Coding Shapes

ACTIVITY OVERVIEW

STEM Focus Area: Robotics

Learning Goal: Demonstrate understanding of computational thinking concepts (sequence, algorithm, testing, debugging) & practice giving precise multi-step directions to solve complex problems.

Youth Learning Targets

- "I understand that robots follow commands."
- "I can control a robot with an algorithm."
- "I understand that there are better ways than others to write an algorithm."
- "I understand there is more than on way to write an algorithm to do something."
- "I understand I can write code quicker if I use loops."

LEARNING ENVIRONMENT

Activity Duration: 20 minutes

Group Size: Large or Small

Age of Youth: Grades 6-9

Guiding Question - What is the question to explore OR the problem or challenge to solve?

How can you guide a robot to complete a task by using computational sequences (called algorithms)? How can we make these tasks easier by using loops?

Throughout this activity, youth will:

- Create an "algorithm" designed to complete a specific task.
- Develop an understanding and observe how robots respond only to the commands given to them.
- Have an opportunity to "debug" or edit/fix a sequence of commands.
- Have an opportunity to utilize "loops" to simplify a code.

Facilitator Prep

- Become familiar with the sphero edu application.
- Ensure Spheros are charged before activity.
- Create the challenge cube dices before activity.

PREPARATION

Materials

- Sphero (one per group)
- Tablet (ipad or android) OR smartphone with "Sphero EDU" app installed (one per group)
- Optional: Projector & tablet optical adapter, to display tablet screen on projector
- Challenge Cube page 4

Room

Ensure each group has enough space on the floor for their Spheros to move and create their intended shapes. It might help to make barriers with bags or boxes to block Spheros from rolling into areas where you don't want them.

Stem Content Learning

- **Robots**: a machine that carries out a series of actions automatically, especially one programmable/programmed by a computer. All robots require an input and a response to the input.
- **Computational Thinking**: a problem solving process that involves logically ordering and analyzing data to create solutions to problems using a series of ordered steps.
- **Algorithms**: a sequence of instruction intended to complete a particular task.
- **Debugging**: correcting a part of an algorithm to better complete its goal.
- **Loop**: a repetition in the code by a specified amount *(eg. Loop 3x, loop 10x, etc)*
- **Blockly code**: often called block code; a visual coding language designed to be user friendly way to create Java code. Used in a variety of applications, including programming robots.

Inquiry

Your primary goal as a facilitator is to encourage youth to explore and observe ways to make more efficient code. You can prompt those discussions with questions like the following:

- How did you know that the code needed debugging?
- What would you need to do in order for the robot to correct its own code?
- How do you know if you have written the right code? Is there more than one correct or efficient way to write a code to complete a particular task?

Literacy Connection: Great books to get youth support learning about Robotics *(available on Amazon).*

- Read *Gabi's if/then Garden*
- Read *How to Code a Sandcastle*

INTRODUCTION TO ACTIVITY (15 MINUTES)

Start with a discussion the participants' experiences with computers and robots

- "Has anyone here seen or interacted with a robot? How do you interact with a robot? Does the robot really "hear" you speak? Does it "understand" what you are saying?"
- "Robots don't hear and understand the same way people do. They operate off "instructions," specific things they're preprogrammed to do. In order to accomplish a task, they need to have a series of instructions (called an **algorithm**) that they can reference."
- "We rely on algorithms in our everyday lives to complete tasks. Brushing your teeth requires an algorithm. What is are the series of steps you need to brush your teeth? What would happen if you did them out of order? "
- Ask for a volunteer to stand. Instruct the volunteer to walk around the room (or table or friend). When they are done walking in a circle, instruct them to do it again, using the exact same words you did the first time. When they are done, instruct the same way again. Then two more times, using the exact same words each time.
- Discuss with the participants: What would have been an easier/more efficient way to have instructed the volunteer to complete this task? Would it have been better to instruct them to have circled the room four times? Explain that when known in advance how many times times a task should be repeated, it would be easier to just state the number of times with the instruction. In coding, repeating an action many times is called looping. This will be useful when creating the shapes with the robots

Introduce the Sphero robots

- Show the youth how to open the Sphero Edu app and start making a code.
- Show the Start button, then explain that when that's pressed, the robot will do what the code tells it to.
- As a group, create a code for the robot to move forward.
    - Which direction should it go? How fast should it go? For how long should it roll?
    - Run the code to show how the robot responds.
    - Then as a group, try to come up with a code for having the robot drive in a direction, then return to the same spot.
    - Run the code to show how the robot responds.
    - Then show the group how to create a loop with the robot to repeat the code a certain number of times.
    - Inquiry: Is the robot doing what it's supposed to do? How can we make the robot move more smoothly? How can we make it run more precisely?
- Explain the activity: when divided into groups, they will try to create the shape rolled on the challenge cube dice using a code.

ACTIVITY ENGAGEMENT (20 MINUTES)

- Divide the participants into groups of 2 or 3.
- First, challenge them each to create a square.
    - How many sides does a square have? What is the angle of a square's corners?
- Once they figure out how to make a square, allow one of them to roll the dice which will select their challenge.
    - If they're having trouble figuring out the correct angle, help them to figure it out with the following formula:
        - *360 degrees / [number of corners] = Input Degrees for given shape*
- After a couple groups complete their shapes, invite them to trade tablets, read the code, predict what shape will be based on the code, then run the code to confirm their prediction.

FINAL REFLECTION AND RELEVANCE (5 MINUTES)

- "Was this easier or harder than you anticipated? What made it challenging/easy?"
- Have the groups compare code from the more complex shapes.
    - "Are these codes the same or different? What does this mean? How do you know if a code is correct? Is there more than one way to write correct code?"
- "Do you ever use algorithms in your everyday life? What are some examples?"
    - *Brushing teeth, making breakfast, taking a shower are all processes that use a sequence of step to accomplish a goal.*
- "What are some professions or circumstances you can see algorithms being useful?"
- "Why is looping important? What would happen without the loop? Would it affect the robots?"
- "What are some things in your home that use code/algorithms?"