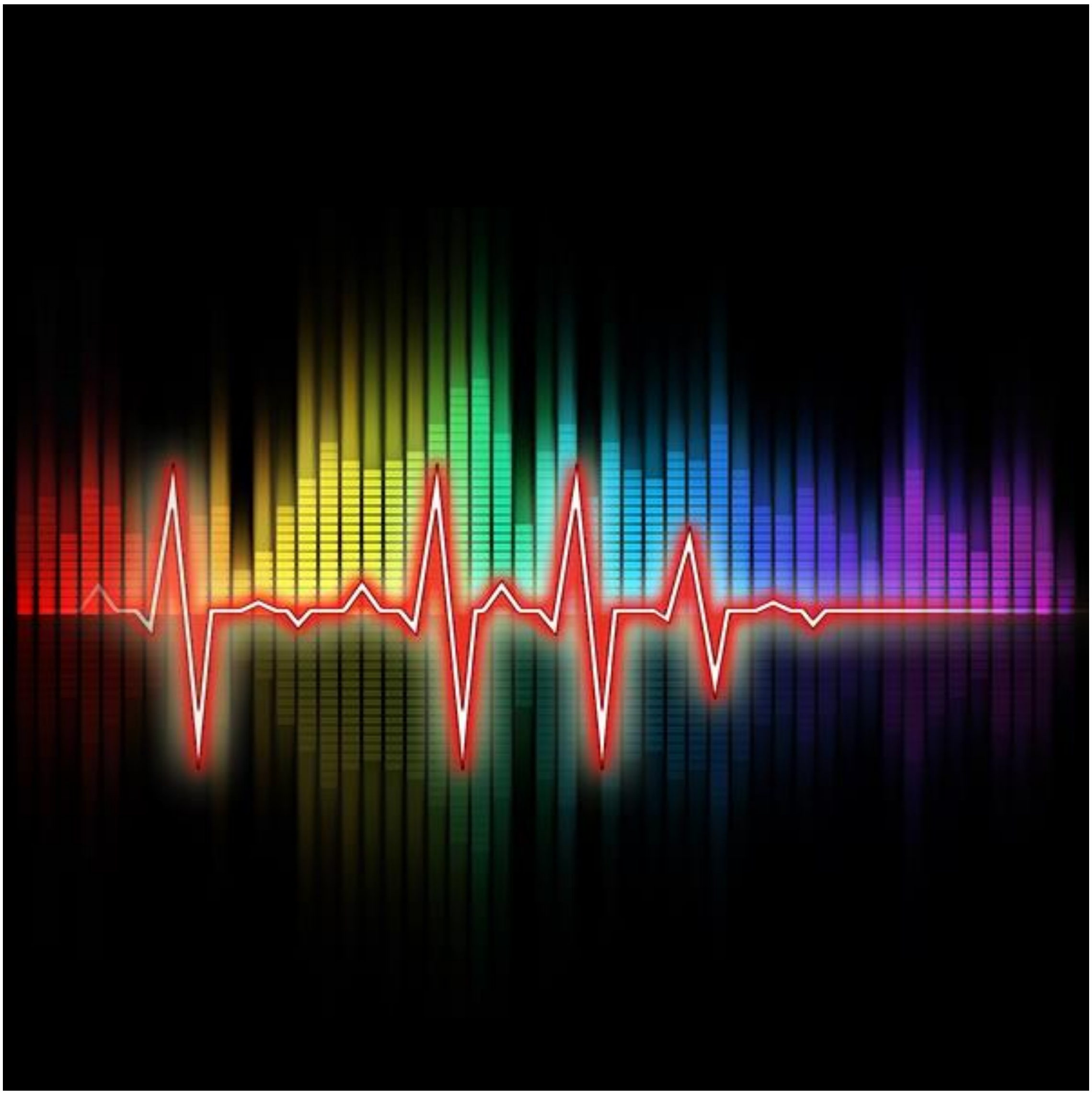
Beat Detection



Versión 3.0, for Unity 3D

3y3.net

Index

Introduction

Beat Detection Algorithm can detect the beats of any audio source in realtime by using frequency spectrum analysis (FFT), audio energy or both simultaneously.

You can use the asset in your games or apps and make them react to the music. The code is high-performance and you can use it on mobile devices and even use several scripts simultaneously to control many audio sources.

Some of the Beat Detection features are:

- Detect beats in real time from any audio source, including microphone
- Advanced high performance FFT algorithm
- Works on mobile
- Detect high, medium and low frequency beats
- May also be used in single mode and only detects energy beats
- Works both on Unity and Unity Pro
- Supports Unity 4.3 and above, including Unity 5
- Full documented API
- Lot of working examples

Quick Start tutorial

Beat Detection asset is really simple to use.

1. Add the BeatDetection script to the object with the audio source you wish to control.

2. Set the 'Beat Mode' to the one you wish.

3. Once the scene were running all the beats will be reported to the Update() function of the script. By default this function will fire the apropriate event.

Beat Modes

Beat Detection can be used in three different modes:

- **Frequency Mode**. The asset will use an advanced FFT algoritm to detect beats in three frequency bands, low, médium and high. The asset will fire an event, either *Kick* for low, *Snare* for médium and *Hit Hat* for high, whenever any of this beats is detected.

- **Energy Mode**. The asset detects simple beats by calculating the amount of energy present in the audio. This mode does nor differenciate among any frequency band and will fire an *Energy* event for every beat detected.

- **Both**. Combines both methods and detects beats by frequency and also by energy. Take in mind that almost all events detected in frequency mode will be also fired in energy mode so it is posible to get this betas twice.

Using events

By default, BeatDetection script will fire an event once any beat is detected. Each event is defined in this event class:

```
public class EventInfo  {
        public EventType messageInfo;
        public BeatDetection sender;
}
```

The event type is a public enum with this format:

```
public enum EventType {
    Energy, Kick, Snare, HitHat
}
```

The events are sent to the callback function, so you will need to register you own callback function to get all this events. You can see examples on how to do in the sample scenes. A simple callback function may be:

```
public void MyCallbackEventHandler(BeatDetection.EventInfo eventInfo) {
         Debug.Log(eventInfo. messageInfo);
}
```

And this is a simple example on how to register the callback function:

```
ObjWithBeat.GetComponent<BeatDetection>().CallBackFunction = MyCallbackEventHandler;
```

But we encourage you to take a look at the examples and/or ask any question to us.

Avoid events by copying the code

If you don't feel comfortable with events and callbacks functions you may simply copy and paste the algorithm and use your own code. There are two examples where you can use the algorithm directly.

Simply take in mind to rename the class name to any other you wish. You can see a good example in the No Event samples.

Examples

The asset comes with four examples divided in two groups. The first group uses event system and the other group has the same samples but without using events.

The first example is a simple beat detection over a recorded audio source. The sample folders are **Event – Example** and No **Event – Example**.

The second example uses microphone as input for beat detection. The sample folders are **Event – Example Mic Input** and No **Event – Example Mic Input**.