MALOKE GAMES

ASSETS

Advanced Aircraft HUD - GUI Pack

High Fidelity HUDs (Head Up Display) for:

Eurofighter

American Standard
(Based mainly on the F-18)

Old Russian HUD Style

Modern Russian

Russian MIG
(Somewhat in between the old and modern ones)

Civilian Aviation Standard

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

➢ **Quick Instructions:**

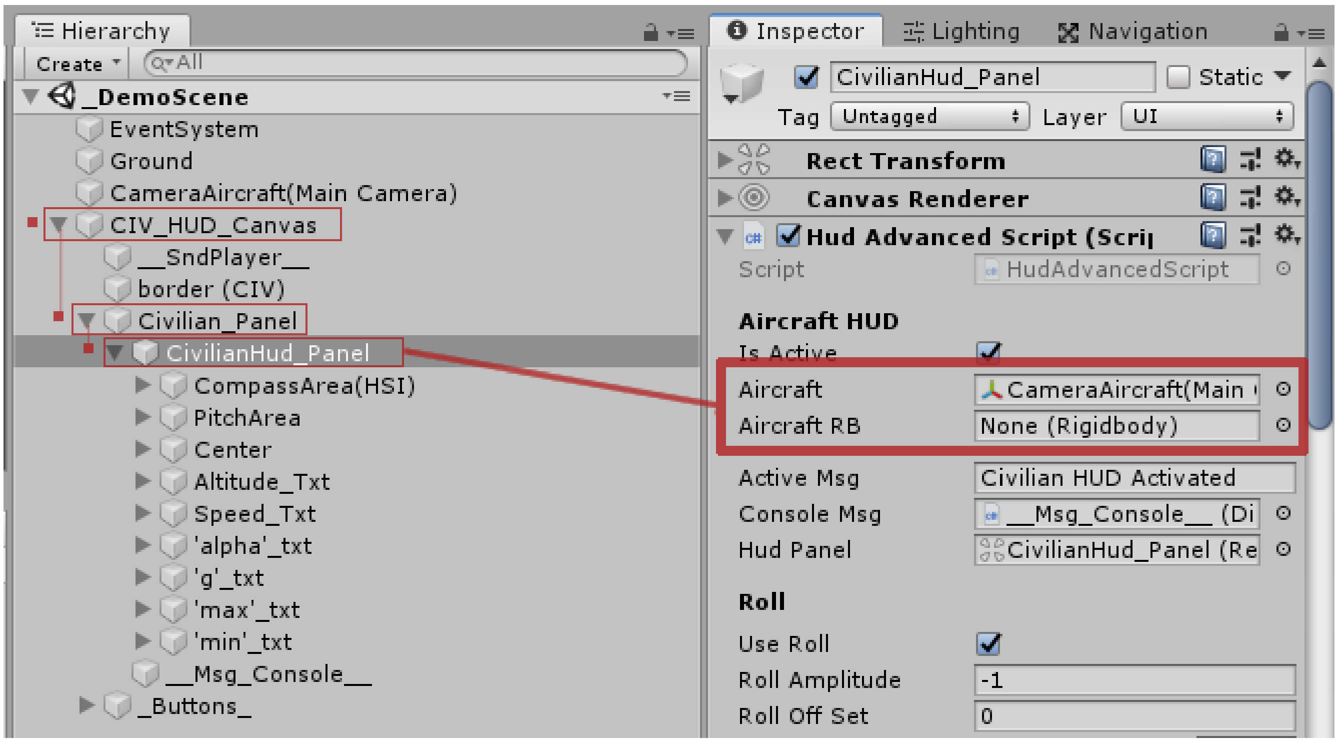
You can find a DemoScene with this asset configured and working straight away for the 6 different HUDs of this pack. The demo uses a very basic camera movement script applied to the main camera that emulates an aircraft movement just to help you visualize properly how it works.

If you want to use on your own project/aircraft you just need to link a reference of your aircraft's *Transform* and/or *RigidBody* to the main script. If you leave these fields empty it will automatically look for the *MainCamera* in the scene.

Just follow these 3 simple steps:

1- Drop on your scene any of the "**HUD.prefab**" prefab.

2- Locate the main script "**HudAdvancedScript**" as shown in the image:



```
_DemoScene
  EventSystem
  Ground
  CameraAircraft(Main Camera)
  CIV_HUD_Canvas
    __SndPlayer__
    border (CIV)
    Civilian_Panel
      CivilianHud_Panel
        CompassArea(HSI)
        PitchArea
        Center
        Altitude_Txt
        Speed_Txt
        'alpha'_txt
        'g'_txt
        'max'_txt
        'min'_txt
      __Msg_Console__
    _Buttons_
```

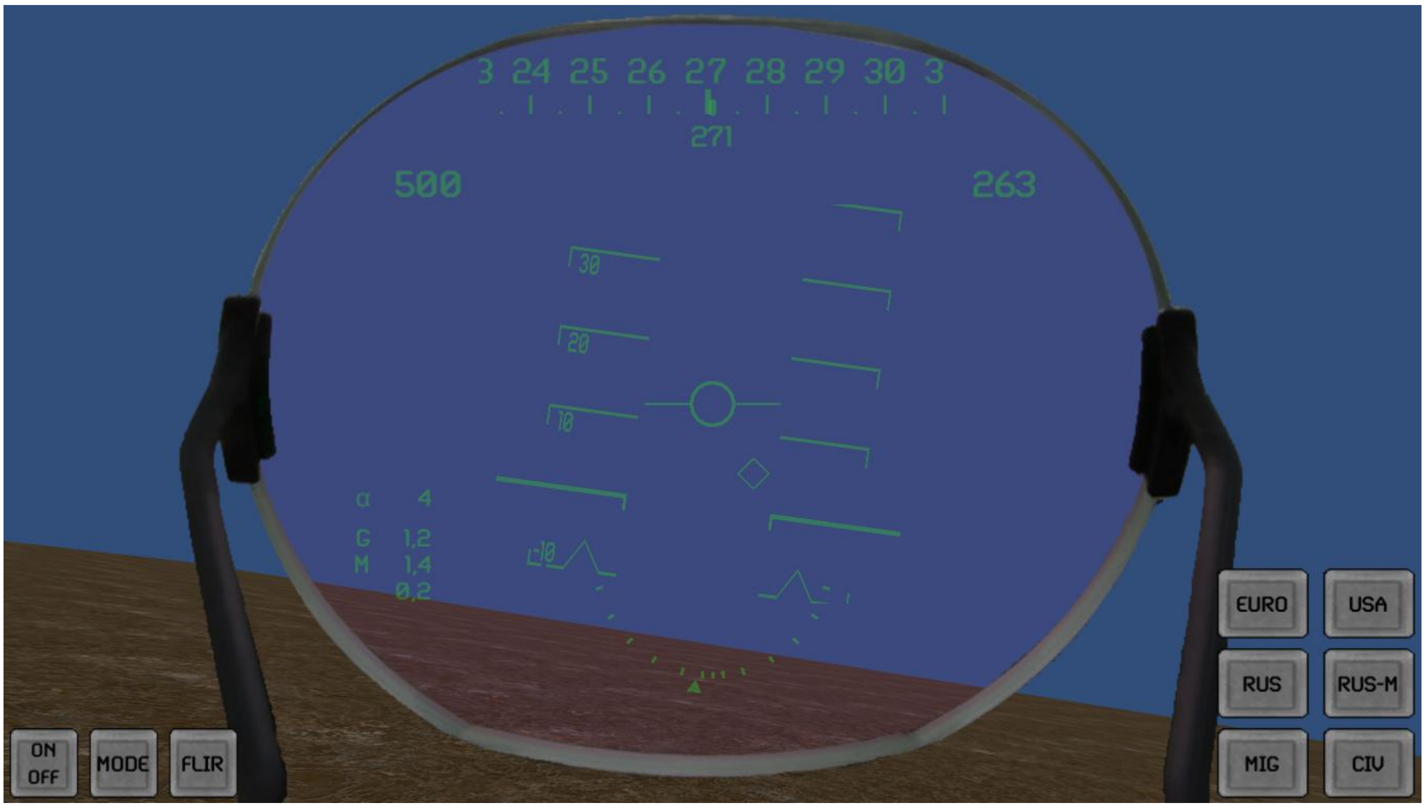
```
CivilianHud_Panel                          Static
Tag Untagged        Layer UI
Rect Transform
Canvas Renderer
Hud Advanced Script (Scri
Script              HudAdvancedScript
Aircraft HUD
Is Active           ✓
Aircraft            CameraAircraft(Main
Aircraft RB         None (Rigidbody)
Active Msg          Civilian HUD Activated
Console Msg         __Msg_Console__ (Di
Hud Panel           CivilianHud_Panel (Re
Roll
Use Roll            ✓
Roll Amplitude      -1
Roll Off Set        0
```

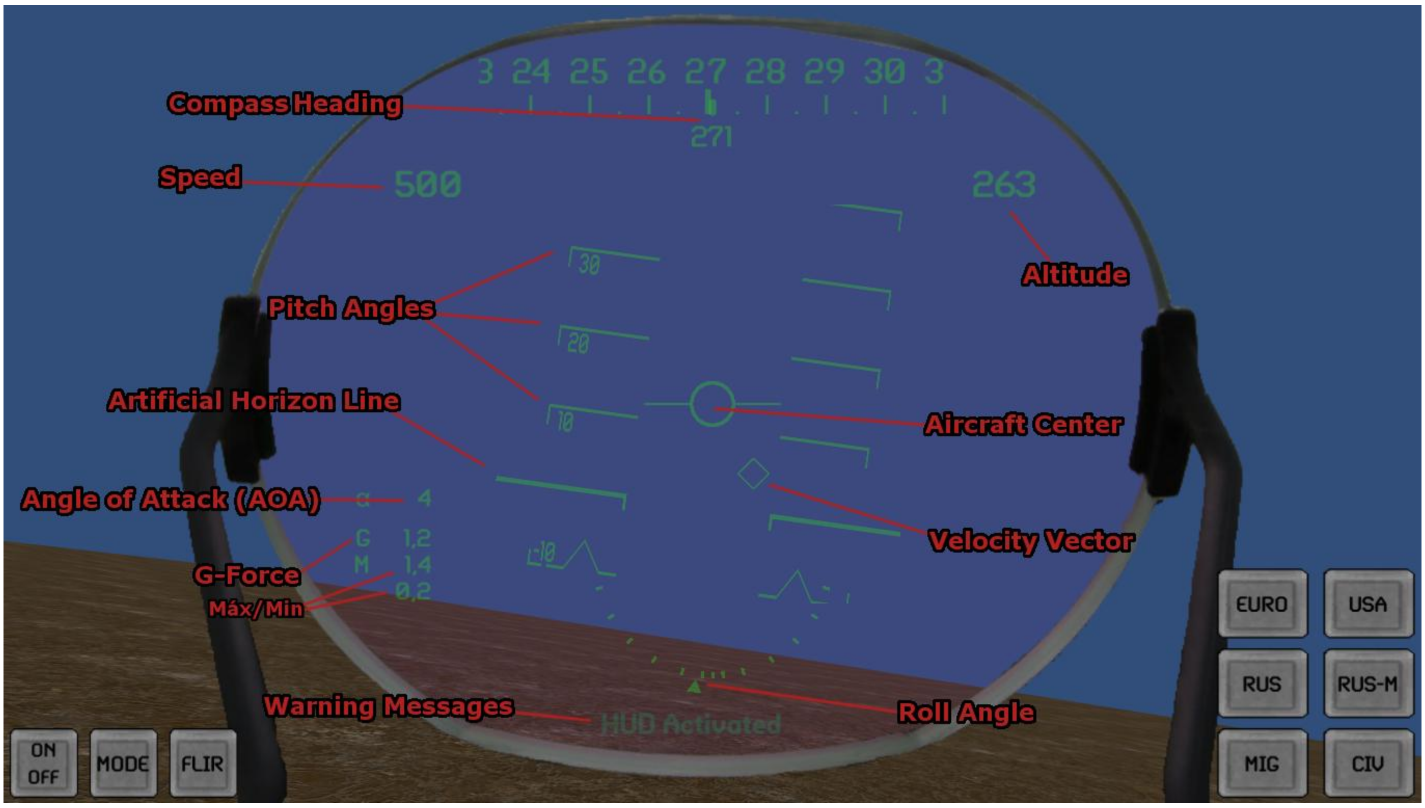
3- Then simply drag your aircraft object to the field "**Aircraft**" and you are ready to go. If your aircraft haves a ***RigidBody*** then drag it too to the "**AircraftRB"** field. Using an ***RB*** will ensure more precision on some physics calculations, but the script can work using ***Transform***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

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

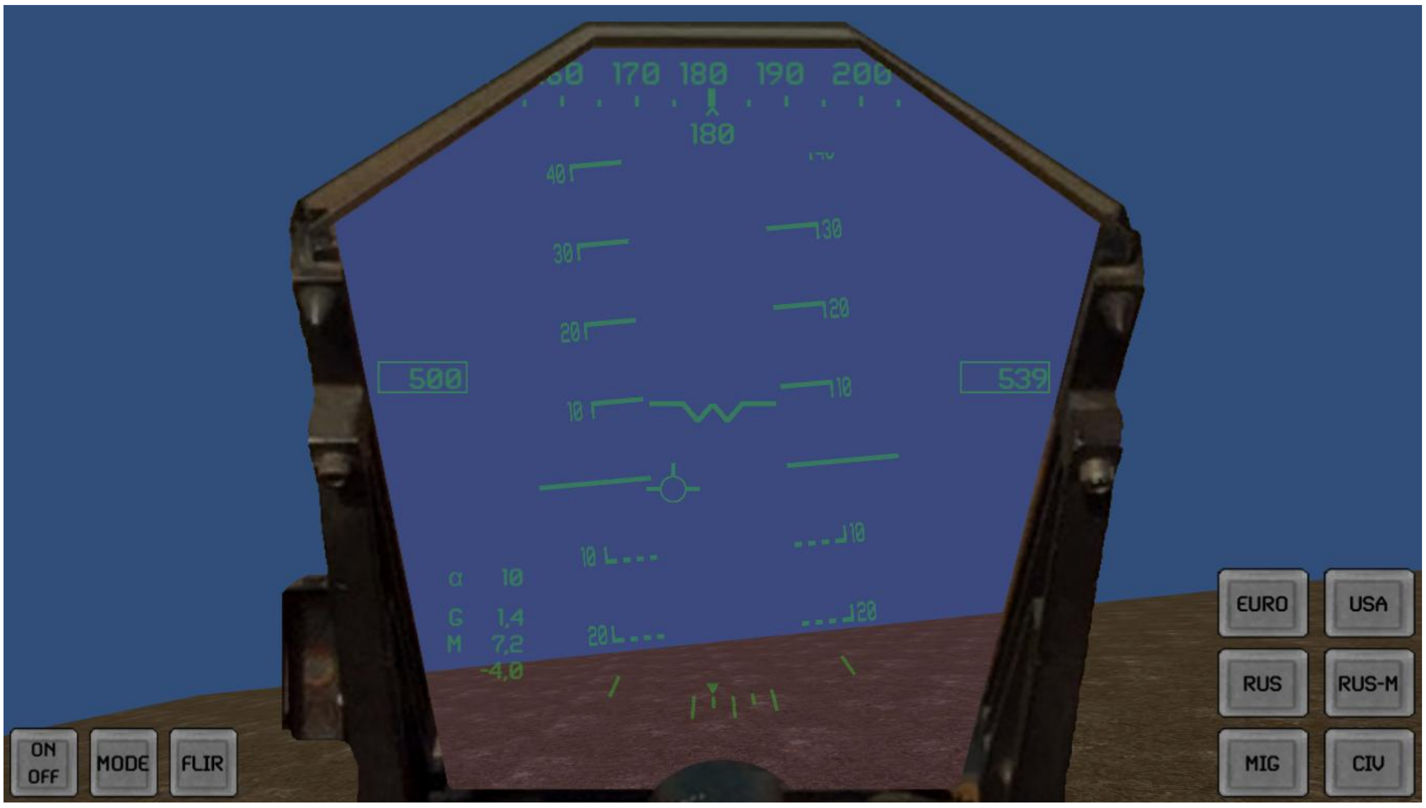
If you want to know more about this asset you can find extra information further on this document.

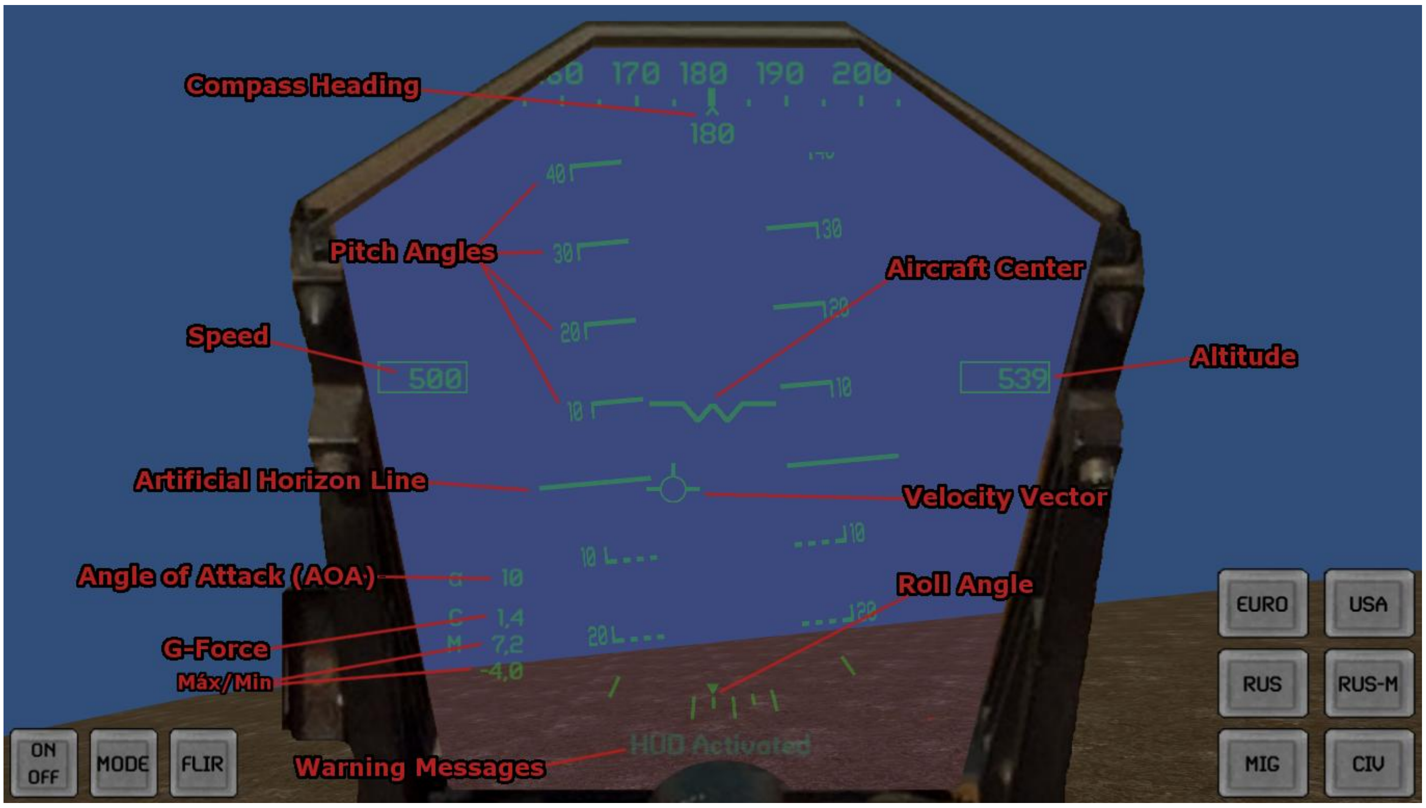
- **Eurofighter HUD Symbology:**



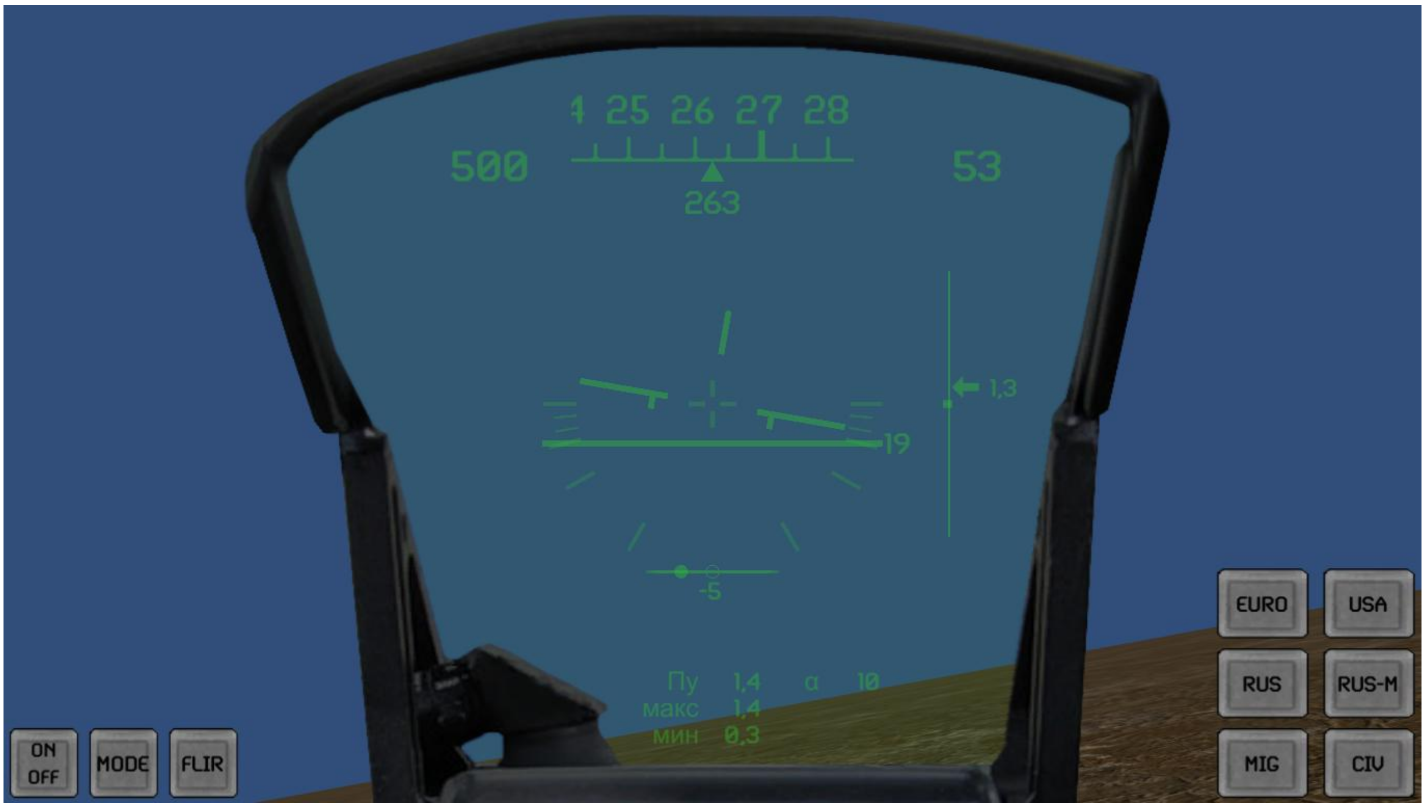


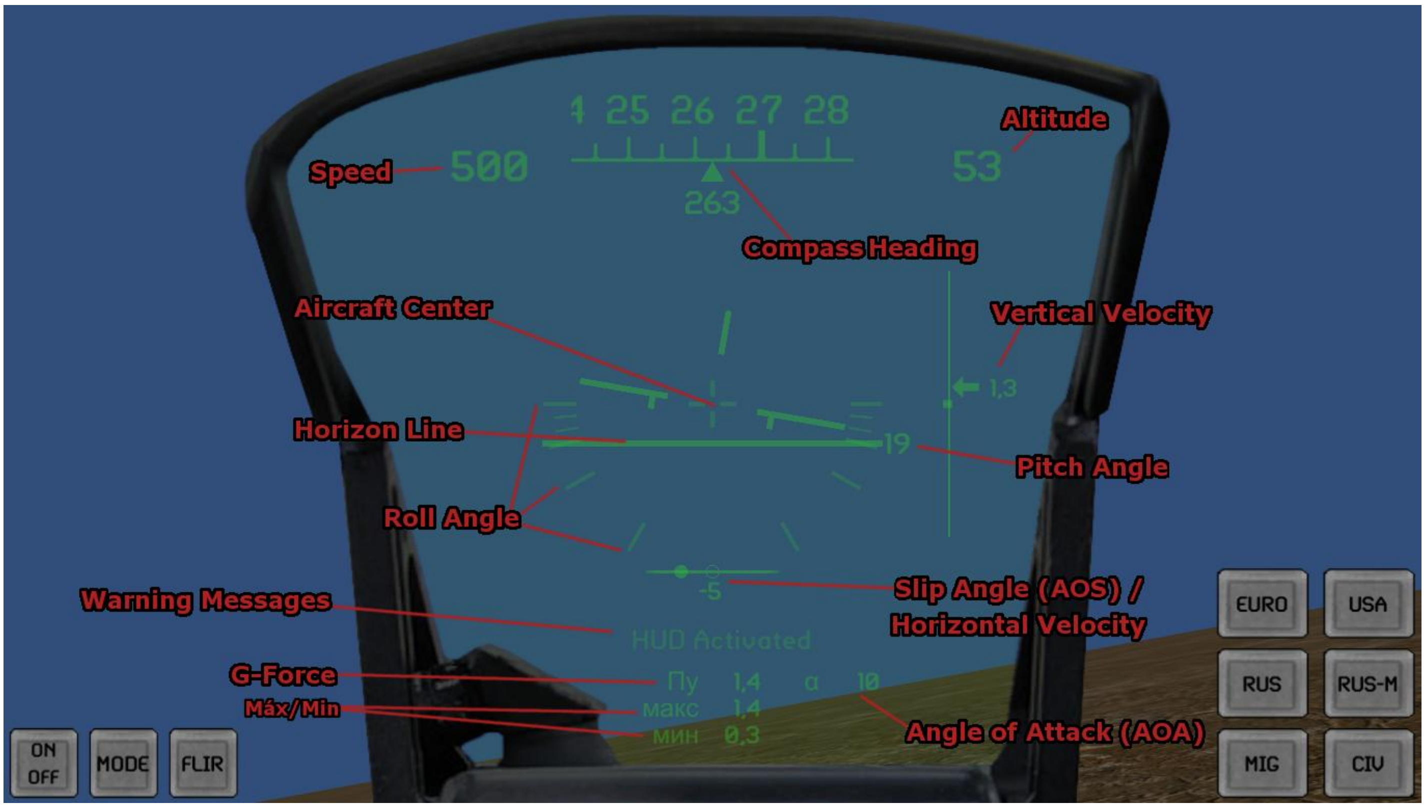
➢ **American HUD Symbology:**



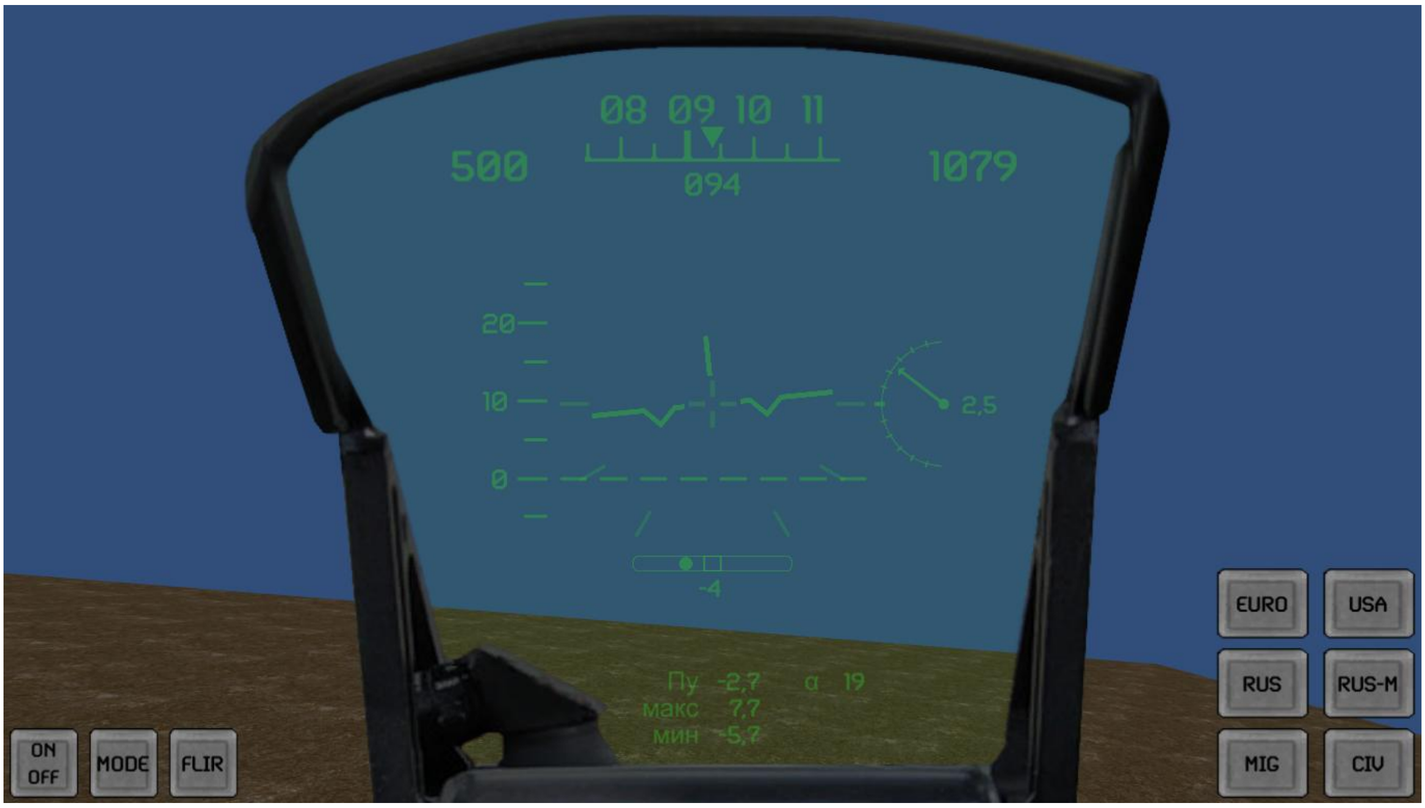


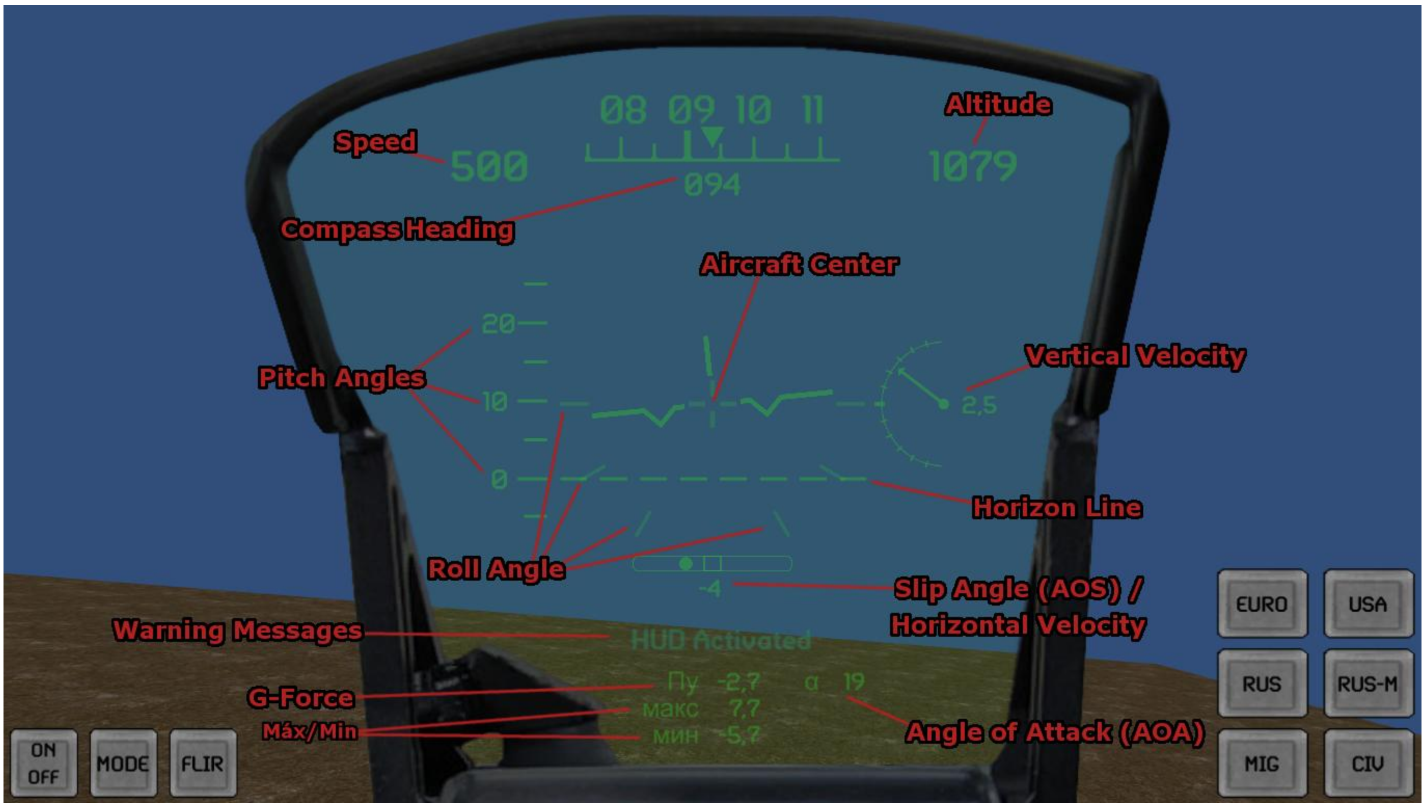
➢ **Russian Old HUD Symbology:**



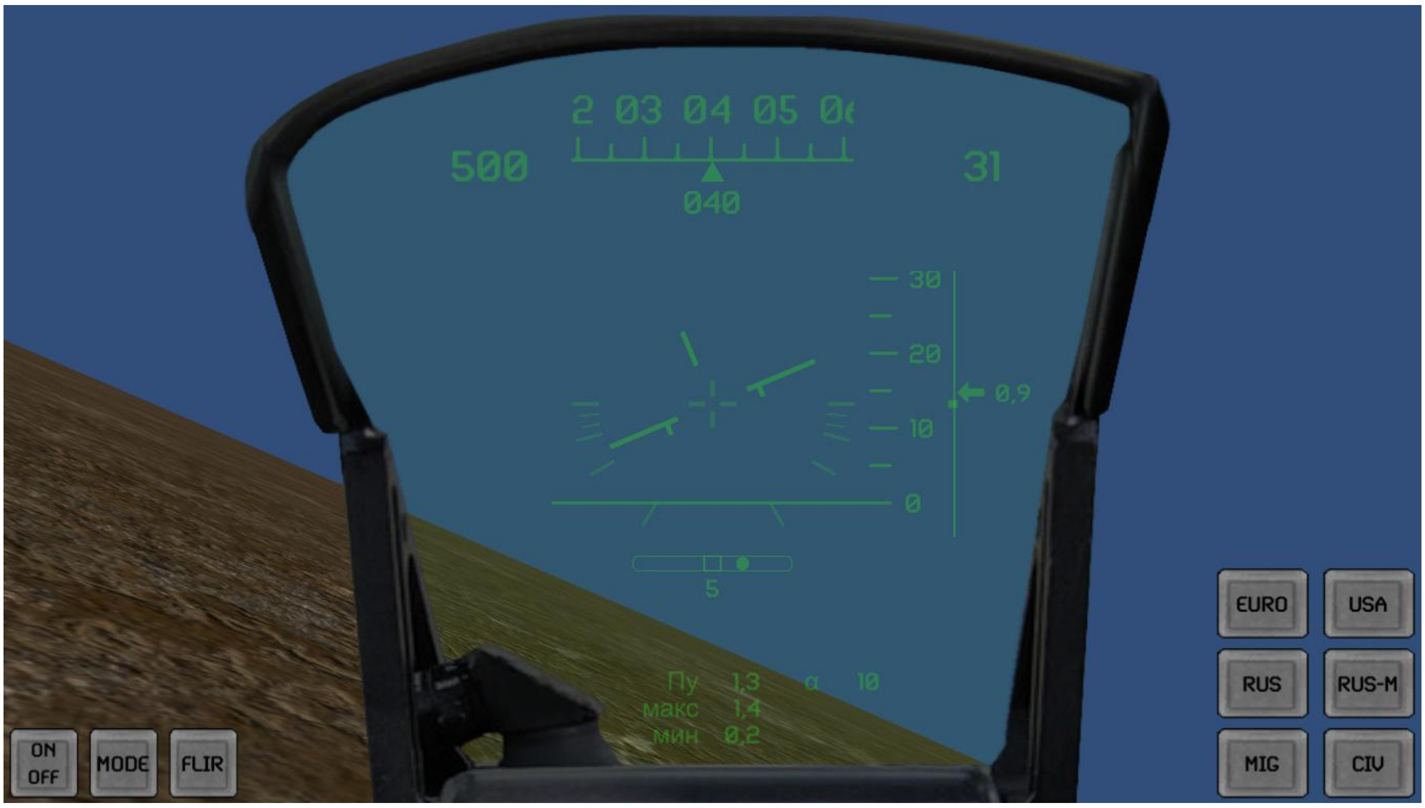


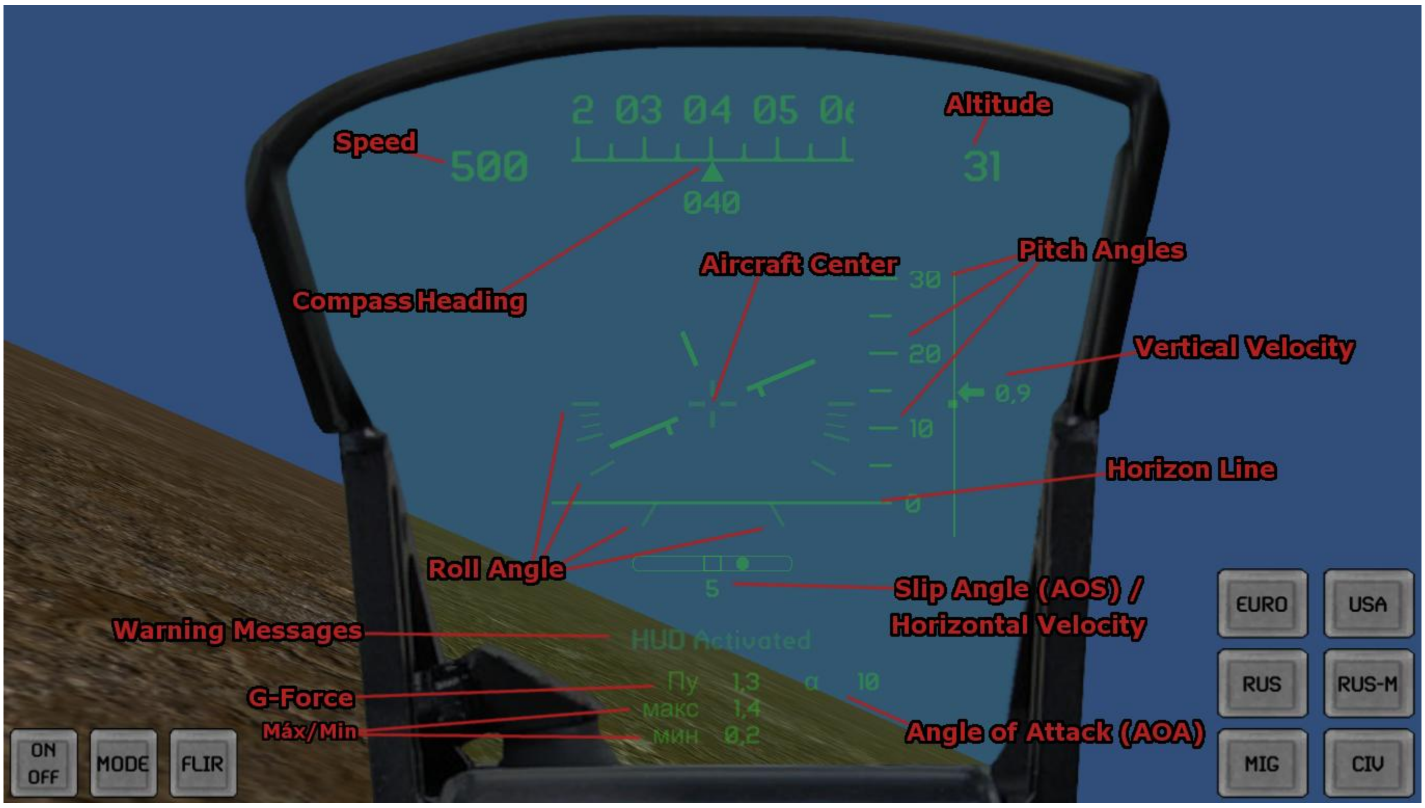
- **Russian Modern HUD Symbology:**



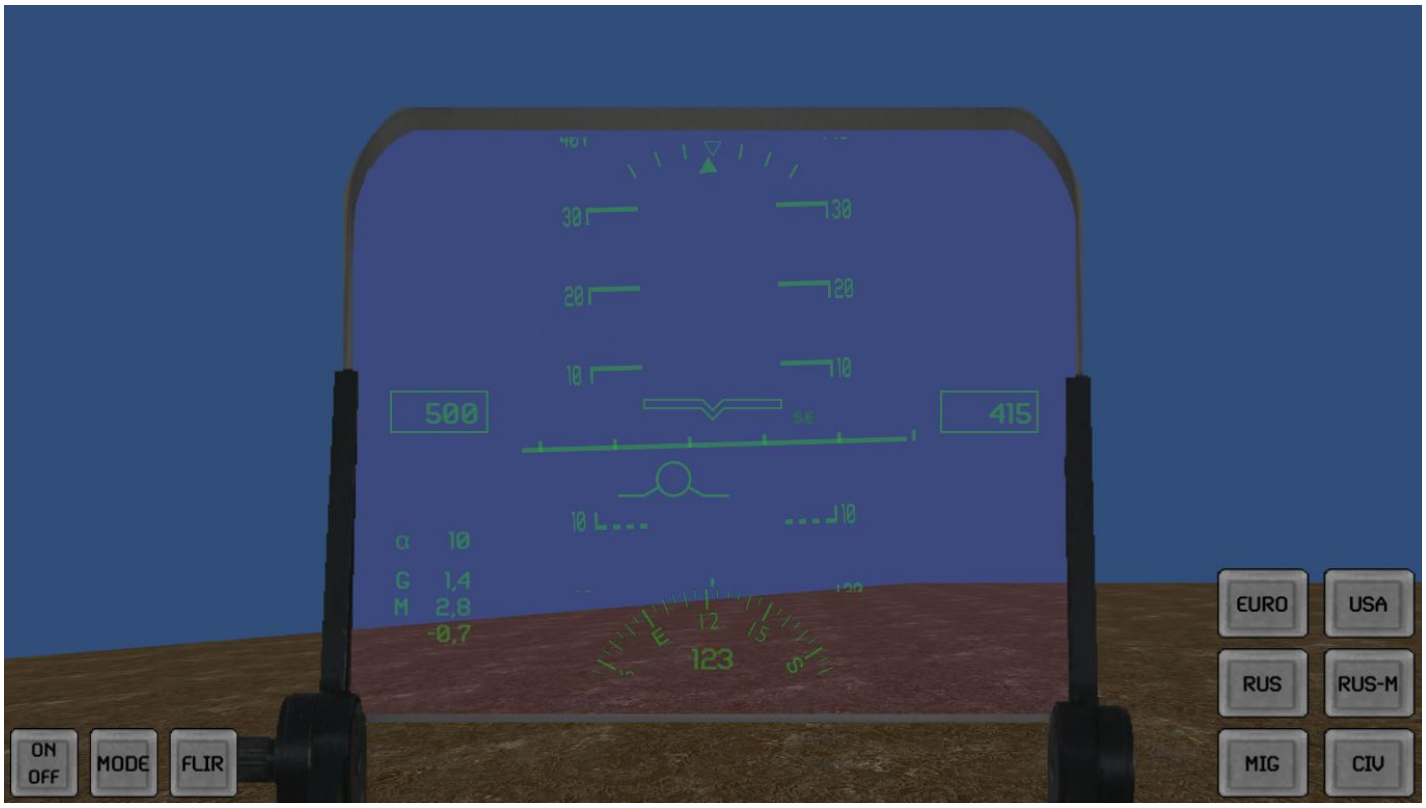


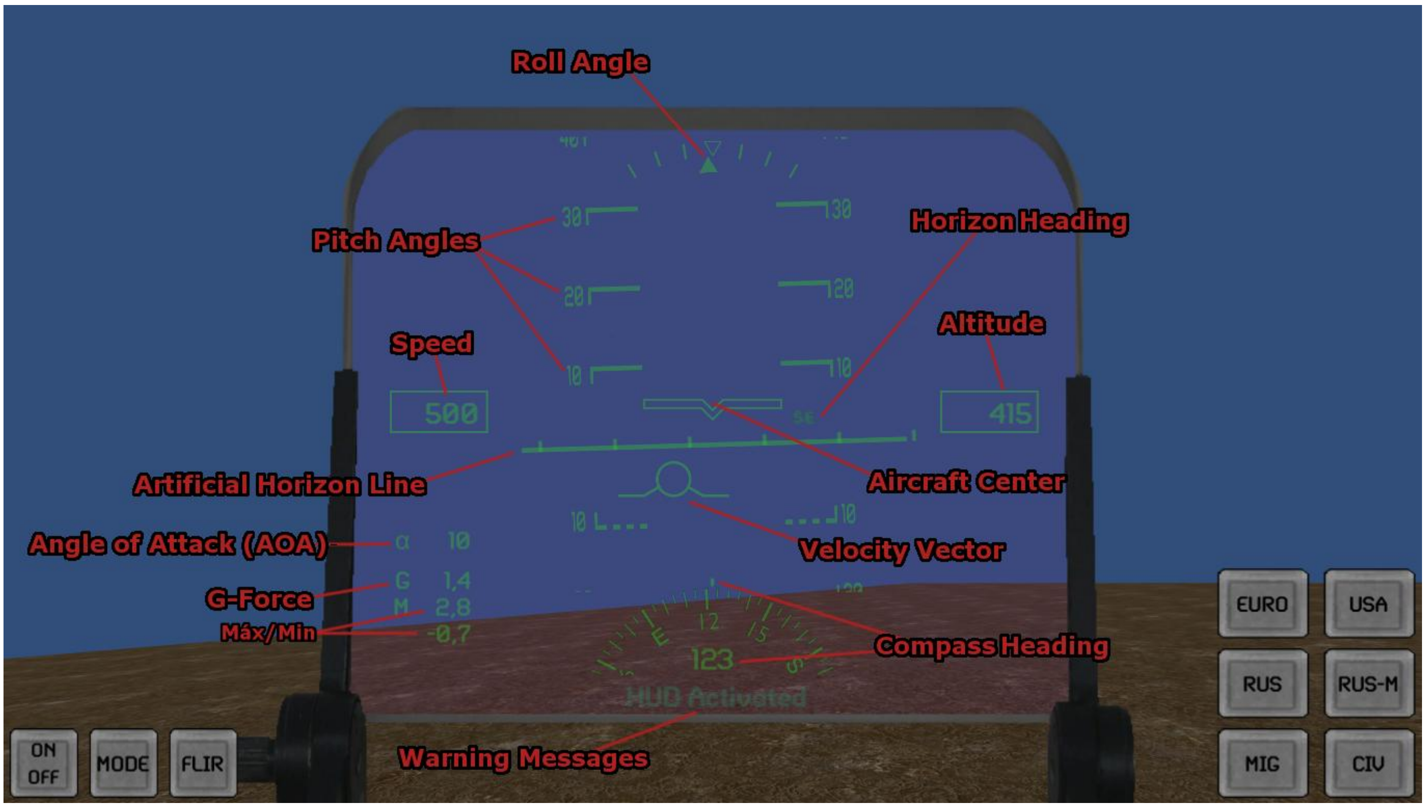
- **Russian MIG HUD Symbology:**





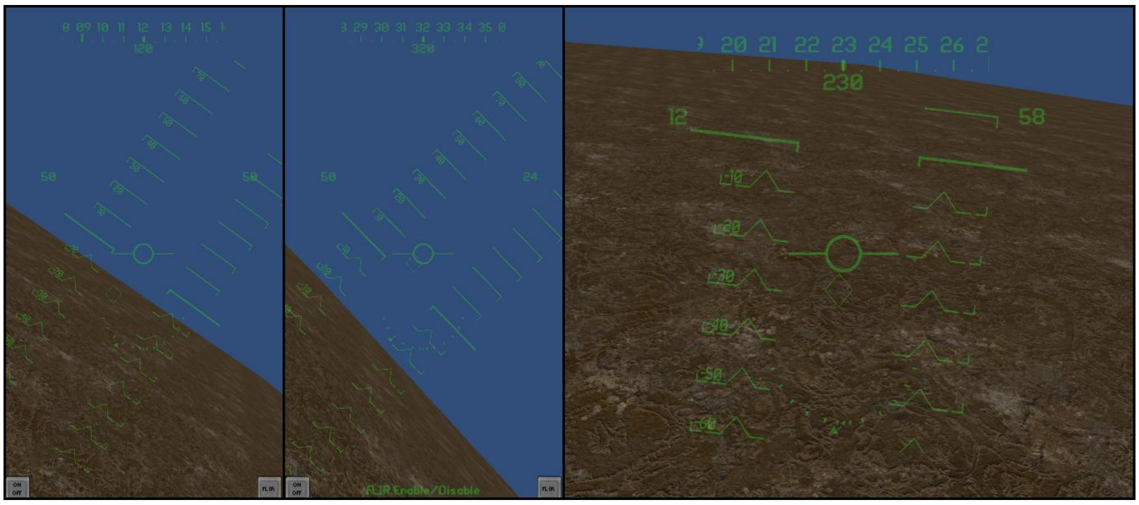
➢ **Civilian HUD Symbology:**



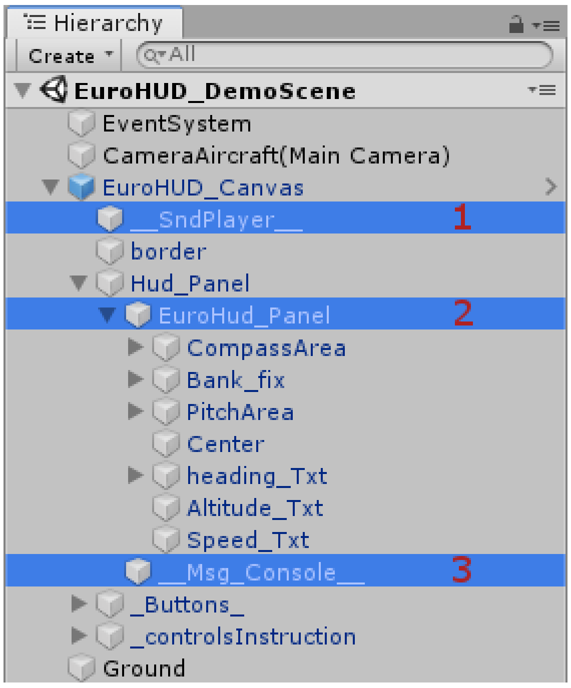


➢ HMD Mode (Helmet-Mounted Display):

In this mode, the HUD glass is hidden and the GUI elements adjust to fit the center of the screen. It can be used as an Arcade version, with third person views or even to simulate a helmet-mounted display (HMD) like those used in the new F-35 figther.



➢ Main Components:



```
Hierarchy
Create ▾  Q▾All
▼ EuroHUD_DemoScene
    EventSystem
    CameraAircraft(Main Camera)
  ▼ EuroHUD_Canvas
      __SndPlayer__                1
      border
    ▼ Hud_Panel
      ▼ EuroHud_Panel              2
        ▶ CompassArea
        ▶ Bank_fix
        ▶ PitchArea
          Center
        ▶ heading_Txt
          Altitude_Txt
          Speed_Txt
        __Msg_Console__            3
    ▶ _Buttons_
    ▶ _controlsInstruction
    Ground
```

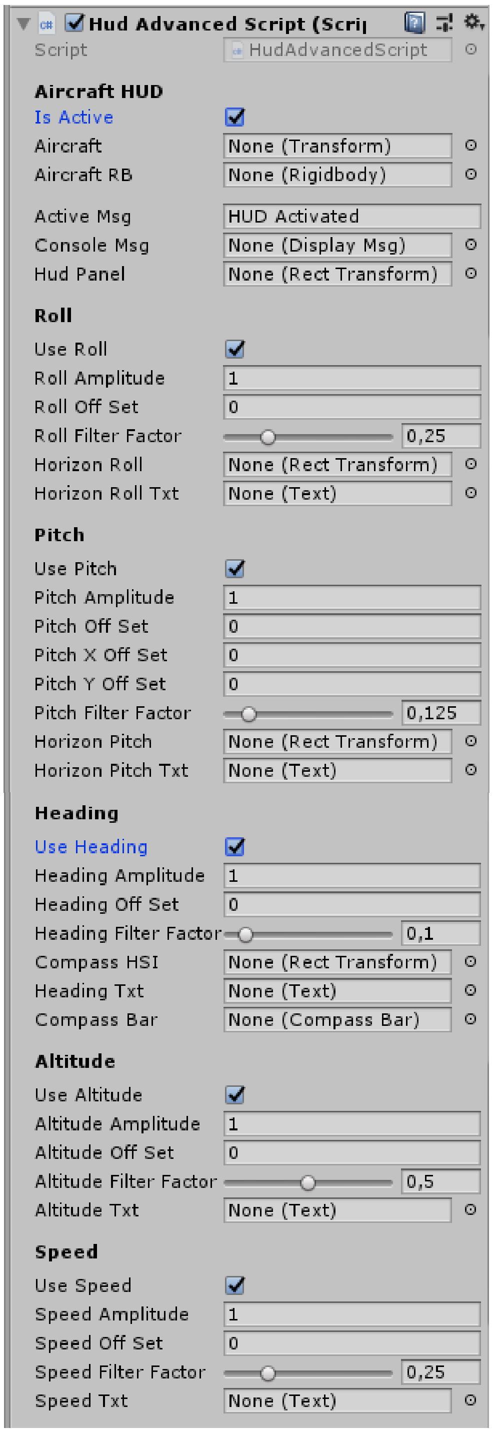
1- Contains script "**SndPlayer**" that manages the HUD sounds.

2- The main script "**HudAdvancedScript**" 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!*

➢ **HudAdvancedScript - Structure in Editor:**



Hud Advanced Script (Scri

Script	HudAdvancedScript
Aircraft HUD	
Is Active	✓
Aircraft	None (Transform)
Aircraft RB	None (Rigidbody)
Active Msg	HUD Activated
Console Msg	None (Display Msg)
Hud Panel	None (Rect Transform)
Roll	
Use Roll	✓
Roll Amplitude	1
Roll Off Set	0
Roll Filter Factor	0,25
Horizon Roll	None (Rect Transform)
Horizon Roll Txt	None (Text)
Pitch	
Use Pitch	✓
Pitch Amplitude	1
Pitch Off Set	0
Pitch X Off Set	0
Pitch Y Off Set	0
Pitch Filter Factor	0,125
Horizon Pitch	None (Rect Transform)
Horizon Pitch Txt	None (Text)
Heading	
Use Heading	✓
Heading Amplitude	1
Heading Off Set	0
Heading Filter Factor	0,1
Compass HSI	None (Rect Transform)
Heading Txt	None (Text)
Compass Bar	None (Compass Bar)
Altitude	
Use Altitude	✓
Altitude Amplitude	1
Altitude Off Set	0
Altitude Filter Factor	0,5
Altitude Txt	None (Text)
Speed	
Use Speed	✓
Speed Amplitude	1
Speed Off Set	0
Speed Filter Factor	0,25
Speed Txt	None (Text)

- "**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 all of them:

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

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

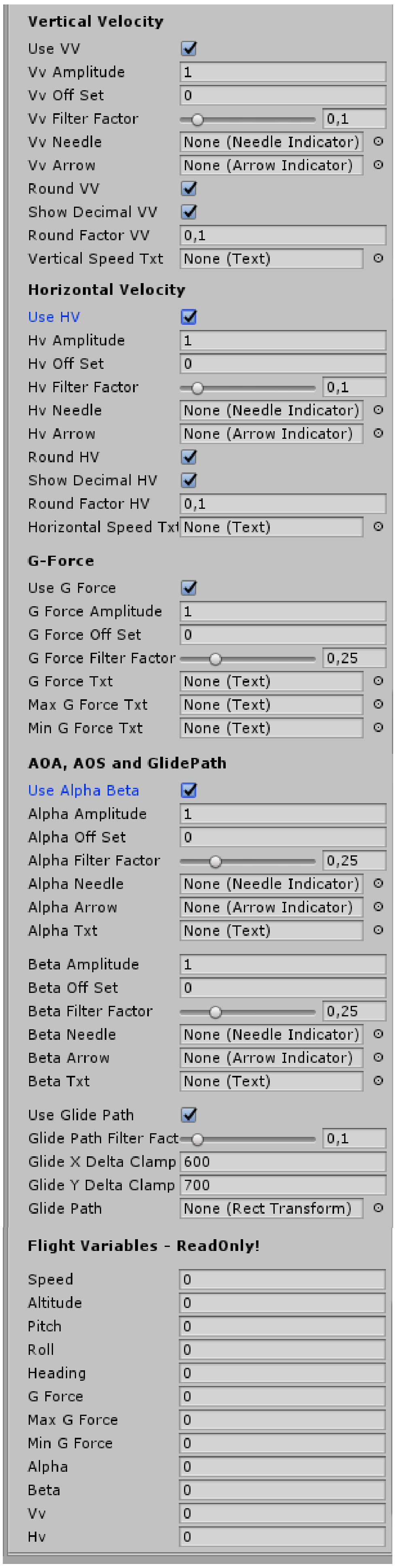
- Note that for **Pitch** variable we have two extra **OffSet** options in **X** and **Y**. These are visual offsets in the position of the Pitch Angles or Horizon Line graphics used on the GUI and not for the actual Pitch value itself.

- 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 too close to 0 it will take a long time to reach the actual value.

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

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

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



Vertical Velocity

Use VV	✓
Vv Amplitude	1
Vv Off Set	0
Vv Filter Factor	0,1
Vv Needle	None (Needle Indicator)
Vv Arrow	None (Arrow Indicator)
Round VV	✓
Show Decimal VV	✓
Round Factor VV	0,1
Vertical Speed Txt	None (Text)

Horizontal Velocity

Use HV	✓
Hv Amplitude	1
Hv Off Set	0
Hv Filter Factor	0,1
Hv Needle	None (Needle Indicator)
Hv Arrow	None (Arrow Indicator)
Round HV	✓
Show Decimal HV	✓
Round Factor HV	0,1
Horizontal Speed Txt	None (Text)

G-Force

Use G Force	✓
G Force Amplitude	1
G Force Off Set	0
G Force Filter Factor	0,25
G Force Txt	None (Text)
Max G Force Txt	None (Text)
Min G Force Txt	None (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	None (Text)
Beta Amplitude	1
Beta Off Set	0
Beta Filter Factor	0,25
Beta Needle	None (Needle Indicator)
Beta Arrow	None (Arrow Indicator)
Beta Txt	None (Text)
Use Glide Path	✓
Glide Path Filter Fact	0,1
Glide X Delta Clamp	600
Glide Y Delta Clamp	700
Glide Path	None (Rect Transform)

Flight Variables - ReadOnly!

Speed	0
Altitude	0
Pitch	0
Roll	0
Heading	0
G Force	0
Max G Force	0
Min G Force	0
Alpha	0
Beta	0
Vv	0
Hv	0

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

- The "**XXArrow**" is the same as the needle, but it uses a reference to the script that controls the visual indication using the arrow translation instead of an angle.

- The "**RoundXX**" option makes that value be rounded by steps using the "**RoundFactorXX**". If round factor is set to "**10**" the value will be shown in steps like **"10, 20, 30, ..."**. If set to "**0.1**" it will be shown like **"0.1, 0.2, 0.3, ..."**.

- The bool "**ShowDecimalXX**" determines if the GUI will show the decimal point or not for this variable.

- The **AOA** (***Angle of Attack***), **AOS** (***Angle of Slip***) and the **GlidePath** (*also called **Velocity Vector** or **flight path vector FPV***) are specific terms on aviation and engineering that are a bit complex to explain here, so here is some Wiki links with more detailed explanation about them:

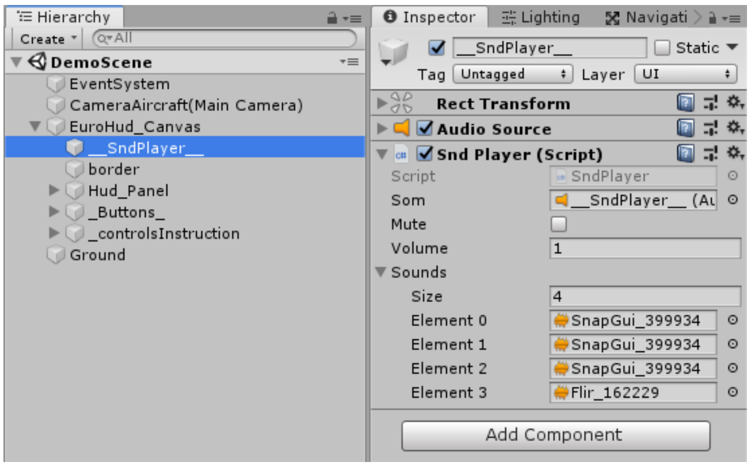
- https://en.wikipedia.org/wiki/Head-up_display#Displayed_data
- https://en.wikipedia.org/wiki/Angle_of_attack
- https://en.wikipedia.org/wiki/Slip_(aerodynamics)

- The "**Glide XY Delta Clamp**" determines how much the **Glide Path** indicator will be clamped from the center position. If you do not wish to clamp it inside the HUD, then you can just set these to some big value.

- On the **Flight Variables** section you can see all current values for all the variables in real time inside the Editor. This is just for debug or tweeking and are **ReadOnly**!

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

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

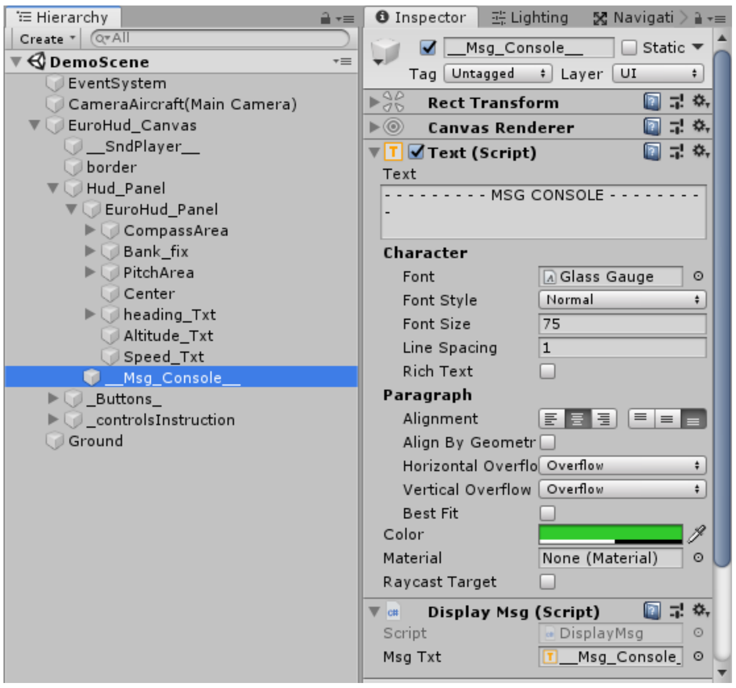


DemoScene
EventSystem
CameraAircraft(Main Camera)
EuroHud_Canvas
__SndPlayer__
border
Hud_Panel
Buttons
_controlsInstruction
Ground

__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

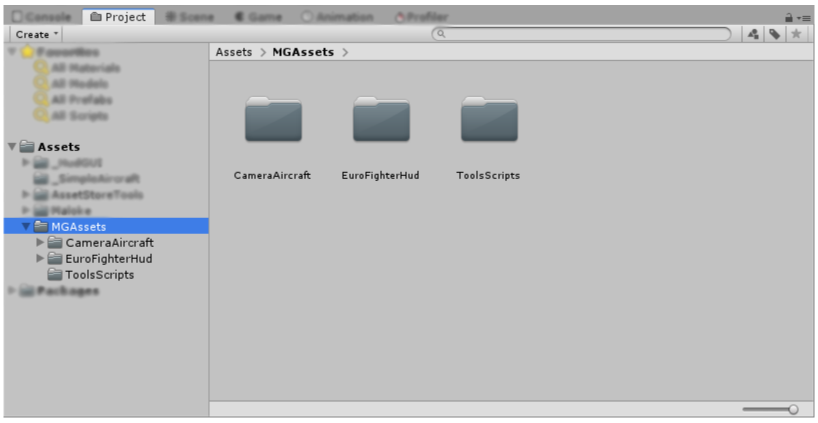
__Msg_Console__ Static
Tag Untagged Layer UI
Rect Transform
Canvas Renderer
Text (Script)
Text
- - - - - - - - - MSG CONSOLE - - - - - - - -
-
Character
Font Glass Gauge
Font Style Normal
Font Size 75
Line Spacing 1
Rich Text
Paragraph
Alignment
Align By Geometr
Horizontal Overflo Overflow
Vertical Overflow 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:

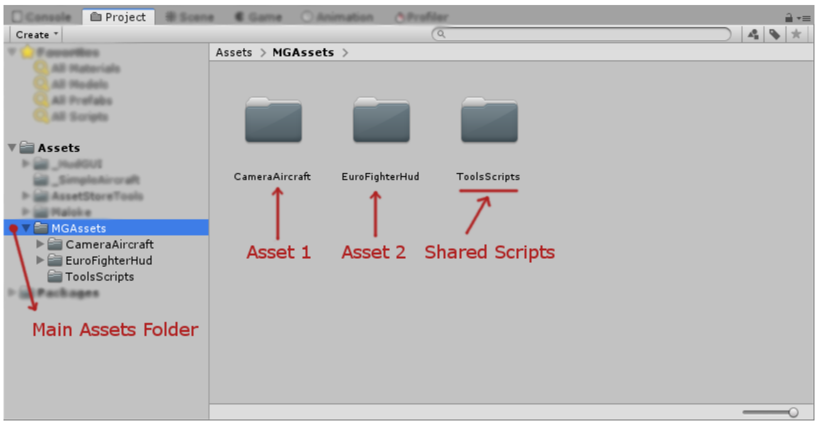
All assets and packages from MalokeGamesAssets will be downloaded/unpacked to a folder called **"MGAssets"** inside the Unity's **"Assets"** root:



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"**.

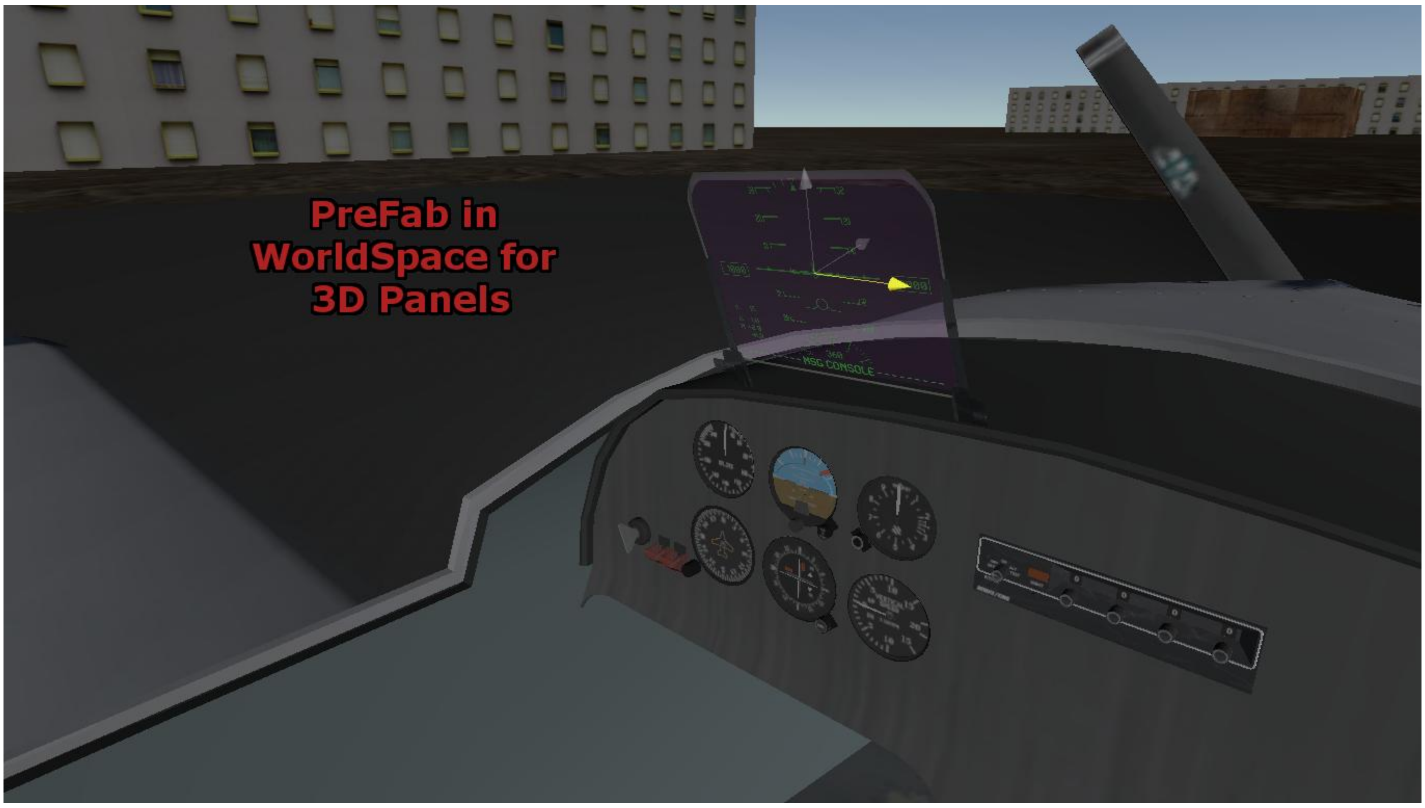
Feel free to explore and use them on your projects too, they are simple but handy!



- **Update 1.1: Prefab in WorldSpace and Bug Fixes**

Fixed issue where extreme G force variation returned NaN and thus locking the G reading.

Added PreFabs configured in WorldSpace for using in 3D Cockpits:


PreFab in
WorldSpace for
3D Panels
MSG CONSOLE