Beat Mapping

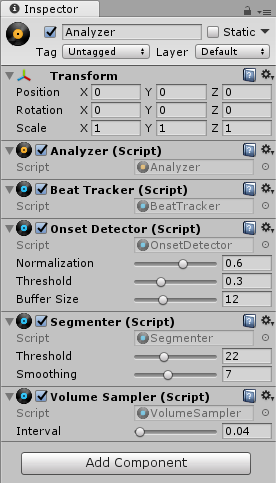
-Some of this can be easily found by looking around the documentation. Thus, some of it is redundant. The reason why this exists is to have a single document that is easy to follow along and understand for our purposes.-

For this, we make use of the [Rhythm Tool](https://assetstore.unity.com/packages/tools/audio/rhythmtool-15679) asset.

From the [documentation](http://hellomeow.net/rhythmtool/documentation/html/53f2927b-71fd-4719-aae5-34b7ff45a9ad.htm), to begin analyzing a song you must create a RhythmAnalyzer.

RhythmAnalyzer “takes an AudioClip, chops it up into small chunks and passes it to different analyses.”

-This part of the documentation is slightly confusing. The picture shown makes it seem like Analyzer is a type of game object that can be added into the hierarchy. It does not help that the name of the object is Analyzer.



What you have to do, instead, is create an empty object to the scene, name it whatever you like, then add the analyzer as a component. This is evident by looking at the example scene “Visualizer”

For our purposes, Beat Tracker is important. “RhythmAnalyzer provides a RhythmData object with several tracks.”

* Beats: Represent the rhythm of the song.

Fields: - bpm

- length: duration of this feature in seconds

- timestamp: time in seconds at which this feature occurs.

We can get the information of what the track is currently playing by listening from the audiosource

AudioSource.timeSeconds

-Gives current playback time.

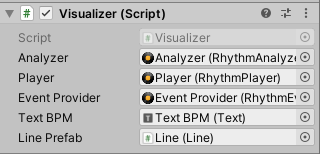
RhythmData.GetFeatures<T>(List<T> features, float start, float end, string trackName)

-Finds all features of type T within tracks that match trackName within a certain

frame.

\*Might be useful, AudioClip.length for length of song.

The analyzer can then be added onto the event script (which we create), which will detect beats.



Here is an event script attached to an empty game object. The purpose is to visualize

beats, onset mapping, etc. We don’t need all of them.

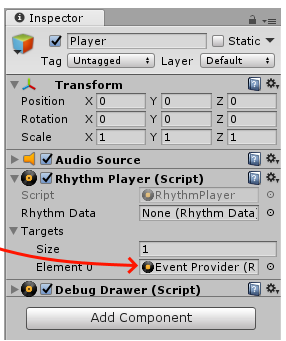
RhythmTool provides an event manager that is useful for detecting beats.

-Create an empty game object, called Player. Attach an audio source to it.

Then, attach a Rhythm Player. This allows it to keep track of the time and RhythmData.

Next, we can add the RhythmEventProvider. Add it to the Rhythm Player’s targets,

change target size to 1.



You can change the offset of the event provider to detect events ahead of time.

Double click the event provider and change. Maximum 10 seconds ahead.

Which events will it react to?

-The ones you subscribe it to

Use: [RhythmEventProvider.Register()](http://hellomeow.net/rhythmtool/documentation/html/M_RhythmTool_RhythmEventProvider_Register__1.htm)

Ex.

public RhythmEventProvider eventProvider;

void Start()

{

eventProvider.Register<Beat>(OnBeat);

}

Private void OnBeat(Beat beat)

{ //Do something}

Note, it is important to unsubscribe from events that will not need to be used, or for components that are destroyed.

void OnDestroy()

{

eventProvider.Unregister<Beat>(OnBeat);

}

**Adding an AudioClip Selector**

Finally, we must choose which song we are playing. This method also allows us to queue

multiple songs.

Simply, add the AudioClipSelector script onto the event manager. Change the size of the

songs playlist, and add the songs you wish to play.

The script requires the same analyzer and player that was added to the event manager.