Before we begin, please download and install Sonic Pi www.sonic-pi.net

Making Music with Code

Jared O'Leary
BootUp PD

What’s the plan?

- Sonic Pi ≈ 1 hour
- Break ≈ 10 minutes
- Scratch ≈ 1 hour and 30 minutes
- Discussion ≈ 20 minutes

Some Context

Technology Classes at Desert Thunder

Jared O'Leary
Arizona State University
Avondale Elementary School District

Sonic Pi

What is Sonic Pi?

- Composing
- Performing
- Improvising
- Aleatoric

Some Bach and an infinite drumset

Hot Cross Buns

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

```
use_bpm 144

define :buns do
  play :e
  sleep 2
  play :d
  sleep 2
  play :c
  sleep 4
end
```

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

Calling our function

```
define :buns do
  play :e
  sleep 2
  play :d
  sleep 2
  play :c
  sleep 4
end

buns()
buns()
```

Starting our next phrase

12. buns()
13. buns()
14.
15. play :c
16. sleep 1

Using repeats

```
buns()
buns()
4.times do
  play :c
  sleep 1
end
```

Using repeats

```
buns()
buns()
4.times do
  play :c
  sleep 1
end
4.times do
  play :d
  sleep 1
end
```

Completing our song

```
buns()
buns()
4.times do
  play :c
  sleep 1
end
4.times do
  play :d
  sleep 1
end
buns()
```

Changing our synth

```
use_bpm 144
use_synth :tri

define :buns do
  play :e
  sleep 2
  play :d
  sleep 2
  play :c
  sleep 4
end
```

Shaping our notes

```
use_bpm 144
use_synth :tri

define :buns do
  play :e, release: 2
  sleep 2
  play :d, release: 2
  sleep 2
  play :c, release: 4
  sleep 4
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

```
use_bpm 144

live_loop :perc do
  sample :bd_haus
  sleep 0.25
end
```

EDM

```
use_bpm 144

live_loop :perc do
  sample :bd_haus if (spread 1, 4).tick
  sleep 0.25
end
```

Adding in another rhythm

```
use_bpm 144

live_loop :perc do
  sample :bd_haus if (spread 1, 4).tick
  sample :elec_bong if (spread 3, 8).look
  sleep 0.25
end
```

…and another

```
use_bpm 144

live_loop :perc do
  sample :bd_haus if (spread 1, 4).tick
  sample :elec_bong if (spread 3, 8).look
  sample :perc_snap if (spread 3, 4).look
  sleep 0.25
end
```

Adjusting our amplitude

```
use_bpm 144

live_loop :perc do
  sample :bd_haus if (spread 1, 4).tick
  sample :elec_bong if (spread 3, 8).look
  sample :perc_snap, amp: 0.3 if (spread 3, 4).look
  sleep 0.25
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

```
use_bpm 144

live_loop :perc do
  sample :bd_haus if (spread 1, 4).tick
  sample :elec_bong if (spread 3, 8).look
  sample :perc_snap, amp: 0.3 if (spread 3, 4).look
  sleep 0.25
end

song()
```

Exploring Sonic Pi

- Sonic Pi's built-in help
  - Tutorials
  - Examples
  - Synths
  - Fx
  - Samples
  - Lang(uage)
- www.JaredOLeary.com/sonic-pi

Let’s Share What We Created!

10 Minute Break

(feel free to ask me questions)

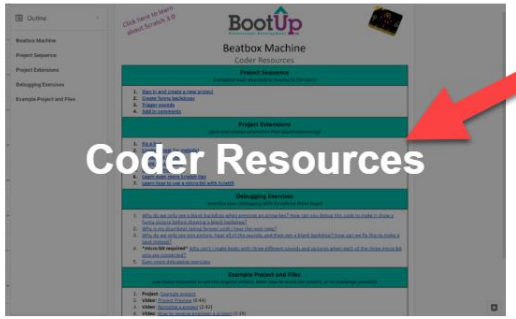
Scratch

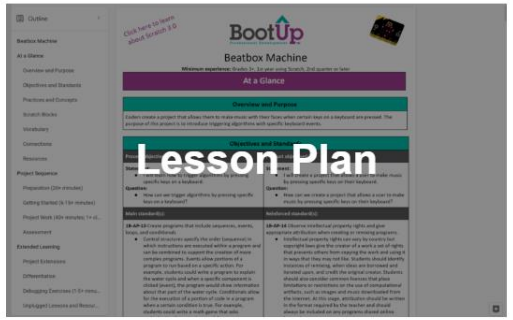

BootUp
Beatbox Machine

How to reach the resources

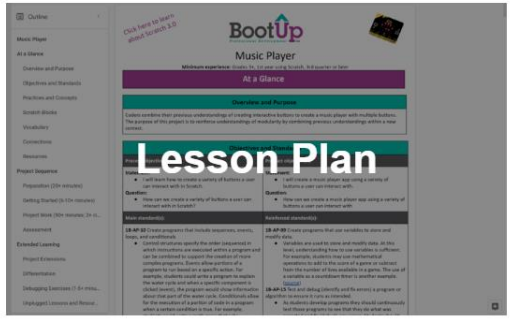
- www.JaredOLeary.com
  - Presentations
    - Making Music with Code (ISTE)

Beatbox Machine With Scratch





Music Player With Scratch



2019

Making Music with Code (ISTE)

A Corpus-assisted Discourse Analysis of Chiptune-related Practices Discussed within Chipmusic.org

Introduction to Ipsative Assessment

2018

Toward Equitable Learning through Rhizomatic Design

Interest-driven Coding Projects

Moving Beyond Puzzles: Project-based Coding

Assessing Coding Projects

Project-based Learning with Scratch

Making Music with Code (CSTA)

Facilitating Multiple Programming Languages in One Space

Interest-driven Coding Projects

2017

Interest-driven Coding and Learning

Augmenting Programmatic Music

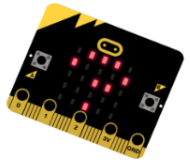
Depression, Suicide, and Music Education

Exploring Music and Video Games

Outline

Click here to learn about Scratch 3.0

BootUp Professional Development™



Beatbox Machine

Coder Resources

Project Sequence

(complete each step before moving to the next)

1. Sign in and create a new project
2. Create funny backdrops
3. Trigger sounds
4. Add in comments

Project Extensions

(pick and choose extensions that sound interesting)

1. Fix a bug
2. Create a beat (or melody)
3. Customize sounds
4. Share your project
5. Create a thumbnail
6. Learn even more Scratch tips
7. Learn how to use a micro:bit with Scratch

Debugging Exercises

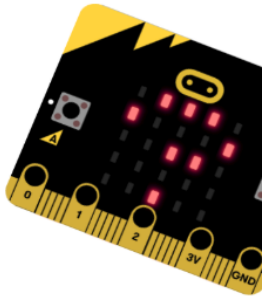
(practice your debugging skills by solving these bugs)

1. Why do we only see a blank backdrop when pressing an arrow key? How can you debug this code to make it show a funny picture before showing a blank backdrop?
2. Why is my drumbeat taking forever until I hear the next note?
3. Why do we only see one picture, hear all of the sounds, and then see a blank backdrop? How can we fix this to make a beat instead?
4. ***micro:bit required*** Why can't I make beats with three different sounds and pictures when each of the three micro:bit pins are connected?
5. Even more debugging exercises

Example Project and Files

ick here to learn
about Scratch 3.0

BootUp
Professional Development™



Beatbox Machine

Coder Resources

Project Sequence

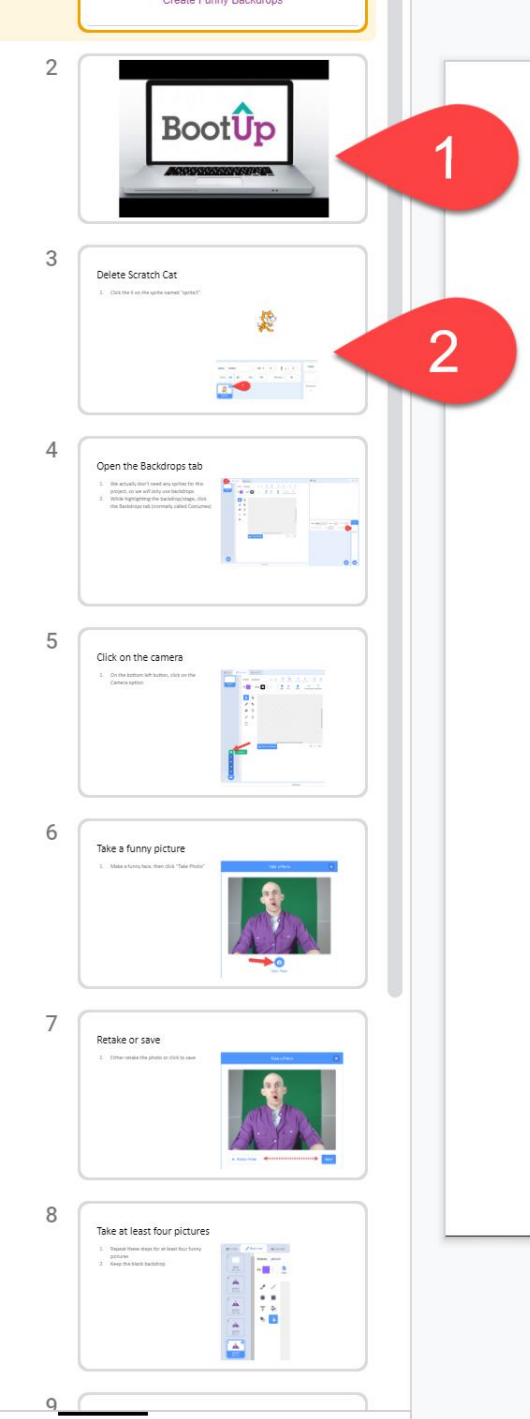
(complete each step before moving to the next)

1. Sign in and create a new project
2. Create funny backdrops
3. Trigger sounds
4. Add in comments

Project Extensions

(pick and choose extensions that sound interesting)

1. Fix a bug



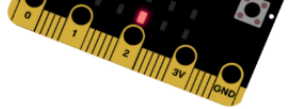
BootUp Professional Development™

Beatbox Machine

Create Funny Backdrops

Click to learn more about Scratch

BootUp Professional Development™



Beatbox Machine

Coder Resources

Project Sequence

(complete each step before moving to the next)

1. Sign in and create a new project
2. Create funny backdrops
3. Trigger sounds
4. Add in comments

1

Project Extensions

(pick and choose extensions that sound interesting)

2

1. Fix a bug
2. Create a beat (or melody)
3. Customize sounds
4. Share your project
5. Create a thumbnail
6. Learn even more Scratch tips
7. Learn how to use a micro:bit with Scratch

Project Extensions

(pick and choose extensions that sound interesting)

1. Fix a bug
2. Create a beat (or melody)
3. Customize sounds
4. Share your project
5. Create a thumbnail
6. Learn even more Scratch tips
7. Learn how to use a micro:bit with Scratch

Debugging Exercises

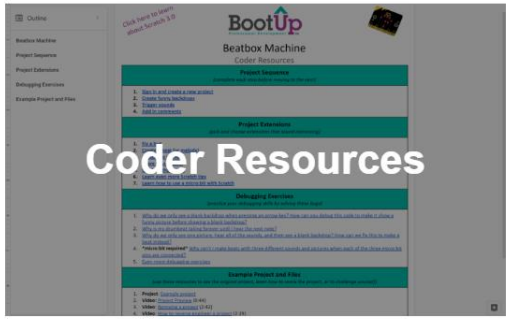
(practice your debugging skills by solving these bugs)

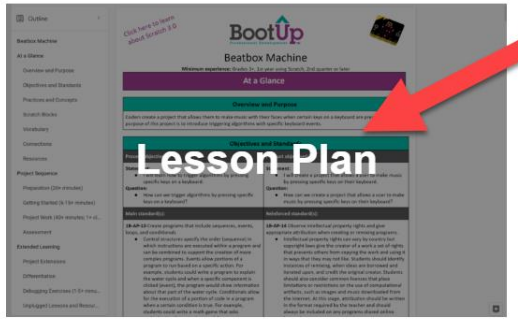
1. Why do we only see a blank backdrop when pressing an arrow key? How can you debug this code to make it show a funny picture before showing a blank backdrop?
2. Why is my drumbeat taking forever until I hear the next note?
3. Why do we only see one picture, hear all of the sounds, and then see a blank backdrop? How can we fix this to make a beat instead?
4. ***micro:bit required*** Why can't I make beats with three different sounds and pictures when each of the three micro:bit pins are connected?
5. Even more debugging exercises

Example Project and Files

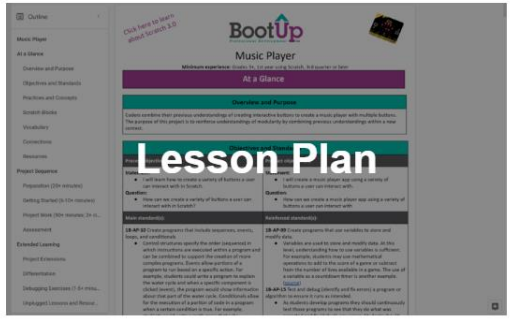
Let’s Share What We Created!

Beatbox Machine With Scratch





Music Player With Scratch



2019

Making Music with Code (ISTE)

A Corpus-assisted Discourse Analysis of Chiptune-related Practices Discussed within Chipmusic.org

Introduction to Ipsative Assessment

2018

Toward Equitable Learning through Rhizomatic Design

Interest-driven Coding Projects

Moving Beyond Puzzles: Project-based Coding

Assessing Coding Projects

Project-based Learning with Scratch

Making Music with Code (CSTA)

Facilitating Multiple Programming Languages in One Space

Interest-driven Coding Projects

2017

Interest-driven Coding and Learning

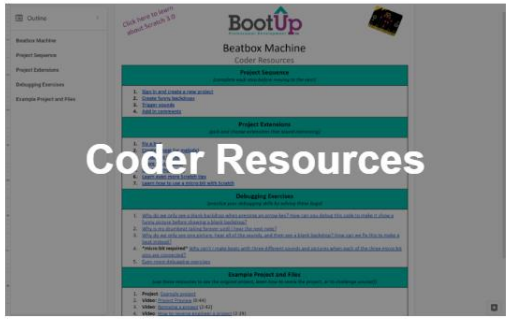
Augmenting Programmatic Music

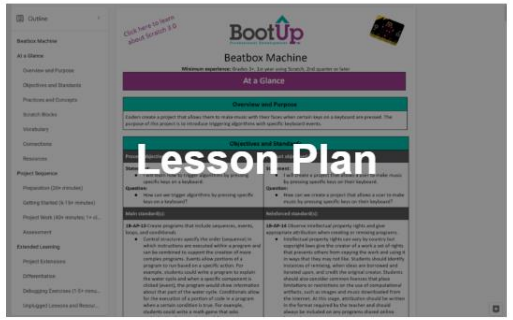
Depression, Suicide, and Music Education

Exploring Music and Video Games

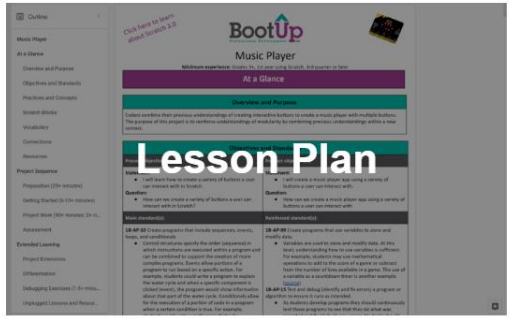

BootUp
Dance Funky
Party
Previous Song
Stop Song
Next Song
Shuffle
Music Player

Beatbox Machine With Scratch





Music Player With Scratch



2019

Making Music with Code (ISTE)

A Corpus-assisted Discourse Analysis of Chiptune-related Practices Discussed within Chipmusic.org

Introduction to Ipsative Assessment

2018

Toward Equitable Learning through Rhizomatic Design

Interest-driven Coding Projects

Moving Beyond Puzzles: Project-based Coding

Assessing Coding Projects

Project-based Learning with Scratch

Making Music with Code (CSTA)

Facilitating Multiple Programming Languages in One Space

Interest-driven Coding Projects

2017

Interest-driven Coding and Learning

Augmenting Programmatic Music

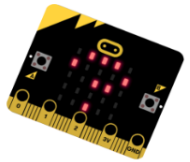
Depression, Suicide, and Music Education

Exploring Music and Video Games

Outline

Click here to learn about Scratch 3.0

BootUp Professional Development™



Music Player

Coder Resources

Project Sequence

(complete each step before moving to the next)

1. Sign in and create a new project
2. Add music and backdrops
3. Create a play and stop toggle
   a. Button review resources
      i. Create custom buttons
      ii. Code your buttons
4. Create next and previous song buttons
5. Indicate a button is pressed
6. Add even more functionality to your music player app
7. Add in comments

Project Extensions

(pick and choose extensions that sound interesting)

1. Control volume with arrow keys
2. Create a shuffle toggle (Advanced)
3. Share your project
4. Create a thumbnail
5. Learn even more Scratch tips
6. Learn how to use a micro:bit with Scratch

Debugging Exercises

(practice your debugging skills by solving these bugs)

1. Why won't the music stop when you press the "stop" button?
2. Why doesn't the "next song" button switch backdrops but doesn't start playing the next song?
3. What's wrong with the play/stop toggle?
4. ***micro:bit required*** Why doesn't the shuffle button turn on or off when the B button is pressed on the micro:bit?
5. Even more debugging exercises

i. Create custom buttons
ii. Code your buttons

4. Create next and previous song buttons
5. Indicate a button is pressed
6. Add even more functionality to your music player app
7. Add in comments

Project Extensions

(pick and choose extensions that sound interesting)

1. Control volume with arrow keys
2. Create a shuffle toggle (Advanced)
3. Share your project
4. Create a thumbnail
5. Learn even more Scratch tips
6. Learn how to use a micro:bit with Scratch

Debugging Exercises

(practice your debugging skills by solving these bugs)


micro:bit Tips
Create a Volume Meter with LEDs

2


First time using a micro:bit?
BootUp
micro:bit

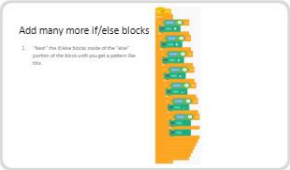
3


BootUp

4


Here's the pattern

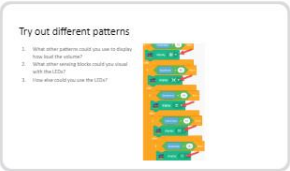
5


Add many more if/else blocks

6

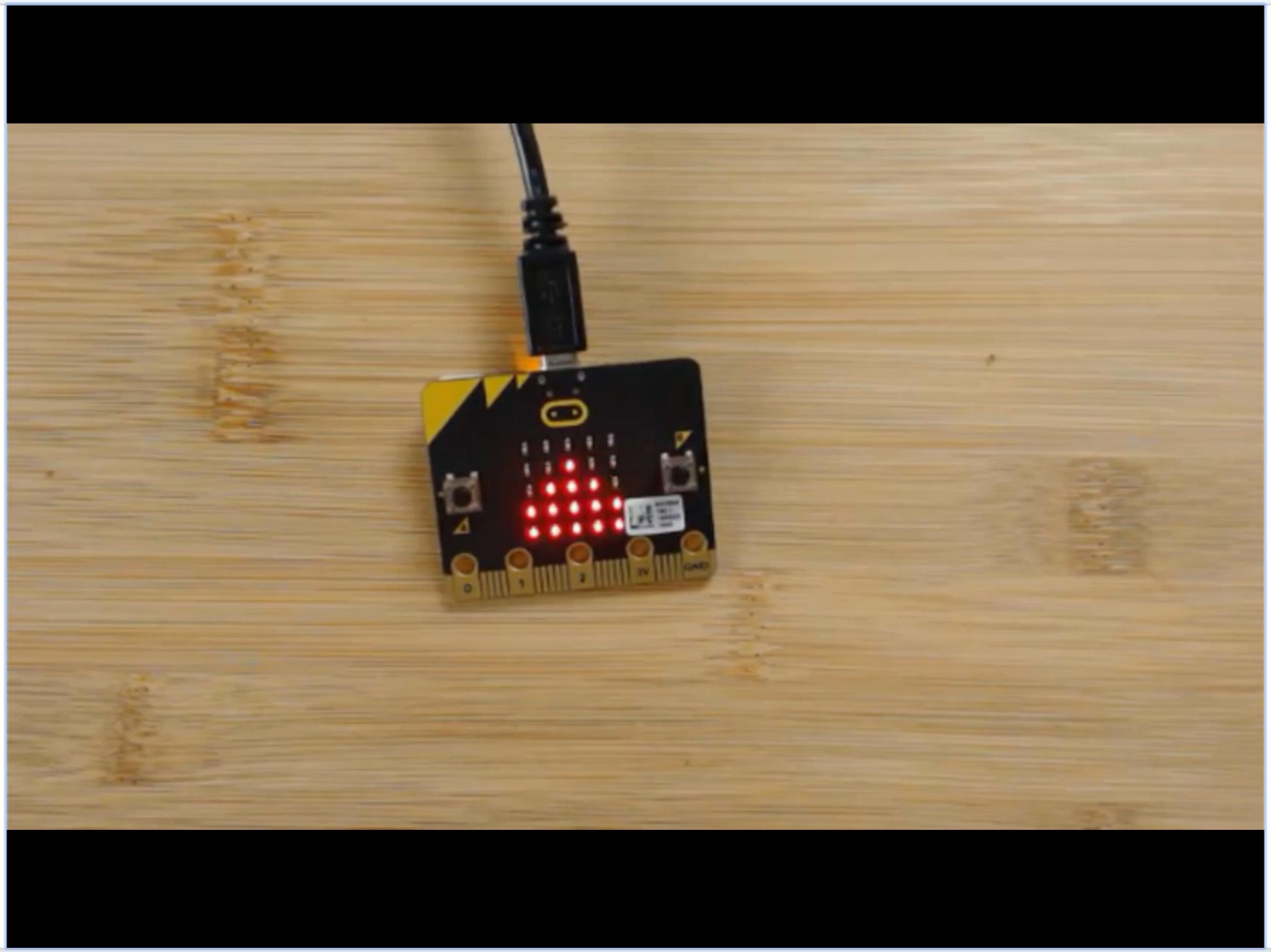

Make the last "else" blank

7

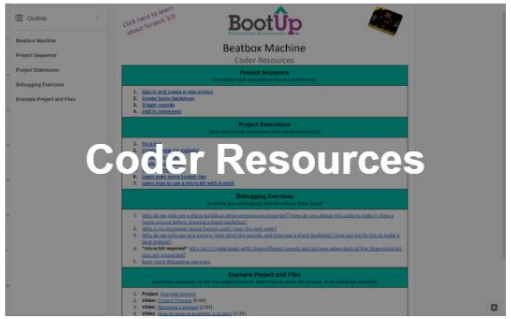

Try out different patterns

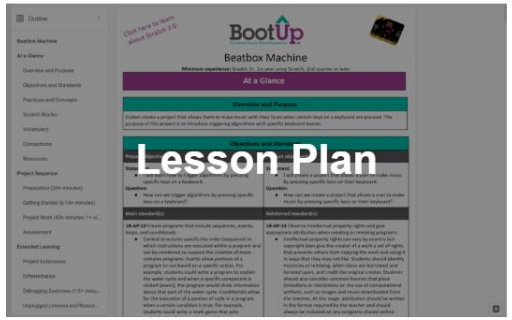
8


Try out the examples on the micro:bit page on Scratch

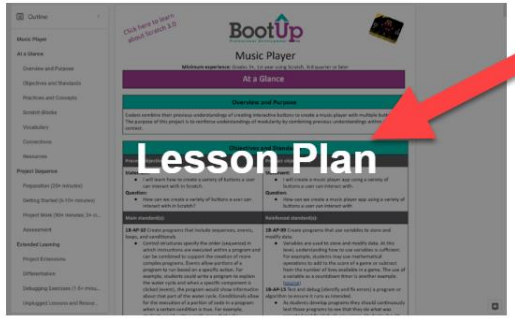


Beatbox Machine With Scratch





Music Player With Scratch



2019

Making Music with Code (ISTE)

A Corpus-assisted Discourse Analysis of Chiptune-related Practices Discussed within Chipmusic.org

Introduction to Ipsative Assessment

2018

Toward Equitable Learning through Rhizomatic Design

Interest-driven Coding Projects

Moving Beyond Puzzles: Project-based Coding

Assessing Coding Projects

Project-based Learning with Scratch

Making Music with Code (CSTA)

Facilitating Multiple Programming Languages in One Space

Interest-driven Coding Projects

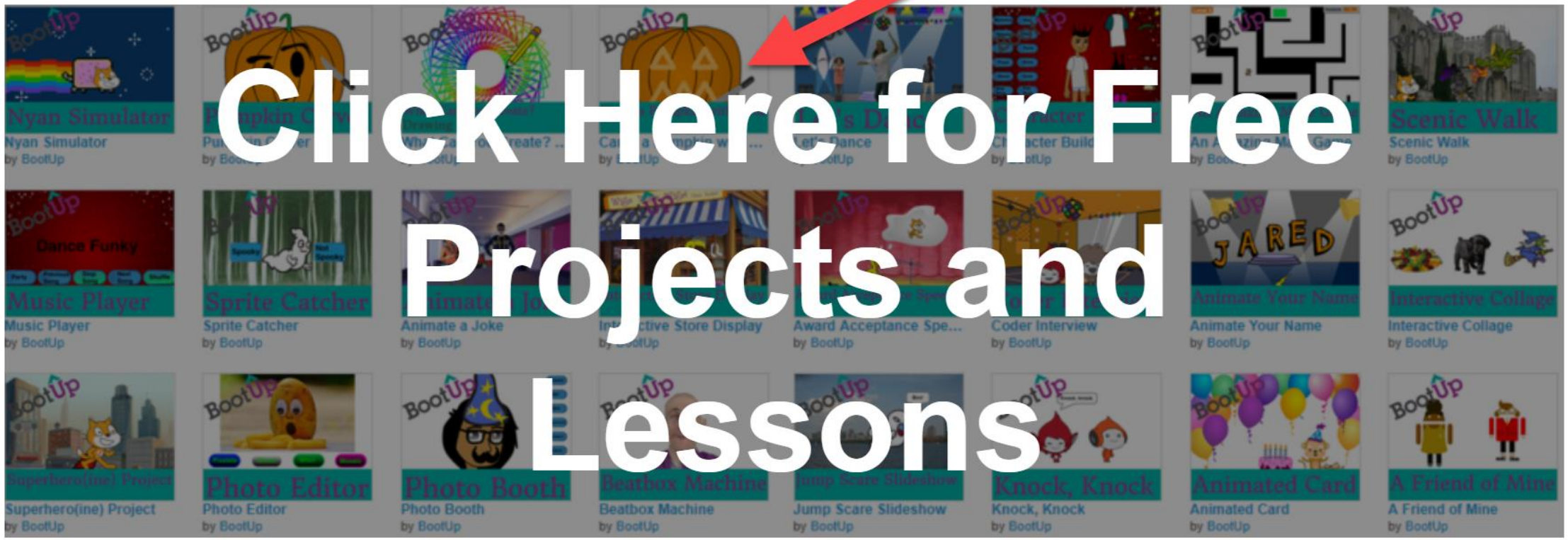
2017

Interest-driven Coding and Learning

Augmenting Programmatic Music

Depression, Suicide, and Music Education

Exploring Music and Video Games

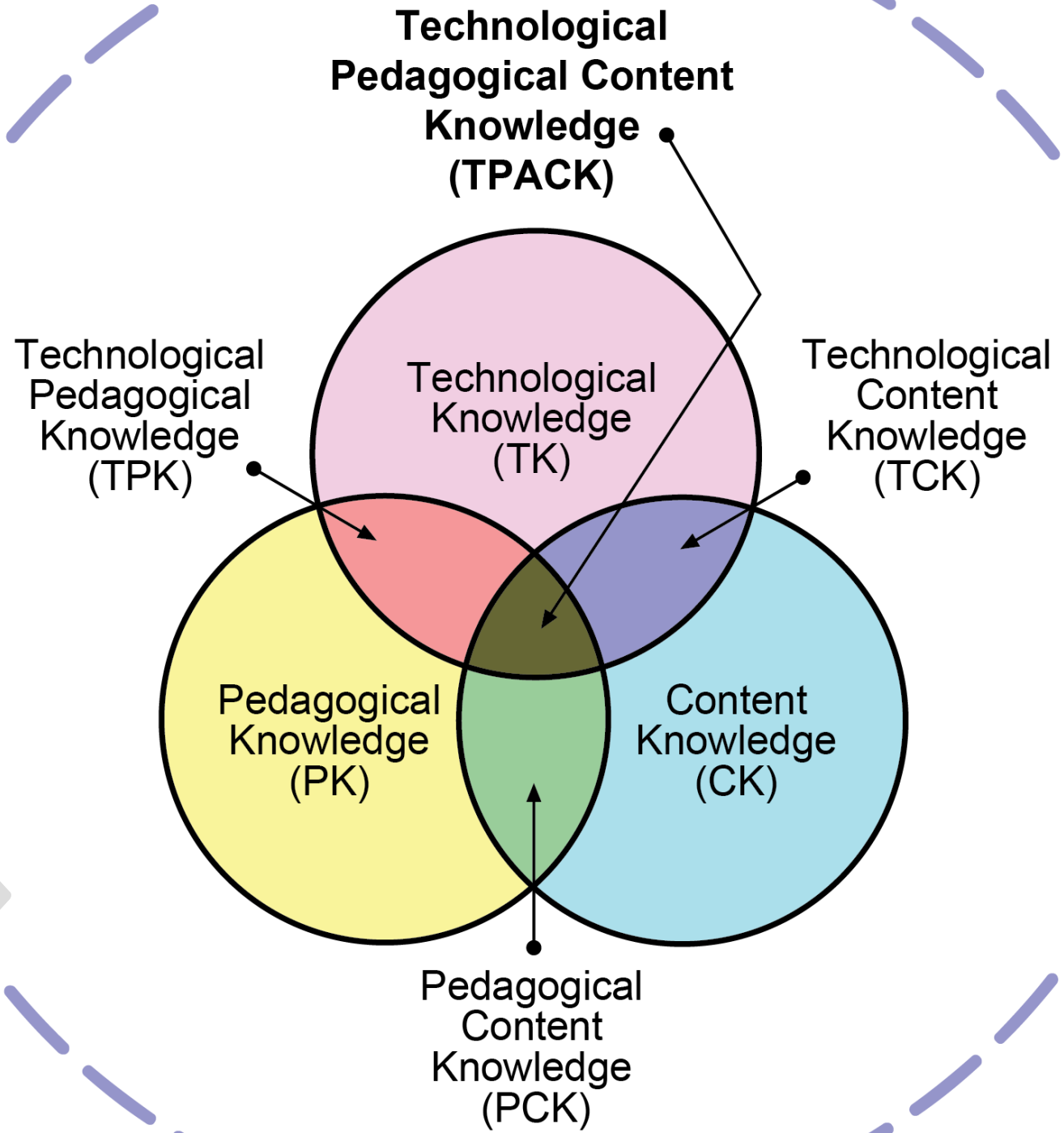


Click Here for Free Projects and Lessons

Let’s Share What We Created!

Discussion

TPACK



ISTE Standards

ISTE STANDARDS FOR COMPUTER SCIENCE

In an increasingly digital world, computer science plays a star role. The ISTE Standards·CSE describe what computer science teachers must kno effectively integrate these essential concepts.

EXPLORE COMPUTER SCIENCE EDUCATOR STAN

1 Knowledge of content	Computer science educators demonstrate knowledge of computer science co important principles and concepts.
2 Effective teaching and learning strategies	Computer science educators demonstrate effective content pedagogical strat the discipline comprehensible to students.
3 Effective learning environments	Computer science educators apply their knowledge of learning environments maintaining safe, ethical, supportive, fair and effective learning environments
Effective professional knowledge and skills	Computer science educators demonstrate professional knowledge and skills i readiness to apply them.

ISTE Standards for

Computer Science

DOWNLOAD

EXPLORE COMPUTER SCIENCE EDUCATOR STANDARDS

1	Knowledge of content	Computer science educators demonstrate knowledge of computer science content and model important principles and concepts. +
2	Effective teaching and learning strategies	Computer science educators demonstrate effective content pedagogical strategies that make the discipline comprehensible to students. +
3	Effective learning environments	Computer science educators apply their knowledge of learning environments by creating and maintaining safe, ethical, supportive, fair and effective learning environments for all students. +
4	Effective professional knowledge and skills	Computer science educators demonstrate professional knowledge and skills in their field and readiness to apply them. +

EXPLORE COMPUTER SCIENCE EDUCATOR STANDARDS

1	Knowledge of content	Computer science educators demonstrate knowledge of computer science content and model important principles and concepts. +
2	Effective teaching and learning strategies	Computer science educators demonstrate effective content pedagogical strategies that make the discipline comprehensible to students. +
3	Effective learning environments	Computer science educators apply their knowledge of learning environments by creating and maintaining safe, ethical, supportive, fair and effective learning environments for all students. +
4	Effective professional knowledge and skills	Computer science educators demonstrate professional knowledge and skills in their field and readiness to apply them. +

EXPLORE COMPUTER SCIENCE EDUCATOR STANDARDS

1	Knowledge of content	Computer science educators demonstrate knowledge of computer science content and model important principles and concepts. +
2	Effective teaching and learning strategies	Computer science educators demonstrate effective content pedagogical strategies that make the discipline comprehensible to students. +
3	Effective learning environments	Computer science educators apply their knowledge of learning environments by creating and maintaining safe, ethical, supportive, fair and effective learning environments for all students. +
4	Effective professional knowledge and skills	Computer science educators demonstrate professional knowledge and skills in their field and readiness to apply them. +

EXPLORE COMPUTER SCIENCE EDUCATOR STANDARDS

1	Knowledge of content	Computer science educators demonstrate knowledge of computer science content and model important principles and concepts. +
2	Effective teaching and learning strategies	Computer science educators demonstrate effective content pedagogical strategies that make the discipline comprehensible to students. +
3	Effective learning environments	Computer science educators apply their knowledge of learning environments by creating and maintaining safe, ethical, supportive, fair and effective learning environments for all students. +
4	Effective professional knowledge and skills	Computer science educators demonstrate professional knowledge and skills in their field and readiness to apply them. +

CT COMPETENCIES

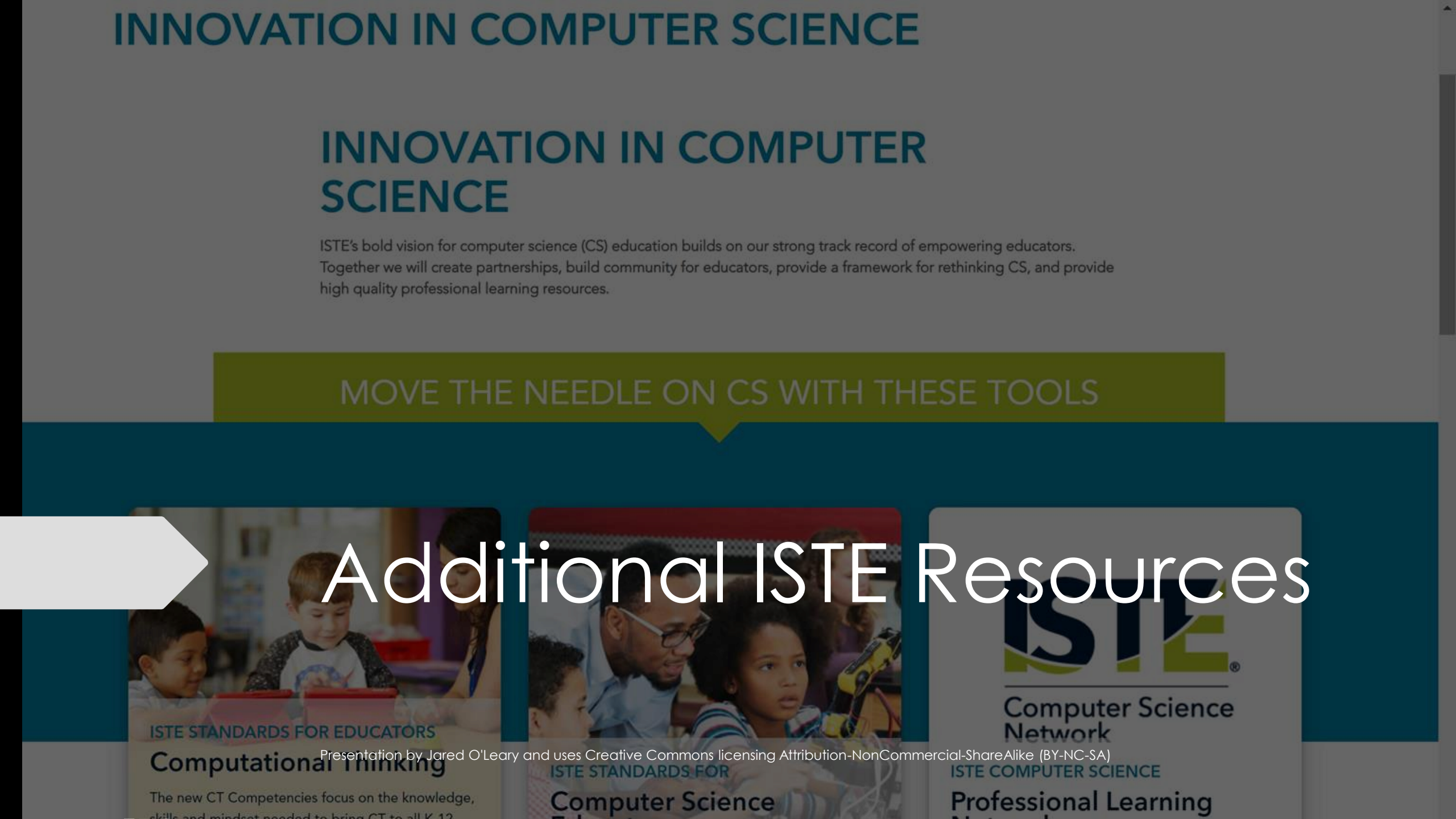
Read how this body of work complements the existing CSTA K-12 Computer Science Standards for Students and the K-12 Computer Science Framework, and why ISTE created the CT Competencies.

1. Computational Thinking (Learner) +
2. Equity Leader (Leader) +
3. Collaborating Around Computing (Collaborator) +
4. Creativity & Design (Designer) +
5. Integrating Computational Thinking (Facilitator) +

EXPLORE THE STUDENT STANDARDS

1	Empowered Learner	Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.
2	Digital Citizen	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.
3	Knowledge Constructor	Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others.
4	Innovative Designer	Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.
5	Computational Thinker	Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
6	Creative Communicator	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.
7	Global Collaborator	Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.

Additional ISTE Resources



Music && Coding

Jared O'Leary
// Multiplicity

Contact Me | Curriculum Vitae | Presentations | Publications

Computer Science Education | Music Education

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

- MAXP/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.

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

- 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.

Session Evaluation

Please take a moment to evaluate this session. Your valuable feedback helps make the overall program stronger and ensures we're meeting your learning needs. Evaluations are also used by the conference program committee to provide feedback to presenters and inform future presentations.

To provide feedback and rate the quality of this session, please use the ISTE19 app or locate the session online using the program search at isteconference.org.

Thank you!

Upcoming sessions I'm presenting

- **Sunday, June 23rd**
  - **Designing and Facilitating a Media Arts and Technology Makerspace**
    - *3:30-4:00 pm in location 122B*
    - *Note, I'm the second half of the session as there is a different presenter from 3:00-3:30*
- **Monday, June 24th**
  - **Project-based Learning with Scratch**
    - *4:00-5:00 pm in location 118B*
    - *Registration code:* ***BYOD243***

Let’s talk or explore

- www.JaredOLeary.com
  - Presentations
    - Making Music with Code (ISTE)