

Introduction to Artificial Intelligence

Assignment 2: Accompaniment Generation

1. Description:

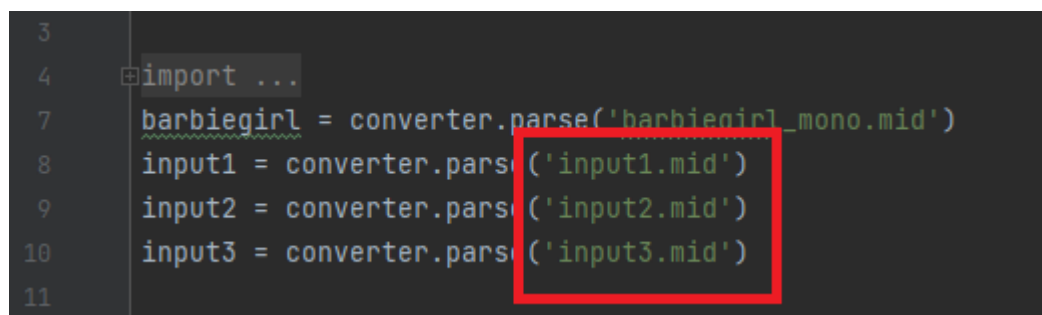
This program creates an accompaniment using a genetic algorithm.

2. How to start:

The program was tested on the Windows 11 operating system. Jupyter notebook was chosen as the development environment, since right inside it you can run midi files and track the result of the algorithm, as well as test individual blocks of the program. For correct operation, you need to install the music21 library. You can do this by running the following code:

```
pip install music21
```

If you want to change the music to which you want to generate the accompaniment, simply at the beginning of the file change the path to the one where the new music is located.



```
3
4 import ...
7 barbiegirl = converter.parse('barbiegirl_mono.mid')
8 input1 = converter.parse('input1.mid')
9 input2 = converter.parse('input2.mid')
10 input3 = converter.parse('input3.mid')
11
```

Or leave it as is and run. Now you have 3 midi output files.

3. How it works:

First of all, we must determine the tonic and mode of the piece. Next, break it down into notes and determine the duration for each. In the next step, we create a frame for the accompaniment, namely, depending on the notes and their duration, we select the main note for each 1/4 measure. At this stage, the genetic algorithm comes to the rescue. Based on our framework for each position, we use a genetic algorithm to generate the most appropriate chord. To do this, we create the first generation of chords, each containing 3 random sounds (from 0 to 127). On the basis of the first generation, we generate the second, on the basis of the second - the third, and so

on until we get the "ideal" chord or close to it. In order to determine how close the current result is to the desired one, we use the fitness function. The fitness function works like this: from the triad of a perfect chord, we subtract the generated values (midi values):

```
def calc_fitness(target, gnome):
    fitness =
    abs(target[0]-gnome[0])+abs(target[1]-gnome[1])+abs(target[2]-gnome[2])
    return fitness
```

4. Result:

As a result of the algorithm, we received three generated files with accompaniments.