~/xmlMusicGen — -zsh

```
Now playing: ('B2', 'D4')
Now playing: ('B2', 'D4', 'G2')
Now playing: ('D4', 'G2')
Now playing: ('E2', 'E4')
Now playing: ('E4', 'G2')
Now playing: ('E4', 'G2')
Now playing: ('A4', 'C5', 'F2', 'F4')
Now playing: ('A2', 'A4', 'C5', 'F4')
Now playing: ('A4', 'B2', 'C5')
Now playing: ('D3', 'F4')
Now playing: ('C5', 'D3', 'E3', 'E5', 'F4')
Now playing: ('C5', 'E3', 'E5')
Now playing: ('C5', 'E3', 'E5')
Now playing: ('C5', 'E3', 'E5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('A4', 'C3', 'C5')
Now playing: ('A4', 'B2', 'B4', 'C3', 'C5', 'G4')
Now playing: ('B2', 'B4', 'G4')
Now playing: ('B2', 'B4', 'G4')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D3', 'D4', 'G3')
Now playing: ('B3', 'D3', 'D4', 'G3')
Now playing: ('B3', 'D3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen % python3 matrixMusic.py extracted_mxl/moon.xml
First measure, grid of notes:
Now playing: ('C3', 'E5')
Now playing: ('E3', 'E5')
Now playing: ('E3', 'E5')
Now playing: ('E5', 'G3')
Now playing: ('D3', 'D5', 'G3')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('A4', 'C5', 'F2', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
Now playing: ('A2', 'A4', 'C3', 'C5', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F3', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen %
```

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

```python
import xmlGenerate
import random
import numpy as np
import fluidsynth
import time

matrix = xmlGenerate.getMatrix()
chord_list = xmlGenerate.getChordList()
chord_index = xmlGenerate.getChordIndex()

# MATRIX MANIPULATION
# Higher = more entropy, lower = less change
# Emphasizes/Minimizes the existing row probability vectors
def scale_temperature(matrix, temperature=1.0):
    assert temperature > 0, "Temperature must be positive"
    log_matrix = np.log(matrix + 1e-9)  # Avoid log(0)
    scaled = np.exp(log_matrix / temperature)
    scaled = np.maximum(scaled, 0)
    scaled /= scaled.sum(axis=1, keepdims=True)
    return scaled

def inject_noise(matrix, epsilon=0.01):
    noisy = matrix + epsilon * np.random.rand(*matrix.shape)
    noisy /= noisy.sum(axis=1, keepdims=True)  # Renormalize rows
    return noisy



matrix = scale_temperature(matrix, 2.5)
# matrix = inject_noise(matrix, 0.0001)


# ———————— TRAJECTORY THROUGH THE ROW STOCHASTIC MATRIX ————————
initial = random.choice(chord_list)
generated = [initial]

for _ in range(1000):
    i = chord_index[initial]
    probs = matrix[i]
    j = np.random.choice(len(chord_list), p=probs)
```

matrixMusic.py 107:45 LF UTF-8 Python main Fetch GitHub Git (1)

~/xmlMusicGen — -zsh

```
Now playing: ('B2', 'D4')
Now playing: ('B2', 'D4', 'G2')
Now playing: ('D4', 'G2')
Now playing: ('E2', 'E4')
Now playing: ('E4', 'G2')
Now playing: ('E4', 'G2')
Now playing: ('A4', 'C5', 'F2', 'F4')
Now playing: ('A2', 'A4', 'C5', 'F4')
Now playing: ('A4', 'B2', 'C5')
Now playing: ('D3', 'F4')
Now playing: ('C5', 'D3', 'E3', 'E5', 'F4')
Now playing: ('C5', 'E3', 'E5')
Now playing: ('C5', 'E3', 'E5')
Now playing: ('C5', 'E3', 'E5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('A4', 'C3', 'C5')
Now playing: ('A4', 'B2', 'B4', 'C3', 'C5', 'G4')
Now playing: ('B2', 'B4', 'G4')
Now playing: ('B2', 'B4', 'G4')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D3', 'D4', 'G3')
Now playing: ('B3', 'D3', 'D4', 'G3')
Now playing: ('B3', 'D3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen % python3 matrixMusic.py extracted_mxl/moon.xml
First measure, grid of notes:
Now playing: ('C3', 'E5')
Now playing: ('E3', 'E5')
Now playing: ('E3', 'E5')
Now playing: ('E5', 'G3')
Now playing: ('D3', 'D5', 'G3')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('A4', 'C5', 'F2', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
Now playing: ('A2', 'A4', 'C3', 'C5', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F3', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen %
```

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

```python
import xmlGenerate
import random
import numpy as np
import fluidsynth
import time

matrix = xmlGenerate.getMatrix()
chord_list = xmlGenerate.getChordList()
chord_index = xmlGenerate.getChordIndex()

# MATRIX MANIPULATION
# Higher = more entropy, lower = less change
# Emphasizes/Minimizes the existing row probability vectors
def scale_temperature(matrix, temperature=1.0):
    assert temperature > 0, "Temperature must be positive"
    log_matrix = np.log(matrix + 1e-9)  # Avoid log(0)
    scaled = np.exp(log_matrix / temperature)
    scaled = np.maximum(scaled, 0)
    scaled /= scaled.sum(axis=1, keepdims=True)
    return scaled

def inject_noise(matrix, epsilon=0.01):
    noisy = matrix + epsilon * np.random.rand(*matrix.shape)
    noisy /= noisy.sum(axis=1, keepdims=True)  # Renormalize rows
    return noisy



matrix = scale_temperature(matrix, 2.5)
# matrix = inject_noise(matrix, 0.0001)


# ———————— TRAJECTORY THROUGH THE ROW STOCHASTIC MATRIX ————————
initial = random.choice(chord_list)
generated = [initial]

for _ in range(1000):
    i = chord_index[initial]
    probs = matrix[i]
    j = np.random.choice(len(chord_list), p=probs)
```

matrixMusic.py 107:45 LF UTF-8 Python main Fetch GitHub Git (1)

~/xmlMusicGen — -zsh

```
Now playing: ('B2', 'D4')
Now playing: ('B2', 'D4', 'G2')
Now playing: ('D4', 'G2')
Now playing: ('E2', 'E4')
Now playing: ('E4', 'G2')
Now playing: ('E4', 'G2')
Now playing: ('A4', 'C5', 'F2', 'F4')
Now playing: ('A2', 'A4', 'C5', 'F4')
Now playing: ('A4', 'B2', 'C5')
Now playing: ('D3', 'F4')
Now playing: ('C5', 'D3', 'E3', 'E5', 'F4')
Now playing: ('C5', 'E3', 'E5')
Now playing: ('C5', 'E3', 'E5')
Now playing: ('C5', 'E3', 'E5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('B4', 'D3', 'D5')
Now playing: ('A4', 'C3', 'C5')
Now playing: ('A4', 'B2', 'B4', 'C3', 'C5', 'G4')
Now playing: ('B2', 'B4', 'G4')
Now playing: ('B2', 'B4', 'G4')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('A3', 'B3', 'D4', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D3', 'D4', 'G3')
Now playing: ('B3', 'D3', 'D4', 'G3')
Now playing: ('B3', 'D3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
Now playing: ('B3', 'D4', 'F3', 'G3')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen % python3 matrixMusic.py extracted_mxl/moon.xml
First measure, grid of notes:
Now playing: ('C3', 'E5')
Now playing: ('E3', 'E5')
Now playing: ('E3', 'E5')
Now playing: ('E5', 'G3')
Now playing: ('D3', 'D5', 'G3')
Now playing: ('D3', 'D5')
Now playing: ('D3', 'D5')
Now playing: ('A4', 'C5', 'F2', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
Now playing: ('A2', 'A4', 'C3', 'C5', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F3', 'F4')
Now playing: ('A4', 'C3', 'C5', 'F4')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen %
```

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

```python
import xmlGenerate
import random
import numpy as np
import fluidsynth
import time

matrix = xmlGenerate.getMatrix()
chord_list = xmlGenerate.getChordList()
chord_index = xmlGenerate.getChordIndex()

# MATRIX MANIPULATION
# Higher = more entropy, lower = less change
# Emphasizes/Minimizes the existing row probability vectors
def scale_temperature(matrix, temperature=1.0):
    assert temperature > 0, "Temperature must be positive"
    log_matrix = np.log(matrix + 1e-9)  # Avoid log(0)
    scaled = np.exp(log_matrix / temperature)
    scaled = np.maximum(scaled, 0)
    scaled /= scaled.sum(axis=1, keepdims=True)
    return scaled

def inject_noise(matrix, epsilon=0.01):
    noisy = matrix + epsilon * np.random.rand(*matrix.shape)
    noisy /= noisy.sum(axis=1, keepdims=True)  # Renormalize rows
    return noisy



matrix = scale_temperature(matrix, 2.5)
# matrix = inject_noise(matrix, 0.0001)


# ———————— TRAJECTORY THROUGH THE ROW STOCHASTIC MATRIX ————————
initial = random.choice(chord_list)
generated = [initial]

for _ in range(1000):
    i = chord_index[initial]
    probs = matrix[i]
    j = np.random.choice(len(chord_list), p=probs)
```

matrixMusic.py 107:45 LF UTF-8 Python main Fetch GitHub Git (1)

Moon River (high temp)

What else can you do?

If you didn’t waste your time on FFT

- Create multiple matrices of one song at different time intervals. Helpful for decomposing a song and understanding how it changes over time.
- Isolate bottom and top staffs to better analyze patterns.
- Create one matrix for only the top staff and create another matrix that represents the transition states from the top staff to the bottom staff and use that matrix to randomize the chord progression.
- Try completely polarizing the matrix (an extreme temperature scaling)
- Use the constructed matrix and completely randomize the probability vectors.