

I initially used sinusoidal
frequency waves.

That also sounded bad.

FluidSynth

A synthesizer.

- FluidSynth allows MIDI files to be played through SoundFonts (a public collection of sounds)
- Thus, to play sounds: a note to midi conversion was necessary

```
93  ✓ def play_chords(chords, duration=0.1, velocity=100):
94      current = set()
95      for chord in chords:
96          print("Now playing:", chord)
97          next_notes = set(filter(None, (note_to_midi(n) for n in chord)))
98          for note in current - next_notes:
99              fs.noteoff(0, note)
100         for note in next_notes - current:
101             fs.noteon(0, note, velocity)
102         time.sleep(duration)
103         current = next_notes
104     for note in current:
105         fs.noteoff(0, note)
```

Direct Music Generation Code

```
48  # ----- MIDI PLAYBACK -----
49  ✓ def note_to_midi(note):
50      if note in ["00", "0", "", None]:
51          return None
52      try:
53          note = note.replace('*', '') # Handle natural sign as a no-op
54
55          # Expect format like "D4b" or "C4#"
56          if len(note) < 2:
57              return None
58
59          pitch_letter = note[0]
60          if note[2:] in ('#', 'b'): # with accidental
61              octave = int(note[1])
62              accidental = note[2]
63          else:
64              octave = int(note[1])
65              accidental = ''
66
67          pitch = pitch_letter + accidental
68
69          # Convert flats to equivalent sharps
70          flat_to_sharp = {
71              'Cb': 'B', 'Db': 'C#', 'Eb': 'D#', 'Fb': 'E',
72              'Gb': 'F#', 'Ab': 'G#', 'Bb': 'A#'
73          }
74          pitch = flat_to_sharp.get(pitch, pitch)
75
76          note_map = ['C', 'C#', 'D', 'D#', 'E', 'F',
77                     'F#', 'G', 'G#', 'A', 'A#', 'B']
78
79          return note_map.index(pitch) + 12 * (octave + 1)
80      except Exception as e:
81          print(f"Error converting note '{note}': {e}")
82          return None
```