**Clock.cs**

This is a class that sends events based on a BPM. It allows you to easily & accurately sync aspects of your game to the tempo (bpm) of your soundtrack. You'll want to use BeatClock to start playing your song, so that it's in sync. If you change songs or BPMs, you just have to call InitializeBPM(newBPM). I'll keep you posted if I make additional improvements to this.

**Editor/AutoAudio.cs**

This is an asset importer UnityEditor extension that identifies common errors in audio files like:

* Clips with clipping - *Digital distortion from exporting or recording audio at too high a level*
* Clips that start with silence - *Messes with game feel, making actions feel latent*
* Clips that start or end with a non-0 audio crossing - *Likely to make clicking/popping sounds on playback, reduce overall sound quality*

Simply right click on an audio file / audio files then choose "AutoAudio Check". Results of the check will be printed to the console. You can also have it run on all audio files automatically on import. Enable this by going to Preferences->AutoAudio & select "Automatically Run on Import".

**Unity - sfxr**

This is an editor asset & series of scripts that implement sfxr (with the [bfxr](http://www.bfxr.net/) improvements) in Unity. Can be super useful for quick prototyping. Original source can be found [here](https://github.com/zeh/usfxr), but I've made some modifications to allow it to be used to generate pitched musical tones. Most of that logic is in my script Freq2Note.cs. Just use the function Note2Freq(Note \_note), and pass that to an sfxr synth. If you want to see an example implementation, check out the Unity project AudioPongGen2.zip in the Projects folder.