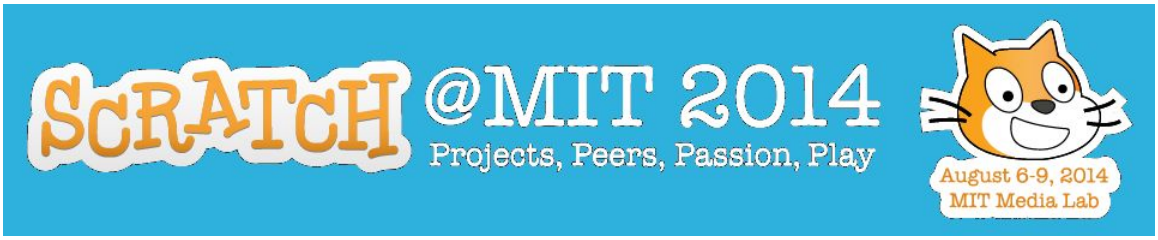
Scratch and Middle School Music

August 7, 2014

A workshop presented at



Workshop Leaders

Gena R. Greher, Ed.D., Prof. of Music Education
Gena_Greher@uml.edu

Jesse M. Heines, Ed.D., Prof. of Computer Science
Jesse_Heines@uml.edu

University of Massachusetts Lowell
Lowell, MA 01854

www.performamatics.org

This work is sponsored by the National Science Foundation
Transforming Undergraduate Education in STEM Program
Division of Undergraduate Education Award No. 1118435

Handout Contents

Workshop Description

This is a hands-on workshop geared specifically to middle school teachers on ways to infuse computing into the music curriculum. We are targeting middle schools because while the arts seem to be victims of budget cuts at the high school level, most states still require all middle school students to take courses in the arts, which of course include music. As a result, middle school music teachers come into contact with a large percentage of the students in their schools. An interdisciplinary approach in which the arts are integrated with STEM — into what has been called STEAM — could therefore include virtually every student in a middle school, having a huge impact on the program's reach and effectiveness within an entire school.

This work has grown out of our NSF-funded Performamatics project, through which we have developed numerous resources, models, and tools that integrate computing and music. Workshop participants will do one, two, or three of the activities of the type that we have used successfully with middle school students.

If time permits, we will conclude with a discussion of the barriers to implementation of interdisciplinary teaching in middle schools and possible ways to address those.

This work is supported by Award No. 1118435 from the National Science Foundation (NSF) Division of Undergraduate Education (DUE). It falls under the TUES program: Transforming Undergraduate Education in STEM (Science, Technology, Engineering, and Mathematics). Any opinions, findings, conclusions, or recommendations expressed in our materials are solely those of the authors and do not necessarily reflect the views of the National Science Foundation.

Workshop Materials

Sequencing MP3 Clips:
"Happy Sequencing Exercise"

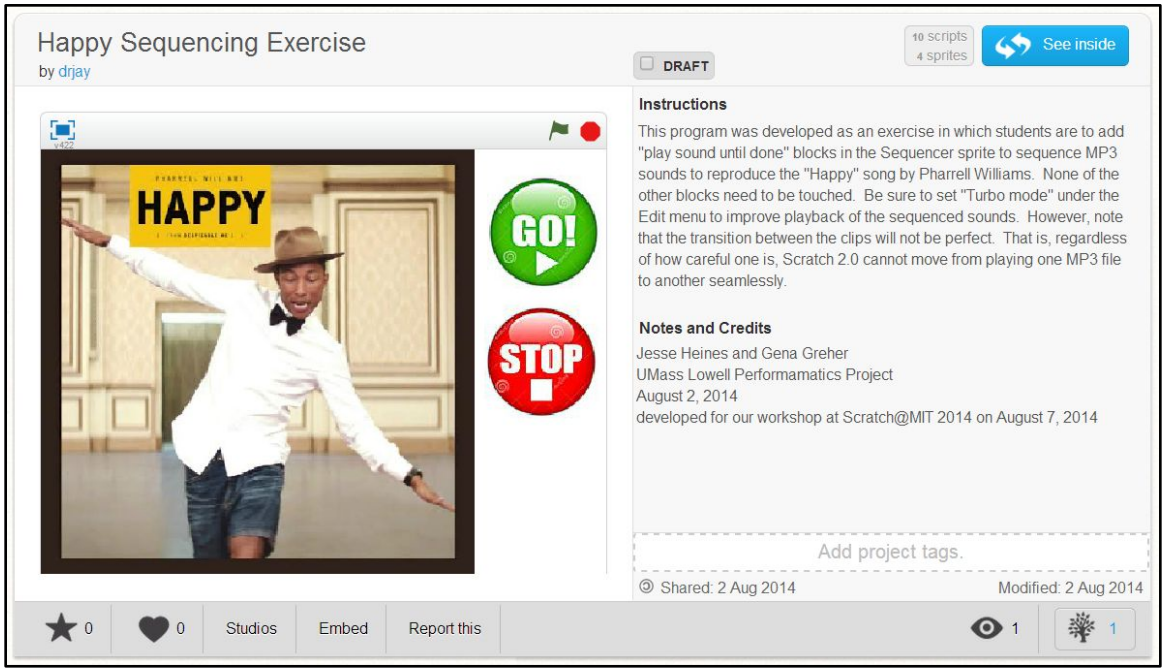
http://scratch.mit.edu/projects/25255474

Purpose

This program is designed to give students experience in critical listening and sequencing. Their task is to add a series of "play sound until done" blocks to a given program to sequence a series of MP3 clips. When properly sequenced, those blocks will play part of the "Happy" song by Pharrell Williams.

Note: *Be sure to set "Turbo mode" under the Edit menu to improve playback.*

Project Page



Sprites

The screen capture below shows the four sprites that comprise the program. The only one that students need to change is the "Sequencer" sprite. This sprite contains all the sound clips under the Sounds tab and has the first five sequenced via "play sound until done" blocks to help students get started.

Additional details on and explanations of these sprites are provided on the following pages.

Scripts in the "Control" Sprite

These scripts stop and play the student's sequence when the space bar and P keys are pressed, respectively. Broadcasts are used to avoid duplicating code in the "Go Button" and "Stop Button" sprites. *There is no need for students to change anything in this sprite.*

Scripts in the "Sequencer" Sprite

This is where students will add "play sound until done" blocks to sequence the MP3 clips under the Sound tab. The first five "play sound until done" blocks are provided to help students get started.

Scripts in the "Go Button" Sprite

These scripts provide functionality to the Go button. *There is no need for students to change anything in this sprite.*

Scripts in the "Stop Button" Sprite

These scripts provide functionality to the Stop button. *There is no need for students to change anything in this sprite.*

Sequencing Solution

Here is the full solution to this exercise. Each listed sound clip is present under the Sounds tab in the "Sequencer" sprite.

For each clip you see an animal name, a number, and a previous name. The animal names are the ones that must be used in the "play sound" blocks. The two additional pieces of information are provided to show how the exercise was developed. The number is the number of the sound clip under the Sounds tab, and the previous name is the name that was used during development to keep all the clips straight.

The previous names were converted to the animal names by manually renaming each clip under the Sounds tab individually. Fortunately, Scratch knows that when a clip is renamed there, it likewise has to rename the clip in all blocks that refer to it. It does this automatically, so you don't have to manually change your code.

Intro / Verse1

1.	geese	#4	was HappyIntro1
2.	giraffe	#6	was Line1
3.	puppy	#8	was 1stInstBreak
4.	lizard	#22	was Line2
5.	donkey	#21	was 2ndInstBreak
6.	horse	#10	was Line3
7.	snail	#5	was 3rdInstBreak
8.	rabbit	#24	was Liine4
9.	lobster	#1	was 4thInstBreak

Chorus

10.	turtle	#28	was Happy1
11.	starfish	#26	was Happy2
12.	dog	#27	was Happy3
13.	clam	#29	was Happy4

Verse 2

14.	kitten	#20	was 5_BadNews
15.	pony	#12	was 5thInstBreak
16.	pig	#16	was 6_GimmeAll
17.	cat	#17	was 7_BadNews
18.	snake	#19	was 8_WarnYa
19.	parrot	#18	was 8thInstBreak
20.	trout	#13	was 9_NoOffense
21.	shark	#3	was 10_Here'sWhy

Chorus

22.	turtle	#28	was Happy1
23.	starfish	#26	was Happy2
24.	dog	#27	was Happy3
25.	clam	#29	was Happy4

Break

26.	lion	#25	was 1_BringMeDown
27.	deer	#15	was 2_BringMeDown
28.	zebra	#7	was 3_BringMeDown
29.	bear	#23	was 4_BringMeDown
30.	cow	#11	was 5_BringMeDown
31.	tiger	#9	was 6_BringMeDown
32.	monkey	#14	was 7_BringMeDown
33.	leopard	#2	was 8_BringMeDown

Chorus

34.	turtle	#28	was Happy1
35.	starfish	#26	was Happy2
36.	dog	#27	was Happy3
37.	clam	#29	was Happy4

Finding Wrong Notes:
"Harry Potter Theme Excerpt With Wrong Notes"

http://scratch.mit.edu/projects/25256909

Purpose

This program also gives students experience in critical listening, but this time the task is to find the wrong notes in an excerpt from the Harry Potter theme. There are three versions of the theme excerpt provided:

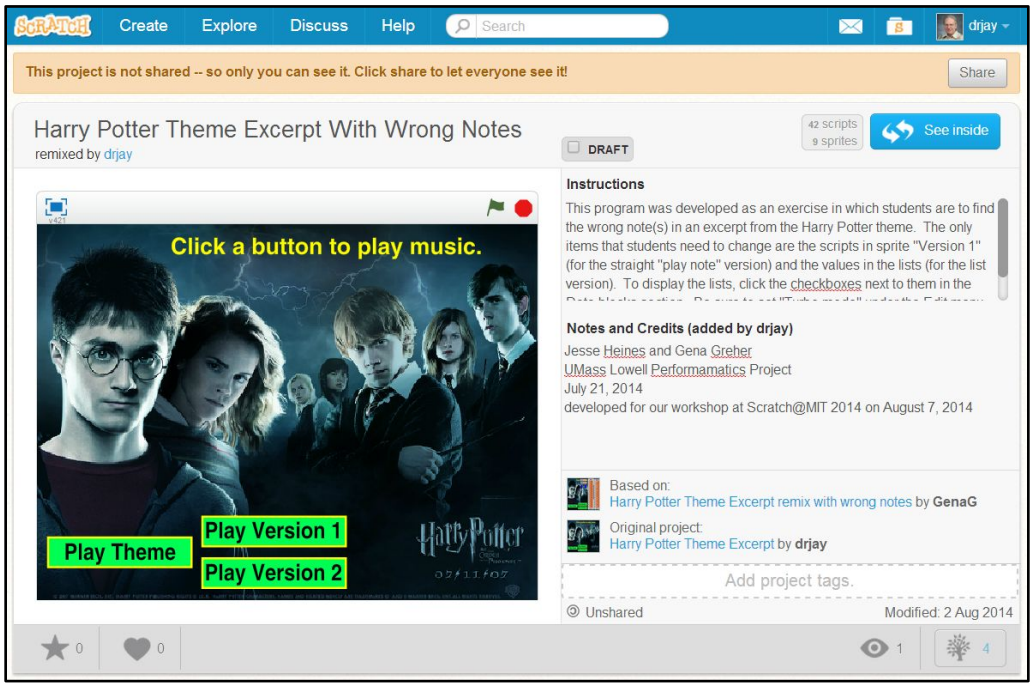
- the orchestral version in the "Play Theme" sprite accessed by clicking the "Play Theme" button
- a version encoded as sequences of "play note" blocks in the "Version1" sprite accessed by clicking the "Play Version 1" button
- a version encoded in two synchronized "Notes" and "Rhythms" lists that are read in the "Version2" sprite and accessed by clicking the "Play Version 2" button

Once students hear the wrong notes and figure out where they are, the only items they need to change to correct the program are two "play note" blocks in sprite "Version 1" (for the "play note" version) or two values in the "Notes" list (for the list version).

Note #1: The lists are not displayed by default. They are displayed when the "Play Version 2" button is clicked. To display them manually so that they can be changed, students must check the checkboxes next to their names in the Data code section.

Note #2: *Be sure to set "Turbo mode" under the Edit menu to improve playback.*

Project Page



Sprites

The program has nine sprites. As mentioned above, to correct the "play note" version, the only sprite that students need to change is the "Version1" sprite. To correct the "lists" version, students need to change values in the global "Notes" list, which can be displayed by checking its corresponding checkbox in the Data code section.

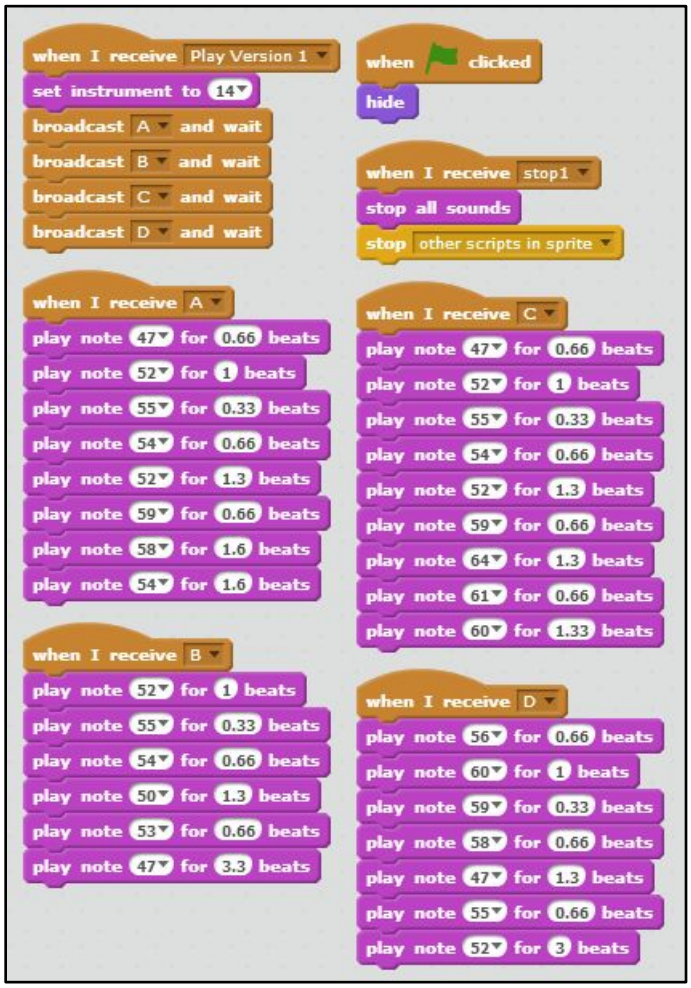
The "Offset Up" and "Offset Down" sprites enable the key of the "lists" version to be shifted up and down, respectively, but clicking these buttons when they appear while Version 2 is being played.

Note: Version 1 and Version 2 *sound* exactly the same. The music is merely encoded differently.

Scripts in the "Version1" Sprite

This is the version encoded as a series of "play note" blocks. The excerpt is broken up into four separate scripts for ease of programming the component musical phrases.

There are two wrong notes in this sprite that students are to find and correct.

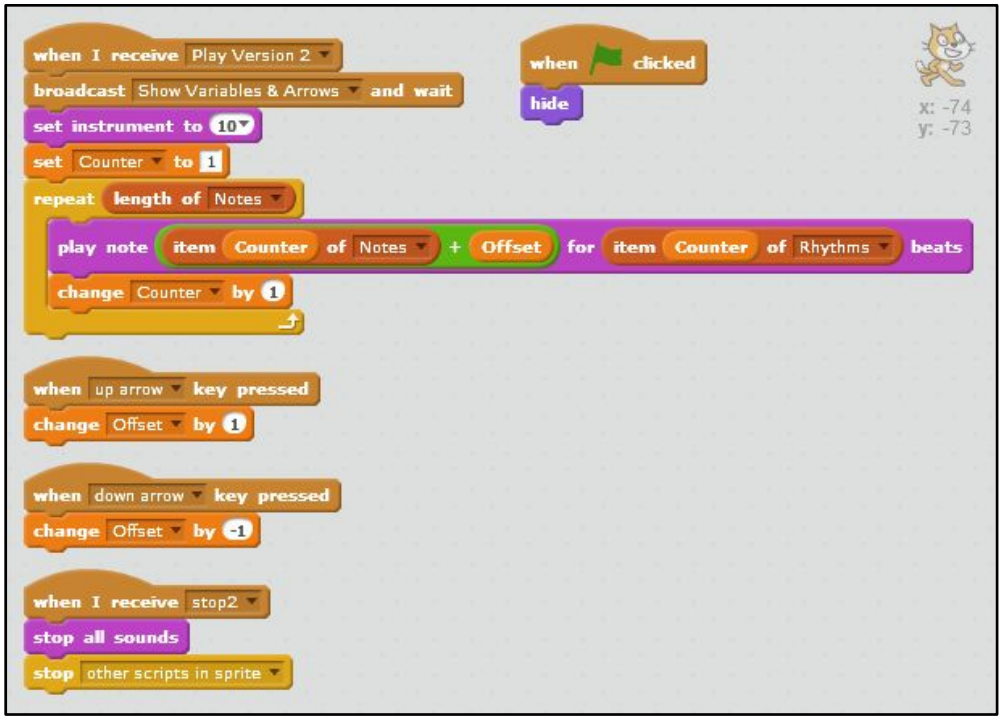


Scripts in the "Version2" Sprite

This is the version that plays the excerpt from two synchronized lists. This version sounds exactly the same as the previous version. It merely shows a different way to encode the music.

However, this version also allows the key to be changed by adding an "offset" to the MIDI value of the note to be played. The offset is initially set to +2 when the green flag is clicked in the "Initialize" sprite (provided below). This raises the key by 2 semi-tones. The offset can be increased by 1 semi-tone by pressing the up arrow key or by clicking the up arrow button (the "Offset Up" sprite) that appears when Version 2 is playing. Likewise, it can be decreased by 1 semi-tone by pressing the down arrow key or by clicking the down arrow button (the "Offset Down" sprite) that appears when Version 2 is playing.

There is no need for students to change anything in this sprite.



The values in the Notes and Rhythms lists are provided on the next page.

There are two wrong note values in the Notes list that students are to find and correct.

Notes List

1. 47
2. 52
3. 55
4. 54
5. 52
6. 59
7. 58
8. 54
9. 52
10. 55
11. 54
12. 50
13. 53
14. 47
15. 47
16. 52
17. 55
18. 54
19. 52
20. 59
21. 64
22. 61
23. 60
24. 56
25. 60
26. 59
27. 58
28. 47
29. 55
30. 52

Rhythms List

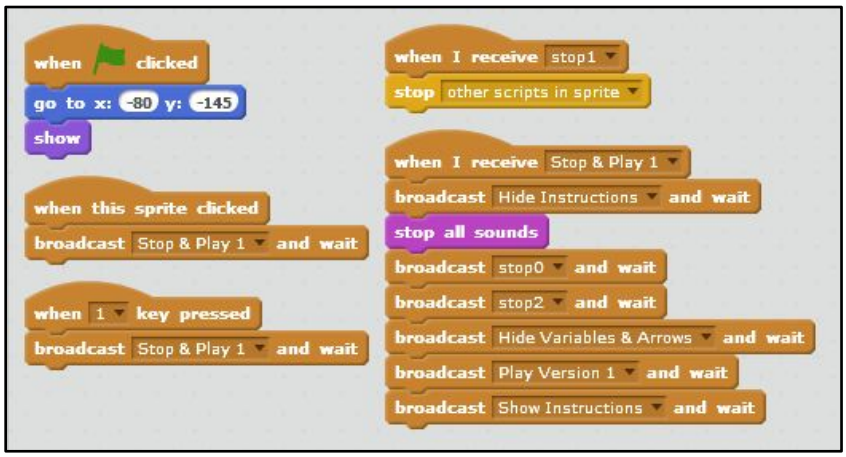
1. 0.66
2. 1.0
3. 0.33
4. 0.66
5. 1.3
6. 0.66
7. 1.6
8. 1.6
9. 1.0
10. 0.33
11. 0.66
12. 1.0
13. 0.66
14. 3.3
15. 0.66
16. 1.0
17. 0.33
18. 0.66
19. 1.3
20. 0.66
21. 1.3
22. 0.66
23. 1.33
24. 0.66
25. 1.0
26. 0.33
27. 0.66
28. 1.3
29. 0.66
30. 3

Scripts in the "Play Theme" Sprite

This sprite provides the button and code to play the recorded excerpt for reference so that students can hear how it is supposed to sound. *There is no need for students to change anything in this sprite.*

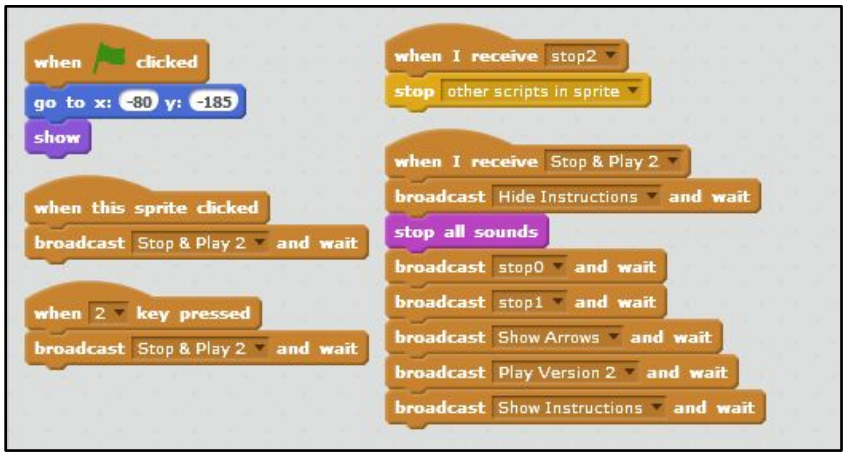
Scripts in the "Play Ver 1" Sprite

This sprite provides the button and code to play the "play note" version of the excerpt. *There is no need for students to change anything in this sprite.*



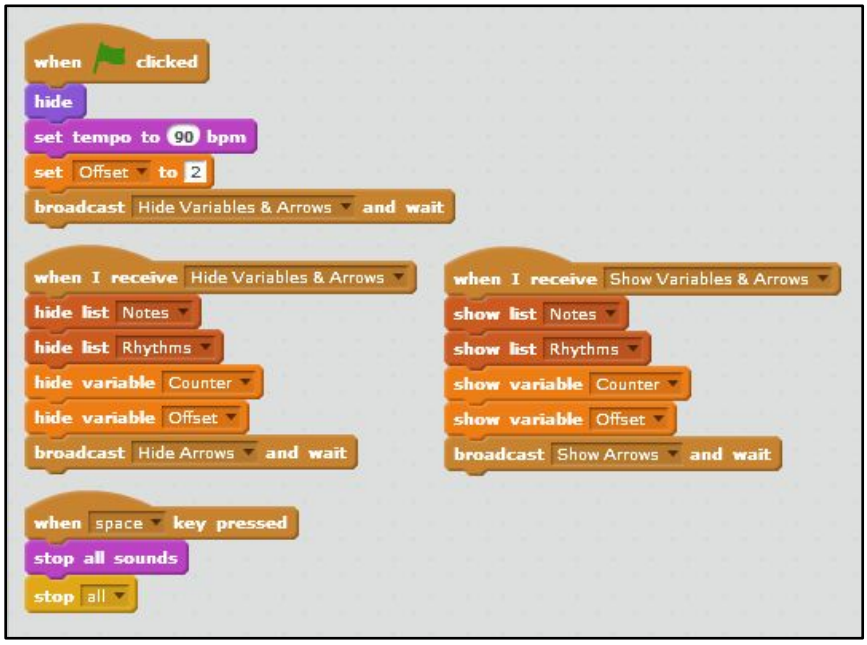
Scripts in the "Play Ver 2" Sprite

This sprite provides the button and code to play the "lists" version of the excerpt. *There is no need for students to change anything in this sprite.*



Scripts in the "Initialize" Sprite

This sprite performs initialization tasks to ensure that the program always looks the same when it starts up and contains code to show and hide the program's lists and variables as well as to stop the program cleanly. *There is no need for students to change anything in this sprite.*

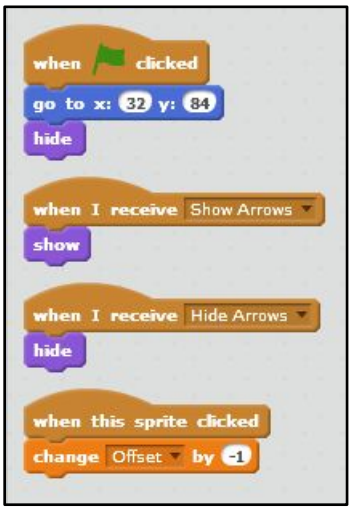


Scripts in the "Offset Up" Sprite

This sprite controls display of the up arrow that, when clicked, changes the key of the "lists" version ***up*** by one semi-tone. *There is no need for students to change anything in this sprite.*

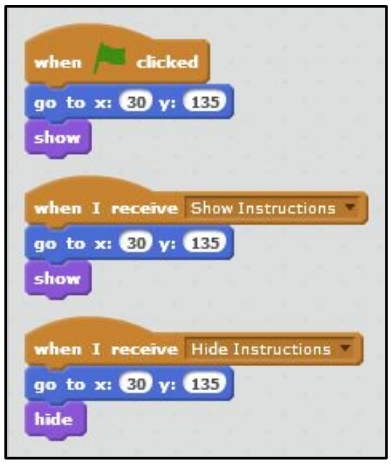
Scripts in the "Offset Down" Sprite

This sprite controls display of the up arrow that, when clicked, changes the key of the "lists" version ***down*** by one semi-tone. *There is no need for students to change anything in this sprite.*



Scripts in the "Instructions" Sprite

This sprite controls display of the main instructions that appear at the top of the window. *There is no need for students to change anything in this sprite.*



Coding Missing Phrases:
"What a Wonderful World in Pieces With Missing Phrases"

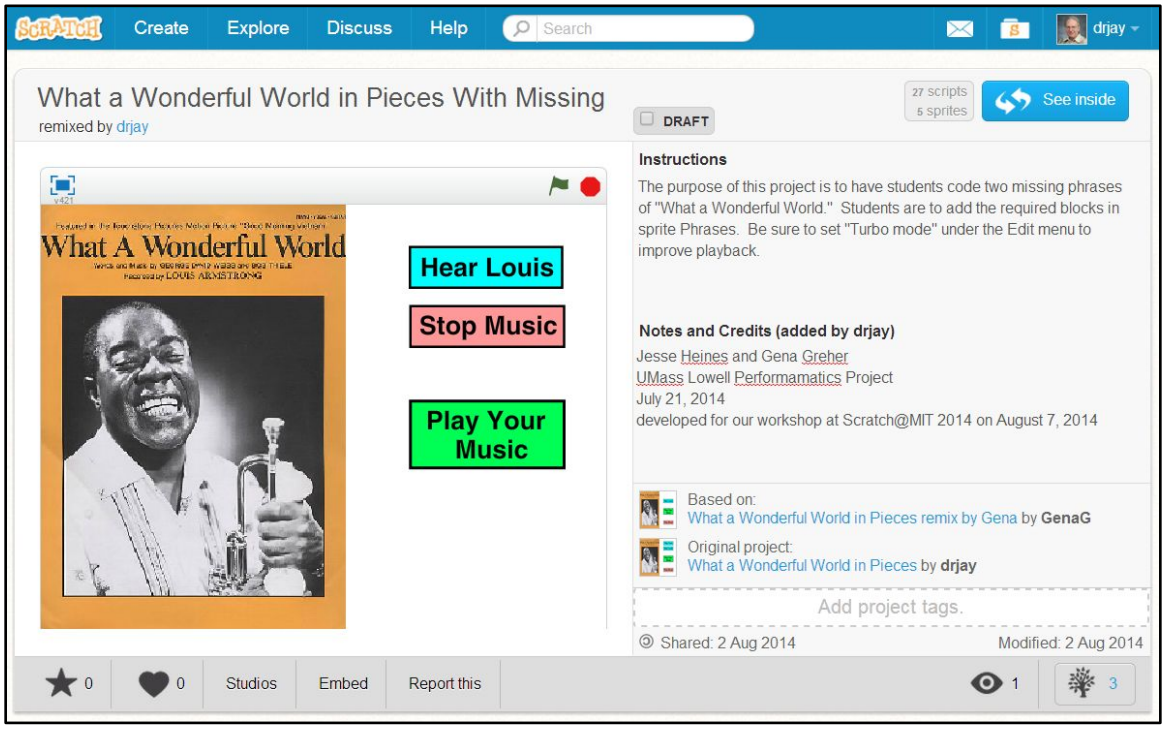
http://scratch.mit.edu/projects/25257125

Purpose

This program is designed to give students experience in critical listening and coding series of "play note" blocks in MIDI scripts. Their task is to identify the two missing phrases in "What a Wonderful World" as sung by Louis Armstrong and code those in the same manner as provided MIDI scripts that play other phrases in the song.

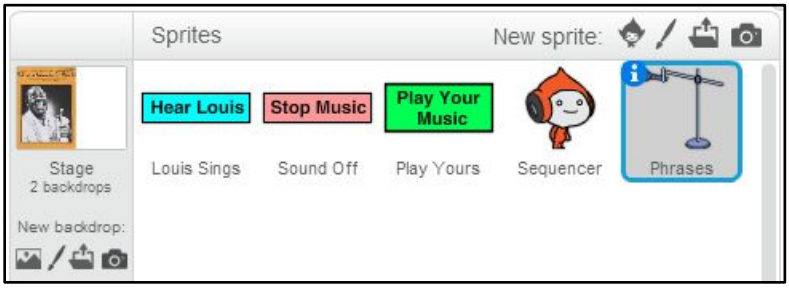
Note: *Be sure to set "Turbo mode" under the Edit menu to improve playback.*

Project Page



Sprites

This program has five sprites. The only one that students need to change is the "Phrases" sprite. This is the one that contains the song's phrases coded as series of "play note" blocks.



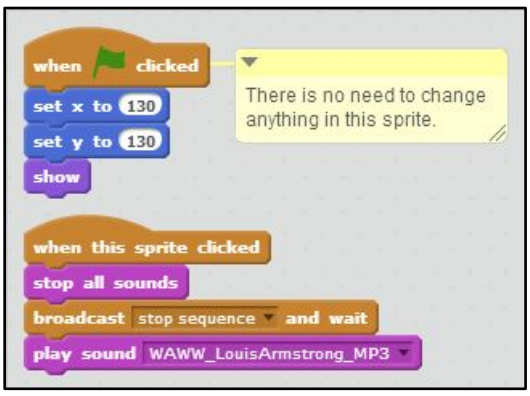
Scripts in the "Louis Sings" Sprite

The first script is executed at program startup to reset the position of the "Hear Louis" button just in case it inadvertently got dragged to a different location the last time the program was run.

The second script is executed when the "Hear Louis" button is clicked. It first stops any sound that is currently playing, which terminates all "play sound," "play drum," and "play note" blocks. It then broadcasts a "stop sequence" message that is "received" in the "Play Yours" sprite to stop execution of the scripts in that sprite. If those scripts continued, the next "broadcast" block would be executed and the music would continue causing both MP3- and MIDI-generated music to be played at the same time.

The third block in the "when this sprite clicked" script plays the MP3 version of Louis Armstrong singing the song excerpt we're working with.

There is no need for students to change anything in this sprite.



Scripts in the "Sound Off" Sprite

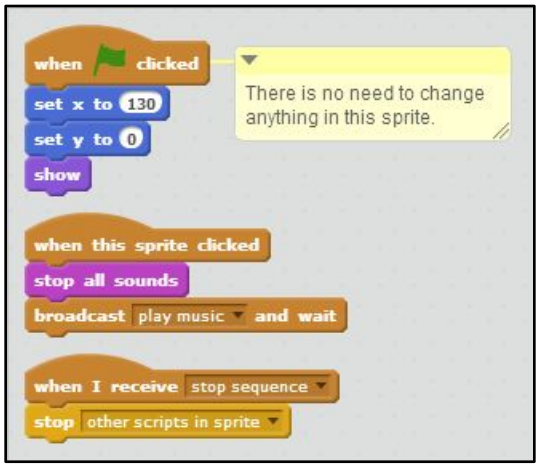
The first script in this sprite performs the same function as the first script in the "Hear Louis" sprite. That is, it resets the "Stop Music" button to ensure that it is positioned properly just in case it got dragged inadvertently. The second script stops the program completely. *There is no need for students to change anything in this sprite.*

Scripts in the "Play Yours" Sprite

The main purpose of this sprite is to trigger playing of the MIDI sequences in the "Sequencer" sprite. This is done by the second script in the sprite.

The first script is analogous to those in the two sprites discussed previously, and the last script merely stops the execution of the other scripts in this sprite when a "stop sequence" message is received.

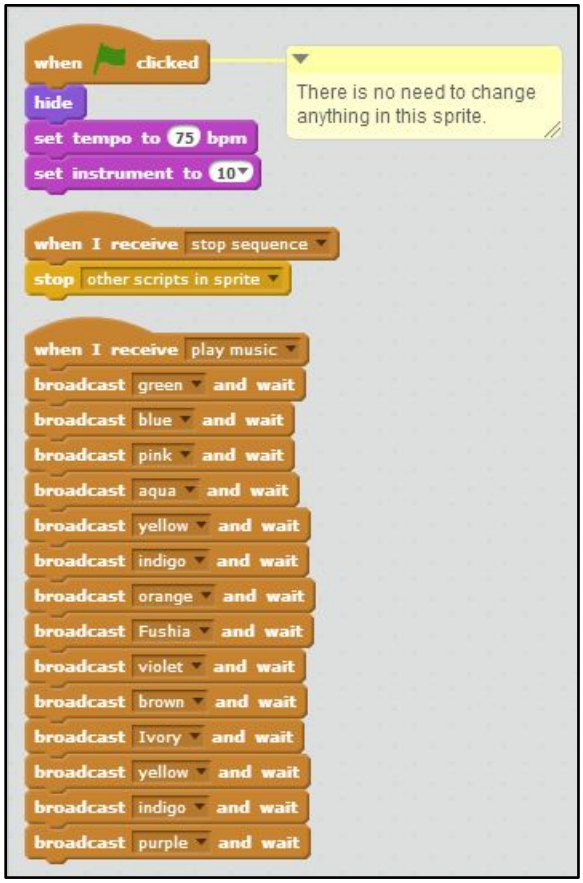
There is no need for students to change anything in this sprite.



Scripts in the "Sequencer" Sprite

This sprite contains the script that plays the phrases in the "Phrases" sprite. *There is no need for students to change anything in this sprite.*

Note: Coding this sequence is the point of the next exercise, "What a Wonderful World in Pieces for Sequencing." Therefore, it doesn't make much sense to give students both "What a Wonderful World" exercises, because the first contains the solution to the second. We suggest that you choose one or the other depending upon the abilities of your students and your teaching goals.



Scripts in the "Phrases" Sprite

This sprite contains all the scripts that play the various phrases in the excerpt we're working with. The students' assignment is

1. to listen to the music by clicking the "Play Your Music" button and identify which phrases of the music are missing,
2. and then to code those phrases by adding "play note" blocks under the appropriate "when I receive" blocks in the same manner as the other scripts in this sprite.

Sequencing MIDI Clips:
"What a Wonderful World in Pieces for Sequencing"

http://scratch.mit.edu/projects/25257878

Purpose

This program is a variation on the previous one. Rather than coding missing phrases, this variation provides all of the phrases in the excerpt we're working with and asks students to arrange them in the proper sequence to play the song.

As mentioned previously, it doesn't make sense to give students both variations of the "What a Wonderful World" exercise, because the previous one contains the solution to this one. We suggest that you choose one or the other depending upon the abilities of your students and your teaching goals.

Note: *Be sure to set "Turbo mode" under the Edit menu to improve playback.*

Project Page

The project page for this exercise is identical to that for the previous exercise.

Sprites

Like the program for the previous exercise, this one also has five sprites. But this time, the only one that students need to change is the "Sequencer" sprite. The "Phrases" sprite in this program contains all the phrases in the excerpt we're working with, including the two that were missing in previous exercise. Therefore, there is no need to change anything in the "Phrases" sprite for this exercise.

Scripts in the First Three Sprites

The "Louis Sings," "Sound Off," and "Play Yours" sprites are identical to those in the previous exercise, so they are not repeated here. *There is no need for students to change anything in these first three sprites.*

Scripts in the "Sequencer" Sprite

This is where students will add "broadcast and wait" blocks to play the phrases in the "Phrases" sprite.

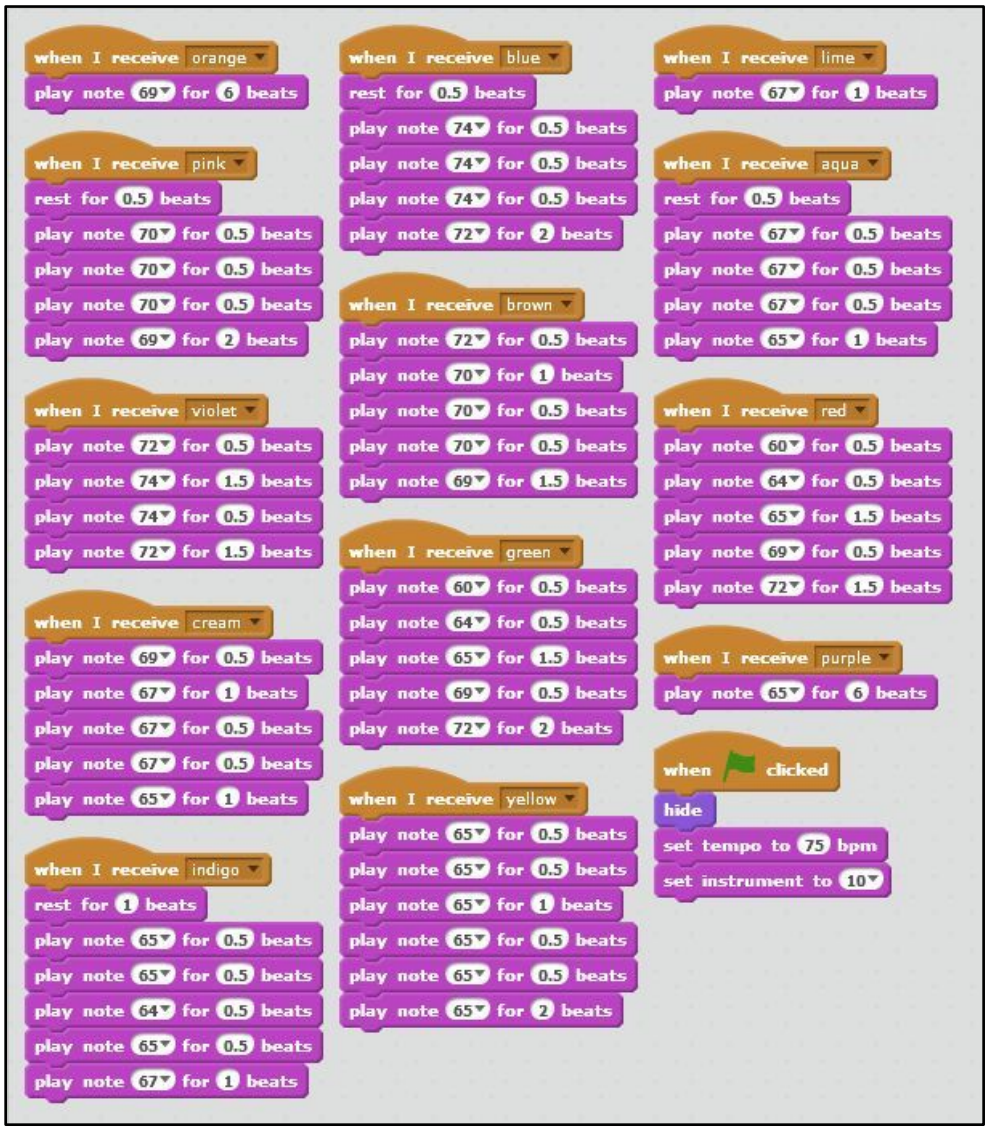
Note: The solution to this exercise, that is, the proper sequence of "broadcast and wait" blocks under the "when I receive [play music]" block, is in the "Sequencer" sprite in the previous exercise.

Scripts in the "Phrases" Sprite

This sprite contains all the phrases in the excerpt we're working with. Students are to reference each of these phrases — and some more than once — in the "broadcast and wait" blocks they add to the "Sequencer" sprite.

There is no need for students to change anything in this sprite.

Note: This sprite contains the solution to the two scripts that are missing from the previous exercise.



University of Massachusetts Lowell
Depts. of Music and Computer Science

Computational Thinking through Computing and Music

an interdisciplinary NSF TUES project

Now Available

Computational Thinking In Sound
by Gena R. Greher and Jesse M. Heines
Oxford University Press, April 2014

To attend our workshop, please complete the **workshop application form**.

Participants' comments on our workshops

Our sixth (and final!) **two-day** interdisciplinary workshop will take place on:

Thursday & Friday, January 15-16, 2015

at the UMass Lowell Inn & Conference Center in Lowell, Massachusetts

June 2014 Post-Workshop Evaluation Form

To apply to attend this last workshop, please complete the **workshop application form**.

Scenes from the June 21-22, 2012, workshop

Our goal is to develop and disseminate ways to enhance students' grasps of computational thinking by engaging them in fundamental concepts that unite computing and music. Our approach leverages students' near universal interest in music as a context and springboard for engaging in rich computational thinking experiences. Prior work in an NSF CPATH project showed this approach to be effective at creating value in both discipline-specific courses for Computer Science and Music majors, as well as General Education courses for all majors. This project will develop additional activities to deepen students' experiences in computing and music, and explore additional techniques for evaluating learning through those activities. The project will also disseminate our work through workshops for pairs of interdisciplinary faculty at 4- and 2-year colleges.

Our materials teach concepts such as modularization by breaking songs down into their components, looping and subroutines by noting where musical phrases are repeated intact and with small variations (requiring parameters), logic flow by creating musical flowcharts, and algorithms by writing programs that generate music. New materials will explore ways to teach more advanced computing concepts such as threads and synchronization by writing programs that play multiple parts simultaneously and use various Application Programmer Interfaces (APIs), allowing us to combine software platforms into systems that to do more than is possible by one alone.

A major component of this project is the sharing of our techniques and materials through sponsored workshops at conferences and on-site at universities where participants attend as a pair: at least one from Computer Science (or another science or engineering department) and one from Music (or another arts department). This will ensure that collaborations begun in the workshops have a foothold on sustainability when the participants return to their own institutions.

This project is supported by Award No. 1118435 from the National Science Foundation (NSF) Division of Undergraduate Education (DUE). It falls under the TUES program: Transforming Undergraduate Education in STEM (Science, Technology, Engineering, and Mathematics). Any opinions, findings, conclusions, or recommendations expressed in our materials are solely those of the authors and do not necessarily reflect the views of the National Science Foundation.

University of Massachusetts Lowell
Depts. of Music and Computer Science

Computational Thinking through Computing and Music

an interdisciplinary NSF TUES project

Jesse Heines, *Principal Investigator*, is a Professor of Computer Science at UMass Lowell with a strong interest in music and its power to interest students in computing. He teaches courses on graphical user interfaces, web programming, and C++, and co-teaches the Sound Thinking course with Gena and Alex. He was PI on an NSF CPATH award and is currently PI on an NSF TUES award. Gena and Jesse are working on a book on interdisciplinary teaching that is currently under contract with Oxford University Press. To keep his music alive, Jesse sings with the Lowell "Gentlemen Songsters" Chapter of the Barbershop Harmony Society.

Gena Greher, *Co-Principal Investigator*, is a Professor of Music Education at UMass Lowell. Her research focuses on creativity and listening skill development in children and examining the influence of integrating multimedia technology in urban music classrooms, as well as in the music teacher education curriculum and School-University partnerships. Recent projects include: a music technology mentor/partnership with UMass Lowell music education students and two local K-8 schools; *SoundScape*, a technology-infused music intervention program for teenagers with autism spectrum disorders; and *Performamatics*, an NSF-supported project linking computer science to the arts. Before entering the education profession and crossing paths with Jesse and Alex, Gena was a music director in advertising, working for several multinational advertising agencies producing the jingles and underscores for hundreds of commercials.

S. Alex Ruthmann, *Co-Principal Investigator*, is an Associate Professor of Music Education and Music Technology at New York University. Beginning as a middle school music teacher and computational musician, he now teaches undergraduate and graduate courses at the intersection of music education, arts computing, and research. He has published on technology-mediated music learning and teaching, children's musical and compositional processes, and fostering learner agency through music and technology. Alex's research explores social/digital media musicianship and creativity, as well as the development of technologies for music learning, teaching and engagement for use in schools and community-based arts+computing programs.

Brendan Reilly, *Research Assistant*, is an undergraduate Computer Science major at UMass Lowell, expecting to complete his B.S. in Computer Science in 2014. He has played bass since grade school, participating in every musical group available to him. After taking a course on Java, however, he decided to go into CS while keeping music in his background.

Contact Information

Jesse M. Heines, *Principal Investigator*
Dept. of Computer Science
University of Massachusetts Lowell
Lowell, MA 01854
e-mail: **Jesse_Heines@uml.edu**
phone: +1 (978) 934-3634

Gena R. Greher, *Co-Principal Investigator*
Dept. of Music
University of Massachusetts Lowell
Lowell, MA 01854
e-mail: **Gena_Greher@uml.edu**
phone: +1 (978) 934-3893

S. Alex Ruthmann, *Co-Principal Investigator*
Dept. of Music & Performing Arts Professions
New York University
New York, NY 10003
e-mail: **Alex.Ruthmann@nyu.edu**
phone: +1 (212) 998-5607

Scott Lipscomb, *Project Evaluator*, is an Associate Professor of Music Education at the University of Minnesota. He teaches courses in music education, music cognition, and music technology. Scott's research areas include facilitation of music learning through technology integration, interactive instructional media development, and multimedia cognition. His work has been published in numerous peer-reviewed journals and edited volumes, and he presents frequently at national and international conferences. Scott is Editor of the *Journal of Technology in Music Learning*, immediate past President of the *Association for Technology in Music Instruction*, Research Committee Chair for the *Technology Institute for Music Educators*, and Treasurer for the *Society for Music Perception and Cognition*.

Fred Martin, *Senior Personnel*, is an Associate Professor of Computer Science and serves as Associate Dean of the College of Sciences at UMass Lowell. He teaches courses in robotics, programming languages, software engineering, and artificial intelligence, and co-taught the Performamatics interdisciplinary Tangible Interaction Design course. In 2006, Fred received an NSF Faculty Early Career Development award (REC- 0546513, $599,943). His present focus is science inquiry, using electronic sensors for data collection and the web for data-sharing and visualization.

Sarah Kuhn, *Senior Personnel*, is a Professor of Psychology at UMass Lowell. She is commited to innovation in learning, particularly blending technical education with the social sciences and arts, which led her to create the UMass Lowell Laboratory for Interdisciplinary Design. Sarah is a member of the Social Science Advisory Board of the NSF


Zachary Robichaud, *Research Assistant*, is a graduate Computer Science major at UMass Lowell expecting to complete his M.S. in Computer Science in 2015. He has been a musician since he was in second grade, playing the guitar, piano, and trombone. Zack received his bachelor's degree in Sound Recording Technology from UMass Lowell in 2012. He hopes to use his knowledge of music and sound recording with his master degree to create and develop music and audio applications.

funded National Center for Women & Information Technology and was a member of the National Research Council Committee on Workforce Needs in Information Technology. She was Co-PI of Project TechForce, an NSF-funded study of women and men working in the Massachusetts software and Internet industry.

University of Massachusetts Lowell
Depts. of Music and Computer Science

Computational Thinking through Computing and Music

an interdisciplinary NSF TUES project

Workshop participants are required to attend in **interdisciplinary pairs**, preferably from the same institution. This will ensure that the workshop itself models interdisciplinary collaboration and produces outcomes that connect directly to participants' own situations.

We welcome professors and instructors from 2- and 4-year colleges and universities. High school teachers should contact the instructors to attend by special arrangement.

Please complete the form below as best you can. Questions should be directed to Jesse Heines at Jesse_Heines@uml.edu.

Fields marked with an asterisk are required.

Which workshop do you wish to attend? *

- ~~COMPLETED *One day:* **April 27, 2012**, 9:00 AM to 3:00 PM, at UMass Lowell **in conjunction with ASEE-NE**~~
- ~~COMPLETED *Two day:* **June 21-22, 2012**, 9:00 AM to 5:00 PM, at UMass Lowell~~
- ~~COMPLETED *Two day:* **January 17-18, 2013**, 9:00 AM to 5:00 PM, at UMass Lowell~~
- ~~COMPLETED *One day:* **March 5, 2013**, 9:00 AM to 5:00 PM, in Denver, CO, **in conjunction with ACM SIGCSE**~~
- ~~COMPLETED *Two day:* **June 20-21, 2013**, 9:00 AM to 5:00 PM, at UMass Lowell~~
- ~~COMPLETED *Two day:* **January 16-17, 2014**, 9:00 AM to 5:00 PM, at UMass Lowell~~
- ~~COMPLETED *Two day:* **June 19-20, 2014**, 9:00 AM to 5:00 PM, at NYU~~
- *Two day:* **January 15-16, 2015**, 9:00 AM to 5:00 PM, at UMass Lowell

Please tell us about yourself...

Your Full Name *	
Institution *	
Department *	
Email Address *	
Work Phone Number *	
Cell Phone Number	
Website URL	
Courses You Teach *	
Please state briefly why you would like to attend this workshop. *	

Please tell us about the other member of your pair...

His or Her Full Name *	
Institution *	
Department *	
Email Address *	
Work Phone Number *	
Cell Phone Number	
Website URL	
Courses He or She Teaches *	

Are you requesting support for travel? * ○ **Yes** ◉ **No**

Travel support is only available for two-day workshops.

Please tell us how you heard of this workshop...

How or where did you hear about this workshop? *	

Confirm Information

Additional Web Resources

Performamatics Project Website
..... http://www.performamatics.org

Computational Thinking in Sound Website
..... http://www.compthinkinsound.org

Sound Thinking Course Website (for the most recent semester)
..... http://soundthinking.uml.edu

Related Scratch User Accounts
Performamatics ..... http://scratch.mit.edu/users/performamatics
Jesse Heines ..... http://scratch.mit.edu/users/drjay
S. Alex Ruthmann ..... http://scratch.mit.edu/users/alexruthmann
NYU Performamatics http://scratch.mit.edu/studios/222541

S. Alex Ruthmann Blog Post re Being More Musical In and With Scratch
..... http://www.alexruthmann.com/blog1/?p=641

MaKey MaKey Website
..... http://makeymakey.com

S. Alex Ruthmann Experiencing Audio page on MaKey MaKey Music
..... http://www.experiencingaudio.org/2012/11/makey-makey-music-workshop-materials.html

MaKey MaKey Musical Construction Kit and links to Ruthmann students' MaKey MaKey music projects
..... http://makeymakeymusicalconstructionkit.blogspot.com

Eric Rosenbaum (MaKey MaKey inventor) Home Page
..... http://web.media.mit.edu/~ericr

Jesse Heines Home Page
..... https://teaching.cs.uml.edu

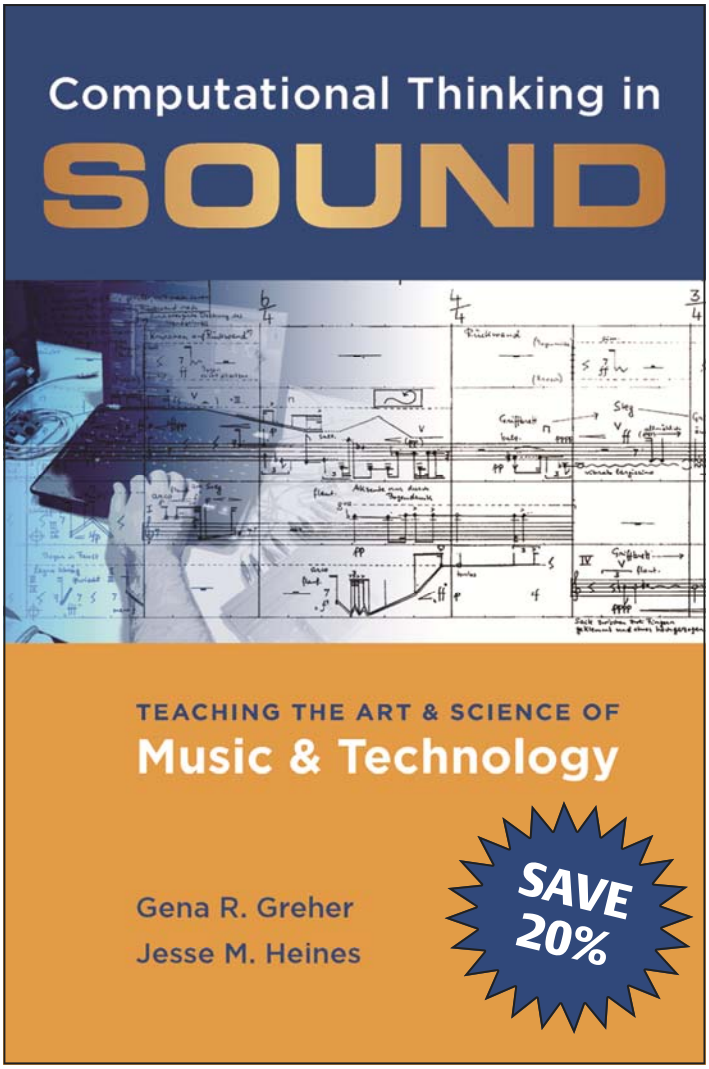
NEW FROM OXFORD

COMPUTATIONAL THINKING IN SOUND

TEACHING THE ART AND SCIENCE OF MUSIC AND TECHNOLOGY

By Gena R. Greher and Jesse M. Heines

With *Computational Thinking in Sound*, veteran educators Gena R. Greher and Jesse M. Heines provide the first book ever written for music fundamentals educators that is devoted specifically to music, sound, and technology. Using a student-centered approach that emphasizes project-based experiences, the book provides music educators with multiple strategies to explore, create, and solve problems with music and technology in equal parts. It also provides examples of hands-on activities that encourage students, alone and in groups, to explore the basic principles that underlie today's music technology and freely available multimedia creation tools. *Computational Thinking in Sound* is an effective tool for educators to introduce students to the complex process of computational thinking in the context of the creative arts through the more accessible medium of music.



March 2014

Hardcover | 272 pp. | ~~$35.00~~ **$28.00**

978-0-199-826190-3

Gena R. Greher is a Professor of Music Education at UMass Lowell. Her research focuses on creativity and listening skill development in children and examining the influence of integrating multimedia technology in urban music classrooms and music teacher education through School-University partnerships.

OXFORD UNIVERSITY PRESS

Jesse M. Heines is a Professor of Computer Science at UMass Lowell. His primary teaching responsibilities include object-oriented programming and graphical user interfaces. His research focuses on computer science education, particularly interdisciplinary approaches that blend computer science with music and other fields to enhance instructional effectiveness in both.

Order online at **www.oup.com/us**. Enter promotion code **28862** to save **20%**