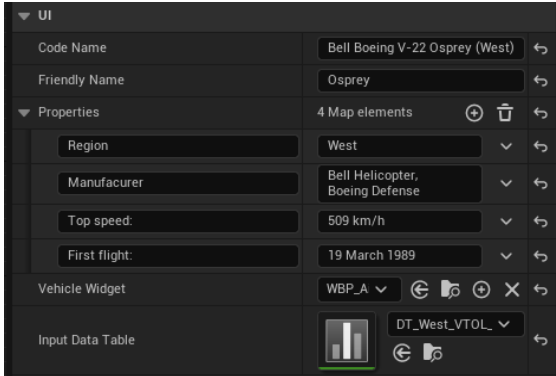
There are no active weapons for this asset.

Widgets setup

This asset uses the Communication Widget component. This includes:

- Description Widget
- Inputs Widget

Please refer to [BP_CommunicationWidget](#) for detailed usage description

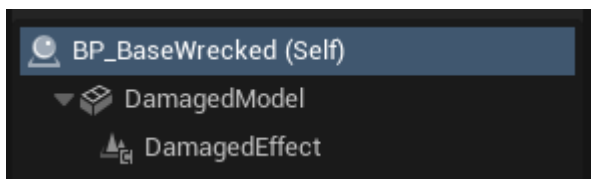


The available mapping of inputs is specified in DT_West_VTOL_V22_Osprey_InputsMapping.

Row Name	Show	FriendlyName	InputName	isAxis	AxisScale	
1	Respawn	True	Respawn	Respawn	False	0.000000
2	ThrustPlus	False	Throttle +	Thrust	True	1.000000
3	ThrustMinus	False	Throttle -	Thrust	True	-1.000000
4	MoveForward	True	Throttle +	MoveForward	True	1.000000
5	MoveBackward	True	Throttle -	MoveForward	True	-1.000000
6	MoveUp	False	Up	MoveUp	True	1.000000
7	MoveDown	False	Down	MoveUp	True	-1.000000
8	MoveRight	True	Yaw +	MoveRight	True	1.000000
9	MoveLeft	True	Yaw -	MoveRight	True	-1.000000
10	RollPlus	True	Roll +	Roll	True	1.000000
11	RollMinus	True	Roll -	Roll	True	-1.000000
12	PitchPlus	True	Pitch +	Pitch	True	1.000000
13	PitchMinus	True	Pitch -	Pitch	True	-1.000000
14	VTOLPlus	True	VTOL Ratio +	Pitch	True	1.000000
15	VTOLMinus	True	VTOL Ratio +	Pitch	True	-1.000000
16	LookAround	True	Look Around	LookAround	True	1.000000
17	LookUp	True	Look Up/Down	LookUp	True	0.500000
18	Fire	False	Fire	Fire	False	0.000000
19	CycleWeapons	False	Cycle Weapons	CycleWeapons	False	0.000000
20	Countermeasure	True	Flares	Countermeasures	False	0.000000
21	SwitchCamera	True	Switch Camera	ToggleCamera	False	0.000000
22	Zoom+	True	Switch Seat+	Zoom +	False	0.000000
23	Zoom-	True	Switch Seat-	Zoom -	False	0.000000
24	Boost	False	Afterburner	Boost	False	0.000000
25	Eject	False	Eject	Eject	False	0.000000
26	Handbrake	False	Airbrake	Handbrake	False	0.000000
27	Gears	True	Gears	Gears	False	0.000000
28	Flaps	True	Flaps	Flaps	False	0.000000
29	Hook	False	Hook	Hook	False	0.000000
30	Canopy	True	Open Doors	Open Canopy	False	0.000000
31	FoldWings	False	Fold Wings	Fold Wings	False	0.000000
32	Drop	False	Drop Fuel Tank	Drop	False	0.000000
33	Spoilers	False	Spoilers	Spoilers	True	1.000000
34	SteerRight	False	Steer Right	Steer	True	-1.000000
35	SteerLeft	False	Steer Left	Steer	True	1.000000
36	LeftBrake	False	Left Brake	LeftBrake	True	1.000000
37	RightBrake	False	Right Brake	RightBrake	True	1.000000
38	CenterBrake	False	Center Brake	CenterBrake	True	1.000000
39	ParkingBrake	False	Parking Brake	ParkingBrake	False	0.000000
40	TrimYawUp	False	Trim Yaw +	TrimYawUp	False	0.000000
41	TrimYawDown	False	Trim Yaw -	TrimYawDown	False	0.000000
42	TrimRollUp	False	Trim Roll +	TrimRollUp	False	0.000000
43	TrimRollDown	False	Trim Roll -	TrimRollDown	False	0.000000
44	TrimPitchUp	False	Trim Pitch +	TrimPitchUp	False	0.000000
45	TrimPitchDown	False	Trim Pitch -	TrimPitchDown	False	0.000000

BP_West_VTOL_V22_Osprey_Wrecked / BP_BaseWrecked

This is the Blueprint actor that spawns/replaces the normal vehicle with a wrecked/damaged version. It includes smoke VFX.



BP_West_VTOL_V22_Osprey_Showcase

The Showcase is a demonstration of the asset’s functionalities. You can use inputs such as sliders, buttons, and checkboxes to demonstrate the vehicle’s functions on the showcase level West_VTOL_V22_Osprey.

HUD/HMD Overview

The airplane uses two UI Widgets for:

- Head Mounted Display HMD: Shown on the external view, simulating projection on the pilot’s helmet.

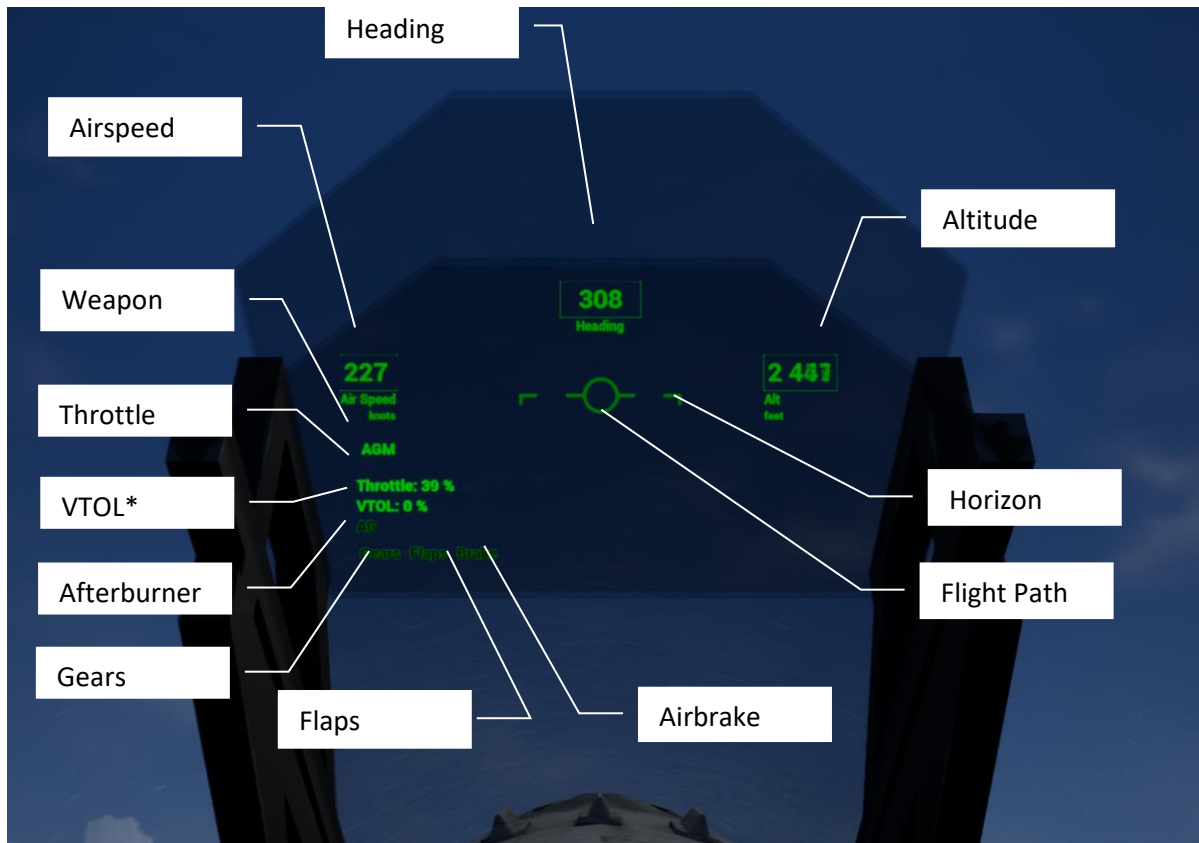
- Head Up Display HUD: Shown on the airplane's HUD projector inside the cockpit.

HMD reference is set in the [BP Communication Widget](#) component.

HUD reference is set in the specific airplane child actor widget component AiplaneNoseHUD.

The functionality of both is the same, they only differ in layout size.

HMD/HUD Layout



*VTOL if available

HMD/HUD Widgets

UI_AirVehicle_Cockpit_HUD

UI_AirVehicle_HMD

Variables:

OwnerRef	Widget owner reference
OwnerJSBSim_Ref	Widget owner reference for JSBSim movement vehicles
AltitudeUnitsType	Enum for altitude units
SpeedUnitsType	Enum for speed units

Functions:

Get Air Speed Text	Updates air speed text value
Get Altitude Text	Updates altitude text value
Get Throttle Text	Updates throttle text
Get VTOL Ratio Text	Updates VTOL ratio text
Get Heading Text	Updates heading text
Get Gears Color and Opacity	Updates gears indicator
Get Flaps Color and Opacity	Updates flaps indicator
Get Brake Color and Opacity	Updates air brake indicator
Get AF Color and Opacity	Updates afterburner indicator
Get Armed Text	Updates selected weapon name
Set Speed Units	Sets speed units
Set Altitude Units	Sets altitude units
Update Horizon Line	Updates artificial horizon line angle
Update Flight Path Indicator	Updates flight path indicator

Events:

Event Construct	Sets owner reference, sets units, hides afterburner indicator if not available, hides VTOL Indicator if not available
Event Tick	Updates flight path indicator, updates horizon line

BP_AirVehicle_JSBSim – Flight Model JSBSim

This is the base aircraft actor with JSBSim that has common functionalities for all air vehicles.

Aircraft variables:

Commands	Struct with command for JSBSim	
Throttle	Indicates current throttle position	0-1 means 0% – 100%
AileronControl	Control of aileron	-1 up to 1 (-100% to 100%)
ElevatorControl	Control of elevators	-1 up to 1 (-100% to 100%)
RudderControl	Control of rudder	-1 up to 1 (-100% to 100%)
YawTrim	Trim of yaw	
PitchTrim	Trim of pitch	
RollTrim	Trim of roll	
TrimStep	Step to adjust trims	
FlapPosition	Indicates flaps position. 0 means up, 1 full extended	0-1 means 0% – 100%
Spoilers	Indicates wing spoilers' position. 0 means up, 1 full extended	0-1 means 0% – 100%
Speed Brake	Indicates air brake position. 0 means up, 1 full extended	0-1 means 0% – 100%
Gear Down	Indicates gears position. 1 means extended, 0 retracted	0-1 means 0% – 100%
GearNormValue	Indicates gears normalized position. 1 means extended, 0 retracted	0-1 means 0% – 100%
WheelSteer	Indicates gears steering turn position	-1 up to 1 (-100% to 100%)
WheelsVelocity	Indicates velocity of landing gear wheels	
Left Brake	Indicates left gear brake position	0-1 means 0% – 100%
Right Brake	Indicates right gear brake position	0-1 means 0% – 100%
Center Brake	Indicates center gear brake position	0-1 means 0% – 100%
Parking Brake	Indicates parking brake position	0-1 means 0% – 100%
CASknts	Indicates calculated air speed	Knots
Altitude ASLft	Indicates altitude from sea level	Feet
EnginesRPM	Indicates engine RPM for each available engine	RPM
Crashed	Indicates crashed state	
ShowDust	Indicates if to show dust VFX	
TerrainDetectionRadius	Radius of dust smoke VFX	
TerrainDetectionOffset	End point of terrain detection	X,Y,Z
DoubleProperties	Double properties to be used with JSBSim	
StringProperties	String properties to be used with JSBSim	

Functions:

ApplyFlightCommands	Sends flight commands from inputs to JSBSim
PrintAircraftProperties	Debug prints aircraft properties from JSBSim
ApplyProperties	Sends properties to JSBSim
ToggleGears	Toggles landing gears
UpdateGearsNormalValue	Updates landing gears normalized position
ToggleFlaps	Toggles flaps
ToggleAirBrake	Toggles air brake
UpdateFlyData	Updates flight data from JSBSim
UpdateEnginesRPM	Updates engine data from JSBSim
DetectTerrain	Updates if terrain is nearby for dust VFX

Events:

BeginPlay	Applies JSBSim properties on begin, gets states of gears and flaps, starts terrain detection
Event Tick	Updates JSBSim properties, commands, and states of the aircraft
UpdateThrottle	Updates throttle value based on input. Calls corresponding event used by child actor
SetPitch	Sets pitch value
SetYaw	Sets yaw value
SetRoll	Sets roll value

InputMapping Graph:

Maps inputs from the Project Settings into the corresponding events in the EventGraph.

BP_Airplane_JSBSim

This is an airplane BP actor with JSBSim movement that has common functionalities for airplanes (not helicopters).

Internal calculation variables:

Following variables are used for internal calculations and states. They are also used to provide values for child Blueprint actor of a specific airplane.

Yaw	Indicates yaw input. -1 up to 1	-1 up to 1 (-100% to 100%)
Roll	Indicates roll input. -1 up to 1	-1 up to 1 (-100% to 100%)
Pitch	Indicates pitch input. -1 up to 1	-1 up to 1 (-100% to 100%)
isAfterburnerActive	Is True if afterburner is active	
Augmentation	Indicates input for engine augmentation for afterburner method 0	
HideHUDOnSecondaryCam	If to hide HUD on secondary camera to prevent duplicity	

Camera variables:

The following is an overview of the variables from the Cameras category.

Important: You should change the variables set up in the corresponding child Blueprint actor of a specific airplane.

CockpitCameraZoomedTransform	Sets Interior camera position and rotation when zoomed. Can be used to switch to second seat.
CockpitCameraUnZoomedTransform	Sets Interior camera position and rotation when in default position. Can be used to switch to default pilot seat.
CockpitCameraZoomedFOV	Sets the field of view of the zoomed camera
CockpitCameraUnZoomedFOV	Sets the field of view of the default camera

Functions:

SetActiveCamera	Removes HMD if in interior view. Optional.
UpdateJSBSimEngines	Updates commands for JSBSim Engines and gets their states

Events:

EventTick	Updates JSBSim engines and afterburners
UpdateYaw	Updates yaw rotation of airplanes based on input. Calls corresponding event used by child actor to set animation in ABP
UpdateRoll	Updates roll rotation of airplanes based on input. Calls corresponding event used by child actor to set animation in ABP
UpdatePitch	Updates pitch rotation of airplanes based on input. Calls corresponding event used by child actor to set animation in ABP
ToggleLights	Switches on and off Lights. Lights will flash when on.
ToggleCanopy	Opens and closes canopy based on input (not supported by JSBSim)
Sets Canopy	Calls corresponding event used by child actor to set animation in ABP
Input Action Zoom+	Zooms the view of fixed interior camera or switches seat
Input Action Zoom-	Un-zooms the view of fixed interior camera or switches seat back
UpdateAim	Update aim of weapons

SetThrottle	Calls corresponding event used by child actor to set animation in ABP
SetAfterburner	Calls corresponding event used by child actor to set animation in ABP

InputMapping Graph:

Maps inputs from Project Settings into the corresponding events in the EventGraph.

BP_BaseAirplaneVehicle – Projectile based Flight Model

This is a base airplane actor that has common functionalities with for airplane vehicles. This includes mainly flight model calculations, HUD/HMD, and inputs handling.

This asset uses a flight model for simple airplane flight as well as VTOL functionality (*It is not intended for helicopter movement. For helicopters, please use one of our helicopter assets with BP_BaseHeliVehicle*).

Flight model is based on calculations of:

- Throttle
- Trust Vectoring
- Drag
- Lift

Final forward, up and right velocity is sent as a vector to Epic's Unreal Engine projectile movement component. To turn this functionality off, just deactivate the aforementioned component.

Basic variables:

The following is an overview of the variables from Basic category.

Important: You should change the variables set up in corresponding child Blueprint actor of a specific airplane.

isFlying	If true, the airplane is set into flying mode inclusive minimum cruising speed. The off value is used for static airplane presentation
TopSpeed	Maximum speed of airplane on 100% throttle without afterburner. In cm/s
TopSpeedAF	Maximum speed with afterburner. In cm/s
has Afterburner	Set True If airplane is equipped with afterburner
isVTOL	Set True if airplane has VTOL capability
HideHMDOOnSecondaryCam	Set True if you want to hide Head Mounted Display (HMD) when looking thru cockpit nose Head Up Display (HUD)
InvertPitch	Set True if you want to invert pitch input values (usually joystick, gamepad users)

Advanced variables:

The following is an overview of the variables from the Advanced category.

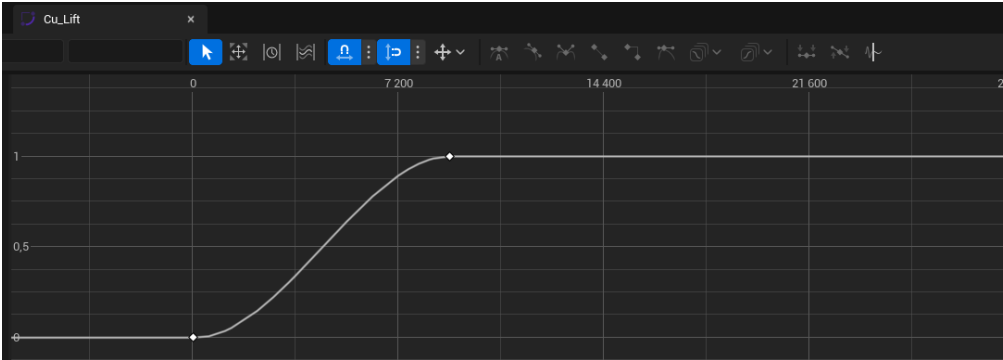
Important: You should change the variables set up in corresponding child Blueprint actor of a specific airplane.

YawMaxTurnRate	Sets maximum turn rate for yaw rotation
RollMaxTurnRate	Sets maximum turn rate for roll rotation
PitchMaxTurnRate	Sets maximum turn rate for pitch rotation
ThrottleRate	Sets how fast throttle will react on change
AccelerationRate	Sets how fast the airplane will accelerate forward
MaxVTOLZSpeed	Sets maximum up velocity when in VTOL mode
FlapsLiftIndex	Sets flaps lift multiplier when flaps are extended
GearsDragIndex	Sets gear drag multiplier when gears are extended
AirBrakeDragIndex	Sets air brake drag multiplier when air brake is extended
TerrainDetectionRadius	Radius of dust smoke VFX
DustMultiplier	Density of dust smoke VFX
TerrainDetectionOffset	End point of terrain detection
LiftCurve	Reference to lift curve. Lift curve can be adjusted to desired lift force. Key 0,0 indicates 0 lift on 0 air speed. Key X,1 indicates lift matching gravity force at X air speed.

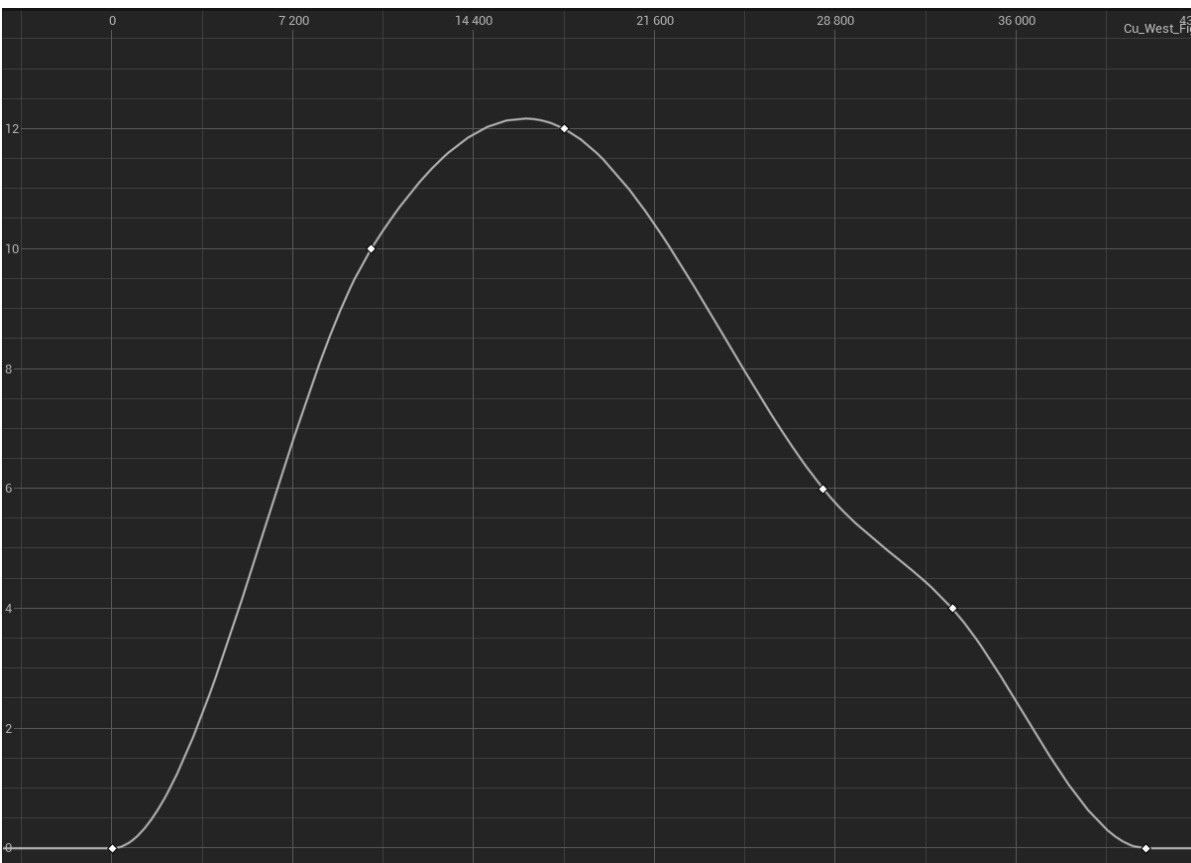
PitchTurnRateCurve

Reference to turn rate curve. Turn rate curve can be adjusted to the desired turn rate. Key 0,0 indicates 0 turn rate on 0 air speed. Key on Y axis indicates turn rate in cm/s at X air speed.

Cu_Lift:



Cu_TurnRate:



Camera variables:

Overview of the variables from the Cameras category.

Important: You should change the variables set up in corresponding child Blueprint actor of a specific airplane.

CockpitCameraZoomedTransform

Sets interior camera position and rotation when zoomed. Can be used to switch to second seat.

CockpitCameraUnZoomedTransform	Sets interior camera position and rotation when in default position. Can be used to switch to default pilot seat.
CockpitCameraZoomedFOV	Sets the field of view of the zoomed camera
CockpitCameraUnZoomedFOV	Sets the field of view of the default camera

Internal calculation variables:

The following variables are used for internal calculations and states. They are also used to provide values for the child Blueprint actor of a specific airplane.

Throttle	Indicates current throttle position	0-1 means 0% – 100%
EngineThrust	Indicates engine thrust	0-1 means 0% – 100%
MarginalVelocity	Indicates marginal velocity achievable with 100% throttle	cm/s
ForwardSpeed	Indicates forward speed (air speed)	cm/s
RightSpeed	Indicates right / left speed	cm/s
UpSpeed	Indicates up / down speed	cm/s
VTOLRatio	Indicates ratio between trust vectoring. 0 means full forward, 1 full up	0-1 means 0% – 100%
Gravity	Indicates gravity force	cm/s
TotalDragIndex	Indicates total drag multiplier – opposing force to forward speed	cm/s
FlapsValue	Indicates flaps position. 0 means up, 1 full extended	0-1 means 0% – 100%
GearsValue	Indicates gears position. 0 means extended, 1 retracted	0-1 means 0% – 100%
AirBrakeValue	Indicates air brake position. 0 means up, 1 full extended	0-1 means 0% – 100%
Yaw	Indicates yaw input. -1 up to 1	-1 up to 1 (-100% to 100%)
Roll	Indicates roll input. -1 up to 1	-1 up to 1 (-100% to 100%)
Pitch	Indicates pitch input. -1 up to 1	-1 up to 1 (-100% to 100%)
isAfterburnerActive	Is True if afterburner is active	
isCameraZoomed	Is True if fixed interior camera is zoomed or on secondary seat	
ShowDust	Indicates if to show dust VFX	

Functions:

Update Engine Thrust	Updates thrust of engines based on throttle. Currently thrust match throttle. Intended for future usage
Update Forward Velocity	Updates forward velocity based on engine thrust, marginal velocity, acceleration rate and VTOL vector
Update VTOLImpact	Updates impact of VTOL vector on up velocity based on VTOL Vector ratio, max VTOL up speed and engine thrust
Update Drag	Updates impact of gear position and air brake position on total drag
Update Lift	Updates lift based on lift curve, forward speed, flaps position, VTOL ratio. Incorporates gravity impact.
SetActiveCamera	Removes HMD if in interior view. Optional.
DetectTerrain	Updates if terrain is nearby for dust VFX
UpdateDust	Updates dust VFX

Events:

BeginPlay	Sets marginal velocity, switch on lights, sets initial values if airplane is spawned as flying, starts terrain detection
UpdateThrottle	Updates throttle value based on input. Calls corresponding event used by child actor

UpdateSpeed	Updates X,Y,Z velocity based on thrust, forward velocity, VTOL impact, drag and lift. Calls Epic's Unreal Engine projectile movement component with final X,Y,Z velocity values
ToggleAfterburner	Turns afterburner on and off. Calls corresponding event used by Child actor to show VFX
UpdateYaw	Updates yaw rotation of airplanes based on input. Calls corresponding event used by child actor to set animation in ABP
UpdateRoll	Updates roll rotation of airplanes based on input. Calls corresponding event used by child actor to set animation in ABP
UpdatePitch	Updates pitch rotation of airplanes based on input. Calls corresponding event used by child actor to set animation in ABP
UpdateVTOLRatio	Updates VTOL ratio of airplane based on input if VTOL capable. Calls corresponding event used by child actor to set animation in ABP
ToggleGear	Extends and retracts landing gears based on input. Calls corresponding event used by child actor to set animation in ABP
ToggleFlaps	Extends and retracts flaps gears based on input. Calls corresponding event used by child actor to set animation in ABP
ToggleAirBrake	Extends and retracts air brake based on input. Calls corresponding event used by child actor to set animation in ABP
ToggleLights	Switches on and off lights. Lights will flash when on
TurnOffLights	Turns off lights
ToggleCanopy	Opens and closes canopy based on input. Calls corresponding event used by child actor to set animation in ABP
ToggleHook	Deploys hook based on input. Calls corresponding event used by child actor to set animation in ABP
ToggleWings	Folds wings based on input. Calls corresponding event used by child actor to set animation in ABP
Input Action Zoom+	Zooms the view of fixed interior camera or switches seat
Input Action Zoom-	Un-zooms the view of fixed interior camera or switches seat back
ToggleRefuel	Opens refuel pipe based on input. Calls corresponding event used by child actor to set animation in ABP
ToggleBombBay	Opens bomb bay based on input. Calls corresponding event used by child actor to set animation in ABP

InputMapping Graph:

This component maps inputs from the Project Settings into corresponding Events in the EventGraph.

BP_WeaponComponent

The weapons component is a shared component used by every Vigilante vehicle to fire and cycle between different weapons.

There are two types of weapons:

1. Trace-based: Based on trace (scan hit) intended for machineguns.
2. Projectile-based: Projectiles intended for bombs, missiles, and larger guns/cannons.

The available weapons are specified in the file [DT_WeaponsTable](#) in the Data folder and are specific for each vehicle.

A fire VFX effect, if needed, is spawned for a specific vehicle child actor BP. It's called by the events 'StartFire' and 'StopFire'.

A hit VFX effect is spawned by BP_WeaponComponent for trace-based weapons and by [BP_BaseProjectile](#) for projectile-based weapons.

Weapons are spawned on the vehicle's sockets and can be adjusted by offsets.

Variables:

Armaments	Array of weapons defined in DT_WeaponsTable
ActiveWeapon	Struct with information about selected active weapon
FiringTimer	Timer handle for fire
ActiveArmaments	Struct with information about active armament
ActiveIndex	Index of active weapon
SocketIndex	Index of active socket

Functions:

Cycle Weapon	Cycles between available armaments weapons
Fire	Fire active weapon from last socket based on tracer (Scanhit) or projectile type. Selects next socket
Start Fire	Fires repetitiously when trigger is hold. Calls effect event
Stop Fire	Stops fire if trigger is released. Stops effect event
Spawn Countermeasures	Spawns countermeasures
Activate Weapon	Get weapon information from DT table
Get Base Vehicle	Gets reference to owning base vehicle
Get Active Weapon Name	Gets active weapon name
Get Fire Location	Gets location where to start fire
Get Fire Rotation	Gets rotation of fire
Get Socket Offset	Gets socket offset to adjust the location and rotation
GetForwardVector	Returns forward vector of active socket
Get Trace End Location	Gets tracer maximum location for tracer (Scanhit) based weapons such as machine guns
Get Trace Start Location	Gets tracer starting location for tracer (Scanhit) based weapons such as machine guns

Events:

BeginPlay	Activates available weapons
SetFireActive	Toggles fire

DT_WeaponsTable

This is a data table for all available weapons for all vehicle types.

Please use the vehicle-specific DT_WeaponTable.

Important Note: Weapons do NOT represent or match exactly real-world specific missiles or bombs and are just for demonstrations purposes only.

Uses FVehicleWeapon Structure:

Friendly Name	UI Display friendly name of weapon
FireGroup	Enum with weapon type – tracer (hitscan) or projectile based
ProjectileClass	Projectile class to spawn in case of projectile based weapons
Firemode	Enum if fire is single, continuous, or burst type
Firerate	Fire Rate
SpawnOffset	General offset to adjust root of weapons. For a detailed attachment offset, use the vehicle specific offset and socket in the weapon component of the vehicle specific Blueprint
SpawnRotation	General rotation to adjust root of weapons. For a detailed attachment rotation, use the vehicle specific offset and socket in the weapon component of the vehicle specific Blueprint
TracerLength	Length of tracer-based weapons only. Projectile-based weapons use their own life span in BP_BaseProjectile
TracerImpactVFX	Reference to Impact VFX of tracer-based weapons only. Projectile-based weapons use their own VFX defined in BP_BaseProjectile

BP_BaseProjectile

Used by the [BP_WeaponComponent](#) to spawn projectile movement-based weapon fire.

Supports static and/or skeletal mesh.

Variables:

BaseVehicle	Reference to base vehicle that spawned the projectile
HitEffectScale	Scale of hit effect
HitEffectRef	Reference to spawned Niagara system on hit event
isFuseActive	Set up by timer to activate the fuse. If True hit effect will be spawned on hit event

Functions:

Initialize	Sets initial projectile speed based on vehicle movement type and its velocity
Spawn Hit Effect	Spawns hit effect
Set Fuse Active	Activates fuse

Events:

Event Begin Play	Sets life span, sets fuse activation timer, enables collisions not to collide on Spawn
Event Hit	Add physics impulse to any physics object on hit. Spawns Hit Effect and destroys

BP_Missile_Projectile

Used by missiles with self-propulsion.

Variables:

EngineStartDelay	Delay to start the engine. Engine start can be delayed not to harm firing vehicle
EngineRunningTime	Time period for engine running in seconds
MissileMaxSpeed	Missile maximum achievable speed in cm/s
AccelerateRate	Acceleration rate of missile

Functions:

Start Engine	Starts engine
Stop Engine	Stops engine and enables free fall
Accelerate	Acceleration to max speed

Events:

Event Begin Play	Starts engine
-------------------------	---------------

BP_BaseBomb_Projectile

Base Bomb with no special functionality used by specific bomb child classes.

BP_BaseCannon_Projectile

Base Canon Projectile with no special functionality used by specific canon projectiles child classes.

Generic Projectiles Examples

This are just examples of generic projectiles for each type.

Important Note: they do NOT represent or match exactly real-world specific missiles or bombs and are just for demonstrations purposes only.

- **BP_BaseCannon_120mm_Projectile**
- **BP_BaseBomb_Bomb_Generic_Projectile**
- **BP_BaseMissile_AA_Generic_Projectile**
- **BP_BaseMissile_AGM_Generic_Projectile**
- **BP_BaseMissile_HeliAGM_Generic_Projectile**
- **BP_BaseMissile_ATGM_DE_Generic_Projectile**
- **BP_BaseMissile_TOW_Generic_Projectile**

BP_CommunicationWidget

WBP_Inputs

A widget that generates UI with input mappings for vehicles.

Variables:

VehicleWidget	Reference to vehicle widget
InputDataTable	Reference to specific vehicle DT table with Inputs
VerticalList	Vertical list to generate child slots widgets for each Input

Functions:

Initialize	Initializes widgets and adds all available axis and actions inputs as WBP_Inputs_Slot children to VerticalList
Get Base Vehicle	Gets reference to owning base vehicle
Get Widget Communicator	Gets reference to owning base vehicle widget communicator component
Get Input Data Table	Returns reference of data table

Events:

Event Begin Play	Initializes content
-------------------------	---------------------

WBP_Inputs_Slot

Widget slot for each input mapping.

Variables:

ActionFriendlyName	Friendly action name to display
InputDisplayName	Input display value name to display

Functions:

Initialize	Initializes values in widget Slot
-------------------	-----------------------------------

Events:

Event Begin Play	Initializes content in slot
-------------------------	-----------------------------

DT_InputsMapping

Stores information for each mapping. Used by WBP_Inputs mapping.

Uses F_InputsKeyMapping Structure:

Show	True if action should be shown in UI for specific vehicle
FriendlyName	Friendly name for action
InputName	Input name defined in project setting input
isAxis	True if input is axis – requires special handling, false on action
AxisScale	Axis scale plus minus

Asset Versions

- Version 2.0.0 – Base version