CS 150: Algorithmic Music Composition
Composition Lab 1

Richard Townsend, Tufts University

Background

In this lab, you will compose your first algorithmic work using the techniques covered so far: generative grammars and serialism. You will investigate these techniques at a deeper level, use them to design a compositional process, and implement your process with the music21 library. You are welcome and encouraged to work with a partner!

Content Requirements

You will submit a composition program called `comp1.py`, and a `written.pdf` write-up that explains your composition's motivation and algorithmic approach.

Write-Up

Your write-up should be short (about 1-2 pages) and include the following:

1. A "header section" specifying the name of your composition, the name of each composer, and an estimate of how many hours you spent working on this lab.
2. An explanation of how to run `comp1.py` (e.g., can it take command line arguments?) and what it should produce when run. Please have this in clearly labeled section that's distinct from the rest of the write-up.
3. A description of your composition: how does it use grammars and/or serialism to generate music? Are you implementing one of the approaches covered in class, or a new approach?
4. A visualization of a small, representative chunk of your composition (sheet music, numbers, drawings). This should be paired with prose explaining how the chunk demonstrates your algorithmic approach (look at the slides for examples).
5. A reflection on the algorithmic and/or aesthetic quality of the piece and how it could be improved or expanded.
6. Anything that gave you trouble while working on this lab (you may omit this if the answer would be "Nothing.").

Program

Your `comp1.py` program must use the music21 library to generate a 30-60 second piece, playing the piece by default but providing the user with the option to generate sheet music. If any additional Python libraries are required to run your program, those must be specified in your write-up too (with instructions of how to install them). NOTE: Richard uses a Linux computer, so if you're hoping to go crazy with new libraries, make sure he can install them first!

Submission and Grading

This assignment will be submitted in two parts:

- Write-up: upload to Gradescope as a `pdf` file.
- Code: submit on Halligan with command `provide comp150amc comp1lab comp1.py`

Your write-up and program will each be graded out of 10 points, with a maximum score of 20 points for the assignment. Each component can earn a maximum of 10 points, with points being earned according to these rubrics (going over 10 points just caps you at 10):

Component	Points	Description
Program	5	Generates a 30-60 second piece of music
Program	3	Uses either grammar- or serialism-based algorithms
Program	1	Uses both grammar- and serialism-based algorithms
Program	1	Uses a novel approach leveraging grammars or serialism
Program	0.5	Good style makes the code readable and easy to understand
Write-up	2	Includes a description of your algorithmic approach(es)
Write-up	2	Includes a visualization and accompanying explanation
Write-up	1	Includes a reflection
Write-up	1	Document is generally free of spelling and grammar errors
Write-up	2	The description is clear and easily understood
Write-up	2	The visualization captures an interesting aspect of your approach