midi2event Unity instructions v1

Reach out to Cadence if you have questions/feature requests!

The Concept:

The midi2event system is a lightweight tool for charting C# events using a MIDI file. This allows you to trigger any code you would like in time with the chart/music.

The Prefab:

The midi2event prefab contains a parent object containing the midi2event Unity script as well as a child object containing an audio source.
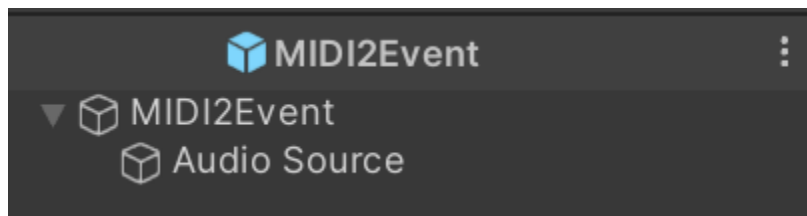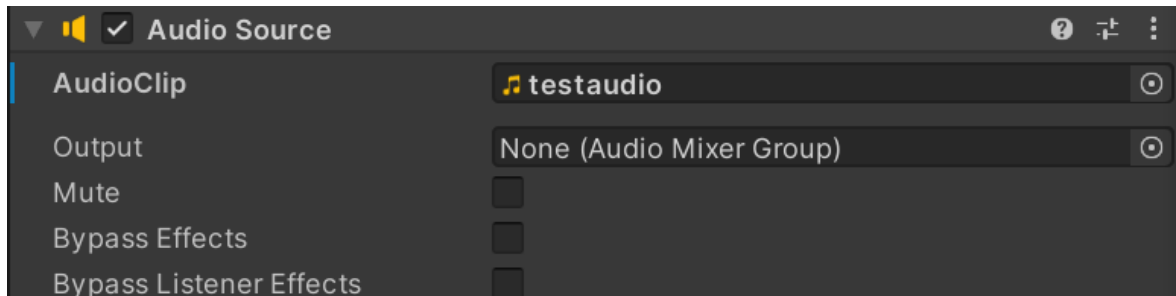

To set up the prefab, do the following:
- Drag the prefab into the scene hierarchy.
- Right click the midi chart you want to use in the Unity project directory, copy the path, and paste it into the Chart Path field of the MIDI2Event game object.
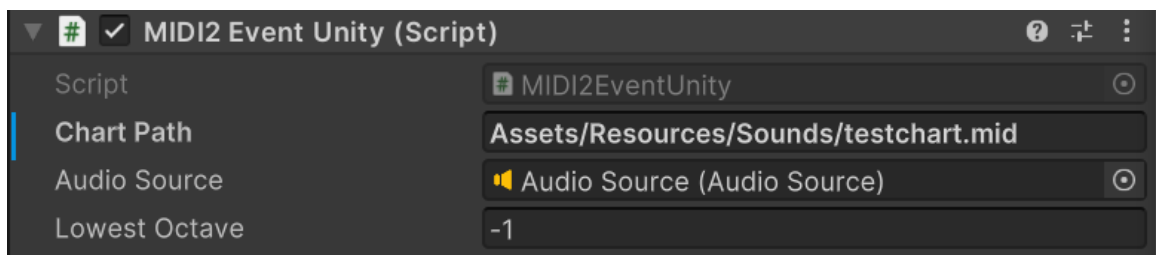
- Drag the audio asset you would like to use with the chart into the child Audio Source's AudioClip field.

- Specify the lowest octave available in your MIDI charting software in the MIDI2Event object's Lowest Octave field. In actual MIDI data, the lowest octave is always stored as -1, but some charting software uses a different lowest octave. This option is here to make working with those software easier.

Using midi2event:

Subscribing:
- To subscribe code to the midi2event system, you must use the Subscribe method. This method has one required and three optional parameters.
    - Action action - The code you want the midi2event system to trigger, in C# Action form.
    - Midi2Event.Notes note - The enum value corresponding to the note pitch you want to subscribe your action to.
    - int octave - The octave number containing the note you want to subscribe your action to.
    - Midi2Event.SubType type - What kind of event you want to subscribe your action to. Right now, you can subscribe to the start of a note, the end of a note, or the end of the chart.
- The Subscribe method returns an Action which you can invoke to unsubscribe the action from the system.
- You should subscribe in OnEnable and store the return values to unsubscribe in OnDisable.

```csharp
// Unity Message | 0 references
private void OnEnable()
{
    unsub = handler.Subscribe(Max, note, octave);
}

// Unity Message | 0 references
private void OnDisable()
{
    unsub.Invoke();
}
```

Playing/Stopping:
- You can start playing a chart and its associated music by using the Play method.
- You can stop a chart and its associated music from playing, while resetting them both, by using the Stop method.

Known Issues/Quirks:
- At least in the editor, starting the system too close to clicking the run button could lead to synchronization issues. It is recommended to have some trigger in-game that calls the Play method.
- Right now, the system only supports type 0 MIDI files that use ticks per quarter note. To my knowledge, this should be the most common format. You can read more [here](#).

Feature Requests:
- As this is a system I am hoping to release at some point, I would like to be the one who makes code modifications to it. If you would like to see a feature that does not already exist that relies on modifying the MIDI2Event class or lower, let me know and I can work on it!