WFCM

- Project Idea
    - Use wave function collapse algorithm to procedurally generate music
- Technologies
    - C++/Java
- Implementation

Implementation Notes

- DATA STRUCTURE
    - need a data structure that will hold note relations (like a table)
        - rows representing unique values
        - columns represent possible notes to transition to
    - maybe just a 2d array?
    - needs to be able to hold note information and have quick lookup times
        - we could make a dictionary between notes/values (using a hash table) to translate note queries into array indices

        - or we could sort the note arrays in some logical order so we can leverage that to get O(1) lookup

- READING MIDI FILE
    - if we use the dictionary approach, probably need 2 passes
        - first pass to create dictionary and count unique values
        - second pass to insert dependencies in the table structure
    - on seeing a new note
        - make a dictionary element
        - add it to unique value count
    - for second pass, after seeing a note following another note, increment the respective table item
- MARKOV CHAIN
    - normalize table elements by dividing each row element by the row sum
    - then a random roll between [0,1] could be used to select a transition
- WFC
    - initialize N notes as a superposition of all notes
        - using an array
        - each note node should have references to prev and next note
    - randomly select a note and define it
        - collapse
            - i.e. remove neighbor notes that cannot exist

            - modify prev and next probabilities based on the current note
            - recursively propagate constraints until no more can be
        - repeat until the piece is finished
- INTERFACE
    - start with command line
    - LOAD .MIDI
        - bring in the file and generate up to MARKOV CHAIN
    - GENERATE
        - generate music of notes
        - play the generated piece
        - SAVE (y/n)
    - HELP
        - give command information