CLEAN ENERGY BRIGHT FUTURES
A Program of BEF

LESSON PLAN

Solar SPRK+

Lesson 2: Sphero Edu Coding

AUTHOR
Deb Frankel
dfrankel@sherwood.k12.or.us

DESCRIPTION
After working on a few Scratch drag and drop programs, participants will transition to Sphero Edu, a comparable drag and drop program for Lesson 2 to prepare to program a Sphero SPRK+ ball to navigate through a maze.

GRADE LEVEL(S)
6, 7, 8

SUBJECT AREA(S)
Computer programming

ACTIVITY LENGTH
5 hours

LEARNING GOAL(S)

1. Students will use block programming and Oval Language in order to make a SPRK+ perform a variety of motion-oriented tasks.
2. Students will combine motion and sound functions using block programming and Oval Language.
3. Students will track useful pieces of sequence code that they can reuse in order to get their SPRK+ to navigate a lengthy maze.

STANDARDS REMINDERS

- Use the array of challenges presented by Sphero as a way to continue having students practice using language such as “criteria for success” and “constraints.” This will help them to identify these much more easily in open ended-engineering challenges.
- Additionally, have a discussion with students about how lines of code used to solve previous problems may come in handy later, determining what types of situations this may occur in. This highlights the importance of noting different methods for reaching success to build background knowledge in problem solving.

CONTENT BACKGROUND

STUDENT BACKGROUND

Students participating in this lesson should be familiar with drag-and-drop programming. This lesson comes immediately after working through Scratch, a similar program to the one that will be used in this lesson

EDUCATOR BACKGROUND

Educators leading this lesson should be familiar with Scratch and Sphero Edu.

REQUIRED MATERIALS

HANDOUTS/PAPER MATERIALS

- Scratch workbook

CLASSROOM SUPPLIES

- N/A

ACTIVITY SUPPLIES (PER GROUP OF 3-4 STUDENTS)

- Computer or tablet that runs Sphero Edu

LESSON PROGRESSION

PLANNING AND PREP

Teachers should work through and be familiar with the programming outlined in the lesson.

LESSON SEQUENCE

Working with Sphero Edu

1. Sphero Edu Block and Oval Codes

   a. Students will need to first download the Sphero Edu App (formerly Lightning Lab) and make an account.
   b. In order to explore some of the functionality of Sphero code, have students work with the "Sphero Edu Block and Oval Code" Worksheet. In this worksheet, they will write down block code from any Sphero Edu program and copy its Oval counterpart.
2. Sphero Edu Startup
   a. Using the "Sphero Edu Startup" worksheet, have students engage in an initial steps to acquaint them with the SPRK+: determine guidelines for usage, wake up the SPRK+, enable Bluetooth, connect the robot, disconnect the robot and disable Bluetooth. They will sign off on each step as they go along.
   b. Have students sign into account and click on Drive. Next have them go to AIM and orient the blue tail light correctly (towards you). Customize the color and drive (speed). Finally have them test out WASD
   c. Next, students will be writing down information on their worksheet about different programs by Sphero. When logged in, go to "By Sphero" under "All Programs." Click on each of the eleven programs listed on the "Sphero Edu Startup" worksheet. Once selected, make changes to the blocks and press start again. Observe how this changes the function of the program.
   d. As a group, write one sentence using your observations about what the program does as well as one sentence about the result of your change to the block.
3. Sphero Edu Activities: Blocks 1, 2, and 3
   a. Students will complete the following activities in Sphero Edu that will help them guide their SPRK+ through a maze.
   b. Assign "Beta Testers," potentially students who have already completed some of these activities, to walk the room to check for completion.
   c. Blocks 1 Activities Objective: Use drag and drop blocks to program a SPRK+ ball to travel in a square and travel in a square with loops. Look at Toss Game blocks to learn about the accelerometer. Incorporate multiple blocks from the three programs to design your own program.
      i. Go to Activities -> All activities -> Blocks 1.
      ii. Watch all three videos as a class and then write own programs
      iii. Complete Blocks 1: Intro: Program a Square
      iv. Complete Blocks 1: Intro: Square with Loops
      v. Complete Blocks 1: Intro: Toss Game
      vi. As a group, have the Beta Testers check off your program and SPRK+ ball running each of these programs.

d. Blocks 2 Activities Objective: You will start by discovering the different methods to control Sphero's lights, and then build some amazing programs to put the lights to work. You will create a program that powers a robotic Jack-o'-Lantern, turn Sphero into a sensor-based spinning top, and get creative with long exposure light art.
    i. Go to Activities -> All activities -> Blocks 1.
    ii. Complete Blocks 2: Lights: Introduction
    iii. Complete Blocks 2: Lights: Jack-o-Lantern
    iv. Complete Blocks 2: Lights: Spinning Top
    v. Complete Blocks 2: Lights: Long Exposure Lollipop
    vi. As a group, have the Beta Testers check off your program and SPRK+ ball running each of these programs.

e. Blocks 3 Objective: This last block will have students investigate how to combine more of the above functionalities to be responsive to motion added to the SPRK+ through the addition of variables to the coding process.
    i. Go to Activities -> All Activities -> Blocks 3
    ii. Complete Blocks 3: Variables: Introduction
    iii. Complete Blocks 3: Variables: Speeding Ticket
    iv. Complete Blocks 3: Variables: Hot Potato
    v. Complete Blocks 3: Variables: Fortune Teller
    vi. As a group, have the Beta Testers check off your program and SPRK+ ball running each of these programs.
    vii. After you've been checked off, make your own program incorporating all three programs. We'll share next end of class and/or next STEM class.

TEXT SECTION

- Students will create sections in their notebook where they can write examples of code as well as findings for specific needs. On the top line of 12 pages write the headings below. Leave three or so blank pages in between each heading page. They may want to create tabs to find each page.:
    - Get Started
    - Actions
    - Controls
    - Operators
    - Comparators
    - Sensors
    - Events

  - Variables
  - Functions
  - Error Codes
  - Known Issues
  - My Programs
- In the inside cover of their spiral, they will need to write the syntax rules of the Oval Language before writing any code. Here are the basics:
  - *"void startProgram()"* starts a program
  - *"{ and }"* will contain all program code, except for global functions and variables
  - *"//"* indicates a comment but does not affect the program logic
  - *";"* ends a statement
  - *"()"* contains a value(s)
  - *" ," separates values*
  - *" "* a tab space starts the first statement in a program
- For the Text 1 Activities, students will need to make sure they are tracking important lines of code throughout their exploration. The progression for this section is below:
  - Go to Activities -> All Activities -> Text 1
  - Complete Introduction
  - Watch Video: Text 1: Hello World Egg Toss
  - Go over Tips and Syntax
  - Complete Text 1: Hello World (Don't copy – Write on own)
  - Complete Text 1: Hello Square! (Don't copy – Write on own)
  - Complete Text 1: Egg Toss (Don't copy – Write on own)
  - Complete Text 1: Your Own Program; or select one from Text 2 or Text 3
- Specific to the Hello World activity above, follow the process below:
  - Open Hello World
  - Like square with loops, repeat the Hello World line four times
  - Add Engine Drive By (or sound of your choice)
  - Start program with a heading offset (like you did before) and add heading so your robot travels in a loop
- Below are helpful pieces of code for your process:

Goal of Sequence	Sequence Code

Getting Started	{
	setRgbLed(0, 0, 255);
	controlSystemTargetSpeed = 60;
	delay(1);
	controlSystemTargetSpeed = 0;
	}

Add While Loop	int _loopCount_1 = 0; // 1 is the unique value
	while (_loopCount_1 < 4) // 4 is the # of loops
	_loopCount_1 = _loopCount_1 + 1;

Add Drive-by sound	playSound(2, 6); // engine drive by sound
	delay(0.05);

Add heading change	controlSystemTargetYaw = 270;
	Make the robot turn:
	controlSystemTargetYaw = controlSystemTargetYaw + 90; //add 90°
	delay(1);

Set Speed (Action); make sure to input a speed in the blank between 0 and 255, then put a delay for how long you want it to move	Set Speed: controlSystemTargetSpeed=__
Stop (Action); always put the number 0 after the equal sign so it wont move.	Stop: controlSystemTargetSpeed=0
Set a color (Action); This is in RGB format, using a number between 0 and 255 for each blank within the parentheses	Set Color: setRgbLed(__, __, __)
Delay (Controls); the highlighted number represents the delay in between your actions. This can be whole or decimal.	Delay: delay(1)

Sound (Action); the numbers inside the parenthesis will come from a chart of numbers equivalent to what you want on the Oval Wiki	Sound: playSound(,)
Set Heading (Action); In the blank space put a bumber between 0-270, this is how you turn. • 0= straight • 90=right • 270=left • 180=back	Set Heading: controlSystemTargetYaw=__

Loops (Controls); adjust the numbers in the sequence to the amount of loops you want in your text	int _loopCount_int# = 0;
	while (_loopCount_int# < #loops)
	_loopCount_int# = _loopCount_int# + 1;

ASSESSMENT AND EXTENSIONS

FORMATIVE ASSESSMENT

Teacher should be making observations and notes throughout the student investigations. A basic understanding of algorithmic thinking and problem-solving will be important to integrate the coding into modeling and simulating a rover environment.