Making Music with Scratch

a workshop at presented at

SIGCSE 2011

March 12, 2011

Jesse M. Heines

Dept. of Computer Science
University of Massachusetts Lowell
heines@cs.uml.edu

John Maloney

Lifelong Kindergarten Group
MIT Media Laboratory
jmaloney@media.mit.edu

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu.
Performamatics is an NSF-funded interdisciplinary project at UMass Lowell. See http://performamatics.org.

Copyright Notice

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Contents

continued on next page

Contents (cont'd)

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Workshop Description

This workshop introduces the music playing and generation abilities of Scratch, a media-rich visual programming system (scratch.mit.edu). It is based on experiences gained using Scratch to teach both music and computer science in an interdisciplinary, college-level GenEd course. As students write programs that make music, they learn CS concepts such as control flow, user interaction, synchronization, real-time programming, and data structures.

Using their own laptops, workshop participants explore progressively complex Scratch programs that incorporate music in a variety of ways (see www.scratchmusic.org), including the use of external sensor devices to make custom musical instruments. The workshop culminates in a concert of participant-created music.

This handout provides the sample programs presented in the workshop and suggestions for extending them.

Workshop Agenda

1. Demonstration of Scratch music capabilities (10 min.)
2. Playing MP3 files from Scratch, demo and hands-on experience (20 min.)
   - Synching music to animations
   - Manipulation of MP3 files using Audacity
3. Playing MIDI notes from Scratch, demo and hands-on experience (30 min.)
   - Creating and playing simple melodies
   - Using loops and broadcasts to structure music
4. Playing MIDI notes using lists, demo and hands-on experience (30 min.)
   - Creating and populating lists
   - Working with rhythm and note lists
5. Synchronizing multiple parts, demo and hands-on experience (40 min.)
   - Techniques that do not work, and those that do
6. Introduction to external sensor devices (30 min.)
   - The Scratch Board and PicoBoard
   - The IchiBoard

continued on next page

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Workshop Agenda (cont'd)

7. Sharing what you've created, "performances" and discussion (20 min.)
    - Using the Scratch website
    - Concert of live performances by participants ☺

Additional Workshop Info and URLs

Participants should download and install:

- **Scratch**
  scratch.mit.edu/download
- **Audacity**
  audacity.sourceforge.net/download
  - download the **1.3 Series (Beta)**

Participants should also download and install the appropriate sensor board drivers for their systems.

- for **PicoBoards**
  www.picocricket.com/whichpicoboard.html
- for **IchiBoards**
  www.cs.uml.edu/ecg/index.php/IchiBoard

Please Note: Scratch does not have access to a MIDI synthesizer on systems running Linux, Ubuntu, etc. Scratch **does** synthesize notes on these systems, but you only get one instrument.

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

About the Workshop Leaders

Jesse Heines is a Professor of Computer Science at the University of Massachusetts Lowell. He has a keen interest in CS education and computer applications in the arts, particularly those in music. This interest was recently supported by NSF award 0722161, "Performamatics: Connecting Computer Science to the Performing, Fine, and Design Arts" (www.nsf.gov/awardsearch/showAward.do?AwardNumber=0722161 and www.performamatics.org). Jesse grew up in a musical household and currently enjoys singing in a barbershop chorus.

John Maloney is the lead programmer for Scratch, a visual programming system developed by the Lifelong Kindergarten group at the MIT Media Lab. Scratch is designed to help students of all ages learn programming and problem solving by creating personally meaningful artifacts such as games, animated stories, and interactive art. John is also an enthusiastic amateur musician.

Important Note on Turbo Speed

The timing of virtually all music scripts can be improved by setting Turbo Speed. To do this, select:

Edit ➔ Set Single Stepping... ➔ Turbo Speed

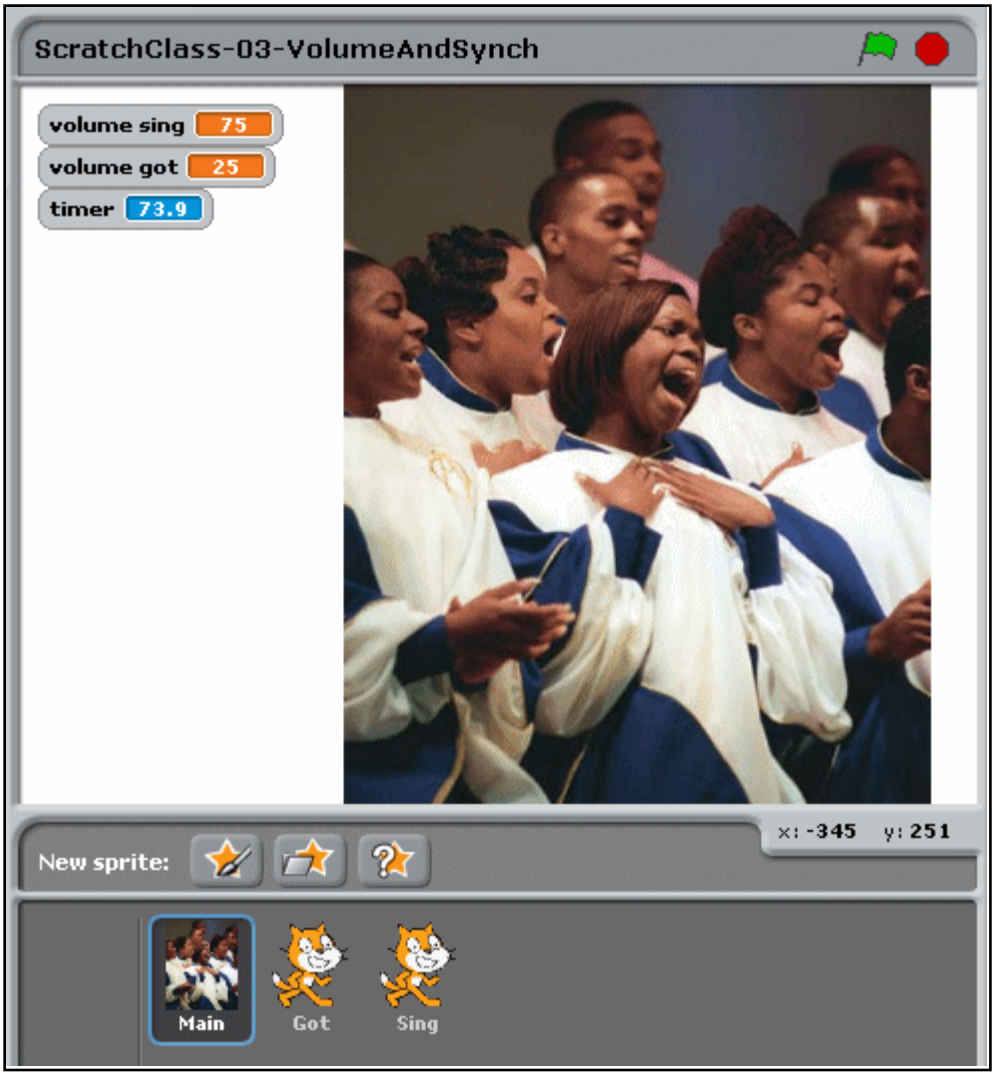
Acknowledgements

Additional contributors to this work include UMass Lowell Profs. Gena Greher, Alex Ruthmann, and Fred Martin, graduate student Mark Sherman, and recent undergraduates Paul Laidler and Charles Saulters.

The materials presented in this workshop is based in part upon work supported by the National Science Foundation under Grant No. 0722161, "CPATH CB: Performamatics: Connecting Computer Science to the Performing, Fine, and Design Arts" and complementary Research Experience for Undergraduates (REU) supplements. Any opinions, findings, and conclusions or recommendations expressed or implied in these materials or the workshop discussion are those of the authors alone and do not necessarily reflect the views of the National Science Foundation.

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Playing and Synchronizing MIDI Files



Volume and Synchronization Concepts

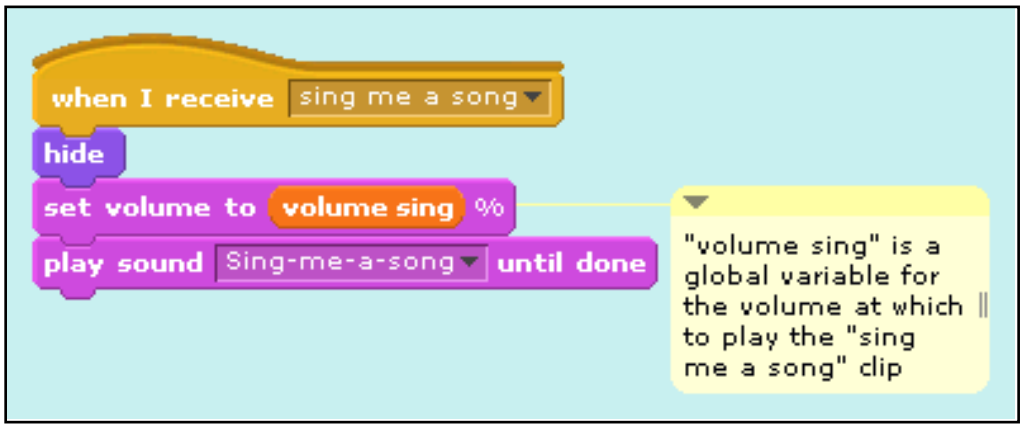
- use of variables when setting the volume
- local vs. global attributes, specifically volume
- use of a control script and broadcasts
- use of the Scratch timer for synchronization

Playing and Synching MIDI Files (cont'd)

MP3 Player Scripts

Script in Sprite "Got"

Script in Sprite "Sing"



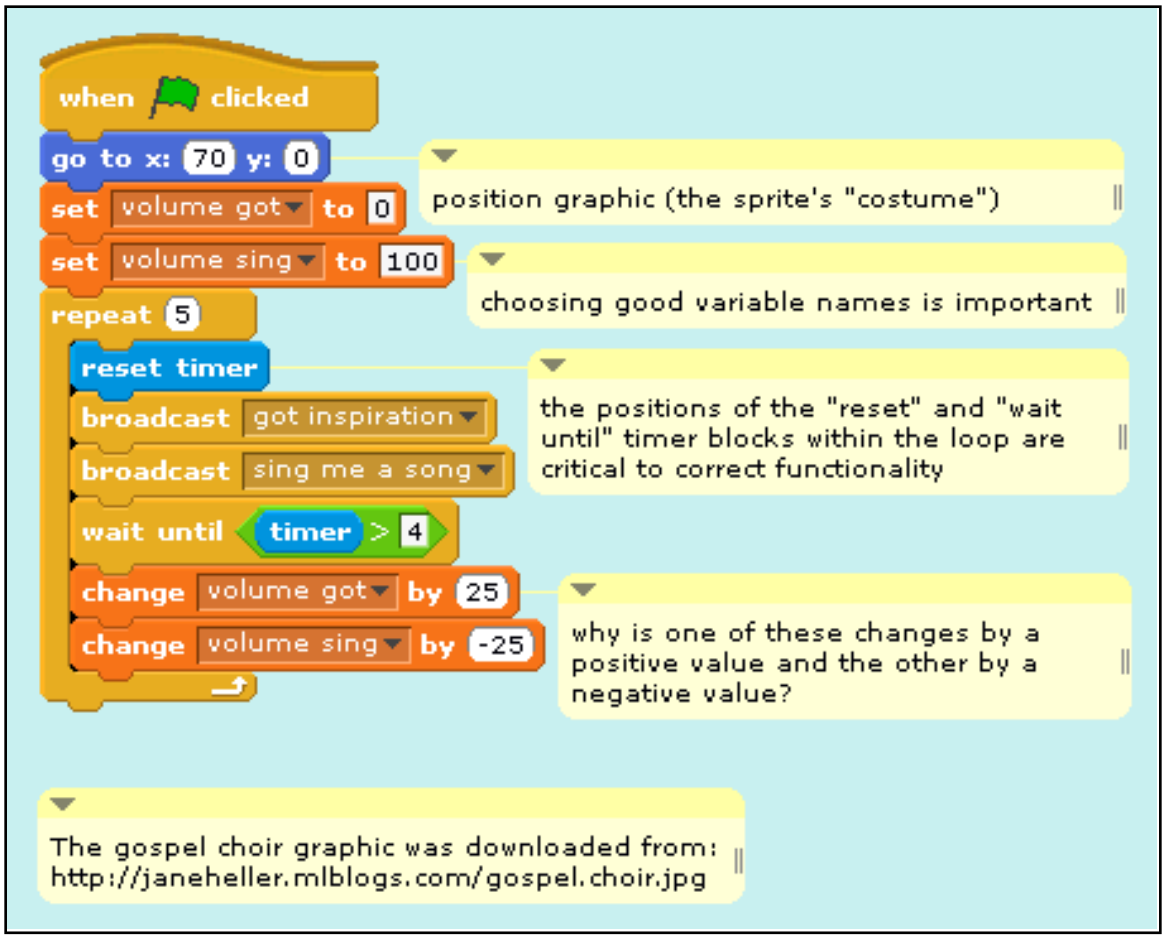
Each script must be in its own sprite to allow volume to be controlled independently.

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Playing and Synching MIDI Files (cont'd)

Control Script

Script in Sprite "Main"

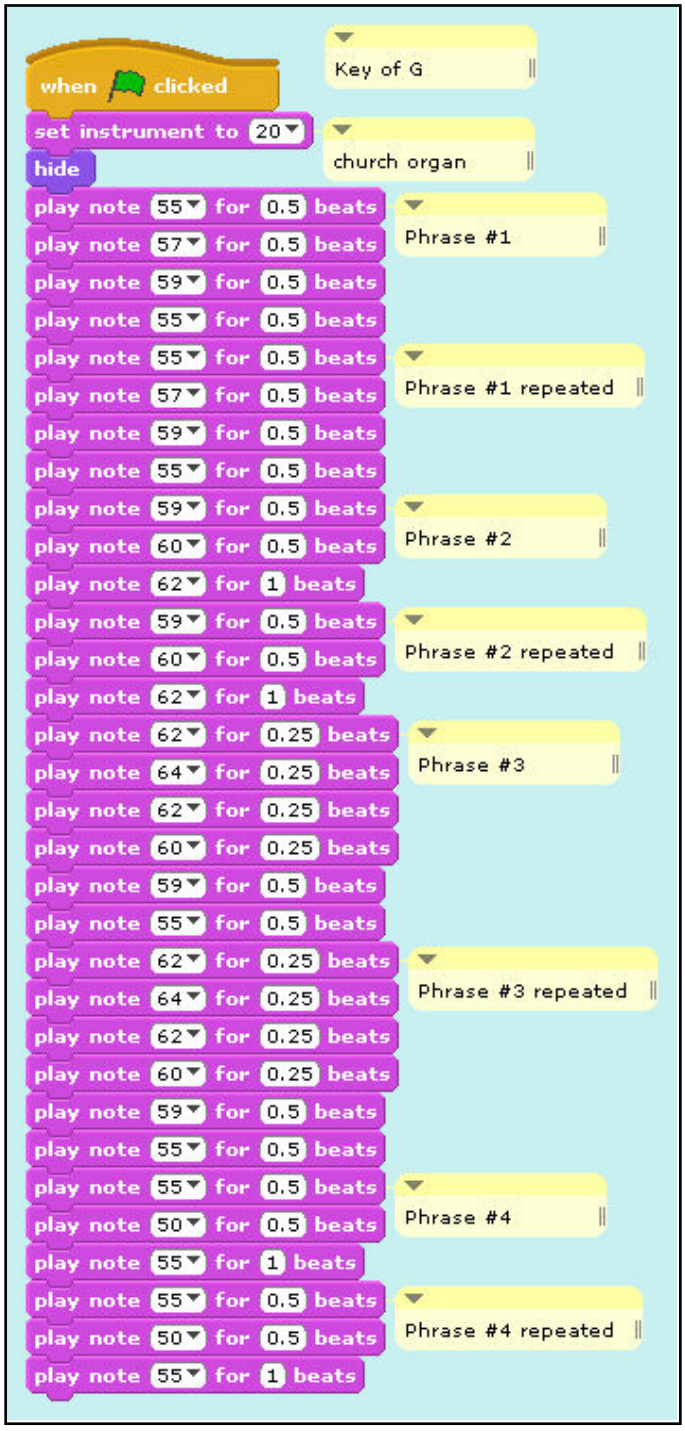


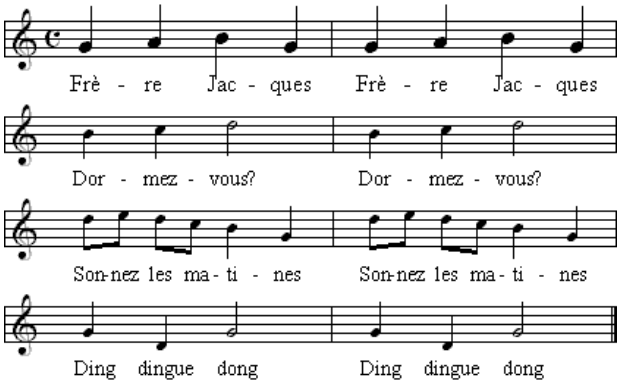
Note the order of the blocks and the critical position of the **change** blocks. Changing the volume parameter before the **wait until** block will cause the volume to be changed while the MP3 is playing. Such behavior may be desirable in other programs, but not this one.

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Frère Jacques

Version 1: Playing Notes

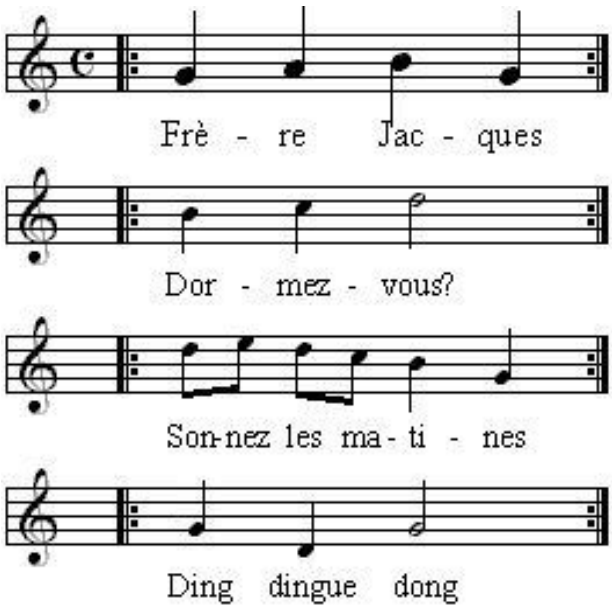




Remember Turbo Speed!

Frère Jacques

Version 2: Using Loops



Remember to set Turbo Speed to improve performance.

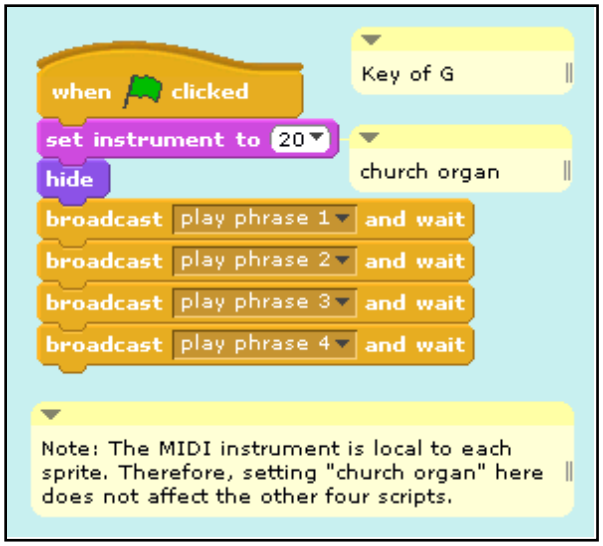
Acknowledgement: The scores on this and the previous page were adapted from www.csdraveurs.qc.ca/musique/flutalors/images/frere.gif and www.mamalisa.com/images/scores/frerejacques.jpg, respectively.

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Frère Jacques

Version 3: Separating Phrases

Main Script



Phrases Scripts (4, cont'd on next page)

#1

Thought: We could set the instrument in each script, but that would contradict the **DRY** programming principle: "Don't Repeat Yourself."

Frère Jacques

Version 3: Separating Phrases (cont'd)

Phrases Scripts (cont'd)

#2

#3

#4

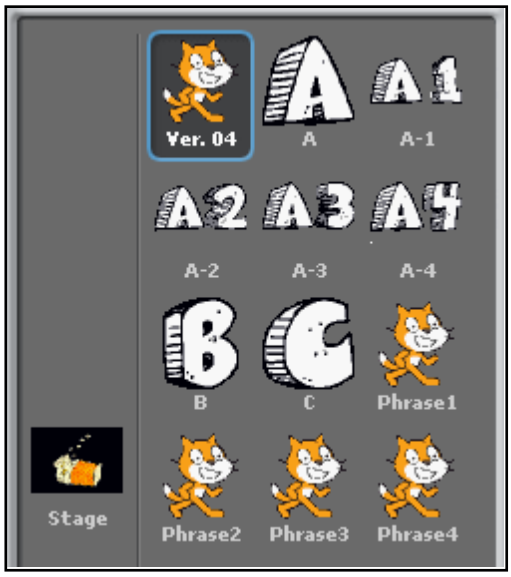
Challenge: How can we set the instrument **JUST ONCE** and have that setting apply to all scripts?

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

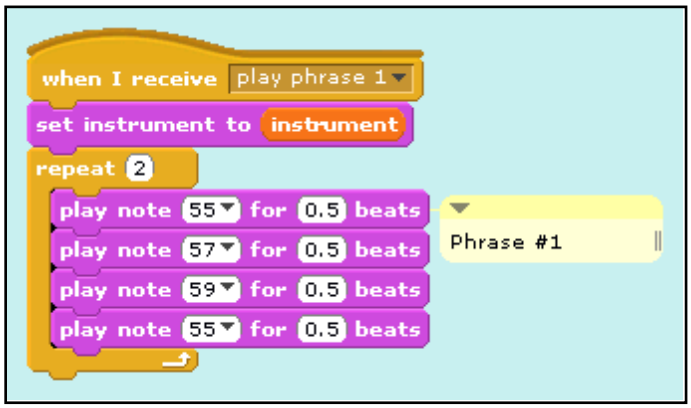
Frère Jacques

Version 4: Playing a Round

Main Script



Phrases Scripts



Note the addition of the **set instrument** block and the use of the **instrument** variable (set in the Main script) as the value to set. Other phrase scripts similarly contain this one revision.

continued on next page

Frère Jacques

Version 4: Playing a Round (cont'd)

Scripts A-1 through A-4

. . .

Others scripts are similar, differing only in **when I receive**.

Control Script A – single threaded

continued on next page

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Frère Jacques

Version 4: Playing a Round (cont'd)

Control Script B – multi-threaded

Control Script C – multi-threaded repeat

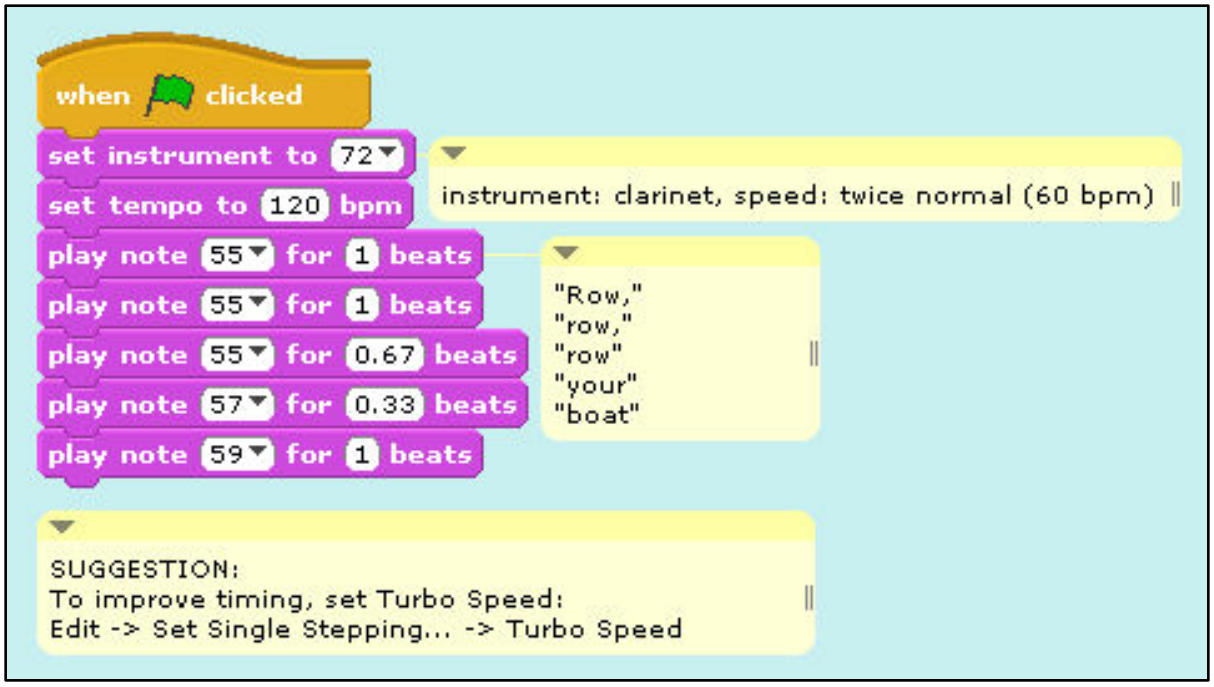
end of Version 4

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

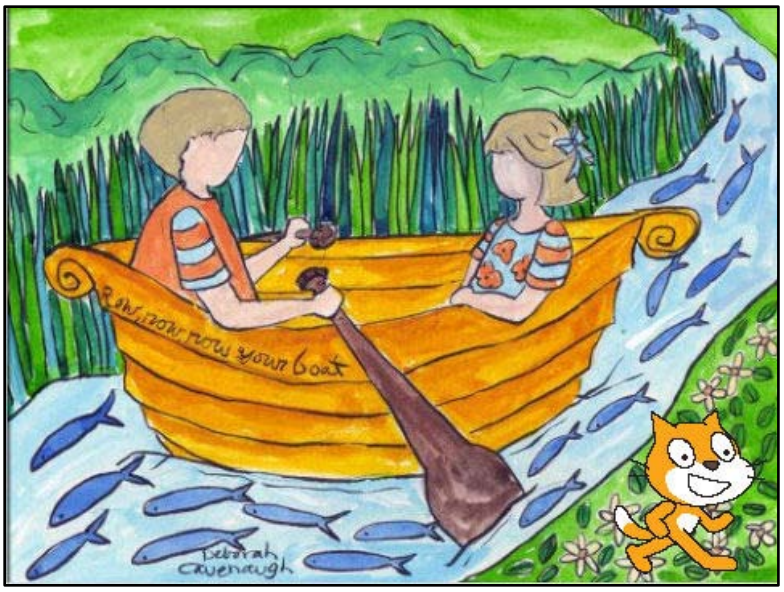
Row, Row, Row Your Boat

Version 1: Playing Notes

Single Script



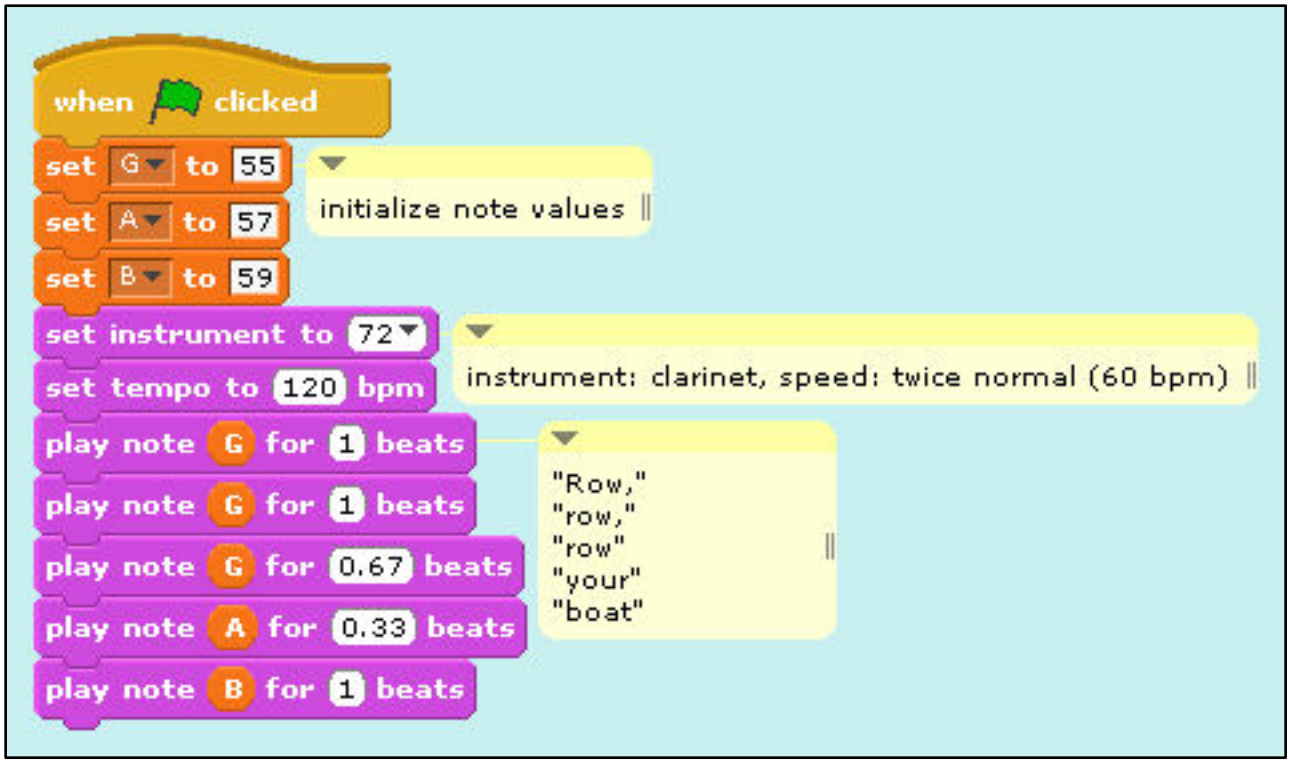
Output Window



Row, Row, Row Your Boat

Version 2: Playing Notes Using Variables

Single Script



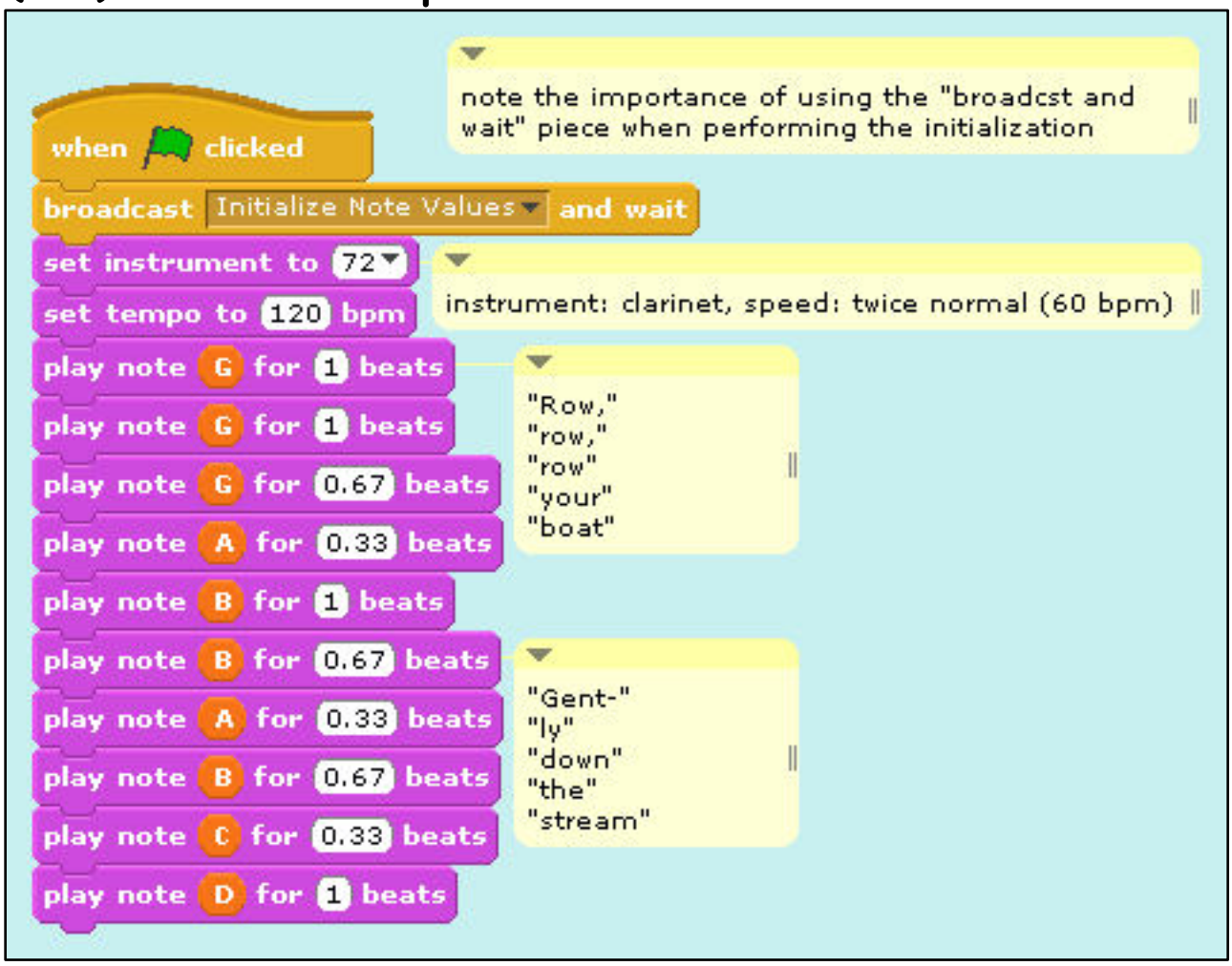
Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Row, Row, Row Your Boat

Version 3: Separating Initialization

Two Scripts

(3a) **Main** Script

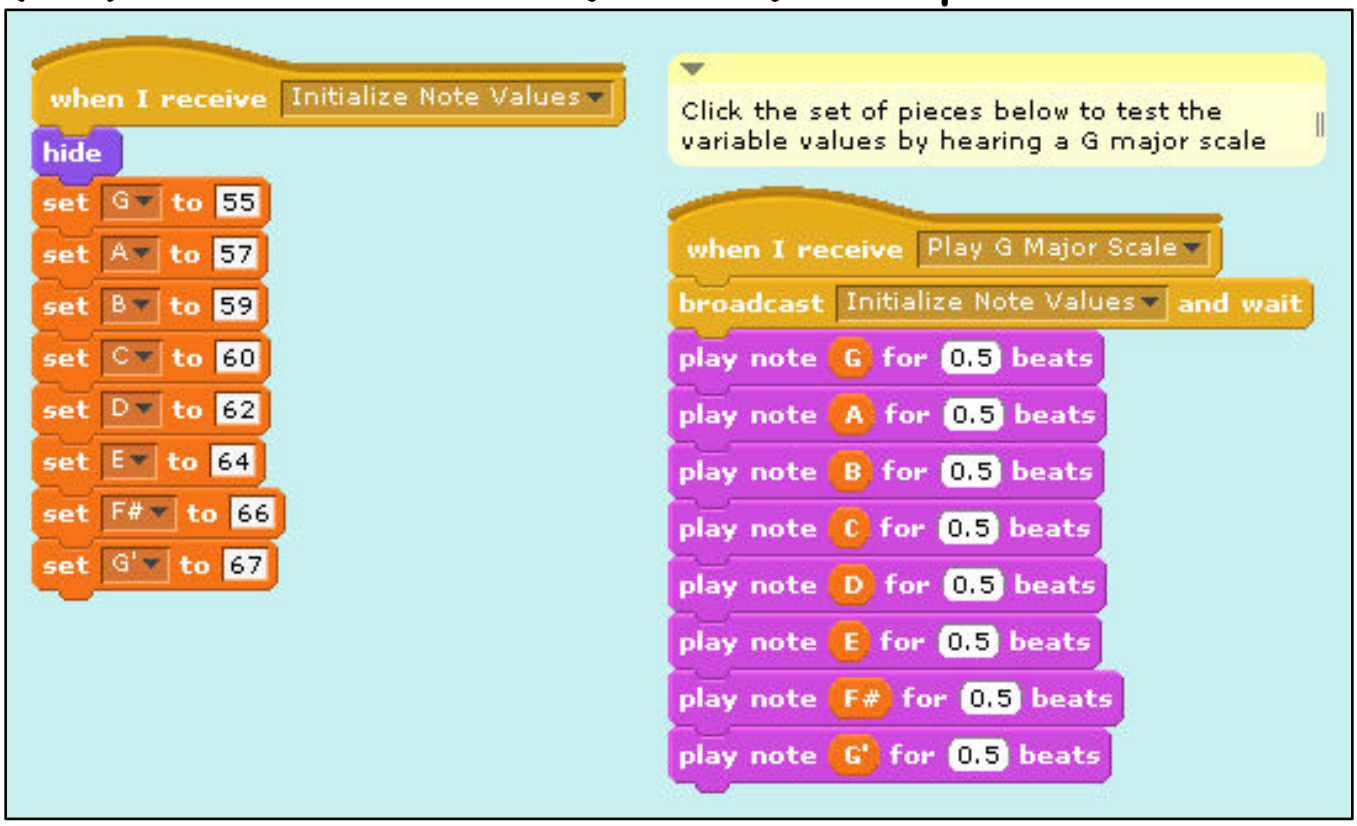


continued on next page

Row, Row, Row Your Boat

Version 3: Separating Initialization (cont'd)

(3b) **Initialization ("Init")** Script



end of Version 3

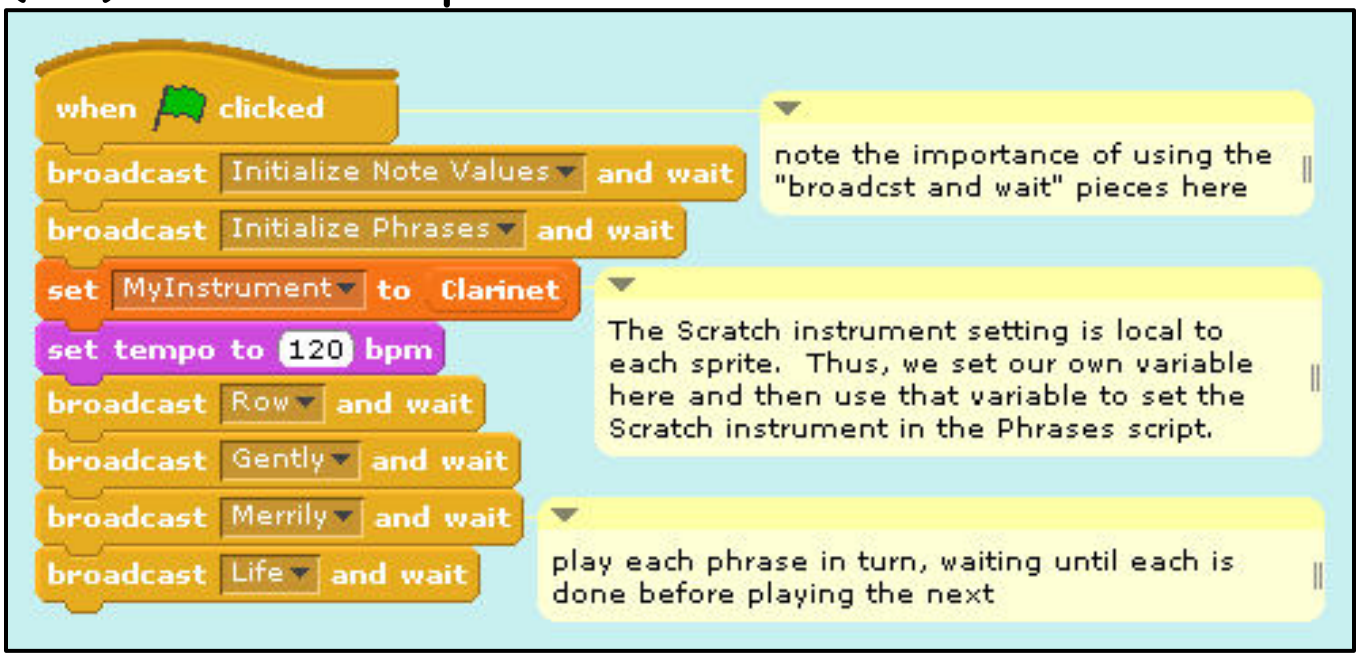
Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Row, Row, Row Your Boat

Version 4: Separating Phrases

Three Scripts

(4a) **Main** Script

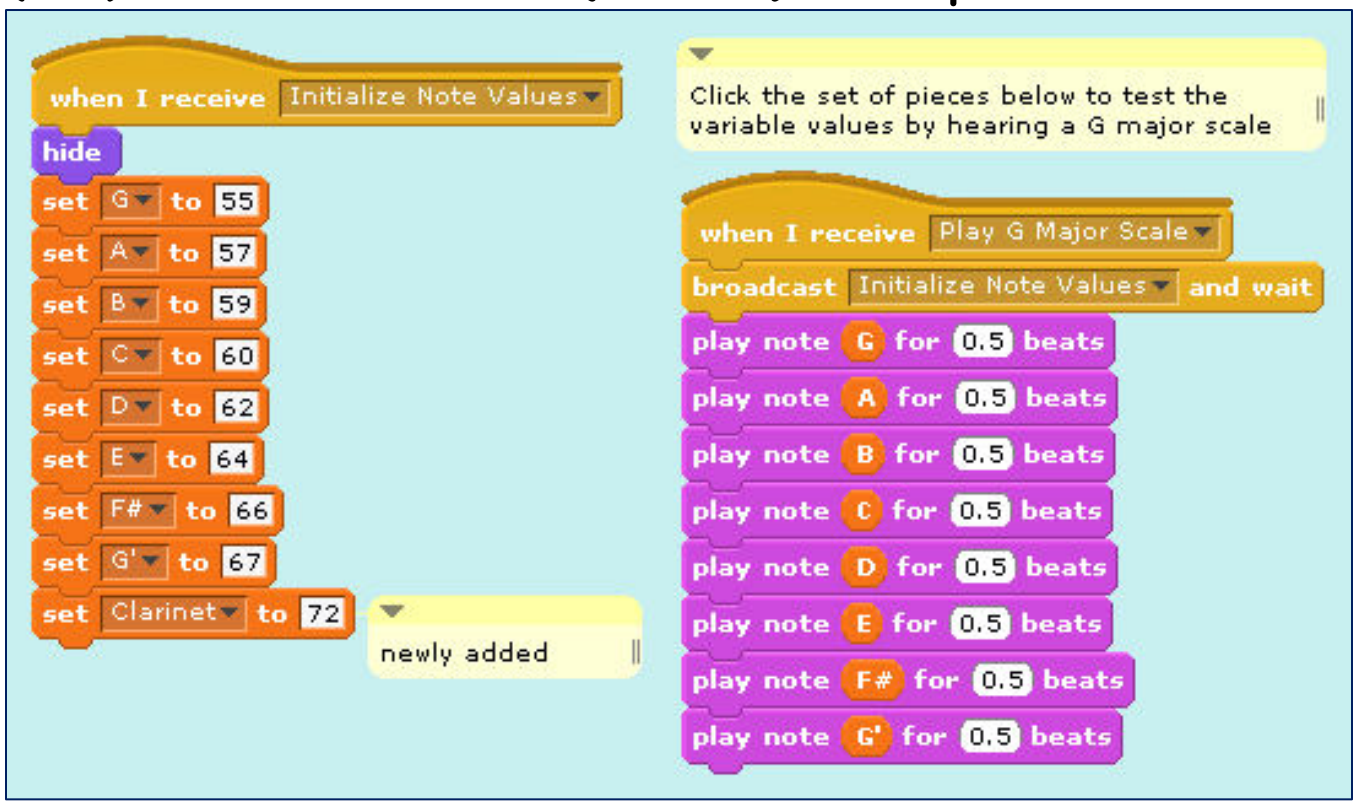


continued on next page

Row, Row, Row Your Boat

Version 4: Separating Phrases (cont'd)

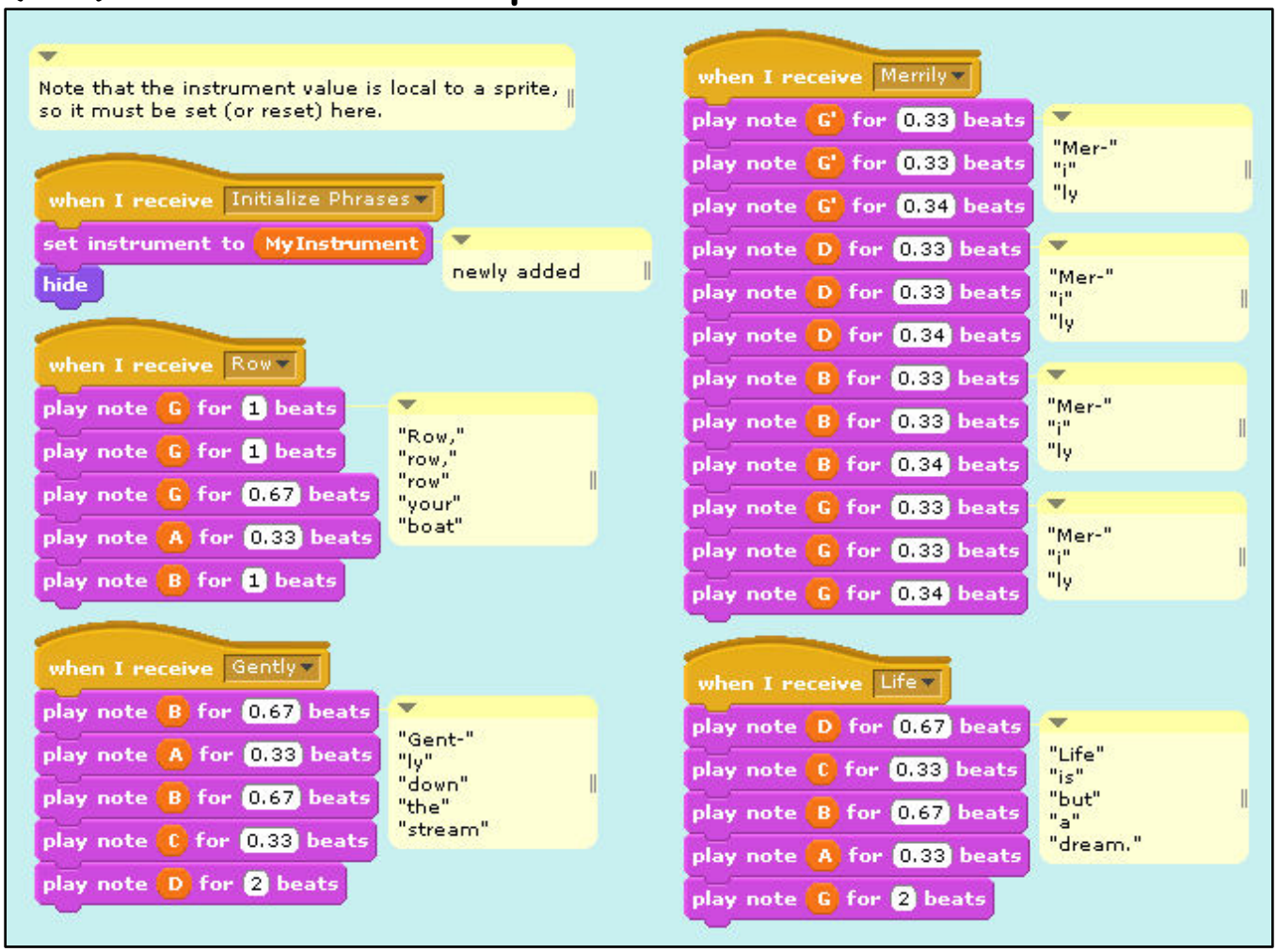
(4b) **Initialization ("Init")** Script



continued on next page

Version 4: Separating Phrases (cont'd)

(4c) **Phrases** Script



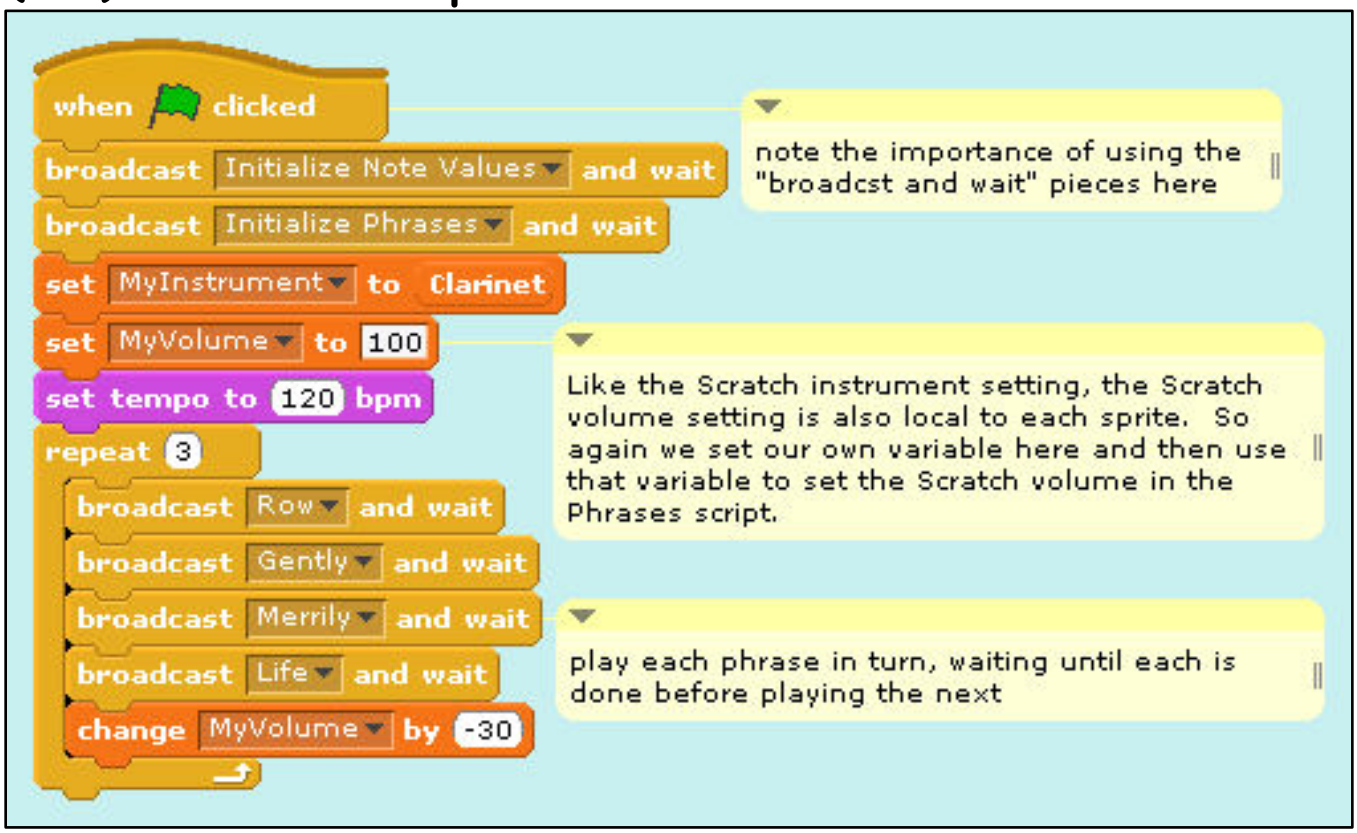
end of Version 4

Row, Row, Row Your Boat

Version 5: Looping and Fading

Three Scripts

(5a) **Main** Script



(5b) **Initialization ("Init")** Script

(*same as on page 28*)

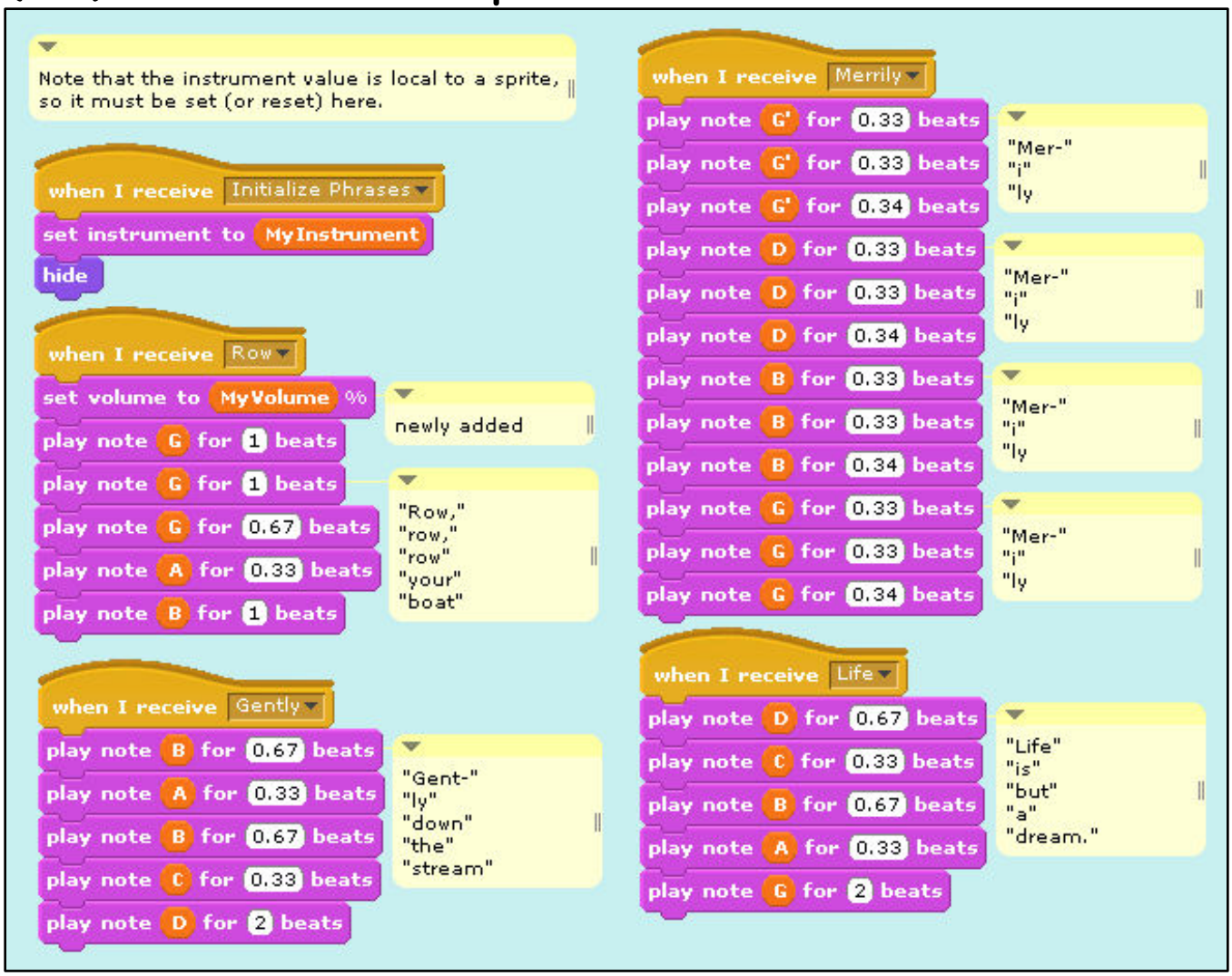
continued on next page

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Row, Row, Row Your Boat

Version 5: Looping and Fading (cont'd)

(5c) **Phrases** Script



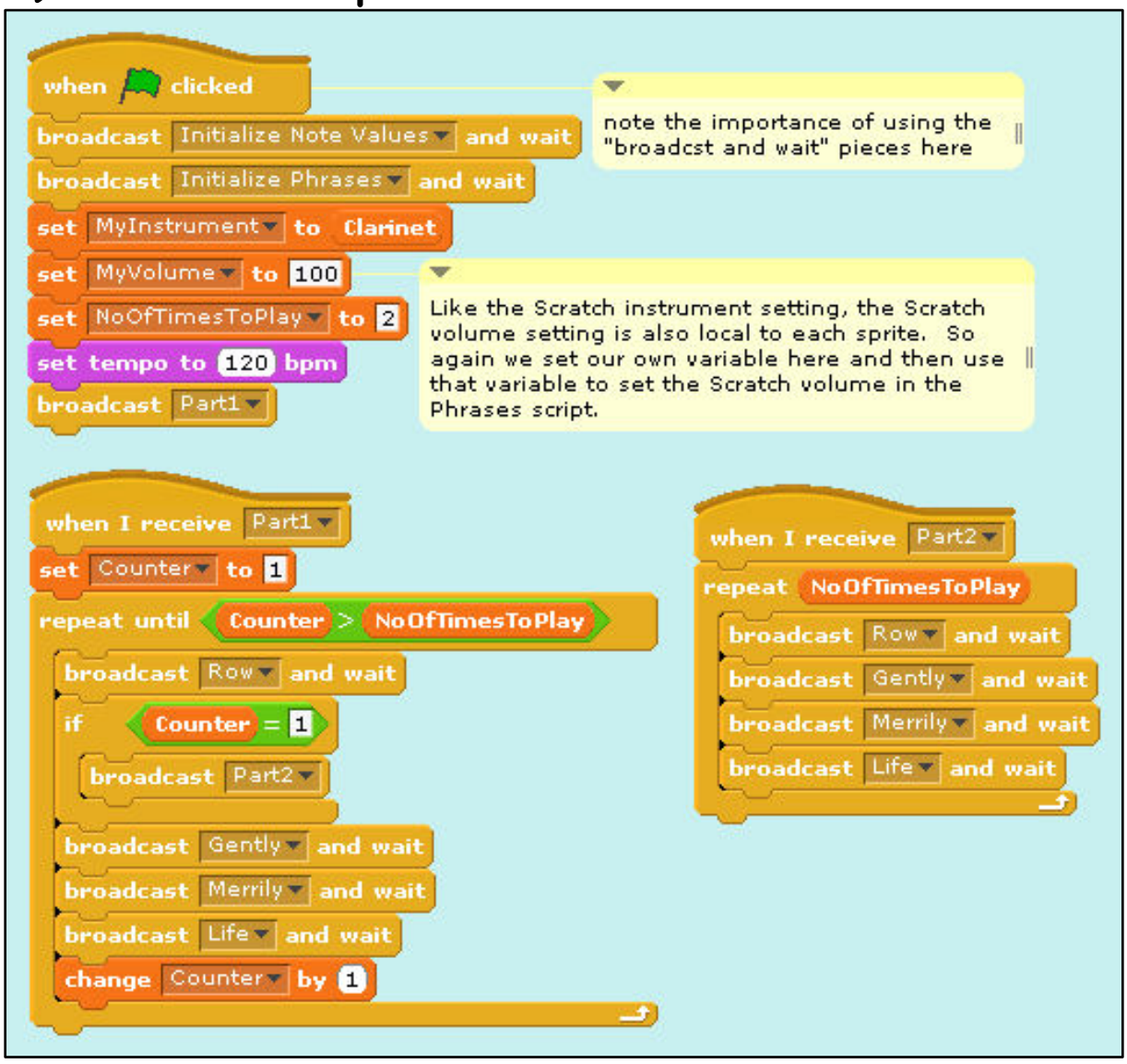
end of Version 5

Row, Row, Row Your Boat

Version 6: Playing a Round with One Instrument

Three Scripts

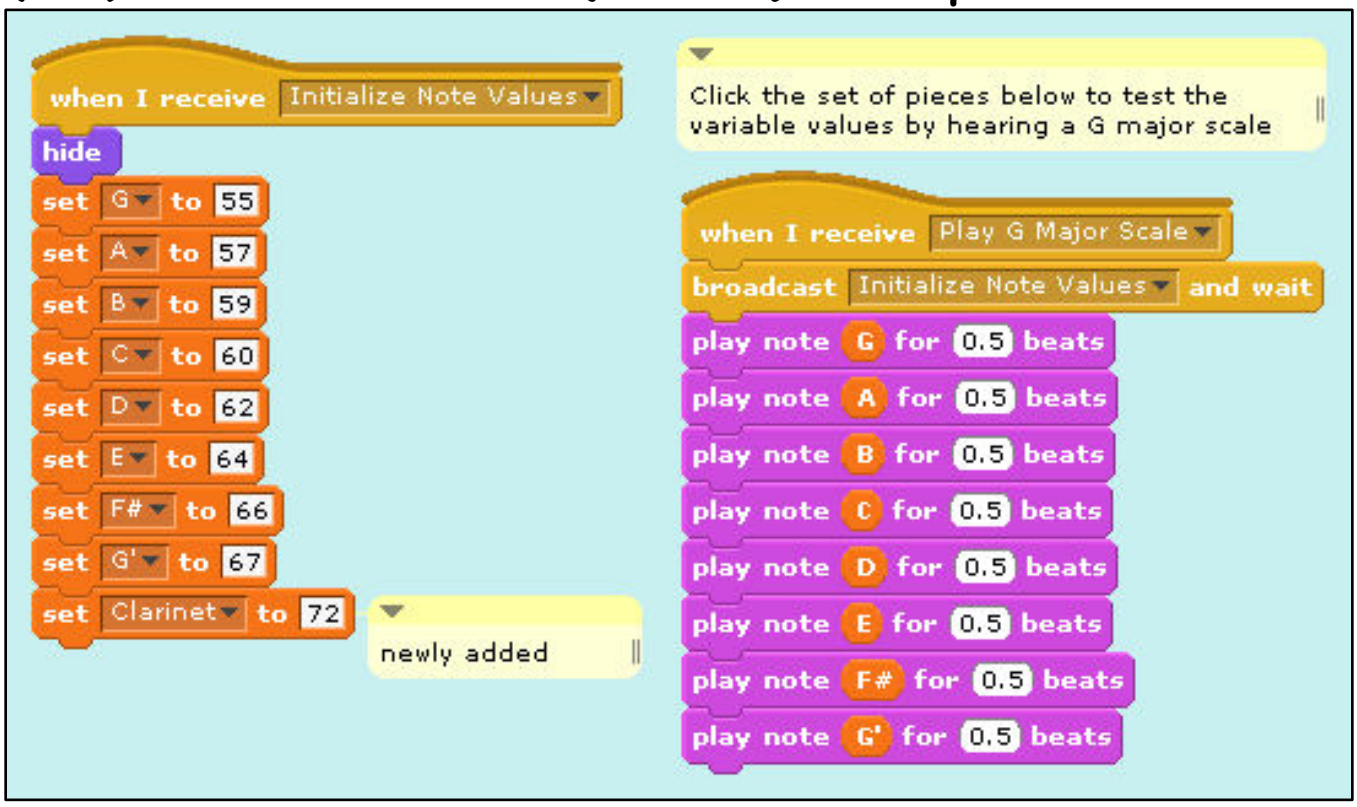
(6a) **Main** Script



Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Version 6: Playing a Round with One Instrument (cont'd)

(6b) **Initialization ("Init")** Script



(6c) **Phrases** Script

(*same as on page 31*)

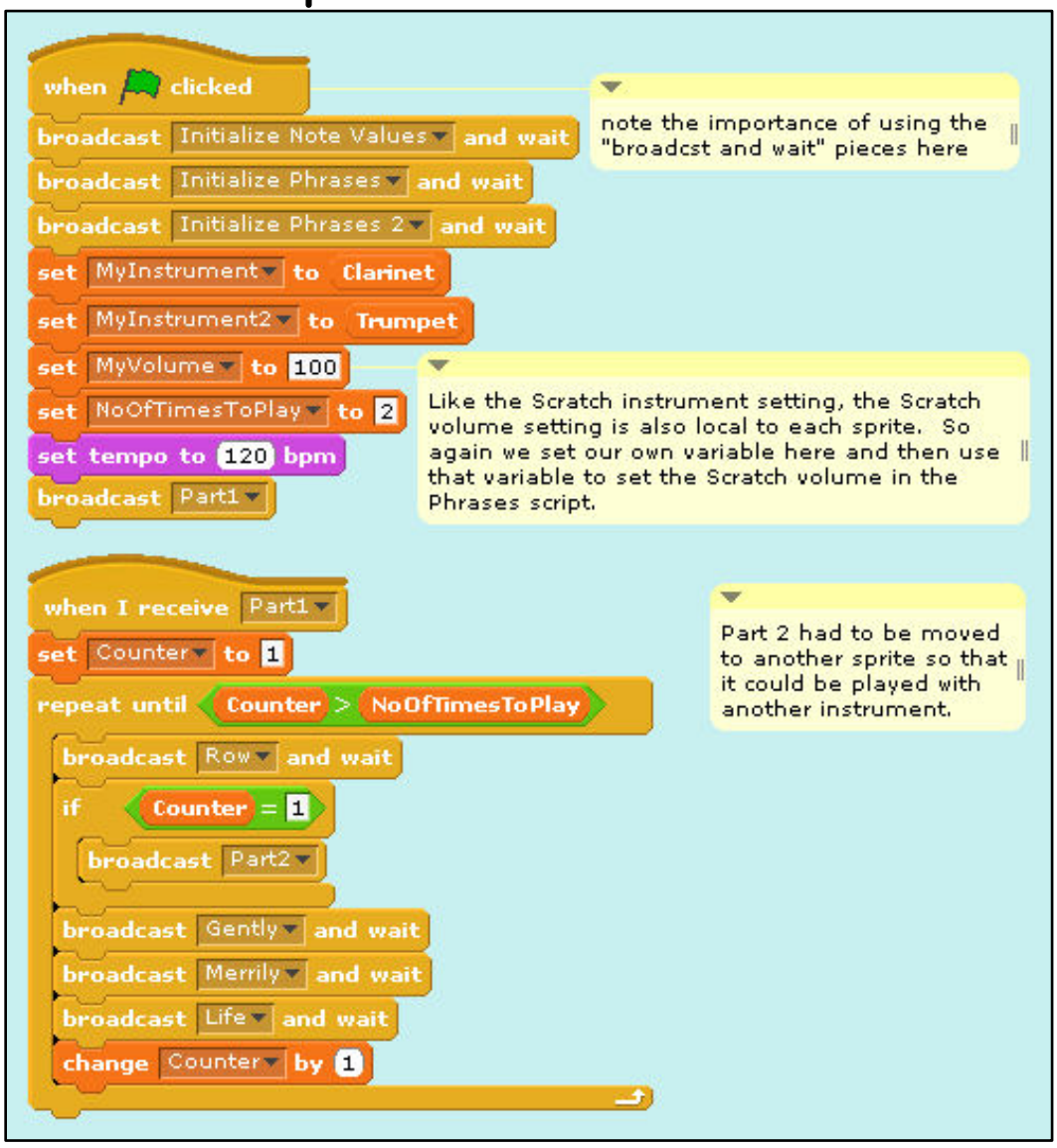
end of Version 6

Row, Row, Row Your Boat

Version 7: Playing a Round with Two Instruments

Five Scripts

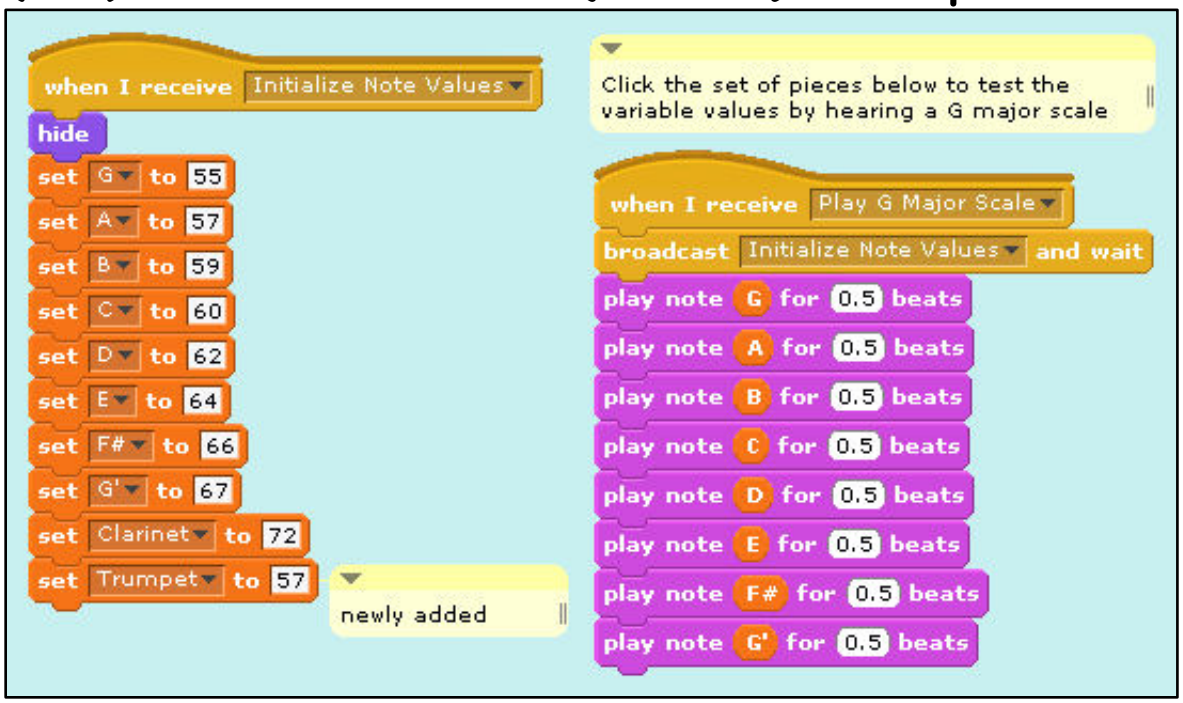
(7a) **Main** Script



Row, Row, Row Your Boat

Version 7: Playing a Round with Two Instruments (cont'd)

(7b) **Initialization ("Init")** Script



(7c) **Phrases** Script

(*same as on page 31*)

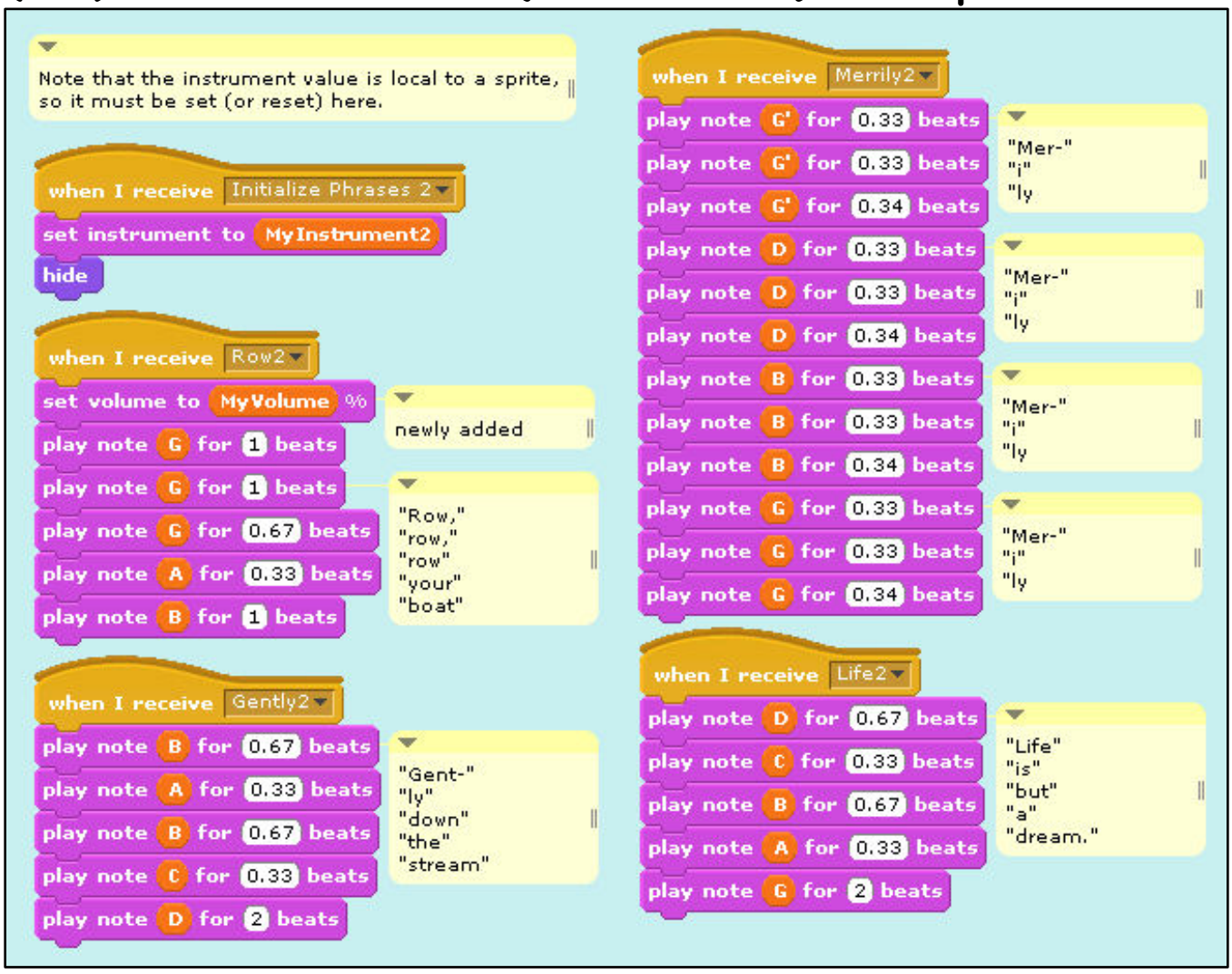
(7d) **Part2** Script ➔

continued on next page

Row, Row, Row Your Boat

Version 7: Playing a Round with Two Instruments (cont'd)

(7e) Instrument2 ("Instru2") Script



end of Version 7

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Row, Row, Row Your Boat

Version 8: Storing Notes and Rhythms in Lists

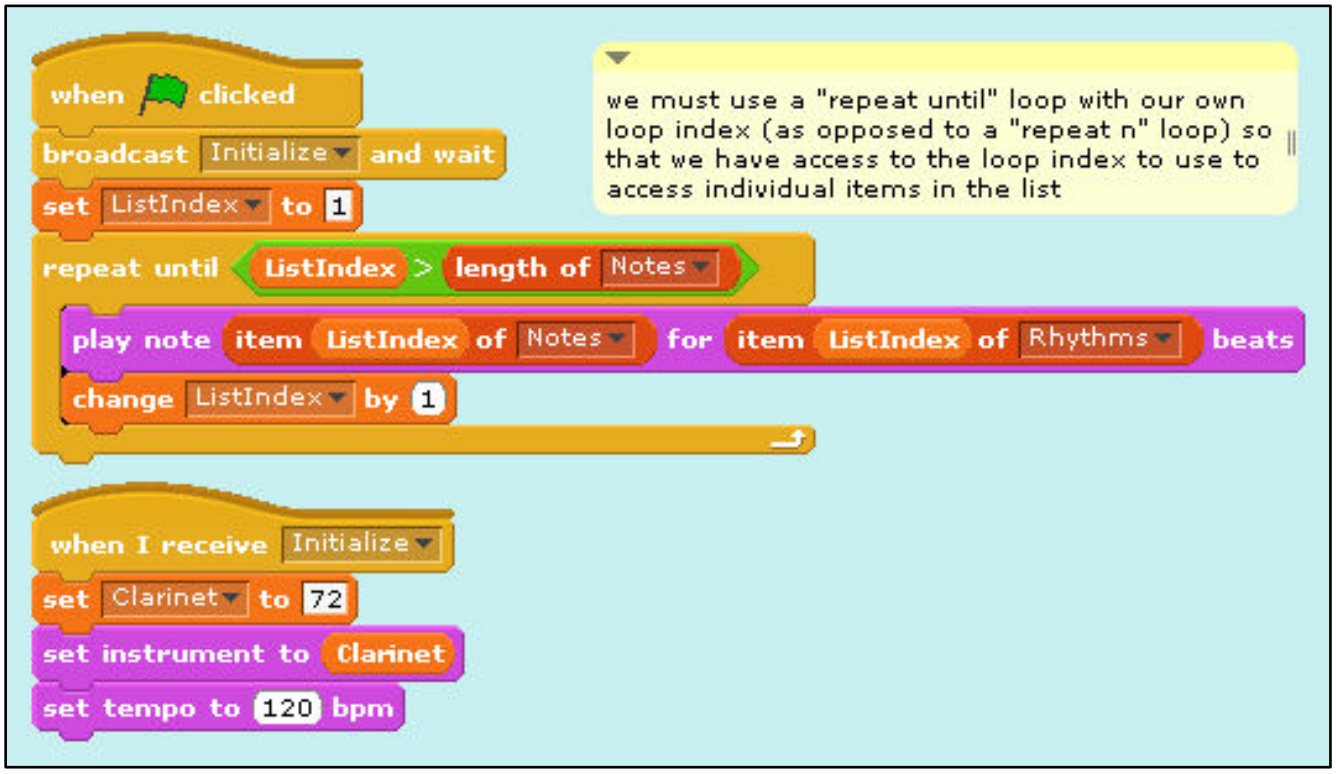
Output Window

continued on next page

Row, Row, Row Your Boat

Version 8: Storing Notes and Rhythms in Lists (cont'd)

Single Script



end of Version 8

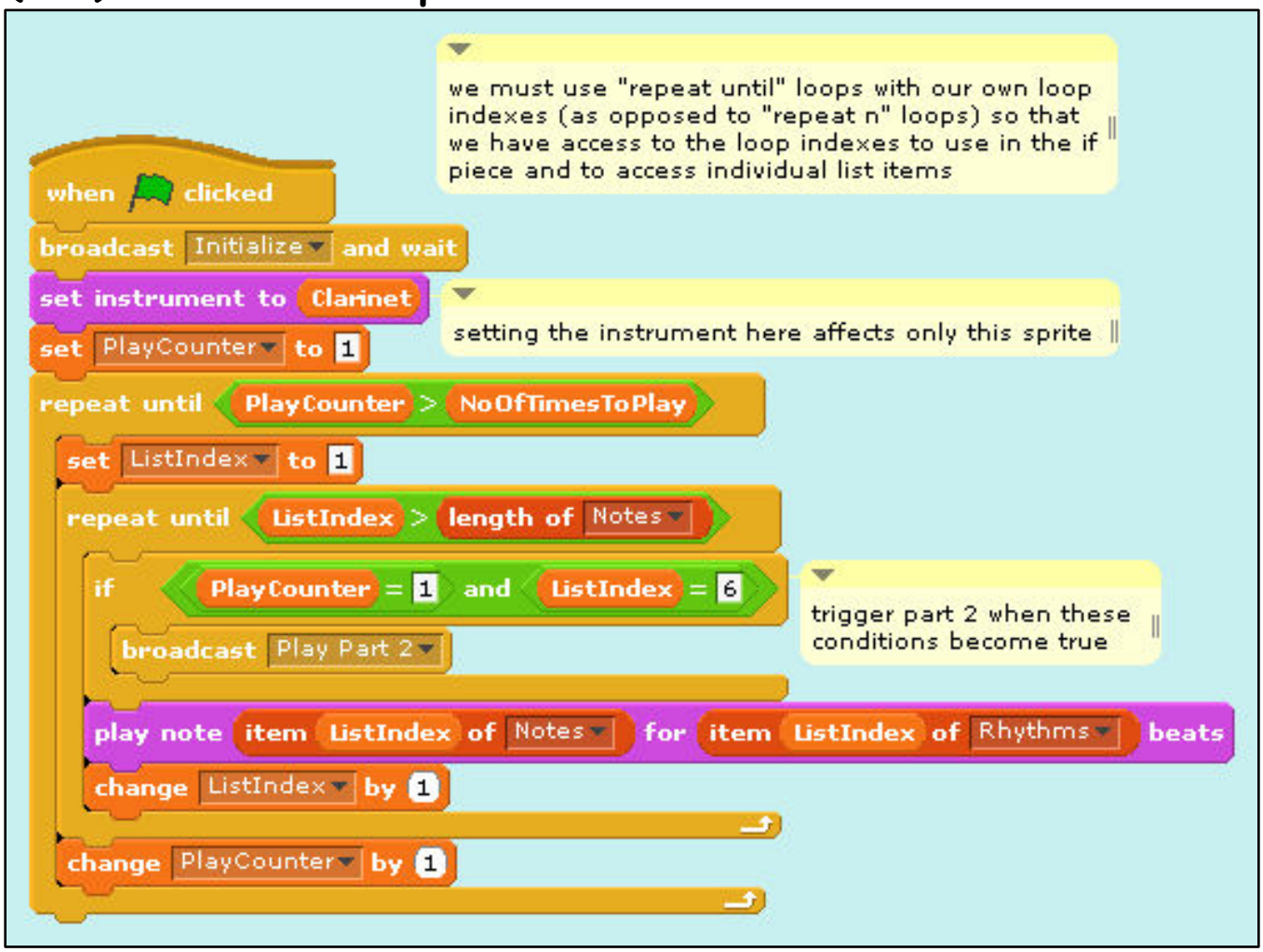
Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Row, Row, Row Your Boat

Version 9: Playing a Round Using Lists

Three Scripts

(9a) **Main** Script



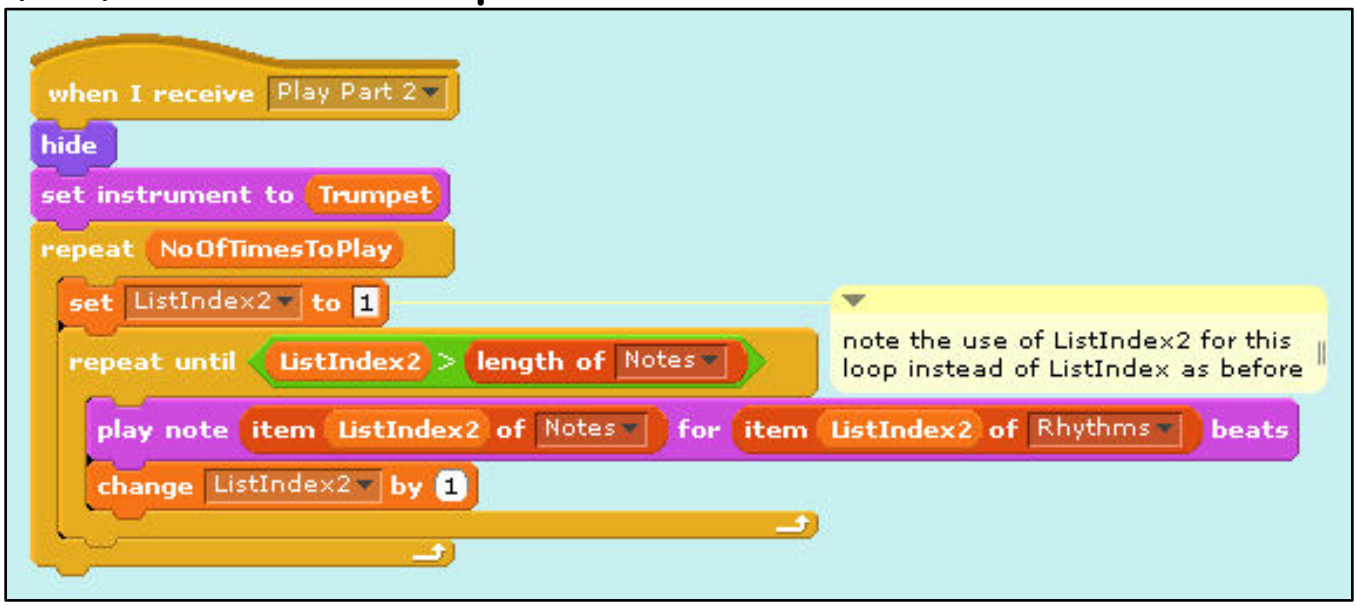
continued on next page

Row, Row, Row Your Boat

Version 9: Playing a Round Using Lists (cont'd)

(9b) **Initialization ("Init")** Script

(9c) **Part2** Script



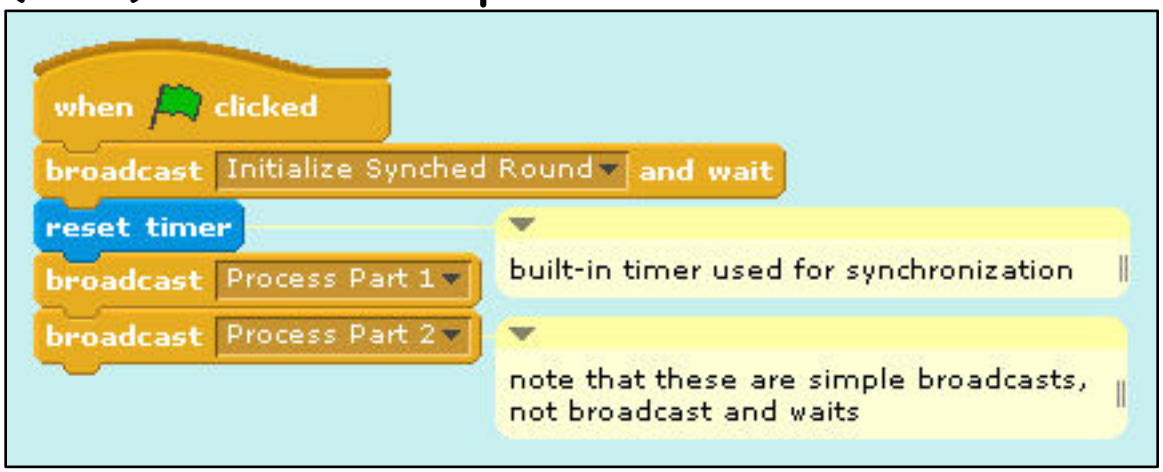
end of Version 9

Row, Row, Row Your Boat

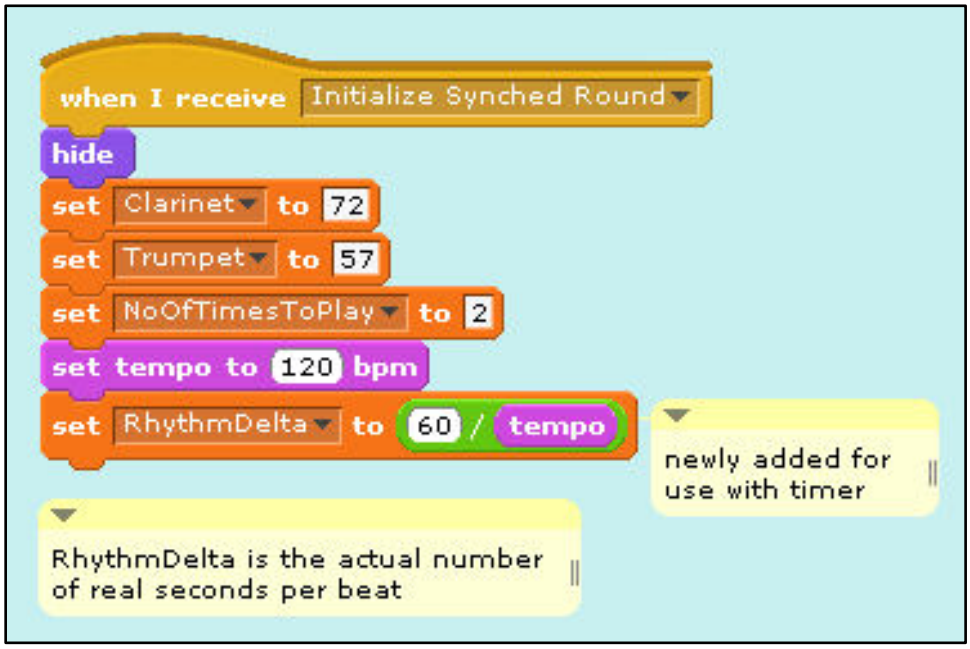
Version 10: Synchronizing Play from Lists

Four Scripts

(10a) **Main** Script



(10b) **Initialization ("Init")** Script

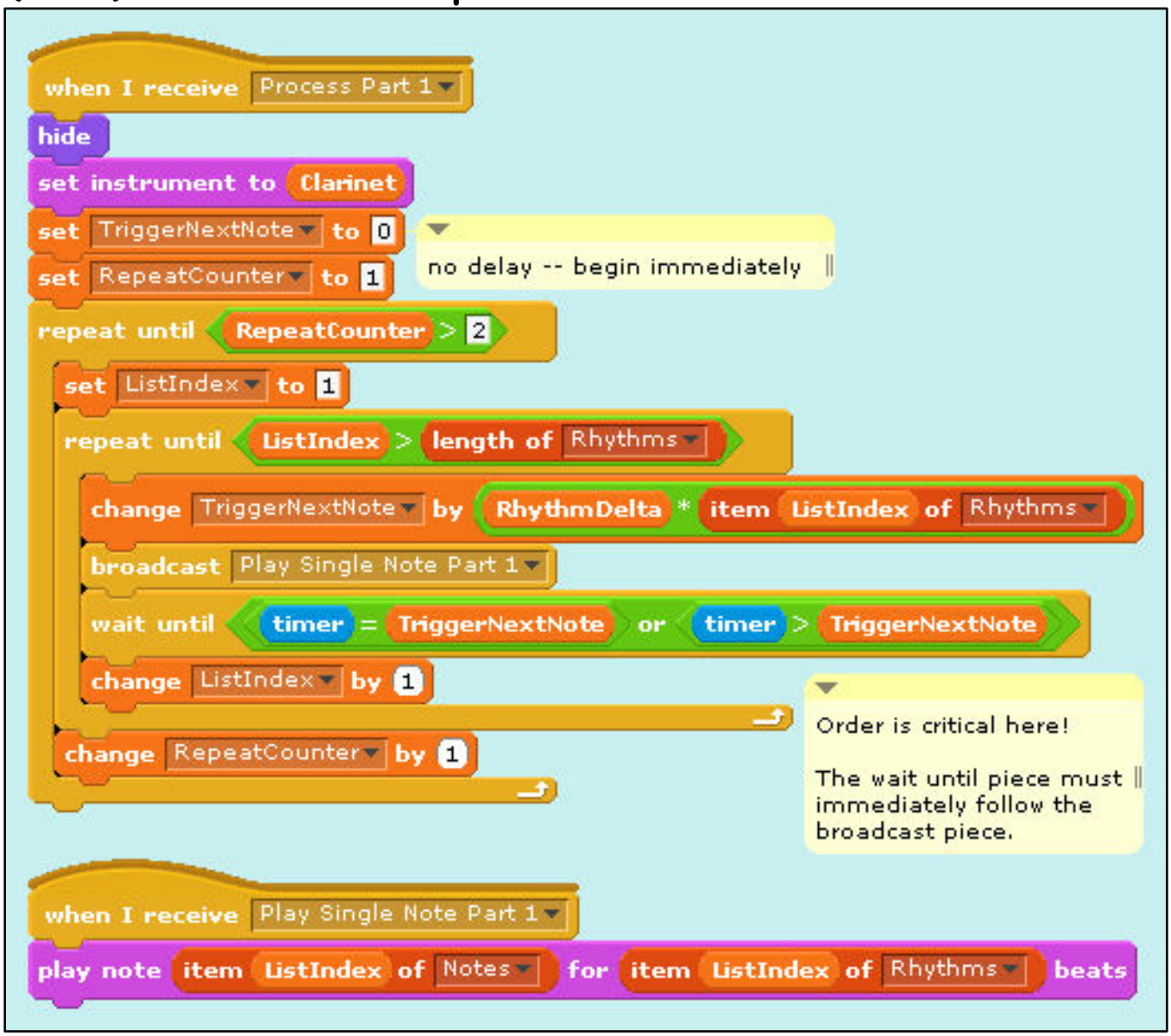


continued on next page

Row, Row, Row Your Boat

Version 10: Synchronizing Play from Lists (cont'd)

(10c) Part 1 Script



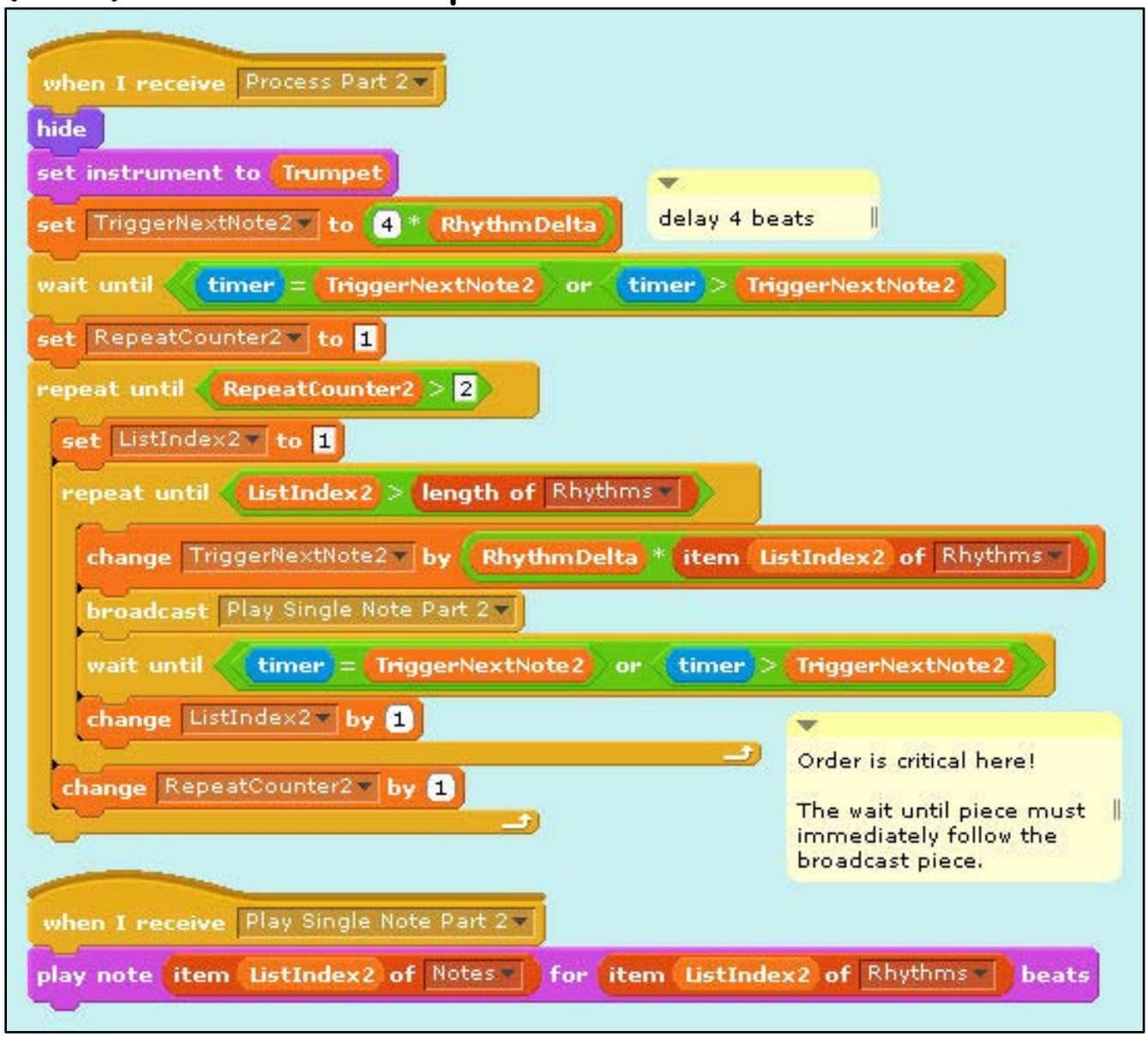
continued on next page

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Row, Row, Row Your Boat

Version 10: Synchronizing Play from Lists (cont'd)

(10d) **Part 2** Script



end of Version 10

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Extending the Examples

1. **Use a variable to set the tempo.**
   - Add a slider to the variable so that you can change the tempo in real time.
   - Find all the places you need to use the variable to reset the tempo when you change it in real time.
   - Which version of playing the round best stays synchronized when you change the tempo?

2. **Transpose the melody to another key.**
   - Create a variable to hold a pitch offset.
   - Find all the places you need to use that variable to play the melody in the new key.

3. **Increase the number of times that the round repeats.**
   - Do the parts stay in synch?

4. **Increase the number of parts that play simultaneously.** (Be sure to set Turbo Speed to do this!)
   - When should each part "come in"?
   - How much should the first beat of each part be offset?

Extending the Examples (cont'd)

5. **Play the melody backwards.**
   - Can you play multiple parts backwards, too?

6. **Increase the number of times that the round repeats.**
   - Do the parts stay in synch?

7. **Increase the number of parts that play simultaneously.** (Be sure to set Turbo Speed before you try this!)
   - When should each part "come in"?
   - How much should the first beat of each part be offset?

8. **Make a round using the G-major scale.**
   - Put the note values for a G-major scale into a list. See page 26 for code that initializes and plays a G-major scale, but remember that you must use the integer values, not the variable names, to play notes from a list.
   - Start Part 2 when Part 1 plays its third note (B, MIDI note #59).
   - Add Part 3, starting when Part 1 plays its fifth not (D, #62).

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Extending the Examples (cont'd)

9. **Play random notes in the G-major scale.**
   - Start with the list created for the previous exercise.
   - Use the "pick random" piece in the Operators group to pick a random note from the list.
   - Play each note for 0.25, 0.50, 0.75, or 1.00 beats, also selected randomly.
   - Does the result sound musical?

10. **Create a program that can play any <u>major</u> scale given any starting note.**
    - Store the starting note in a variable.
    - For a major scale, the number of half-tones between each note is:
      2, 2, 1, 2, 2, 2, 1
    - Another way to think about this is:
      Do + 2 ➔ Re + 2 ➔ Mi + 1 ➔ Fa + 2 ➔
      Sol + 2 ➔ La + 2 ➔ Ti + 1 ➔ Do
    - Create a list containing the changes between the notes, and then use a loop to process the list and play the scale.

Extending the Examples (cont'd)

11. **Create a program that can play any <u>harmonic minor</u> scale given any starting note.**
    - For a harmonic minor scale, the number of half-tones between each note is:
      2, 1, 2, 2, 1, 3, 1
    - Create a new list containing these changes, but use the same loop that you created for the previous exercise to play this scale.

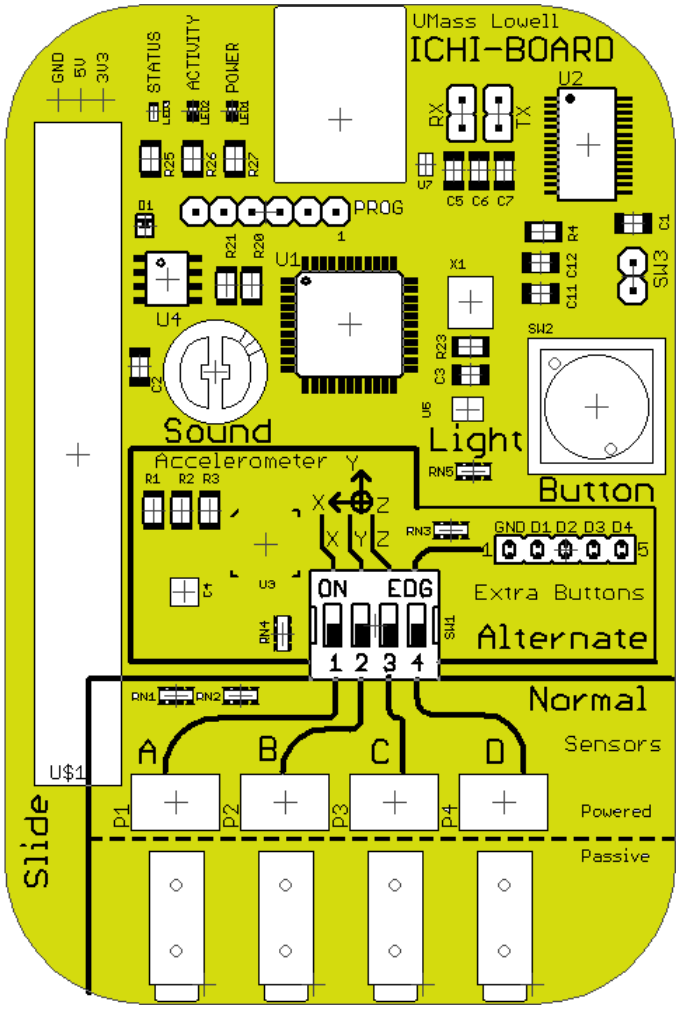
12. **Create a program to play a major chord.**
    - A major chord is the 1st, 3rd, and 5th notes of the scale, usually complemented by the octave above the 1st note. Thus, a G-major scale has notes G (#55), B (#59), D (#62), and G' (#67).
    - Another way to think about this is to compute the half-tone difference from the starting note: 0, 4, 7, 12.
    - Set a starting note and then use a "broadcast" to play the four notes simultaneously.

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

The IchiBoard

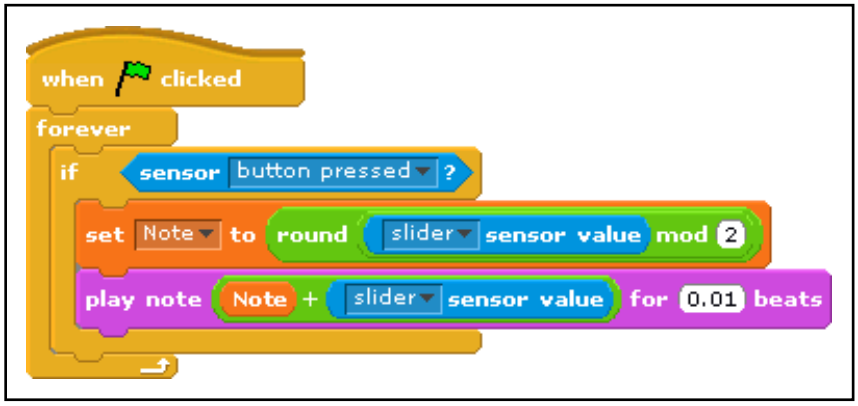
Board Layout

(courtesy of Mark Sherman, UMass Lowell Computer Science Engaging Computing Group)



Scratch Code for an IchiBoard Musical Instrument

(courtesy of Alex Ruthmann)



Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

The PicoBoard

CRICKET

With PicoCrickets, you can create musical sculptures, interactive jewelry, dancing creatures, and other playful inventions.

MAKE YOUR CREATIONS COME TO LIFE!

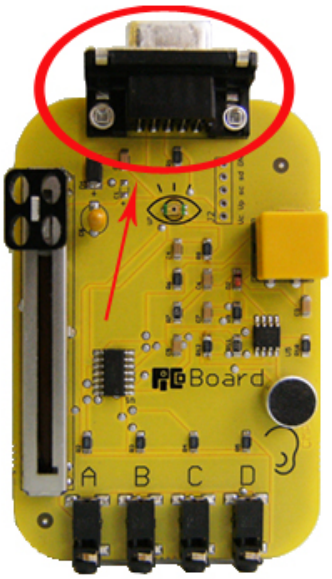
What Is It? | Order | Ideas | Support | Educators | About PICO | News

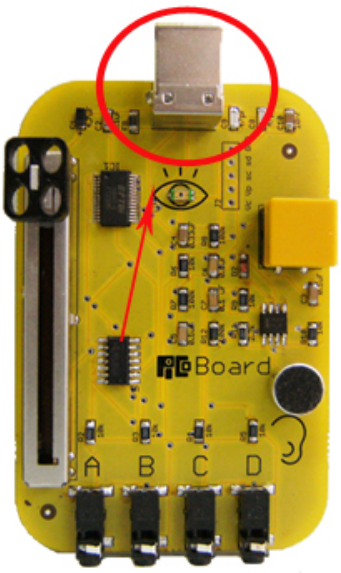
PICOBOARD SET UP

Which version of the PicoBoard do you have?

Serial
Does your PicoBoard connect to your computer using a Serial to USB Cable? All PicoBoards purchased before June 2009 are Serial PicoBoards.

USB
Does your PicoBoard connect to your computer using a USB to USB Cable? All PicoBoards purchased after June 2009 are USB PicoBoards.





info@playfulinvention.com

CREATE · PROGRAM · PLAY

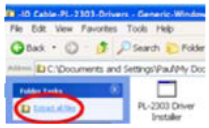
PicoBoard Serial to USB Setup

Windows Instructions

Windows XP (and older) users: Download the PicoBoard **Windows Driver (1471 KB)**

Windows Vista/7 users should just plug the USB to Serial cable in their computer and let the operating system choose the correct driver. Then skip to step #3 below.
If you do not have internet access on your computer, download the PicoBoard **Windows Driver (1471 KB)**.

1 First, open the file that you just downloaded. Click the link that says "Extract all files".



2 Then double-click on the file that you just extracted, and follow the on-screen instructions.

Mac Instructions

Mac OS X Driver (61 KB)

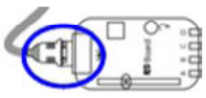
1 First, open the file that you just downloaded by clicking the magnifying glass in the Download panel.

2 Then double-click on the file that you just extracted, and follow the on-screen instructions.

Both Operating Systems:

3 Connect the USB part of your USB-Serial Cable to a USB port on your computer.

4 Connect the serial part of your USB-Serial Cable to the serial port on the PicoBoard.



5 Read **Getting Started with PicoBoards** to start making projects in **Scratch** with your PicoBoard.

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

PicoBoard USB to USB Setup

Windows Instructions

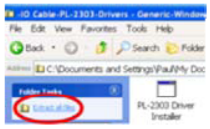
Windows XP users: Download the PicoBoard **Windows Driver (1.71 MB)**

Windows Vista/Windows 7 users should just plug the USB cable in their computer and let the operating system choose the correct driver. Then skip to step #3 below.
If you do not have internet access on your computer, download the PicoBoard **Windows Driver (1.71 MB)**.

Mac Instructions

Mac OS X Driver (419 KB)

1 First, open the file that you just downloaded. Click the link that says "Extract all files".



1 First, open the file that you just downloaded by clicking the magnifying glass in the Download panel.

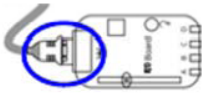
2 Then double-click on the file that you just extracted, and follow the on-screen instructions.

2 Then double-click on the file that you just extracted, and follow the on-screen instructions.

Both Operating Systems:

3 Connect the USB Cable to a USB port on your computer.

4 Connect the other part of the USB Cable to the USB port of the PicoBoard.



5 Read **Getting Started with PicoBoards** to start making projects in **Scratch** with your PicoBoard.

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Computing and Music: What Do They Have in Common?

Computing and music share deep structural similarities. For starters, both rely on notational symbol systems. Programming loops are typically delineated with opening and closing curly brackets { }, parentheses, or levels of indentation. Music loops are delineated with begin and end repeat signs 𝄆 𝄇 or initiated by "D.S." (Italian: *dal segno*), which instructs musicians to "repeat back to the sign," typically designated as 𝄋. As in programming, musical iteration can also make use of loop control variables. For example, Figure 1 shows a loop in which the music changes the second time through.

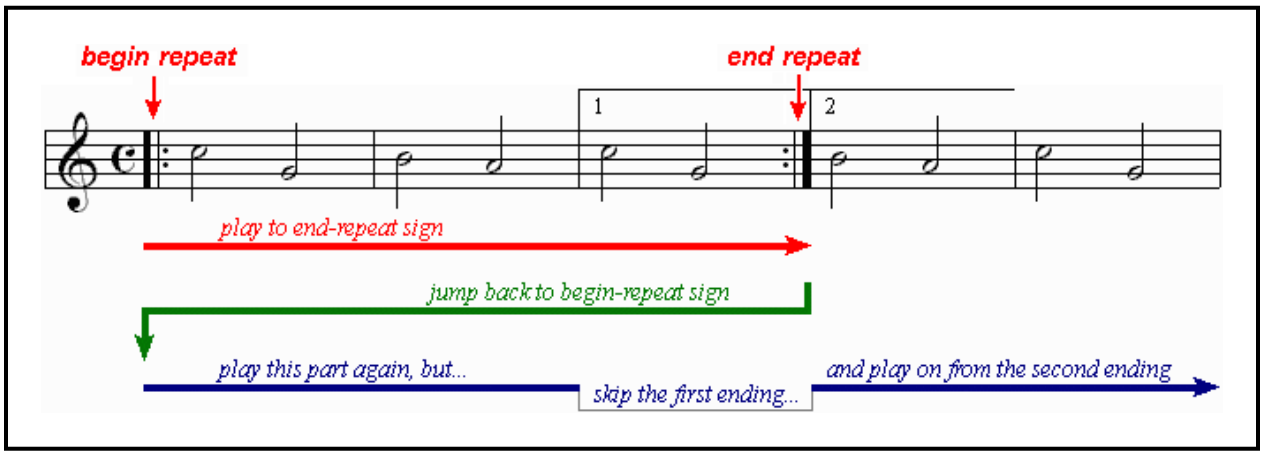


Figure 1. Musical iteration with a loop control variable. [6]

Both computing and music have logic and flow. Figure 2 shows the logic one student saw in The Beatles' All You Need Is Love. If one were to turn this flowchart into a computer program, it would not only contains loops, but if and switch statements as well.

One can also go the other way, converting musical concepts into computer programs. For example, the Scratch [2, 3] program in Figure 3a plays Jimmy Page's famous guitar riff from Led Zeppelin's Kashmir. This code works properly, but consider the many computational thinking (CT) concepts learned by transforming the code in Figure 3a to 3b, which plays exactly the same riff.

Computing and Music (cont'd)

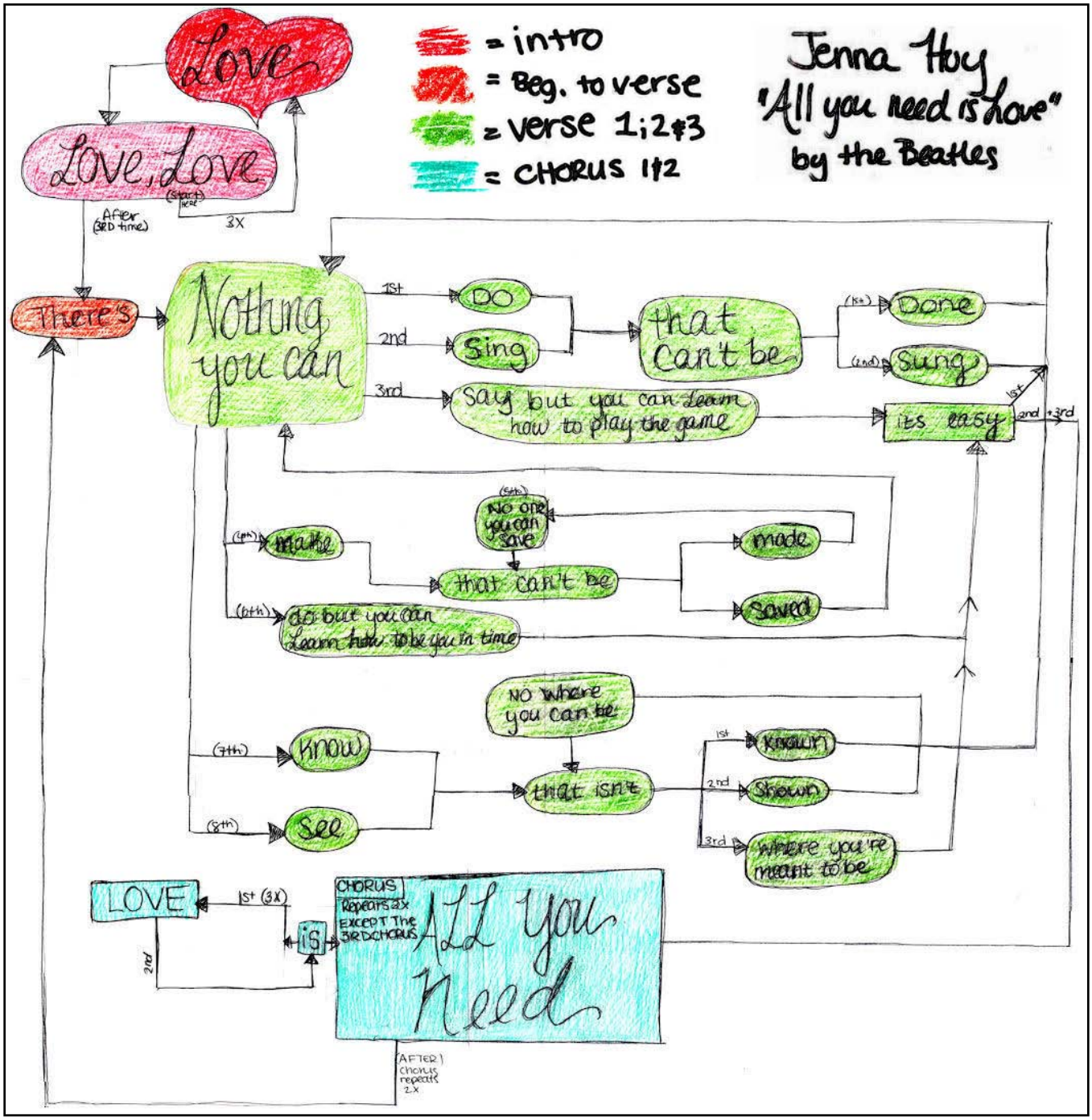
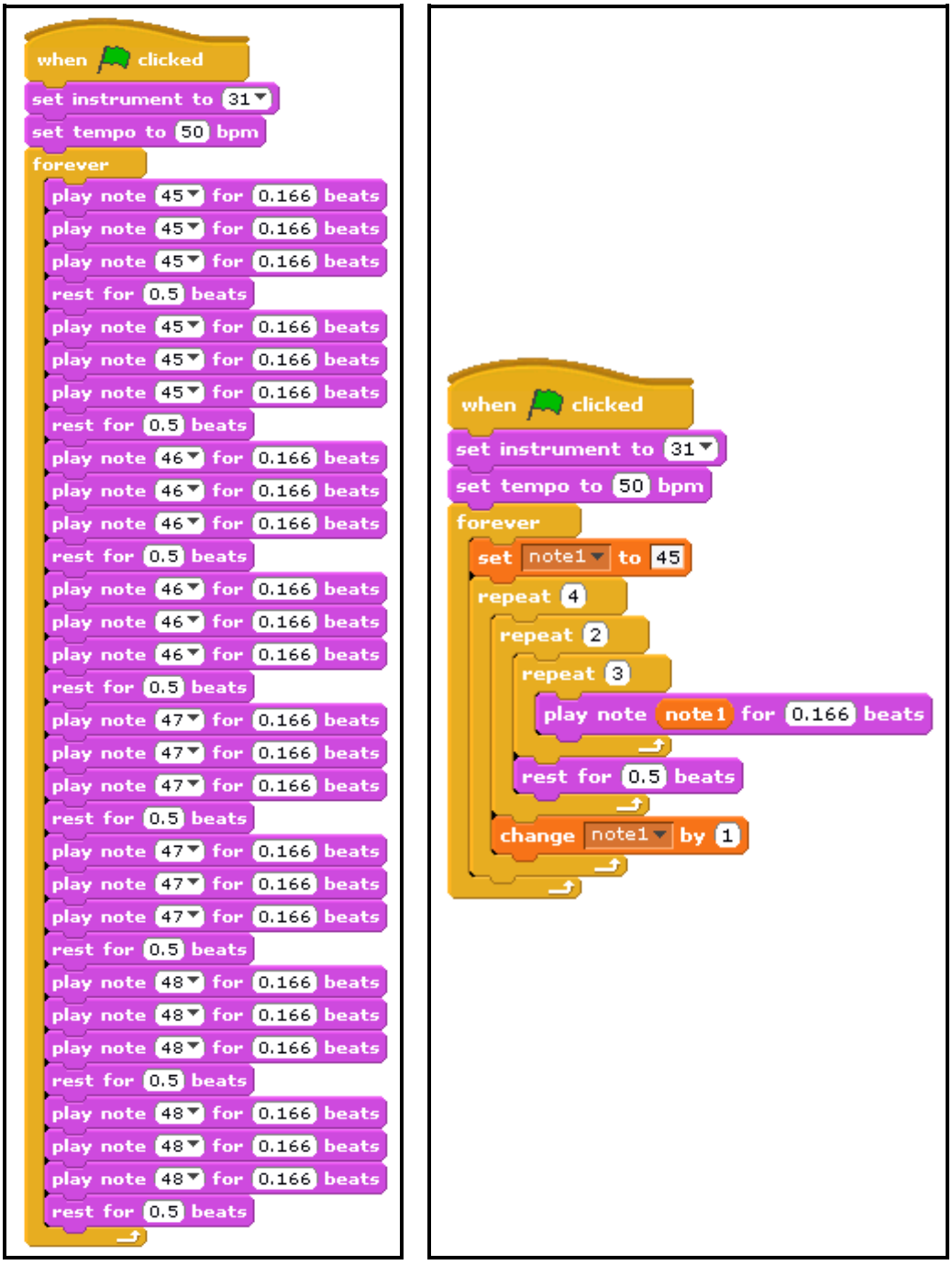


Figure 2. A song flowchart. [1]

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Computing and Music (cont'd)

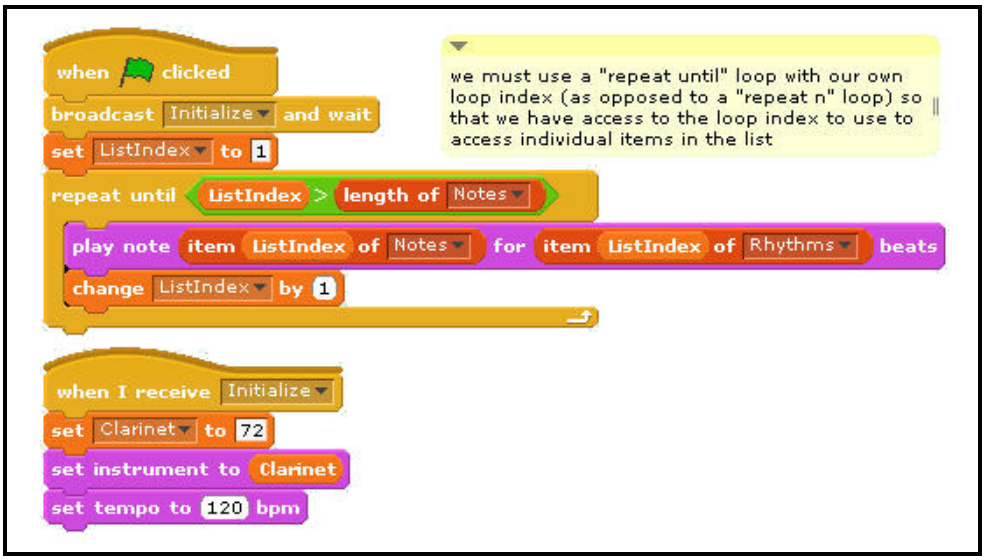


3a 3b

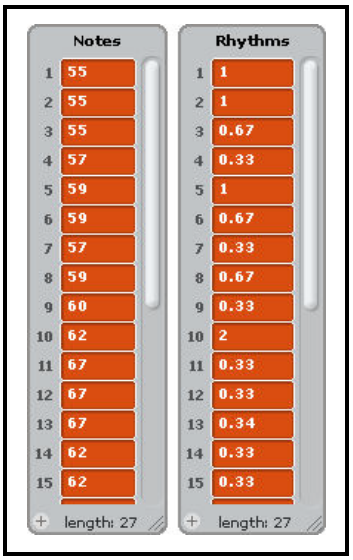
Figure 3. Two versions of Jimmy Page's *Kashmir* riff programmed in Scratch. [4]

Computing and Music (cont'd)

List and array data structures can be used to represent pitches and durations. Figure 4 shows an array (or indexed list) of MIDI note values paired with an array of note durations (in fractions of beats) that plays part of Row, Row, Row Your Boat. Using such structures, one can explore synchronization when the values are read by multiple threads with entrances staggered in time, resulting in the performance of a canon (or round).



4a



4b

Figure 4. Processing Scratch lists of notes and rhythms for *Row, Row, Row Your Boat*. [5]

References Cited

[1] Hoy, J. (2010). *Song flowchart for The Beatles' "All You Need is Love."* Created for a course assignment in "Sound Thinking."

[2] MIT Scratch Team (2009). *Scratch*. scratch.mit.edu accessed Dec. 21, 2009.

[3] Resnick, M., Maloney, J., Monroyhernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). *Scratch Programming for All*. Comm. of the ACM 52(11):60-67.

[4] Ruthmann, S.A. (2009). *Computational Zeppelin*. scratch.mit.edu/projects/alexruthmann/736779 accessed Jan. 5, 2010.

[5] Ruthmann, S.A., & Heines, J.M. (2010). *Exploring Musical and Computational Thinking Through Musical Live Coding in Scratch*. Scratch@MIT. Cambridge, MA.

[6] Smith, D.E. (1997). *Repeats, Second Endings, and Codas*. www.scenicnewengland.net/ uitar/notate/repeat.htm accessed Dec. 25, 2009.

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Computer Science, Math, and Music: Concepts Covered in Scratch

Computer Science

- statements
- sequential control flow
- iteration
- conditional execution
- arithmetic operators
- Boolean operators
- objects
- concurrency
- variables
- lists
- event handling
- user interaction
- optimization

Math

- positive and negative numbers
- real numbers
- decimal notation
- built-in functions with inputs
- angles
- Cartesian coordinates
- trigonometric operators
- random numbers

Music

- pitch
- rhythm (as duration)
- melodic fragments
- modes and scales
- polyphony
- synchronization
- harmony
- composing
- performing
- transposition
- balance and dynamics
- digital audio (as sound files)
- MIDI notes and timbres
- tempo
- form and structural analysis

Additional Readings

Ruthmann, S.A., Heines, J.M., Greher, G.R., Laidler, P., & Saulters, C. (2010). **Teaching Computational Thinking through Musical Live Coding in Scratch**. *41st ACM SIGCSE Technical Symposium on CS Education.* Milwaukee, WI, March 12, 2010.

http://teaching.cs.uml.edu/~heines/academic/papers/2010sigcse/SoundThinking-SIGCSE-2010.pdf

This paper discusses our ongoing experiences in developing an interdisciplinary general education course called Sound Thinking that is offered jointly by our Dept. of Computer Science and Dept. of Music. It focuses on the student outcomes we are trying to achieve and the projects we are using to help students realize those outcomes. It explains why we are moving from a web-based environment using HTML and JavaScript to Scratch and discusses the potential for Scratch's "musical live coding" capability to reinforce those concepts even more strongly.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., and Eastmond, E. (2010). **The Scratch Programming Language and Environment**. ACM Transactions on Computing Education 10(4). Article 16.

http://web.media.mit.edu/~jmaloney/papers/ScratchLangAndEnvironment.pdf

Scratch is a visual programming environment that allows users (primarily ages 8 to 16) to learn computer programming while working on personally meaningful projects such as animated stories and games. A key design goal of Scratch is to support self-directed learning through tinkering and collaboration with peers. This article explores how the Scratch programming language and environment support this goal.

Martin, F., Greher, G.R., Heines, J.M., Jeffers, J., Kim, H.J., Kuhn, S., Roehr, K., Selleck, N., Silka, L., and Yanco, H. (2009). **Joining Computing and the Arts at a Mid-Size University**. *2009 Conference of the Consortium for Computing Sciences in Colleges — Northeastern Region (CCSCNE 2009).* Plattsburgh, NY, April 24, 2009.

http://teaching.cs.uml.edu/~heines/academic/papers/2009ccscne/JoiningComputingAndArts.pdf

This paper describes two NSF-funded collaborations among faculty members in the Computer Science, Art, Music, and English departments at a public university in the Northeast USA. Our goal has been to create undergraduate learning opportunities across the university, focusing on connecting computer science to

Additional Readings (cont'd)

creative and expressive domains. In past publications, we have focused on student learning outcomes. This paper reports on the motivations, opportunities, and challenges for the faculty members involved.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, B. (2009). **Scratch: Programming for All**. *Communications of the ACM* 52(11):60-67.

http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf

"Digital fluency" should mean designing, creating, and remixing, not just browsing, chatting, and interacting. In this article we discuss the design principles that guided our development of Scratch and our strategies for making programming accessible and engaging for everyone.

Heines, J.M., Greher, G.R., & Kuhn, S. (2009). **Music Performamatics: Interdisciplinary Interaction**. *40th ACM SIGCSE Technical Symposium on CS Education.* Chattanooga, TN, March 7, 2009.

http://teaching.cs.uml.edu/~heines/academic/papers/2009sigcse/fp119-heines.pdf

This paper describes how a graphical user interface (GUI) programming course offered by the Dept. of Computer Science (CS) was paired with a general teaching methods course offered by the Dept. of Music in an attempt to revitalize undergraduate CS education and to enrich the experiences of both sets of students. The paper provides details on the joint project done in these classes and the evaluation that assessed its effect on the curriculum, students, and professors.

Urban, J. (organizer), Heines, J.M., Fox, E.A., & Taylor, H.G. (2009). **Panel on Revitalized Undergraduate Computing Education**. *40th ACM SIGCSE Technical Symposium on CS Education.* Chattanooga, TN, March 5, 2009.

http://teaching.cs.uml.edu/~heines/academic/papers/2009sigcse/sigcse2009panel-JMH-accepted.pdf

There is an imbalance in the supply and demand for computing professionals that has generated shortages in meeting personnel needs within industry. A major program was developed by the U.S. National Science Foundation to encourage innovations in undergraduate computing education. There are a variety of new projects that are revitalizing undergraduate computing education. One approach

Additional Readings (cont'd)

to such revitalization is the introduction of interdisciplinary courses to expand the scope of computing education. The basic idea is to have students from various disciplines work together on computing projects to expand their educational horizons and make computing courses more appealing. This panel brings together research managers with educators who have developed and taught interdisciplinary courses with these goals in mind.

Heines, J.M., Jeffers, J., & Kuhn, S. (2008). **Performamatics: Experiences With Connecting a Computer Science Course to a Design Arts Course**. *The International Journal of Learning* 15(2):9-16.

http://teaching.cs.uml.edu/~heines/academic/papers/2008learning/AsPublished-IntlJrnlLearning.pdf

This paper describes our efforts to stem the tide of declining CS enrollments by introducing innovations into our curriculum to give students more flexibility in course selection, especially in the freshman and sophomore years. Our approach is based on a partnership between the CS and Art, Music, and English departments in the area of exhibition and performance technologies.

In addition to describing our work, this paper provides the results of an evaluation conducted by an independent research. It reports on the impact this work has had on the CS and Art students and their respective projects, as well as on the professors and the way they teach their courses. It also describes steps that are being taken to improve the courses in the future.

Related Websites

Performamatics Website and Scratch Gallery and YouTube Channel

http://www. performamatics.org → http://teaching.cs.uml.edu/Performamatics/
http://www.scratchmusic.org → http://scratch.mit.edu/galleries/view/90913
http://www.youtube.com/performamatics

Scratch Projects by Performamatics People

http://scratch.mit.edu/users/alexruthmann (Music Prof. Alex Ruthmann)
http://scratch.mit.edu/users/drjay (CS Prof. Jesse Heines)
http://scratch.mit.edu/users/performamatics (additional collections)

Scratch Software

http://scratch.mit.edu (home page)
http://scratch.mit.edu/download (download page)
http://scratch.mit.edu/forums (discussion forums)

Scratch Resources for Teaching and Teachers

http://scratched.media.mit.edu (learn – share – connect for educators)

Scratch Project Galleries

http://scratch.mit.edu/channel/featured (featured projects)
http://scratch.mit.edu/galleries/browse/newest (members' personal galleries)

Scratch Information and Support

http://info.scratch.mit.edu/Support/Get_Started (getting started instructions)
http://info.scratch.mit.edu/sites/infoscratch.media.mit.edu/files/file/ScratchGettingStartedv14.pdf (Getting Started Guide)
http://info.scratch.mit.edu/Support/Reference_Guide_1.4 (Reference Guide)
http://info.scratch.mit.edu/Support (support page)
http://info.scratch.mit.edu/Video_Tutorials (video tutorials)
http://info.scratch.mit.edu/Support/Scratch_Cards (single-topic lessons)

Lifelong Kindergarten Group and Collaborators' Websites

http://llk.media.mit.edu (John Maloney and Mitchel Resnick)
http://teaching.cs.uml.edu (Jesse Heines)
http://www.alexruthmann.com (Alex Ruthmann)