Sequence of Plans

Title: User-Oriented Symbolic Algorithmic Composition

First Plan - February 9, 2023

Task 0 through 2 create the necessary knowledge and infrastructure needed for creating a music individual consisting of three different melodies. The initial constraint system will guide melody generation, with music theory rules for harmonization of the alternate melody with the primary melody, as well as confinements for octaves and note durations. Task 3 mutates all three melodies in a music individual. Task 4 preps for interactive selection by printing the melodies in a format fitted for EasyABC, so the user can copy/paste and listen on there. Task 5 and 6 implement and optimize interactive selection, where the user acts as the fitness method by ranking a subset of individuals with separate ranks for melody/alternate melody pairs and bassline. Task 7 implements crossover based on the two rankings, hence “Double-Crossover.” Tasks 8, 10, 11, 12, and 13 follow the RGB project implementation. Task 9 implements a method for evolving melodies faster without having to ask the user every generation. Task 14 expands the knowledge base of keys and scales. Task 15 drafts an analysis of the program by altering variables of the genetic algorithm and testing user-friendliness.

- **Task 0** - Project Description and Motivation
- **Task 1 -** Initial Constraint System and Melody String Generation
- **Task 2** - Music Individual Class
- **Task 3 -** Mutation
- **Task 4** - Display Melodies Method for EasyABC
- **Task 5** - Interactive Selection/User-Centric Fitness
- **Task 6 -** Analyze User-Friendliness Selection
- **Task 7** - “Double-Crossover” (based on best rankings for melody pairs and bassline)
- **Task 8 -** Population Class
- **Task 9 -** User/Random Selection Based on Generation Number
- **Task 10** - Population-based Mutation Methods
- **Task 11** - Copy
- **Task 12** - Population-based Crossover Methods
- **Task 13** - Finish GA Implementation
- **Task 14** - Expand Knowledge Base of Keys and Scales
- **Task 15** - Analysis of Music Samples and Collaboration/User-Friendliness

Second Plan - February 14, 2023

- **Task 0** - Project Description and Motivation
- **Task 1 -** Initial Constraint System and Melody String Generation
- **Task 2** - Music Individual Class
- **Task 3 -** Mutation
- **Task 4** - Display Melodies Method for EasyABC
- **Task 5** - Interactive Selection/User-Centric Fitness
- **Task 6 -** Analyze User-Friendliness Selection & Add to Constraint System
- **Task 7** - "Double-Crossover" (based on best rankings for melody pairs and bassline)
- **Task 8 -** Population Class
- **Task 9 -** User/Random Selection Based on Generation Number
- **Task 10** - Population-based Mutation Methods
- **Task 11** - Copy
- **Task 12** - Population-based Crossover Methods
- **Task 13** - Finish GA Implementation
- **Task 14** - Expand Knowledge Base of Keys and Scales
- **Task 15** - Analysis of Music Samples and Collaboration/User-Friendliness

Third Plan – March 1st, 2023

I moved the implementation of the Population Class from Task 8 to Task 5 since it is needed – maybe not required, but makes it easier – to test selection. I moved the User-Friendliness analysis to Task 8, so I could implement Double-Crossover first. I also moved the User/Random Selection Based on Generation Number after the population methods for testing purposes.

- **Task 0** - Project Description and Motivation
- **Task 1 -** Initial Constraint System and Melody String Generation
- **Task 2** - Music Individual Class
- **Task 3 -** Mutation
- **Task 4** - Display Melodies Method for EasyABC
- **Task 5 -** Population Class
- **Task 6** - Interactive Selection/User-Centric Fitness
- **Task 7** - Analyze User-Friendliness Selection & Add to Constraint System
- **Task 8 -** "Double-Crossover" (based on best rankings for melody pairs and bassline)
- **Task 9** - Population-based Mutation Methods
- **Task 10** - Copy
- **Task 11** - Population-based Crossover Methods
- **Task 12 -** User/Random Selection Based on Generation Number
- **Task 13** - Finish GA Implementation
- **Task 14** - Expand Knowledge Base of Keys and Scales
- **Task 15** - Analysis of Music Samples and Collaboration/User-Friendliness

Fourth Plan – March 12th, 2023

After performing the user-friendliness analysis, I felt that a separate task was needed to address concerns I had while testing interactive selection. Thus, I added Task 8 as a separate task to handle these concerns.

- **Task 0** - Project Description and Motivation
- **Task 1 -** Initial Constraint System and Melody String Generation
- **Task 2** - Music Individual Class
- **Task 3 -** Mutation
- **Task 4** - Display Melodies Method for EasyABC
- **Task 5 -** Population Class
- **Task 6** - Interactive Selection/User-Centric Fitness
- **Task 7** - Analyze User-Friendliness Selection
- **Task 8** - Switch to Two Melodies & Add to Constraint System
- **Task 9 -** "Double-Crossover" (based on best rankings for melody pairs and bassline)
- **Task 10** - Population-based Mutation Methods
- **Task 11** - Copy
- **Task 12** - Population-based Crossover Methods
- **Task 13 -** Finish GA Implementation
- **Task 14** - User/Random Selection Based on Generation Number
- **Task 15** - Expand Knowledge Base of Keys and Scales
- **Task 16** - Analysis of Music Samples and Collaboration/User-Friendliness

Fifth Plan – April 6th, 2023

The fine-tuning of parameters deserves its own task, so it is now Task 15. The addition of new scales to the constraint system has been removed. I also added Task 16 based on feedback in class.

- **Task 0** - Project Description and Motivation
- **Task 1 -** Initial Constraint System and Melody String Generation
- **Task 2** - Music Individual Class
- **Task 3 -** Mutation
- **Task 4** - Display Melodies Method for EasyABC
- **Task 5 -** Population Class
- **Task 6** - Interactive Selection/User-Centric Fitness
- **Task 7** - Analyze User-Friendliness Selection
- **Task 8** - Switch to Two Melodies & Add to Constraint System
- **Task 9 -** "Double-Crossover" (based on best rankings for melody pairs and bassline)
- **Task 10** - Population-based Mutation Methods
- **Task 11** - Copy
- **Task 12** - Population-based Crossover Methods
- **Task 13 -** Finish GA Implementation
- **Task 14** - User/Random Selection Based on Generation Number
- **Task 15 -** Generate Collection of Compositions
- **Task 16** - Analysis of Music Samples and Collaboration/User-Friendliness