

[illegible]

```

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

```


[illegible]

```
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
```


[illegible]

```

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

```

All I want for Christmas

Temperature scaling