GENESONG

The following song generation process is done thirty times per book:

SONG GENERATION PROCESS

1. A key is chosen. I decided to limit the book's choices to major keys, particularly Ab major, Eb major, Bb major, D major, C major, F major, and G major. This was mostly because minor keys would require different chord progressions, and would mean a lot more work and conditional statements. Throughout the book, every pitch was represented by an integer, with middle C being 0 and each half step up being an increment of 1. The keys for each song were represented by the integer of their base note, so Ab was -5, Eb was 3, etc. For this project, I disallowed changing keys mid-song (I also chose not to actually put key signatures at the beginning of each line, but rather to put sharps and flats on each individual note, just for simplicity's sake).

2. A chord progression is chosen. I limited the options to progressions with four steps, just so that the music would all fit a general form. I picked progressions that are pleasant/generally popular: I-ii-IV-V , I-vi-ii-V, I-IV-V-V, I-V-vi-IV, and I-vi-IV-V.

3. A rhyme scheme is chosen. I wanted to keep these very simple, so that the rhymes can be identified despite what will be fairly nonsense lyrics. So, the only options are "ABAB", "AAAA", and "AABB." The rhyme schemes affect not only the lyrics, but the music, because in general, rhyming lines in songs tend to feature parallel rhythms. That is not a rule in music, but for this project, I decided to make it one.

4. A "rhythm line" is generated. When a rhythm line is created, it is told how many beats it needs to fill. This amount will be four times the number of measures in the line (which is a random int between 3 and 5 inclusive). All songs are 4/4 time, just to keep things simple. A rhythm line knows how many elements (notes and rests) there are, how long each one lasts, and whether each element is a note or a rest. This was all randomly generated, but with a few parameters: the total length must be the length that was passed in (12, 16, or 20), the only possible lengths for elements are 0.5, 1, 2, 3, and 4 (I didn't want to deal with dotted quarter notes, ties, etc, but as you'll see I failed in that), the line cannot start with a rest, the line cannot end with a rest, and a rest cannot be longer than 1 beat (with such short lines and so little musical complexity, I didn't want to make anyone sit through a long rest).

5. More rhythm lines are generated such that there are four per verse and four per chorus, and lines that rhyme have identical rhythm lines.

6. Each note is given an integer pitch based on what chord it is from (which is calculated from the key combined with where we are in the chord progression). Each line starts with the base note of said chord, just to have a nice musical landmark every few measures. Rests are given a pitch value of 100 (which is invalid).

7. Lyrics are generated via RiTa's random word generator. I had hoped to make these coherent Markov-based sentences, but making those fit syllabically with the music and also fit in the rhyme scheme was too difficult to achieve (I tried a loop that kept re-generating lines until one worked, and would repeat this process for each song, and it often ran for several minutes with no positive results. This happened no matter what texts I put into the Markov chain, and for time's sake I had to simplify things). So, each note gets one one-syllable word, about 80% of which are from RiTa's lexicon, and 20% of which are randomly pieced together from an array I made of word beginnings (eg. "br","ch", "st") and word ends (eg. "ess", "ind", "ay"). If a word is at the end of the line, it will 100% of the time be one of the piece-meal words. Then, if it needs to rhyme with something else, it is just given the same word ending and a different beginning. When a line is complete, the first word is capitalized and a random punctuation mark is added. Random hyphens are also added (only after words that are non-lexicon) to imitate multiple-syllabled words. (Three sets of lyrics are generated for each line in the verses, one for the chorus).

8. A title is chosen for the song from the first 1-4 words of the chorus.

Once all 30 songs are generated, we have to display each one. Here are those steps.

SONG DISPLAY PROCESS

1. Print the title at the top of the page (if this is a verse page). Then, find your starting point (passed in as a parameter). Calculate how far apart every element in this line needs to be, based on how wide the page is and how many elements there are.

2. For each element, figure out what image you need (based on whether or not it’s a rest, how long the element is, and what pitch the element is). If you have room in the measure (kept track of by a measure-beat-counter--each measure needs exactly four beats), just put that image on the page, calculating its x position based on which element in the line this is, and its y position based on its pitch and the distance between the staff lines). Also determine whether you

need to add a flat or sharp. If you do NOT have room, put down however much you have room for, draw the measure line, and then put down the rest, then draw a tie between them. This part of the code was much more difficult than I thought, because it forced me to use all the things I was trying to avoid before, like dotted quarter notes, ties, and trying to express things like "3.5 beats" with as little superfluous notation as possible. When I used multiple note symbols for one element, I sometimes had to cram them together (see below) so as not to exceed the amount of horizontal space that one element was given. I should have done more to avoid situations like this in my original generation of the rhythms. Ultimately, I accounted for this by reducing the maximum number of possible measures per line from 7 to 5, just to ensure a reasonable amount of space. Also, the lyrics for each note were added to the document at the same time as their note and at the same x position.

3. Do the above step for every line. Then, add things like the staff lines themselves, the treble clef, the time signature, the repeat signs, the page number, and (this was a late addition) the chord names above the lines.

4. A fun addition: add a randomly generated adverb from RiTa's lexicon at the beginning of the piece. I did this mostly to poke fun at sheet music that demands to be played "nonchalantly" or "youthfully" or other vague terms like that. Some of the generated adverbs made some sense in context, and others (see below) did not, like "architecturally," or "ironically." I prefer it this way.

Once all of the songs were displayed (actually, this is done first, but I didn't want to wreck the narrative I had going), a table of contents is generated using the title of every song and its respective page number. This is all put together and exported as one .pdf file. The cover was printed separately.