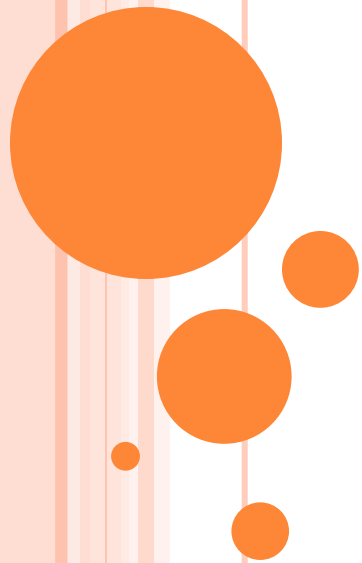


Maze Challenge



Pre-Activity Quiz

- 1. What is a design challenge?**
- 2. How do you program a robot to move 2 feet forward in a straight line?**

Pre-Activity Quiz **Answers**

1. What is a design challenge?

A design challenge starts with problem you want to solve. You think through the challenge logically and plan a design solution. Using suitable and available materials and following the steps in the engineering design process, you create, test and evaluate your best solution to the challenge. As necessary, you iterate (repeat) this process until a successful design is achieved.

2. How do you program a robot to move 2 feet forward in a straight line?

Calculate the distance that the robot moves forward for one motor rotation. Let's say, it moves X inches with one motor rotation. Then divide 2 feet, which is 24 inches, by X , that is, calculate $24/X$ and program the motor to rotate $24/X$.

What Is a Design Challenge?

Day 1 - 50 minutes

In our modern world, challenges are everywhere!

How can we waste less? How can we harness solar energy and other renewable energy more effectively?

How can improve transportation? How can we build roads and bridges? How can we design a house that is not too expensive? How can we build smarter cars? How can we use technology to make our cars safer?

These are big ideas that engineers and scientists work on to help improve the world we live in.

We will investigate how to complete a few design challenges through two different methods:

- Creating different robot designs to help it better complete the challenges
- Creating programs to help the robot complete the design challenge

Your Engineering Challenge:

To build and program a robot to travel through a maze.

- We will look at different robot designs (with and without sensors) in order to determine which is **more reliable** and which is **faster**.
- Both the **design** of the robot and how it's **programmed** are important for this challenge!

Let's Look at Some Basics

- Before we start the design challenge, it is important to understand how and why we need to be careful with designing the robot and designing its program.
- Also, it is very important that we understand how a robot follows instructions so we can understand how to program it.
- Let's do an activity to help us understand two important ideas:
 1. How a robot follows instructions
 2. The importance of sensors

Let's Look at Some Basics (continued)

Maze 1 is a 3-ft wide path in a corner of the classroom.

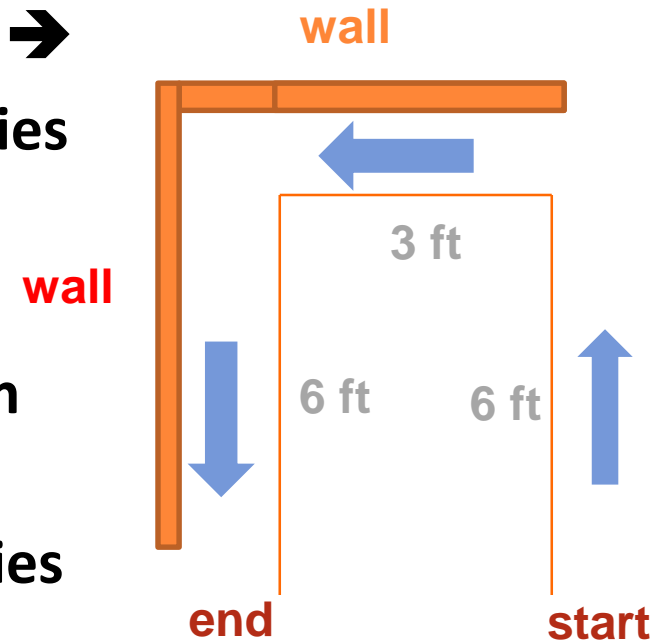
It looks like this diagram →

Use tape (or boxes) to mark the boundaries shown in fine lines.

Blindfold a student “robot” and have him stand at the maze start.

Have a student “programmer” give a series of commands to instruct the “robot” to complete the maze *without touching the maze edges* (keeping hands at his/her sides).

Command examples: go forward X steps, turn left, etc.



Let's Look at Some Basics (continued)

What did you observe?

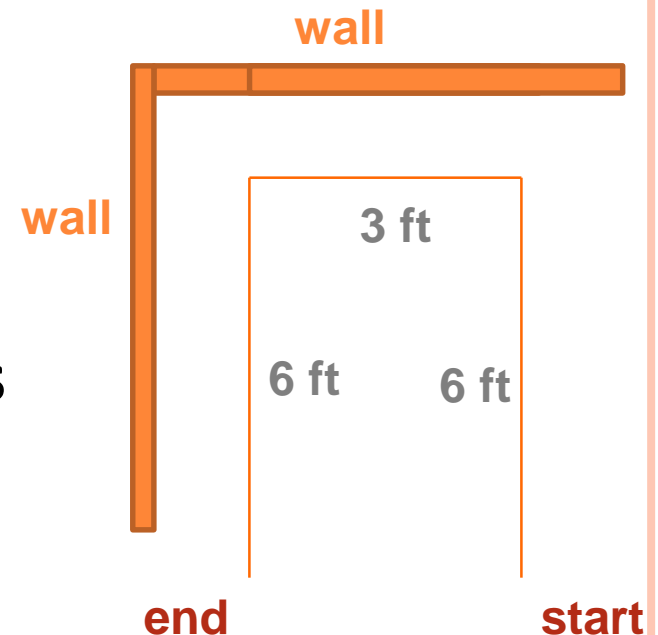
- Did the “programmer” get the blindfolded “robot” through the maze?
- Did the “commander” always tell the “robot” to go the correct number of steps?

Often, it is easier for the “programmer” to give instructions if s/he can tell the “robot” to go forward until s/he senses something

By doing it that way, the “programmer” does not have to worry about telling the “robot” exactly how many steps to move.

Let's Look at Some Basics (continued)

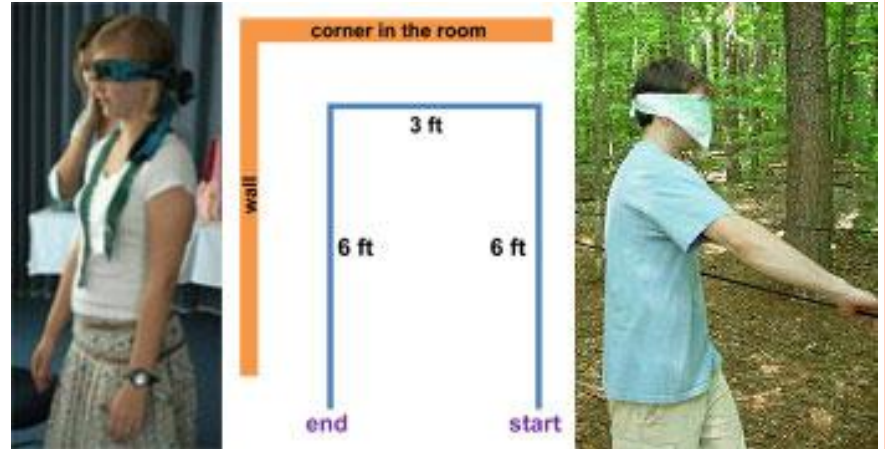
- Choose a different volunteer “robot” to be blindfolded.
- This time, permit the volunteer “robot” to stretch his/her hands out in front to sense when s/he is approaching a wall.
- Have another student give commands to instruct the volunteer to get through the maze
- Now, commands such as “go forward until you sense a wall” are allowed.



Let's Look at Some Basics (continued)

What did you observe?

- Was it easier to give instructions the first time or second time?



- It is generally easier to tell a robot to go forward until it senses something (like a wall) than to tell it exactly how far it should go before turning. This approach also tends to be faster.

Let's Look at Some Basics (continued)

What else did you observe?

- Did you notice that the “commander” had to make sure to tell the “robot” **EVERYTHING?**
...**which direction to move, how far or how long to move, when to turn, which way to turn, etc.**
- Likewise, the LEGO MINDSTORMS EV3 intelligent brick must be given exactly the same information. **It knows nothing and will do exactly what you tell it to do.** If you make a programming error, the robot will follow that incorrect command, and the fault is with the programmer, and not the robot!

Before you start...

- Before we start our maze challenges, let's learn the **relationship** between motor (or wheel) **rotation** and the corresponding **distance** the robot travels.
- *Why is this important?* Because, when we have the robot travel the maze *without* any sensors, we must tell it **how much distance to move**.
- Use your results from the *Master Driver* activity to inform your programming! (Recall that this is the activity where you determined the distance traveled by the taskbot for every rotation of the EV3 motor.)

A Real Maze

Day 2 - 50 minutes

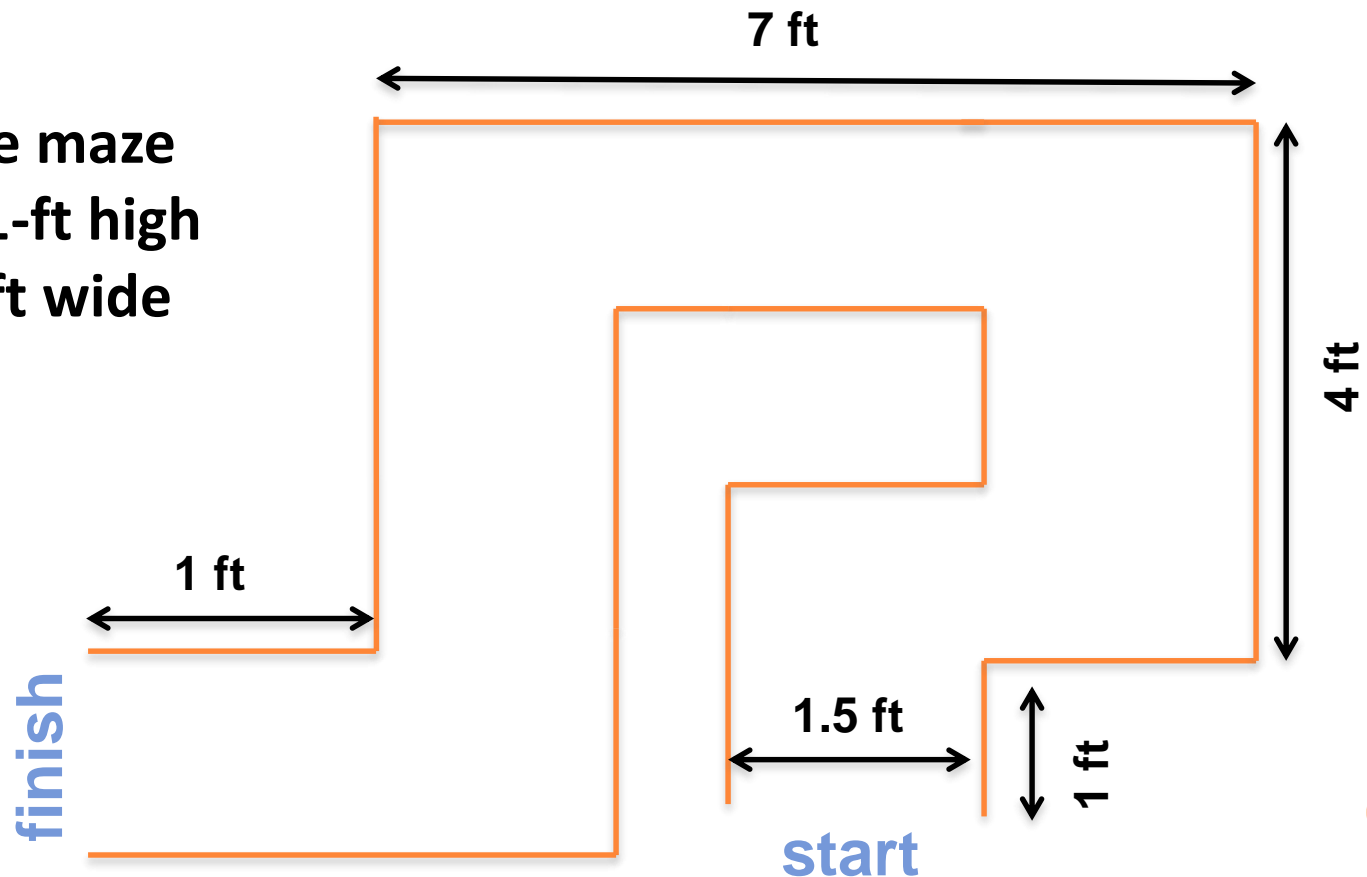
Challenge 1: Make the LEGO robot navigate the maze *without using any sensors*

Challenge 2: Make the LEGO robot navigate the maze using the sound sensor, the touch sensor, and ultrasonic sensor *at least once each*

The Challenges Maze

Suggested dimensions for the maze are shown in the diagram below, with “start” and “finish” locations indicated. ↓

Make the maze at least 1-ft high and 1.5-ft wide



Challenge 1

To program the taskbot to travel the maze *without* using sensors.

- For this first challenge, the robot must go through the maze *without using sensors!*
- How will you program the robot to do this?
- Think back to the *Master Driver* activity and what you learned about the **relationship** between the **number of rotations** of the motor and the **distance the robot moved**.
- You will be provided with a measuring stick or tape measure for this challenge. *Be resourceful!*

Challenge 2

To program the taskbot to travel the maze using the sound sensor, touch sensor, and ultrasonic sensor each *at least once*.

- For this challenge, your robot must use the touch sensor, sound sensor and ultrasonic sensor *at least once each* when navigating the maze.
- Think back to how we did basic programming with the touch sensor, ultrasonic sensor and sound sensor, and combine that information to get the robot to travel through the maze.
- In the robot documentation, find information about attaching sensors to the robot. Or, come up with your own designs for attachment also for this challenge. *Be creative!*

Post-Activity Quiz

- 1. What types of problems did you encounter when trying to complete the design challenges?**
- 2. How did you change the design of your robot or your programs to help you complete the tasks?**

Post-Activity Quiz **Answers**

1. **What types of problems did you encounter when trying to complete the design challenges?**

Calculating how many rotations to use in order to have the robot move a measured distance in the maze.

Recalling how to begin programming; how to program each sensor.

2. **How did you change the design of your robot or your programs to help you complete the tasks?**

We re-attached the sensors to position them correctly to receive the necessary input information.

Through troubleshooting we corrected some programming errors.

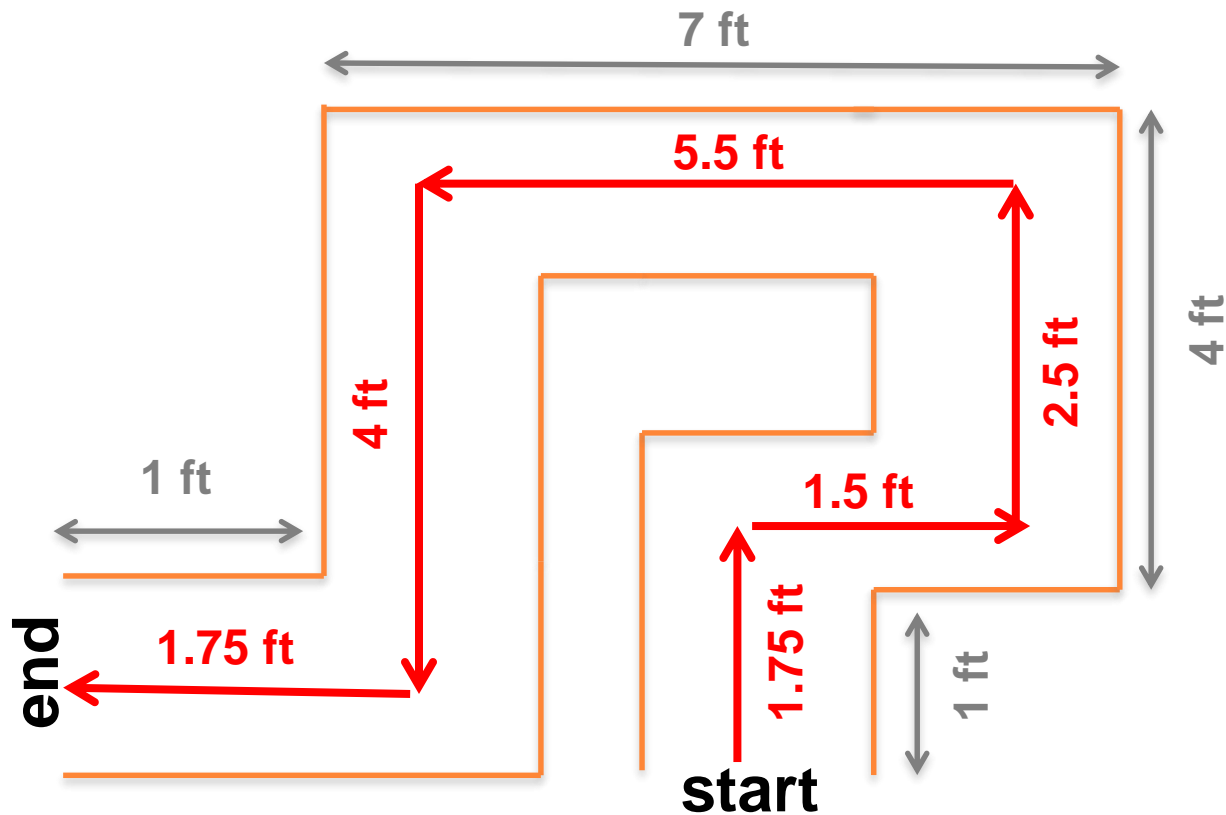
Challenge 1

Maze without Sensors **Solution**

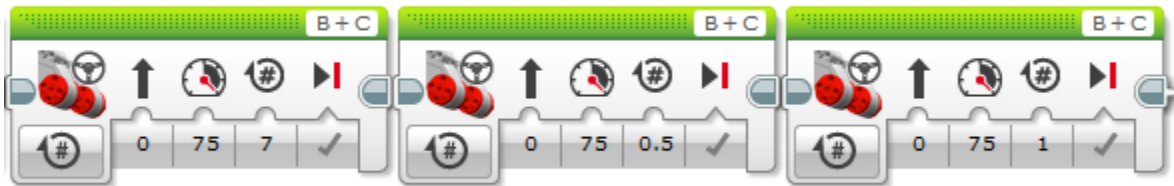
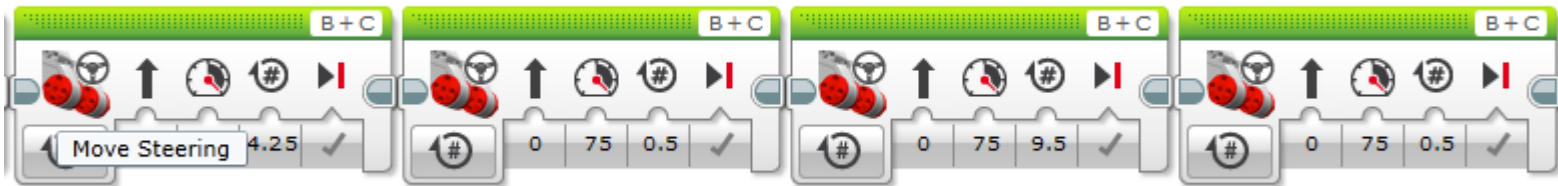
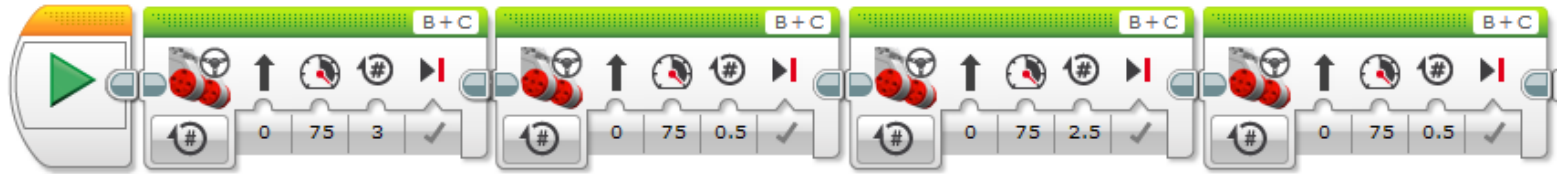
- Recall the relation between rotations and distance:
For each rotation, the robot travels a certain distance (inches or cms) and we can use *multiplication* to approximate how far the robot will travel.
- So, we must measure each section of the maze and guess how many rotations we need based off each distance!
- This may not work perfectly the first time, but the number of rotations can be modified so it eventually works.

Challenge 1 Program **Solution**

In an effort to keep the robot as far from the walls as possible, calculate path distances so that the robot stays in the center of each corridor (.75 ft away from the wall).



Challenge 1 Program Solution



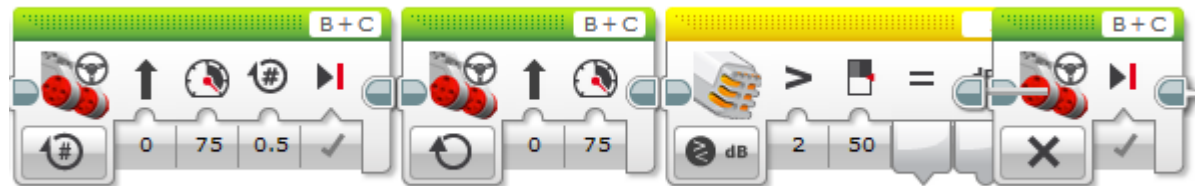
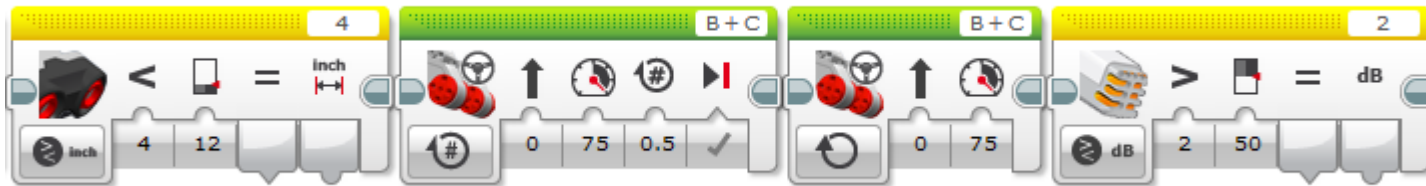
Challenge 2

Maze Using Sensors **Solution**

How to complete the maze by using the LEGO robot sound, touch and ultrasonic sensors:

- Remember, since we can use the sensors, we do not have to worry about distance and guessing how many rotations we need!
- For each section of the maze, we can program the forward and backward movement blocks using unlimited, and then immediately place a sensor block after the movement block.
- We must use the ultrasonic and sound sensor once in this program, then we can choose to use the sound sensor for the rest of the maze or the ultrasonic sensor for the rest of the maze or both.

Challenge 2 Program Solution



Note: Block 20 is not strictly necessary.

Vocabulary

design: Loosely stated, the art of creating something that does not exist.

engineering: The use of science and mathematics to solve problems to improve the world around us.

