code jumper

Lesson 4

Threads

Code Jumper Curriculum: Lessons

Lesson 4
Threads

American Printing House for the Blind

Code Jumper Curriculum: Lessons

Published by American Printing House for the Blind
1839 Frankfort Avenue, Louisville, KY 40206
www.aph.org | info@aph.org

Overview

Lesson Objectives

Students will:

- Understand what a thread is
- Understand that computer programs are created using multiple threads
- Understand how to create a program containing multiple threads

Expected Outcomes

Students will:

- All students: Explain what threading is and identify threads in an existing program
- Most students: Recreate a program with two threads using a sample as a model
- Some students: Create an original program that combines threading and sequencing

Lesson Plan Structure

- Unplugged Activity
- Guided Code Jumper Activity
- Exploration
- Standards and Check for Understanding

Resource

- Code Jumper Tutorial Videos
  - The Hub: https://www.youtube.com/watch?v=KGb51PW9zJQ&lis=
  - Play and Pause Pod: https://www.youtube.com/watch?v=446jCw8qcDI&t=
- Code Cards

Key Vocabulary

- **Thread:** A thread is one set of commands or tasks that can be carried out at the same time as another set of commands.

Unplugged Activity: Introduction to Threads

Objective

Introduce the concept of a thread and how, when using Code Jumper, two separate sequences can play at the same time.

Materials

- Thread Sound Cards (located at the end of the lesson)
- 11 or more student participants (1 student Hub, two threads of 5 students each)

Instruction

This activity can be done as a whole group with a smaller class or in a larger class as two or more different groups of Hubs with two threads each.

1. Explain that a **thread** is a single line of commands and that in this activity students are going to perform two separate threads at the same time.
2. Select one student to represent the Hub.
3. Explain that their task is to say “run” when the two threads are ready to present their sequences at the same time.
4. Choose five students to represent Thread group #1 and five students to represent Thread group #2. (If needed, increase the number of students in each thread, as long as there are the same number of students in each group.)

5. Provide each student in Thread group #1 with a Sound Card.
6. Provide each student in Thread group #2 with a Sound Card.
7. Explain that each thread group needs to practice together with their Sound Cards, going in sequential order. Give each group five minutes to practice making their assigned sound in order to make their complete thread.
8. Explain to each thread group that they will be presenting their individual thread group sequence in front of the rest of the group.
    a. Start by having all students in thread group #1 stand side-by-side and link arms at the elbows.
    b. Instruct the first student in Thread group #1 to link elbows with the student who is representing the Hub.
    c. When the student who is representing the Hub says "run," Thread group #1 will perform their sequence of sounds.
    d. Next, have all students in Thread group #2 stand side-by-side and link arms at the elbows.
    e. The first student in Thread group #2 will link elbows with the student who is representing the Hub.
    f. When the student who is representing the Hub says "run," Thread group #2 will perform their sequence of sounds.
9. Explain to students that in Code Jumper, two separate sequences can play at the same time when they are both plugged into or linked with the Hub.
10. Explain to students that now each thread group will remain linked and connected to their Hub and will perform their sequences at the same time when the student representing each Hub says "run."

   a. This may take a few tries with the entire group. Repeat the process until you feel students are able to run the thread sequences simultaneously.

Closure

Review the concept of a thread of code and discuss with students how they think threads could be used together. For instance, instead of two threads simply running at the same time, they could complement each other by either playing alternating beats, sounds, and melodies, or by playing sounds at the same time, like words with music or harmony with melody.)

Guided Activity: Creating Threads with Code Jumper

In this activity, students will use multiple threads to create a program using Code Jumper.

Objective

- Students will be introduced to the Pause pod
- Students will create a simple program that uses two threads
- Students will create a program and then transfer it to Code Jumper

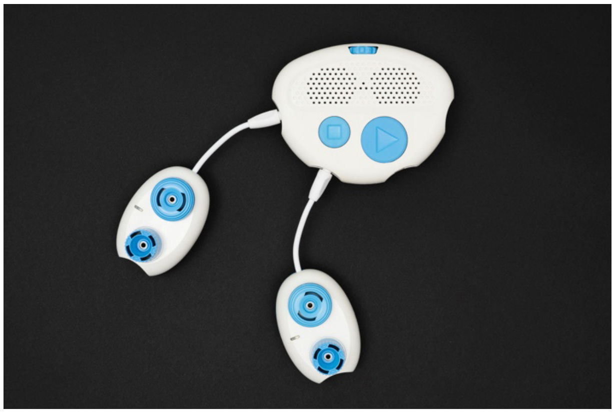
Vocabulary

- **Thread:** A thread is one set of commands or tasks that can be carried out at the same time as another set of commands.

Materials

- Code Jumper Kit:
    - Hub
    - Eight Play pods

- Two Pause pods
- Computer Science Journal



[Figure Caption:] This photo shows the Code Jumper Hub with a Play pod connected to Port 1 and another Play pod connected to Port 2.

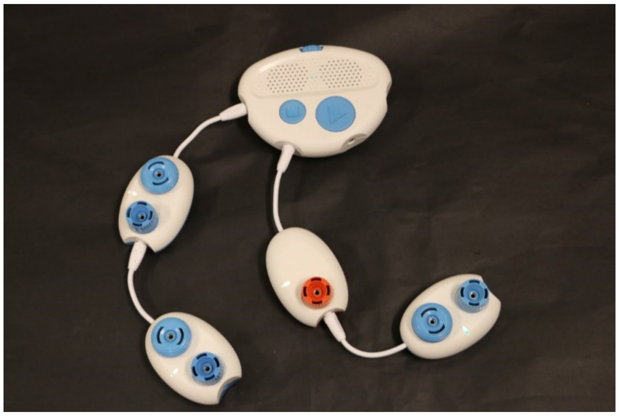
1. Divide students into groups of two or three.
2. Review the Ports on the Hub numbered 1-4, from left to right, demonstrating that they correspond to the threads in the App numbered 1-4.
3. Review the functions of the Pause and Play pods. Remind students that the function is what each specific type of Pod does (Example: The Play pod makes a sound, the Pause pod creates a pause, and the parameter is what sound the Pod is set to within a specific Sound Set.
4. Model creating two sequences of commands. Connect two Play pods to a Port in the Hub, then connect two more Play pods to a different Port in the Hub. Note: You can use any two Ports on the Hub; the Ports do not have to be adjacent to each other.
5. Change each of the threads to a different Sound Set. Do not run a program at this time.
6. Ask students to make a prediction about what will happen when they run the code now that they have physically created the code in their Computer Science Journals.

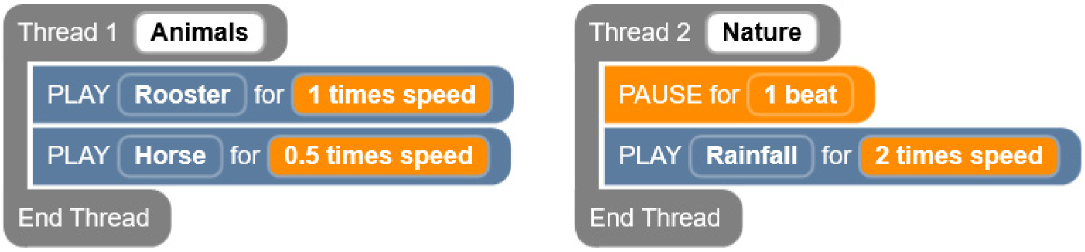
7. In pairs or groups of three, ask students to map out a program with two threads using up to eight Play pods and three Pause pods. Strategies for creating the program:

   a. In their Computer Science Journals, create a matrix (example):

Thread 1: Animal Sounds	Thread 2: Nature
Command: Play Parameter: Sound Rooster Speed: 0.5	Command: Pause Parameter: 1 beat
Command: Play Parameter: Horse Speed: 0.5	Command: Play Parameter: Rainfall, Speed: 2 times

   b. Record in the Computer Science Journal the arrangement





```
Thread 1  Animals
    PLAY  Rooster  for  1 times speed
    PLAY  Horse  for  0.5 times speed
End Thread
```

```
Thread 2  Nature
    PAUSE for  1 beat
    PLAY  Rainfall  for  2 times speed
End Thread
```

[Figure Caption:] At the top is a photo of a Code Jumper program, with two Play pods connected to the Hub at Port 1, and one Pause pod and one Play pod (in that order) connected to the Hub at Port 2. Below this photo is a screenshot of a Code Jumper program with two threads. Under Thread 1 Animals, the commands read, PLAY Rooster for 1 times speed, PLAY Horse for 0.5 times speed; the commands are followed by End Thread. Under Thread 2 Nature, the commands read, PAUSE for 1 beat, PLAY Rainfall for 2 times speed; the commands are followed by End Thread.

8. Once students have created the code in their Computer Science Journals, ask them to create the code using Code Jumper. Ask them not to run the code (press Play), but to wait until all the correct functions and parameters are set.
9. Ask the students to run the code and trace the program with their hands. They may notice that they need to use two hands to follow the sequence. Tip! Remind students to touch the Pods on the sides, not the Dials, while tracing the code. Have them play the program several times and discuss with their group. If there are several group members, ask one student to follow along in their Computer Science Journal to see if what they created matches the original sequence.
10. Ask students to make observations about what happened when they pressed Play and traced the code. What did you hear? Did sounds overlap? In what order did you hear the sounds?
11. Explain that each set of commands they just created is called a thread. Explain that a thread allows several sets of commands to be carried out at the same time. Explain that this is like doing two or more actions at the same time.
12. Give students several minutes to explore and record observations when making changes to the Sound Dial and Duration Dial to see that changes on one thread do not affect the other thread. Ask students: What happens when you unplug one thread and press Play? What will happen if you change the sound duration or a sound? How do the

sounds line up when changes are made? Does one thread end before the other? What are strategies you can use to make certain sounds play at the same time?

13. Have the students write the definition of **thread** in their Computer Science Journals. (Definition: A thread is one set of commands or tasks that can be carried out at the same time as another set of commands.)

Closure

1. Ask students to explain what threading is and to give an example of how they used it in this lesson. (Threading is where two or more separate sequences of commands are played or performed at the same time. They created threads when they plugged Pods into separate Ports on the Hub.)
2. Ask students if they can identify examples of threading in their everyday life. (Examples: When people work in a kitchen to prepare different parts of a meal, when one member of a band plays the drums while another member plays the piano, etc.)

Exploration

Objective

In this activity, students will create a program with eight Play pods, set the sounds on each Pod, and run and trace the program.

Vocabulary

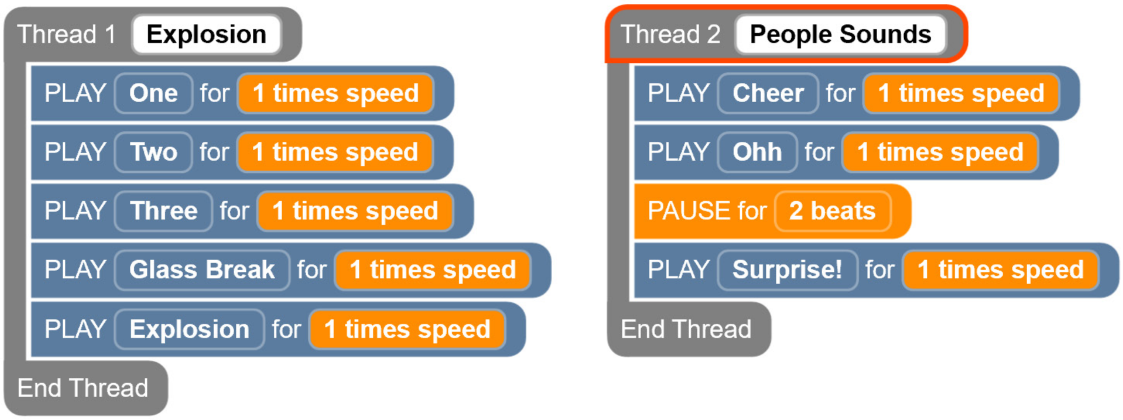
- Thread: A thread is one set of commands or tasks that can be carried out at the same time as another set of commands.

Materials

- Code Jumper Kit
  - Eight Play pods
  - Three Pause pods

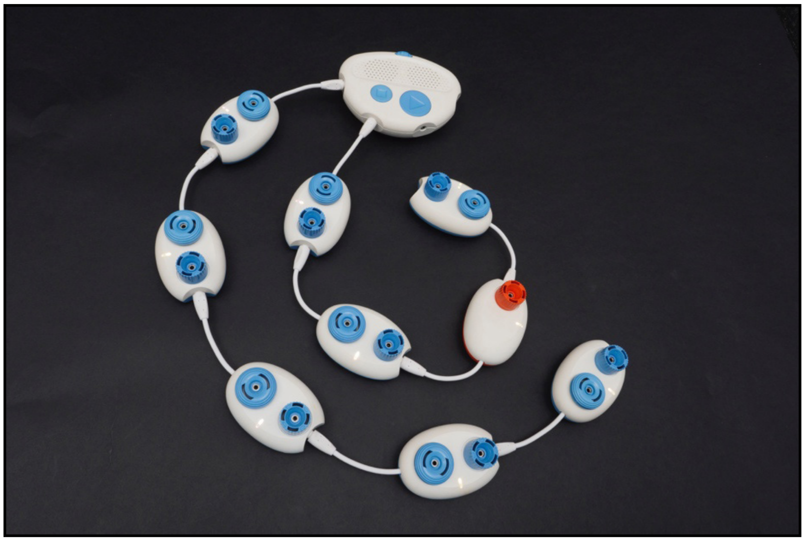
Instruction

1. Ask students to create their own story using multiple threads. Tell students that they can add sound effects to a story using multiple threads.
2. Instruct students to write the code for their story in their Computer Science Journals and compare the finished program to their written code.
3. Give students an example of a multi-thread program. (Examples: Adding in background sounds to a poem or using the Pause pods with two different Sound Sets to count down for an explosion.
4. Example:



```
Thread 1 Explosion
  PLAY One for 1 times speed
  PLAY Two for 1 times speed
  PLAY Three for 1 times speed
  PLAY Glass Break for 1 times speed
  PLAY Explosion for 1 times speed
End Thread
```

```
Thread 2 People Sounds
  PLAY Cheer for 1 times speed
  PLAY Ohh for 1 times speed
  PAUSE for 2 beats
  PLAY Surprise! for 1 times speed
End Thread
```



[Figure Caption:] At the top is a screenshot of the Code Jumper app showing a program with two threads. Under Thread 1 Explosion, the commands read, PLAY One for 1 times speed, PLAY

Two for 1 times speed, PLAY Three for 1 times speed, PLAY Glass Break for 1 times speed, PLAY Explosion for 1 times speed; the commands are followed by End Thread. Under Thread 2 People Sounds, the commands read, PLAY Cheer for 1 times speed, PLAY Ohh for 1 times speed, PAUSE for 2 beats, PLAY Surprise! for 1 times speed; the commands are followed by End Thread. Below this screenshot is a photo of a Code Jumper program, with five Play pods connected to the Hub at Port 1, and two Play pods, a Pause pod, and another Play pod (in that order) connected to the Hub at Port 2.

Closure

Ask students to write in their Computer Science Journals how threading can be useful for creating a program.

Standards and Check for Understanding

CSTA K-12 Computer Science Standards*

- 1A-AP-12: Develop plans that describe a program's sequence of events, goals, and expected outcomes.
- 1B-AP-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

National Curriculum of England*:

Key Stage 1:

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- Use logical reasoning to predict the behavior of simple programs

- Use technology purposefully to create, organize, store, manipulate and retrieve digital content
- Recognize common uses of information technology beyond school
- Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies

Key Stage 2:

- Use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- Use technology safely, respectfully and responsibly; recognize acceptable/unacceptable behavior; identify a range of ways to report concerns about content and contact.

Closing Activities and Check for Understanding

Have students write down or touch-type in their Computer Science Journals, or verbally give a sequence of instructions that is divided into threads for an everyday activity .

Possible examples:

- Musicians playing in a band
- Cooks preparing ingredients for a meal in a kitchen
- Workers building different parts of a building

- Families each doing their morning routines to get ready to leave the house.

Check for Understanding	**Completed**
Student can identify a task that consists of separate tasks being done at the same time.	Yes / No
Student can break the task down into separate sequences and threads	Yes / No
Student can explain the benefit of being able to perform separate threads at the same time.	Yes / No

*Computer Science Teachers Association (2017). CSTA K-12 Computer Science Standards, Revised 2017. Retrieved from http://www.csteachers.org/standards

*Education, Department for. "National Curriculum in England: Computing Programmes of Study." GOV.UK, 11 Sept. 2013, www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study

Unplugged Activity Sound Cards

Thread Card #1:

Sound: Snap fingers

Thread Card #2:

Sound: Clap hands slowly

Thread Card #3:

Sound: Tweet

Thread Card #4:

Sound: Buzz, buzz

Thread Card #5:

Sound: Stomp feet

Thread Card #6:

Sound: Meow

Thread Card #7:

Sound: Clap hands quickly

Thread Card #8:

Sound: Shh, shh

Thread Card #9:

Sound: Rub hands together

Thread Card #10:

Sound: Snap then clap

Unplugged Sound Cards Text Only:

Thread Card #1:

Sound: Snap fingers

Thread Card #2:

Sound: Clap hands slowly

Thread Card #3:

Sound: Tweet

Thread Card #4:

Sound: Buzz, buzz

Thread Card #5:

Sound: Stomp feet

Thread Card #6:

Sound: Meow

Thread Card #7:

Sound: Clap hands quickly

Thread Card #8:

Sound: Shh, shh

Thread Card #9:

Sound: Rub hands together

Thread Card #10:

Sound: Snap then clap

Unplugged Activity Code Cards

- Thread Code Cards for Unplugged activity
- These cards can be printed to use with students
- Code Card: Dr. Foster

```
THREAD 2 Dr Foster
    PLAY Doctor Foster at 1 times speed
    PLAY Went to Gloucester at 1 times speed
    PLAY In a shower of rain at 1 times speed
    PLAY He stepped in a puddle at 1 times speed
    PLAY Right up to his middle at 1 times speed
    PLAY And never went there again! at 1 times speed
END THREAD

THREAD 4 Nature
    PAUSE for 1 beat
    PAUSE for 1 beat
    PLAY Rainfall at 2 times speed
    PAUSE for 1/2 a beat
    PLAY Splash at 1 times speed
END THREAD
```

Sample program set up and run by the teacher: Twinkle, Twinkle with Vocals

- Sound Sets: **MIDI Instruments: Piano** at Thread 1; **Sample Sounds: Twinkle, Twinkle** at Thread 2

```
THREAD 1 Piano
    PLAY C5 for 1/2 a beat
    PLAY C5 for 1/2 a beat
    PLAY G5 for 1/2 a beat
    PLAY G5 for 1/2 a beat
END THREAD

THREAD 2 Twinkle, Twinkle
    PLAY Twinkle 1 for 1 times speed
    PLAY Twinkle 2 for 1 times speed
END THREAD
```

code jumper

For more resources, visit codejumper.com

AMERICAN PRINTING HOUSE

1839 Frankfort Avenue
Louisville, Kentucky 40206
502-895-2405 • 800-223-1839
www.aph.org • info@aph.org