R2B09 ACTIVITY LOG N02 *in* input_dataset/Dianne/Sampler.py

* Dianne -> acronym for **D**ataset **I**nitiation **A**lgorithm for **N**ominal **N**ote **E**xtrapolation. The project constructs the initial chord dataset.
* No person named Dianne made any of this code.
* This file randomly samples from the chord dataset. **Source code begins below.** Language is **Python 3.6.2**.

```
import scale
import chord
import ChordTypes
import random
listOfChordsOfChordTypes = []
listOfChords = []
sampleChords = []
flatSampleChords = []
ct = []
SAMPLE_SIZE = 3 # temporary value for now

def chordgen():
    for rootNote in scale.TWELVE_NOTE_SCALE:
        for chordType in scale.CHORD_TYPES:
            listOfChords.append(chord.Chord(rootNote, chordType))
    print(listOfChords)

def sample(SAMPLE_SIZE):
    # Makes list of list of chords. The top level list is broken down into lists of chords grouped by type
    for type in range(0, ChordTypes.NUMBER_OF_CHORD_TYPES):
        listOfChordsOfChordTypes.append([])
        for chord in range(0, ChordTypes.NUMBER_OF_CHORDS, ChordTypes.NUMBER_OF_CHORD_TYPES):
            listOfChordsOfChordTypes[type].append(listOfChords[chord + type])
    print(listOfChordsOfChordTypes)
    print(len(listOfChordsOfChordTypes))
    for ctListIndex in range(len(listOfChordsOfChordTypes)):
        sampleChords.append(random.sample(listOfChordsOfChordTypes[ctListIndex], SAMPLE_SIZE))
    flatSampleChords = [_chord_ for _chordList_ in sampleChords for _chord_ in _chordList_]
    print(sampleChords)
    print(flatSampleChords)
    print("Sample size:", str(len(flatSampleChords)), "(" , str(ChordTypes.NUMBER_OF_CHORD_TYPES), "chord types *",
SAMPLE_SIZE, "chords per type)")

if __name__ == "__main__":
    chordgen()
    if SAMPLE_SIZE < 12:
        sample(SAMPLE_SIZE)
    else:
        print("Sample invalid.")
```