Overview: This activity is all about getting Sphero through a maze via precise programming. Getting Sphero up and running and then developing this program will involve lots of test runs and failures. While the basic activity may be intriguing, the real exploration comes in the various challenges that can be added to customize the experience. Students experiencing difficulty and frustration should be encouraged to try new things, consult resources and fellow students rather than given quick answers.

Objectives:

-Learn how a new technology works by ***tinkering*** with it.

-Use block code to create a program for Sphero

-Use trial and error to measure time and speed needed to create a successful ***algorithm*** for ***autonomous*** navigation through the maze

Vocabulary:

***Tinker***: Play with or handle an object. Motivated by curiosity.

***Algorithm:*** A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

***Autonomous***: Action on its own, without intervention.

Materials needed:

Handouts for binder, Sphero Bolts, tablets with Sphero Edu app installed, print-outs/cut outs of maze tiles, challenge cards, color gates and event stars. Cardboard ramps for the elevation challenge

Activity:

Intro: Demonstrate a Sphero through a maze with a preprogrammed algorithm. Survey the students, how may have programmed Sphero before today? Introduce challenge, “You will need to create an algorithm that will allow Sphero to autonomously navigate a maze you design. To do this you will need to tinker with Sphero and with The block coding environment of the app, Sphero edu. You will also need to use trial and error to determine the speed and time needed for Sphero to travel your maze”

Hand out materials and Spheros to programmers. Each student or pair of programmers will need:

-1 sphero Bolt

-1 tablet with Sphero Edu App installed

Starter set of maze tiles:

-1 set of start/finish maze tiles #1 and #2

-3 straight path tiles #3

-3 90 degree turn tiles #4

\*Have plenty of additional #3 and #4 tiles available for any programmers that want them.

Exploration:

Note to instructors and student workers. This activity is designed for programmers to learn through trial and experimentation. While support from facilitators is necessary, efforts should be made to avoid “telling” or “showing” how to do something. programmers should be encouraged to keep trying new things instead of being told what to do. See the useful questions section for prompts that can be used to help programmers create their own knowledge.

Programmers use handouts, posted instructions and support from instructors to:

-Create a maze

-Connect Sphero Bolt to tablet with bluetooth

-Open a new block code program in Sphero Edu

-Create a program in Sphero Edu and test it on Sphero

-Calibrate Sphero using the “Aim” setting in Sphero Edu

-Test and iterate to create an autonomous program to navigate Sphero from Start to Finish

Extensions:

These materials are available for programmers who progress quickly through the activity OR who have previous experience with Sphero or have already made a maze for Sphero.

Challenge Cards:

These cards can be handed out to or selected by students. Instructors can customize the challenge for the student as needed.

Challenge Card #1

-Program Sphero to automatically return to the start before each trial run

Challenge Card #2

-Add one or more non-90 degree turns (maze tile #5)

Challenge Card #3

-Add one or more curves

Challenge Card #4

-Add one of the ramps to your maze

Challenge Card #5

-Create a new maze and program it using text-based coding language

Challenge Card #6 (for seriously high-flyers)

-Using the intersection tile (tile #7), create a maze with multiple solutions (at least 3) and program Sphero to randomly choose a solution each time through. This is boss mode.

Challenge Card #7

-Program Sphero to change color based on its speed

Challenge Card #8

-Use [ChatGPT-4 to write JavaScript code for your Bolt](https://edu.sphero.com/cwists/preview/71443x) (\*\*\*Use Klyn’s computer for this)

Challenge Card #9

-Build a maze that will allow you to program Sphero to complete it using one loop block/function that repeats 6 times

Color Gates:

These are strips of colored paper that can be added to the mazes. Place a color gate (or 3 or 5 depending on the student’s need for challenge) on the maze and instruct the designers to either:

* Program Sphero to show the color of the color gate BEFORE it crosses the gate (easier)
* Program Sphero to show the color of the gate AS it crosses the gate but show white the rest of the time (more difficult)

Event Stars:

These star-shaped cut-outs can be placed at a desired location on the programmer’s maze. Each star will describe an event that the programmer can incorporate into their algorithm to occur at the exact location of the event star.

Star #1

Dance Party~ Have Sphero pause here and use movement, light and sound to appear as if dancing at a wild party

Star #2

Speed Trap~ Sphero MUST be traveling slower than speed 15 when it crosses this star

Star #3

Speed Demon~ Sphero MUST be traveling faster than speed 40 shen it crosses this star

Star #4

Nap time~ Sphero is tired! Use the LED display, sound, and motion to show us that Sphero pauses here to take a nap.

Star #5

Create your own event~ Make something cool and interesting happen here, you decide what it is!

Sharing out:

For the last 10 minutes, students should take turns viewing each other's mazes and solutions. Instructors will use this time to ask questions that prompt students to synthesize learning (see the questions below for suggestions)

Useful questions to help on their exploration:

Why do you think this isn’t working as expected?

Have you checked your handout or the board for useful tips and tricks?

What were you hoping would happen?

What happens first, next, next …last in your program? What’s missing?

Can you read your program to me? Did you notice anything missing?

Remember, a computer algorithm can’t “make up” additional information, or make connections and inferences like a human brain, are there any instructions missing?

Did you “aim” the device?

Useful questions to help programmers synthesize the experience into future skill building:

What frustrated you about this activity?

What did you notice about someone else's maze and program that you thought was interesting?

How do you feel when an activity is too easy? What do you do when something is too easy?

How is communicating instructions to a programmed device (like Sphero) different from communicating with another human?

What is something you think is important for new programmers to know about an algorithm?

How did it feel to tinker and learn how the Sphero works compared to being told or shown how it works?

What was a problem you encountered in this activity and what did you do to try and solve your problem?

What does it take to be a “good” computer programmer?

If you were going to teach your grandparent how to program a Sphero, what do you think would be the most difficult part to teach?

How was programming Sphero similar/different to other programming experiences you’ve had?