

Project #2 Programming

Due: Fri., Nov. 7, 11:59 pm¹

CHOOSE ONE of the following three *algorithmic composition* project options, or Option 4:

Option 1: Create a generative drum beat

Option 2: Create a generative *n*-voice canon

Option 3: Sonify genetic data

Option 4: Another type of Max project (with instructor permission)

For Options 1-3, template files are available for download at:

<https://max.reginaldbain.com>

GUIDELINES

1. Starting with the template file: “Random_Drum_Beat_Template.maxpat”
Create a Max app that generates an interesting cyclic drum beat using random numbers and General MIDI (GM) percussion instruments. You must significantly change at least the following musical parameters in the template file: tempo/pulse (108 1 16); cycle length (5 and 4, respectively); percussion instrument selections (43 47 50 37 60 61); weighted-random durations (8 16 16 16) and probabilities; and velocity humanize factor (12%).²
2. Starting with the template file: “Random_4-voice_Canon_Template.maxpat”³
Create a Max app that generates an interesting *n*-voice canon using randomly selected musical parameters and GM instruments. You must significantly change (at least) the following musical parameters the template file: tempo/pulse (120 1 16), pipe delay times (0 3000 5000 8000), scale (Pentatonic), velocity (64), pulse duration (1000), and GM instrument selections (12 13 14 16).
3. Starting with the template file: “DNA_Sonification_Template.maxpat”⁴
Create a Max app that generates interesting music from a DNA sequence. You must significantly change (at least) the following musical parameters in the template file: metro rate, DNA nucleotide base to musical parameter (pitch, velocity, and duration), harmonization factors, and GM instrument. Optionally, you might want to use a different DNA sequence file from NCBI’s GenBank repository.

Other requirements:

- Your music out should NOT sound like the original template file’s musical output.
- Try to extend the template file in some musically or algorithmically significant way.
- Use comments (c) to document your code.
- Design an attractive *user interface* (UI) for your app.⁵
- 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.

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

² The template program’s settings are indicated within parentheses.

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

⁴ 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/>>.

⁵ 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 Style & Methodology:

- Strive for well-organized, easy-to-read, clearly commented code.
- Horizontally/vertically *align* all objects (Arrange > Align).
- Use *segmented patch cords* (Cmd-y) as appropriate.
- Where appropriate, use the “self-commenting code” style; e.g., use number boxes, menus, dials, etc. to show musical parameter values as they change.
- 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 in this regard.
- Be sure to use Max’s interactive *object Help* feature as you code.
- Be sure to use Max’s Overdrive (Options > Overdrive) option to give computational priority to sound generation tasks.

Submission Guidelines

1. 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 musical parameters (pitch, velocity, duration, timbre, etc.) in full detail. Also be sure to fully explain what features you added to the template file.

2. Files

- **Give your project a title** – one that creatively describes your musical or algorithmic approach.
- Using a **comment** object, put the *project’s title* and *your name* at the top of the patcher.

Upload your supporting paper and files into the *Programming Project* Blackboard assignment by the deadline. Be sure to include:

- Your supporting paper .docx or .pdf
- Your completed .maxpat file patcher
- Any dependent files; e.g., the “UserScales2.coll” text file, the DNA sequence file (FASTA format), etc.⁶

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.

⁶ Optionally, you may upload a .zip archive that contains all of your files.