MALOKE GAMES ASSETS

Eurofigther GUI

HUD (Head Up Display)

HMD (Helmet-Mounted Display)

- **Contact: Maloke7-Games@yahoo.com.br**
- **Portfolio:** https://maloke.itch.io/

- **Quick Instructions:**

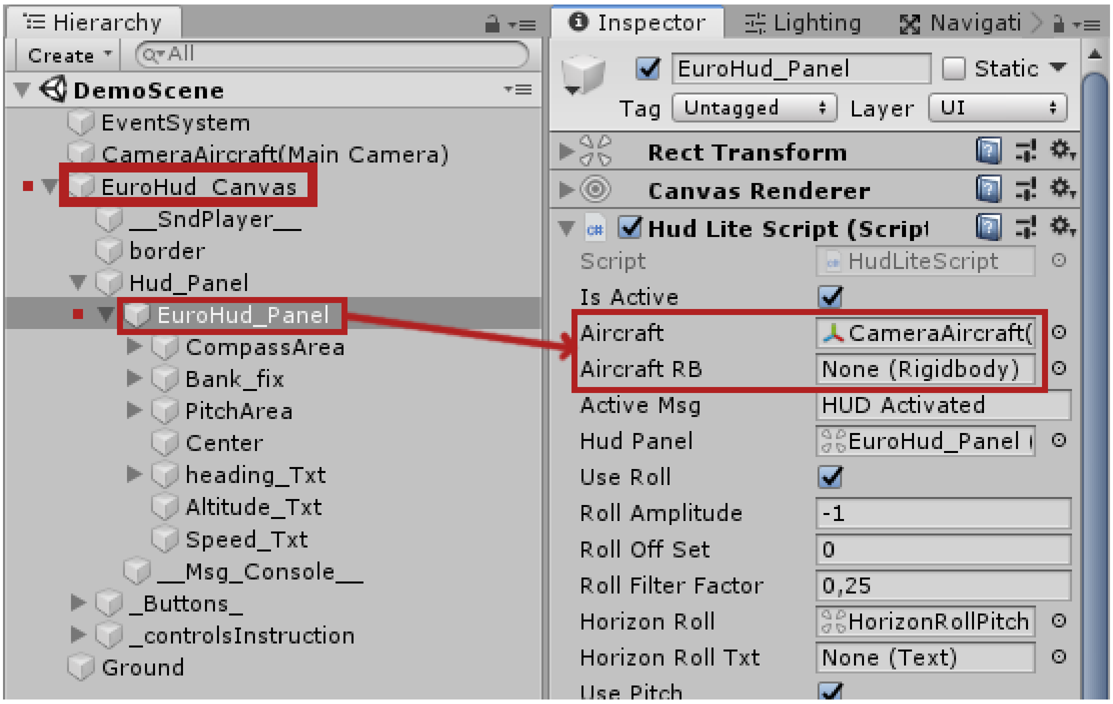
You can find 2 DemoScenes (one for **HUD** and other for **HMD** mode) with this asset configured and working straight away. The demo scene uses a very basic camera movement script applied to the main camera. It emulates an aircraft movement to help you visualize properly how it works.

If you want to start using the Eurofigther HUD/HMD on your own project you just need to link a reference of your aircraft's *Transform* to the main script.

Just follow these 3 simple steps:

- Drop on your scene the "**EuroHUD_Canvas**" or "**EuroHMD_Canvas**" prefab.

- Then locate the main script "**HudLiteScript**" on the prefab as show in the image:



☑ EuroHud_Panel □ Static
Tag Untagged Layer UI

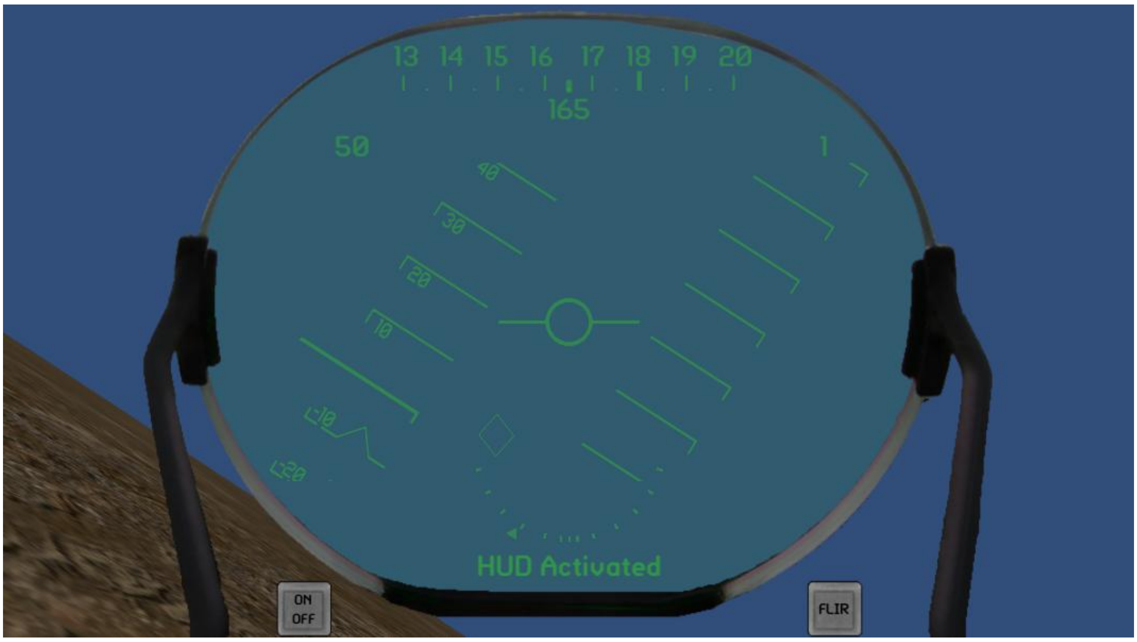
Script HudLiteScript
Is Active ☑
Aircraft CameraAircraft(
Aircraft RB None (Rigidbody)
Active Msg HUD Activated
Hud Panel EuroHud_Panel (
Use Roll ☑
Roll Amplitude -1
Roll Off Set 0
Roll Filter Factor 0,25
Horizon Roll HorizonRollPitch
Horizon Roll Txt None (Text)
Use Pitch ☑

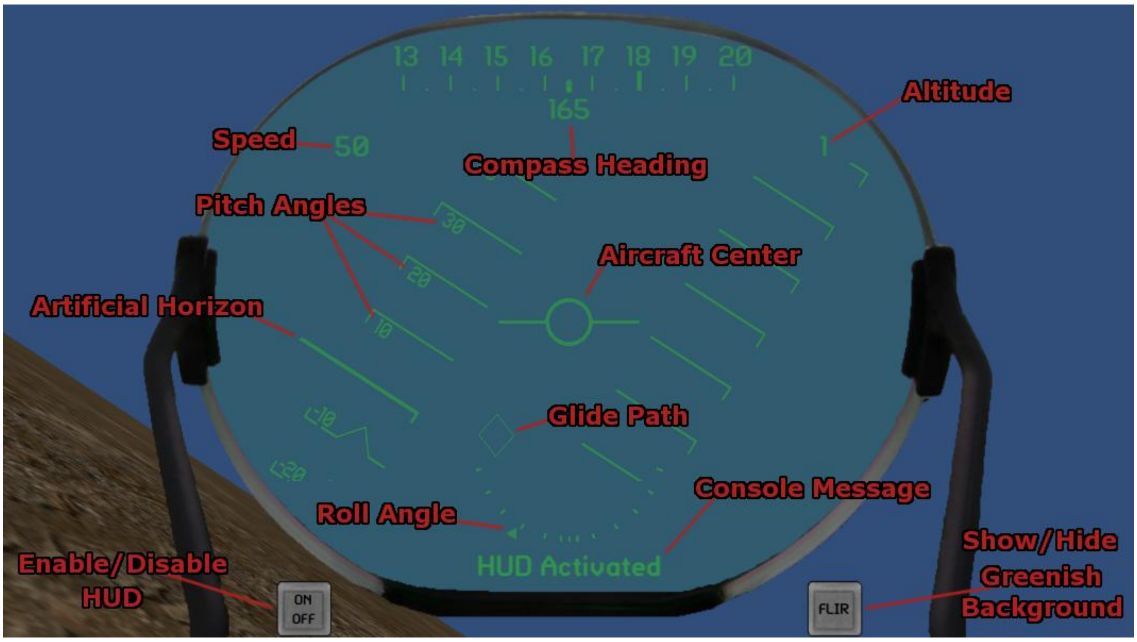
- Then simply drag your aircraft object to the field "**Aircraft**" and you are ready to go. If your aircraft haves a ***RigidBody*** then drag it too to the "**AircraftRB"** field. Using a ***RB*** will ensure more precision on the physics calculations, but the script can work using ***Transform*** only.

That's it, now you are ready to go!

If you want to know more about this asset you can find extra information further on this document.

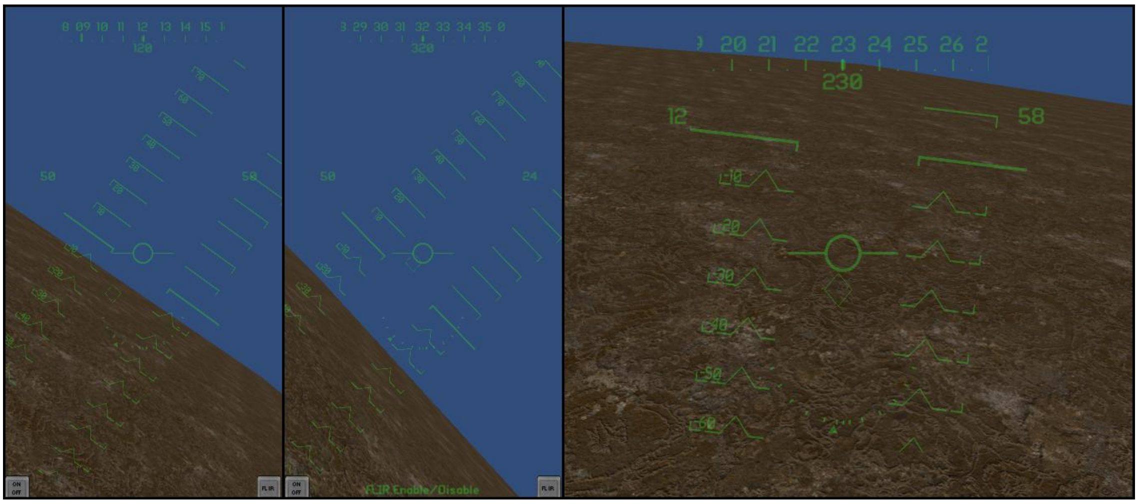
- **Eurofighter HUD Symbology:**



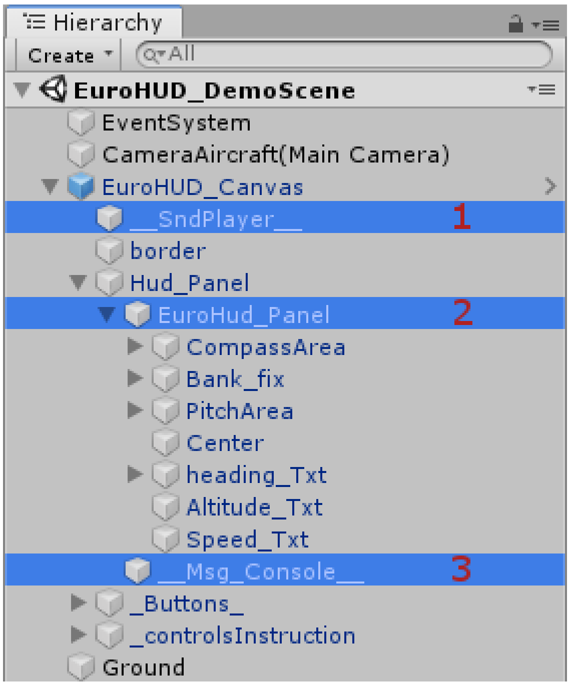


- **HMD Mode (Helmet-Mounted Display):**

In this mode, you don't have the HUD glass and the GUI elements adjust to fit the whole screen. It can be used as an Arcade version or even to simulate a helmet-mounted display (HMD) like those used in the new F-35 figther.



- **Main Components:**



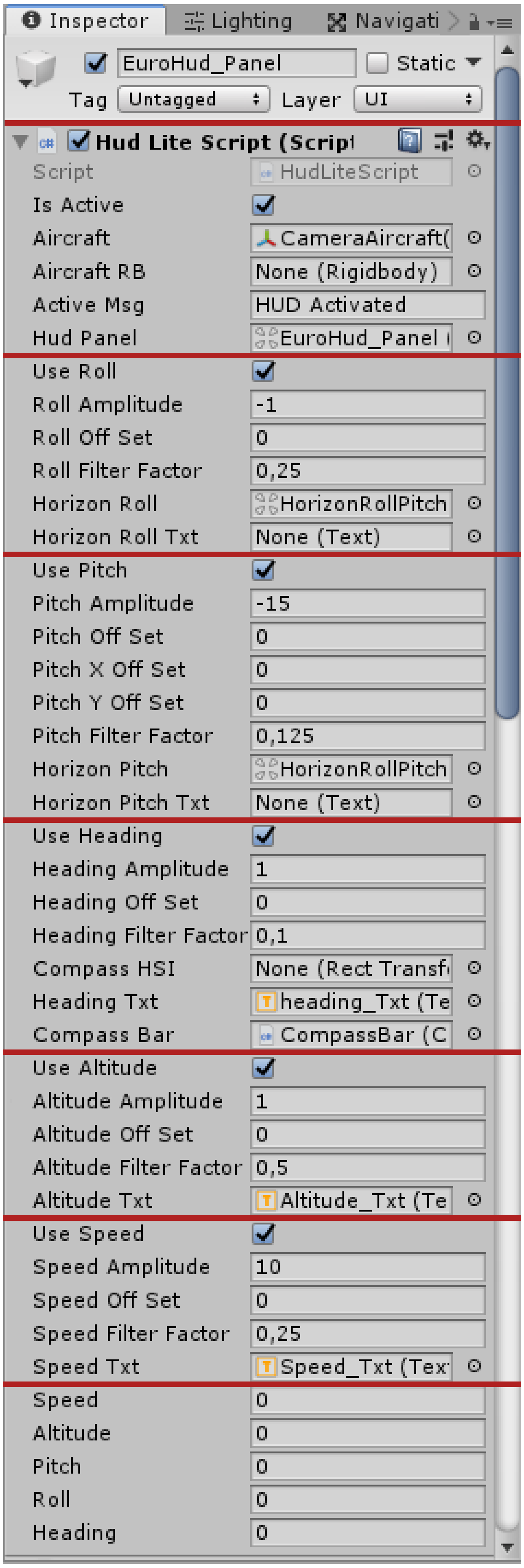
1- Contains script "**SndPlayer**" that manages the HUD sounds.

2- The main script "**HudLiteScript**" which controls the whole HUD and calculations.

3- Contains the script "**DisplayMsg**" which is responsible to send/show messages to the console on the bottom of the HUD.

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

➢ HudLiteScript - Structure in Editor:



☑ EuroHud_Panel □ Static

Tag Untagged Layer UI

☑ Hud Lite Script (Script)

Script	HudLiteScript
Is Active	☑
Aircraft	CameraAircraft(
Aircraft RB	None (Rigidbody)
Active Msg	HUD Activated
Hud Panel	EuroHud_Panel (
Use Roll	☑
Roll Amplitude	-1
Roll Off Set	0
Roll Filter Factor	0,25
Horizon Roll	HorizonRollPitch
Horizon Roll Txt	None (Text)
Use Pitch	☑
Pitch Amplitude	-15
Pitch Off Set	0
Pitch X Off Set	0
Pitch Y Off Set	0
Pitch Filter Factor	0,125
Horizon Pitch	HorizonRollPitch
Horizon Pitch Txt	None (Text)
Use Heading	☑
Heading Amplitude	1
Heading Off Set	0
Heading Filter Factor	0,1
Compass HSI	None (Rect Transf
Heading Txt	heading_Txt (Te
Compass Bar	CompassBar (C
Use Altitude	☑
Altitude Amplitude	1
Altitude Off Set	0
Altitude Filter Factor	0,5
Altitude Txt	Altitude_Txt (Te
Use Speed	☑
Speed Amplitude	10
Speed Off Set	0
Speed Filter Factor	0,25
Speed Txt	Speed_Txt (Tex
Speed	0
Altitude	0
Pitch	0
Roll	0
Heading	0

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

- "**Aircraft**"(***Transform***) and "**AircraftRB**"(***RigidBody***) are the references to your flying gameObject which will be used to calculate all the values displayed on the GUI.

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

-**HudPanel** is an internal reference for the *panel* itself.

Each section below corresponds to a flight variable and the following pattern applies to them:

-The *bool* values "**useXXX**" determine if that variable will be calculated.

- The "**xxxAmplitude**" is a value multiplied to that variable after calculations. 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 be used to unit conversion or to fix objects orientation properly.

- All the "**xxxFilterFactor**" values are used to smooth the value shown *(works as a **lowpass filter**)*. If set to 1 it will have no filtering at all.

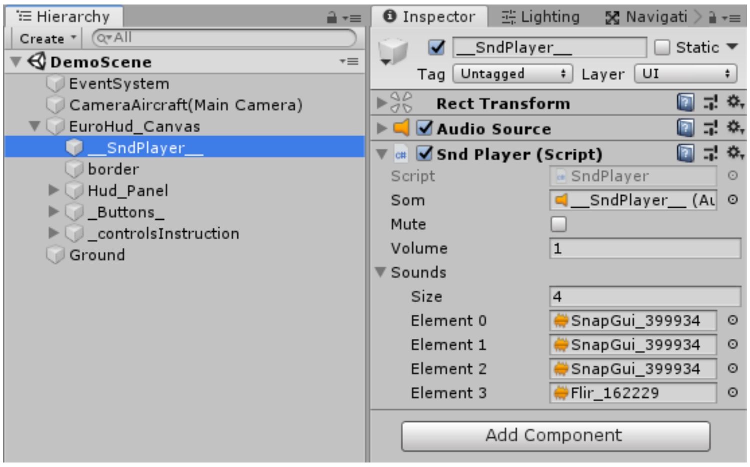
- The "**Horizonxxx**" and other ***RectTransform*** entries are references to the GUI object used to represent that variable visually.

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

In this last section you can find the current value of all the flight variables (**Read Only**!)

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

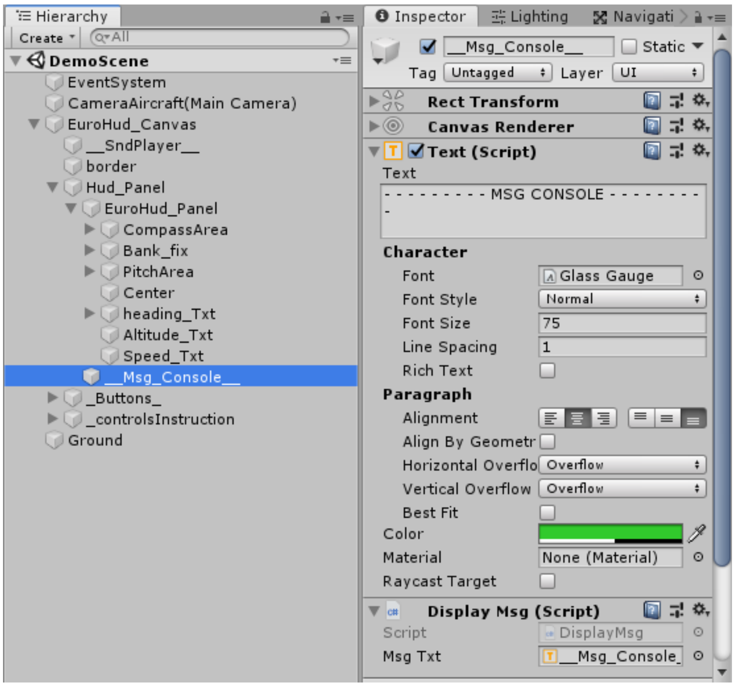
If you wish extra functionalities, you can make use of these components by script calling statics methods:



☑ __SndPlayer__ □ Static
Tag Untagged Layer UI
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)



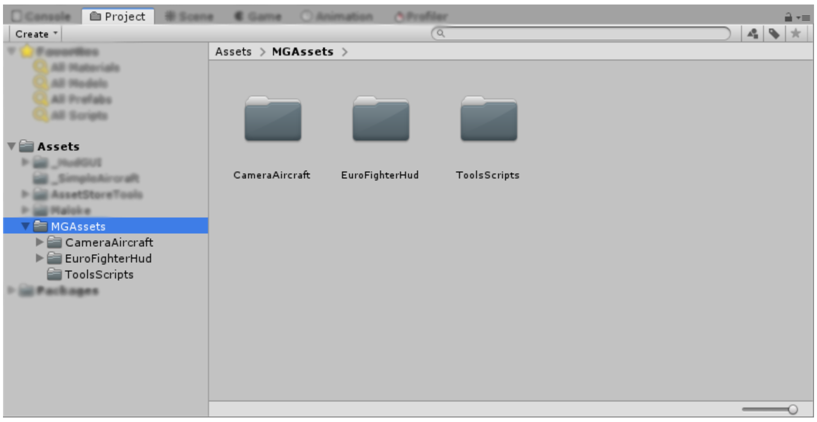
☑ __Msg_Console__ □ Static
Tag Untagged Layer UI
Text
- - - - - - - - - MSG CONSOLE - - - - - - - -
-
Character
Font Glass Gauge
Font Style Normal
Font Size 75
Line Spacing 1
Rich Text □
Paragraph
Alignment
Align By Geometr □
Horizontal Overflo Overflow
Vertical Overflow Overflow
Best Fit □
Color
Material None (Material)
Raycast Target □
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:**

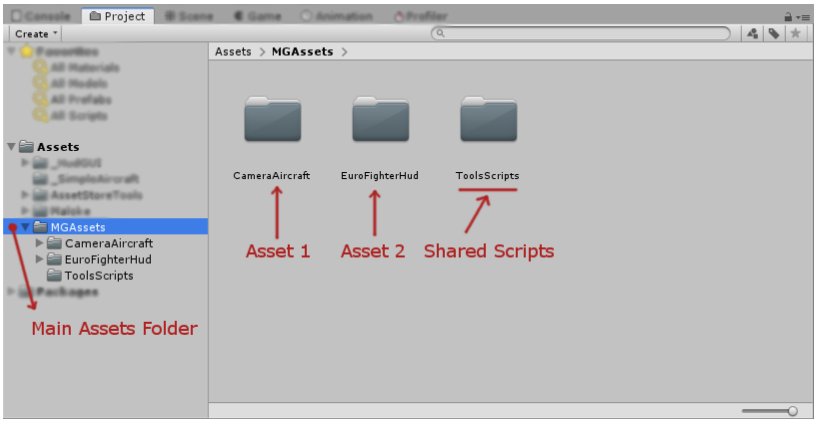
All assets and packages from us will be always downloaded/unpacked to a folder called **"MGAssets"** inside the Unity's **"Assets"** root:



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