

Project #1 Programming w/ Max & MIDI

Part A *Preparatory Homework*

Due Date

Mon., Oct. 7, 2019

Directions

Project #1 has two parts: Part A & Part B. Part A asks you to write 10 short Max programs and store them in your personal folder on the Studio B hard drive. Part A is a preparatory assignment that reviews all of the Max and MIDI programming concepts we have covered in class. Be sure to note that hints are provided in the footnotes. Part B is a small-scale creative project that you will demo in class on the designated in-class presentation day.

Reading

It is assumed that you have read the Cycling '74 *Max: Getting Started* handout (see Learning Max) and MIDI reading assignments available on the class website (see Learning MIDI).

MIDI Device and Port Assignments

Use the following MIDI port assignments and variables for this assignment:

	Input Device	Output Device
Kurzweil PC88 (PC88)	a	a
Roland JV-2080 (JV-2080)	b	b

Unless otherwise specified, all MIDI output should be directed to the MIDI channel 1 of the Roland JV-2080. The JV-2080 should be set to General MIDI (GM) mode.

Exercises

Write the following 10 Max *patchers*. Create a “Max” folder in your personal MUSC 336 folder. Save each exercise in a separate patcher file named: *x.maxpat*, where *x* is the exercise number (e.g., 1.maxpat). Use Max’s **comment** object to *document* each significant component of your program.

1. Write a *program* that uses three integer number boxes to display the *MIDI pitch* (Display Format: MIDI C4),¹ *key velocity*, and *channel* of any note you play on the PC88 *keyboard controller*.
2. Write a program that displays *control change*, *after touch* and *pitch bend* information for the PC88.²
3. Write a program that *maps* all PC88 *MIDI data* to the JV-2080 *tone generator* so that you can play the JV-2080 via the PC88’s keyboard. (To be sure your patcher works, turn the PC88’s *master volume* all the way down and play something.)
4. Write a program that echos anything you play *n* milliseconds (ms) later. Use Max’s **pipe** object to implement the “echo”. Add a number box to the program that allows the user to change delay time (*n*). Finally, use a **loadbang** object and appropriate Max *message* to set the *default* delay time to 2 seconds when the program loads.
5. Use the **midiparse** object to construct a “raw” *Note on* MIDI message that plays the pitch A4 with key velocity 90.³ Also construct the corresponding *Note off* message. The help file demonstrates how to construct this type of low-level MIDI message. (Be sure to test that the Note on/Note off messages actually work.)

¹ Use the Inspector (Cmd-i) to accomplish this.

² HINT: Use the **midiparse** object, copy and paste program elements from the Max help file as required.

³ As you will see in the interactive Help file, the syntax for the Note on message is: 144, 67, 90.

6. Using two large *number boxes*,⁴ write a program that converts a *decimal number* to its equivalent *binary representation*.⁵ Set the range of the decimal number box to: Min. 0; Max. 127. Use comment boxes and color to help the user differentiate between the two numeric representations.
7. Write a *real-time interactive* program that takes data values generated by the PC88's *pitch bend* wheel (0-127), passes the values through a number box (an example of *self-commenting code*), displays them graphically using a **dial** object, and plays the corresponding MIDI pitches (0-127) on the JV-2080. Each note should have a default key velocity of 90 and duration of 50 ms.⁶ Finally, add the code necessary to allow the user to control the key velocity (0-127) via the *modulation wheel*.⁷
8. Write a *generative algorithm* that plays *pseudorandom* pitches between C4 and B4 inclusive at a rate of four-notes per second and displays them in real-time on a **kslider** (a piano keyboard "slider") and **nslider** (a music notation "slider"). Make the default key velocity 108. In order to produce *staccato* note, set the default duration 100 ms. Finally, add the code necessary to produce random key velocity values in the range 64–127 inclusive.⁸
9. Copy, paste and expand the previous program to play in *octaves*.⁹ Add a number box that will allow the user to change the rate of the **metro** object, constraining the the range of the number box to a minimum of 10 ms. and maximum of 2 seconds. Finally, make the duration of each note twice the metro rate so that note durations overlap (a *legato* effect).
10. Write a program that allows you to use the *expression pedal* to control the *volume* of a General MIDI (GM) organ voice.¹⁰ Use a vertical **slider** object to provide the user with visual confirmation of the current volume setting. Be sure that the range of the slider is 0-127. Use a **pgmout** object to send the necessary *program change* command to the JV-2080's GM instrument bank to select a GM organ voice when the program loads.¹¹

File Submission

Store all of your work in a sub-folder called "Max" in your personal MUSC 336 folder (as described above). Using a Max **comment** (c) object, **put your full name on the top-level of the Part B patcher**.

Programming Style

Be sure to use Max's interactive object Help feature as you work. Also be sure to:

- Horizontally/vertically align objects
- Use the "self-commenting code" style: e.g. use number boxes to show object output where appropriate
- Comment your code
- Use Max's Arrange menu and segmented patch cords where appropriate

Grading

Part A will be checked for completion but will not be assigned a grade. You are encouraged to bring any questions you to class for discussion.

⁴ Set the font size to 18-point, and be sure to fix the width of both boxes so you can see all of the numbers.

⁵ Use the Inspector (Cmd-i) to accomplish this.

⁶ Use a **makenote-noteout** object pair to play the notes and set these *arguments*.

⁷ The modulation wheel is *continuous controller* 1 (cc: 1).

⁸ The objects required are **metro**, **random**, **+**, **makenote** and **noteout**. Note: '**random** 64' will produce values between 0 and 63, inclusive. Adding (**+**) 64 to the results will produce values in the range 64-127.

⁹ Adding (**+**) 12 to a pitch number will produce an octave-related pitch.

¹⁰ The expression pedal is cc: 11. Volume is cc: 7.

¹¹ Use the **loadbang** object.