Unity Android

You can download the package here
Create a Csharp script `MidiEventHandler` that inherits from `MonoBehaviour` and `IMidiEventHandler`. In the awake and start methods add these lines for instantiating the plugin :

```
private void Awake()
{
    gameObject.AddComponent<MidiManager>();
}

private void Start()
{
    MidiManager.Instance.RegisterEventHandler(this);
}
```

Then implements all the methods of `IMidiEventHandler` :

```
[SerializeField] private Text text;
// Called for all midi commands, to receive raw midi data, including before NoteOn and Note
public void RawMidi(sbyte command, sbyte data1, sbyte data2)
{
    string output = string.Format("MIDI command: {0:x2} {1:x2} {2:x2}", command, data1, dat
    Debug.Log(output);
    text.text += output + Environment.NewLine;
}

// Called when a midi event is received
public void RawMidi(sbyte a, sbyte b, sbyte c)
{
    Debug.Log("RawMidi a " + a + " b " + b + " c " + c);
    text.text += "RawMidi a " + a + " b " + b + " c " + c + Environment.NewLine;
}

// Called when you plug a midi note is down
public void NoteOn(int note, int velocity)
{
    Debug.Log("Note On " + note + " velocity " + velocity);
    text.text += "Note On " + note + " velocity " + velocity + Environment.NewLine;
}

// Called when you plug a midi note is released
public void NoteOff(int note)
{
    Debug.Log("Note off " + note);
    text.text += "Note off " + note + Environment.NewLine;
}

// Called when you plug a midi device
public void DeviceAttached(string deviceName)
{
    Debug.Log("Device Attached " + deviceName);
    text.text += "Device Attached " + deviceName + Environment.NewLine;
}
// Called when you unplug a midi device
public void DeviceDetached(string deviceName)
{
    Debug.Log("Device Detached " + deviceName);
    text.text += "Device Detached " + deviceName + Environment.NewLine;
}
```

In the editor, add a Text and add the previous script into your scene.

License