

...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

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

TTS

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

Now playing: ('C2', 'C4', 'G3')




```
eadXML.py    xmlGenera...    matrixMusi...    requiremen...    refactored...    db.py    mxlConvert...    musicC
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.")
```

Moon River

```
...— Python matrixMusic.py extracted_mx1/christmas.xml + read
Now playing: ('A2', 'A3') 74
Now playing: ('A3', 'C3', 'F3') 75
Now playing: ('F2', 'F3') 76
Now playing: ('F2', 'F3') 77
Now playing: ('F2', 'F3') 78
Now playing: ('F2', 'F3') 79
Now playing: ('F2', 'F3') 80
Now playing: ('A2', 'C3', 'F3') 81
Now playing: ('A2', 'C3', 'F3') 82
Now playing: ('A3', 'C4', 'F1', 'F4') 83
Now playing: ('A3', 'C4', 'F1', 'F4') 84
Now playing: ('A3', 'C4', 'F1', 'F4') 85
Now playing: ('A3', 'C4', 'F1', 'F4') 86
Now playing: ('A3', 'C4', 'F1', 'F4') 87
Now playing: ('A3', 'C4', 'F1', 'F4') 88
Now playing: ('A3', 'C4', 'F1', 'F4') 89
Now playing: ('A3', 'C4', 'F1', 'F4') 90
Now playing: ('A3', 'C4', 'F1', 'F4') 91
Now playing: ('A3', 'C4', 'F1', 'F4') 92
Now playing: ('A3', 'C4', 'F1', 'F4') 93
Now playing: ('A3', 'C4', 'F1', 'F4') 94
Now playing: ('A3', 'C4', 'F1', 'F4') 95
Now playing: ('A3', 'C4', 'F1', 'F4') 96
Now playing: ('A3', 'C4', 'F1', 'F4') 97
Now playing: ('A3', 'C4', 'F1', 'F4') 98
Now playing: ('A3', 'C4', 'F1', 'F4') 99
Now playing: ('A3', 'C4', 'F1', 'F4') 100
Now playing: ('A3', 'C4', 'F1', 'F4') 101
Now playing: ('A3', 'C4', 'F1', 'F4') 102
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen % python
3 matrixMusic.py extracted_mx1/christmas.xml
First measure, grid of notes:
Now playing: ('B4', 'D3', 'D5', 'G3', 'G5') 105
Now playing: ('00',) 106
Now playing: ('00',) 107
Now playing: ('00',) 108
Now playing: ('00',) 109
Now playing: ('00',) 110
Now playing: ('00',) 111
Now playing: ('00',) 112
Now playing: ('00',) 113
Now playing: ('D2', 'D4', 'D5')
```

```
ML.py      xmlGenera... matrixMusi... requiremen... refactored... db.py      mxlConvert... musicGen.py

note_map = ['C', 'C#', 'D', 'D#', 'E', 'F',
            'F#', 'G', 'G#', 'A', 'A#', 'B']

    return note_map.index(pitch) + 12 * (octave + 1)
except Exception as e:
    print(f"Error converting note '{note}': {e}")
return None

SOUNDFONT_PATH = "../FluidR3_GM/FluidR3_GM.sf2"
fs = fluidsynth.Synth()
fs.start()
sfid = fs.sfload(SOUNDFONT_PATH)
fs.program_select(0, sfid, 0, 0)

def play_chords(chords, duration=0.1, velocity=100):
    current = set()
    for chord in chords:
        print("Now playing:", chord)
        next_notes = set(filter(None, (note_to_midi(n) for n in chord)))
        for note in current - next_notes:
            fs.noteoff(0, note)
        for note in next_notes - current:
            fs.noteon(0, note, velocity)
        time.sleep(duration)
        current = next_notes
    for note in current:
        fs.noteoff(0, note)

try:
    play_chords(note_sequences, duration=0.2)
except KeyboardInterrupt:
    print("Playback interrupted by user (Ctrl+C).")
finally:
    fs.delete()
    print("Fluidsynth resources cleaned up.")
```

All I want for Christmas