

# GRAPHICAL LOOPING DIGITAL ACTIVITY MATERIAL

## Contents:

- **Lesson Plan (for teachers)**
- **Worksheet (for students)**
- **Worksheet Solutions (for teachers)**
- **Review Sheet (for students)**

These draft lesson plans, worksheets, worksheet solution guides, and review sheets are intended to accompany digital activities developed as part of *Thinking Outside the Box: Integrating Dynamic Mathematics to Advance Computational Thinking for Diverse Student Populations*, a National Science Foundation's STEM+C/Core R&D Programs #1543062. These draft versions correspond to those used in classroom research February - March 2017. As drafts, no warrants are made about these materials' completeness or correctness.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. You are free to copy and distribute it for non-commercial purposes only, but not to distribute derivative works based on it.

December 2017  
SRI International  
<http://csforall.sri.com>

# VELA LESSON #1:

## Graphical Looping (Non-Scratch)

### What comes before

Nothing; this lesson has no prerequisites.

### What comes After

Simple Loops (Scratch) Lesson

### Intended outcomes:

Students will be able to:

- Identify a real life process or object as a repeating pattern - identify the unit of repetition in the loop, and possible pre-loop and post-loop actions/objects
- Predict the resultant pattern/sequence of actions based on a “repeat generator”
- Determine how many times a loop needs to run based on given requirements
- Calculate how values change in a loop based on different initial values and change rules

### New Vocabulary Used/Introduced:

Repetition/ Looping	Doing an action or set of actions multiple times
Repeat Unit	The smallest part that repeats (Initially called “inner story” in discussions)
Pattern generator	A mechanism for creating any step of a pattern
Iteration	One cycle of the loop (Not heavily used in this lesson plan but could be introduced in discussions)

### Lesson plan schematic:

#### Part 1: Swimming panel rearranging:

1. Open <http://csforall.sri.com> and go to **Graphical Looping**.

2. Have students open the **Arrange** section of **Swimming pool stories**. Then ask student pairs to try and rearrange the four given panels to tell a sensible story.
3. Give the class a minute or two to rearrange the panels. Then, conduct a whole group discussion, facilitated with an overhead projection of the scene. Ask a few student pairs how they arranged the panels and recreate the scenes so that they are visible to the entire class. Then, ask those student pairs the following questions:
  - a. What story did you tell? (Try to discourage implausible ones and validate plausible ones.)
  - b. What evidence did you use to choose which panels went where? (Formalize what makes an arrangement “logical”/“plausible.”) Why did you not organize the panels in a different way?
4. Next, when the class has agreed on a particular arrangement of panels, ask the following questions and support discussion converging toward answers:
  - a. When someone “reads” the story you “wrote,” how far will they think the swimmer swam? (Possible answers: twice the length of the pool, deep to shallow end of pool and back; Discourage numeric answers since we do not know what the length of the pool is)
  - b. How could you extend the story so it that it felt like she swam twice as far? Assume you had access to more panels of each type. (Try to channel this toward “what would these additional pictures show?” and the realization that they’d show basically the same thing as two of the current pictures; just they’d show it *again*—i.e that we can *repeat* the “**inner story**” while keeping the same “outer story”) Note that some students may want to repeat all 4 panels; help them realize that you do not need to come out and dive into the pool for every additional lap you swim.
5. Finally, on an overhead projector, visually segregate the story into three parts
  - a. What goes before
  - b. The ‘**inner story**’ we might repeat, to increase the length of the swim
  - c. What comes after

Point out how these three parts have (and require) their own internal sequentiality too; e.g. in the inner story, the swimmer must swim to the shallow end before swimming back to the deep end, because she *starts* in the deep end.

Introduce the vocabulary “**repeat unit**”.

**A “Repeat Unit”** is the smallest part of the pattern that is repeating (and would go inside the repeat loop).

Point out that the inner story here represents the **repeat unit**. It is the part of the story or the scenario or the pattern that we want to repeat if needed.

**Identify purpose of rest of lesson; looking at how computers use repetition to generate a variety of different ‘patterns’—or experiences—from one single pattern-generator**

## Part 2: Swim generator microworld:

1. Choose **Back** in your browser to reopen the last page, or go back to <http://csforall.sri.com> and choose **Graphical Looping** again.
2. Ask students to click **Generate** under **Swimming Pool Stories** to introduce students to the repeating pattern generator where the number of repeats is, by default, set to 2. Show them the *pattern*, which is vertical, and the *results* generated by that pattern, which is displayed horizontally below. Tell students “the 2 here means the inner story in the *Pattern* gets repeated 2 times in the *Result*,” pointing out to them each of these parts in turn. Tell students that the *pattern* with a *repeat count* is a more efficient way of representing that the swimmer swam two pool lengths. This pattern generator creates the same results as the panel pattern being discussed in Part 1 - Step 3.

Explain that the “swim generator” can be used to *repeat the inner story* an arbitrary number of times. Point out the generator’s BEFORE, DURING, and AFTER parts, and have them explore how it works to generate different stories as they change the *repeat* number (blue box at the bottom of the screen). Again, point out that the two swimming panels inside the Repeat represent the “**repeat unit**” for this swim generator.

Show them what happens when you set **repeat=0** and explain that when you talk about loops, **repeat=1** actually means you do it once. (Students may be confused about the idea of “repeating” something when the swimmer travels the pool only once, which is why you should use language like “The *pattern* gets repeated one 1 time in the *result*.”)

**2. Making Real World Connections--** Explain the “repeat generator” works with any type of story, pattern, or recipe with a repeating component; not just the swimming one. Have them come up with some examples of such repeating patterns in their lives, and on the board help them think how they might structure a “pattern generator” to generate them. For example,

Repeat for 100 times:

Wake up  
Brush  
Get ready  
Have breakfast  
Get to school  
Repeat for 7 periods  
    Check subject  
    Go to correct classroom  
  
Get home  
Do homework

Play  
Eat Dinner  
Go to bed

3. Then ask students to answer the **questions 1 and 2 of the Worksheet (PAIRED ACTIVITY)**.

For the teacher:Swimming Pool Q & A

**Q:** Why have a swimming pool story that can be ordered many ways in Part 1?

**A:** Much of choosing how to express a computer program involves making decisions about how to *organize* and *order* the elements of a story or experience. There are often lots of different correct ways to write a program, but you need to know what the clear beginning, middle, and end of the story is before you try to write the program. For this swimming pool example, there may be more than one “right answer” in terms of stories that can be told out of these four panels. The important student experience here is realizing that a specific sequence is essential to a specific (possible) meaning, and that certain parts have to come *before* or *after* other parts in order to “make sense.”

For the *Generate* part of the activity, we will stick to one (possible) order, so we all are doing the same thing—though that doesn’t mean only *that* order was more “correct” in the *Arrange* part of the activity.

**Q:** Why use pool lengths in Part 2? Isn’t the first part of the Worksheet just a big test of multiplication skills?

**A:** In programming, loops get used not just to model repeating processes, but also to count various quantities that change over repetition. For example, the *checkout line at the grocery store* executes a conceptual loop of the form “pick up next item from cart, add item price to total price, put item in a bag.” After looping over *all items*, we want to know not just the *number of items* but (more importantly) *the total price of all items*, and also (probably as a side-effect) *the number of grocery bags required to store all items*. We include pool lengths to show that the simple number of loop repetitions is not always the only number we care about in a loop!

Part 3: Putting it into words (Square generator)

1. Again go **back** to Graphical Looping at <http://csforall.sri.com>.
2. Ask students to click the third link, **Generating Squares (in Scratch)**. Explain to them that what they are looking at is a *computer program written in the Scratch language*, and just like the comic book story about swimmers, this program is made of individual pieces—*statement blocks* instead of *comic book panels*—that you read in sequence, from top to bottom. “Once the statements are in sequence, just like the comic book, they

*tell a story*. In fact, the job of a computer program is to *tell a sequential story to the computer about what to do*. So let's figure out, what does this story tell the computer to do?"

3. Explain to students that they will learn about the first gold "When [Flag] Clicked" statement, and the two green "Pen up/Pen down" statements, when they start programming themselves. For now, we are interested in the *blue* action statements. Ask student pairs to physically act out these blue Scratch actions, in sequence, and determine what sort of pattern or shape they create. Give students several minutes to come to the understanding that this sequence describes *the perimeter of a square*—a student acting out the computer instructions should trace out a square on the floor (or on a piece of paper), with each of the four sides of the square being "ten steps" in length and each of the sides being separated from the previous one by a 90 degree turn.
4. Ask students if they can see how to use a *repeating pattern* to express the same set of instructions more efficiently. Listen to and guide the conversation appropriately: in the blue instructions, there is clearly a two-step sequence, *move* and *turn*, that gets repeated four times.
5. Then, project this screen, and click the "Show structure" button. Students can follow along by doing the same themselves. This demonstrates how we might organize the program into several repetitions of a basic "Repeat Unit", with a "before loop" and "after loop" section of the code surrounding it.
6. Next, click the button on the right called "With a repeat loop". This shows how the Scratch code on the left can be written more efficiently using a 'repeat' loop, using exactly the same structure students met in Part 2. If this is their first exposure to Scratch computer code, let them know they've just successfully interpreted—and more efficiently rewritten!—their first Scratch program!

Finally, ask students to **individually answer the questions in their review sheets**.

Names: \_\_\_\_\_ and

\_\_\_\_\_

### **Graphical looping - Worksheet**

Q1. The length of the pool is 10 meters. In one repeat, the swimmer travels two pool-lengths (from one side to the other side of the pool, and back).

Complete the following table:

Number of times swimming pattern repeats in the <b>'Results'</b>	Total Distance swum (in meters)
1	
2	
3	
100	
9500	

Q2. The swimmer has a goal of swimming 120 meters. The length of the pool is 10 meters. Using the swim generator, write how many times will you need to repeat the inner story (that is, swimming from one end of the pool to the other, and back)?

\_\_\_\_\_

Names: \_\_\_\_\_ **SOLUTIONS** \_\_\_\_\_ and

---

### Graphical looping - Worksheet

Q1. The length of the pool is 10 meters. In one repeat, the swimmer travels two pool-lengths (from one side to the other side of the pool, and back).

Complete the following table:

Number of times swimming pattern repeats in the 'Results'	Total Distance swum (in meters)
1	20
2	40
3	60
100	$100 * 20 = 2000$
9500	$9500 * 20$

Q2. The swimmer has a goal of swimming 120 meters. The length of the pool is 10 meters. Using the swim generator, write how many times will you need to repeat the inner story (that is, swimming from one end of the pool to the other, and back)?

$$120/20 = 6$$

---



Name: \_\_\_\_\_

**Graphical Looping - Review questions**

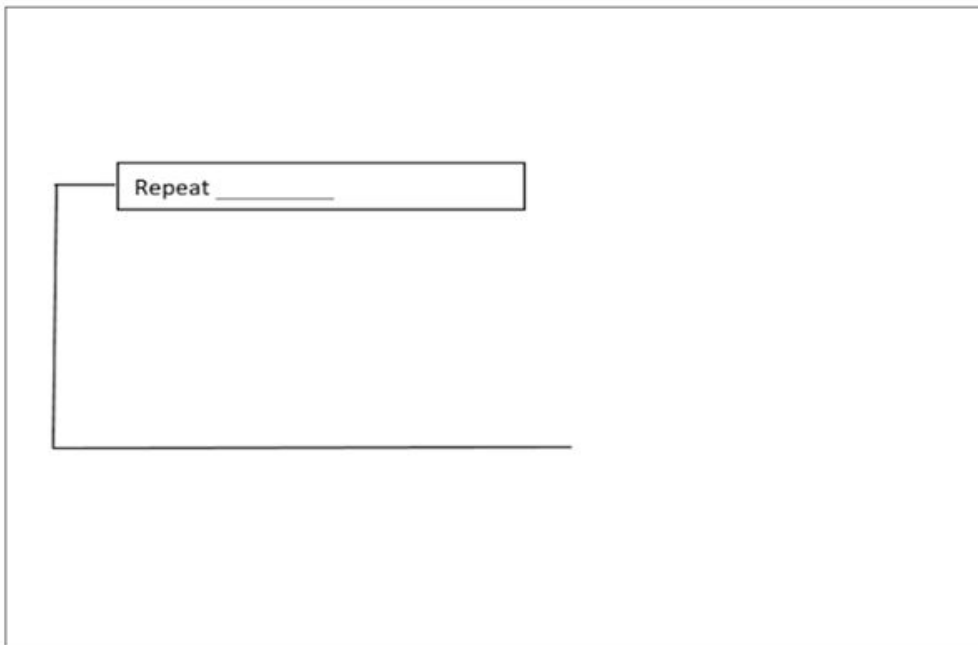
Q1. A pattern is shown below.



a. Draw the next step (step 4) in the pattern.

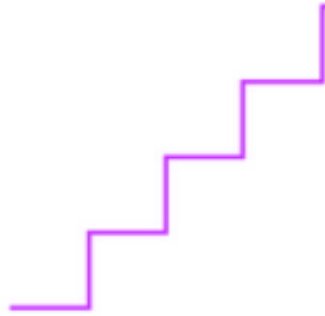
b. What will step 10,000 of the pattern look like?

c. Complete this drawing using the given shapes (triangle, rectangle and hexagon) to show what goes BEFORE, INSIDE, or AFTER the following REPEAT LOOP that describe a pattern-generator for the above pattern. The pattern generator should be able to generate Step 1, 2, 3, 4, .. 10,000 or any other step number of the pattern.



Q2. A **“Repeat Unit”** is the shortest part of a pattern that is repeating (and would go inside the repeat loop).

a. In the picture above, circle the **“repeat unit”**, and draw it here.



b. In the picture above, draw a square around the **“repeat unit”**.

