Immersive Intro to Computer Science with Python + Applications to Music

Mark Goldstein

July and August 2020 at NEC

1 Practical stuff

- Mondays (lecture) and Wednesdays (interactive), both 3pm-4:30pm EST, July 6 to August 19 through TBA Zoom link.
- instructor info: Mark Goldstein, goldstein@nyu.edu (see point about Piazza, better than email)
- Will be recorded, **let me know ASAP if recording the session is *not OK***. You can choose whether to participate with audio/video, audio, or none, according to your preference.
- We will use Piazza as a discussion platform (signup link here),(regular link here). Easiest for me to keep track of all course-related questions there. You do need to make an account.
- I will keep course content (slides, code examples) posted at this github page (don't need to make account).

2 What is this?

This is a summer course on the fundamentals of computer science, with applications to music analysis, musicology, and composition.

3 What is Computer Science?

This says "computer science" or "computation" and not just "programming". Why?

3.1 Larger Picture

Our emphasis is principles of computation: learning to specify a problem for a computer and finding algorithmic solutions, i.e. a prescribed set of logical instructions that a computer (or person) could follow, for such problems. Some questions:

- Are there problems that don't admit algorithmic solutions? (turns out, yes, see mathematicians Gödel and Turing). If possible, which resources do we need to solve a problem? (theoretical CS research)
- Can computation enable novel kinds of collaboration across disciplines? Is it intuitive or frustrating to use? (human-computer interaction research)
- Can the computer learn its own behavior from a set of a given examples instead of being explicitly programmed? (machine learning / AI research).

3.2 More practically

We should ask if becoming friends with computation is useful to us. We can try to see it as a general means of solving problems in our everyday lives:

- automatically renaming all the `PHOTO_16798.IMG` filenames to something more useful, e.g. based on location of photo or even the computer's guess about the content of the photo.

ranging to specific problems in our fields of study:

- Noname's rapping style is right on the threshold of speaking and pitched singing. Which harmonies are in there?
- What is common to all of the music listened to by this Spotify user?
- I have an elaborate plan for how rhythm in my composition will evolve. Do I really have to write it out manually, or can I prototype it more quickly?

3.3 But what is programming?

"Programming" is the act of writing these intentions in a particular language (e.g. Python). This is useful, but not the whole picture. That said, we do need to learn *some* language, and Python is a broadly applicable, popular, and easy language to use.

4 Creative Inspiration

computer science is a kind of universal language. It facilitates collaborations among researchers and creatives across disciplines. Here is one example:

> A landscape architect is working with a composer on a commission for the city of Boston to build a music garden. The designer shares with the composer a computer script that renders a model of the garden that simulates how the sun lighting changes the garden throughout the day, and how the garden plants and colors evolve through the seasons. The composer can not only appreciate the garden plans as musical inspiration, but can directly compute aspects of spatial and temporal proportion in the garden and use these as parameters of a composition that likewise evolves at several scales- over the course of minutes for a garden tourist and over the course of weeks or months for the Boston regular.

5 Structure of Course

5.1 Part one

A compact introduction to CS. We will learn Python and basic aspects algorithmic problem solving. After this, a student should feel ok writing a script in Python that iterates through text files representing Bach MIDI scores and answering a question about whether their exists a particular sequence of intervals in the corpus.

5.2 Part two

We will focus on applications such as conducting novel music analyses using statistics and such as generating scores algorithmically. We will use a few tools developed recently in academia:

- Music21 [1] is a tool for computer-assisted Theory and Musicology, enabling Corpus-wide statistical analyses, developed at MIT.
- Abjad [2] helps composers generate scores according to abstract rules, allowing them to generate a template for a piece and quick change some of its inner content
- Librosa [3] allows helps build information retrieval systems that infer genre, chord transcription systems. apps like Shazam, automatic segmentation of musical form.

If time allows, we will fill the last part of the course with material meant to stimulate research and composition projects for the upcoming year at NEC.

[1] https://web.mit.edu/music21/, https://mta.mit.edu/person/michael-scott-cuthbert
[2] http://abjad.mbrsi.org/
[3] https://librosa.github.io/librosa/

6 Syllabus

Numbers correspond to weeks. Each week has a lecture on Monday and an interactive lab on Wednesday.

1. **What is Computer Science?** First, high-level on theory and practice of computation. Computer science is more than just programming?. Programming by speaking/writing in plain English. Python basics. Data types (integers, floats, strings, lists, lists of lists) and control flow logic (if-then conditionals statements, while loops). Importing a library. Numerical computing using Numpy.

2. **Abstraction.** Writing functions. Higher-order functions (functions that operate on functions). Recursion. Python Objects/Classes, Some text processing like computing the frequency of words in the text of a given author. Do good authors use more vowels?

3. **Probability basics** plus randomness in Python to generate rhythms (stochastic music exercise from Stratis' Xenakis class). more numerical computing using Numpy. Students should leaving know what a conditional probability distribution is.

4. **Symbolic Representations of Music.** Music 21 library for Corpus-wide queries. Students should prepare a novel query for the following week.

5. **Audio Representations of Music.** Librosa library for music information retrieval, computing the Short Time Fourier Transforms to analyze frequency content, and maybe segmenting compositions.

6. **Symbolic Again.** Abjad library for algorithmically generating/prototyping scores. Students should generate a score for the following week.

7. **Machine Learning and Music**. Discuss natural language modeling and its extensions to music. Discuss Google Magenta's recent work on music modeling and generation. How to predict the next chords. For the following week, students should bring discussion points for how they want to use computation/programming for the upcoming academic year at NEC and beyond.