



...Gen — Python matrixMusic.py extracted\_mx1/moon.xml +

```
Now playing: ('B3', 'E4', 'G4')
Now playing: ('B3', 'D3', 'E4', 'G3', 'G4')
Now playing: ('A2', 'A4', 'C4#', 'E4')
Now playing: ('A4', 'C3#', 'C4#', 'E4')
Now playing: ('A4', 'C3#', 'C4#', 'E4')
Now playing: ('A5', 'C3#', 'E3')
Now playing: ('A5', 'E3')
Now playing: ('A3', 'G5')
Now playing: ('E3', 'E5')
Now playing: ('E5', 'G3')
Now playing: ('D3', 'D5', 'G3')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('A4', 'C5', 'F2', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
Now playing: ('A4', 'C5', 'F3', 'F4')
Now playing: ('A4', 'C5', 'E3', 'F3', 'F4')
Now playing: ('A4', 'C5', 'E3', 'F4')
Now playing: ('A3', 'D3', 'D4', 'F4')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
[fooMUSIC] ethansie@Ethans-MacBook-Pro xmlMusicGen % python
3 matrixMusic.py extracted_mx1/moon.xml
First measure, grid of notes:
Now playing: ('A3', 'A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A2', 'A3', 'D3', 'D4', 'F3', 'F4')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('E4', 'G1')
Now playing: ('C2', 'C4', 'G3')
```

readXML.py | xmlGenera... | matrixMusi... | requiremen... | refactored... | db.py | mxlConvert... | musicGen.py | log.txt | debug.py | forgetTest....

```
74
75         note_map = ['C', 'C#', 'D', 'D#', 'E', 'F',
76                      'F#', 'G', 'G#', 'A', 'A#', 'B']
77
78     return note_map.index(pitch) + 12 * (octave + 1)
79 except Exception as e:
80     print(f"Error converting note '{note}': {e}")
81     return None
82
83
84
85
86 SOUNDFONT_PATH = "../FluidR3_GM/FluidR3_GM.sf2"
87 fs = fluidsynth.Synth()
88 fs.start()
89 sfid = fs.sfload(SOUNDFONT_PATH)
90 fs.program_select(0, sfid, 0, 0)
91
92 def play_chords(chords, duration=0.1, velocity=100):
93     current = set()
94     for chord in chords:
95         print("Now playing:", chord)
96         next_notes = set(filter(None, (note_to_midi(n) for n in chord)))
97         for note in current - next_notes:
98             fs.noteoff(0, note)
99         for note in next_notes - current:
100             fs.noteon(0, note, velocity)
101         time.sleep(duration)
102         current = next_notes
103     for note in current:
104         fs.noteoff(0, note)
105
106 try:
107     play_chords(note_sequences, duration=0.2)
108 except KeyboardInterrupt:
109     print("Playback interrupted by user (Ctrl+C).")
110 finally:
111     fs.delete()
112     print("Fluidsynth resources cleaned up.")
113
```

matrixMusic.py 94:16

LF UTF-8 Python ⚡ main ⚡ Fetch ⚡ GitHub ⚡ Git (1) ⚡

matrixMusic.py 94:16

LF UTF-8 Python ⚡ main ⚡ Fetch ⚡ GitHub ⚡ Git (1) ⚡

TTS

TTS

TTS



```
...Gen — Python matrixMusic.py extracted_mxl/moon.xml +
```

```
readXML.py xmlGenera... matrixMusi... requiremen... refactored... db.py mxlConvert... musicGen.py log.txt debug.py forgetTest...
```

```
Now playing: ('B3', 'E4', 'G4')
Now playing: ('B3', 'D3', 'E4', 'G3', 'G4')
Now playing: ('A2', 'A4', 'C4#', 'E4')
Now playing: ('A4', 'C3#', 'C4#', 'E4')
Now playing: ('A4', 'C3#', 'C4#', 'E4')
Now playing: ('A5', 'C3#', 'E3')
Now playing: ('A5', 'E3')
Now playing: ('A5', 'E3')
Now playing: ('A3', 'G5')
Now playing: ('E3', 'E5')
Now playing: ('E5', 'G3')
Now playing: ('D3', 'D5', 'G3')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('A4', 'C5', 'F2', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F3', 'F4')
Now playing: ('A4', 'C5', 'F3', 'F4')
Now playing: ('A4', 'C5', 'E3', 'F3', 'F4')
Now playing: ('A4', 'C5', 'E3', 'F4')
Now playing: ('A3', 'D3', 'D4', 'F4')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
```

```
[fooMUSIC] ethansie@Ethans-MacBook-Pro xmlMusicGen % python]
3 matrixMusic.py extracted_mxl/moon.xml
First measure, grid of notes:
Now playing: ('A3', 'A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A2', 'A3', 'D3', 'D4', 'F3', 'F4')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('E4', 'G1')
```

```
74
75         note_map = ['C', 'C#', 'D', 'D#', 'E', 'F',
76                         'F#', 'G', 'G#', 'A', 'A#', 'B']
77
78     return note_map.index(pitch) + 12 * (octave + 1)
79 except Exception as e:
80     print(f"Error converting note '{note}': {e}")
81     return None
82
83
84
85
86 SOUNDFONT_PATH = "../FluidR3_GM/FluidR3_GM.sf2"
87 fs = fluidsynth.Synth()
88 fs.start()
89 sfid = fs.sfload(SOUNDFONT_PATH)
90 fs.program_select(0, sfid, 0, 0)
91
92 def play_chords(chords, duration=0.1, velocity=100):
93     current = set()
94     for chord in chords:
95         print("Now playing:", chord)
96         next_notes = set(filter(None, (note_to_midi(n) for n in chord)))
97         for note in current - next_notes:
98             fs.noteoff(0, note)
99         for note in next_notes - current:
100             fs.noteon(0, note, velocity)
101         time.sleep(duration)
102         current = next_notes
103     for note in current:
104         fs.noteoff(0, note)
105
106 try:
107     play_chords(note_sequences, duration=0.2)
108 except KeyboardInterrupt:
109     print("Playback interrupted by user (Ctrl+C).")
110 finally:
111     fs.delete()
112     print("Fluidsynth resources cleaned up.")
113
```

```
...Gen — Python matrixMusic.py extracted_mxl/moon.xml +
```

```
readXML.py xmlGenera... matrixMusi... requiremen... refactored... db.py mxlConvert... musicGen.py log.txt debug.py forgetTest...
```

```
Now playing: ('B3', 'E4', 'G4')
Now playing: ('B3', 'D3', 'E4', 'G3', 'G4')
Now playing: ('A2', 'A4', 'C4#', 'E4')
Now playing: ('A4', 'C3#', 'C4#', 'E4')
Now playing: ('A4', 'C3#', 'C4#', 'E4')
Now playing: ('A5', 'C3#', 'E3')
Now playing: ('A5', 'E3')
Now playing: ('A5', 'E3')
Now playing: ('A3', 'G5')
Now playing: ('E3', 'E5')
Now playing: ('E5', 'G3')
Now playing: ('D3', 'D5', 'G3')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('A4', 'C5', 'F2', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F3', 'F4')
Now playing: ('A4', 'C5', 'F3', 'F4')
Now playing: ('A4', 'C5', 'E3', 'F3', 'F4')
Now playing: ('A4', 'C5', 'E3', 'F4')
Now playing: ('A3', 'D3', 'D4', 'F4')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
```

```
[fooMUSIC] ethansie@Ethans-MacBook-Pro xmlMusicGen % python]
3 matrixMusic.py extracted_mxl/moon.xml
First measure, grid of notes:
Now playing: ('A3', 'A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A2', 'A3', 'D3', 'D4', 'F3', 'F4')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('E4', 'G1')
```

```
74
75         note_map = ['C', 'C#', 'D', 'D#', 'E', 'F',
76                         'F#', 'G', 'G#', 'A', 'A#', 'B']
77
78     return note_map.index(pitch) + 12 * (octave + 1)
79 except Exception as e:
80     print(f"Error converting note '{note}': {e}")
81     return None
82
83
84
85
86 SOUNDFONT_PATH = "../FluidR3_GM/FluidR3_GM.sf2"
87 fs = fluidsynth.Synth()
88 fs.start()
89 sfid = fs.sfload(SOUNDFONT_PATH)
90 fs.program_select(0, sfid, 0, 0)
91
92 def play_chords(chords, duration=0.1, velocity=100):
93     current = set()
94     for chord in chords:
95         print("Now playing:", chord)
96         next_notes = set(filter(None, (note_to_midi(n) for n in chord)))
97         for note in current - next_notes:
98             fs.noteoff(0, note)
99         for note in next_notes - current:
100             fs.noteon(0, note, velocity)
101         time.sleep(duration)
102         current = next_notes
103     for note in current:
104         fs.noteoff(0, note)
105
106 try:
107     play_chords(note_sequences, duration=0.2)
108 except KeyboardInterrupt:
109     print("Playback interrupted by user (Ctrl+C).")
110 finally:
111     fs.delete()
112     print("Fluidsynth resources cleaned up.")
113
```





...Gen — Python matrixMusic.py extracted\_mxl/moon.xml

```

Now playing: ('B3', 'E4', 'G4')
Now playing: ('B3', 'D3', 'E4', 'G3', 'G4')
Now playing: ('A2', 'A4', 'C4#', 'E4')
Now playing: ('A4', 'C3#', 'C4#', 'E4')
Now playing: ('A4', 'C3#', 'C4#', 'E4')
Now playing: ('A5', 'C3#', 'E3')
Now playing: ('A5', 'E3')
Now playing: ('A5', 'E3')
Now playing: ('A3', 'G5')
Now playing: ('E3', 'E5')
Now playing: ('E5', 'G3')
Now playing: ('D3', 'D5', 'G3')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('A4', 'C5', 'F2', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F3', 'F4')
Now playing: ('A4', 'C5', 'F3', 'F4')
Now playing: ('A4', 'C5', 'E3', 'F3', 'F4')
Now playing: ('A4', 'C5', 'E3', 'F4')
Now playing: ('A3', 'D3', 'D4', 'F4')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.

```

(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen % python

```

3 matrixMusic.py extracted_mxl/moon.xml
First measure, grid of notes:
Now playing: ('A3', 'A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A3', 'C4', 'E3')
Now playing: ('A2', 'A3', 'D3', 'D4', 'F3', 'F4')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A2', 'A3', 'D4', 'F3')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('A3', 'D4', 'G1')
Now playing: ('E4', 'G1')
Now playing: ('C2', 'C4', 'G3')

```

readXML.py | xmlGenera... | matrixMusi... | requiremen... | refactored... | db.py | mxlConvert... | musicGen.py | log.txt | debug.py | forgetTest...

```

74
75         note_map = ['C', 'C#', 'D', 'D#', 'E', 'F',
76                         'F#', 'G', 'G#', 'A', 'A#', 'B']
77
78     return note_map.index(pitch) + 12 * (octave + 1)
79 except Exception as e:
80     print(f"Error converting note '{note}': {e}")
81     return None
82
83
84
85
86 SOUNDFONT_PATH = "../FluidR3_GM/FluidR3_GM.sf2"
87 fs = fluidsynth.Synth()
88 fs.start()
89 sfid = fs.sfload(SOUNDFONT_PATH)
90 fs.program_select(0, sfid, 0, 0)
91
92 def play_chords(chords, duration=0.1, velocity=100):
93     current = set()
94     for chord in chords:
95         print("Now playing:", chord)
96         next_notes = set(filter(None, (note_to_midi(n) for n in chord)))
97         for note in current - next_notes:
98             fs.noteoff(0, note)
99         for note in next_notes - current:
100             fs.noteon(0, note, velocity)
101         time.sleep(duration)
102         current = next_notes
103     for note in current:
104         fs.noteoff(0, note)
105
106 try:
107     play_chords(note_sequences, duration=0.2)
108 except KeyboardInterrupt:
109     print("Playback interrupted by user (Ctrl+C).")
110 finally:
111     fs.delete()
112     print("Fluidsynth resources cleaned up.")
113

```

# Moon River

