MALOKE GAMES

ASSETS

DRONE SIMULATION

- Contact: Maloke7-Games@yahoo.com.br
- Full Portfolio: https://maloke.itch.io/
- AssetStore: https://assetstore.unity.com/publishers/26634

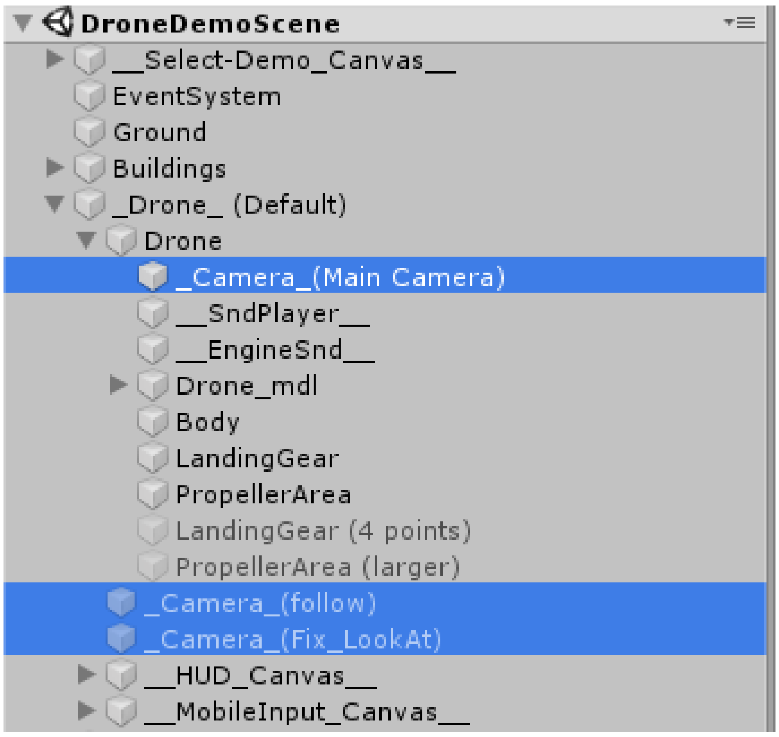
➢ **Quick Instructions:**

You can find a DemoScene with this asset configured and working straight away for many different modes as examples. To add the Drone to your own game you ***just need to drop one of the PreFabs into your scene*** and you are Ready to Go!

Simple steps:

1- Drop on your scene any of the "X_**Drone_(*type*).prefab**" prefab.

2- Make sure you don't have any other MainCamera active on the scene otherwise it could override one of the Drone's own cameras. Just disable other cameras and everything should work!

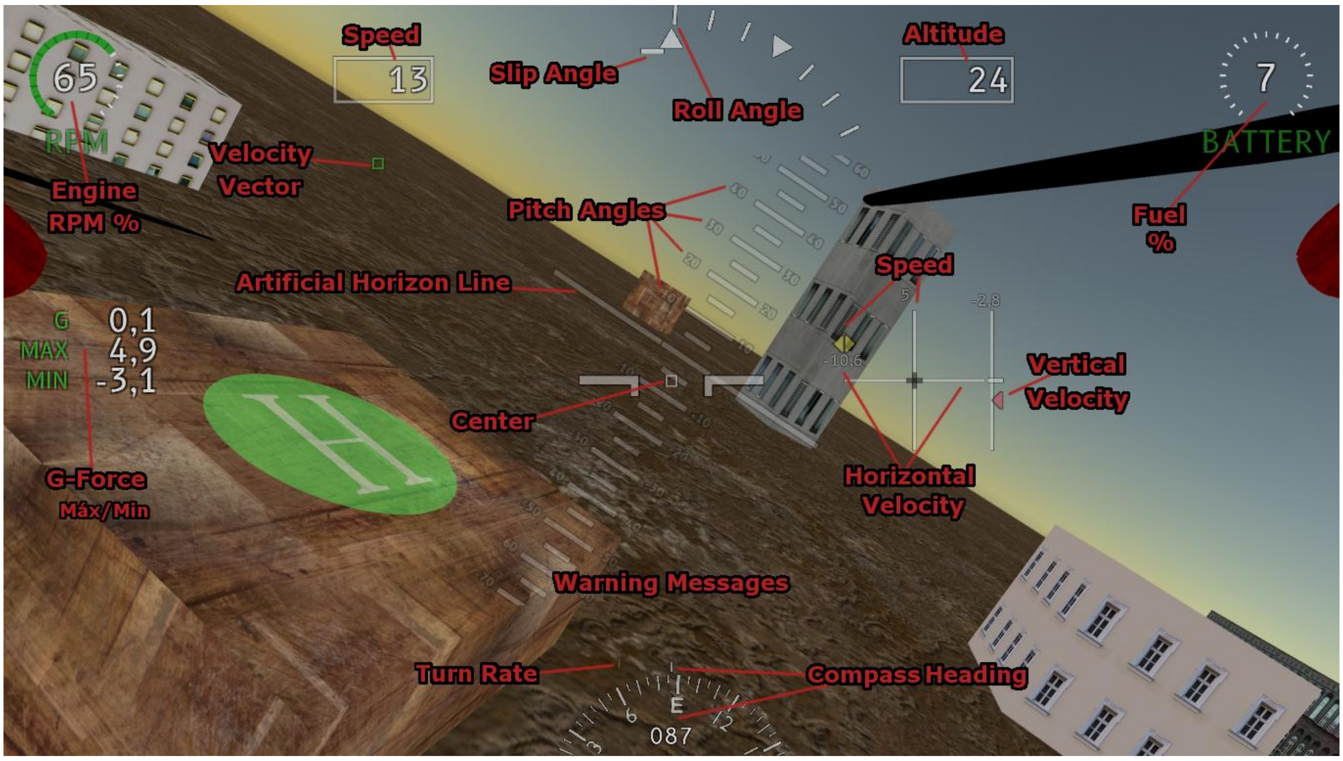


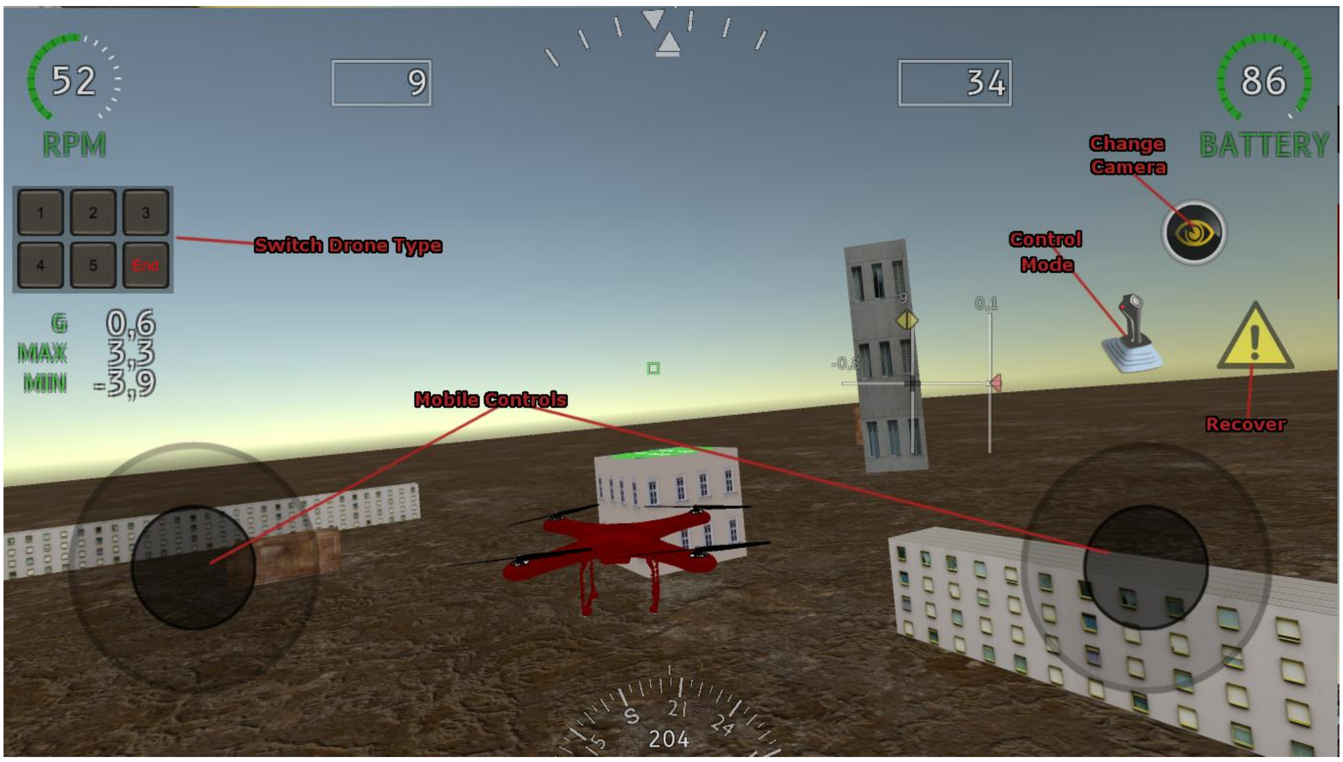
```
DroneDemoScene
  __Select-Demo_Canvas__
  EventSystem
  Ground
  Buildings
  _Drone_ (Default)
    Drone
      _Camera_(Main Camera)
      __SndPlayer__
      __EngineSnd__
      Drone_mdl
      Body
      LandingGear
      PropellerArea
      LandingGear (4 points)
      PropellerArea (larger)
    _Camera_(follow)
    _Camera_(Fix_LookAt)
    __HUD_Canvas__
    __MobileInput_Canvas__
```

This asset includes ***5 Prefab Templates*** with variations on each one. You can start with a prefab and then customize your own version. We recommend that you break prefab instance before making modifications to avoid overwriting the original asset example.

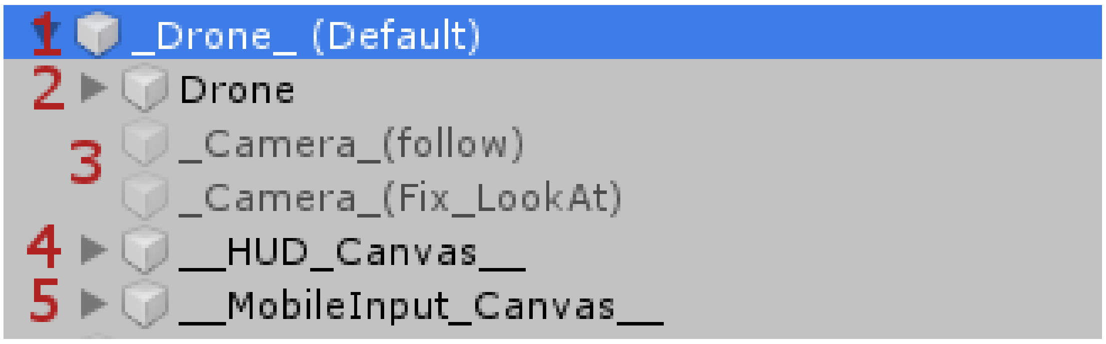
If you want to know more about this asset and how to customize or tweak it more in depth, you can find extra information further on this document.

➢ **Instruments Symbology:**





➢ **Main Components:**


1 _Drone_ (Default)
2 Drone
3 _Camera_(follow)
Camera(Fix_LookAt)
4 __HUD_Canvas__
5 __MobileInput_Canvas__

1- The Main root of the PreFab.

2- It contains the **DroneMainScript** which you can configure the Drone's behavior and tweak all necessary values there.

3- Alternative cameras. Inside component 2 you can also find the First-Person Camera gameObj.

4- It contains the **DroneHUD** script that controls the *HUD instruments* and ***Canvas***, the **Sounds**, and **Messages**. You don't need to do any adjustments in this script.

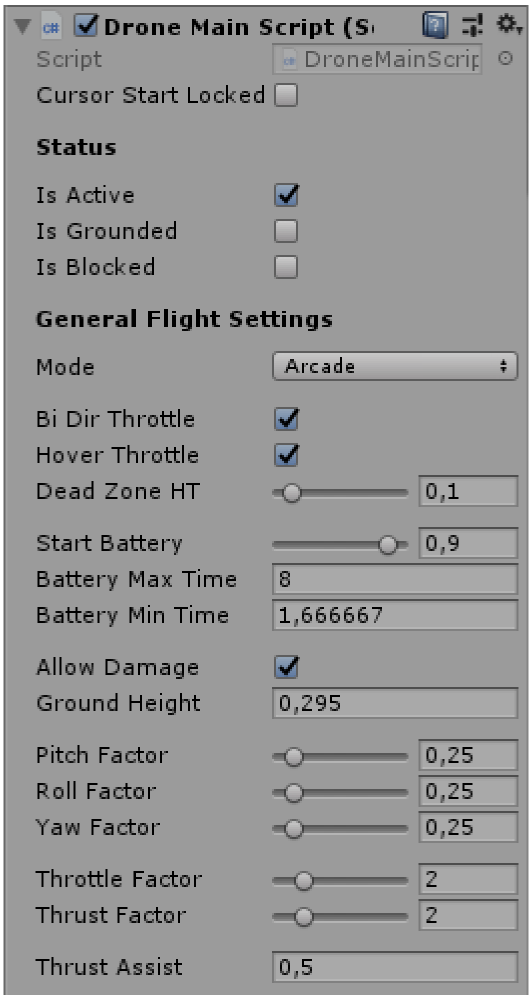
5- It contains the ***Canvas*** for the Mobile Input Controls. Inside you can locate the **DroneInputMobile** script and *configure the axis as you wish*.

**** ATTENTION!! - DO NOT DELETE THE (DATA CONSOLE) inside HUD Canvas!***

Other instruments get some values from it and deleting it may create missing references. Other instruments can be deleted if not used but make sure to check for missing references for instruments and reconnect them on the **DroneHUD** script if necessary.

*****_Some components are used "under the hood" by the asset and do not require any setup or configuration... but fell free to use them for extra purposes on your project if you like!_

➢ **DroneMainScript - Editor Parameters:**



Drone Main Script (S	✓
Script	DroneMainScrip
Cursor Start Locked	☐
Status	
Is Active	✓
Is Grounded	☐
Is Blocked	☐
General Flight Settings	
Mode	Arcade
Bi Dir Throttle	✓
Hover Throttle	✓
Dead Zone HT	0,1
Start Battery	0,9
Battery Max Time	8
Battery Min Time	1,666667
Allow Damage	✓
Ground Height	0,295
Pitch Factor	0,25
Roll Factor	0,25
Yaw Factor	0,25
Throttle Factor	2
Thrust Factor	2
Thrust Assist	0,5

- **isActive**: Indicates that the script is enabled.

-**isGrounded**: Indicates the Drone is firmly on the ground.

-**isBlocked**: Used when out of fuel or propeller damaged and indicates that the Drone can't fly.

General Flight Settings

Modes: You have 3 modes of Control

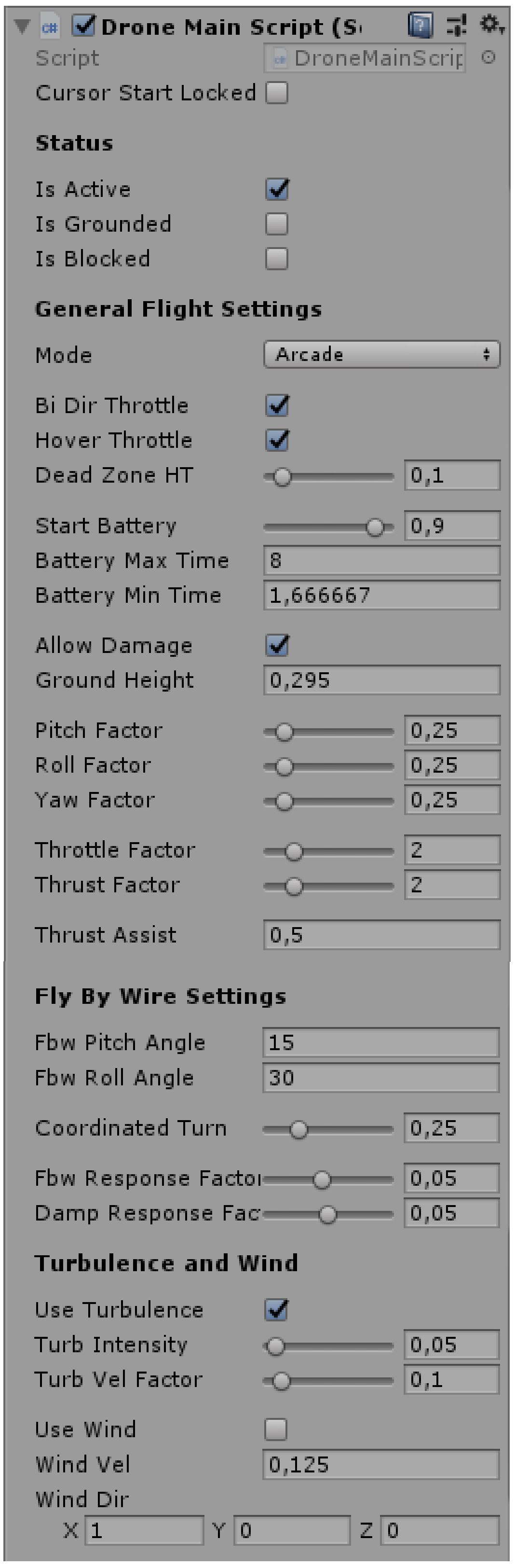
- **Manual**: (aka Acro Mode) with full 6 DOF control where you can tumble and fly upside down (if ***Bi-Directional*** throttle is also activated). It needs time to master the flight, recommended for Simulations.

- **Fly-By-Wire**: It acts as a real FBW system, you still fly the same realistic physics but the script modulates your input to avoid tumbling and it Auto Stabilizes the drone if you leave the controls. With ***Hover-Throttle*** activated you still get a very realistic feeling but it's much easier to control (You can still tumble but only if you hit hard on something)

- **Arcade**: With both ***Bi-Directional*** and ***Hover-Throttle*** activated anyone can Easily Fly. Good for Cinematic Flight, Action-Arcade GameStyle or even for just exploring a large Tech-Demo scene where you don't want your player to focus on actually controlling the Drone itself.

- BiDirThrottle: This enables the drone to have a downward force by reversing the propeller direction. This is also known as ***3D Mode***. You need this to fly upside down.

- Hover Throttle: This raises the throttle RPM automatically to counterweight the Drone's weight, so you can easily hover without constant adjustments. Depending on other settings, this can also auto-trim the throttle to ensures that a 50% throttle axis position corresponds to the hover state. The **DeadZoneHoverThrottle** option gives you some dead zone in this state.



Drone Main Script (S

Script	DroneMainScrip
Cursor Start Locked	☐
Status	
Is Active	✓
Is Grounded	☐
Is Blocked	☐
General Flight Settings	
Mode	Arcade
Bi Dir Throttle	✓
Hover Throttle	✓
Dead Zone HT	0,1
Start Battery	0,9
Battery Max Time	8
Battery Min Time	1,666667
Allow Damage	✓
Ground Height	0,295
Pitch Factor	0,25
Roll Factor	0,25
Yaw Factor	0,25
Throttle Factor	2
Thrust Factor	2
Thrust Assist	0,5
Fly By Wire Settings	
Fbw Pitch Angle	15
Fbw Roll Angle	30
Coordinated Turn	0,25
Fbw Response Factoı	0,05
Damp Response Fac	0,05
Turbulence and Wind	
Use Turbulence	✓
Turb Intensity	0,05
Turb Vel Factor	0,1
Use Wind	☐
Wind Vel	0,125
Wind Dir	X 1 Y 0 Z 0

-StartBattery: This is the % value of fuel/battery when the Drone is started.

-Battery Max/Min Time: These values correspond to how much *time in minutes* the Drone can fly. ***MaxTime*** determines how long it can operate *on Idle Throttle position*, and ***MinTime*** is how long operating *at 100% RPM*. ***Set both to Zero if you don't want to consume battery.***

- AllowDamage: Disable this if you don't want the *propeller to be damaged*. Inside the ***PreFab*** you can *find alternative colliders* to make the collision harder or easier to happen.

- GroundHeight: This is used by a RayCast to determine if the Drone is landed on the ground. If you switch the prefab model or have a larger drone you should update this value.

- Pitch/Roll/Yaw Factors: These values determine the intensity of these maneuvers in each axis (They are Mass and Gravity normalized).

-Throttle/Thrust Factors: These values determine the intensity of the Drone movement to gain ***Speed (Thrust)*** and ***Altitude (Throttle)***. **ThrustAssist** is used only on Manual mode to generate an extra horizontal thrust to facilitate maneuvers (They are Mass and Gravity normalized).

Fly-By-Wire

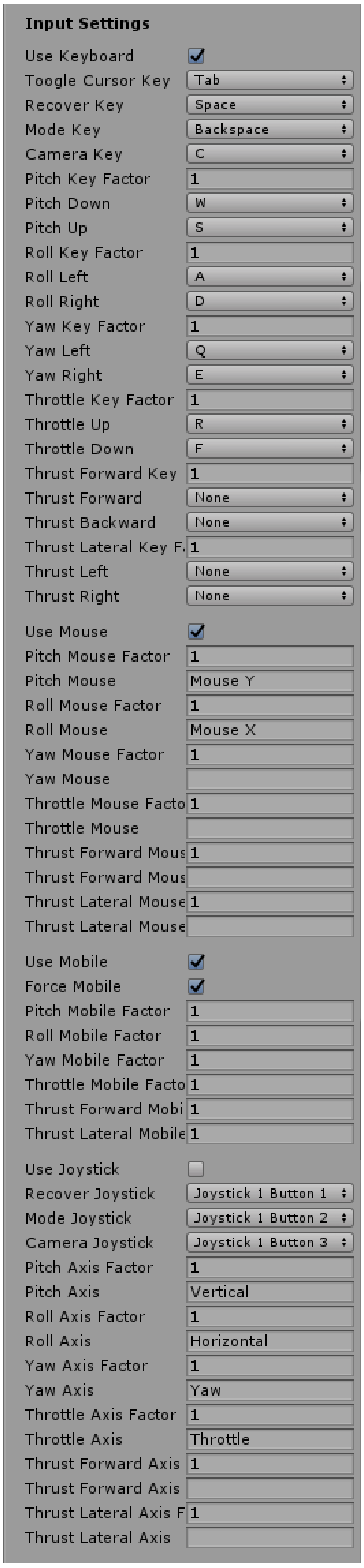
- Pitch/Roll Angle: Determines the máx target angle the Drone should be allowed to tumble during FBW mode.

- CoordinatedTurn: On ***Arcade*** mode if you roll the Drone this will *add some yaw* movement to help coordinate the turn, making it easier to control. You can set the amount of *"turning help"* with this slider.

- Fbw Response/Damp Factors: These values sets how fast FBW system will try to control the drone. Damp factors avoid overshoot but excessive damp can make the response too slow. You don't need to alter these values for the default prefabs.Tweak these values with care on your custom drone!

Turbulence and Wind

On this section you can set the amount of turbulence (**TurbIntensity**) and how much it increases with the speed (**TurbVelFactor**). You can also set the Wind velocity and its direction in a Vector3 **WindDir**.

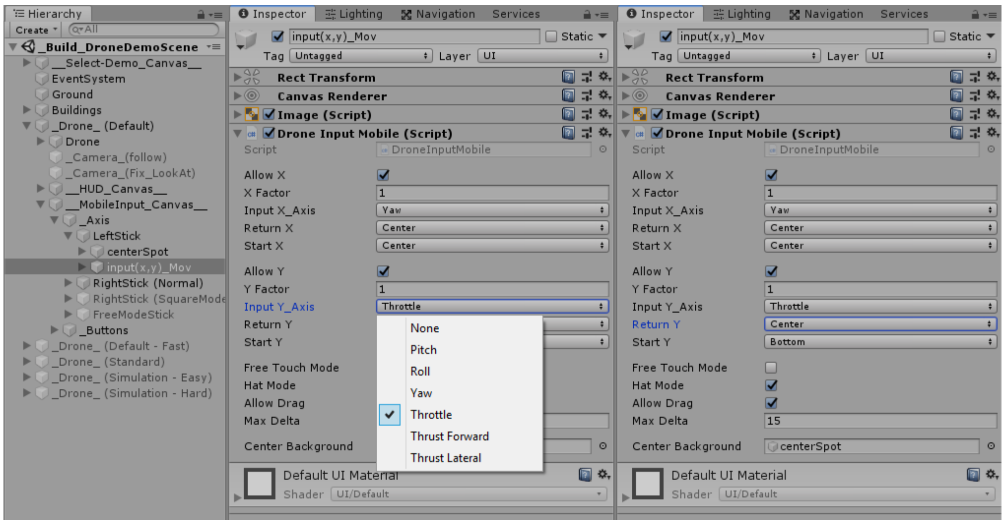


Input Settings

Use Keyboard	✓
Toogle Cursor Key	Tab
Recover Key	Space
Mode Key	Backspace
Camera Key	C
Pitch Key Factor	1
Pitch Down	W
Pitch Up	S
Roll Key Factor	1
Roll Left	A
Roll Right	D
Yaw Key Factor	1
Yaw Left	Q
Yaw Right	E
Throttle Key Factor	1
Throttle Up	R
Throttle Down	F
Thrust Forward Key	1
Thrust Forward	None
Thrust Backward	None
Thrust Lateral Key F	1
Thrust Left	None
Thrust Right	None
Use Mouse	✓
Pitch Mouse Factor	1
Pitch Mouse	Mouse Y
Roll Mouse Factor	1
Roll Mouse	Mouse X
Yaw Mouse Factor	1
Yaw Mouse	
Throttle Mouse Facto	1
Throttle Mouse	
Thrust Forward Mous	1
Thrust Forward Mous	
Thrust Lateral Mouse	1
Thrust Lateral Mouse	
Use Mobile	✓
Force Mobile	✓
Pitch Mobile Factor	1
Roll Mobile Factor	1
Yaw Mobile Factor	1
Throttle Mobile Facto	1
Thrust Forward Mobi	1
Thrust Lateral Mobile	1
Use Joystick	
Recover Joystick	Joystick 1 Button 1
Mode Joystick	Joystick 1 Button 2
Camera Joystick	Joystick 1 Button 3
Pitch Axis Factor	1
Pitch Axis	Vertical
Roll Axis Factor	1
Roll Axis	Horizontal
Yaw Axis Factor	1
Yaw Axis	Yaw
Throttle Axis Factor	1
Throttle Axis	Throttle
Thrust Forward Axis	1
Thrust Forward Axis	
Thrust Lateral Axis F	1
Thrust Lateral Axis	

Input Settings

In this section, you can customize all the inputs for **Keyboard**, **Mouse**, **Joystick** and **Mobile** *(you can also customize each individual mobile axis and its auto-return feature by locating the script DroneMobileInput inside the __**MobileInput_Canvas**__ object inside the prefab).*



_Build_DroneDemoScene
- __Select-Demo_Canvas__
- EventSystem
- Ground
- Buildings
- _Drone_ (Default)
  - Drone
  - _Camera_(follow)
  - _Camera_(Fix_LookAt)
  - __HUD_Canvas__
  - __MobileInput_Canvas__
    - _Axis
      - LeftStick
        - centerSpot
        - input(x,y)_Mov
      - RightStick (Normal)
      - RightStick (SquareMode
      - FreeModeStick
    - _Buttons
- _Drone_ (Default - Fast)
- _Drone_ (Standard)
- _Drone_ (Simulation - Easy)
- _Drone_ (Simulation - Hard)

input(x,y)_Mov — Static — Tag: Untagged — Layer: UI

Rect Transform
Canvas Renderer
Image (Script)
Drone Input Mobile (Script)

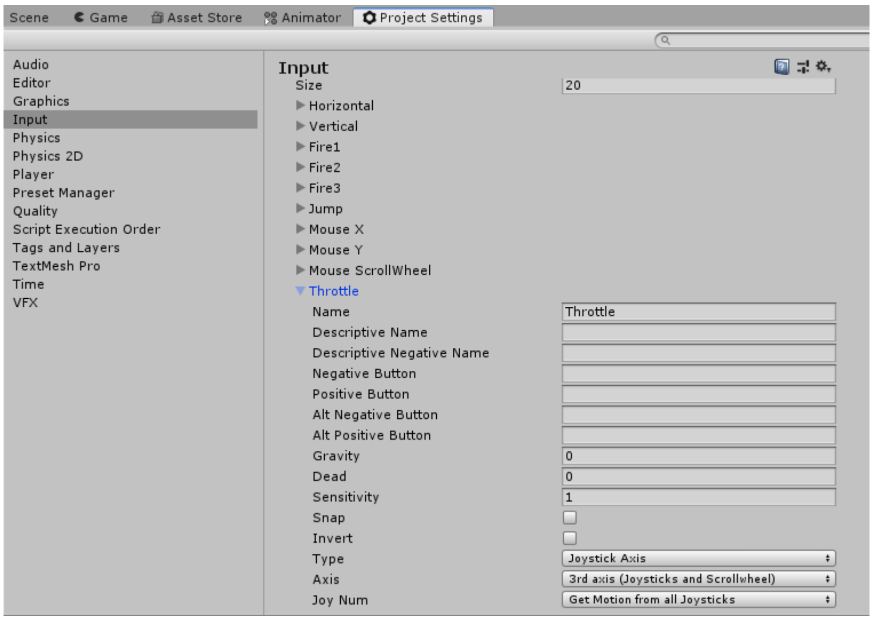
	Left inspector	Right inspector
Script	DroneInputMobile	DroneInputMobile
Allow X	✓	✓
X Factor	1	1
Input X_Axis	Yaw	Yaw
Return X	Center	Center
Start X	Center	Center
Allow Y	✓	✓
Y Factor	1	1
Input Y_Axis	Throttle	Throttle
Return Y		Center
Start Y		Bottom
Free Touch Mode		
Hat Mode		✓
Allow Drag		✓
Max Delta		15
Center Background		centerSpot

Dropdown: None, Pitch, Roll, Yaw, [✓ Throttle], Thrust Forward, Thrust Lateral

Default UI Material — Shader: UI/Default

The **Factor** fields allow you to reverse them by setting it to -1 or you can also clamp them by setting it to lower values between 0-1.

Note that for the **Joystick axis** they must exist in the Unity's Input section, otherwise you will get an error. The axis for ***Throttle*** and ***Yaw*** are not added by default, so you should create them as depicted in the image below:

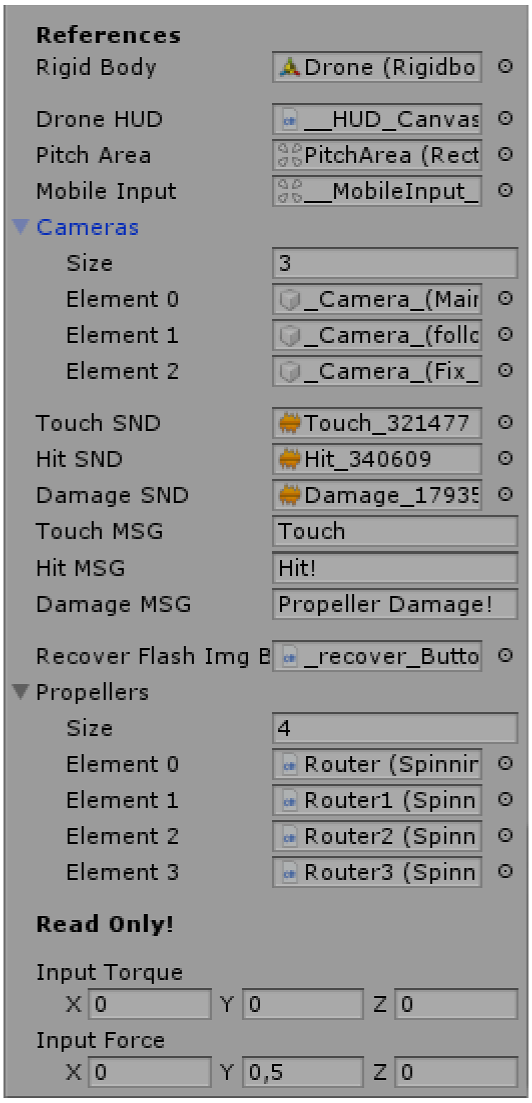


Input

Size: 20

Horizontal
Vertical
Fire1
Fire2
Fire3
Jump
Mouse X
Mouse Y
Mouse ScrollWheel
Throttle

Name	Throttle
Descriptive Name	
Descriptive Negative Name	
Negative Button	
Positive Button	
Alt Negative Button	
Alt Positive Button	
Gravity	0
Dead	0
Sensitivity	1
Snap	
Invert	
Type	Joystick Axis
Axis	3rd axis (Joysticks and Scrollwheel)
Joy Num	Get Motion from all Joysticks



References

Rigid Body	Drone (Rigidbo
Drone HUD	__HUD_Canvas
Pitch Area	PitchArea (Rect
Mobile Input	__MobileInput_
▼ Cameras	
Size	3
Element 0	_Camera_(Mair
Element 1	_Camera_(follc
Element 2	_Camera_(Fix_
Touch SND	Touch_321477
Hit SND	Hit_340609
Damage SND	Damage_17935
Touch MSG	Touch
Hit MSG	Hit!
Damage MSG	Propeller Damage!
Recover Flash Img B	_recover_Butto
▼ Propellers	
Size	4
Element 0	Router (Spinnir
Element 1	Router1 (Spinn
Element 2	Router2 (Spinn
Element 3	Router3 (Spinn

Read Only!

Input Torque
X 0 Y 0 Z 0

Input Force
X 0 Y 0,5 Z 0

References

This final section is used for the script's internal references and for customizing Sounds, Messages as:

- **Cameras**: Reference to the 3 cameras GameObject used by the Drone.

- **Touch/Hit/Damage** sounds used for these events.

- **MSGs** displayed during the above events.

-**RecoverFlashImgBut**: Used to flash the Recover button if the Drone is blocked.

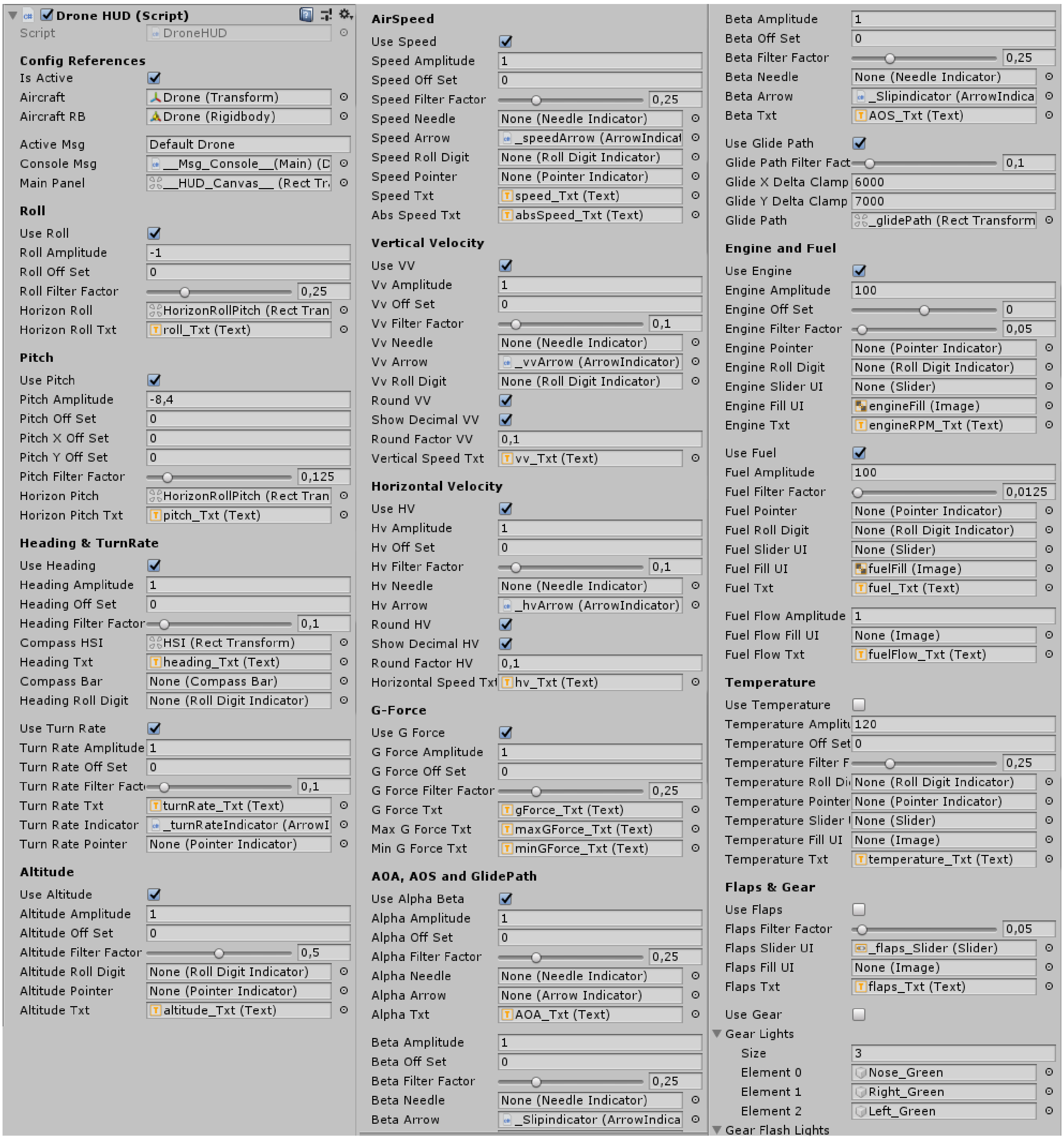
- **Propellers**: Reference to the Spinning script of each propeller. Used to make them spin.

Read Only!

Here you can see the Inputs for debug purpouses.

➢ DroneHUD Script:

You don't need to adjust any value in this script to use the PreFab Drones, but in case you want to customize further, then here is an explanation of how this script is structured:



✓ **Drone HUD (Script)**	
Script	DroneHUD
Config References	
Is Active	✓
Aircraft	Drone (Transform)
Aircraft RB	Drone (Rigidbody)
Active Msg	Default Drone
Console Msg	__Msg_Console__(Main) (D
Main Panel	__HUD_Canvas__ (Rect Tr
Roll	
Use Roll	✓
Roll Amplitude	-1
Roll Off Set	0
Roll Filter Factor	0,25
Horizon Roll	HorizonRollPitch (Rect Tran
Horizon Roll Txt	roll_Txt (Text)
Pitch	
Use Pitch	✓
Pitch Amplitude	-8,4
Pitch Off Set	0
Pitch X Off Set	0
Pitch Y Off Set	0
Pitch Filter Factor	0,125
Horizon Pitch	HorizonRollPitch (Rect Tran
Horizon Pitch Txt	pitch_Txt (Text)
Heading & TurnRate	
Use Heading	✓
Heading Amplitude	1
Heading Off Set	0
Heading Filter Factor	0,1
Compass HSI	HSI (Rect Transform)
Heading Txt	heading_Txt (Text)
Compass Bar	None (Compass Bar)
Heading Roll Digit	None (Roll Digit Indicator)
Use Turn Rate	✓
Turn Rate Amplitude	1
Turn Rate Off Set	0
Turn Rate Filter Fact	0,1
Turn Rate Txt	turnRate_Txt (Text)
Turn Rate Indicator	_turnRateIndicator (ArrowI
Turn Rate Pointer	None (Pointer Indicator)
Altitude	
Use Altitude	✓
Altitude Amplitude	1
Altitude Off Set	0
Altitude Filter Factor	0,5
Altitude Roll Digit	None (Roll Digit Indicator)
Altitude Pointer	None (Pointer Indicator)
Altitude Txt	altitude_Txt (Text)

AirSpeed	
Use Speed	✓
Speed Amplitude	1
Speed Off Set	0
Speed Filter Factor	0,25
Speed Needle	None (Needle Indicator)
Speed Arrow	_speedArrow (ArrowIndicat
Speed Roll Digit	None (Roll Digit Indicator)
Speed Pointer	None (Pointer Indicator)
Speed Txt	speed_Txt (Text)
Abs Speed Txt	absSpeed_Txt (Text)
Vertical Velocity	
Use VV	✓
Vv Amplitude	1
Vv Off Set	0
Vv Filter Factor	0,1
Vv Needle	None (Needle Indicator)
Vv Arrow	_vvArrow (ArrowIndicator)
Vv Roll Digit	None (Roll Digit Indicator)
Round VV	✓
Show Decimal VV	✓
Round Factor VV	0,1
Vertical Speed Txt	vv_Txt (Text)
Horizontal Velocity	
Use HV	✓
Hv Amplitude	1
Hv Off Set	0
Hv Filter Factor	0,1
Hv Needle	None (Needle Indicator)
Hv Arrow	_hvArrow (ArrowIndicator)
Round HV	✓
Show Decimal HV	✓
Round Factor HV	0,1
Horizontal Speed Txt	hv_Txt (Text)
G-Force	
Use G Force	✓
G Force Amplitude	1
G Force Off Set	0
G Force Filter Factor	0,25
G Force Txt	gForce_Txt (Text)
Max G Force Txt	maxGForce_Txt (Text)
Min G Force Txt	minGForce_Txt (Text)
AOA, AOS and GlidePath	
Use Alpha Beta	✓
Alpha Amplitude	1
Alpha Off Set	0
Alpha Filter Factor	0,25
Alpha Needle	None (Needle Indicator)
Alpha Arrow	None (Arrow Indicator)
Alpha Txt	AOA_Txt (Text)
Beta Amplitude	1
Beta Off Set	0
Beta Filter Factor	0,25
Beta Needle	None (Needle Indicator)
Beta Arrow	_Slipindicator (ArrowIndica

Beta Amplitude	1
Beta Off Set	0
Beta Filter Factor	0,25
Beta Needle	None (Needle Indicator)
Beta Arrow	_Slipindicator (ArrowIndica
Beta Txt	AOS_Txt (Text)
Use Glide Path	✓
Glide Path Filter Fact	0,1
Glide X Delta Clamp	6000
Glide Y Delta Clamp	7000
Glide Path	_glidePath (Rect Transform
Engine and Fuel	
Use Engine	✓
Engine Amplitude	100
Engine Off Set	0
Engine Filter Factor	0,05
Engine Pointer	None (Pointer Indicator)
Engine Roll Digit	None (Roll Digit Indicator)
Engine Slider UI	None (Slider)
Engine Fill UI	engineFill (Image)
Engine Txt	engineRPM_Txt (Text)
Use Fuel	✓
Fuel Amplitude	100
Fuel Filter Factor	0,0125
Fuel Pointer	None (Pointer Indicator)
Fuel Roll Digit	None (Roll Digit Indicator)
Fuel Slider UI	None (Slider)
Fuel Fill UI	fuelFill (Image)
Fuel Txt	fuel_Txt (Text)
Fuel Flow Amplitude	1
Fuel Flow Fill UI	None (Image)
Fuel Flow Txt	fuelFlow_Txt (Text)
Temperature	
Use Temperature	☐
Temperature Amplitu	120
Temperature Off Set	0
Temperature Filter F	0,25
Temperature Roll Di	None (Roll Digit Indicator)
Temperature Pointer	None (Pointer Indicator)
Temperature Slider I	None (Slider)
Temperature Fill UI	None (Image)
Temperature Txt	temperature_Txt (Text)
Flaps & Gear	
Use Flaps	☐
Flaps Filter Factor	0,05
Flaps Slider UI	_flaps_Slider (Slider)
Flaps Fill UI	None (Image)
Flaps Txt	flaps_Txt (Text)
Use Gear	☐
Gear Lights	
Size	3
Element 0	Nose_Green
Element 1	Right_Green
Element 2	Left_Green
Gear Flash Lights	

- "**isActive**" determines if the script should be active.

- "**Aircraft**"(***Transform***) is the references to your aircraft gameObject which will be used to calculate all the values displayed on the instruments. If you leave it empty, then the script will automatically look for the **MainCamera** to calculate from there.

- **ActiveMsg** is the *string* displayed on the console when this instruments panel is activated.

- The others are just references to each GUI element used internaly by the script and doesn't require any setup.

Each section corresponds to a ***Flight Variable*** and the following pattern applies to all of them:

-The *bool* values "**useXXX**" determine if that variable will be used by the script.

- The "**xxxAmplitude**" is a value multiplied to that variable after calculations and before showing it on the GUI. It can work as a ***Scale*** or as a ***unit conversion factor***.

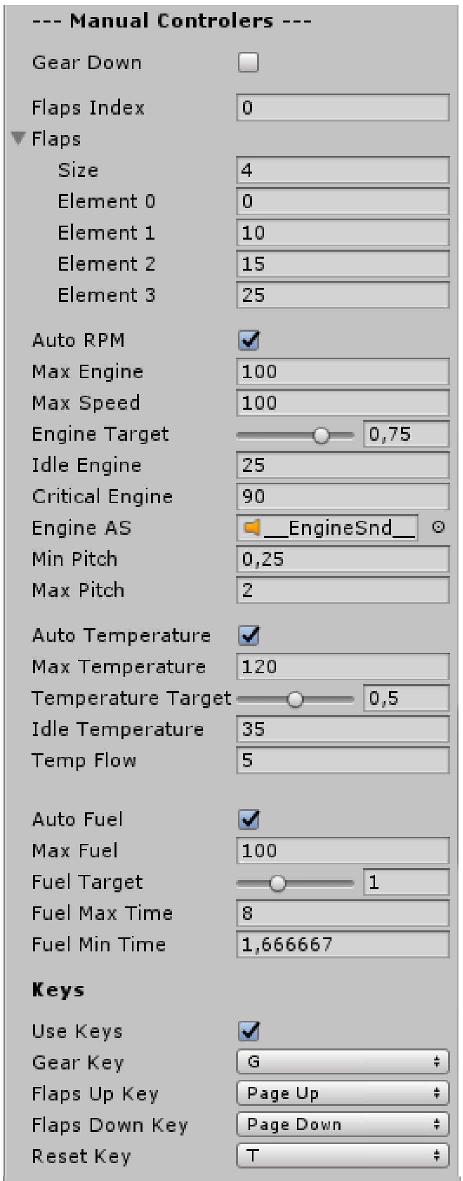
- The "**xxxOffSet**" is a value added to the variable after calculation. It can also be used *to* ***unit conversion*** or simple adjusts.

- All the "**xxxFilterFactor**" values are used to smooth the value shown *(works as a* ***lowpass filter*****)**. If set to **1** it will ***show direct value*** and will have no filtering at all. If set closer to **0** it ***will be smoothed and take more time to reach the final value***.

- The "**CompassBar**" is a reference to the script that controls the compass sliding position to indicate current heading.

- All "**xxxTXT**" are references to a ui *text* component that represent the variable in text format on the **DataConsole**.

- The "**XXPointer**" is a reference to the script that controls the visual indication of that value using the angle of the UI pointer instrument.



--- Manual Controlers ---

Gear Down	☐
Flaps Index	0
Flaps	
Size	4
Element 0	0
Element 1	10
Element 2	15
Element 3	25
Auto RPM	✓
Max Engine	100
Max Speed	100
Engine Target	0,75
Idle Engine	25
Critical Engine	90
Engine AS	__EngineSnd__
Min Pitch	0,25
Max Pitch	2
Auto Temperature	✓
Max Temperature	120
Temperature Target	0,5
Idle Temperature	35
Temp Flow	5
Auto Fuel	✓
Max Fuel	100
Fuel Target	1
Fuel Max Time	8
Fuel Min Time	1,666667

Keys

Use Keys	✓
Gear Key	G
Flaps Up Key	Page Up
Flaps Down Key	Page Down
Reset Key	T

- On **the Manual Controllers** section you can set the instruments value manually or let the script calculate it automatically. Also, you can configure some of the values for the Flaps, Engine RPM, Fuel and Key Controls.

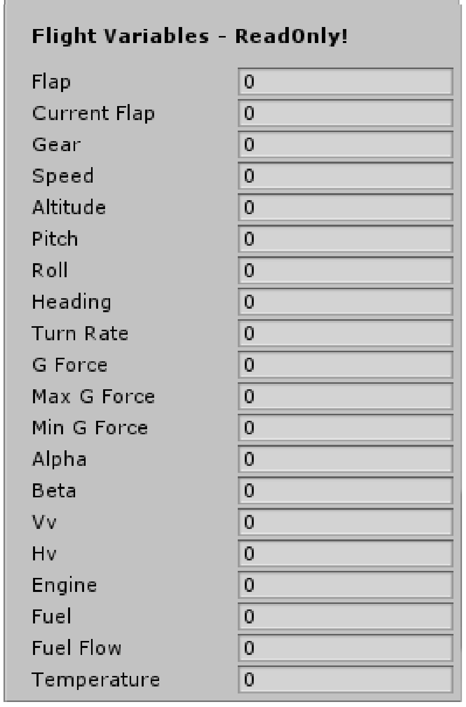
-The ***bool*** values "**AutoXXX**" determine if that variable will be calculated by the script or if you wish to set it manually, for instance, if you already have an asset that calculates these values and you wish *only to use the GUI* to show that value.

- The **"xxxTarget"** sliders are normalized between ***0-1*** and let you *set the desired value to be shown* by the instrument. In auto mode these values will be automatically controlled by the script.

- The **"EngineAS"** is a reference to the **AudioSource** used for the **engine sounds**. The **audio pitch** will automatically follow from **Min** to **MaxPitch** value depending on current engine RPM. If you already have an asset that controls engine sounds for you, then you can leave this field ***empty***.

- The **FuelMaxTime** and **FuelMinTime** determine how much time in minutes takes *to consume 100% of the fuel* when **Engine RPM** is at **Idle** and **Máx RPM** *and interpolates quadratically* in-between them according to current **EngineRPM**.

- The Keys sections lets you customize what keys can be used for manipulating the GUI instruments controls. Set the **UseKeys** to ***false*** if you do not want the player to change anything during gameplay.



Flight Variables - ReadOnly!

Flap	0
Current Flap	0
Gear	0
Speed	0
Altitude	0
Pitch	0
Roll	0
Heading	0
Turn Rate	0
G Force	0
Max G Force	0
Min G Force	0
Alpha	0
Beta	0
Vv	0
Hv	0
Engine	0
Fuel	0
Fuel Flow	0
Temperature	0

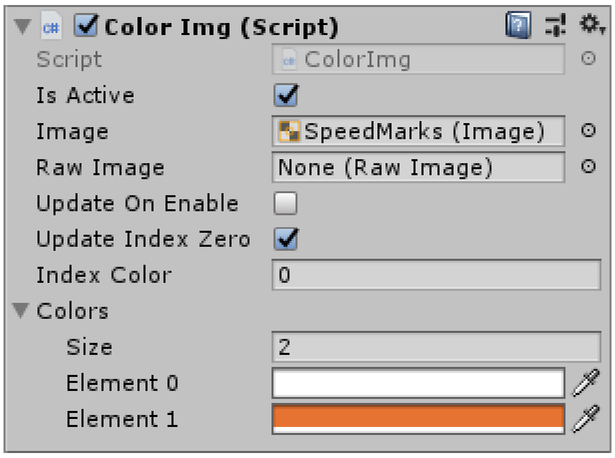
- The last section shows **Current Variables** values for all the variables in real time inside the Editor.

This is just for debug or tweeking and are **ReadOnly**!

You can also visualize these variables during runtime using the **(DATA CONSOLE)**.

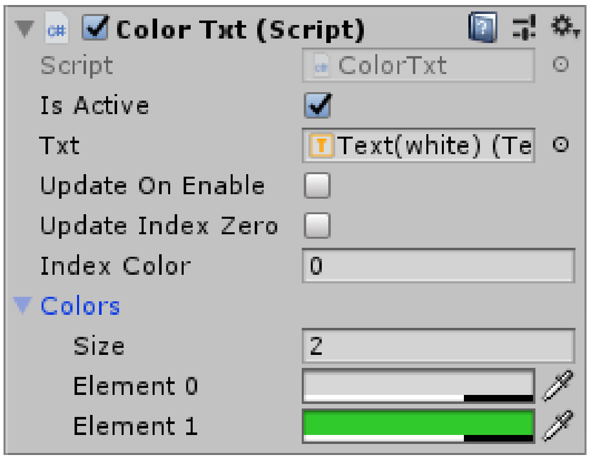
➢ **(Extra Feature) Custom Color (ColorImg + ColorTxt Scripts):**

If you want to improve the asset, it includes 2 scripts that you can use to ***customize*** colors during runtime *for **all instruments*** by editing the fields of the **ColorImg *script*** as shown below:



Color Img (Script)

Script	ColorImg
Is Active	✓
Image	SpeedMarks (Image)
Raw Image	None (Raw Image)
Update On Enable	☐
Update Index Zero	✓
Index Color	0
Colors	
Size	2
Element 0	⬜
Element 1	🟧



Color Txt (Script)

Script	ColorTxt
Is Active	✓
Txt	Text(white) (Te
Update On Enable	☐
Update Index Zero	☐
Index Color	0
Colors	
Size	2
Element 0	⬜
Element 1	🟩

Then you can call the methods ***setColor*** or ***toogleColor*** and it will change or toogle the image component color between **Element0** and **Element1**.

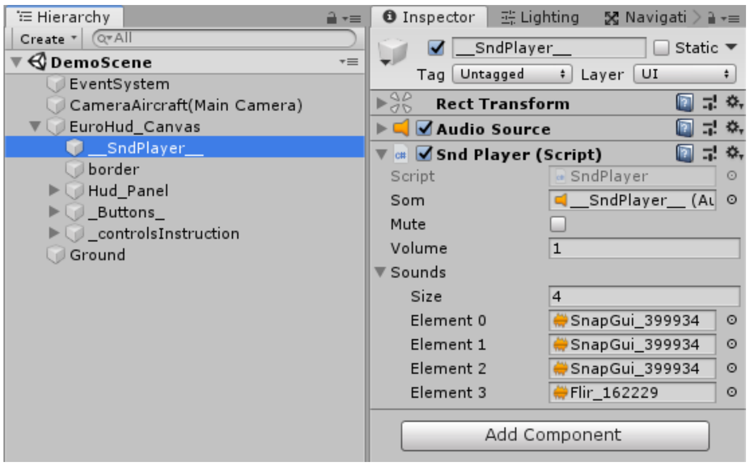
The option **"UpdateIndexZero"** will automatically allocate the current color used in the editor for the component to the **Element0** during runtime.

A ***quick way*** to change light colors of all components is to type **"ColorImg"** on the **Hierarchy search field** and then selet all the gameObjects containing it to edit then simultaneously to your desired color.

Exactly the same applies to **Text** C**olors** using the **ColorTxt** script in the same matter.

➢ **Secondary Components (Sound and Message Console) :**

If you wish extra functionalities, you can make use of these components by script calling statics methods:



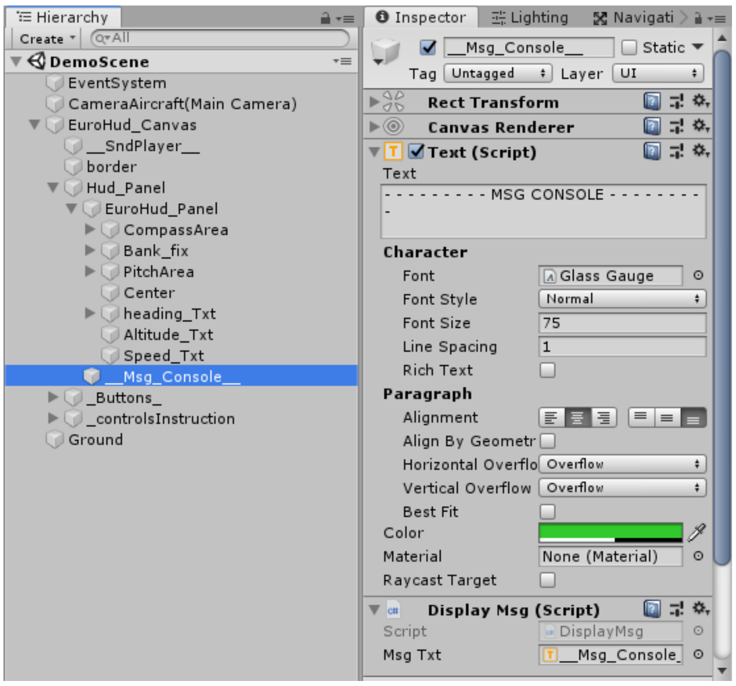
DemoScene
- EventSystem
- CameraAircraft(Main Camera)
- EuroHud_Canvas
  - __SndPlayer__
  - border
  - Hud_Panel
  - _Buttons_
  - _controlsInstruction
- Ground

__SndPlayer__ — Static
Tag: Untagged — Layer: UI

Rect Transform
Audio Source
Snd Player (Script)
- Script: SndPlayer
- Som: __SndPlayer__ (Au
- Mute: ☐
- Volume: 1
- Sounds
  - Size: 4
  - Element 0: SnapGui_399934
  - Element 1: SnapGui_399934
  - Element 2: SnapGui_399934
  - Element 3: Flir_162229

```
-public static void play(int index);
-public static void play(AudioClip clip, float volume = 1f);
```

(Plays the sound listed on array Sounds with index position or the audioclip itself)



DemoScene
- EventSystem
- CameraAircraft(Main Camera)
- EuroHud_Canvas
  - __SndPlayer__
  - border
  - Hud_Panel
    - EuroHud_Panel
      - CompassArea
      - Bank_fix
      - PitchArea
      - Center
      - heading_Txt
      - Altitude_Txt
      - Speed_Txt
    - __Msg_Console__
  - _Buttons_
  - _controlsInstruction
- Ground

__Msg_Console__ — Static
Tag: Untagged — Layer: UI

Rect Transform
Canvas Renderer
Text (Script)
- Text: - - - - - - - - - MSG CONSOLE - - - - - - - - -
- Character
  - Font: Glass Gauge
  - Font Style: Normal
  - Font Size: 75
  - Line Spacing: 1
  - Rich Text: ☐
- Paragraph
  - Alignment
  - Align By Geometr: ☐
  - Horizontal Overflo: Overflow
  - Vertical Overflow: Overflow
  - Best Fit: ☐
- Color: 🟩
- Material: None (Material)
- Raycast Target: ☐

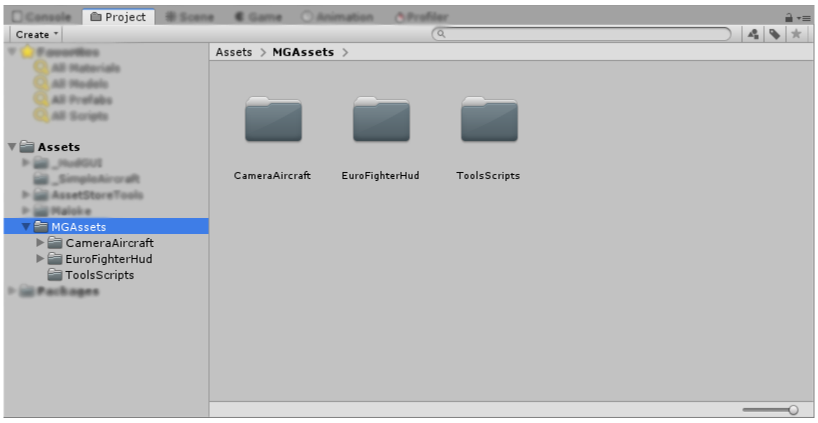
Display Msg (Script)
- Script: DisplayMsg
- Msg Txt: __Msg_Console_

```
- public void displayMsg(string msg = "");
- public void displayQuickMsg(string msg = "");
- public static void show(string msg = "", float timed = 0);
```

Displays a string message on the bottom of the HUD for an amount of seconds (quick is 5s).

➢ Asset's Folders Organization:

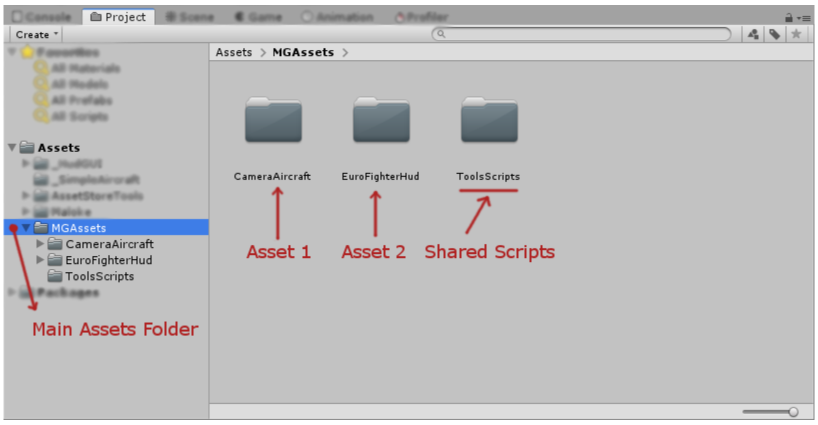
All assets and packages from MalokeGamesAssets will be downloaded/unpacked to a folder called **"MGAssets"** inside the Unity's **"Assets"** root:



Inside **"MGAssets"** you will find a separate folder for each asset package and all their specific resources (like data, scripts, textures, sprites, prefabs, demo scenes and so on...) will be found inside and organized in their respective subfolders.

Notice that some assets may use "under the hood" some general scripts and shared funcionalities, so for this reason (and to avoid duplicity or accidental deletion) you will find all this shared tools inside a folder called **"ToolsScripts"**.

Feel free to explore and use them on your projects too, they are simple but handy!


Project
Create
Assets > MGAssets >
Assets
MGAssets
CameraAircraft
EuroFighterHud
ToolsScripts
CameraAircraft
EuroFighterHud
ToolsScripts
Asset 1
Asset 2
Shared Scripts
Main Assets Folder