

## Project #2 Programming

Due: Fri., Nov. 3, 11:59 pm<sup>1</sup>

CHOOSE ONE of the following four project options:

1. Create a randomly-generated drum beat
2. Create a randomly-generated 4-voice canon
3. Sonify genetic data
4. Other Max programming project (with instructor permission)

---

### INSTRUCTIONS

---

1. Starting with the template file: “Random\_Drum\_Beat\_Template.maxpat”

**Create a generative music app that generates an interesting drum beat** using randomly selected General MIDI (GM) percussion instruments. You must change *at least* the following in the template file: *tempo* (108), *cycle length* (5), percussion *instrument* selections (43 47 50 37 60 61), weighted-random *duration* selections (8 16 16 16), and *velocity humanize* factor (12%).<sup>2</sup>

2. Starting with the template file: “Random\_4-voice\_Canon\_Template.maxpat”<sup>3</sup>

**Create a generative music app that generates an interesting 4-voice canon** using randomly selected musical parameters and GM. You must change (at least) the following in the template file: *tempo*, **pipe** *delay times*, *scale*, *duration*, and *instrument* selections.

3. Starting with the template file: “DNA\_Sonification\_Template.maxpat”<sup>4</sup>

**Create a generative music app that generates interesting music from a DNA sequence.** You must change (at least) the following in the template file: *metro* rate, *DNA nucleotide base-to-musical-parameter mappings* (for pitch, velocity, and duration), *harmonization*, and *instrumentation*.

#### Other Requirements

- Extend the template file in some musically or algorithmically significant way.
- Use comments (c) to document your code.
- Design an attractive *user interface* for your app.<sup>5</sup>
- Colorize objects (Object > Color) to highlight the structure of your code or to create visual interest.
- Add colored **panel** objects and text that highlight important features of the user interface.

\* \* \*

---

<sup>1</sup> The *Programming Project Proposal* is due Fri., Oct. 27, at 11:59 pm.

<sup>2</sup> Program defaults are indicated within parentheses.

<sup>3</sup> This template utilizes the **ScaleMap2** subpatcher, which requires the text file “UserScales2.coll” be located in the same directory as the main patcher.

<sup>4</sup> Option 3 requires the DNA sequence FASTA file “insulin.txt” be located in the same directory as the main patcher. Other DNA sequence FASTA files may be downloaded at NCBI’s GenBank: <<https://www.ncbi.nlm.nih.gov/genbank/>>.

<sup>5</sup> For example, align all objects, use segmented patch cords for message flow clarity, add **slider** and/or **dial** objects, add colored **panel** objects, etc.

## Programming Guidelines & Suggestions

- Strive for well-organized, easy-to-read, clearly commented code.
- Horizontally/vertically *align* all objects (Arrange > Align).
- Where appropriate, use the “self-commenting code” style; e.g., use number boxes, menus, dials, etc. to show musical parameter values as they change.
- Use *segmented patch cords* (Cmd-y) as appropriate.
- Add slider/dial objects to graphically show data.
- Be sure to back up your work. It’s nice to have a recent working version of your patcher to return to if everything suddenly stops working. Edit > Undo (Cmd-z) may also be helpful.
- Be sure to use Max’s interactive *object Help* feature as you program.
- When you run your patcher, be sure to use Overdrive (Options > Overdrive) mode. This gives computational priority to sound generation tasks.

## Submission Guidelines

### Supporting Paper

Write an informal supporting paper (typed, 2-3 pages, double spaced, bullet-point style) that describes your artistic and technical goals. Be sure to discuss your selections of MIDI parameters (pitch, velocity, duration, timbre, etc.) in full detail. Also be sure to fully explain what you added to the template.

### File Submission

- Give your project a title – one that appropriately describes the music.
- Using a **comment** object, put the project’s title and your name at the top of the patcher.
- Upload the following files into the Programming Project Blackboard assignment by the deadline:
  - Your completed .maxpat file
  - Any required dependent files; e.g., “UserScales2.coll” (text file), a DNA sequence file (FASTA format), etc.
  - Supporting paper (.docx)

## Grading

Supporting Paper – 50%

- Followed directions
- Clarity of artistic/technical descriptions
- Effort
- Code – 25%
  - Followed directions
  - Code is well documented and objects are well organized
  - Patch cords and **send/receive** objects are easy to follow
  - Program runs, syntax/flow is correct, and code does what the supporting paper says
- Creativity – 25%
  - The “quality” of the overall sound
  - The “look” of the patcher (e.g., alignment of objects, use of panel, color, fonts, etc.)
  - Innovative ideas/approaches

*We will discuss exemplary student projects in class.*