README.md

Unity Core Haptics Plugin for iOS 13+

This folder contains a plugin to play iOS Core Haptics vibrations from Unity.

Requirements

- Unity 2019.3+
- iOS 13+
- Device with Core Haptics support such as iPhone 8, X/XS/XR, and 11 (and all their variations)

Setup for Custom Haptics

AHAP files
If you plan on using custom AHAP files, you **must** include it within `"Assets/StreamingAssets"` and refer to it relative to `"StreamingAssests"`. For example, if you have a file stored at `"Assets/StreamingAssets/path/to/Drums.ahap"`, you can play it like so:

```
UnityCoreHapticsProxy.PlayHapticsFromFile("path/to/Drums.ahap");
```

Audio Files
If you have an audio file in `"Assets/StreamingAssets/path/to/Drums.wav"`, the AHAP file's `"EventWaveformPath"` key should have the value `"path/to/Drums.wav"`. Again, the path is relative to `"StreamingAssets"`.

References
For more details on the format of AHAP files, please refer to Apple's documentation on Representing Haptic Patterns in AHAP Files.

[Optional] For an understanding of how Streaming Assets work, please see: https://docs.unity3d.com/Manual/StreamingAssets.html.

How to use the plugin

There an included example scene with scripts that you can try yourself in the Example folder. Note that you will need to replace the relativeAHAP path for one of the cubes to a valid one using your own AHAP file.

The following example shows how to call transient, continuous, and preset (AHAP) haptics:

```
// Must be included in any file that wants to call Core Haptics functions
using UnityCoreHaptics;

float intensity = 1f; // 0 to 1
float sharpness = 1f; // 0 to 1
float duration = 2f; // in seconds

// Check if iOS device supports core haptics
if (UnityCoreHapticsProxy.SupportsCoreHaptics()) {
```

```
    // Assumes we have a file at path Assets/StreamingAssets/Drums.ahap
    string pathToDrums = "Drums.ahap";

    // Play transient (one-time) haptics
    UnityCoreHapticsProxy.PlayTransientHaptics(intensity, sharpness);

    // Play continuous haptics
    UnityCoreHapticsProxy.PlayContinuousHaptics(intensity, sharpness, duration);

    // Play haptics from custom AHAP file
    UnityCoreHapticsProxy.PlayHapticsFromFile(pathToDrums);
}
```

Note that the first haptic call could cause a frame spike issue because the plugin automatically creates a haptic engine at this time. To avoid this lag, it is recommended that you create the engine at the start of your scene before playing haptics like so:

```
// This should only be called one time in your app
UnityCoreHapticsProxy.CreateEngine();
```

You can also listen to events when an engine is created or throws errors like so:

```
// Listen to engine creation events
UnityCoreHapticsProxy.OnEngineCreated += () => {
    Debug.Log("Engine created!");
    // You are now set to play haptics!
};

// Listen to engine error events
UnityCoreHapticsProxy.OnEngineError += () => {
    Debug.LogError("Engine error!");
    // Handle errors here
};
```