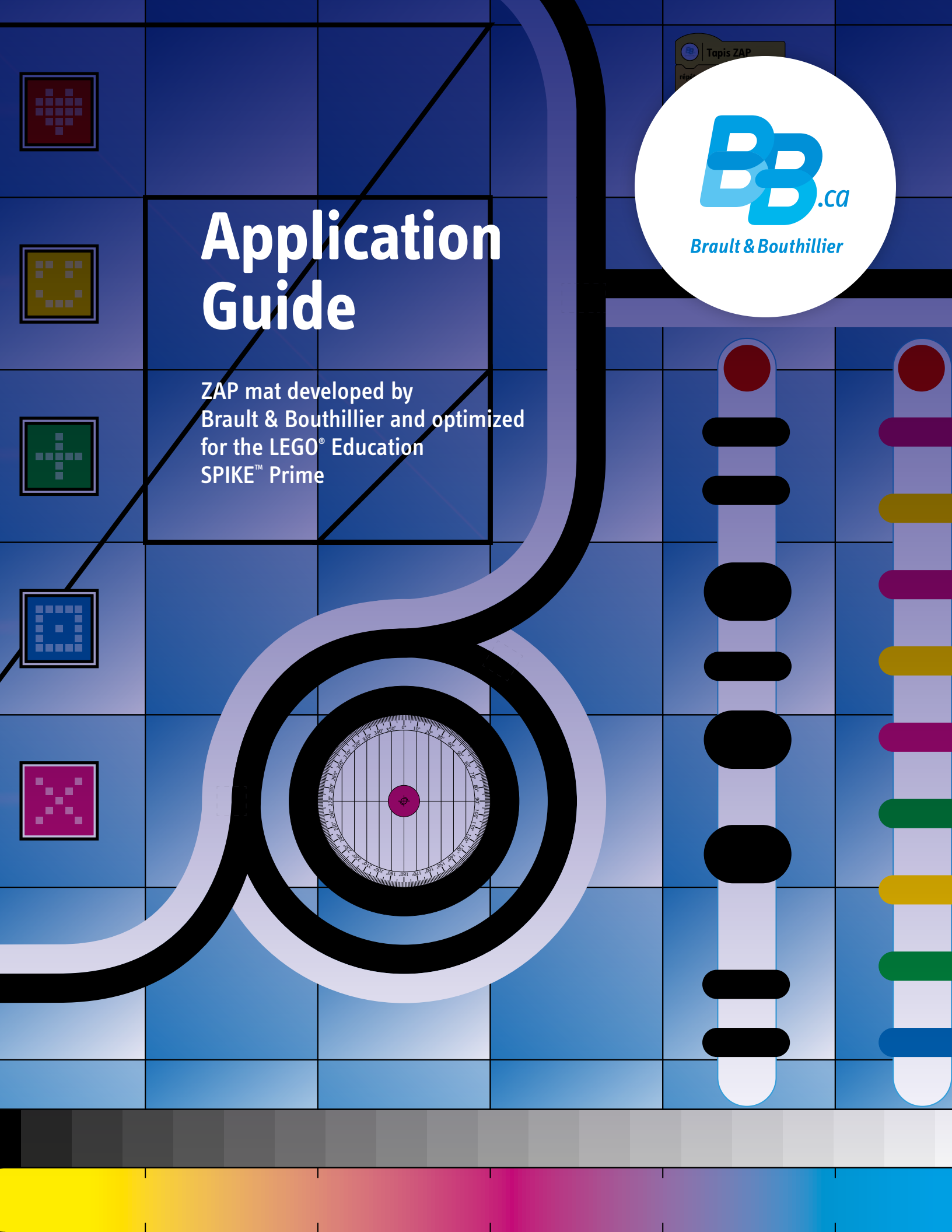
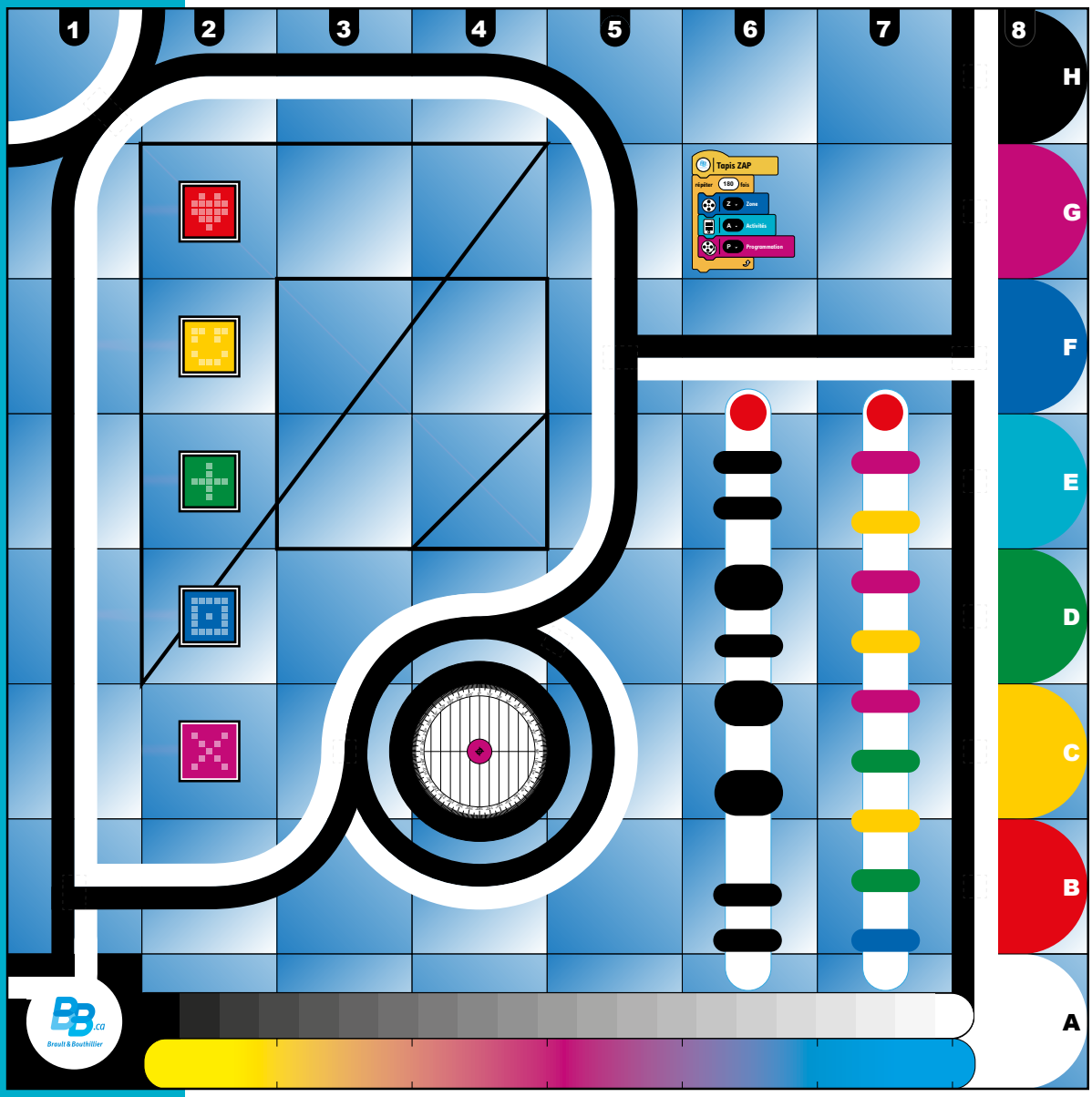
# Application Guide

ZAP mat developed by Brault & Bouthillier and optimized for the LEGO® Education SPIKE™ Prime

Brault & Bouthillier






Designed and written by
**Marc-André Caron**

# Activities for Inspiration

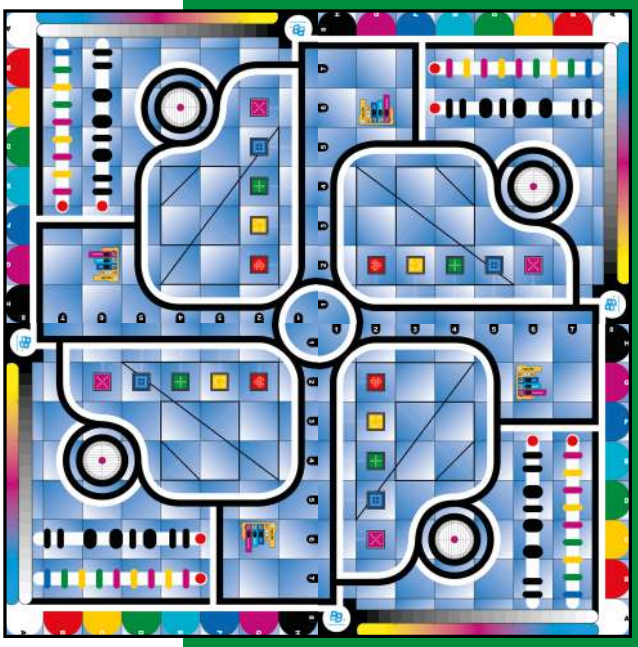
1. **Stop on Red**
   Move a robot forward and program it to stop when it detects a specific colour.
2. **Around the Square**
   Move a robot forward and turn it using the loop.
3. **Follow the Line**
   Make a robot move forward along the path (the black-and-white line)
4. **Find the Way**
   Following the black-and-white line, a robot changes its trajectory by detecting another colour.

# Introduction

The Programming Activities Zone (ZAP) is specially designed for use with the LEGO® EDUCATION SPIKE™ Prime set. This guide outlines four starter activities. However, you should not limit yourself to these four activities, as the ZAP mat has additional features that can be used for many other challenges.

## With this mat, students can:

- Estimate a distance using the 15-cm squares
- Locate themselves spatially using the coordinates in the margin
- Work on moving in squares and triangles
- Work on rotation angles using the protractor on C4
- Use the colour sensor with eight detectable colours
- Work with reflected light percentages using the black gradient on A2–A7
- Work on colour composition with the yellow-magenta-cyan gradient on A2–A7 using graphing features
- Have a robot count the black marks on A6–F6
- Have a robot count the coloured marks on A7–F7
- Follow the paths (black-and-white line) starting from the base on A1 over to, for example, one of the colours in column 8
- Place objects on the squares on B2–G2 and move them over to their colour in column 8, either in a straight line or following the paths



It is also possible to put two, three or four mats together to create an even larger challenge area. The paths will then be connected to the portion of the circle on H1, and the base on A8 will be connected to the path on H8. **The Find the Way** activity shows how to get a robot to change its trajectory.

This guide offers educational activities for introduction to robotics with SPIKE™ Prime and the ZAP mat.

Before using the mat, we recommend that you at least complete the **"Start"** activities in the SPIKE™ App and familiarize yourself with the **"Going the Distance"** activity in the supplementary lessons.

### ATTENTION

**This guide is not intended to be a complete training program or teaching guide that provides all the answers.** The activities presented in this guide are ideas for challenges that should be improved and integrated into other activities. You do not need to have all the solutions before you start; let the students surprise you!

# 1. Stop on red

## Scenario

When we walk down a hallway or drive a car, our eyes look around to help us find our way and, most importantly, tell us when to stop. If we see a stop sign or a red light, we know to stop.

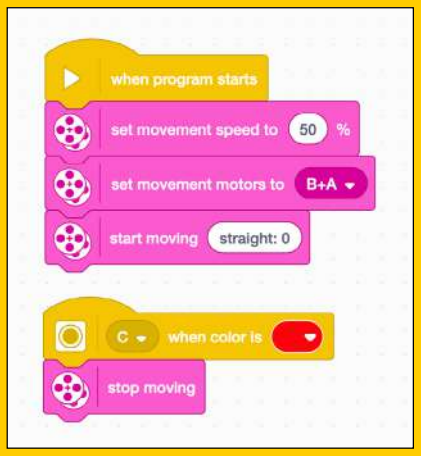
Robots can use a colour sensor to detect a colour and then follow their program to stop. We will be doing the same thing.

## Build

In the SPIKE™ App building instructions, use the Rhino plan to build a driving base or invent one of your own. Then, add the colour sensor and point it to the ground at a height of two beams.

## Program

Use the SPIKE™ App to program your robot to move in a straight line and to stop when it passes over red on the ZAP mat. You can use the example code to get started.




A two-stack program

## Discussion

**Questions to ask when returning to the group:**

→ Did all teams set up the colour sensor the same way?
- Why?
- What are the best techniques ?

→ Did you have trouble programming the colour sensor?
- Which port did you plug it into?
- Did you adjust your program accordingly?

→ Did you try different speeds?
- Was the robot as precise when moving faster?

## A Step Further

Now that your robot stops on red, why not try the following challenges?

→ Make the robot stop on a different colour.

→ Add sound once the robot stops.

→ Could the robot stop on yellow, start back up, and stop again on red?

→ Try programming the same thing with a single stack.
Is it possible?

# 2. Around the Square

## Scenario

A plane figure can be identified by looking at its number of sides, the length of these sides, and the angle that connects them. A square will have four equal sides connected by 90-degree angles.

**Here is a challenge that will allow you to work on three interesting aspects with your Rhino:**

1. how to move a specific distance forward
2. how to make a 90-degree turn
3. how to use a programming loop.

To do this, we will program our robot to follow the path of the square from F3 to E4.

## Build

In the SPIKE™ App building instructions, use the Rhino plan to build a driving base or invent one of your own. For this activity, do not add sensors.

## Program

Use the SPIKE™ App to program your robot to follow the path of the square from F3 to E4

**There are two possible methods:**

1. "Advance, turn, advance, turn, advance, turn, advance, turn"
2. Repeat "advance, turn" four times.

## Discussion

**Questions to ask when returning to the group:**

→ Did all teams use the same measurements?
- Speed, cm, rotation, degrees, etc.

→ Which speed provides a more accurate angle: fast or slow?

## A Step Further

**Now that you can move in a square, can you move in the following shapes?**

→ Octagon

→ Triangle

→ Heptagon

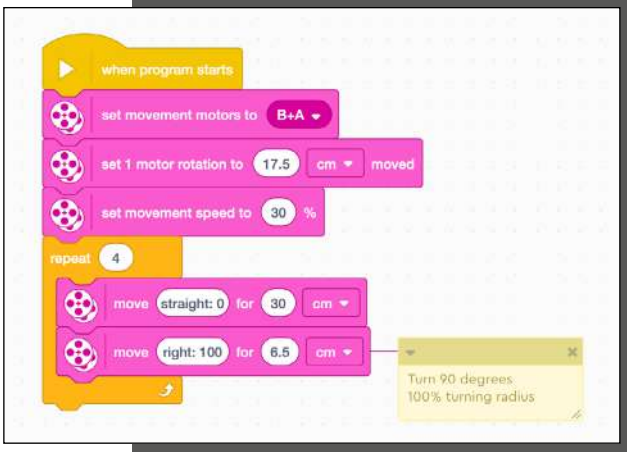
→ Pentagon

→ Five-pointed star

→ Is it possible to make your robot turn using the gyro sensor?

### Turning radius?

For more information, see the reference sheet at the end of this guide.




### How to make a robot turn 90 degrees on itself

Use the movement block set to a turning radius of 100 (right or left) and determine how far you need to turn to reach 90 degrees. Let students test their robots on the mat's protractor and have them share their methods and solutions.

# 3. Follow the Line

## Scenario

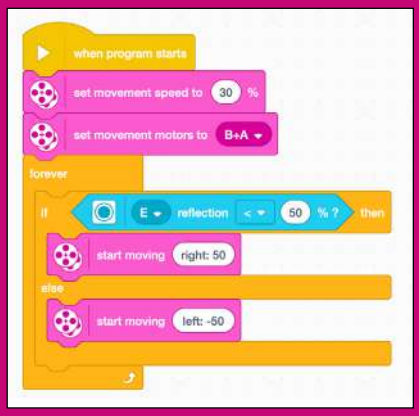
When we ride our bike on a bike path, we are not really moving in a straight line. If the bike moves a little too far left, we turn it a little to the right. If it moves too far right, we turn it a little to the left. Our eyes serve as sensors to guide us.
To steer a robot along a path, an easy solution is to draw a black line on the ground. The robot does not really see a line; it only detects one colour at a time. On the ZAP mat, we made a black and a white line that run alongside each other. We will be building and programming a robot that follows these lines while constantly correcting its trajectory–much like when we ride a bike.

## Build

In the SPIKE™ App building instructions, use the Rhino plan to build a driving base or invent your own. Then, add the colour sensor and point it to the ground at a height of two beams.

## Program

Use the SPIKE™ App to program your robot to turn to the right if it detects black (or less than 50% light reflection) and to turn to the left if it does not. You can use the example code to get started. Try the program on the ZAP mat starting at G1 towards B1 and observe what happens. Do you need to adjust the speed and steering angle? The goal here is a trial-and-error process to find the best solution.




Speed, percentage of reflected light, and turning radius can be adjusted to improve your program.

## Discussion

**Questions to ask when returning to the group:**

- → Did all teams modify their code parameters in the same way? What works best?
- → Were you able to complete the loop on the mat?
- → What happens when the robot gets to the 90-degree angle on B1?

## A Step Further

- → What happens if you swap the black and white in your code? Does the robot follow the line the same way? Does it move in the other direction?
- → How can you make your robot move faster and more smoothly while following the line? In the app's "Competition Ready" unit, try "Training Camp 3: Reacting to Lines" for an expert solution.

# 4. Find the Way

## Scenario

When we drive on a road, we always come to an intersection at some point. A stop sign tells us to stop, and we are forced to make a decision: go straight, turn right or turn left. This decision will depend on our destination; where do we want to go?

## The Challenge

**Getting your robot to follow a path, turn at intersections and stop at a specific destination is a rather complex challenge.**
**The steps can be summarized as follows:**

1. **Place your robot on A1 (starting area).**
2. **Following the path that intersects with C1, your robot needs to get to F5...**
3. **...and stop on the yellow rectangle.**
4. **Turn to the left and follow the line...**
5. **...and stop on F8 on the red rectangle.**
6. **Turn to the right and follow the line...**
7. **...and stop in front of the yellow box on C8. There is no rectangle in front of C8.**
8. **Turn and stop on C8 (+ do a victory dance!)**

## Prerequisite

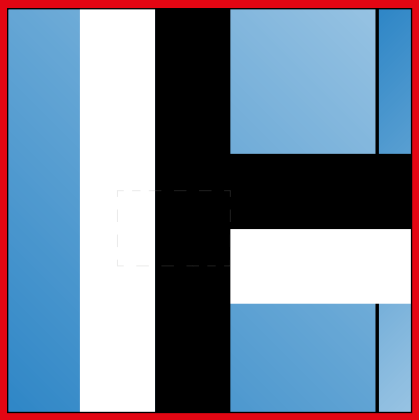
You must have completed the previous activities.

## Build

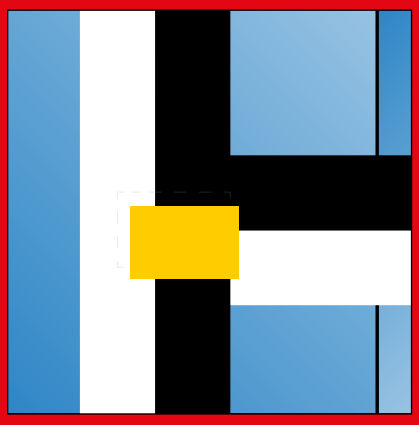
In the SPIKE™ App building instructions, use the Rhino plan to build a driving base or invent your own. Then, add the colour sensor and point it to the ground at a height of two beams

## Prepare the ZAP Mat

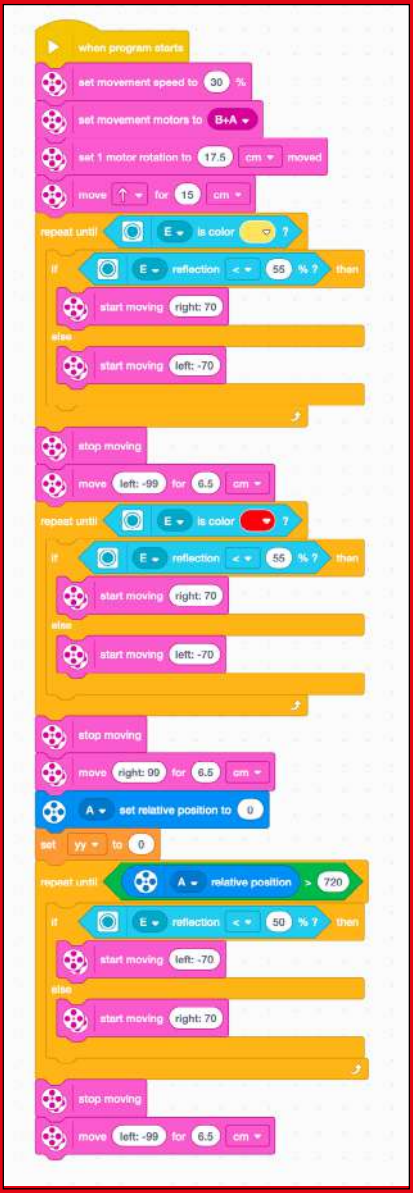
You will notice dotted rectangles on the ZAP mat's black-and-white line, specifically at intersections. Use construction paper in the same colours as the mat (such as red and yellow) to cut out rectangles of the same size. Using tape or sticky tack, stick a yellow paper rectangle at the intersection on F5 and a red rectangle on F8. These will serve as your road signs.



Locate the dotted areas along the paths.



By covering them with coloured paper, you are creating markers for your robot.




An example of programming a complex challenge

## Program

If you think about it, the first intersection (without a coloured paper) on B1 is easy to get through. Instead of following a line, simply move forward one rotation (or 15 cm) in a straight line, and then start following the line.

The robot follows the line as in the previous activity. However, instead of setting the program to an infinite loop, set it to "repeat until <>." In the <> space, insert the colour sensor and request the yellow colour on the first stop. Right after the loop, have the robot stop moving and turn left.

You can repeat the same operation to get to the red paper and then turn right.

Moving to the yellow box on C8 is more difficult. You will need the "More Motors" code extensions to get there. Set the relative position of motor A (or B) to zero. The robot will then follow the line until the relative position of A is greater than 720 degrees, which is the number of degrees the motor moves to get to the yellow box.

## Discussion

**Questions to ask when returning to the group:**

- → Did you complete the challenge all at once, or in parts?
- → Did you share solutions with other teams?

## A Step Further

**Now that you have successfully completed a complex challenge:**

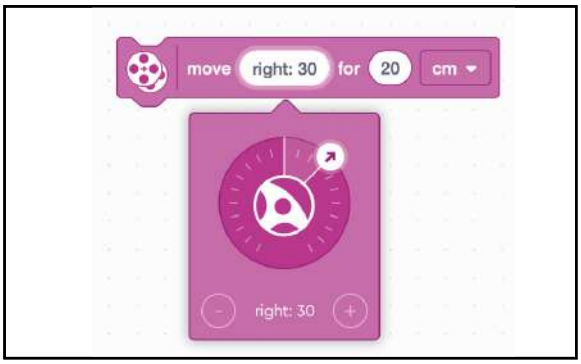
- → Would you be able to return to the starting point following the same path?
- → Would you like to add coloured rectangles to other points on the mat?

# 5. Ultimate Challenge

Place LEGO® bricks (4 x 4 x 2) on the magenta box on C2 and design a robot that can deliver them to the box of the same colour on G8. Do the same with the other squares and colours. How many can you do?

**Psst!**
The ZAP mat logo says to "repeat 180 times."
Do you know why? Simply because there are 180 school days in a year.

# Exemples of **turning radius** with SPIKE Prime




This block starts moving a Driving Base forward with the possibility of steering.

The value 0 will make the Driving Base go straith forward.

The values -99 and 99 will make the Driving Base turn on one wheel.

The values -100 and 100 will make the Driving Base pivot on itself.




Marc-André Caron - Brault & Bouthillier