BOTS FOR KIDS

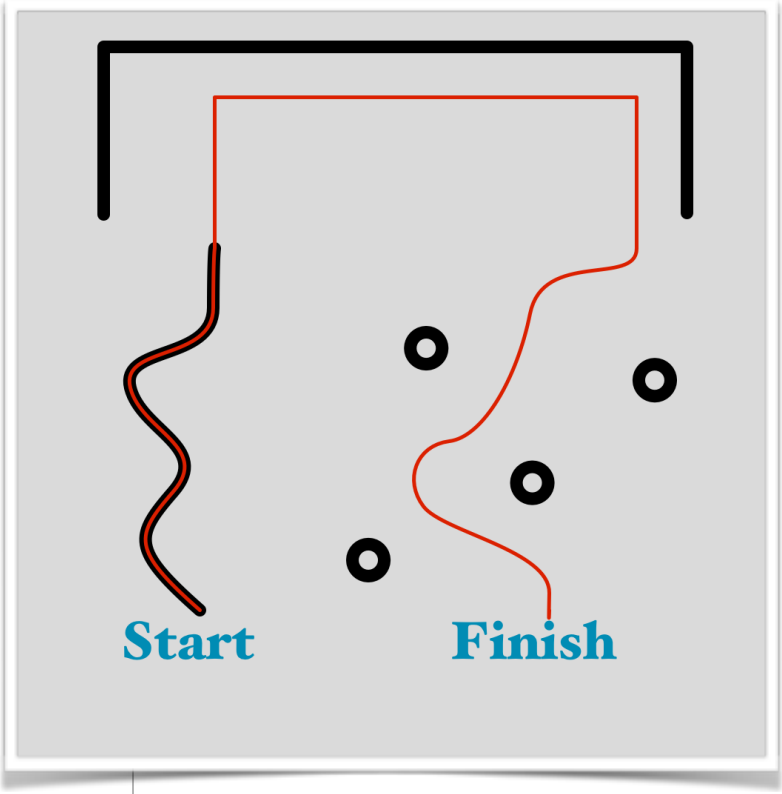
CHALLENGE! | Coding the Trials | Lines | Distance | Maze | Switching

CHALLENGE!

Your bots are built. You know some code. This week, we will program the bots to get through an obstacle course. Can you make it through the trials?

Trial 1: The Line

The first trial will be for the bot to automatically follow a line on the floor. The bot will be placed at the beginning of the line, and must follow the line to the end all on its own. We will be working on this challenge together as a class. At the end of the line, switch modes to the second trial!



Trial 2: The Box

The end of the line will point the bot towards a wall. The bot must turn around within a rectangular area without hitting the walls using the bot's distance sensor. This trial must also be done automatically, once the mode has been set. Once the bot has turned around and left the wall area, it meets the final trial.

Trial 3: The Maze

To end the challenge, the user must manually control the bot with the computer keyboard. Try not to knock over any of the towers, and guide your bot to the finish line.

At the finish line, stop the bot. Success!

Coding the Trials

Don't worry, you don't have to do this all on your own. We will go through the first part together. The rest of this newsletter will also provide many hints for writing the code. You can team up with someone, and feel free to ask a more experienced person for help.

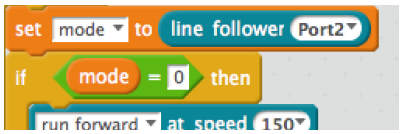
In addition, you will be provided with mBlock code that will give you 3 modes and let you switch between them with the 1, 2, and 3 keys on your keyboard. Each sprite will be a mode for you to program.

The Line

To follow a line, we use the line following sensor at the bottom front of the robot. Each side has a light that you can't see, but that the robot can detect and use to sense the brightness of the ground under it. A Roomba uses these same sensors to detect edges and dirt, but we'll use them to follow a line.

The first thing you need to do is set up a variable. A variable lets the program remember something, and gives it a name. In the "Data&Blocks" panel, press the "Make a Variable" button and type "direction" to make a variable with that name. It will also create a few blocks for setting and getting the variable.

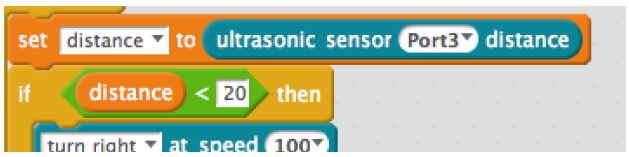
Now we can read the value from our line following sensor, and make decisions based on what it tells us.



Hint: The line follower module uses 0 for forward, 1 for left, and 2 for right.

The Box

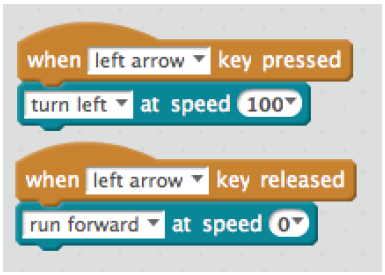
To escape the box, we use the distance sensor. It can be used almost the same as the line sensor, except that it only gives us one value: the distance to the wall, in centimeters. Create a new variable for it, and tell your robot to do something in response.



The Maze

In the maze area, you will control the robot yourself, using your keyboard. We wrote a program last week that allowed the robot to move forward and turn based on pressing keys, but there was still a lot of crashing.

One of the problems was that once a key was pressed, the bot kept going until you told it to do something else. For more precise control, you can detect when a key is released using the "When [space] key is released" block. When a key is pressed, perform the movement, and when it is released, stop the movement.

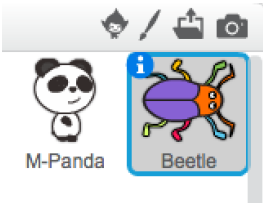


You can add any other moves you think will help you.

Switching

The trials come in three parts, and each part is a different program.

To split our program up, we will put each program in a different sprite that will be provided. If you want to do this yourself, use the "new sprite" button.



The sprites will send messages to each other so all of them know what mode they're in.

It will be your job to fill in the "forever" loop. Good luck!