Progressive Music Examples

prepared for a workshop at

Scratch@MIT

Friday, August 13, 2010

S. Alex Ruthmann

Prof. of Music Education

Alex_Ruthmann@uml.edu

Jesse M. Heines

Prof. of Computer Science

Jesse_Heines@uml.edu

University of Massachusetts Lowell

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.

Program Source Locations

http://scratch.mit.edu/users/performamatics

Gallery: "Scratch@MIT, August 13, 2010"
http://scratch.mit.edu/galleries/view/90913

Table of Examples

Important Note

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

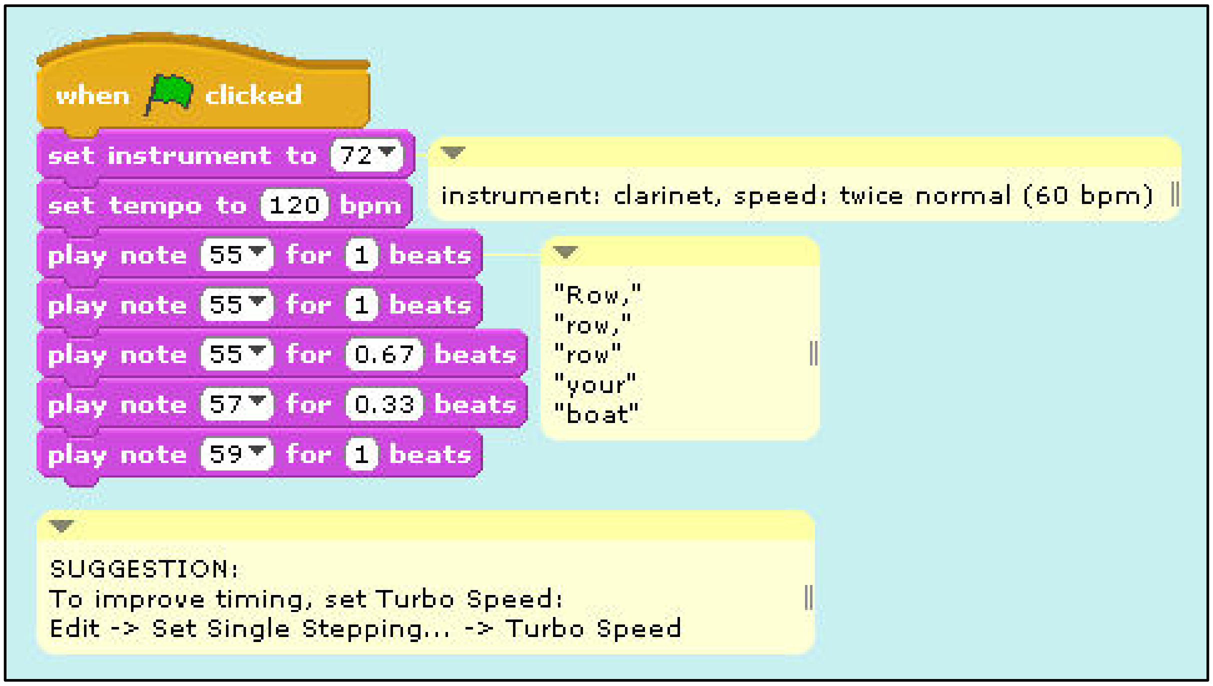
Edit ➔ Set Single Stepping... ➔ Turbo Speed

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.

Progressive Music Examples

No. 1: Playing Notes

Single Script



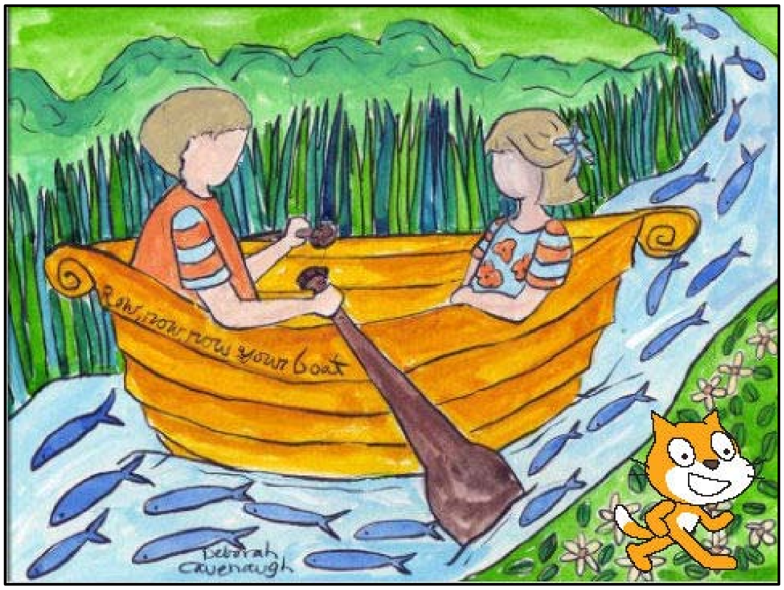
```
when ⚑ clicked
set instrument to 72
set tempo to 120 bpm
play note 55 for 1 beats
play note 55 for 1 beats
play note 55 for 0.67 beats
play note 57 for 0.33 beats
play note 59 for 1 beats
```

instrument: clarinet, speed: twice normal (60 bpm)

"Row,"
"row,"
"row"
"your"
"boat"

SUGGESTION:
To improve timing, set Turbo Speed:
Edit -> Set Single Stepping... -> Turbo Speed

Output Window

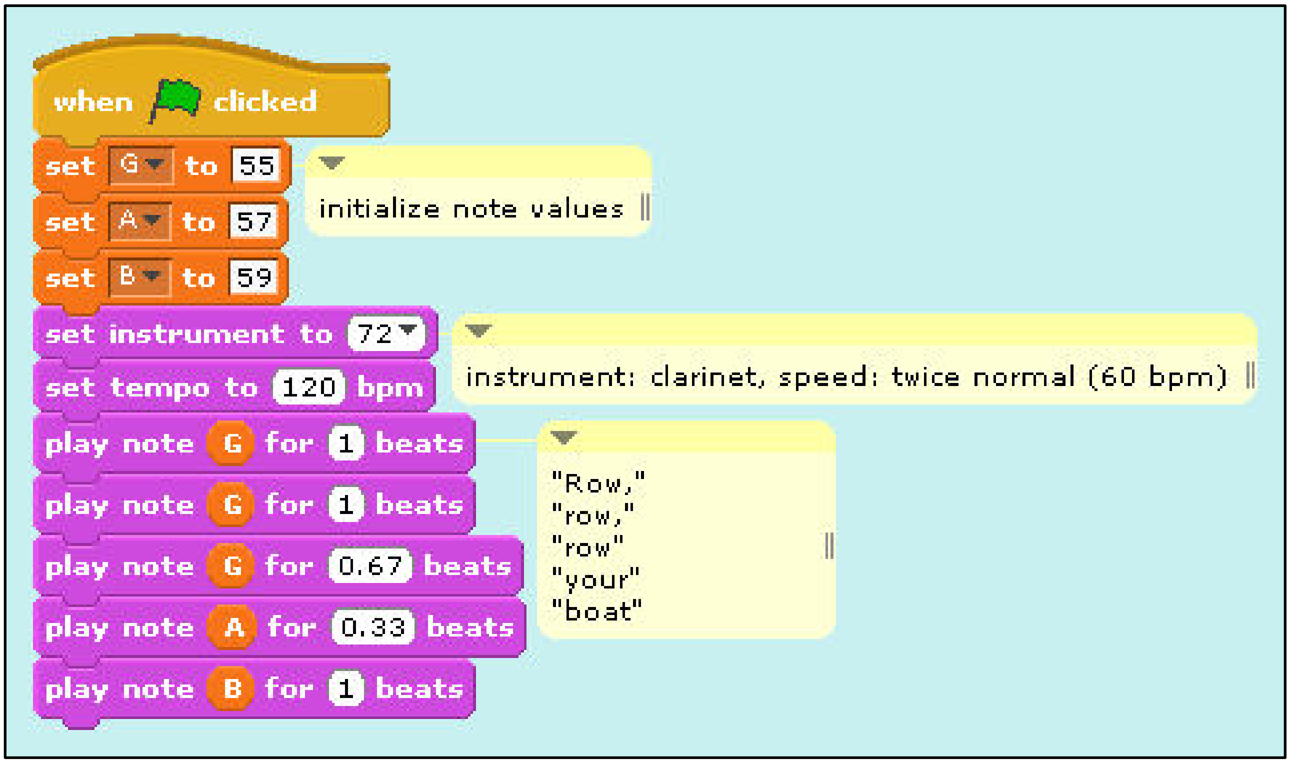


Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Progressive Music Examples

No. 2: Playing Notes Using Variables

Single Script



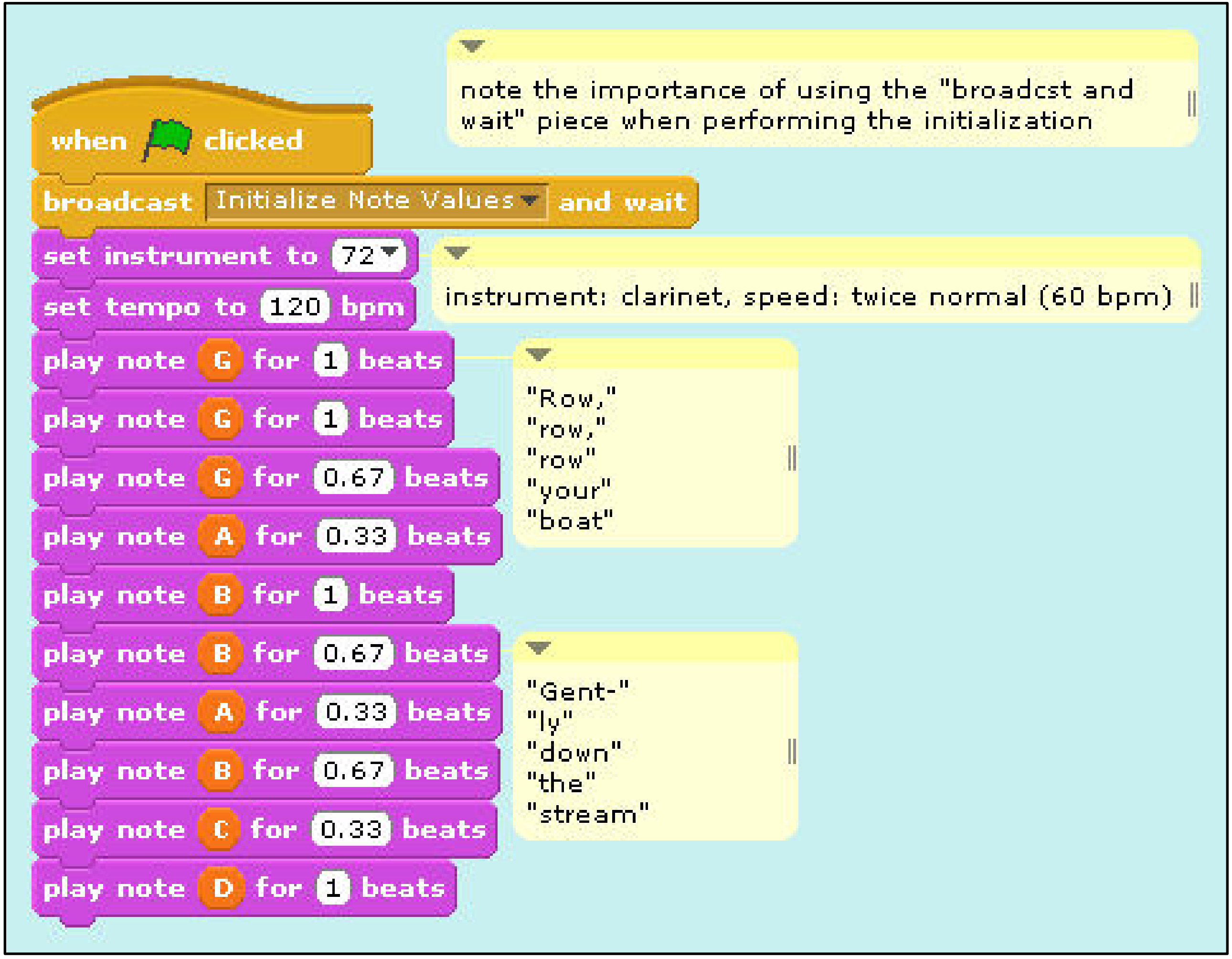
```
when ⚑ clicked
set G to 55                       initialize note values
set A to 57
set B to 59
set instrument to 72              instrument: clarinet, speed: twice normal (60 bpm)
set tempo to 120 bpm
play note G for 1 beats           "Row,"
play note G for 1 beats           "row,"
play note G for 0.67 beats        "row"
play note A for 0.33 beats        "your"
play note B for 1 beats           "boat"
```

Progressive Music Examples

No. 3: Separating Initialization

Two Scripts

(3a) **Main** Script



```
when [green flag] clicked
broadcast [Initialize Note Values] and wait
set instrument to (72)
set tempo to (120) bpm
play note (G) for (1) beats
play note (G) for (1) beats
play note (G) for (0.67) beats
play note (A) for (0.33) beats
play note (B) for (1) beats
play note (B) for (0.67) beats
play note (A) for (0.33) beats
play note (B) for (0.67) beats
play note (C) for (0.33) beats
play note (D) for (1) beats
```

note the importance of using the "broadcst and wait" piece when performing the initialization

instrument: clarinet, speed: twice normal (60 bpm)

"Row,"
"row,"
"row"
"your"
"boat"

"Gent-"
"ly"
"down"
"the"
"stream"

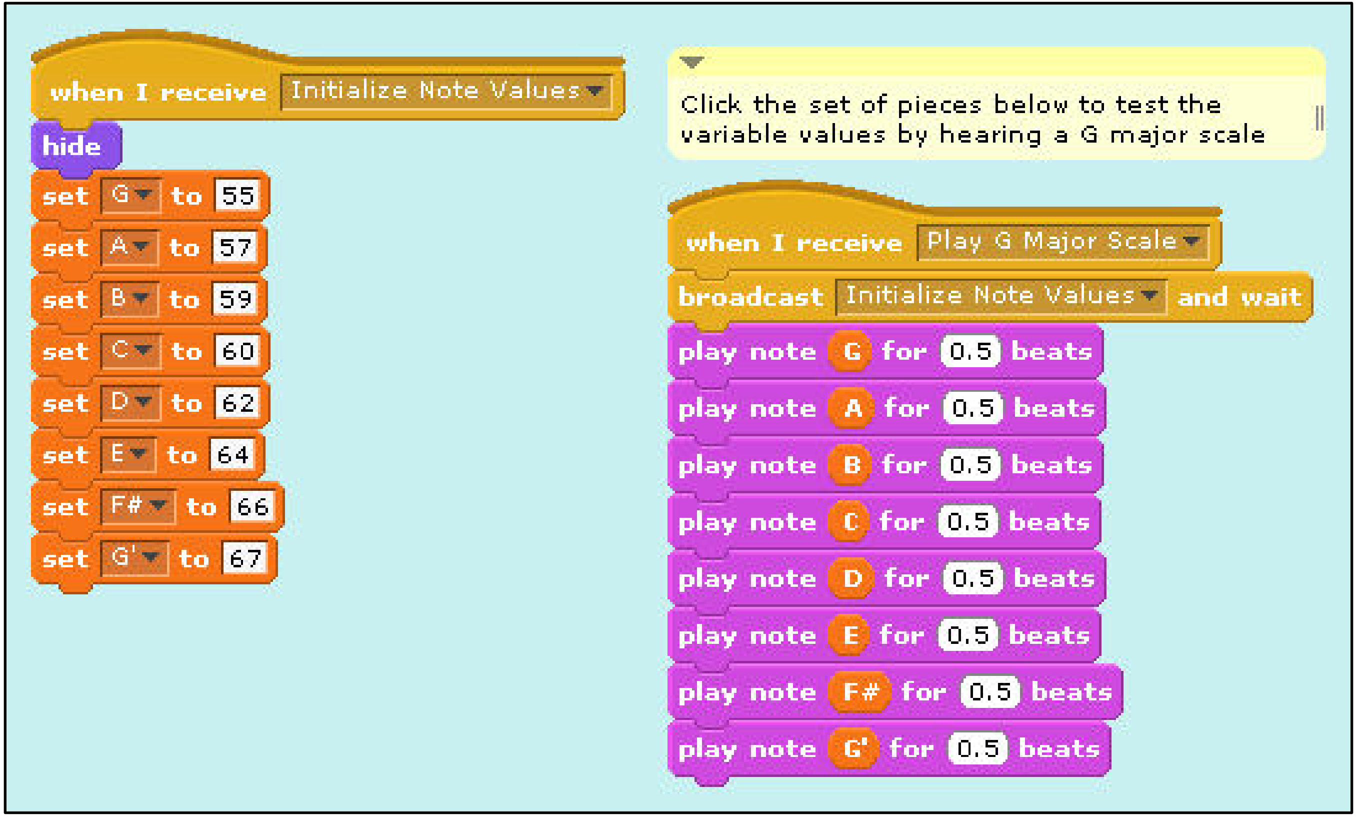
continued on next page

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.

Progressive Music Examples

No. 3: Separating Initialization (cont'd)

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



```
when I receive [Initialize Note Values ▾]
hide
set [G ▾] to 55
set [A ▾] to 57
set [B ▾] to 59
set [C ▾] to 60
set [D ▾] to 62
set [E ▾] to 64
set [F# ▾] to 66
set [G' ▾] to 67
```

Click the set of pieces below to test the variable values by hearing a G major scale

```
when I receive [Play G Major Scale ▾]
broadcast [Initialize Note Values ▾] and wait
play note (G) for 0.5 beats
play note (A) for 0.5 beats
play note (B) for 0.5 beats
play note (C) for 0.5 beats
play note (D) for 0.5 beats
play note (E) for 0.5 beats
play note (F#) for 0.5 beats
play note (G') for 0.5 beats
```

end of Example 3

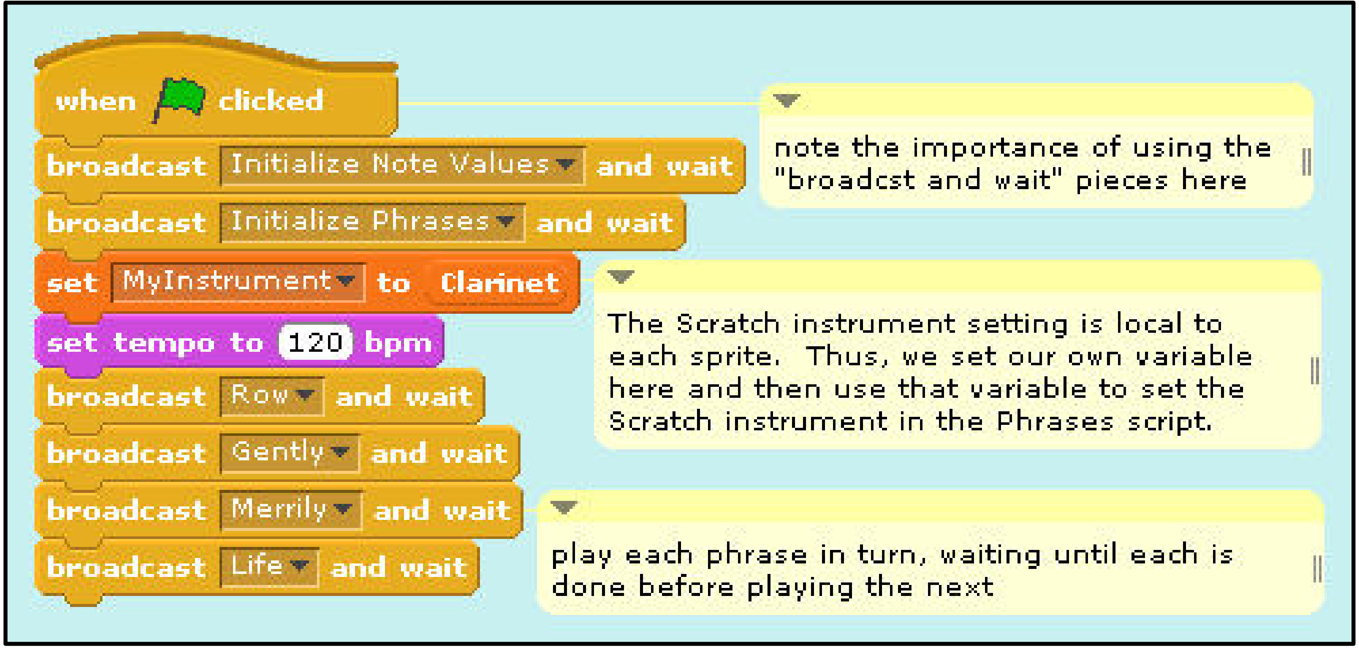
Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Progressive Music Examples

No. 4: Separating Phrases

Three Scripts

(4a) **Main** Script



```
when [flag] clicked
broadcast [Initialize Note Values] and wait
broadcast [Initialize Phrases] and wait
set [MyInstrument] to Clarinet
set tempo to (120) bpm
broadcast [Row] and wait
broadcast [Gently] and wait
broadcast [Merrily] and wait
broadcast [Life] and wait
```

note the importance of using the "broadcst and wait" pieces here

The Scratch instrument setting is local to each sprite. Thus, we set our own variable here and then use that variable to set the Scratch instrument in the Phrases script.

play each phrase in turn, waiting until each is done before playing the next

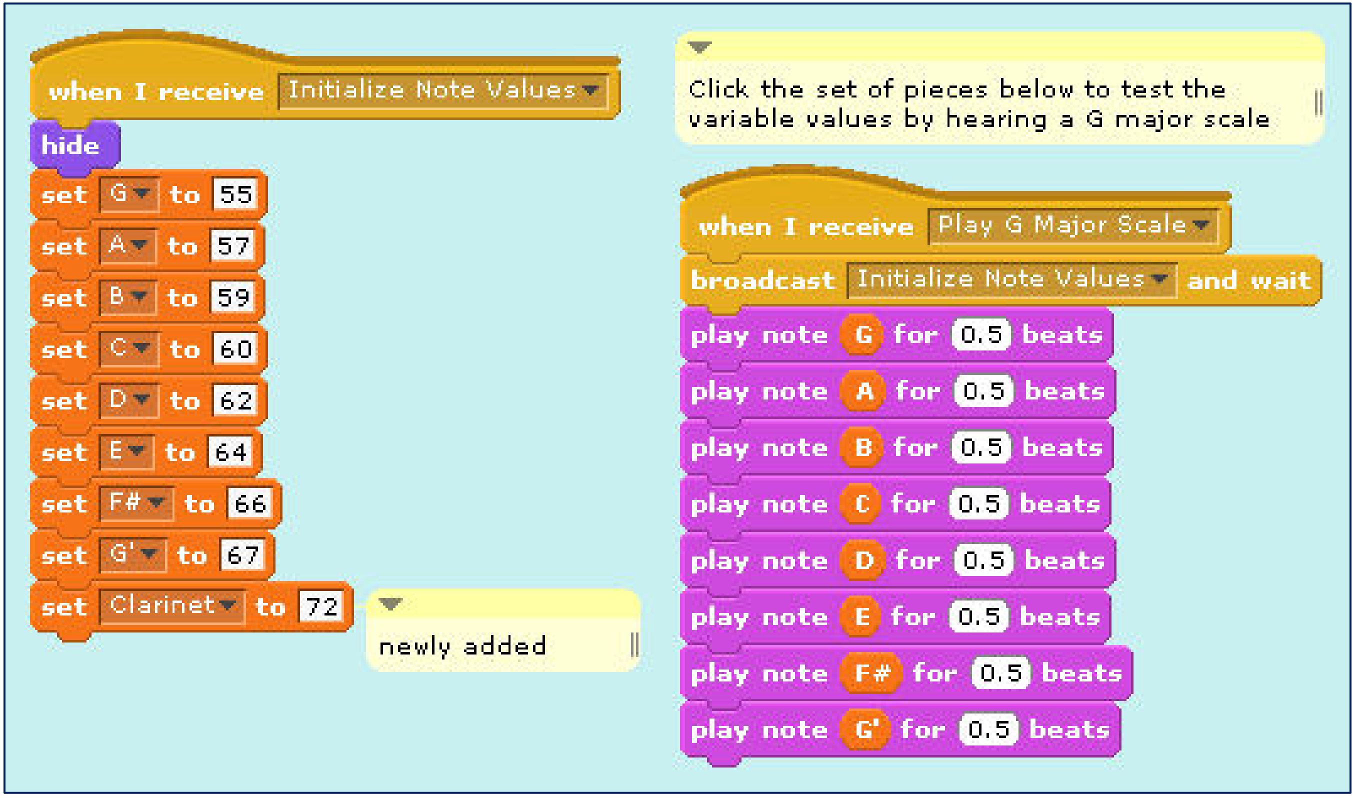
continued on next page

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.

Progressive Music Examples

No. 4: Separating Phrases (cont'd)

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



```
when I receive [Initialize Note Values]
hide
set [G] to 55
set [A] to 57
set [B] to 59
set [C] to 60
set [D] to 62
set [E] to 64
set [F#] to 66
set [G'] to 67
set [Clarinet] to 72        newly added
```

Click the set of pieces below to test the variable values by hearing a G major scale

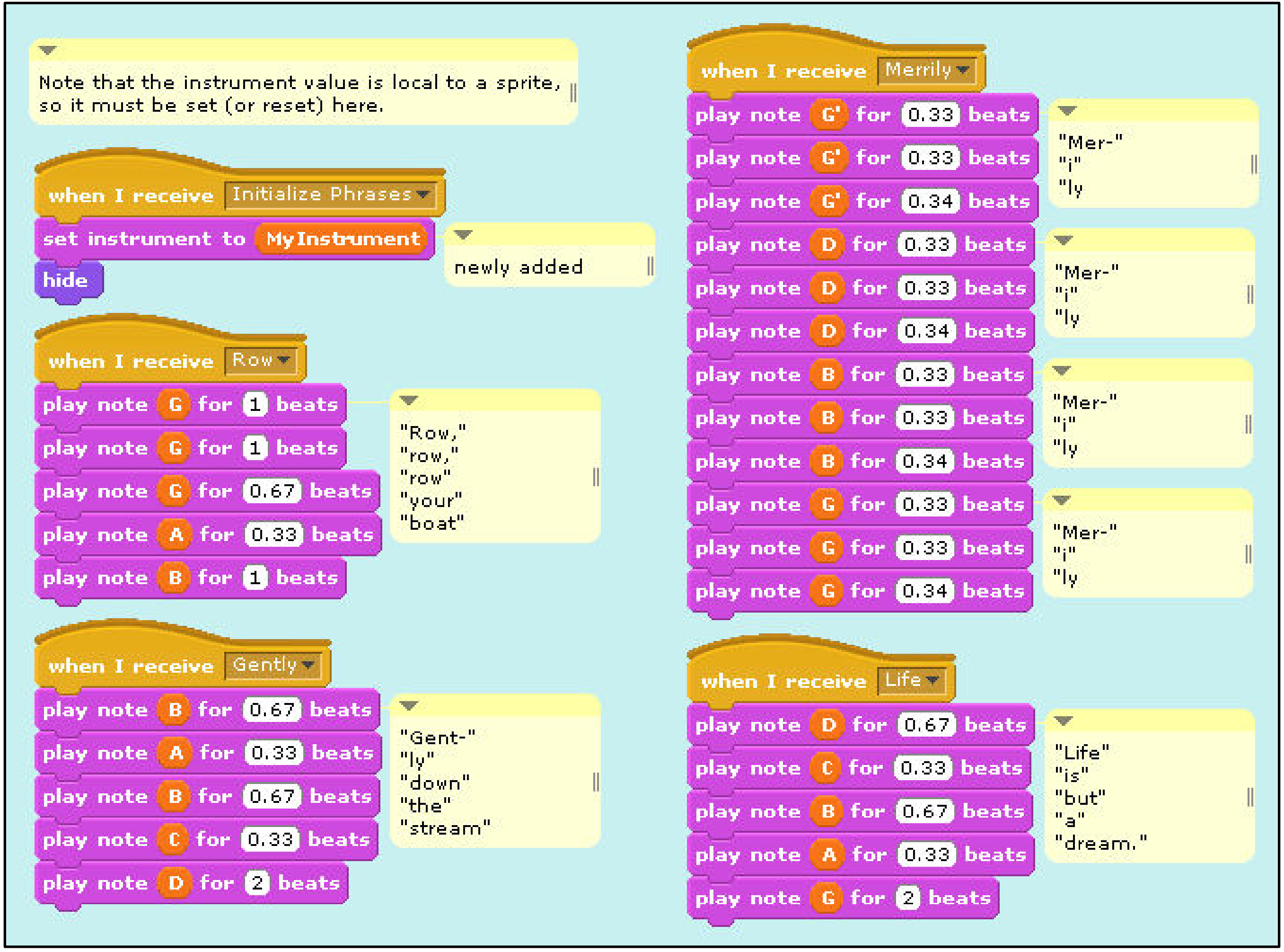
```
when I receive [Play G Major Scale]
broadcast [Initialize Note Values] and wait
play note (G) for 0.5 beats
play note (A) for 0.5 beats
play note (B) for 0.5 beats
play note (C) for 0.5 beats
play note (D) for 0.5 beats
play note (E) for 0.5 beats
play note (F#) for 0.5 beats
play note (G') for 0.5 beats
```

continued on next page

Progressive Music Examples

No. 4: Separating Phrases (cont'd)

(4c) Phrases Script



```
Note that the instrument value is local to a sprite,
so it must be set (or reset) here.

when I receive [Initialize Phrases]
set instrument to (MyInstrument)      newly added
hide

when I receive [Row]
play note (G) for (1) beats           "Row,"
play note (G) for (1) beats           "row,"
play note (G) for (0.67) beats        "row"
play note (A) for (0.33) beats        "your"
play note (B) for (1) beats           "boat"

when I receive [Gently]
play note (B) for (0.67) beats        "Gent-"
play note (A) for (0.33) beats        "ly"
play note (B) for (0.67) beats        "down"
play note (C) for (0.33) beats        "the"
play note (D) for (2) beats           "stream"

when I receive [Merrily]
play note (G') for (0.33) beats       "Mer-"
play note (G') for (0.33) beats       "i"
play note (G') for (0.34) beats       "ly
play note (D) for (0.33) beats        "Mer-"
play note (D) for (0.33) beats        "i"
play note (D) for (0.34) beats        "ly
play note (B) for (0.33) beats        "Mer-"
play note (B) for (0.33) beats        "i"
play note (B) for (0.34) beats        "ly
play note (G) for (0.33) beats        "Mer-"
play note (G) for (0.33) beats        "i"
play note (G) for (0.34) beats        "ly

when I receive [Life]
play note (D) for (0.67) beats        "Life"
play note (C) for (0.33) beats        "is"
play note (B) for (0.67) beats        "but"
play note (A) for (0.33) beats        "a"
play note (G) for (2) beats           "dream."
```

end of Example 4

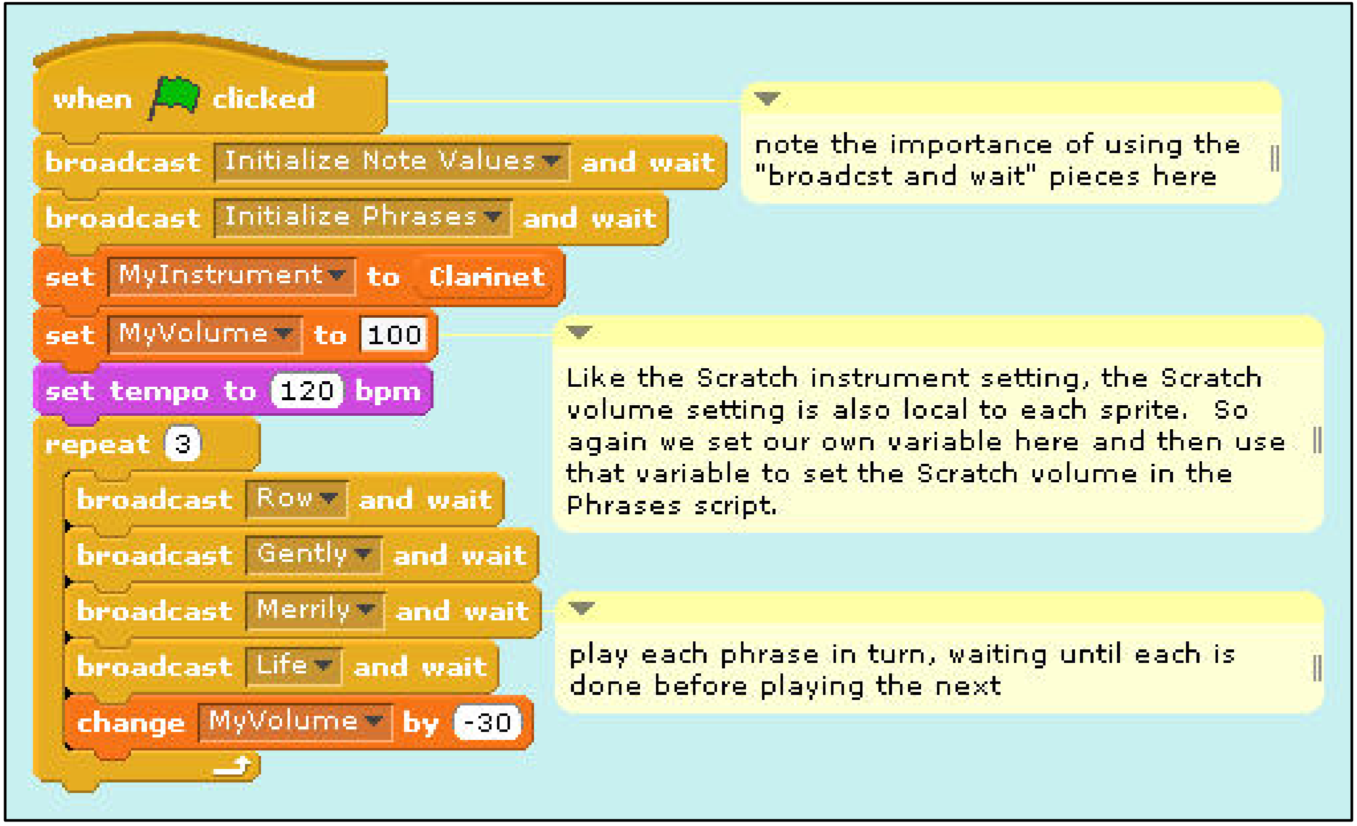
Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab.
See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass
Lowell funded by the National Science Foundation. See http://performamatics.org.

Progressive Music Examples

No. 5: Looping and Fading

Three Scripts

(5a) **Main** Script



```
when [flag] clicked
broadcast [Initialize Note Values] and wait
broadcast [Initialize Phrases] and wait
set [MyInstrument] to Clarinet
set [MyVolume] to 100
set tempo to 120 bpm
repeat 3
    broadcast [Row] and wait
    broadcast [Gently] and wait
    broadcast [Merrily] and wait
    broadcast [Life] and wait
    change [MyVolume] by -30
```

note the importance of using the "broadcst and wait" pieces here

Like the Scratch instrument setting, the Scratch volume setting is also local to each sprite. So again we set our own variable here and then use that variable to set the Scratch volume in the Phrases script.

play each phrase in turn, waiting until each is done before playing the next

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

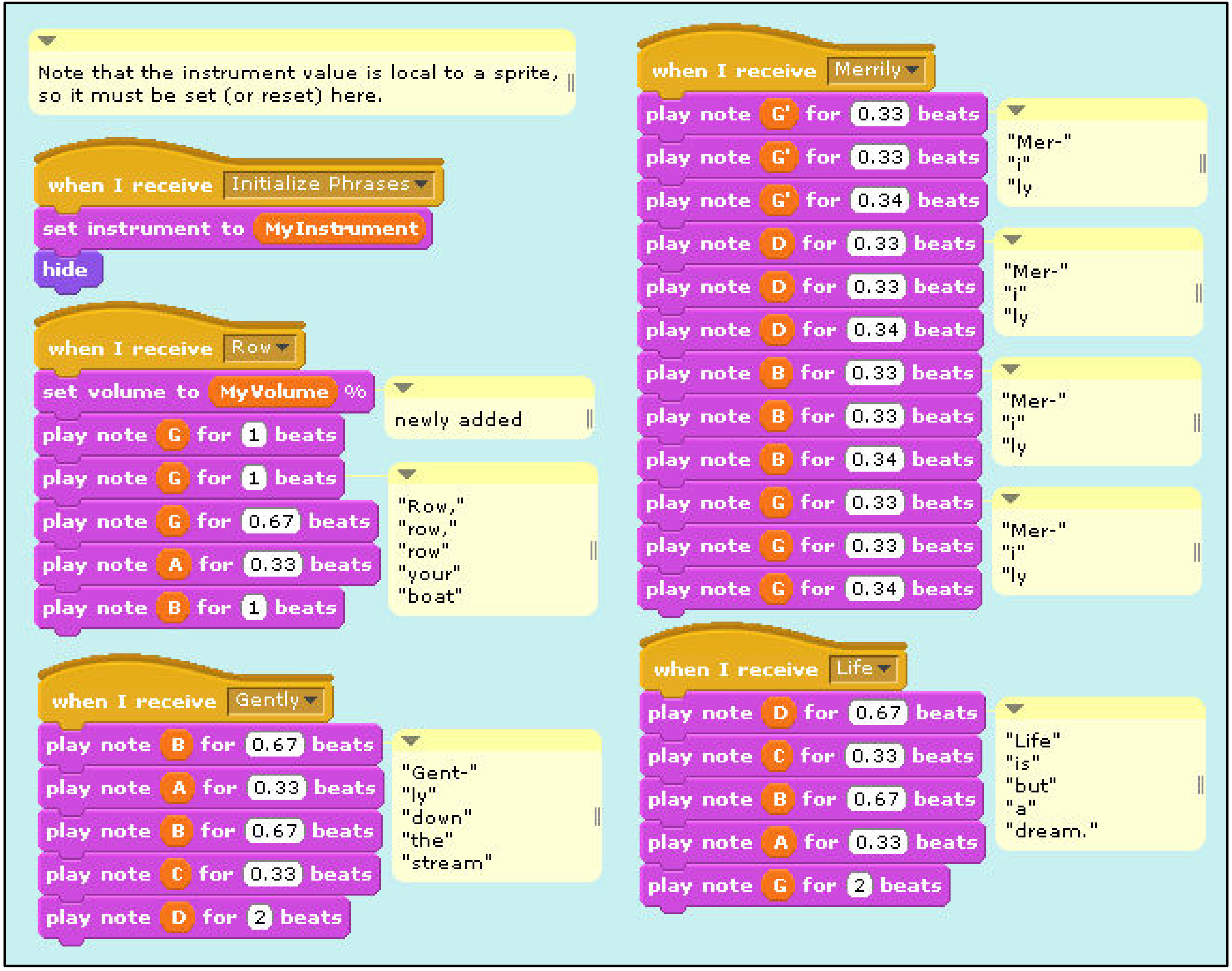
(*same as on page 12*)

continued on next page

Progressive Music Examples

No. 5: Looping and Fading (cont'd)

(5c) **Phrases** Script



```
Note that the instrument value is local to a sprite,
so it must be set (or reset) here.

when I receive [Initialize Phrases]
set instrument to (MyInstrument)
hide

when I receive [Row]
set volume to (MyVolume) %          newly added
play note (G) for (1) beats
play note (G) for (1) beats          "Row,"
play note (G) for (0.67) beats       "row,"
play note (A) for (0.33) beats       "row"
play note (B) for (1) beats          "your"
                                     "boat"

when I receive [Gently]
play note (B) for (0.67) beats       "Gent-"
play note (A) for (0.33) beats       "ly"
play note (B) for (0.67) beats       "down"
play note (C) for (0.33) beats       "the"
play note (D) for (2) beats          "stream"

when I receive [Merrily]
play note (G') for (0.33) beats      "Mer-"
play note (G') for (0.33) beats      "i"
play note (G') for (0.34) beats      "ly
play note (D) for (0.33) beats       "Mer-"
play note (D) for (0.33) beats       "i"
play note (D) for (0.34) beats       "ly
play note (B) for (0.33) beats       "Mer-"
play note (B) for (0.33) beats       "i"
play note (B) for (0.34) beats       "ly
play note (G) for (0.33) beats       "Mer-"
play note (G) for (0.33) beats       "i"
play note (G) for (0.34) beats       "ly

when I receive [Life]
play note (D) for (0.67) beats       "Life"
play note (C) for (0.33) beats       "is"
play note (B) for (0.67) beats       "but"
play note (A) for (0.33) beats       "a"
play note (G) for (2) beats          "dream."
```

end of Example 5

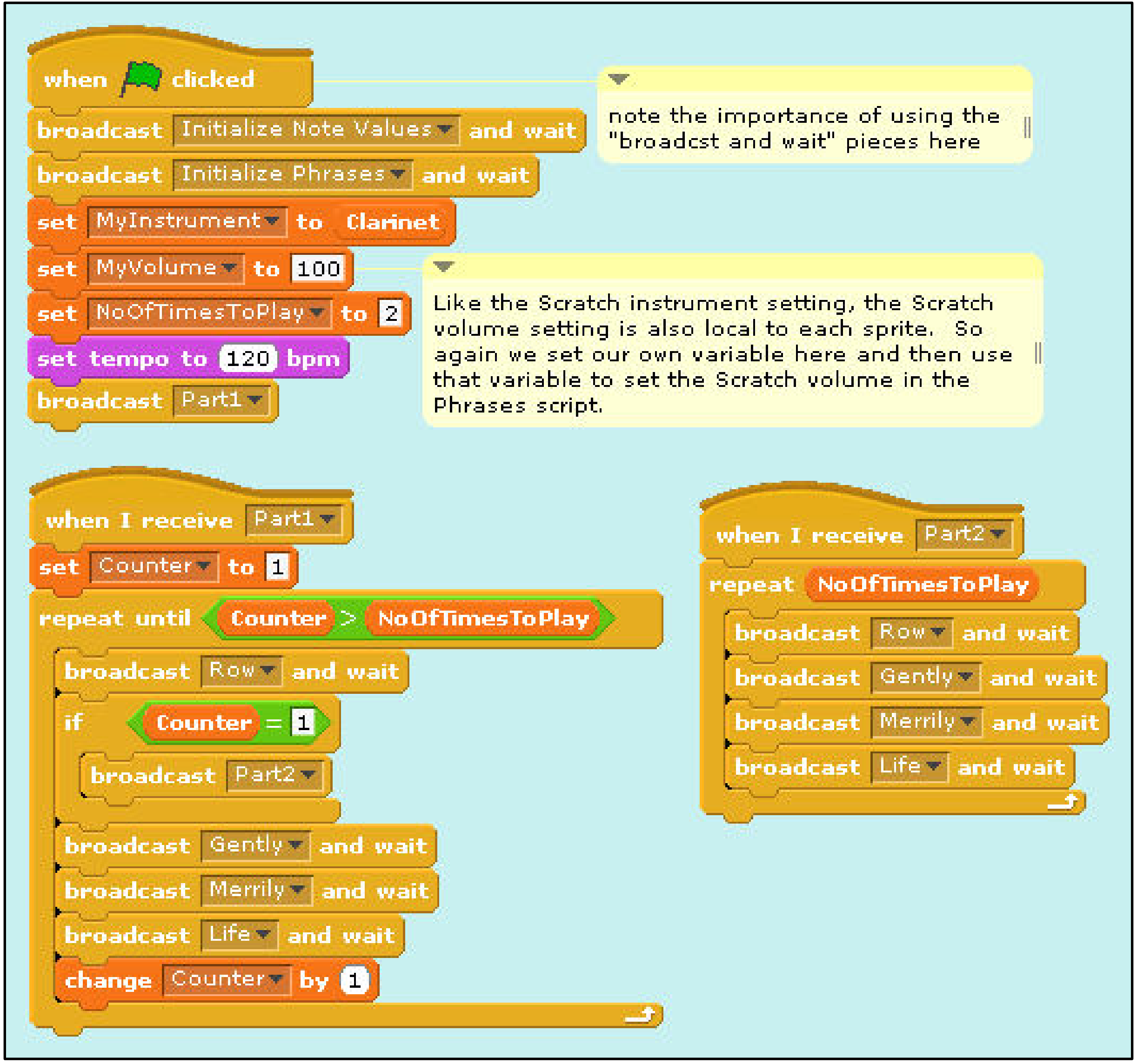
Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab.
See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass
Lowell funded by the National Science Foundation. See http://performamatics.org.

Progressive Music Examples

No. 6: Playing a Round with One Instrument

Three Scripts

(6a) **Main** Script



```
when ⚑ clicked
broadcast Initialize Note Values ▾ and wait
broadcast Initialize Phrases ▾ and wait
set MyInstrument ▾ to Clarinet
set MyVolume ▾ to 100
set NoOfTimesToPlay ▾ to 2
set tempo to 120 bpm
broadcast Part1 ▾
```

note the importance of using the "broadcst and wait" pieces here

Like the Scratch instrument setting, the Scratch volume setting is also local to each sprite. So again we set our own variable here and then use that variable to set the Scratch volume in the Phrases script.

```
when I receive Part1 ▾
set Counter ▾ to 1
repeat until < Counter > NoOfTimesToPlay >
    broadcast Row ▾ and wait
    if < Counter = 1 >
        broadcast Part2 ▾
    broadcast Gently ▾ and wait
    broadcast Merrily ▾ and wait
    broadcast Life ▾ and wait
    change Counter ▾ by 1
```

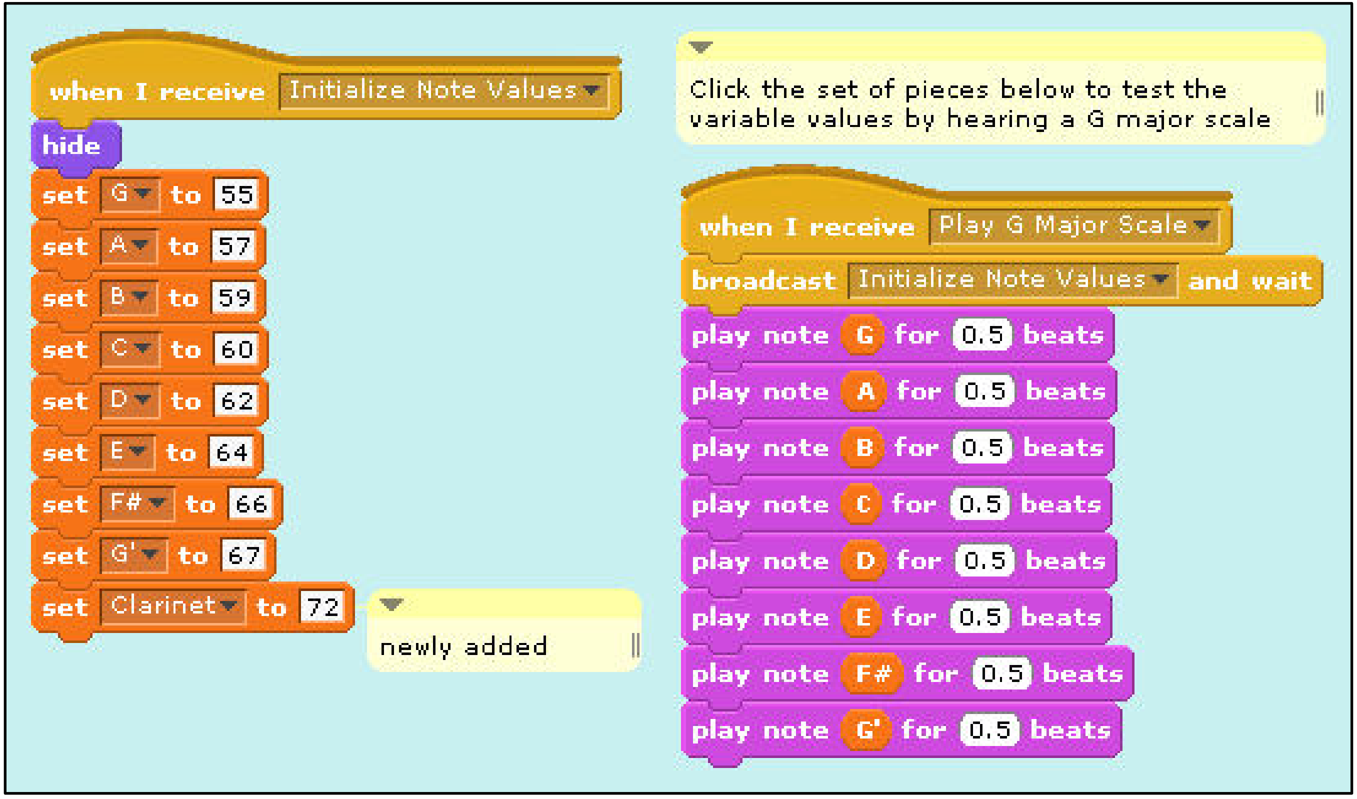
```
when I receive Part2 ▾
repeat NoOfTimesToPlay
    broadcast Row ▾ and wait
    broadcast Gently ▾ and wait
    broadcast Merrily ▾ and wait
    broadcast Life ▾ and wait
```

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.

Progressive Music Examples

No. 6: Playing a Round with One Instrument (cont'd)

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



```
when I receive [Initialize Note Values ▾]
hide
set [G ▾] to 55
set [A ▾] to 57
set [B ▾] to 59
set [C ▾] to 60
set [D ▾] to 62
set [E ▾] to 64
set [F# ▾] to 66
set [G' ▾] to 67
set [Clarinet ▾] to 72
```

newly added

Click the set of pieces below to test the variable values by hearing a G major scale

```
when I receive [Play G Major Scale ▾]
broadcast [Initialize Note Values ▾] and wait
play note (G) for 0.5 beats
play note (A) for 0.5 beats
play note (B) for 0.5 beats
play note (C) for 0.5 beats
play note (D) for 0.5 beats
play note (E) for 0.5 beats
play note (F#) for 0.5 beats
play note (G') for 0.5 beats
```

(6c) **Phrases** Script

(*same as on page 16*)

end of Example 6

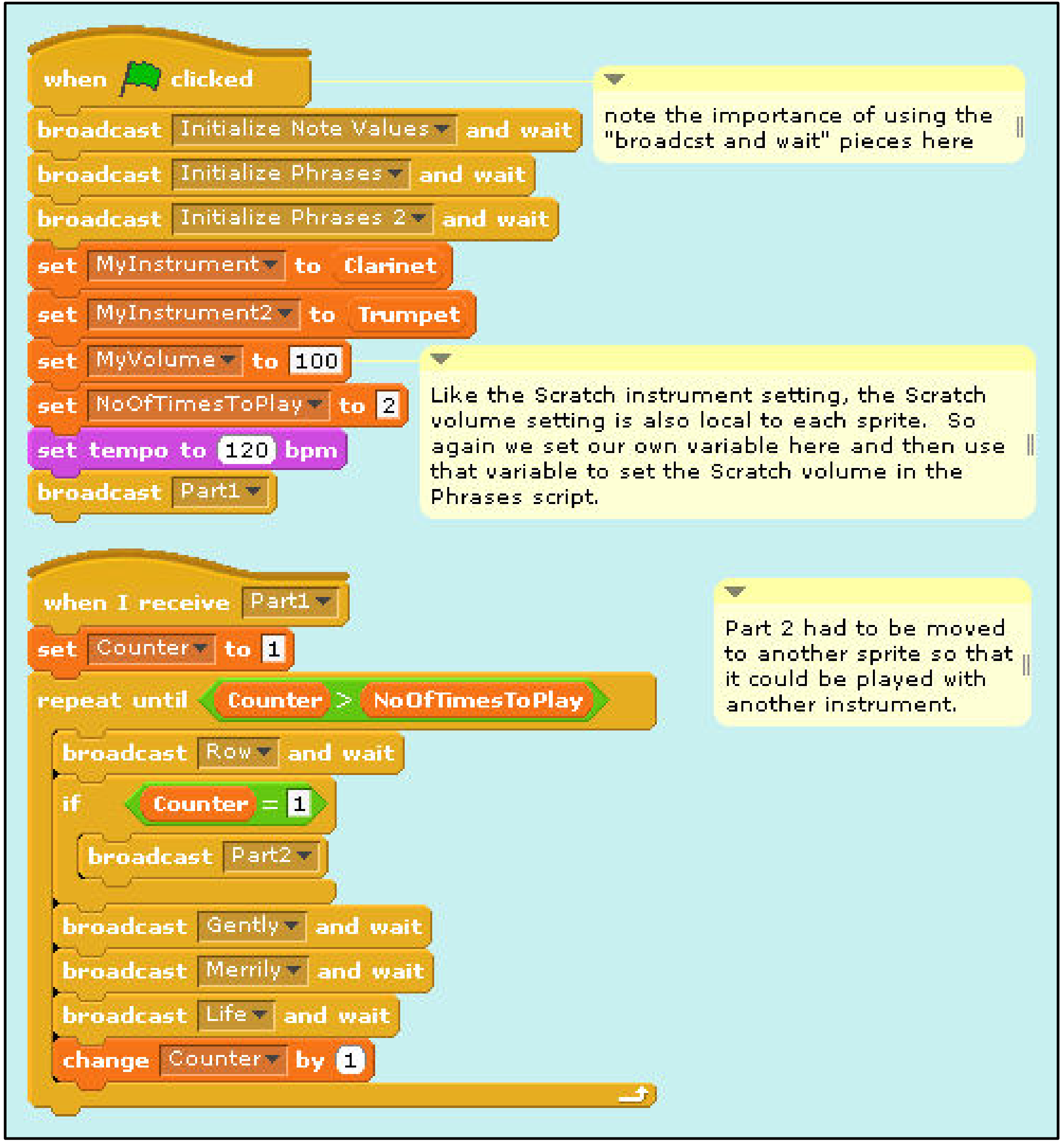
Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu. Performamatics is an interdisciplinary project at UMass Lowell funded by the National Science Foundation. See http://performamatics.org.

Progressive Music Examples

No. 7: Playing a Round with Two Instruments

Five Scripts

(7a) **Main** Script



```
when ⚑ clicked
broadcast [Initialize Note Values ▾] and wait
broadcast [Initialize Phrases ▾] and wait
broadcast [Initialize Phrases 2 ▾] and wait
set [MyInstrument ▾] to (Clarinet)
set [MyInstrument2 ▾] to (Trumpet)
set [MyVolume ▾] to (100)
set [NoOfTimesToPlay ▾] to (2)
set tempo to (120) bpm
broadcast [Part1 ▾]

when I receive [Part1 ▾]
set [Counter ▾] to (1)
repeat until <(Counter) > (NoOfTimesToPlay)>
    broadcast [Row ▾] and wait
    if <(Counter) = (1)>
        broadcast [Part2 ▾]
    broadcast [Gently ▾] and wait
    broadcast [Merrily ▾] and wait
    broadcast [Life ▾] and wait
    change [Counter ▾] by (1)
```

note the importance of using the "broadcst and wait" pieces here

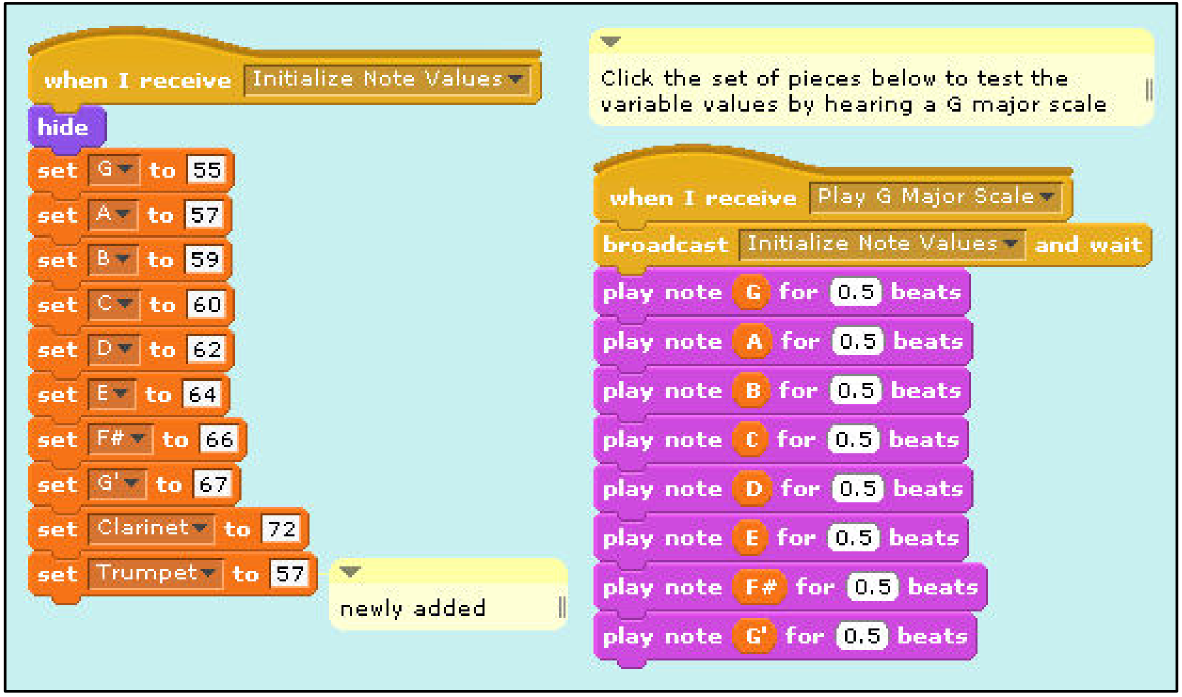
Like the Scratch instrument setting, the Scratch volume setting is also local to each sprite. So again we set our own variable here and then use that variable to set the Scratch volume in the Phrases script.

Part 2 had to be moved to another sprite so that it could be played with another instrument.

Progressive Music Examples

No. 7: Playing a Round with Two Instruments (cont'd)

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



```
when I receive [Initialize Note Values]
hide
set G to 55
set A to 57
set B to 59
set C to 60
set D to 62
set E to 64
set F# to 66
set G' to 67
set Clarinet to 72
set Trumpet to 57        newly added
```

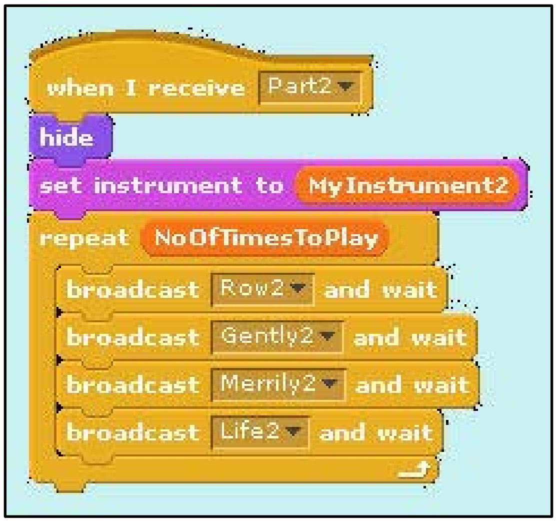
```
Click the set of pieces below to test the
variable values by hearing a G major scale

when I receive [Play G Major Scale]
broadcast [Initialize Note Values] and wait
play note (G) for (0.5) beats
play note (A) for (0.5) beats
play note (B) for (0.5) beats
play note (C) for (0.5) beats
play note (D) for (0.5) beats
play note (E) for (0.5) beats
play note (F#) for (0.5) beats
play note (G') for (0.5) beats
```

(7c) **Phrases** Script

(*same as on page 16*)

(7d) **Part2** Script ➔



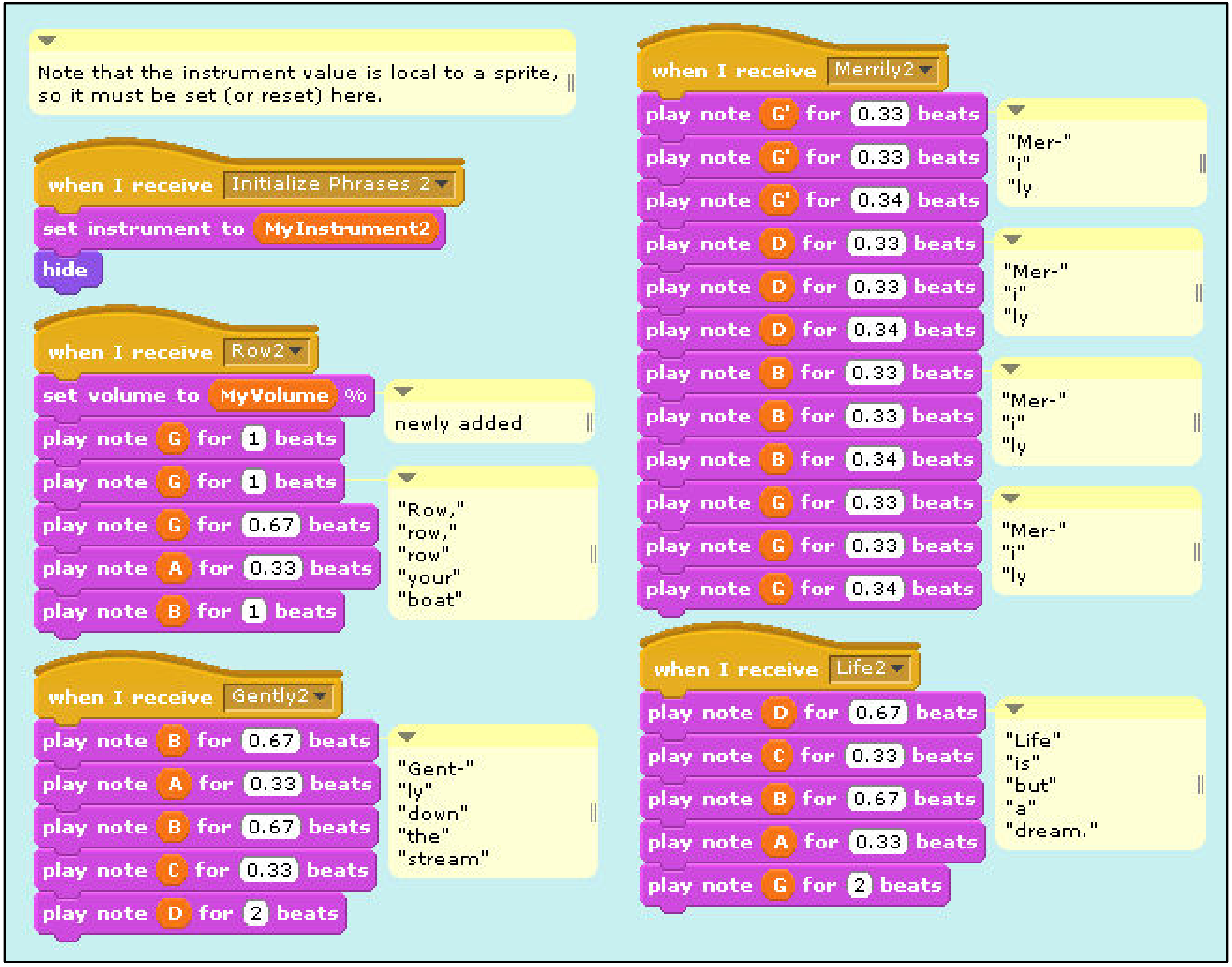
```
when I receive [Part2]
hide
set instrument to (MyInstrument2)
repeat (NoOfTimesToPlay)
    broadcast [Row2] and wait
    broadcast [Gently2] and wait
    broadcast [Merrily2] and wait
    broadcast [Life2] and wait
```

continued on next page

Progressive Music Examples

No. 7: Playing a Round with Two Instruments (cont'd)

(7e) **Instrument2 ("Instru2")** Script



```
Note that the instrument value is local to a sprite,
so it must be set (or reset) here.

when I receive [Initialize Phrases 2]
set instrument to (MyInstrument2)
hide

when I receive [Row2]
set volume to (MyVolume) %          newly added
play note (G) for (1) beats
play note (G) for (1) beats         "Row,"
play note (G) for (0.67) beats      "row,"
play note (A) for (0.33) beats      "row"
play note (B) for (1) beats         "your"
                                    "boat"

when I receive [Gently2]
play note (B) for (0.67) beats      "Gent-"
play note (A) for (0.33) beats      "ly"
play note (B) for (0.67) beats      "down"
play note (C) for (0.33) beats      "the"
play note (D) for (2) beats         "stream"

when I receive [Merrily2]
play note (G') for (0.33) beats     "Mer-"
play note (G') for (0.33) beats     "i"
play note (G') for (0.34) beats     "ly
play note (D) for (0.33) beats      "Mer-"
play note (D) for (0.33) beats      "i"
play note (D) for (0.34) beats      "ly
play note (B) for (0.33) beats      "Mer-"
play note (B) for (0.33) beats      "i"
play note (B) for (0.34) beats      "ly
play note (G) for (0.33) beats      "Mer-"
play note (G) for (0.33) beats      "i"
play note (G) for (0.34) beats      "ly

when I receive [Life2]
play note (D) for (0.67) beats      "Life"
play note (C) for (0.33) beats      "is"
play note (B) for (0.67) beats      "but"
play note (A) for (0.33) beats      "a"
play note (G) for (2) beats         "dream."
```

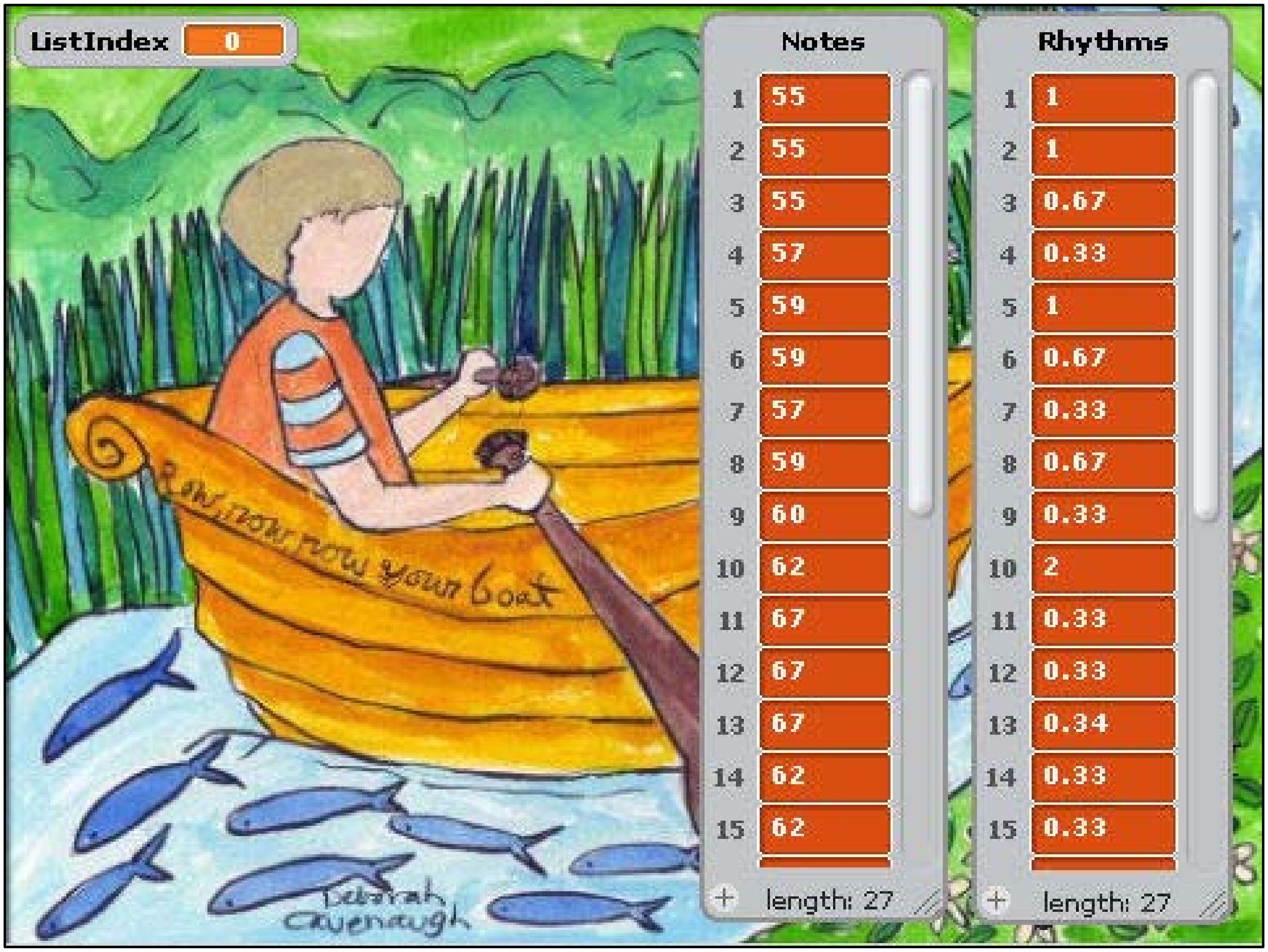
end of Example 7

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.

Progressive Music Examples

No. 8: Storing Notes and Rhythms in Lists

Output Window



ListIndex 0

	Notes	Rhythms
1	55	1
2	55	1
3	55	0.67
4	57	0.33
5	59	1
6	59	0.67
7	57	0.33
8	59	0.67
9	60	0.33
10	62	2
11	67	0.33
12	67	0.33
13	67	0.34
14	62	0.33
15	62	0.33

Notes: length: 27

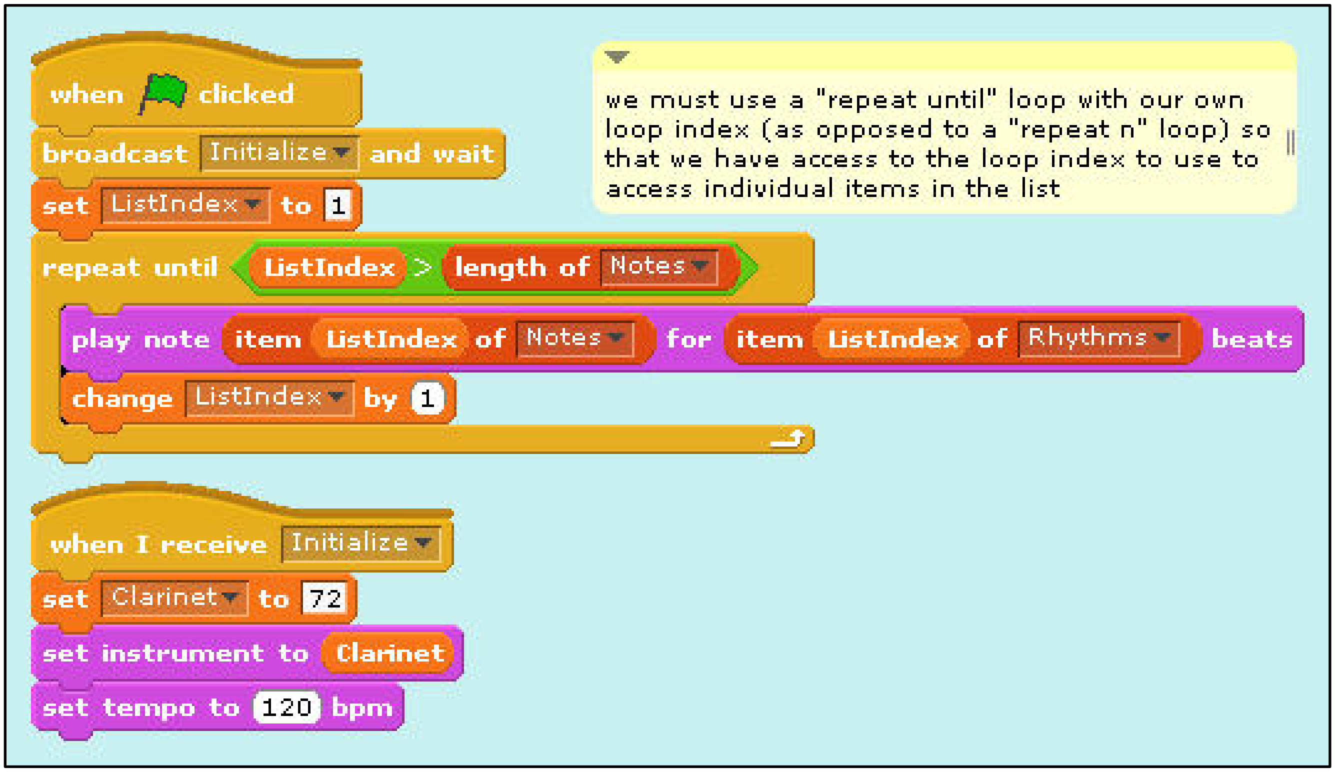
Rhythms: length: 27

continued on next page

Progressive Music Examples

No. 8: Storing Notes and Rhythms in Lists (cont'd)

Single Script



```
when ⚑ clicked
broadcast [Initialize] and wait
set [ListIndex] to 1
repeat until ((ListIndex) > (length of [Notes]))
    play note (item (ListIndex) of [Notes]) for (item (ListIndex) of [Rhythms]) beats
    change [ListIndex] by 1

when I receive [Initialize]
set [Clarinet] to 72
set instrument to (Clarinet)
set tempo to 120 bpm
```

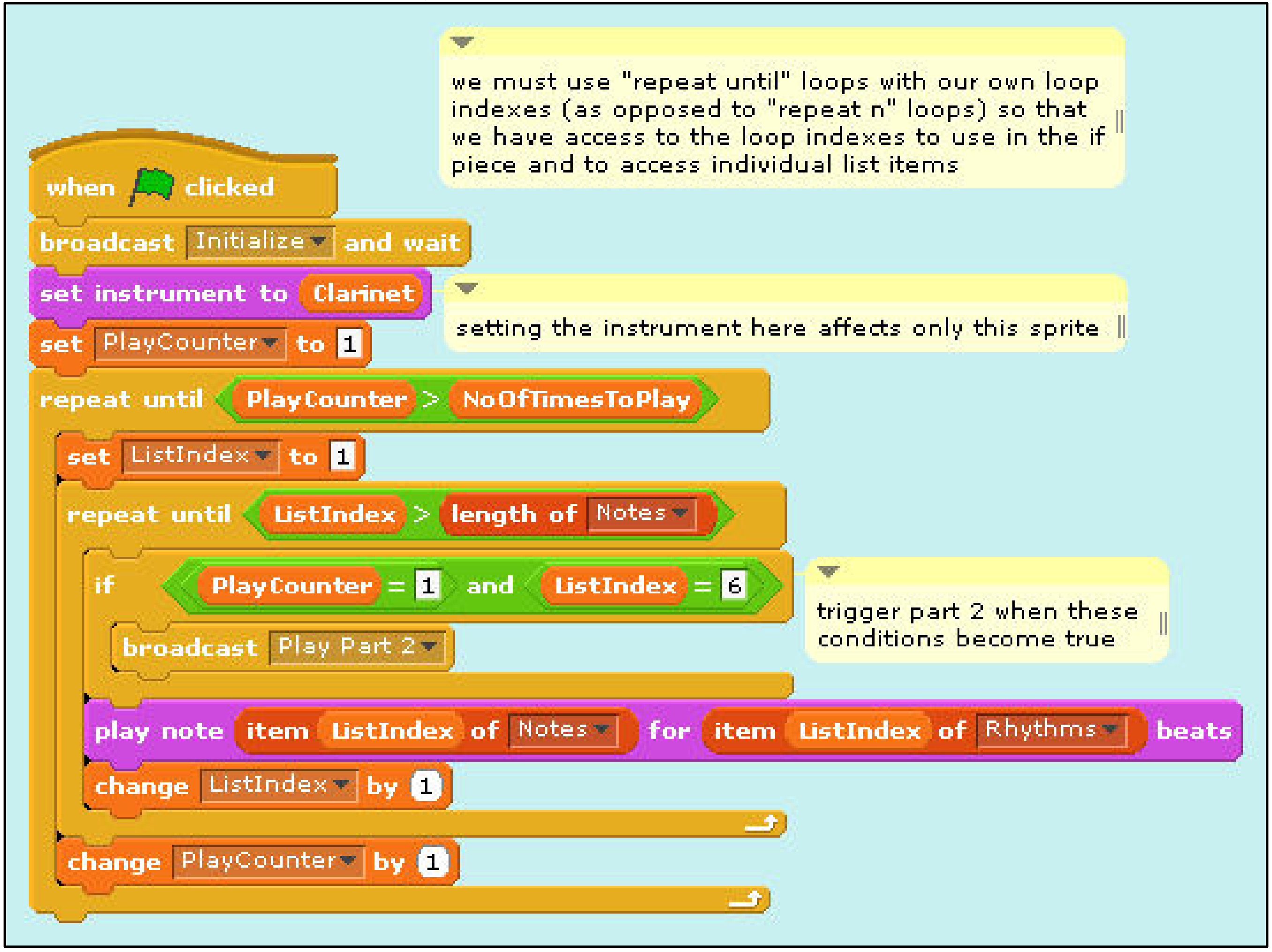
we must use a "repeat until" loop with our own loop index (as opposed to a "repeat n" loop) so that we have access to the loop index to use to access individual items in the list

end of Example 8

No. 9: Playing a Round Using Lists

Three Scripts

(9a) **Main** Script



we must use "repeat until" loops with our own loop indexes (as opposed to "repeat n" loops) so that we have access to the loop indexes to use in the if piece and to access individual list items

```
when 🏁 clicked
broadcast Initialize and wait
set instrument to Clarinet
set PlayCounter to 1
repeat until PlayCounter > NoOfTimesToPlay
    set ListIndex to 1
    repeat until ListIndex > length of Notes
        if PlayCounter = 1 and ListIndex = 6
            broadcast Play Part 2
        play note item ListIndex of Notes for item ListIndex of Rhythms beats
        change ListIndex by 1
    change PlayCounter by 1
```

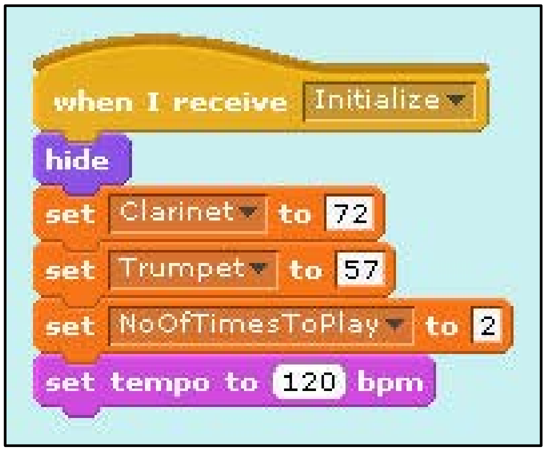
setting the instrument here affects only this sprite

trigger part 2 when these conditions become true

continued on next page

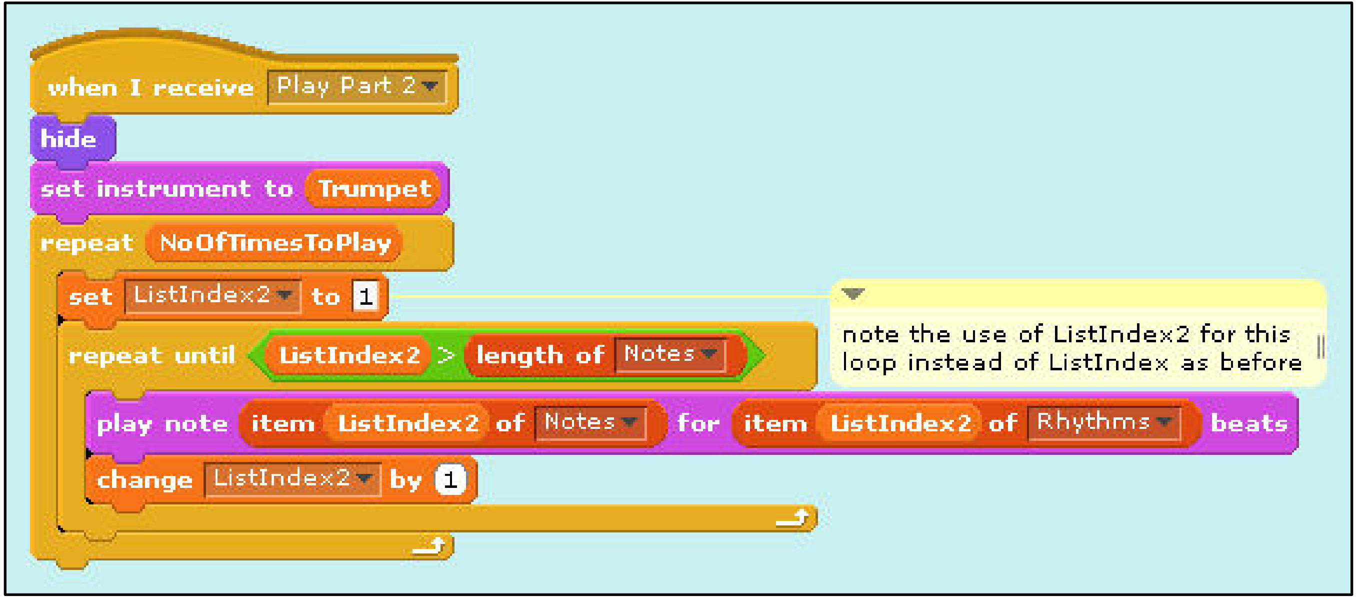
No. 9: Playing a Round Using Lists (cont'd)

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



```
when I receive [Initialize ▾]
hide
set [Clarinet ▾] to [72]
set [Trumpet ▾] to [57]
set [NoOfTimesToPlay ▾] to [2]
set tempo to (120) bpm
```

(9c) **Part2** Script



```
when I receive [Play Part 2 ▾]
hide
set instrument to (Trumpet)
repeat (NoOfTimesToPlay)
    set [ListIndex2 ▾] to [1]
    repeat until <(ListIndex2) > (length of [Notes ▾])>
        play note (item (ListIndex2) of [Notes ▾]) for (item (ListIndex2) of [Rhythms ▾]) beats
        change [ListIndex2 ▾] by (1)
```

note the use of ListIndex2 for this loop instead of ListIndex as before

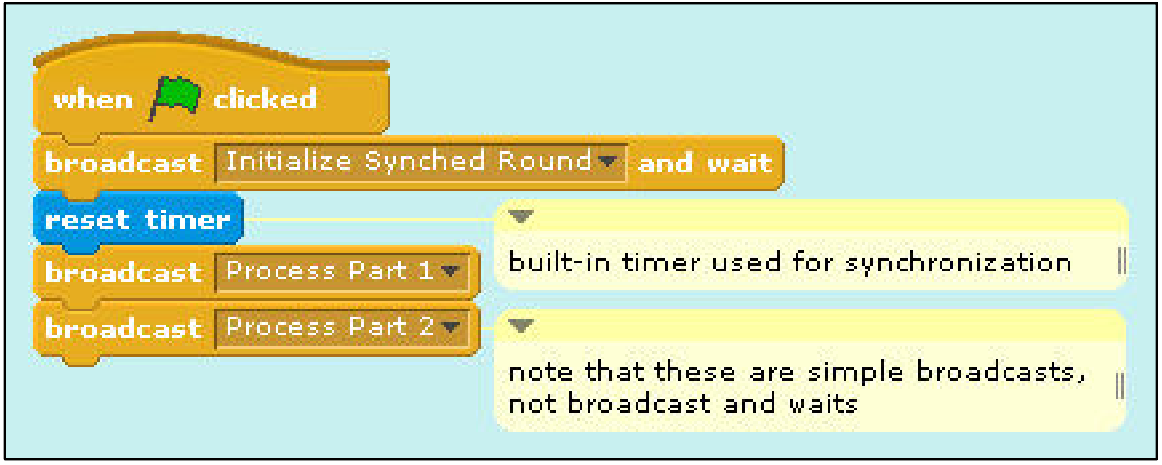
end of Example 9

Progressive Music Examples

No. 10: Synchronizing Play from Lists

Four Scripts

(10a) **Main** Script

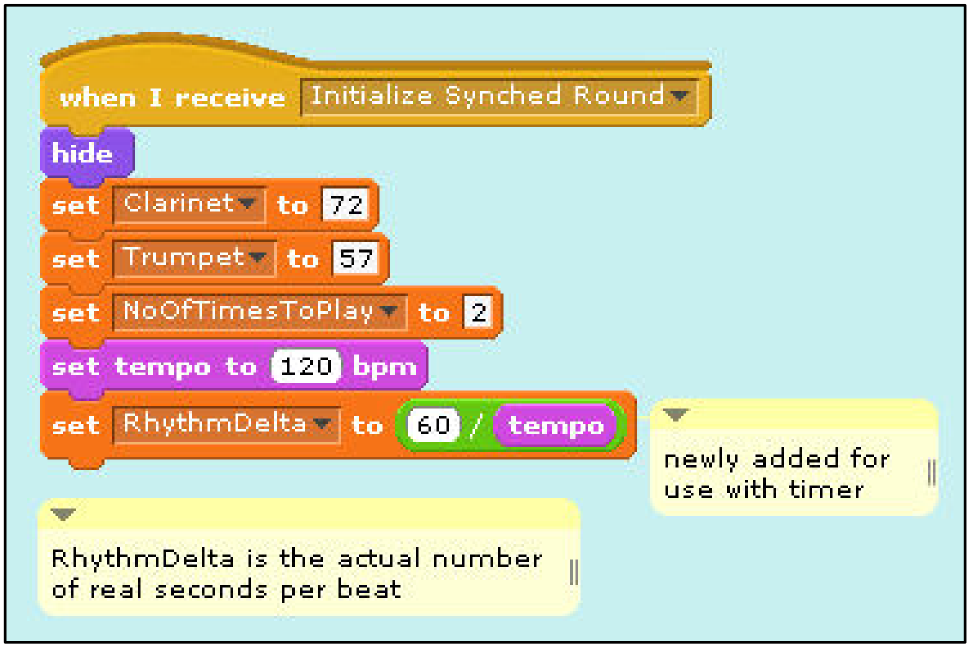


```
when (flag) clicked
broadcast [Initialize Synched Round ▾] and wait
reset timer
broadcast [Process Part 1 ▾]
broadcast [Process Part 2 ▾]
```

built-in timer used for synchronization

note that these are simple broadcasts, not broadcast and waits

(10b) **Initialization (“Init”)** Script



```
when I receive [Initialize Synched Round ▾]
hide
set [Clarinet ▾] to 72
set [Trumpet ▾] to 57
set [NoOfTimesToPlay ▾] to 2
set tempo to 120 bpm
set [RhythmDelta ▾] to (60 / tempo)
```

newly added for use with timer

RhythmDelta is the actual number of real seconds per beat

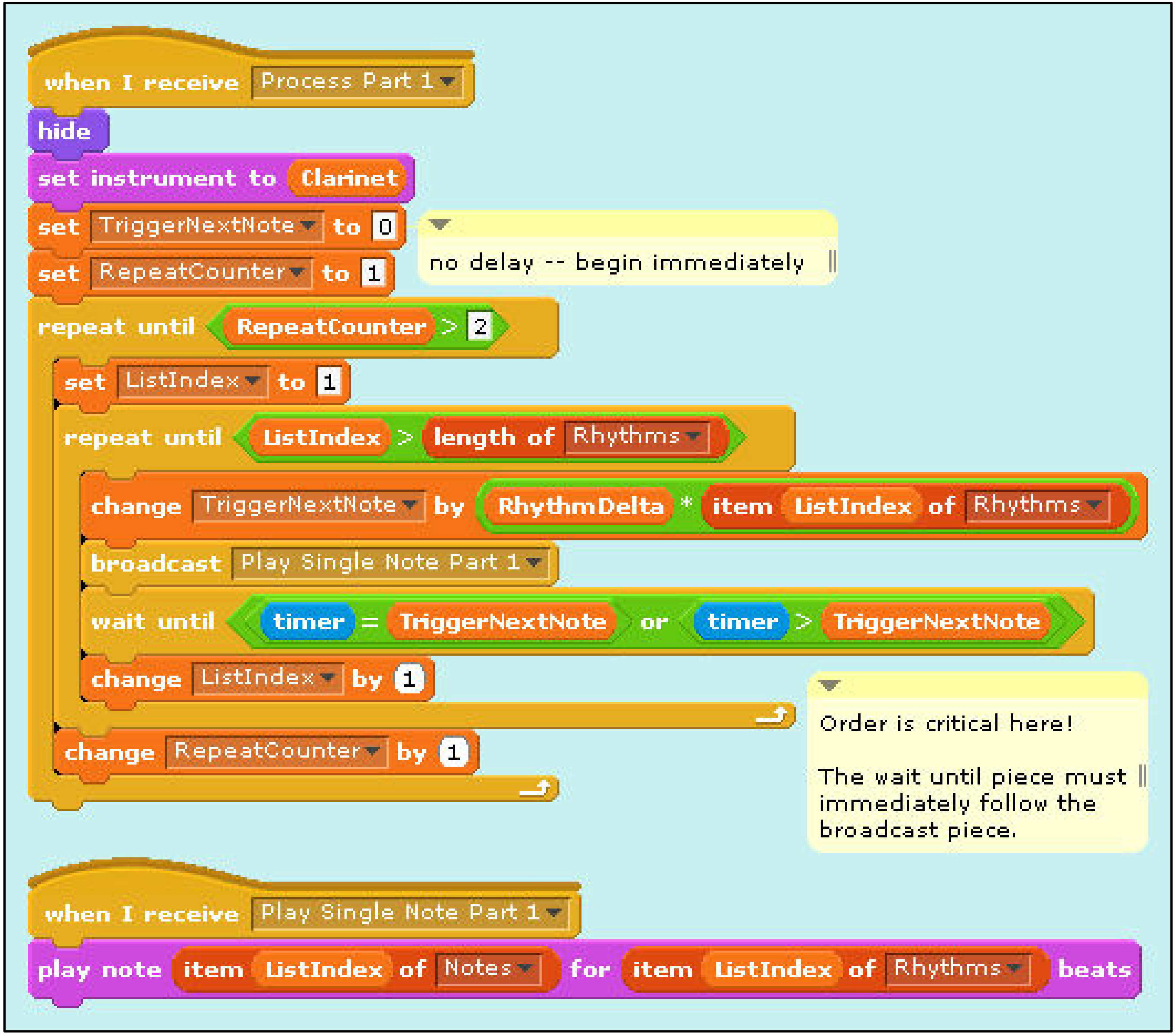
continued on next page

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.

Progressive Music Examples

No. 10: Synchronizing Play from Lists (cont'd)

(10c) **Part 1** Script



```
when I receive [Process Part 1]
hide
set instrument to (Clarinet)
set [TriggerNextNote] to [0]
set [RepeatCounter] to [1]
repeat until <(RepeatCounter) > [2]>
    set [ListIndex] to [1]
    repeat until <(ListIndex) > (length of [Rhythms])>
        change [TriggerNextNote] by ((RhythmDelta) * (item (ListIndex) of [Rhythms]))
        broadcast [Play Single Note Part 1]
        wait until <<(timer) = (TriggerNextNote)> or <(timer) > (TriggerNextNote)>>
        change [ListIndex] by (1)
    change [RepeatCounter] by (1)

when I receive [Play Single Note Part 1]
play note (item (ListIndex) of [Notes]) for (item (ListIndex) of [Rhythms]) beats
```

no delay -- begin immediately

Order is critical here!

The wait until piece must immediately follow the broadcast piece.

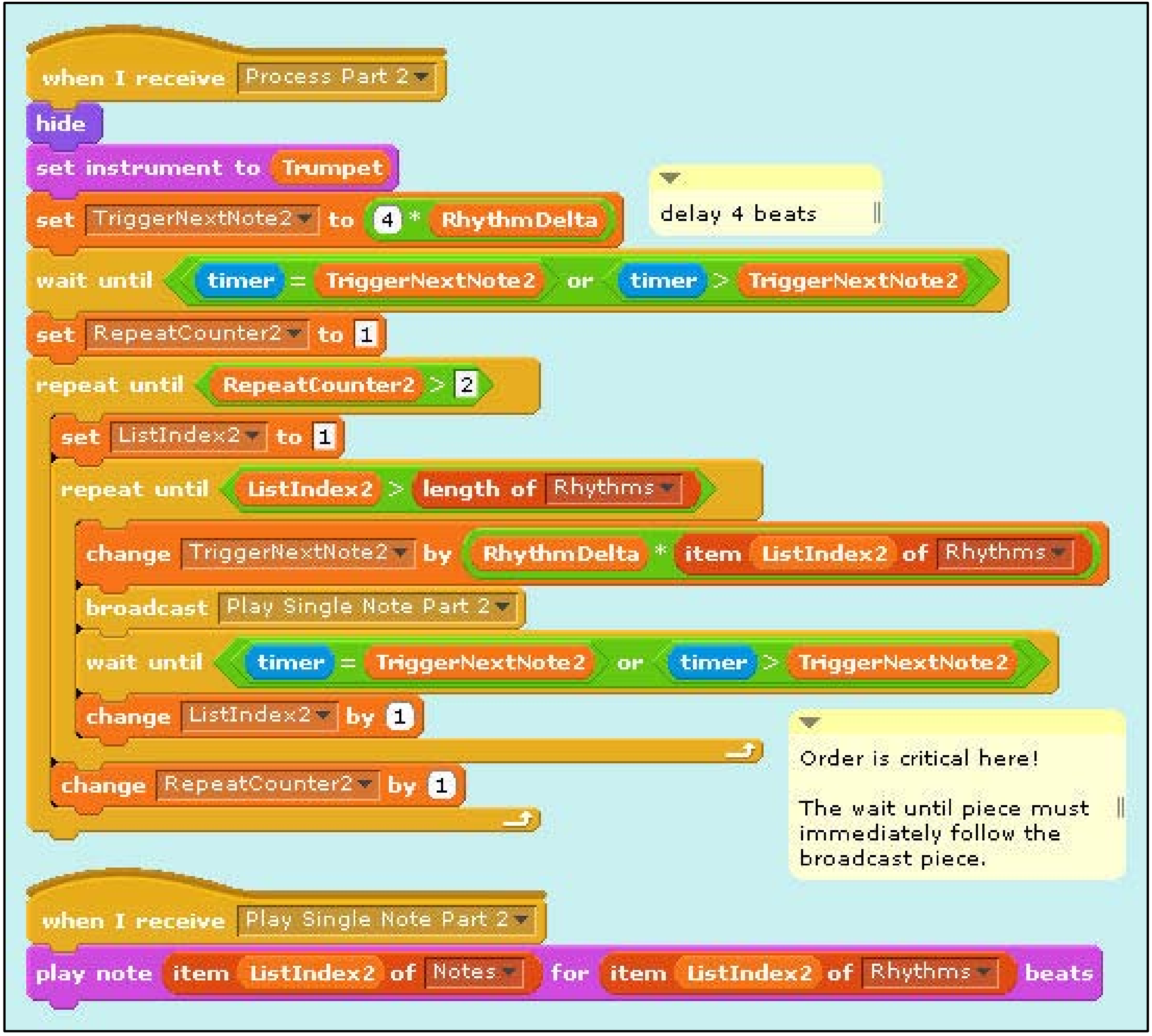
continued on next page

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.

Progressive Music Examples

No. 10: Synchronizing Play from Lists (cont'd)

(10d) **Part 2** Script



```
when I receive [Process Part 2]
hide
set instrument to (Trumpet)
set [TriggerNextNote2] to ((4) * (RhythmDelta))
wait until <<(timer) = (TriggerNextNote2)> or <(timer) > (TriggerNextNote2)>>
set [RepeatCounter2] to (1)
repeat until <(RepeatCounter2) > (2)>
    set [ListIndex2] to (1)
    repeat until <(ListIndex2) > (length of [Rhythms])>
        change [TriggerNextNote2] by ((RhythmDelta) * (item (ListIndex2) of [Rhythms]))
        broadcast [Play Single Note Part 2]
        wait until <<(timer) = (TriggerNextNote2)> or <(timer) > (TriggerNextNote2)>>
        change [ListIndex2] by (1)
    end
    change [RepeatCounter2] by (1)
end

when I receive [Play Single Note Part 2]
play note (item (ListIndex2) of [Notes]) for (item (ListIndex2) of [Rhythms]) beats
```

delay 4 beats

Order is critical here!

The wait until piece must immediately follow the broadcast piece.

end of Example 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.

Ideas for 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.

Progressive Music Examples

Ideas for Extending the Examples (cont'd)

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 before you try this!)
   - When should each part "come in"?
   - How much should the first beat of each part be offset?

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

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.

Ideas for Extending the Examples (cont'd)

6. Make a round using the G-major scale.

- Put the note values for a G-major scale into a list. See page 10 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).

Ideas for Extending the Examples (cont'd)

7. 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?

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.

Ideas for Extending the Examples (cont'd)

8. **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.

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.

Ideas for Extending the Examples (cont'd)

9. **Create a program that can play any harmonic minor 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.

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.

Ideas for Extending the Examples (cont'd)

10. 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.