```
... - Python matrixMusic.py extracted_mxl/christmas.xml
                                                                                                                                                                                          forgetTest....
                                                             readXML.py
                                                                                     matrixMusi... requiremen... refactored..
                                                                        xmlGenera...
                                                                                                                                       mxlConvert... musicGen.py
                                                                                                                                                                   log.txt
Now playing: ('A2', 'A3')
Now playing: ('A3', 'C3', 'F3')
                                                                           note_map = ['C', 'C#', 'D', 'D#', 'E', 'F',
Now playing: ('F2', 'F3')
                                                                                         'F#', 'G', 'G#', 'A', 'A#', 'B']
Now playing: ('F2', 'F3')
Now playing: ('F2', 'F3')
Now playing: ('F2', 'F3')
                                                                           return note_map.index(pitch) + 12 * (octave + 1)
Now playing: ('F2', 'F3')
Now playing: ('F2', 'F3')
                                                                      except Exception as e:
Now playing: ('F2', 'F3')
Now playing: ('A2', 'C3', 'F3')
                                                                           print(f"Error converting note '{note}': {e}")
Now playing: ('A2', 'C3', 'F3')
                                                                           return None
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  SOUNDFONT_PATH = "../FluidR3_GM/FluidR3_GM.sf2"
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  fs = fluidsynth.Synth()
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  fs.start()
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  sfid = fs.sfload(SOUNDFONT_PATH)
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  fs.program_select(0, sfid, 0, 0)
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  def play_chords(chords, duration=0.1, velocity=100):
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                      current = set()
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                      for chord in chords:
                                                             94
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           print("Now playing:", chord)
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           next_notes = set(filter(None, (note_to_midi(n) for n in chord)))
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           for note in current - next_notes:
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                                fs.noteoff(0, note)
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           for note in next_notes - current:
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                                fs.noteon(0, note, velocity)
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           time.sleep(duration)
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           current = next_notes
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
                                                                      for note in current:
[(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen % python]
                                                                           fs.noteoff(0, note)
3 matrixMusic.py extracted_mxl/christmas.xml
First measure, grid of notes:
Now playing: ('B4', 'D3', 'D5', 'G3', 'G5')
Now playing: ('00',)
                                                                  try:
Now playing: ('00',)
                                                                      play_chords(note_sequences, duration=0.2)
Now playing: ('00',)
Now playing: ('00',)
                                                                 except KeyboardInterrupt:
Now playing: ('00',)
                                                                      print("Playback interrupted by user (Ctrl+C).")
Now playing: ('00',)
Now playing: ('00',)
                                                            110 finally:
Now playing: ('00',)
                                                                      fs.delete()
Now playing: ('00',)
Now playing: ('00',)
                                                                      print("Fluidsynth resources cleaned up.")
Now playing: ('00',)
Now playing: ('D2', 'D4', 'D5')
                                                                                                                                        LF UTF-8 Python 🖟 main 🖒 Fetch 🎧 GitHub 🗢 Git (1) 🌣
                                                            matrixMusic.py 94:16
```



```
... - Python matrixMusic.py extracted_mxl/christmas.xml
                                                                                                                                                                                          forgetTest....
                                                             readXML.py
                                                                                     matrixMusi... requiremen... refactored..
                                                                        xmlGenera...
                                                                                                                                       mxlConvert... musicGen.py
                                                                                                                                                                   log.txt
Now playing: ('A2', 'A3')
Now playing: ('A3', 'C3', 'F3')
                                                                           note_map = ['C', 'C#', 'D', 'D#', 'E', 'F',
Now playing: ('F2', 'F3')
                                                                                         'F#', 'G', 'G#', 'A', 'A#', 'B']
Now playing: ('F2', 'F3')
Now playing: ('F2', 'F3')
Now playing: ('F2', 'F3')
                                                                           return note_map.index(pitch) + 12 * (octave + 1)
Now playing: ('F2', 'F3')
Now playing: ('F2', 'F3')
                                                                      except Exception as e:
Now playing: ('F2', 'F3')
Now playing: ('A2', 'C3', 'F3')
                                                                           print(f"Error converting note '{note}': {e}")
Now playing: ('A2', 'C3', 'F3')
                                                                           return None
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  SOUNDFONT_PATH = "../FluidR3_GM/FluidR3_GM.sf2"
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  fs = fluidsynth.Synth()
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  fs.start()
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  sfid = fs.sfload(SOUNDFONT_PATH)
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  fs.program_select(0, sfid, 0, 0)
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                  def play_chords(chords, duration=0.1, velocity=100):
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                      current = set()
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                      for chord in chords:
                                                             94
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           print("Now playing:", chord)
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           next_notes = set(filter(None, (note_to_midi(n) for n in chord)))
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           for note in current - next_notes:
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                                fs.noteoff(0, note)
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           for note in next_notes - current:
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                                fs.noteon(0, note, velocity)
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           time.sleep(duration)
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                           current = next_notes
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
                                                                      for note in current:
[(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen % python]
                                                                           fs.noteoff(0, note)
3 matrixMusic.py extracted_mxl/christmas.xml
First measure, grid of notes:
Now playing: ('B4', 'D3', 'D5', 'G3', 'G5')
Now playing: ('00',)
                                                                  try:
Now playing: ('00',)
                                                                      play_chords(note_sequences, duration=0.2)
Now playing: ('00',)
Now playing: ('00',)
                                                                 except KeyboardInterrupt:
Now playing: ('00',)
                                                                      print("Playback interrupted by user (Ctrl+C).")
Now playing: ('00',)
Now playing: ('00',)
                                                            110 finally:
Now playing: ('00',)
                                                                      fs.delete()
Now playing: ('00',)
Now playing: ('00',)
                                                                      print("Fluidsynth resources cleaned up.")
Now playing: ('00',)
Now playing: ('D2', 'D4', 'D5')
                                                                                                                                        LF UTF-8 Python 🖟 main 🖒 Fetch 🎧 GitHub 🗢 Git (1) 🌣
                                                            matrixMusic.py 94:16
```

```
... — Python matrixMusic.py extracted_mxl/christmas.xml
                                                                                                                                                                                       forgetTest....
                                                            readXML.py
                                                                        xmlGenera...
                                                                                    matrixMusi... requiremen... refactored...
                                                                                                                                     mxlConvert... musicGen.py
                                                                                                                                                                 log.txt
Now playing: ('A2', 'A3')
Now playing: ('A3', 'C3', 'F3')
                                                                          note_map = ['C', 'C#', 'D', 'D#', 'E', 'F',
Now playing: ('F2', 'F3')
                                                                                        'F#', 'G', 'G#', 'A', 'A#', 'B']
Now playing: ('F2', 'F3')
Now playing: ('F2', 'F3')
Now playing: ('F2', 'F3')
                                                                          return note_map.index(pitch) + 12 * (octave + 1)
Now playing: ('F2', 'F3')
Now playing: ('F2', 'F3')
                                                                     except Exception as e:
Now playing: ('F2', 'F3')
Now playing: ('A2', 'C3', 'F3')
                                                                          print(f"Error converting note '{note}': {e}")
Now playing: ('A2', 'C3', 'F3')
                                                                          return None
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                 SOUNDFONT_PATH = "../FluidR3_GM/FluidR3_GM.sf2"
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                 fs = fluidsynth.Synth()
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                 fs.start()
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                 sfid = fs.sfload(SOUNDFONT_PATH)
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                 fs.program_select(0, sfid, 0, 0)
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                 def play_chords(chords, duration=0.1, velocity=100):
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                     current = set()
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                     for chord in chords:
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                            94
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                          print("Now playing:", chord)
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                          next_notes = set(filter(None, (note_to_midi(n) for n in chord)))
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                          for note in current - next_notes:
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                               fs.noteoff(0, note)
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                          for note in next_notes - current:
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                               fs.noteon(0, note, velocity)
Now playing: ('A3', 'C4', 'F1', 'F4')
Now playing: ('A3', 'C4', 'F1', 'F4')
                                                                          time.sleep(duration)
Now playing: ('A3', 'C4', 'F1', 'F4')
^CPlayback interrupted by user (Ctrl+C).
                                                                          current = next_notes
Fluidsynth resources cleaned up.
                                                                     for note in current:
[(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen % python]
                                                                          fs.noteoff(0, note)
3 matrixMusic.py extracted_mxl/christmas.xml
First measure, grid of notes:
Now playing: ('B4', 'D3', 'D5', 'G3', 'G5')
Now playing: ('00',)
                                                                 try:
Now playing: ('00',)
                                                                     play_chords(note_sequences, duration=0.2)
                                                                                                                                                                         All I want
Now playing: ('00',)
Now playing: ('00',)
                                                                 except KeyboardInterrupt:
                                                                                                                                                                        All I Walle
Now playing: ('00',)
                                                                     print("Playback interrupted by user (Ctrl+C).")
                                                                                                                                                                                        for
Now playing: ('00',)
                                                           110 finally:
Now playing: ('00',)
Now playing: ('00',)
                                                                     fs.delete()
                                                                                                                                                                     Christmas
Now playing: ('00',)
                                                                     print("Fluidsynth resources cleaned up.")
Now playing: ('00',)
Now playing: ('00',)
Now playing: ('D2', 'D4', 'D5')
                                                                                                                                      LF UTF-8 Python 🖟 main 🖒 Fetch 🎧 GitHub 🗢 Git (1) 🌣
                                                           matrixMusic.py 94:16
```

## Temperature scaling