

Mentor sheet
Week 5
Line-Following, 2 sensors

1. Introduction of the Rescue Challenge
2. Line following with 2 Light sensors (strategy)
3. Line following with 2 light sensors (programming)

Equipment needed:
2 light sensors attached to the front of a robot that will straddle a black line 15 mm wide.

This week, several parts of the lesson plan will map directly into the student worksheet. Make sure that students follow along so that they have a conceptual understanding of line following with 2 sensors.

1. Rescue Challenge (Read through this so that you can give the specifics to your kids. They don't have to know all of the details right now, but you should).

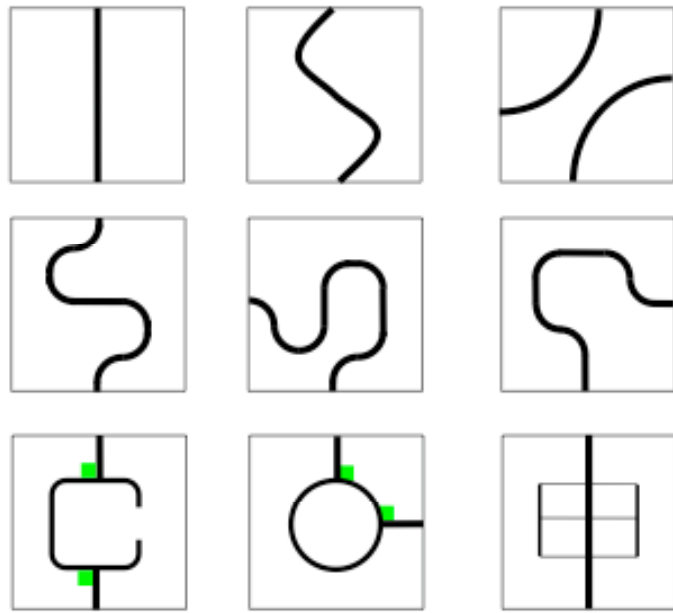
The challenge for December 1 will be a rescue challenge based on the Australian RoboCup Junior Rescue Challenge (<http://www.robocupjunior.org.au/>). Robots will navigate a series of tiles (examples later) following a black line on a white background to a green “chemical spill.” The track will consist of no less than 6 and no more than 12 tiles. In the green area, a victim (aluminum can weighted to 60g) will be present. The rules are summarized below. The official rules from RoboCup Junior Australia are attached at the end of this document.

Scoring

Teams will receive 10 points for each tile that the robot successfully navigates. Teams will receive 10 points for identifying the victim (Beeping, lighting up, or purposefully approaching), and 10 additional points for moving the victim outside of the chemical spill. If a robot stops following a line for any reason, teams may restart or skip the tile. Teams may restart as many times as desired within 120 seconds without penalty and will receive the score of the highest run. Should a team feel they are unable to complete a tile, they may skip to the next tile for a 5-point deduction. Ties will be broken first by number of tiles skipped and second by time. For example, the team with the fewest tiles skipped will be first, and if that doesn't break the tie, the team with the lowest overall time will be first.

Tiles

There will be 3-5 tracks of increasing difficulty utilizing the tiles in the following figure. Track 1 will only use tiles from the first 2 rows, track two will draw from the first 3 rows, and track 3+ will draw from all available tiles. The bridge will have a maximum incline of 15 degrees. The speed bumps will have a maximum height of 5 mm, and the water tower will be some immovable object (weighing significantly more than the victim) that the robot must navigate around. The green squares on the Dead End, Roundabout, and Gridlock tiles are shortcut indicators that may or may not help you as you strategize and program.



Dead End

Roundabout

Bridge

Pool A



Gridlock

Speed Bumps

Water Tower

Tiles

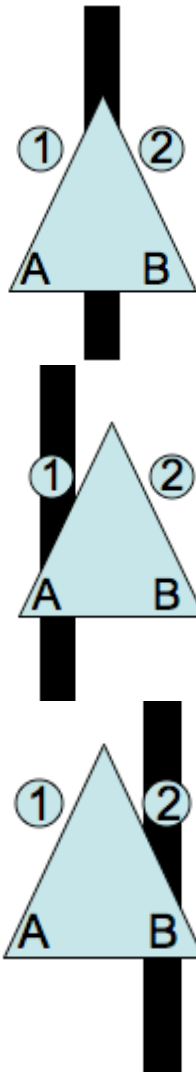
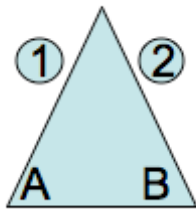
2. 2 sensor line-following

By now, everyone should understand the basics of 1 light sensor line following. Have your group trace the path of a line-following robot with 1 light sensor on their worksheet.



What are some problems with this design? Ans. Choppy, difficulty with tight turns, only follow 1 side of the line, etc...

Students have the following 4 diagrams on their worksheet. The first is a configuration of a robot where 1 and 2 represent light and A and B denote the. Students are then asked to fill in a table with light sensor readings and actions for each engine. Suggested tables are as follows.



Sensor 1	Sensor 2	Motor A	Motor B	Notes
White	White	Forward	Forward	
Black	White	i. Stop ii. Reverse	i. Forward ii. Forward	For i and ii you'll want to wait until Sensor 1 reads white. Strategy ii may be helpful in tight spaces
White	Black	i. Forward ii. Forward	i. Stop ii. Reverse	For i and ii you'll want to wait until Sensor 2 reads white. Strategy ii may be helpful in tight spaces

You on an observant student may have noticed that the case when both sensors read black is absent from the table. We will revisit it once we have developed a working program.

3. Programming the 2 line follower

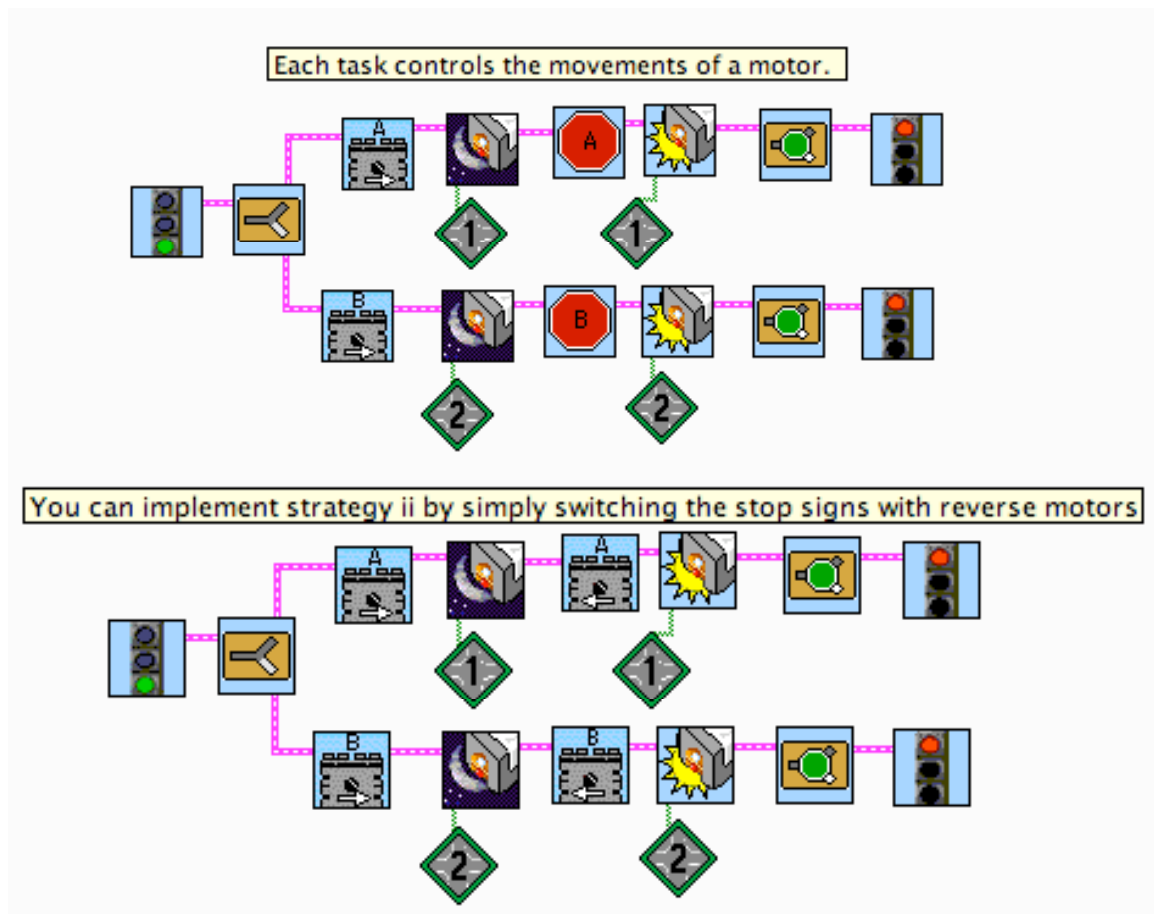
There are infinite ways to program a 2 line follower, but I will provide you with some commented code using events, task splits, and if statements. Your students have the same code without all of the icons and will be asked to fill it in. While it's your students'

choice on what to do, the Task Split and Event programs lend themselves to relatively easy manipulation if you're adding a touch or distance sensor, as may be added to rescue the victim.

Note: Each light sensor based icon needs a numerical constant for the threshold between black and white.

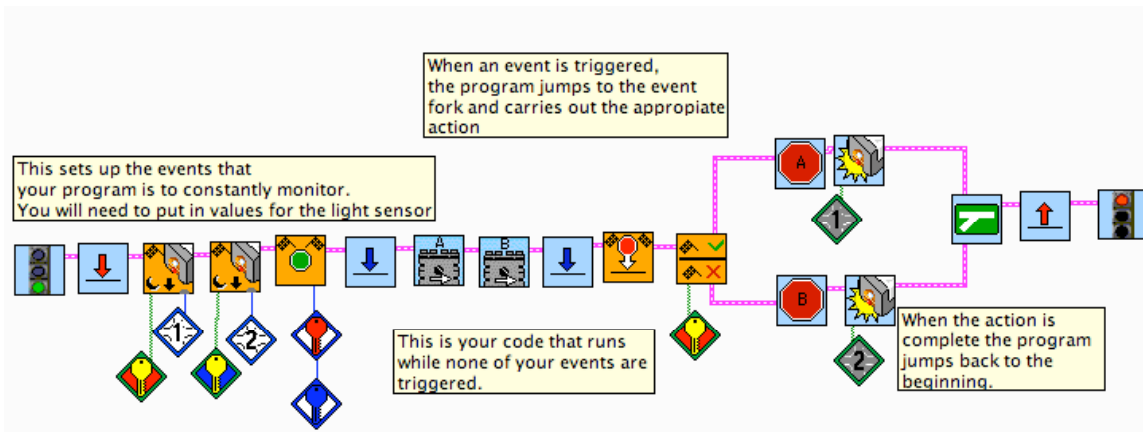
Task Splits

We used task splits last week to show how you could program multiple things to occur at once. In these examples, each task controls the movements of a motor.



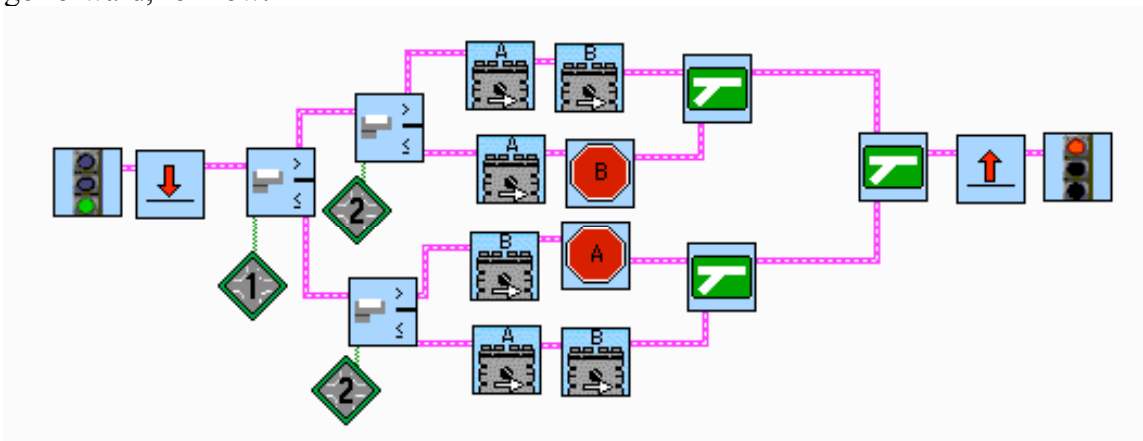
Events

Although we have not formally looked at events, the structure is relatively straightforward. You set up events that you want your program to monitor for and assign them a key of a certain color. Then you start to monitor for your events (Green stop sign with flags), and insert code for what you want to have happen when nothing is triggered. Then you set up an Event Land and specify what you want to have happen when the event is triggered. Once that action is complete, you can start all over again.



Nested If Statements (Forks)

If your students understood forks from a few weeks ago, this program should be easy. You first check to see the value of sensor 1 and then the value of sensor 2, and then perform the desired action. In the lowest fork, where both sensors are black, we have it go forward, for now.



4. Going Farther – When can both sensors read black?

Look at the following tiles and work with your students to come up with strategies to continue following the line when it reaches these tiles. Remember, the green square will have a reading in between black and white, so it may be useful to use light sensor forks to determine if the sensor is reading white or green, or green or black.

