# 3 - Methodology

## 3.1 - Reading in MIDI files

* Read in midi file
* Midi file made up of various events
  + Note start
  + Note end
  + How hard the note is pressed
* Extract note on/off events
* Created specific struct to hold the information
* As midi files don’t have specific information for rests
* Take the difference between notes as rest, create them as normal notes but the pitch is -1
* The key the inputted song is in is also known
* To make change keys easier later on each note’s pitch gets reduced down to the key of C (if original pitch is A, each note’s pitch is subtracted by 9, the semitone difference between A and C)
* Would also allow multiple songs to be combined at the read in stage

## 3.2 - Markov Chains

### 3.2.1 - Frequency Distributions

The next step in the process is to calculate the frequency distribution of the notes pairs in the inputted song. For each note pair it checks if it is a unique pairing, if this is not the case it increases the frequency counter for that pairing by 1, if it is unique then it creates a new instance of DependHolder and adds that to the list of note pairs. For each note it then sums the number of possible next notes.

### 3.2.2 - Choosing Notes

When choosing notes some stuff happens.

## 3. - Questionnaire