## DESCRIPTION

This asset enables you to vibrate your mobile on the WebGL platform, because at the time of writing, Unity's official API doesn't work. You can control how long you want to keep the mobile vibrating, as well as define a vibration sequence if you'd like.

## BROWSER SUPPORT

Currently only browsers running on Android are supported. It's not working on the current Firefox version(Fenix), and it doesn't work on any Apple OS (iOS, iPadOS, macOS). I tested on Chrome, Opera, Edge, and Samsung Internet(all on Android), working fine.

## HOW DOES IT WORK?

This plugin leverages the existing web Vibration API. The method overloads reflect how it works on the web.

There are only 2 ways to vibrate your mobile:

1. Use a number to tell how long you want to vibrate the mobile for, in milliseconds. Example: *Vibrate(500)*, will vibrate your phone for 500ms. If you just call *Vibrate()*(no arguments) it will vibrate for 100ms.
2. Use a vibration sequence. Example: *Vibrate(new uint[] {200, 500, 200})*, will vibrate your phone for 200ms, stop for 500ms, and vibrate again for 200ms.

Calling *Vibrate(0)* or *Vibrate(new uint[] {})* have the exact same effect, which is to interrupt any current vibration or vibration sequence.

*Vibrate* returns true if the device is successfully vibrated(or stops vibrating), and false if not. *Vibrate* always returns false on browsers that don't support vibration. Please note that the return value is somewhat untrustworthy. Even though it's not currently working on Firefox, it returns true. On chrome for desktop, it also returns true(and my desktop doesn't vibrate...lol). However, it does return false on iOS.

There is one additional method, *isSupported*, which returns *true* if the browser supports vibration, and *false* if not.

## IMPORTANT

The vibration functionality is unlocked with a button click. After you unlock the functionality, you can use it anytime you want. It will always work on a button click, but if you want to use the vibration in some other situation, you must make sure that the user

clicks on any button beforehand, even if the button doesn't do anything. As long as the user needs to click on any button in your game before using vibration, it will work. The button doesn't even need to be from the game, but from the template. If you use Unity 2019 for example, the default template has an *ok* button that you need to press before starting the scene. This is enough to unlock vibration.