

Moving Beyond Sequential Learning

Jared O'Leary
BootUp PD

Moving beyond sequential learning

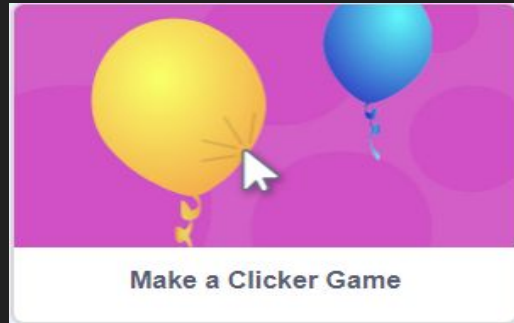
How much experience do you have with Scratch?

- No experience
- Some experience (e.g., I've tried it a couple of times)
- A moderate amount of experience (e.g., I teach it every now and then)
- A lot of experience (e.g., I teach it regularly)

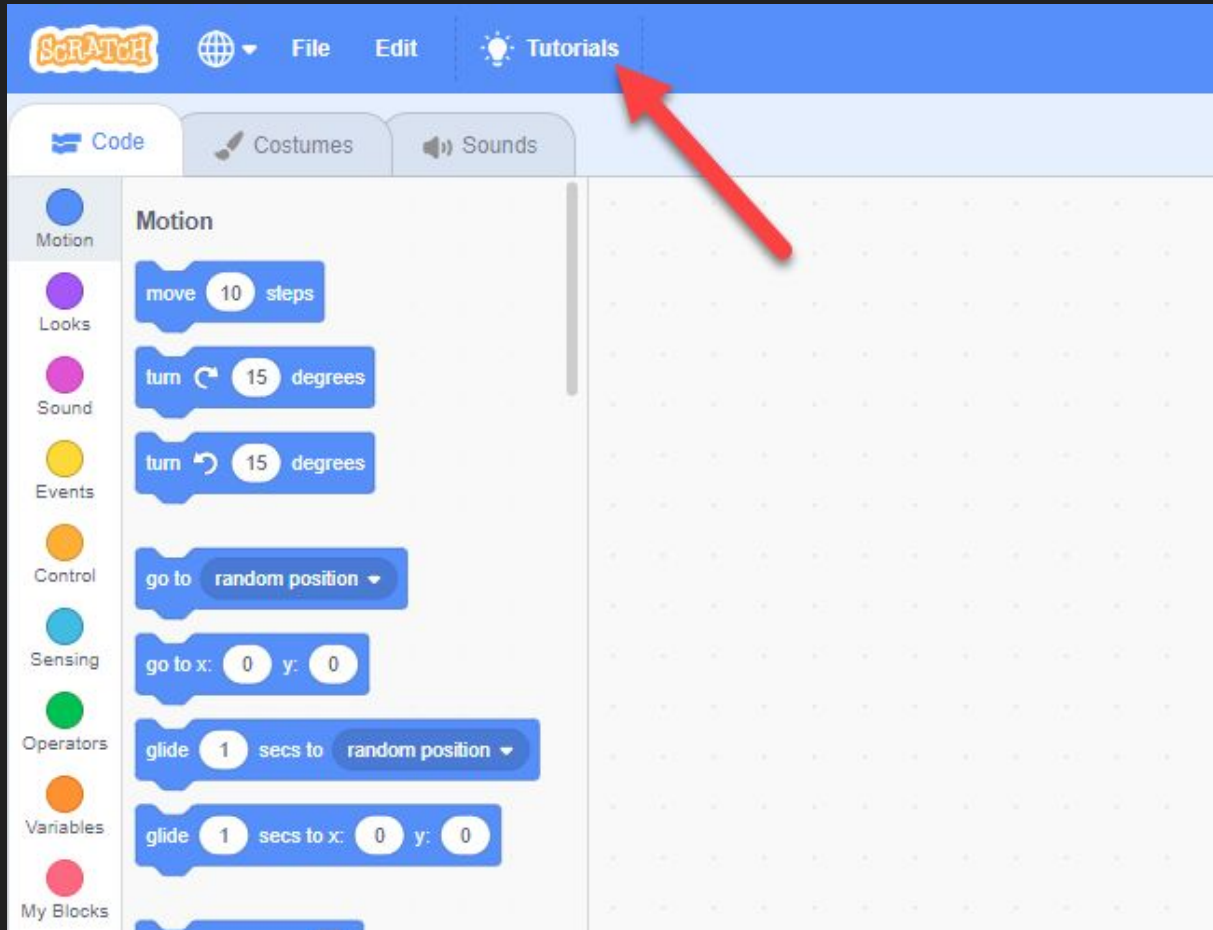
Submit

How to reach the resources

- [Direct link is in the chat](#)
- www.JaredOLeary.com
 - Presentations
 - Moving Beyond Sequential Learning









Make a Chase Game



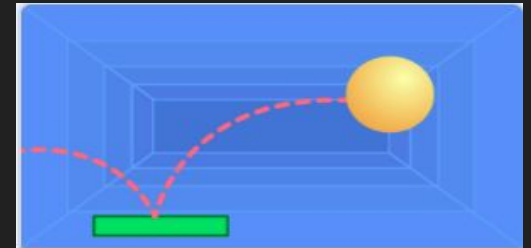
Animate an Adventure Game



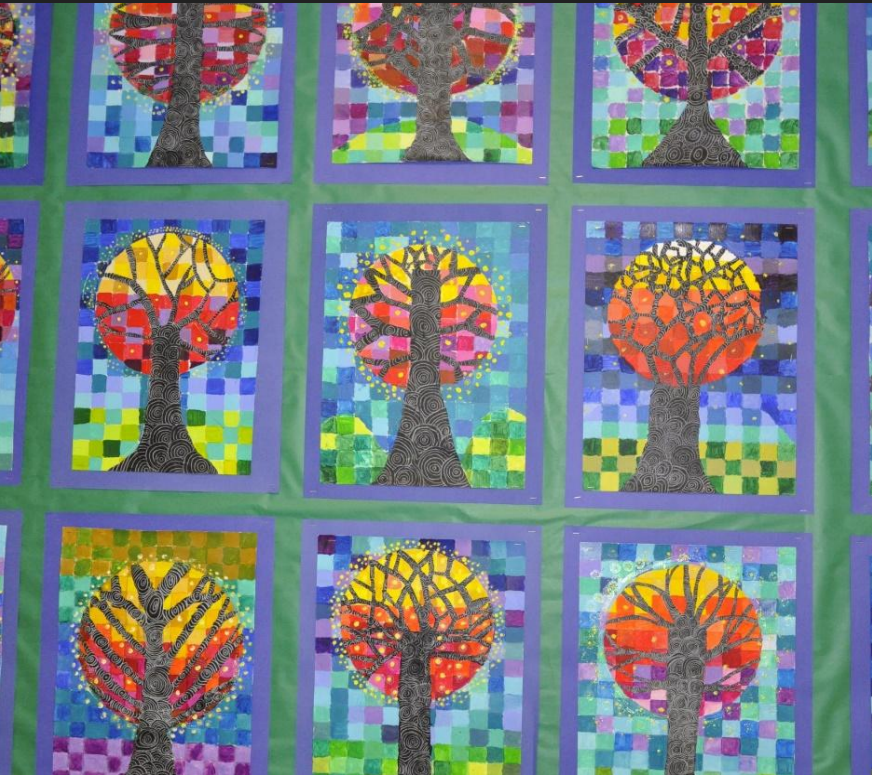
Make a Clicker Game



Make it Fly



Pong Game





Getting Started



Animate a Name



Imagine a World



Make Music



Create A Story



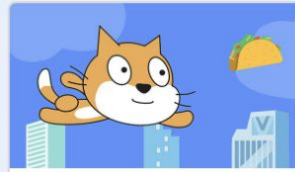
Make a Chase Game



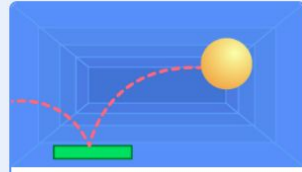
Animate A Character



Make a Clicker Game



Make it Fly



Pong Game



Code a Cartoon



Animate an Adventure Game



Video Sensing



Create Animations That Talk



Talking Tales



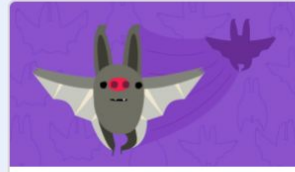
Add a Sprite



Add a Backdrop



Change Size



Glide Around



Record a Sound



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

Potential project prompts

- Can you create a school appropriate project that...
 - ...helps someone?
 - ...is scary, funny, exciting, boring, musical, silly, relaxing, or colorful?
 - ...solves a problem you see in the world?
 - ...reminds you of a special event, story, or place?
 - ...you can give as a gift to someone else?
 - ...you can use for another class?



Nyan Simulator

Nyan Simulator
by BootUp



Pumpkin Carver

Pumpkin Carver
by BootUp



What Can You Create? ... Drawing

What Can You Create? ...
by BootUp



Carve a Pumpkin with ...

Carve a Pumpkin with ...
by BootUp



Let's Dance

Let's Dance
by BootUp



Character Builder

Character Builder
by BootUp



An Amazing Maze Game

An Amazing Maze Game
by BootUp



Scenic Walk

Scenic Walk
by BootUp



Music Player

Music Player
by BootUp



Sprite Catcher

Sprite Catcher
by BootUp



Animate a Joke

Animate a Joke
by BootUp



Interactive Store Display

Interactive Store Display
by BootUp



Award Acceptance Spe...

Award Acceptance Spe...
by BootUp



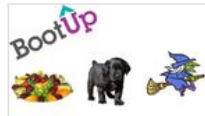
Coder Interview

Coder Interview
by BootUp



Animate Your Name

Animate Your Name
by BootUp



Interactive Collage

Interactive Collage
by BootUp



Superhero(ine) Project

Superhero(ine) Project
by BootUp



Photo Editor

Photo Editor
by BootUp



Photo Booth

Photo Booth
by BootUp



Beatbox Machine

Beatbox Machine
by BootUp



Jump Scare Slideshow

Jump Scare Slideshow
by BootUp



Knock, Knock

Knock, Knock
by BootUp



Animated Card

Animated Card
by BootUp



A Friend of Mine

A Friend of Mine
by BootUp

1. Look at the project options linked to in the chat
2. Click on a project that looks interesting
3. Follow the steps under “project sequence”
4. Post questions in the chat or ask to share your audio/video

The screenshot shows a 'Project Sequence' interface. At the top, there is a yellow header with the text 'Project Sequence' and a downward arrow. Below the header, a note says '(complete each step before moving to the next)'. A list of steps is shown: '1. Sign in and remix this project' (with a purple link) and '2. Turn a sprite into button' (with a teal arrow). The main content area features a white slide with the 'BootUp Professional Development' logo, the text 'Interactive Art', and the subtitle 'Turn a Sprite Into a Button'. A 'Google Slides' icon is visible in the bottom right of the slide. Below the slide, a list of steps continues: '3. Make a sprite talk', '4. Create hidden sprites', and '5. Add in comments' (all with teal arrows). At the bottom, there is a purple header for 'Project Extensions' and a rightward arrow.

Animate Your Name

Experience: 1st year, 1st quarter

Practice: Creating computational artifacts, Testing and refining computational artifacts, and Communicating about computing

Concept: Algorithms and Control

Length: 60+

Watch this first

At a glance

Project
sequence

Extended
learning

Coder resources

Project Lesson Overview

 Lesson plan
 Overview video

If this is your first time navigating our lessons, please take the time to watch this video to learn how our lessons are formatted and how to quickly navigate between sections.

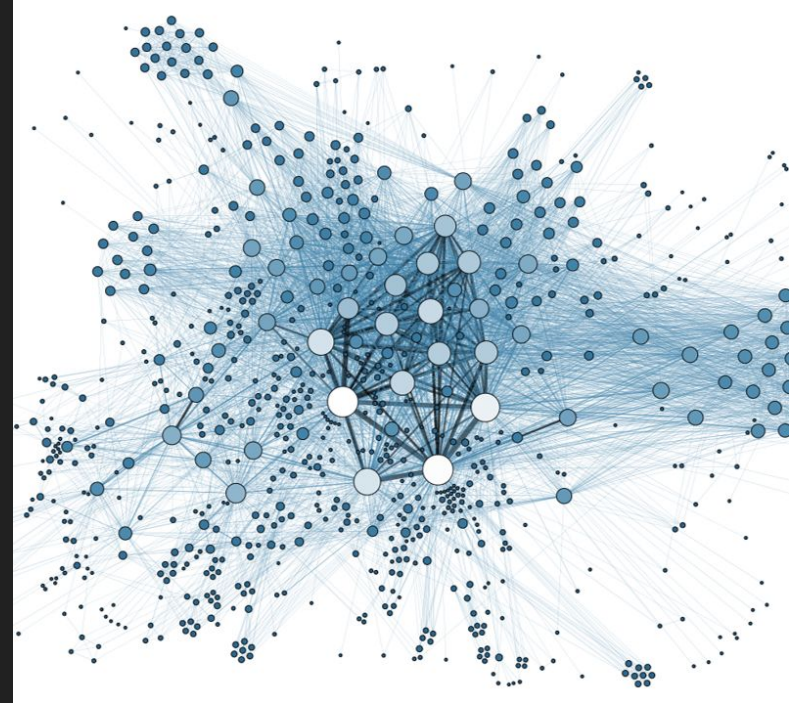
Animate Your Name - Project Preview



Step 1

Step 2

Step 3



Sequential Design	Rhizomatic Design
Group-based learning	Individualized learning
Standards-driven	Interest-driven
Learning CS concepts and practices within a predetermined sequence	Exploring and creating through a multitude of CS concepts and practices
The teacher or curricula determines the group's path	Each student determines their own path
Teachers can stay one lesson ahead of students without being overwhelmed	Teachers should frontload much of their understanding of content knowledge before starting
The teacher's role is to guide students from one step to the next	The teacher's role is to facilitate student learning through discovery and inquiry
Direct instruction is usually from a teacher to a group of students	Direct instruction can be from a teacher or resource to an individual student
Easier to grade and assess	Harder to grade and assess
Administrators are likely familiar with this approach	Administrators might not be familiar with this approach

00100000 00100011 01000011 01010011 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 00100011 01000011 01010011 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 00100011 01000011 01010011 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 01001011 00111000
00100000 01010000 01101111 01100100 01100011 01100001 01110011
01110100 00100000 01001011 00111000

#CSK8
Podcast
with
Jared O'Leary

[75: Rhizomatic Learning with Catherine Bornhorst, Jon Stapleton, and Katie Henry](#)

Affinity spaces characteristics

1. Affinity spaces share a common endeavor
2. Affinity spaces are not segregated by age
3. Affinity spaces are not segregated by experience
4. Affinity spaces encourage, but do not require, active participation
5. Interaction transforms content within an affinity space
6. Affinity spaces encourage both intensive and extensive knowledge
7. Affinity spaces encourage individual and distributed knowledge
8. Affinity spaces encourage dispersed knowledge
9. Affinity spaces encourage and honor tacit knowledge
10. Affinity spaces encourage a multitude of engagement
11. Affinity spaces have multiple routes to status
12. Leadership is porous and leaders are resources

Please note this is a preprint version and should not be cited. To read the published chapter, visit <https://jaredoleary.com/publications/applications-of-affinity-space-characteristics-in-music-education>

Applications of Affinity Space Characteristics in Music Education

Jared O'Leary

Introduction

This chapter explores affinity spaces as a framework for informal spaces or groups with a shared common endeavor and questions how characteristics of such spaces may inform music education practices. The chapter begins with a description of the twelve characteristics of an affinity space—a framework useful for understanding diverse social contexts that may enable or support music learning—and provides a brief illustration of how we might use affinity spaces as a lens for exploring social interactions within an example online music space. After this description, I question the framework of affinity spaces in relation to music-learning communities of practice; although similar, I describe how the two frameworks differ and provide suggestions for selecting a framework for implementation in future research and practice. The remainder of this chapter discusses potential applications in educational contexts by discussing how I applied each of the characteristics in a middle school elective class, as well as a discussion on affinity spaces as a heuristic. The chapter concludes with a discussion on broader implications and considerations for the field of music education.

Questions to consider for each affinity space characteristic:

- 1. Affinity spaces share a common endeavor**
 - a. How might elementary CS classes encourage a multitude of identities such as gamer, actor, musician, creator, artist, composer, reviewer, performer, manufacturer, journalist, listener, student, teacher, and more within a shared space?
 - b. How might we encourage young coders to create projects for their peers or community?
 - c. How might young coders document these experiences in order to share processes, successes, and moments of growth or understanding?
 - d. Might young coders provide constructive critique on the projects we create in these spaces?
 - e. How might we engage in hyphenated forms of coding where young coders shift through a variety of identities and engagement (e.g., designer, artist, programmer, gamer, writer, etc.)?
 - i. How might we assess learning in a space with a multitude of CS related identities?
 - f. When is the focus on individualized, small group, or large group learning of coding concepts and understandings?
 - g. If young coders in a class did not share common interests with their peers, how might we utilize social media to connect them with other coders who share similar interests?
- 2. Affinity spaces are not segregated by age**
 - a. How might coding classes or communities remove unnecessary segregation by age?
 - b. What should CS educators and facilitators consider when creating spaces with a broad range of ages?
 - c. How might people participate in CS spaces where young and old shift between roles of teacher and student?
 - d. How might CS sequences or cycles adapt or expand to include interaction or participation across age levels?
 - e. How might age-based classes or communities interact and learn with other classes of different age groups in either synchronous or asynchronous contexts?
 - f. In what ways might social platforms and networks assist with these forms of communication?
- 3. Affinity spaces are not segregated by experience**
 - a. What kinds of expertise are valued in our classes or communities?
 - b. When are we unintentionally supporting a narrow understanding of what it means to be an expert in CS or coding?
 - c. What are potential affordances and constraints of segregating coders by experience or expertise?
 - d. How might those with more experience teach those with less, and when might

Let's chat and explore the resources rhizomatically

- [Direct link is in the chat](#)
- www.JaredOLEary.com
 - Presentations
 - Moving Beyond Sequential Learning