WHY?

Unity's OpenURL works just fine. However, it is limited because it works as if you were calling window.open with *_self* as the *windowName* parameter. This asset uses *window.open* under the hood as well but it exposes all of its parameters. This means that you can open links on new tabs/windows using *_blank* as the *windowName* parameter, for example, and much more.

HOW DO I USE IT?

First thing to do is attach the *LaunchURLWebGL* script to any gameobject of your scene. You only have 4 methods. The first 3 only work on the WebGL platform. The fourth is cross platform:

1. *public static extern void launchURL(string url = "", string windowName = "_blank", string windowFeatures = "")*

See window.open for more information about each parameter.

2. *public static void launchURLSelf(string url = "")*

It's just a wrapper for method 1, but with *windowName* set to *"_self"*. This opens a URL on the current window/tab. It's no different than using OpenURL.

3. *public static void launchURLBlank(string url = "")*

It's just a wrapper for method 1, but with *windowName* set to *"_blank"*. This opens a URL on a new window/tab.

4. *public static void OpenURL(string url)*

It works like Unity's OpenURL if the target platform is not WebGL. If the target platform is WebGL, it works like method 3.

Since you can't call methods directly in the inspector if they have more than 1 parameter, and you are rarely(if ever) going to use the *windowFeatures* parameter, I created methods 2, 3 and 4 for your convenience.

There is one important detail to notice here. You **MUST** call all methods on a **pointerdown** event. Not doing so may result in the click failing to launch the URL if you're opening on the same tab, and unnecessarily prompting the user to allow pop-up if

opening the URL on a new window/tab. The example scene shows how to easily use a pointerdown event.

ANY REQUIREMENTS?

Your browser needs to support window.open, that's it.