**- Graphy -**

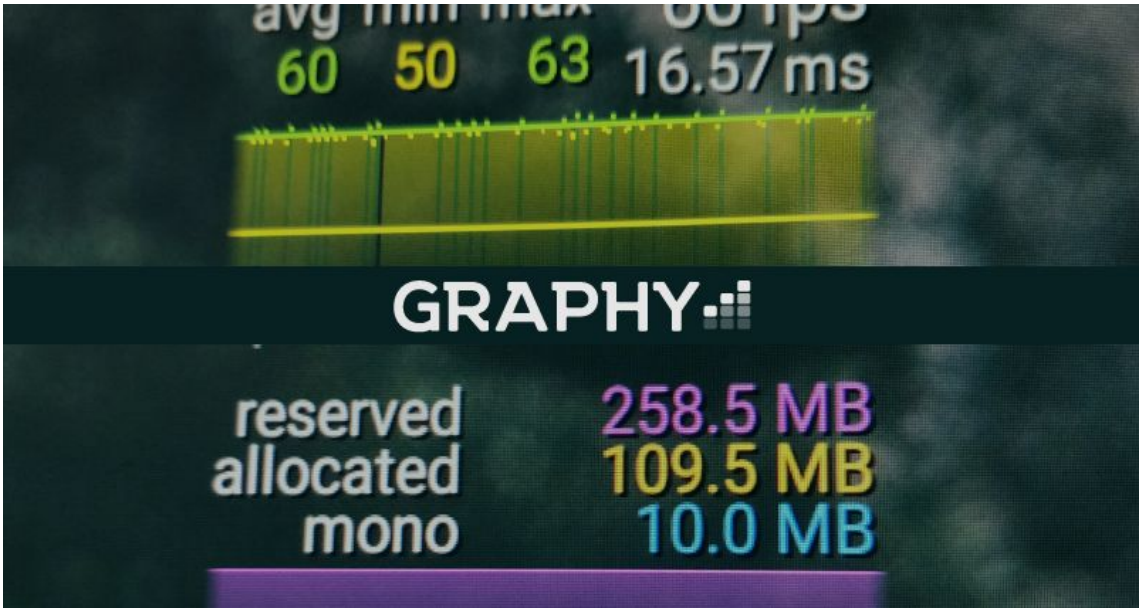
**Ultimate Stats Monitor & Debugger**

**Asset version:** v2.0.1

**Author:** Martín Pane (Tayx)

**Contact information:**

- ➔ Email (martintayx@gmail.com)
- ➔ Twitter (@tayx94)
- ➔ Discord (https://discord.gg/2KgNEHK)




Hi, and thank you for downloading **Graphy**!

I hope you find this asset as useful as I have to debug and monitor your Unity projects. In here you will find a detailed overview of all the functions and options Graphy offers you. If something is still not clear after reading this, or if you want to request a feature or report a bug, scroll down to the bottom and you will find links to contact me through various ways.

- ➔ **Introduction:** Graphy is a set of tools that offers the following:
  - ◆ FPS graph & counter.
  - ◆ Memory (RAM) graph & counter.
  - ◆ Audio spectrometer graph & dB counter.
  - ◆ Advanced device data.
  - ◆ Debugger script.

- ➔ **How to use:**
  - ◆ Just drag the prefab from “Tayx/Graphy - Ultimate Stats Monitor/Prefab” into the scene you want!
  - ◆ It will survive scene changes, it uses the Singleton pattern and DontDestroyOnLoad.
  - ◆ Be careful not to try to access it from code if it’s not instantiated in the scene, it won’t work.

➔ **Common module features:**

- **Module Position:** defines the position where the module will be located.
- **Module state:**
  - **FULL:** complete module with all its features.
  - **TEXT:** hides the graphs and just shows the text.
  - **BASIC:** shows just the basic module information.
  - **BACKGROUND:** monitors the data in the background, making it accessible from code.
  - **OFF:** turns the module completely off, making it inaccessible from code. Careful, this also breaks functionality with the **debugger** (that uses the monitors to read data).
- **Graph color:** sets the color of the graph.
- **Graph resolution:** sets the amount of points the graph is divided into. Be careful with very high graph resolutions with low resolution screens, the graph may have issues drawing all the points.
- **Text update rate:** defines how many times per second the text updates.

➔ **FPS module:**

- **Thresholds:** values used to switch between the different colors associated with the text and graph.
- **Time to reset min/max values:** Time (in seconds) to reset the minimum and maximum frame rates if they don't change in the specified time. Set to 0 if you don't want it to reset.


➔ **RAM module:** has no specific controls.

➔ **Audio module:**

- **Find audio listener:** this will only apply if AudioListener is null.
  - **ALWAYS:** constantly scans for the AudioListener component in the main camera.
  - **ON SCENE LOAD:** scans for the AudioListener in the main camera every time a scene loads.
  - **NEVER:** never scans for the AudioListener.
- **Audio Listener:** reference to the AudioListener you want to use.
- **FFT Window:** Used to reduce leakage between frequency bins/bands. Note, the more complex the window type, the better the quality, but reduced speed. Simplest is rectangular. Most complex is BlackmanHarris.
- **Spectrum size:** Has to be a power of 2 between 128-8192. The higher the sample rate, the less precision, but also more impact on performance. Careful with mobile devices.

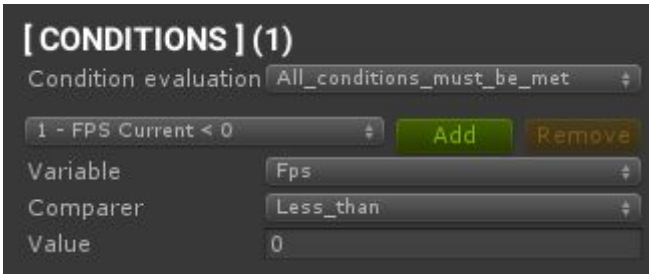

➔ **Advanced data module:** shows information about the device.

- **Screen data:** resolution, refresh rate.
- **Window data:** resolution, refresh rate, dpi.
- **Graphics API:** OpenGL/DirectX/etc.
- **GPU:** VRAM, Max texture size, Shader level.
- **RAM:** system available memory.
- **OS:** operating system.

➔ **The Debugger:** this component is a powerful feature that allows you to set a number of conditions, that, if met, will start a chain of actions defined by you. It's designed around Debug Packets. Each Debug Packet can have different conditions and actions.

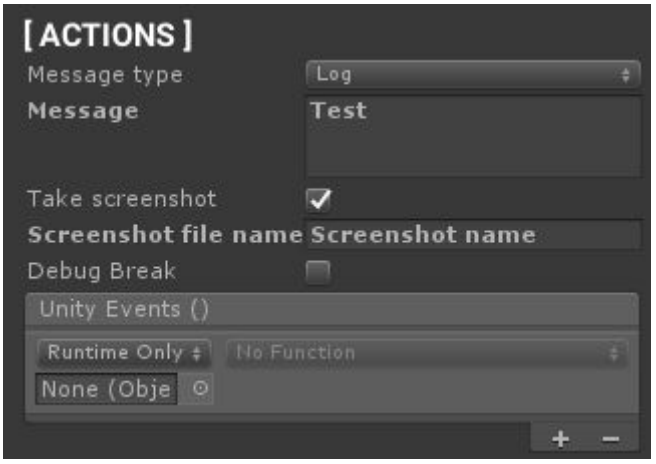
- **Common parameters:**
  - **Active:** If not active, it will be skipped when the debugger checks Debug Packets.
  - **ID:** Optional, but it's used to operate with Debug Packets from code.
  - **Execute once:** If true it will only be executed once, then it will delete itself.
  - **Init sleep time:** Time to wait before checking this Debug Packet.
  - **Sleep time after execute:** Only applies if "Execute Once" is false. Time to wait before checking this Debug Packet after it's been executed.


- **Conditions:**
  - **Condition evaluation:** defines if all conditions must be met, or just one of them.
  - **Variable:** which variable to compare against the value specified.
  - **Comparer:** <, <=, ==, >=, >.
  - **Value:** float value used to compare against the variable specified.




- **Actions:**
  - **Message type:** standard types of console log: Log, Warning, Error.
  - **Message:** text message you want to send to the console.
  - **Take screenshot:** takes a screenshot.
  - **Screenshot file name:** screenshot name. Avoid this characters: "* . \" / \\ [ ] : ; | = ," (without the quotes). It will have the date appended at the end to avoid overwriting. The location where the image is written to can include a directory/folder list. With no directory/folder list the image will be written into the Project folder. On mobile platforms the filename is appended to the persistent data path.
  - **Debug break:** pauses the Unity Editor.
  - **Unity events:** a Unity event wrapper that can be serialized and modified from the Unity Editor.

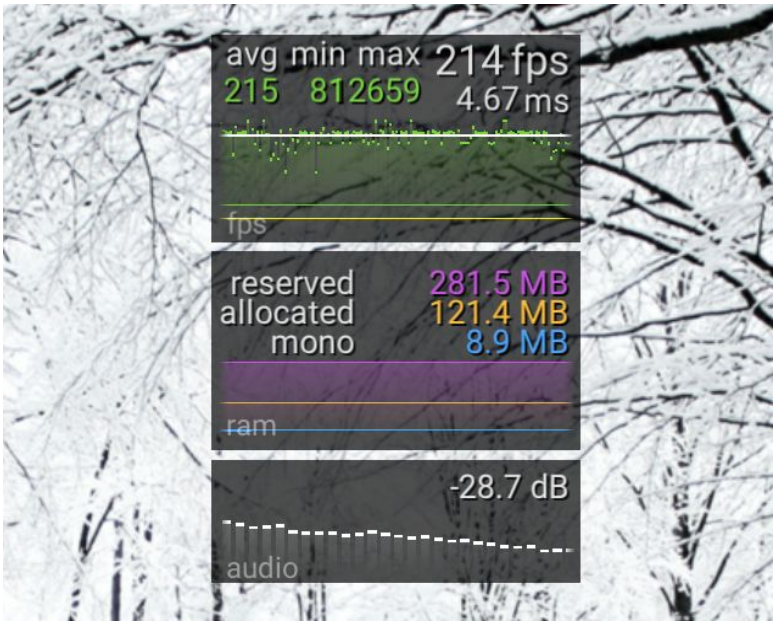
- **System.Action (only accessible from code):** call your own methods from code adding System.Action to the Debug Packet you want, retrieving it with its ID.




➔ **Graphy Mode:**
- ◆ **FULL:** unlocks all the features to their maximum potential.
- ◆ **LIGHT:** limits some small things to improve compatibility with older hardware (for example, the maximum graph resolutions is reduced).

➔ **Keep Alive:** if true, Graphy will survive scene changes. Careful, if Graphy is set as a child of another object, both will survive scene changes.

➔ **Background**
- ◆ If ticked, a background overlay will be applied to all modules. The color will be the selected one.



➔ **Hotkeys**

- **Toggle modes:** switches between various layouts and modules.
- **Toggle active:** toggles Graphy ON/OFF


➔ **Scripting**

- **Add the namespace:** Add “using Tayx.Graphy;” to the top of your .cs file.
- **GraphyManager:** through this class you can access all the stats related variables, like CurrentFPS, AllocatedMemory, etc.
- **GraphyDebugger:** through this class you can access the various Debug Packets to deactivate them, add System.Action callbacks, etc.

```
// Use this for initialization
0 references
void Start ()
{
    var fps = GraphyManager.Instance.CurrentFPS;

    GraphyDebugger.Instance.RemoveAllDebugPacketsWithId(2);
}
```

- **Advanced stuff**:
  - Add callbacks to an existing Debug Packet:

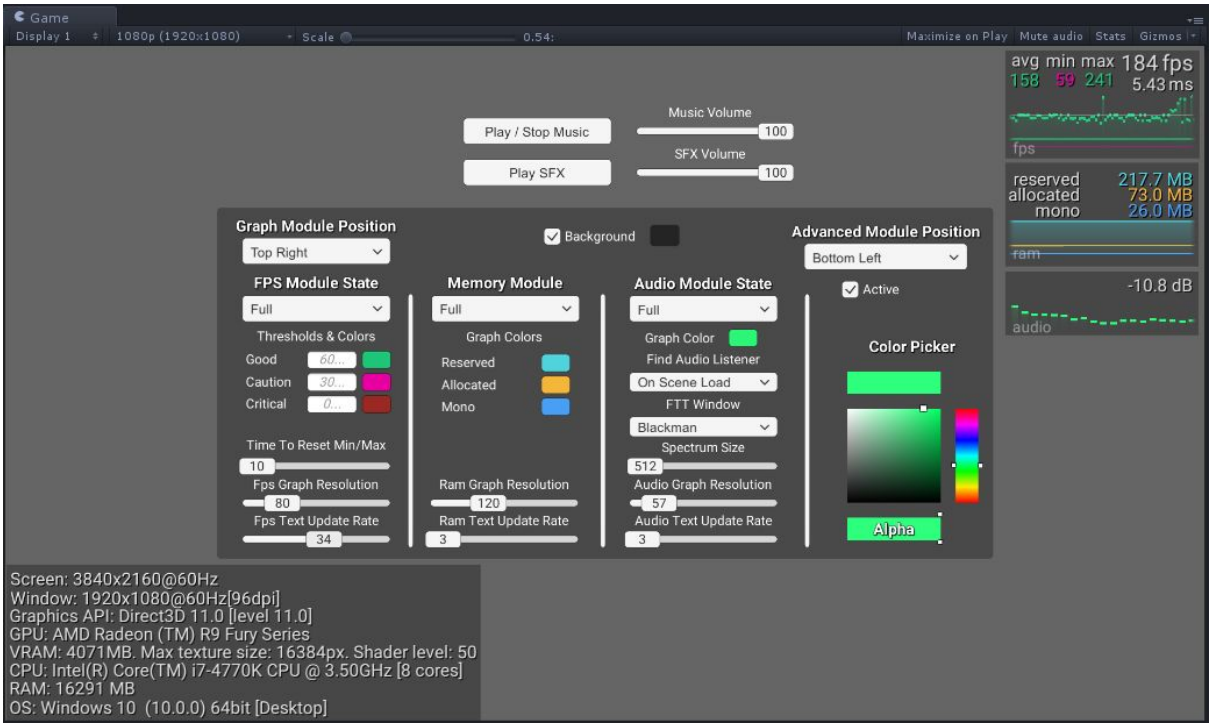
```
Action TestAction = null;

TestAction += TestMethod;

GraphyDebugger.Instance.AddCallbackToFirstDebugPacketWithId(TestAction, 5);
```

  - Add a new Debug Packet:

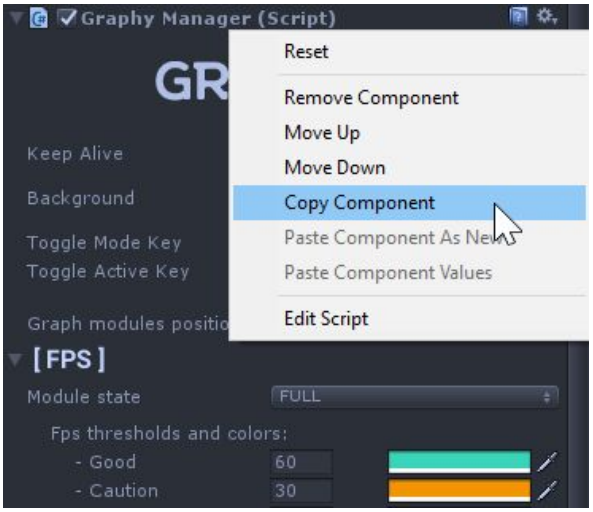
```
Action TestAction = null;

TestAction += TestMethod;

GraphyDebugger.Instance.AddNewDebugPacket
(
    5,
    new GraphyDebugger.DebugCondition()
    {
        Comparer = GraphyDebugger.DebugComparer.Equals_or_greater_than,
        Value = 45f,
        Variable = GraphyDebugger.DebugVariable.Fps_Avg
    },
    GraphyDebugger.MessageType.Warning,
    "Message Warning Test",
    true,
    TestAction
);
```

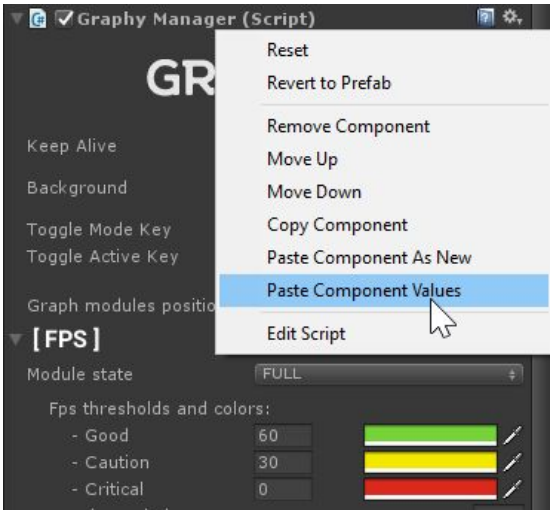
➔ **Customization Scene:**

- It can be found at: *"Assets\Tayx\Graphy - Ultimate Stats Monitor\Scene\Customize Graphy"*
- Allows setting all the parameters in runtime




- Once you have the desired values, just copy the component, exit play mode, and paste the component values, like this:



I've you've read until here, thank you! I hope that you enjoy Graphy and that it makes your developer life easier. It would be highly appreciated if you leave a review in the Asset Store, and if you want to **contact me,** don't hesitate to get in touch through:

- ➔ Email (martintayx@gmail.com)
- ➔ Twitter (@tayx94)
- ➔ Discord (https://discord.gg/2KgNEHK)

**FAQ**:

- ➔ Why does the FPS value in the Stats windows in the Game tab different from Graphy's FPS value?
  - ◆ *That value is an estimation of the FPS you would get in a build, without the Editor overhead. Graphy reports the current real FPS value.*
- ➔ Do you have an approximate ms figure on typical midrange desktop hardware, with all features enabled?
  - ◆ *Yes, less than 0.1 ms in a build and around 0.3-0.7 ms in the Editor due to the extra Editor overhead.*
- ➔ How is Graphy rendered?
  - ◆ *It's being rendered through the Unity UI. Graphy is contained in a standard Canvas, and you have complete control over it.*