

Before we begin, please  
download and install Sonic Pi  
[www.sonic-pi.net](http://www.sonic-pi.net)



# Making Music with Code

Jared O'Leary  
BootUp PD



# What's the plan?

- Scratch - 1 hour and 15 minutes
- Break - 10 minutes
- Sonic Pi - 1 hour and 15 minutes
- Discussion - 20 minutes



# Some Context

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)



# Technology Classes at Desert Thunder

Jared O'Leary  
Arizona State University  
Avondale Elementary School District



BootUP

Beatbox Machine

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)

# How to reach the resources

- ▶ [www.JaredOLEary.com](http://www.JaredOLEary.com)
  - ▶ Presentations
    - ▶ Making Music with Code



A decorative graphic on the left side of the slide. It features a grey arrow pointing right at the top, and several thin, curved black lines that sweep upwards and to the right, overlapping the arrow and the text area.

# Beatbox Machine Process

- ▶ Project sequence
  - ▶ Create funny backdrops
  - ▶ Triggering sounds
- ▶ Project extensions
- ▶ Debugging
- ▶ Lesson plan and coder resources





# Let's Share What We Created!

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)

# 10 Minute Break

If you haven't already, please  
download and install Sonic Pi  
[www.sonic-pi.net](http://www.sonic-pi.net)

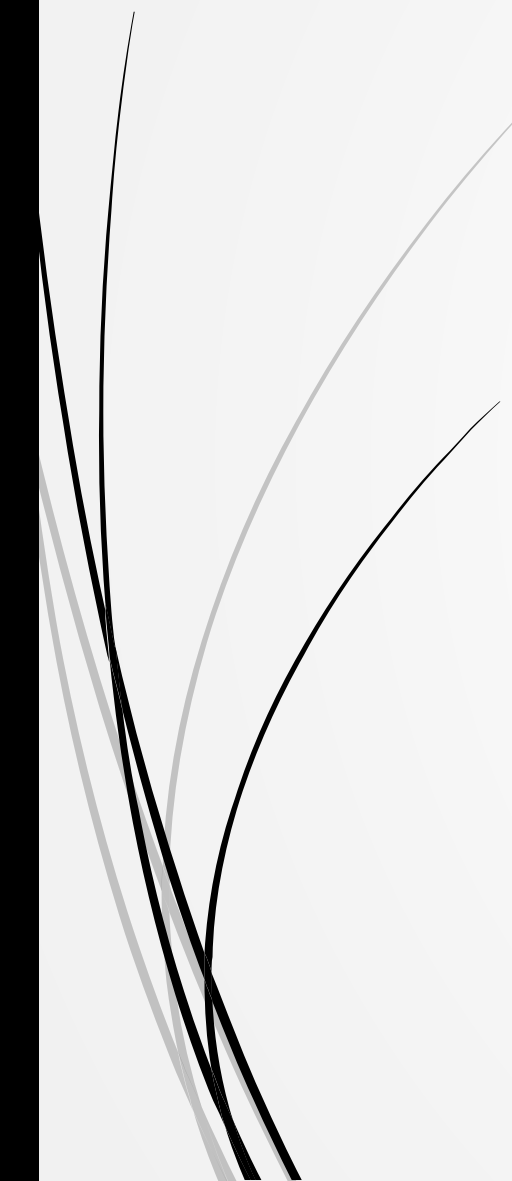


# Sonic Pi

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)



# What is Sonic Pi?

- Composing
  - Performing
  - Improvising
  - Aleatoric
- 



# Some Bach and an infinite drumset

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)



# Hot Cross Buns

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)



# Setting our tempo

1. `use_bpm 144`



# Adding our notes

1. `use_bpm 144`
- 2.
3. `play :e`
4. `play :d`
5. `play :c`





# Separating our notes

1. use\_bpm 144
- 2.
3. play :e
4. sleep 2
5. play :d
6. sleep 2
7. play :c
8. sleep 4



# Defining a function

1. `use_bpm 144`
- 2.
3. `define :buns do`
4. `play :e`
5. `sleep 2`
6. `play :d`
7. `sleep 2`
8. `play :c`
9. `sleep 4`
10. `end`



Congratulations,  
you recreated Cage's 4' 33"!



# Calling our function

3. define :buns do

4. play :e

5. sleep 2

6. play :d

7. sleep 2

8. play :c

9. sleep 4

10. end

11.

12. buns()

13. buns()



# Starting our next phrase

12. buns()

13. buns()

14.

15. play :c

16. sleep 1



# Using repeats

12. buns()

13. buns()

14. 4.times do

15. play :c

16. sleep 1

17. end



# Using repeats

```
12. buns()  
13. buns()  
14. 4.times do  
15.   play :c  
16.   sleep 1  
17. end  
18. 4.times do  
19.   play :d  
20.   sleep 1  
21. end
```



# Completing our song

```
12. buns()  
13. buns()  
14. 4.times do  
15.   play :c  
16.   sleep 1  
17. end  
18. 4.times do  
19.   play :d  
20.   sleep 1  
21. end  
22. buns()
```





# Changing our synth

1. use\_bpm 144
2. use\_synth :tri
- 3.
4. define :buns do
5.   play :e
6.   sleep 2
7.   play :d
8.   sleep 2
9.   play :c
10.  sleep 4
11. end



# Shaping our notes

1. use\_bpm 144
2. use\_synth :tri
- 3.
4. define :buns do
5.   play :e, release: 2
6.   sleep 2
7.   play :d, release: 2
8.   sleep 2
9.   play :c, release: 4
10.  sleep 4
11. end



# Adding effects

13.

14. `with_fx :echo do`

15. `buns()`

16. `buns()`

.....

24. `buns()`

25. `end`



# In a different buffer

1. `use_bpm 144`



# Creating our loop

1. use\_bpm 144
- 2.
3. live\_loop :perc do
4. end



# Metal

1. use\_bpm 144
- 2.
3. live\_loop :perc do
4.   sample :bd\_haus
5.   sleep 0.25
6. end



# EDM

1. use\_bpm 144
- 2.
3. live\_loop :perc do
4.   sample :bd\_haus if (spread 1, 4).tick
5.   sleep 0.25
6. end



# Adding in another rhythm

1. use\_bpm 144
- 2.
3. live\_loop :perc do
4.   sample :bd\_haus if (spread 1, 4).tick
5.   sample :elec\_bong if (spread 3, 8).look
6.   sleep 0.25
7. end





## ...and another

1. use\_bpm 144
- 2.
3. live\_loop :perc do
4.   sample :bd\_haus if (spread 1, 4).tick
5.   sample :elec\_bong if (spread 3, 8).look
6.   sample :perc\_snap if (spread 3, 4).look
7.   sleep 0.25
8. end

# Adjusting our amplitude

1. `use_bpm 144`
- 2.
3. `live_loop :perc do`
4.     `sample :bd_haus if (spread 1, 4).tick`
5.     `sample :elec_bong if (spread 3, 8).look`
6.     `sample :perc_snap, amp: 0.3 if (spread 3, 4).look`
7.     `sleep 0.25`
8. `end`

# Back in our original buffer

12.

13. **define :song do**

14.   with\_fx :echo do

15.     buns()

16.     buns()

.....

25.     buns()

26.   end

**27. end**



# Press Run for Cage's encore

(there is a purpose for this)



# Hip cross buns

1. use\_bpm 144
- 2.
3. live\_loop :perc do
4.   sample :bd\_haus if (spread 1, 4).tick
5.   sample :elec\_bong if (spread 3, 8).look
6.   sample :perc\_snap, amp: 0.3 if (spread 3, 4).look
7.   sleep 0.25
8. end
- 9.
10. **song()**



# Exploring with Sonic Pi

- ▶ Sonic Pi's built-in help
  - ▶ Tutorials
  - ▶ Examples
  - ▶ Synths
  - ▶ Fx
  - ▶ Samples
  - ▶ Lang(uage)
- ▶ [www.JaredOLEary.com/sonic-pi](http://www.JaredOLEary.com/sonic-pi)



# Let's Share What We Created!

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)



🔍 Search

## Music & Coding?

- If you've skimmed through my website, you've probably noticed some of my research interests and nexus I enjoy exploring: music, coding, technology, video games, participatory culture, and so on. I have divided my website into two main sections (computer programming and music education) in order to assist with finding resources I share with others. Despite this organizational divide, I believe the two overlap in ways that few discuss in either field. Borrowing from common programming syntax, I have chosen to label this page "music & coding" because I believe when someone codes music and sound projects, we cannot have an understanding of one without the other. Meaning, one uses (or develops) an understanding of both music and code when engaging in music and sound related coding projects. The following sections intend to parse out some of the specific music & coding examples found within the computer programming section.

## MAX/MSP

- MAX/MSP is a graphical programming language used by composers and artists to create interactive music and art software, installations, compositions, and more. I have used the language to create a few music tools and software that I have shared on my website. All of these projects are geared toward a high school and above level of understanding music & coding.
  - [Click here](#) to view all of the project files.

# Music & Coding

## Scratch

- Scratch is a block-based programming language developed by MIT and used by elementary through professional programmers to create and share media arts programs. I, and the kids I work with, have developed a variety of music and sound programs in Scratch. All of these projects are geared toward an elementary and above level of understanding music & coding.
  - [Click here](#) to check out some of the music and sound projects (and more) I have developed.
  - [Click here](#) to check out some of the music and sound projects the kids I work with have developed.
  - [Click here](#) to check out a music and sound studio that Jesse Rathgeber and myself curate.

## Sonic Pi

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)

- Sonic Pi is a platform that uses the programming language "Ruby" to create live music through code. All of these projects are geared toward an elementary and above level of understanding music & coding.



# Upcoming sessions I'm presenting

- **Monday, July 9<sup>th</sup>**

- **Project-based Learning with Scratch**

- *2:15-3:15 pm in room 216*

- **Tuesday, July 10<sup>th</sup>**

- **Assessing Coding Projects**

- *10:00-10:20 am in room 213*

- **Moving Beyond Puzzles: Project-based Coding**

- *10:40-11:00 am in room 212*

# Let's talk or explore

- ▶ [www.JaredOLEary.com](http://www.JaredOLEary.com)
  - ▶ Presentations
    - ▶ Making Music with Code

