AI Assignment 2 Report

Created by Matvey Abramov - m.abramov@innopolis.university

Music Structure

I started with creating classes for common music concepts: `Note`, `Pitch`, `PitchClass` and `Scale`.

- `PitchClass` represents one of 12 possible notes
- `Pitch` represents `PitchClass` together with octave
- `Note` represents `Pitch`, velocity and duration info
- `Scale` represents minor or major scales, it can generate chords from root note which fit to scale. Logic for generating chords I took from assignment description. I use root note and integer notation for major, minor, dim, sus2 and sus4 chords for generation

Later I added `possible_scales` function. It finds all unique notes in melody and chooses scales which have all these notes

Working with Midi

Next step is convert midi to my music classes and vica versa. I use python package `mido` for working with midi.

Genetic Algorithm (GA)

Firstly I wrote general structure of all GA:

- `Gene` abstract class, can mutate
- `Chromosome` stores genes. Has following methods:
  - `mutate` mutates some genes and randomly chooses 2 genes and swaps them
  - `crossover` produces 2 chromosomes. Takes 2 chromosomes-parents, splits them in 2 parts, swaps parts
  - `generate` returns chromosome using provided function for generating gene

- `GA` takes fitness function, gene generating function and optional selection function (default is tournament selection). Has following methods:
  - `run` generates starting population, and starts the loop of generating fitnesses, selecting new population, crossover and mutations and updating population

Next, I wrote music specific `ChordGene`, which is an array of 3 notes. It mutates by randomly selecting chord from scale fitted chords.

Chromosome size determined by melody length, where each chord playing in each bar of the melody.

Fitness function

Firstly, I create `bar_to_notes` dictionary quickly find which notes are played chord. I have several fitness functions which are all combined into one:

- `fit_to_scale` rewards note of the chords on being on scale
- `fit_to_scale_chords` rewards the chord for being one of the fitted in scale chords
- `fit_to_current_notes` punishes dissonance intervals and fits chords to notes in bar
- `punish_same_notes` punishes less then 3 notes in chord
- `num_of_unique_chords` rewards if number of unique chords between MIN and MAX (4 and 6 by default)
- `same_chord` rewards same chord playing again

Run

I run my genetic algorithm for 2000 iterations, with the population size of 20 with tournament selection and mutation chance of chromosome and gene is 0.5. It took about 3-4 runs for each melody to find results which are most appealing to me. Output2 is the best of them