RoboCup Junior Australia

MIGHTY MAISY MAZE FOR EV3 AND SPIKE PRIME

DAVID MUSGRAVE, RESCUE MAZE NATIONAL COORDINATOR

RoboCup Junior AUSTRALIA

Overview of Competition

- “Real world” robot challenge
- Modular, flexible game design
- Accessible to a wide range of robot platforms
- Many possible solutions
- Easy for beginners but with a high ceiling
- Simple scoring
- Engaging and fun

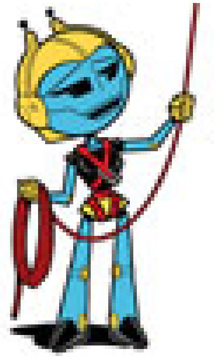
Now even easier for beginners!

CAN BE COMPETITIVE WITH A LEGO EV3 OR SPIKE ROBOT

CAN USE EV3 OR SPIKE CLASSROOM SCRATCH PROGRAMMING

PATHWAY TO MORE ADVANCED SOLUTIONS

LOTS OF DESIGN FLEXIBILITY



Scenario

- There has been a disaster in a factory building.

- Several workers are trapped inside but it is not safe to send in rescue teams. Tokens have been placed to represent trapped (green) and seriously injured (red) victims

- The role of your robot is to enter the building, identify the number and classification of victims, and exit the building.

- Your robot will then report on the victims found.

The Competition

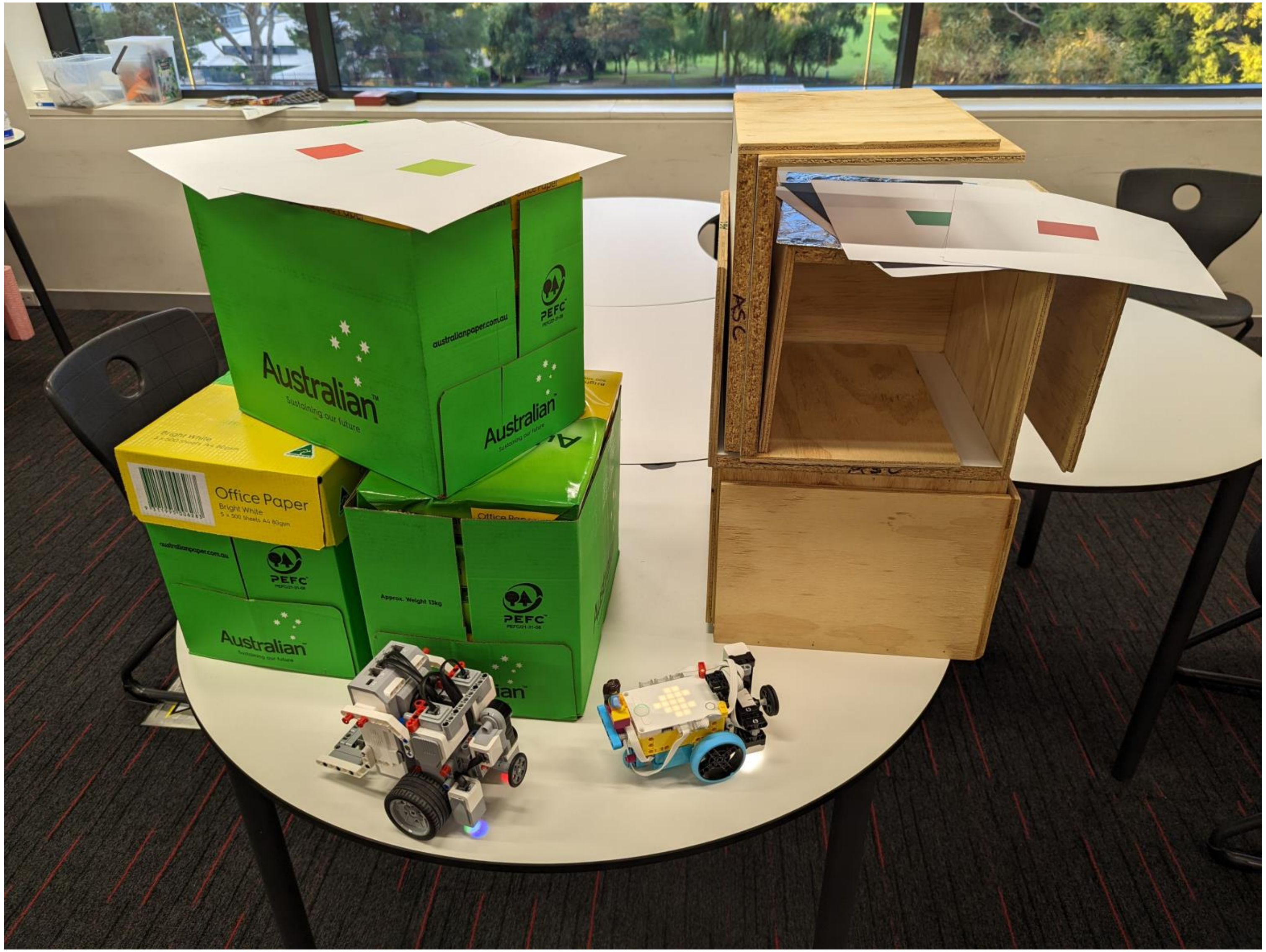
- You have 2 minutes to calibrate your robot, find the victims and exit the maze.
- There is a minimum of 5 victims in the maze. You get points for each one you find. The red victims are worth more points than the green
- Bonus points for exiting and for accurately reporting the number for each type of victim
- The robot must avoid black holes in the ground
- The robot can be return to last found victim if stuck without penalty but program cannot be restarted
- Robot run can be restarted with loss of points at any time
- READ THE RULES !!!!

The Maze

- A4 Maze designed for low barrier to entry into competition.

- Open Maze usable if you already have it.

- New flat pack Open Maze design coming soon.



The Maze – What you Need – A4 Maze

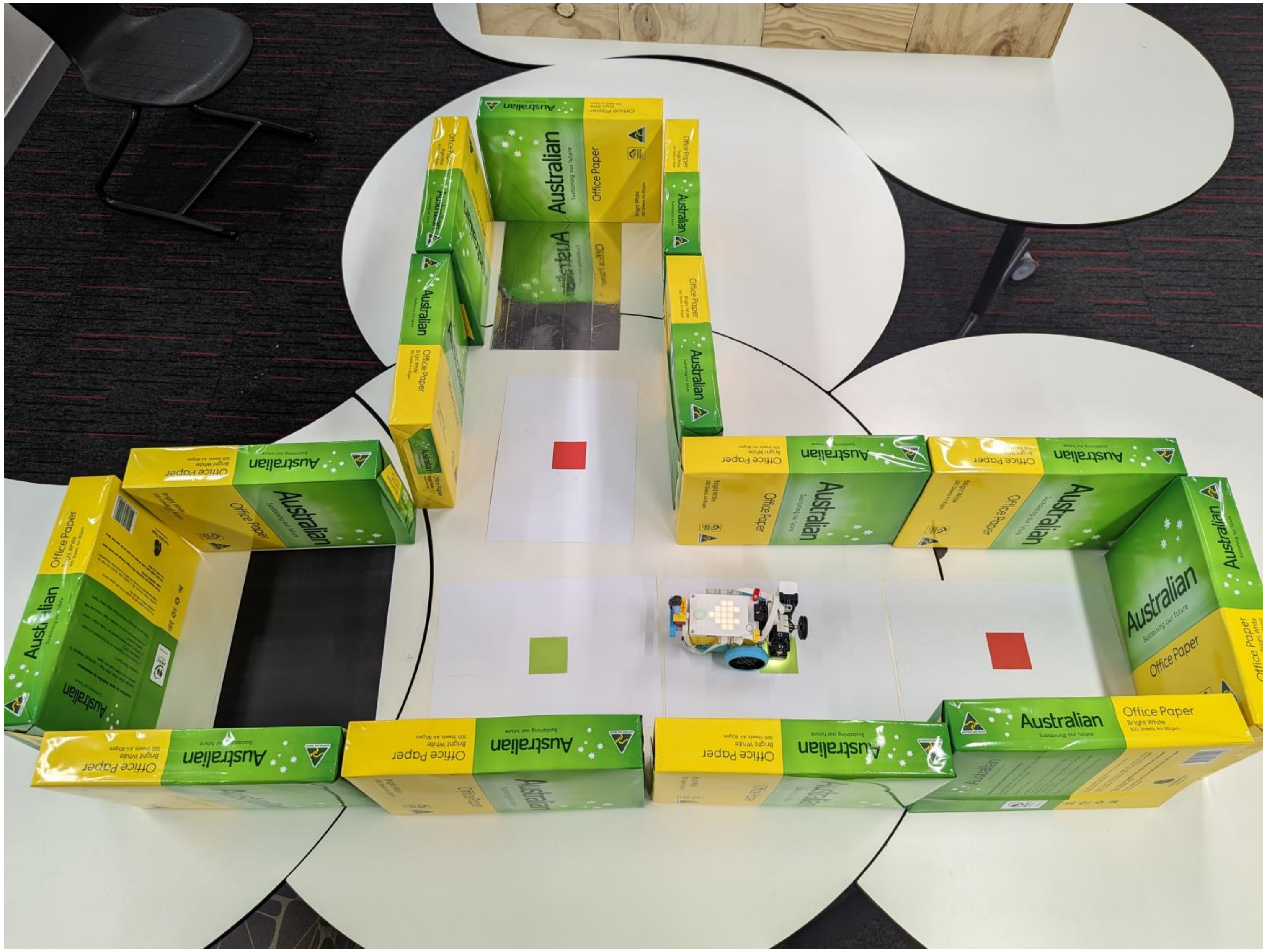
- The simplest way to get started is using reams of A4 paper on their side for walls.

- For the coloured victim marker squares, reflective start/end tile and black hole tile you can use the course instructions PDF available from the Rescue Maze Challenge page of the Robocup Junior Website:

- https://www.robocupjunior.org.au/rescue-maze/

The Maze – What you Need – A4 Maze

- A4 Maze:

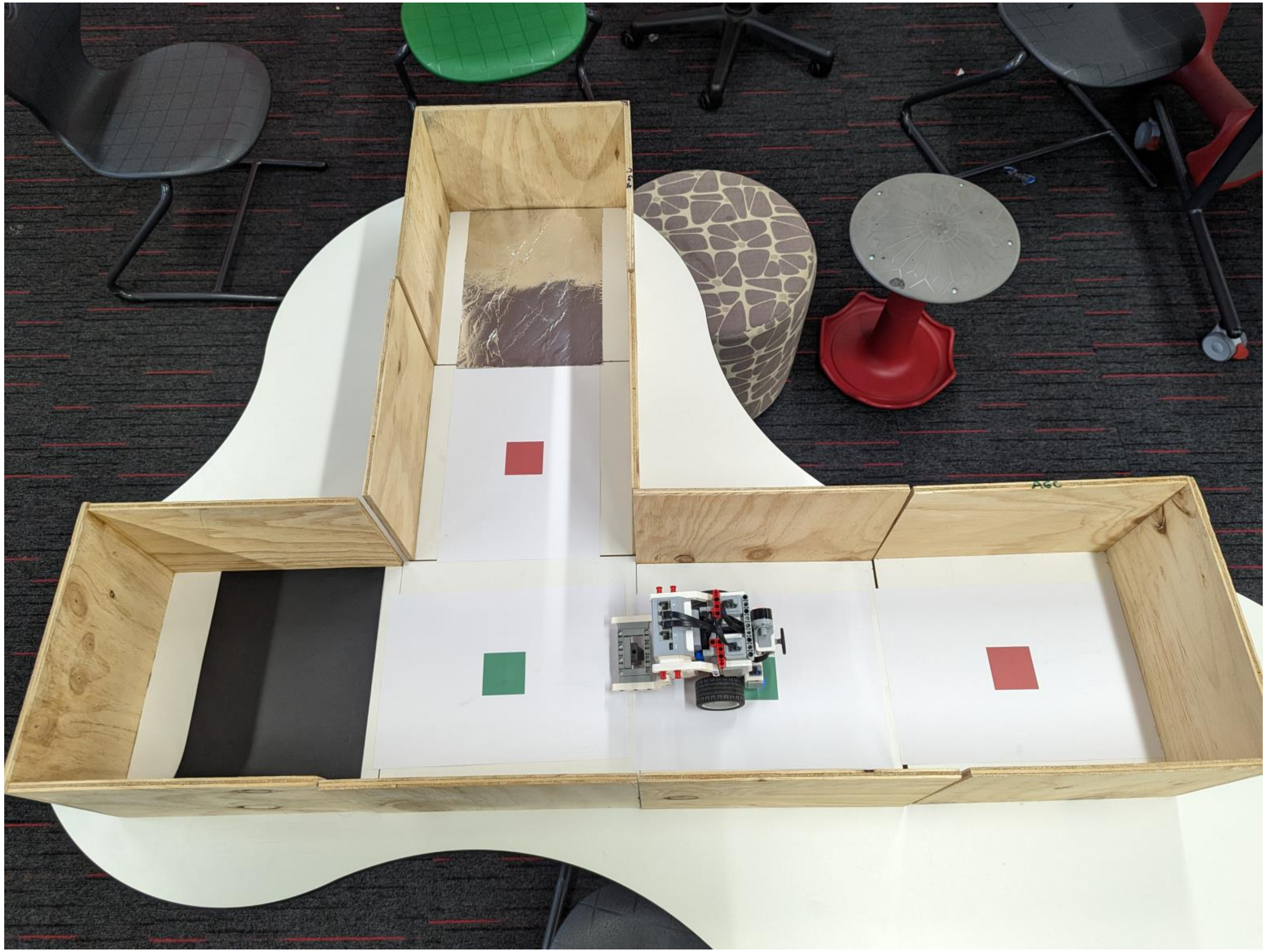


The Maze – What you Need – Open Maze

- If you have Open Maze course tiles, they can also be used.
- The same victim markers can be used as before.
- If you have reflective tiles and black tiles for the Open Maze course, they can be used instead.

The Maze – What you Need – Open Maze

- Open Maze:

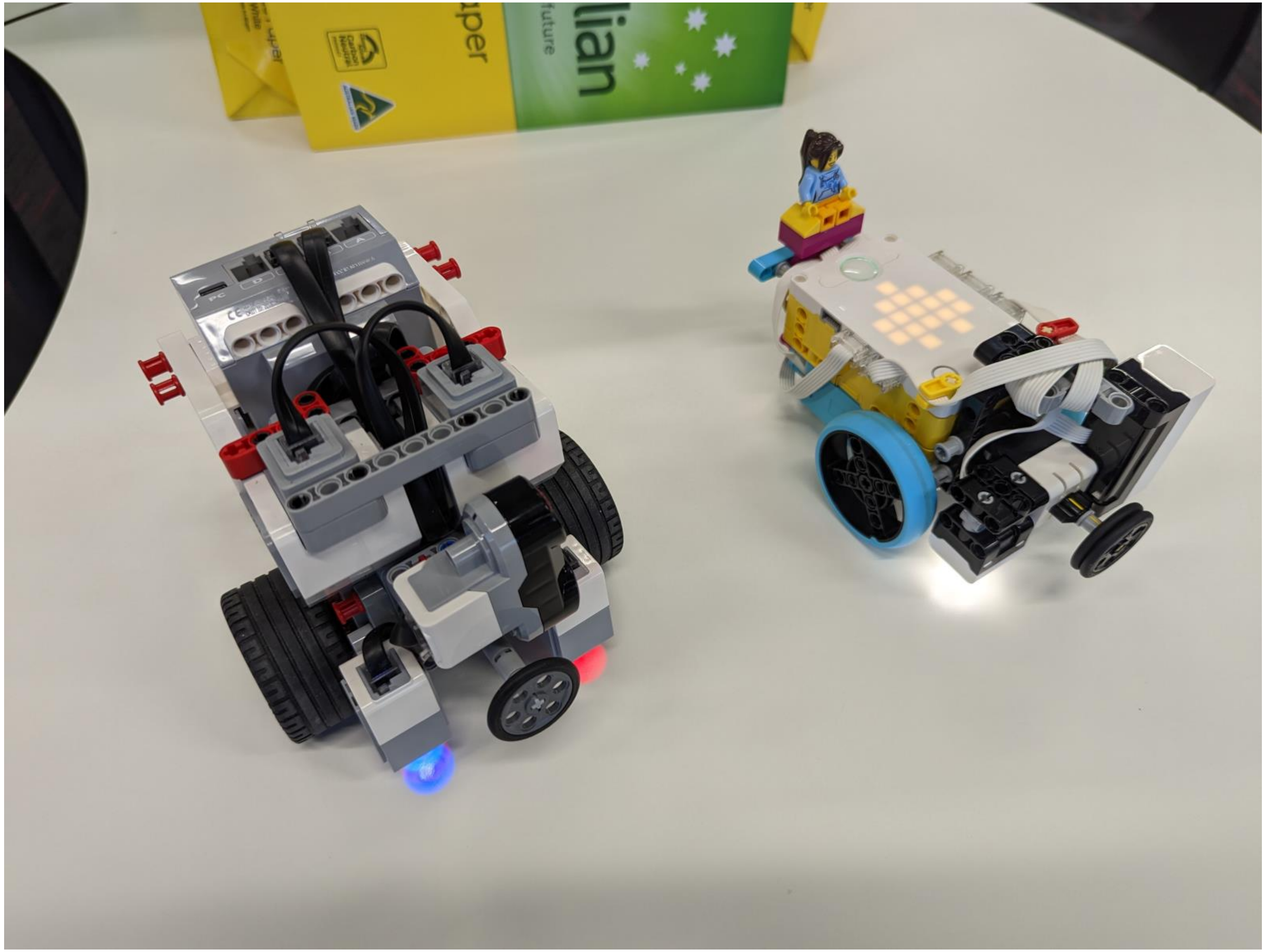


The Robot

Does not have to be Lego

However, Lego EV3 or Spike Prime is a great way to get started in robotics

Can be built with single set

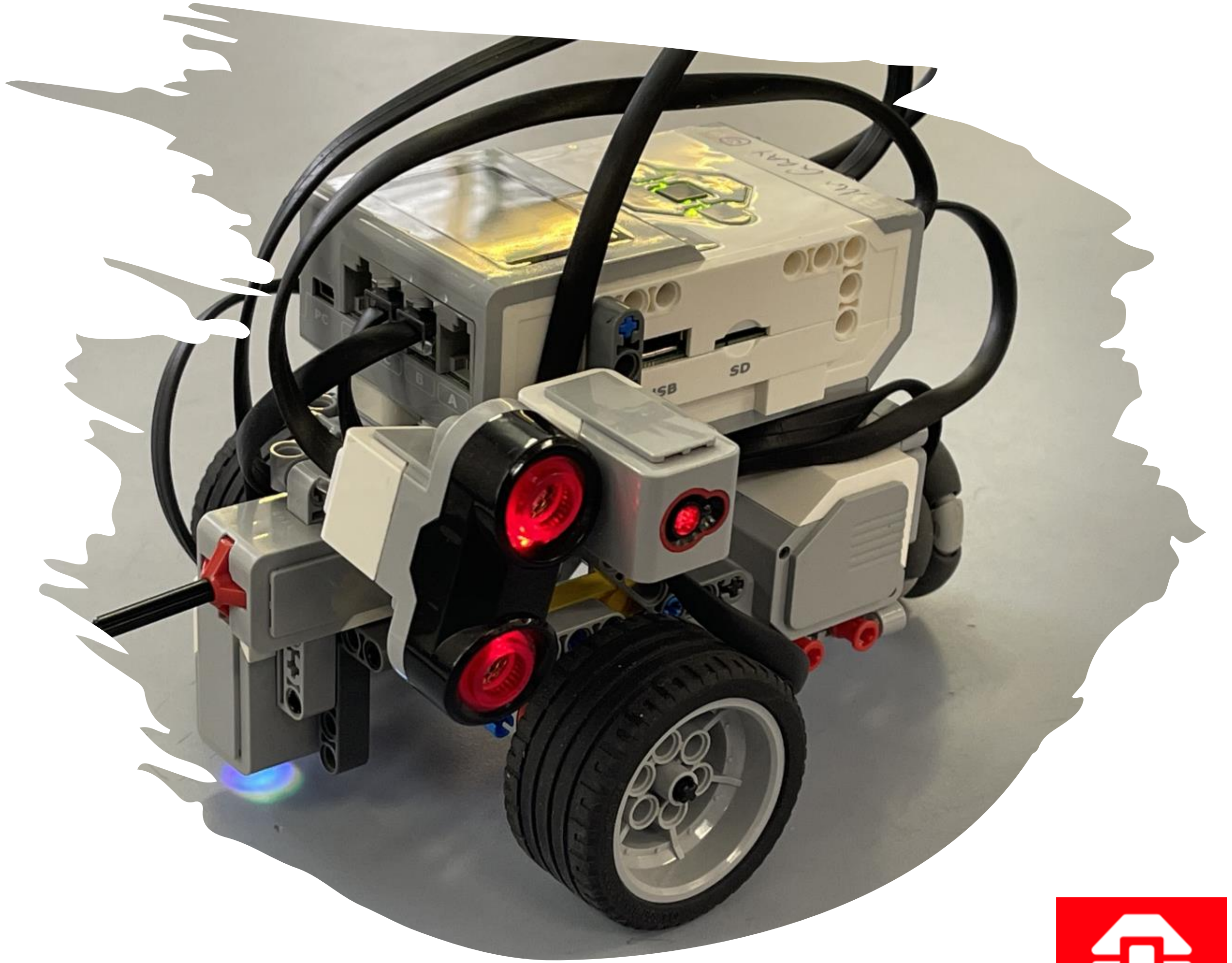


The Robot – what you need

Basic EV3 robot

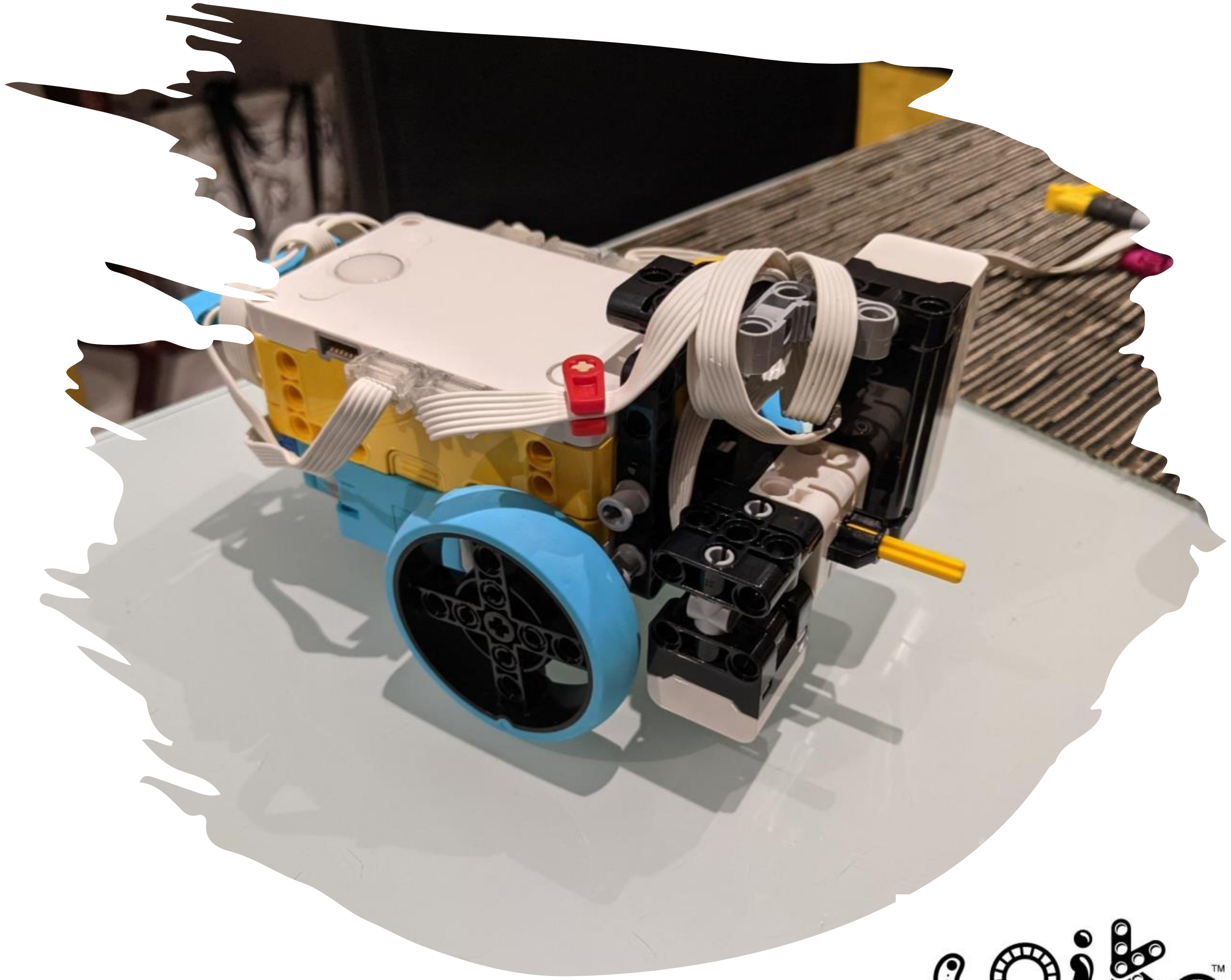
- 1 X Colour sensors (pointing down). Can add a second Colour sensor, if desired*
- 1 X Ultrasonic sensor (facing wall)
- 1 X Touch sensor on front of the robot

* Second colour sensor can help with detecting silver reflective tile



The Robot – what you need

- Basic Spike Prime robot
  - 1 X Colour sensors (pointing down)
  - 1 X Ultrasonic sensor (facing side)
  - 1 X Touch sensor on front of the robot



Maze Algorithm

- Keep it simple and work on one element at a time.
- Make your wall follower as smooth as possible. You want to stay at a constant distance from the wall.
- The rate at which the robot turns is important for reliable performance
- A good development strategy is

✓ Follow a wall

✓ Detect an obstacle in front

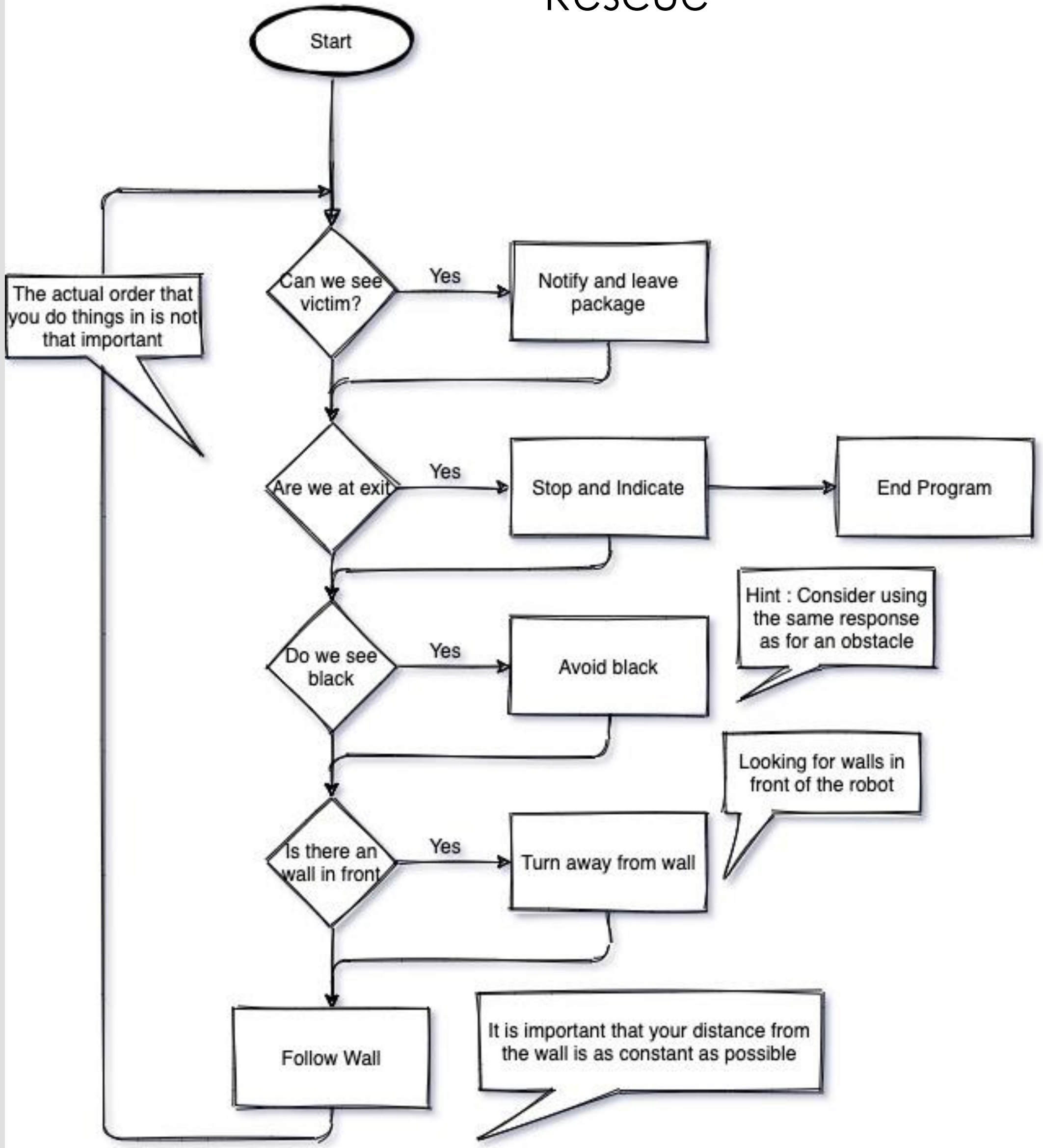
✓ Detect a black hole

✓ Find a victim

✓ Find exit and report

We "nest" a series of if … else statements <u>one at a time</u> to get the robot behavior that we are after.

Mighty Maisy Maze Rescue



Start

Can we see victim?

Yes

Notify and leave package

The actual order that you do things in is not that important

Are we at exit

Yes

Stop and Indicate

End Program

Do we see black

Yes

Avoid black

Hint : Consider using the same response as for an obstacle

Is there an wall in front

Yes

Turn away from wall

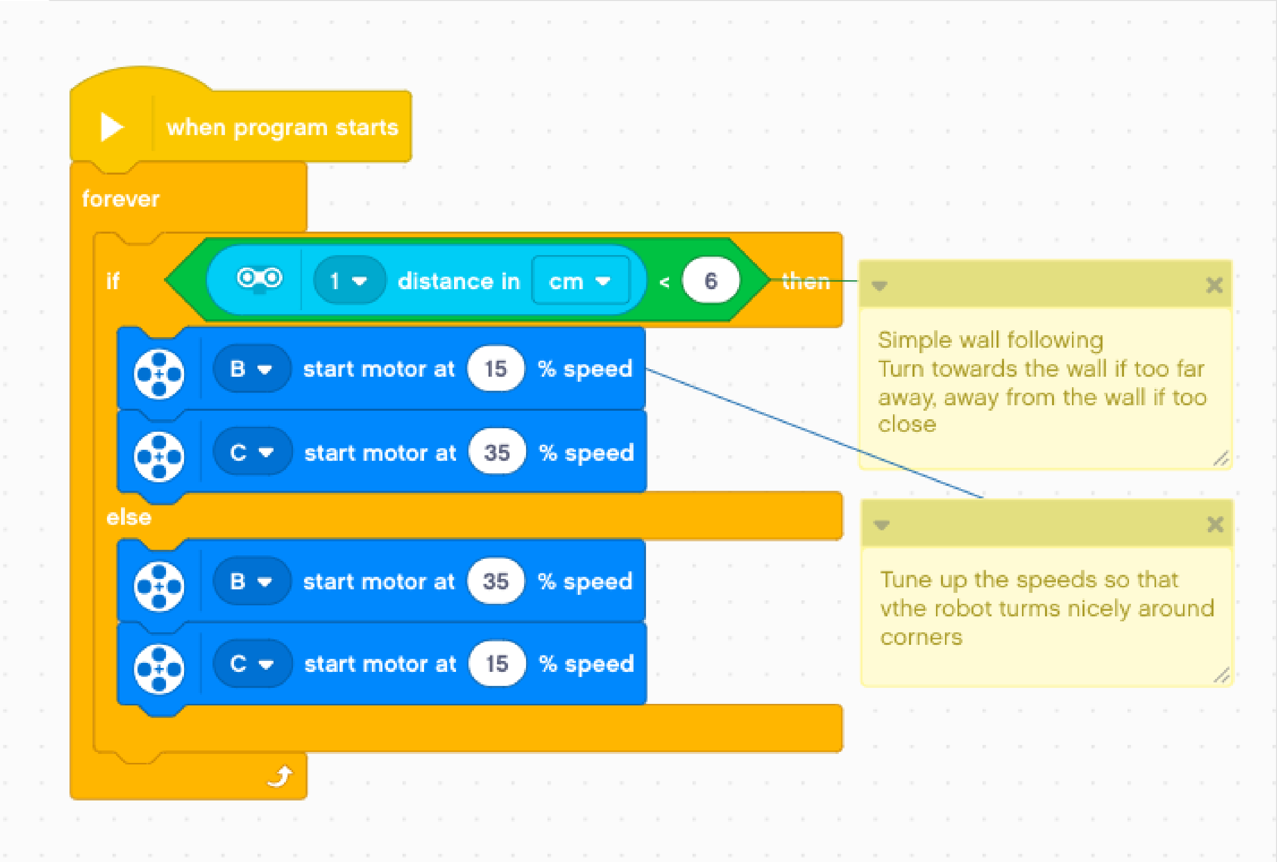
Looking for walls in front of the robot

Follow Wall

It is important that your distance from the wall is as constant as possible

Simple Wall Following

- Wall following is simple. The robot either steers towards the wall if too far away or away from the wall if too close.
- The key issue is carefully calibrating your robot turn rates. The robot should follow a constant arc around a wall divider.
- Individual motor speed control works better than steering for EV3.



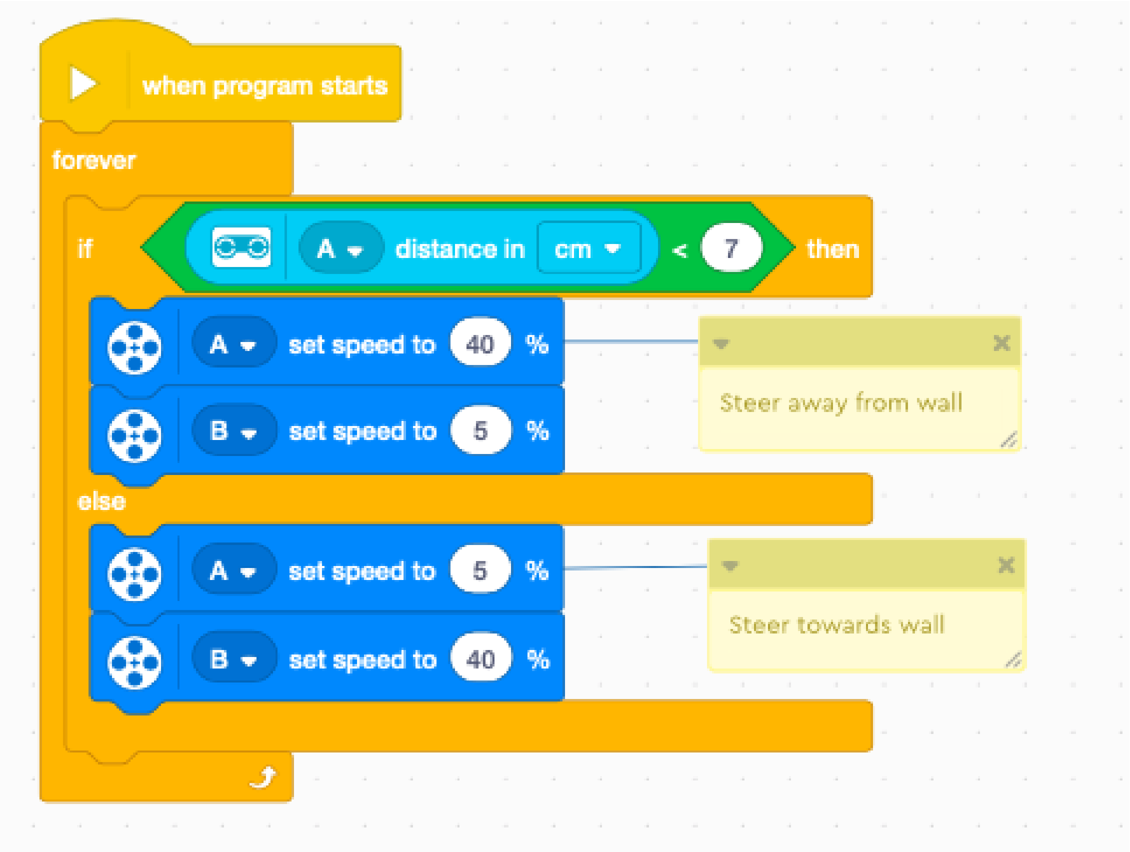
```
when program starts
forever
    if 1 distance in cm < 6 then
        B start motor at 15 % speed
        C start motor at 35 % speed
    else
        B start motor at 35 % speed
        C start motor at 15 % speed
```

Simple wall following
Turn towards the wall if too far away, away from the wall if too close

Tune up the speeds so that vthe robot turms nicely around corners

Simple Wall Following

- Wall following is simple. The robot either steers towards the wall if too far away or away from the wall if too close.
- The key issue is carefully calibrating your robot turn rates. The robot should follow a constant arc around a wall divider.
- Individual motor speed control or steering works for Spike.



```
when program starts
forever
    if  A distance in cm < 7  then
        A set speed to 40 %
        B set speed to 5 %
    else
        A set speed to 5 %
        B set speed to 40 %
```

Steer away from wall

Steer towards wall

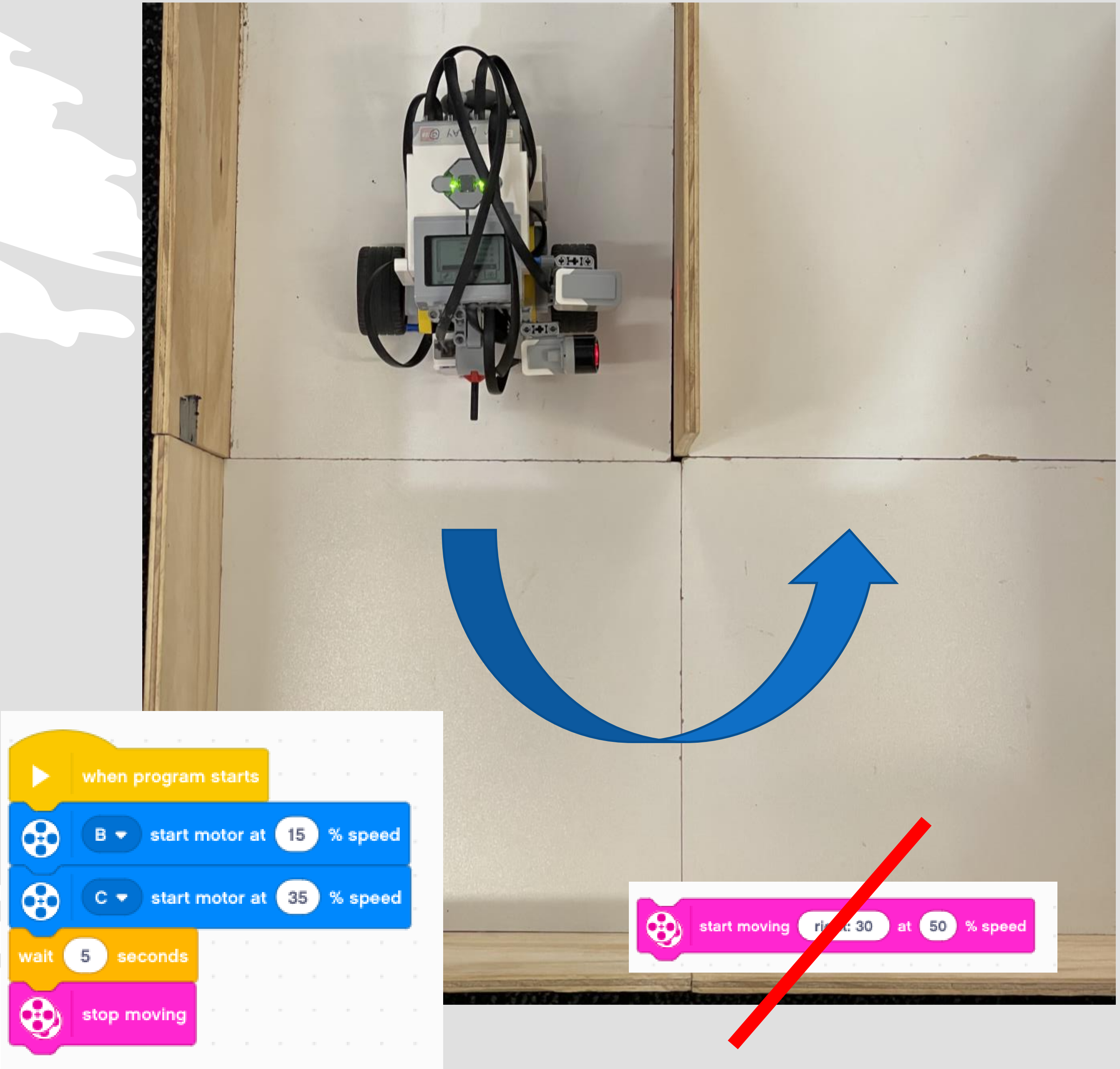
Video: ON/OFF

Video: PROPORTIONAL

Video: FAST PROPORTIONAL

Setting the Turning radius

- When your robot comes to the end of a wall it will try to turn towards where it thinks the wall is.
- Setup your motor speeds so that your robot will do a smooth arc around the wall.
- There is a relationship between speed and turning plus every robot has different characteristics. Play around until you get it right.
- Individual motor speed works much better than using "steering" (for EV3).

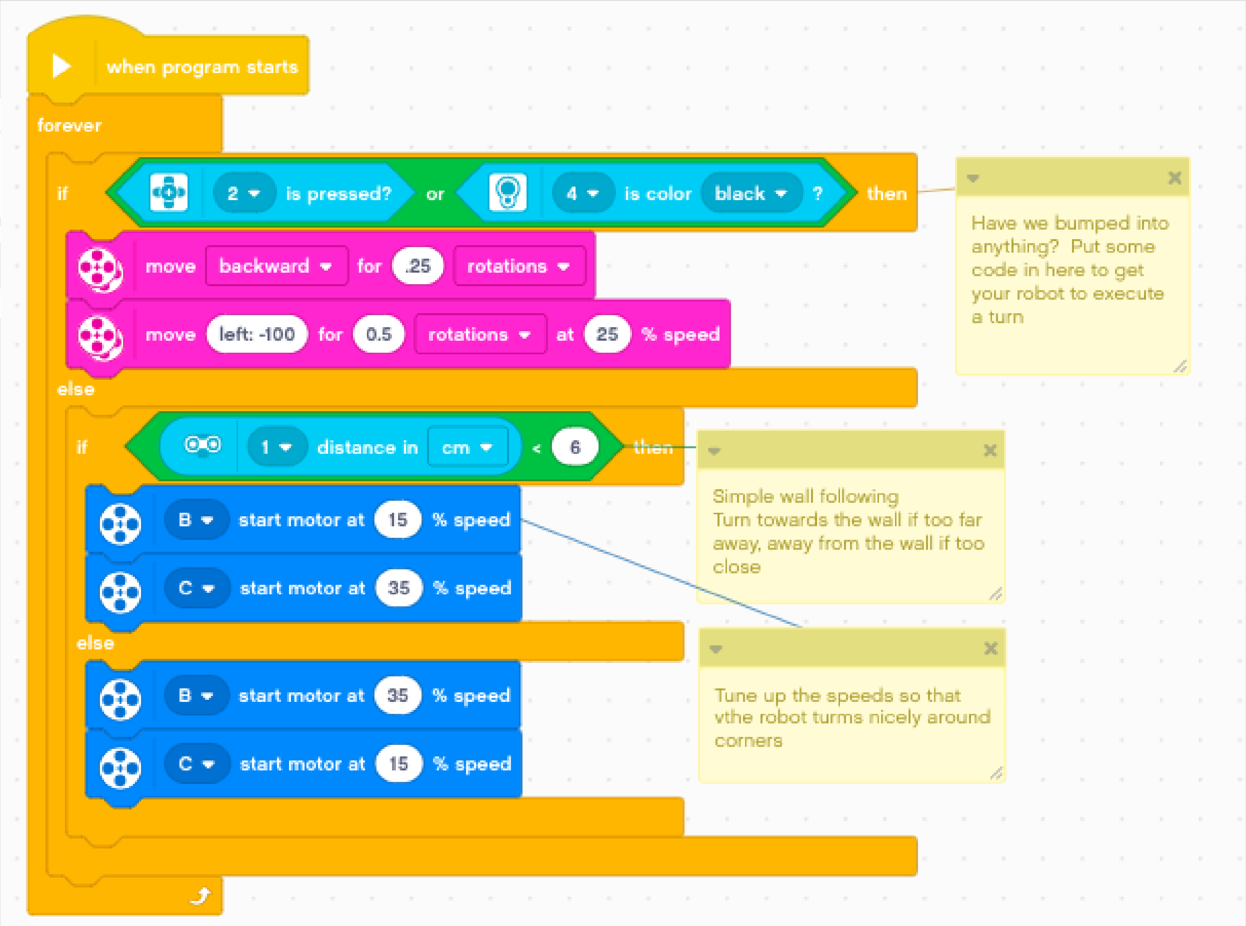


```
when program starts
B ▾ start motor at 15 % speed
C ▾ start motor at 35 % speed
wait 5 seconds
stop moving
```

~~start moving right: 30 at 50 % speed~~

Obstacle Avoidance

- When you see an obstacle in front of the robot you need to react in some way.
- This can be a wall or an obstruction. Even a black hole can be treated as an obstacle as seen in the example on the right.
- If you see a wall in front of you for example you could reverse a little and then turn away from the wall.
- Keep the movements small – it is not unusual to take a few attempts at getting around the obstacle.
- With this simple program you have a robot that can solve a basic maze!



```
when program starts
forever
    if 2 is pressed? or 4 is color black ? then
        move backward for .25 rotations
        move left: -100 for 0.5 rotations at 25 % speed
    else
        if 1 distance in cm < 6 then
            B start motor at 15 % speed
            C start motor at 35 % speed
        else
            B start motor at 35 % speed
            C start motor at 15 % speed
```

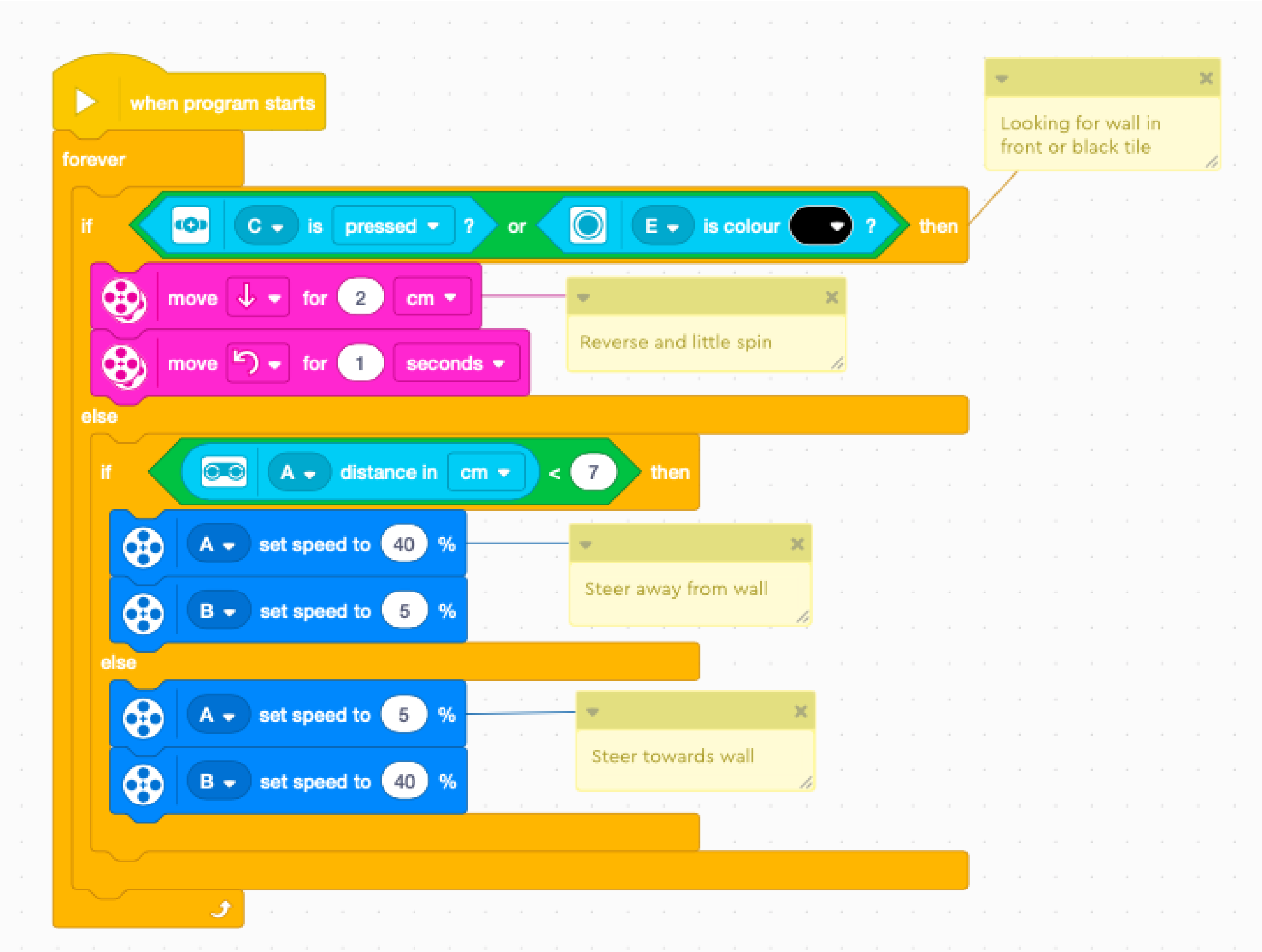
Have we bumped into anything? Put some code in here to get your robot to execute a turn

Simple wall following
Turn towards the wall if too far away, away from the wall if too close

Tune up the speeds so that vthe robot turns nicely around corners

Obstacle Avoidance

- When you see an obstacle in front of the robot you need to react in some way.
- This can be a wall or an obstruction. Even a black hole can be treated as an obstacle as seen in the example on the right.
- If you see a wall in front of you for example you could reverse a little and then turn away from the wall.
- Keep the movements small – it is not unusual to take a few attempts at getting around the obstacle.
- With this simple program you have a robot that can solve a basic maze!



```
when program starts
forever
    if < C is pressed ? or E is colour ● ? > then
        move ↓ for 2 cm
        move ↺ for 1 seconds
    else
        if < A distance in cm < 7 > then
            A set speed to 40 %
            B set speed to 5 %
        else
            A set speed to 5 %
            B set speed to 40 %
```

Looking for wall in front or black tile

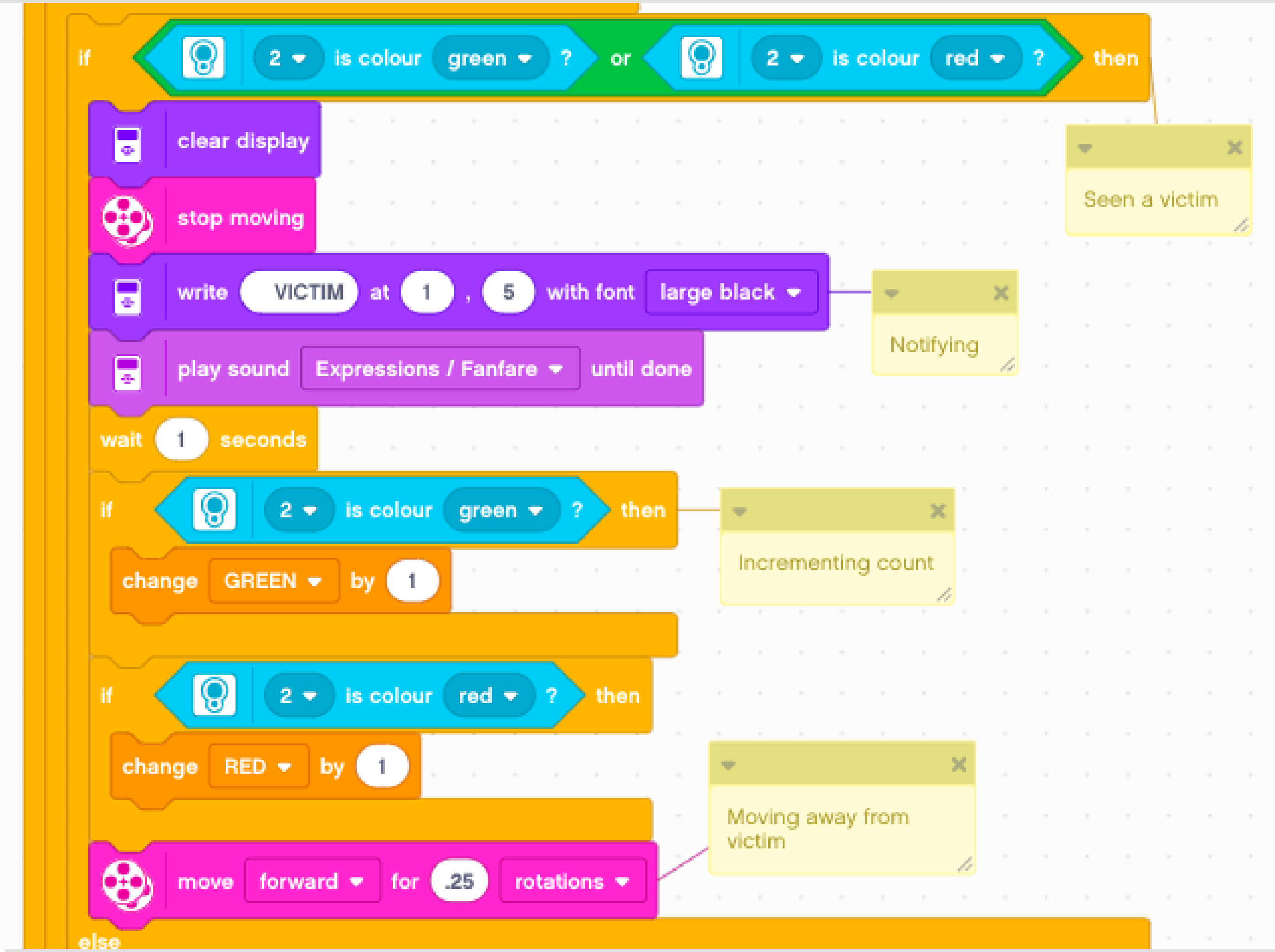
Reverse and little spin

Steer away from wall

Steer towards wall

Finding the Victim

- The victims are represented by 50mm square markers in the middle of each passage through the maze. They are coloured red or green to indicate if they are in critical condition or just need to be located.
- This means we can use the Colour Sensor to identify the victim.
- When you find a victim you need to stop for 1 second and indicate in some clear way.
- It may be a good idea to move off the victim before continuing – why?
- A constant wall following distance is critical



```
if 2 is colour green ? or 2 is colour red ? then
    clear display
    stop moving
    write VICTIM at 1 , 5 with font large black
    play sound Expressions / Fanfare until done
    wait 1 seconds
    if 2 is colour green ? then
        change GREEN by 1
    if 2 is colour red ? then
        change RED by 1
    move forward for .25 rotations
else
```

Seen a victim

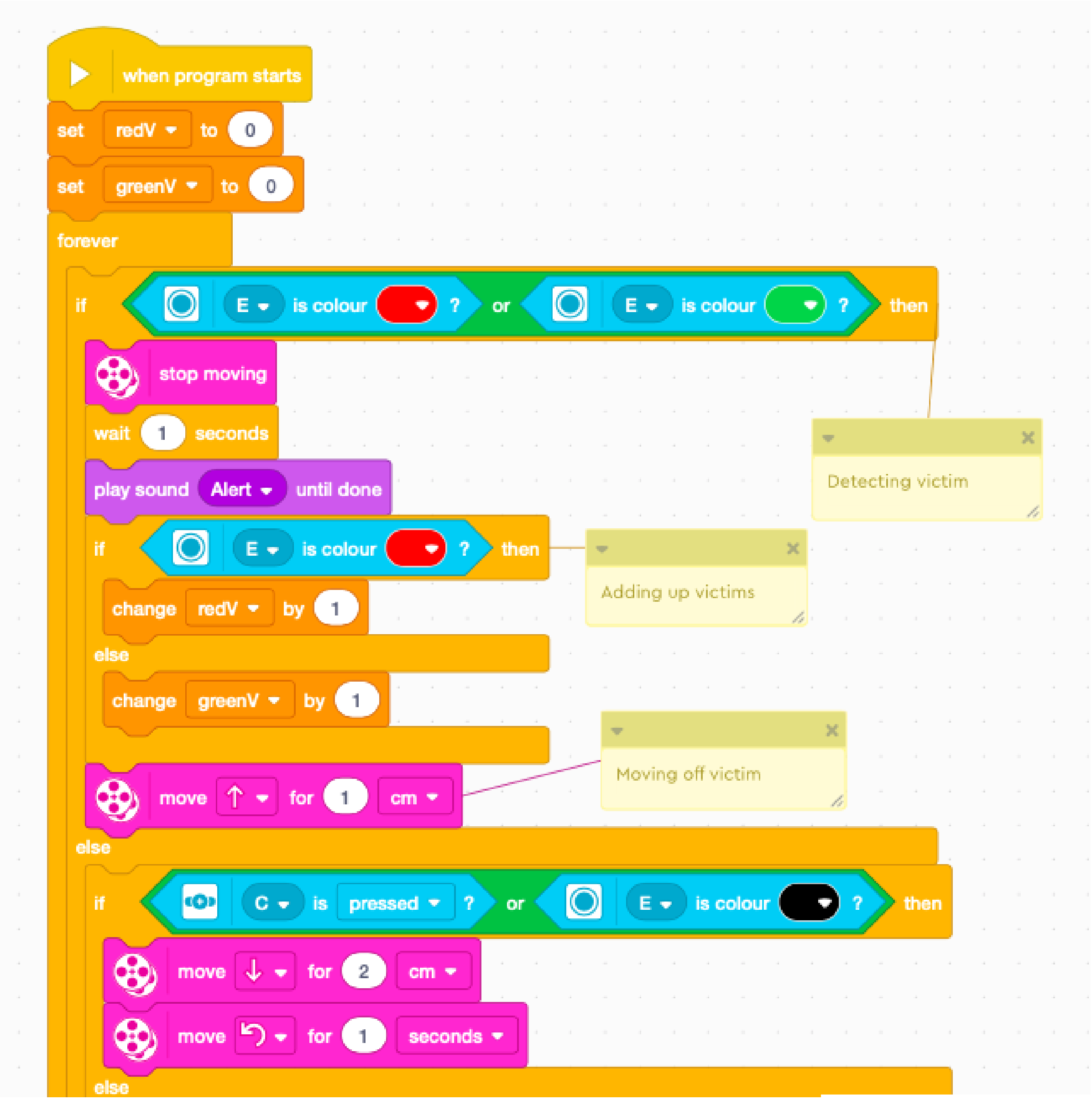
Notifying

Incrementing count

Moving away from victim

Finding the Victim

- The victims are represented by 50mm square markers in the middle of each passage through the maze. They are coloured red or green to indicate if they are in critical condition or just need to be located.
- This means we can use the Colour Sensor to identify the victim.
- When you find a victim you need to stop for 1 second and indicate in some clear way.
- A constant wall following distance is critical



```
when program starts
set redV to 0
set greenV to 0
forever
    if E is colour 🟥 ? or E is colour 🟩 ? then
        stop moving
        wait 1 seconds
        play sound Alert until done
        if E is colour 🟥 ? then
            change redV by 1
        else
            change greenV by 1
        move ↑ for 1 cm
    else
        if C is pressed ? or E is colour ⬛ ? then
            move ↓ for 2 cm
            move ↺ for 1 seconds
        else
```

Detecting victim

Adding up victims

Moving off victim

Finding the Exit tile

The start/exit tile is shiny and highly reflective.

See if you can work out some way to sense it but only actually exit when you return back to it.

Hint ….. put in another nested if … then based on reflected light to notify your success and turn the robot off.

Now it is time to display how many and what type of victims you have found.

Hint: Detecting Silver on a Spike Robot will need to show Block Extensions and enable More Sensors. Then use Raw Light Sensor Block.

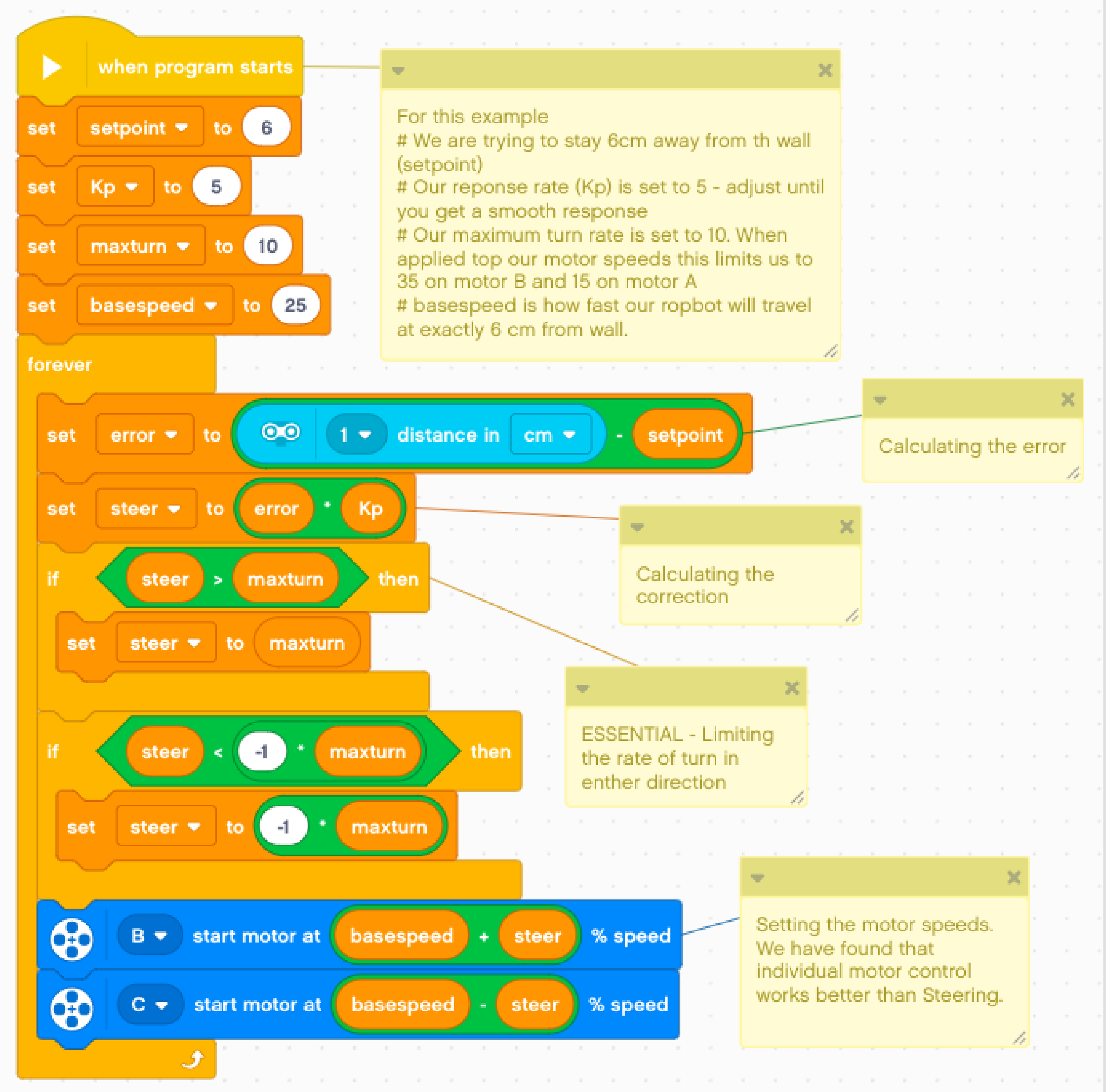
SCORING

ACTION	POINTS	COMMENTS
Found green victim	10 points per victim	Robot must stop for 1 sec.
Found red victim	25 points per victim	Robot must stop for 1 sec.
Exit Bonus	25 points	Robot must stop on exit
Reporting Bonus green victims	25 points	Robot must stop on exit and display the correct number of green victims
Reporting Bonus red victims	50 points	Robot must stop on exit and display the correct number of red victims

The champion teams is decided by the cumulative total over several rounds

A better Wall Follower

- As already noted an accurate wall follower is essential to increasing the reliability of your victim detection.
- The best way to do this is to use a proportional control algorithm.
- Seems tricky at first but pretty easy once you get the hang of it.
- Essentially you speed up the wall side motor and reduce the speed to the other to move away from the wall.
- You do the opposite if you get too far away.
- **Hints : Make sure you put in something to limit the rate at which the robot turns**
- **Play with the Kp and basespeed until you get the response that you want.**



```
when program starts
set setpoint to 6
set Kp to 5
set maxturn to 10
set basespeed to 25
forever
    set error to (1 distance in cm) - setpoint
    set steer to error * Kp
    if steer > maxturn then
        set steer to maxturn
    if steer < -1 * maxturn then
        set steer to -1 * maxturn
    B start motor at basespeed + steer % speed
    C start motor at basespeed - steer % speed
```

For this example
We are trying to stay 6cm away from th wall (setpoint)
Our reponse rate (Kp) is set to 5 - adjust until you get a smooth response
Our maximum turn rate is set to 10. When applied top our motor speeds this limits us to 35 on motor B and 15 on motor A
# basespeed is how fast our ropbot will travel at exactly 6 cm from wall.

Calculating the error

Calculating the correction

ESSENTIAL - Limiting the rate of turn in enther direction

Setting the motor speeds. We have found that individual motor control works better than Steering.

Important Links

Everything you need will be accessible through the RoboCup junior Australia web site and from the workshop materials link (including this presentation).

- Workshop Materials: http://tinyurl.com/RCJWARescue

- RCJA Website: https://www.robocupjunior.org.au/
- Rescue Maze: https://www.robocupjunior.org.au/rescue-maze/

- David's Robotics Portal: http://winthropdc.com/Robotics/

- EV3 Standard Design: https://www.robocupjunior.org.au/wp-content/uploads/2021/08/Maze-EV3-Standard-Design.pdf
- Spike Standard Design Video: https://youtu.be/7wiqk-wr-wo