ROBOCUP JNR RESCUE TEACHING AND LEARNING GUIDE

Jessica Cathro

MINISTRY OF INSPIRATION

Contents

Robocup Jnr Teaching and Learning Guide

This handbook has been designed to support teachers to use the Robocup Jnr Rescue competition to teach their classes about relevant Math, Digital Technology and Technology concepts.

While information from this handbook can be given to students, it has not been designed for students.

Introducing Robocup Jnr Rescue

Robocup Jnr Rescue is a nationwide competition which encourages students to design and build a robot capable of autonomously navigating a maze, turning the correct way at an intersection, avoiding obstacles, locating and moving a "victim". **It works best with teams of 2** and **students can range in age from 7 – 18**. This works best with Lego Mindstorms or Arduino robots and students do need a degree of comfort with programming in either Arduino or a block based program.

More information on the competition can be found at the Robocup Jr NZ website: https://www.robocupjunior.org.nz/

Key Competencies

Thinking: *Students need to determine the design for their robotics; they need to break the problem into relevant parts, they need to create a solution for each part; they need to articulate a clear description of their learning journey. They need to reflect on what they have created and how they identified and solved problems. They need to manage their time and tasks, deciding who should do what and when and in what order.*

Using Language, symbols and texts: *They will explain their code and how they arrived at their solution. They will use digital technology to code their robots, learning a new language as they apply it.*

Managing self: *Students need to manage their expectations and their failures. They need to identify the challenge outcomes and work towards that goal. They need to deal with repeated failures. They need to work effectively with others in their team. They need to investigate why their programme has failed, identify possible causes and fix them. They need to develop a plan, manage their time and effectively contribute towards their team goal, taking overall responsibility for a variety of tasks.*

Relating to Others: *Throughout the project students need to consider their teammates opinions, feelings and emotions. They need to consider how they motivate their teammates, how they criticise and praise teammates, how they can encourage teammates to complete tasks on time and to a high quality. They will reflect on what they can do to change their team dynamic. They need to share their ideas with teammates and with the judging panel.*

Participating and contributing: *This project allows schools to draw in parents and whanau as practice audiences prior to the competition. They are exposed to new career paths and opportunities.*

Curriculum Links

Broadly speaking the Robocup Jnr Rescue competition covers the following curriculum links. As you move through the guide we have suggested ways that you can extend the step being tackled to deliver this to older students. Robocup Jnr allows students to work on a physical challenge so students can begin to understand quite complex concepts at a relatively young age, so please don't be hesitant if you think that some curriculum links are only for high school students.

Cross Curriculum Links

Level	Math	Technology	Digital Technology
1	Give and follow instructions for movement that involve distances, directions, and half or quarter turns Describe their position relative to a person or object	Outline a general plan to support that development of an outcome, identifying appropriate steps and resources	Computational thinking progress outcome 1 In authentic contexts and taking account of end-users, students use their decomposition skills to break down simple non-computerised tasks into precise, unambiguous, step-by-step instructions (algorithmic thinking). They give these instructions, identify any errors in them as they are followed, and correct them (simple debugging).
2	Create and use appropriate units and devices to measure length, area, turn and time	Develop a plan that identifies the key stages and the resources required to complete an outcome	Computational thinking progress outcome 2 In authentic contexts and taking account of end-users, students give, follow, and debug simple algorithms in computerised and non-computerised contexts. They use these algorithms to create simple programs involving outputs and sequencing (putting instructions one after the other) in age-appropriate programming environments.
3	Use linear scales and whole numbers of metric units for length, area, angle and time Use a co-ordinate system or the language of direction and distance to specify	Undertake planning to identify the key stages and resources required to develop an outcome. Revisit planning to include reviews of progress and identify implications for subsequent decision making	Computational thinking progress outcome 3 In authentic contexts and taking account of end-users, students decompose problems into step-by-step instructions to create algorithms for computer programs. They use logical thinking to predict the behaviour of the programs, and they understand that there can be more than one algorithm for the same problem. They develop and debug simple programs that use inputs, outputs,

	locations and describe paths		sequence, and iteration (repeating part of the algorithm with a loop).
4	Use appropriate scales, devices and metric lengths for length, area, angle and time	Undertake planning that includes reviewing the effectiveness of past actions and resourcing, exploring implications for future actions and accessing of resources, and consideration of stakeholder feedback to enable the development of an outcome	Computational thinking progress outcome 4 In authentic contexts and taking account of end-users, students decompose problems to create simple algorithms using the three building blocks of programing: sequence, selection, and iteration. They implement these algorithms by creating programs that use inputs, outputs, sequence, basic selection using comparative operators, and iteration. They debug simple algorithms and programs by identifying when things go wrong with their instructions and correcting them, and they are able to explain why things went wrong and how they fixed them.
5		Use planning tools to support and justify planning decisions that will see the development of an outcome through to completion	Designing digital outcomes progress outcome 4 In authentic contexts, students investigate and consider possible solutions for a given context or issue. With support, they use an iterative process to design, develop, store and test digital outcomes, identifying and evaluating relevant social, ethical and end-user considerations. They use information from testing and apply appropriate tools, techniques, procedures and protocols to improve the quality of the outcomes and to ensure they are fit-for-purpose and meet end-user requirements.
6	Measure at a level of precision appropriate to the task		

Getting started – what you need

Tools

- Wire strippers
- Soldering Iron
- Multimeter
- Tape

Materials

- 2 x colour sensors
- 1x ultrasonic sensor
- At least two motors to drive
- A brain capable of being programmed

This teaching and learning guide is laid out with each step following sequentially. However, the reality of working through Robocup Jnr Rescue process means that some steps should be delivered together and that some steps will be repeated. Below is a recommendation for which steps should be combined.

General Topic	Steps to combine
Learning to manage the project	Steps 1 – 3
Building and Designing the Robot	Step 4
Line following	Step 5
Object Avoidance	Step 6
Turn on Green	Steps 7
Finding Silver	Step 8
Find the can	Step 9
The interview	Step 10

Curriculum Links:
Level 1: Outline a general plan to support that development of an outcome, identifying appropriate steps and resources

Level 2: Develop a plan that identifies the key stages and the resources required to complete an outcome

Level 3: Undertake planning to identify the key stages and resources required to develop an outcome. Revisit planning to include reviews of progress and identify implications for subsequent decision making

Level 4: Undertake planning that includes reviewing the effectiveness of past actions and resourcing, exploring implications for future actions and accessing of resources, and consideration of stakeholder feedback to enable the development of an outcome

Level 5: Use planning tools to support and justify planning decisions that will see the development of an outcome through to completion

Key Competencies:

- Thinking
- Managing Self

Vocabulary List:

- Scrum master
- Sprint
- Impediment
- Scrum Board
- Review

Items and equipment you need:

- Device
- Internet access
- Images

We can now:

- Create a Scrum board
- Run a Sprint meeting

Step 1: Project Management SCRUM

Pre-activities:

- **Watch the video here about project design yourself**
https://www.youtube.com/watch?v=H0_yKBitO8M

- **Show your students the videos on SCRUM**
https://www.youtube.com/watch?v=TRcReyRYIMg
https://www.youtube.com/watch?v=iJ_sl6J8PRg

a) Discuss the different roles, SCRUM Master, development team
b) How is this different from how you normally tackle a project

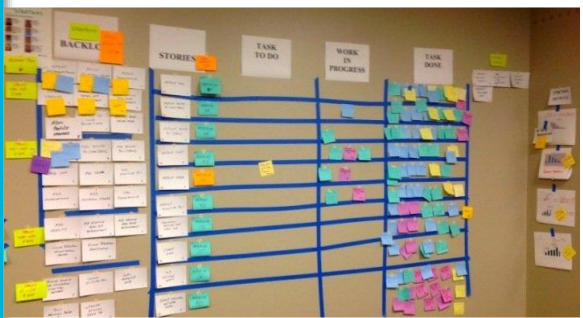
- **Do the marshmallow challenge with your class**
https://www.sciencelearn.org.nz/resources/1696-the-marshmallow-challenge

a) Discuss why some teams were successful
b) Discuss how this would be better with SCRUM
c) Create vocab posters with key vocabulary

- **Discuss how you normally approach group work and what is good/bad about that**

The first three steps should be read and reviewed and delivered together.

SCRUM is a way of managing a project that breaks the project down into small chunks to be completed in a sprint – usually a two week period. A SCRUM Master (the teacher) keeps the teams on track with their timing, supports the regular meetings and helps teams to develop the SCRUM culture.



Scrum starts with a SCRUM board. The SCRUM board has five main columns: the backlog, to do, doing, done and impediments (or stuck).

The **backlog** is a wish list of all the big things that need to get done. For example: build the robot, line following, turn on green, find the can, obstacle avoidance

The **To Do** column lists the more specific tasks that make up the big task. For example: the big task of "build the robot" becomes:

- Identify what materials are required
- Decide on a design
- Build the robot base
- Attach the colour sensors
- Attach the ultrasonic sensor

Each of these tasks needs to go on a separate post it note (or equivalent).

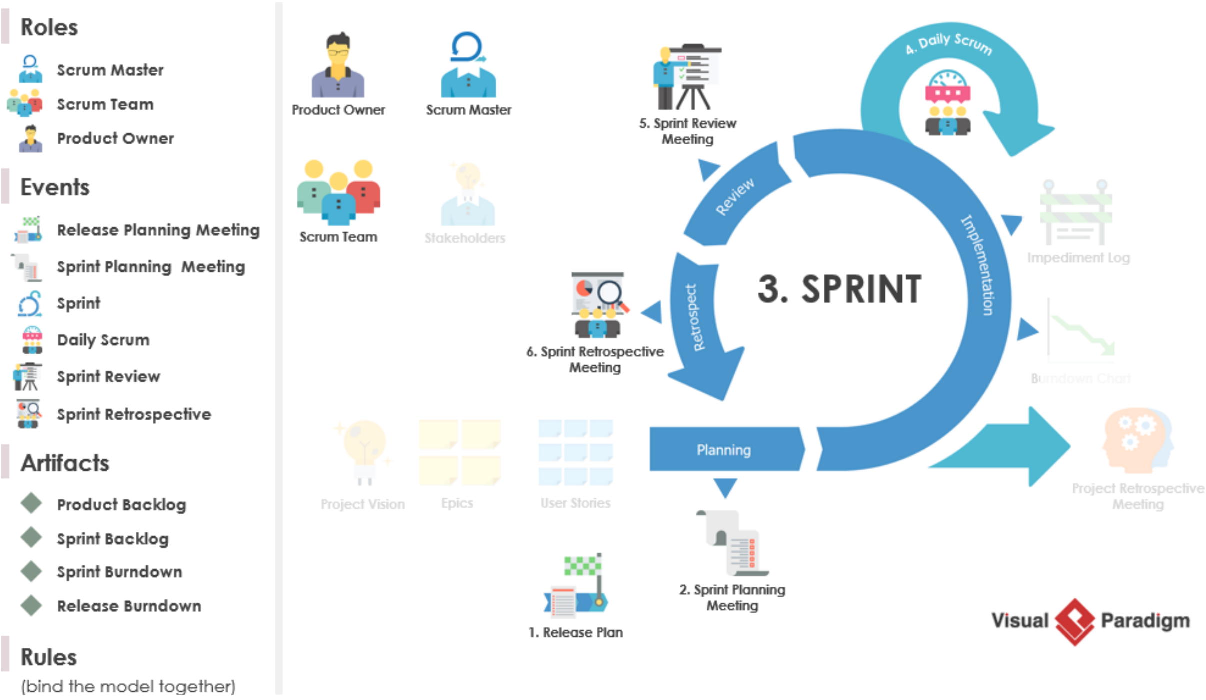
The **doing** column should contain the tasks that each person will do during the sprint. Students will need help to manage their workload effectively. However, as they work through the process they will get an idea for how much they can do during a sprint.

A **sprint** is usually two weeks long but you can determine the length of a sprint based on your time availability and the age of the students. In class I have used a sprint as 4 x 1 hour sessions or two weeks of 1 hour sessions.

The **Done** column is where completed tasks go. You will need to help students to determine what done means. It may pay to make a note of the one or two conditions that make an item done, especially if there is potential for argument. For example; "attach the ultrasonic sensor" means that it is at the front of the robot, not obstructing the colour sensors, being front facing, not having anything in front of it, at a suitable height to see the water bottle and the can.

The **impediments** column is for when students get stuck. They can take their task and put it in the impediment column. I would recommend that they use this for when they need help from you, and use their Scrum meetings to ask for help from their team mates. If you continue to recommend to students to put their task in the impediment section, you can quickly see who needs help and with what rather than being inundated with random "help please".

There are three key SCRUM meetings. The **Sprint planning** (1st meeting), the **Daily Scrum** and the **Scrum Review/Retrospective**.



Sprint Planning (15mins MAX)
Sprint planning is the first meeting in the Sprint:

- Quickly read through all the to do list items
- Each person select two items that they will work on – add their names to the items and move into the doing section
- Ask: does anyone need anything (resources, like devices, internet access, library time, an expert, paper etc.). Captain to take note and communicate these needs to the teacher
- Ask: is anyone concerned about anything or have something that may affect our timing this sprint (like trips out of school, feeling sick etc.). Captain to take note and suggest who might be able to help out
- End meeting

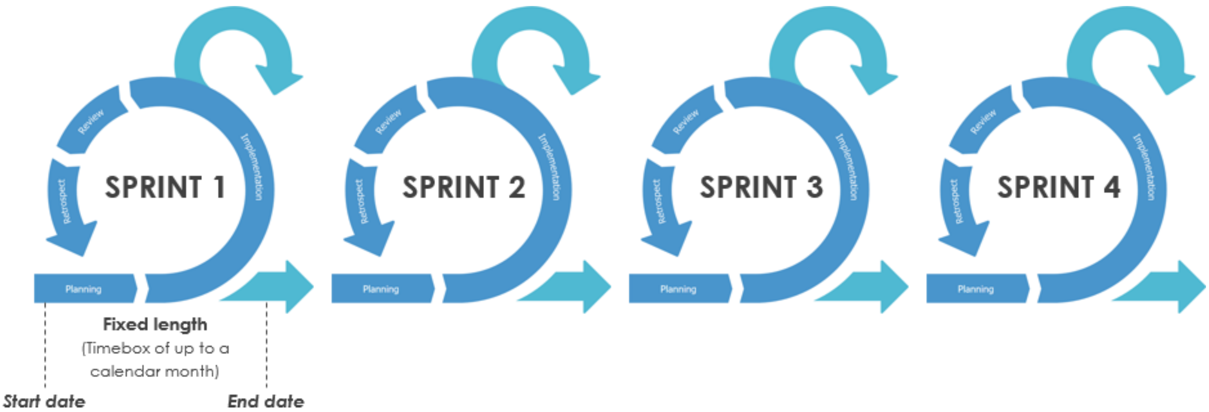
Daily SCRUM meeting (10mins MAX)
The Daily Scrum should happen every time the team meets. This could be daily, or just the hours that you work on the project. In my year 9 class this happened three times a week. The person who runs the meeting will rotate each day.

- Ask each person:
  - What task are you working on (should be a 10 second reply)
  - Do you need anything?
  - Is there anything that will stop you finishing things?
  - Do you have anything to celebrate (finished a tricky task, found out something cool, someone else helped you out)
- End meeting – make sure everyone has moved tasks into Doing and Done and have their name on them

Sprint Review and Retrospective (10mins MAX)
The Sprint Review and Retrospective are about understanding how to improve for next time. While Scrum has these as two separate meetings I have combined them here for efficiency.

- Quickly read through the Done items
- Is there anything outstanding? - move these tasks into the next Sprint
- Ask:
  - What helped us get things done on time? (Headphones in so I can't get distracted, working in 10 min blocks etc.)
  - What stopped us getting things done on time? (talking too much, too much arguing, someone was sick)
- Ask: (if they are running behind)
  - What can we do to catch up? (Someone who finishes early help out the others, do extra time if away etc.)
- End the meeting

Students will need to be guided through this process as it is very different from most group work they have done. For the first few Sprints ask the questions yourself and have the teams discuss. For the next few sprints rotate through the groups each day as they do their meetings, starting with the teams who need more guidance.



Using SCRUM

1) Identify how you/your team would like to manage your SCRUM board – will this be electronic (such as Trello, or Monday.com) or physical (a whiteboard or poster board)
2) Where will your SCRUM board be located so that everyone can access it (including Mentors, teachers, every member of the team)

Curriculum Links:
Level 1: Outline a general plan to support that development of an outcome, identifying appropriate steps and resources

Level 2: Develop a plan that identifies the key stages and the resources required to complete an outcome

Level 3: Undertake planning to identify the key stages and resources required to develop an outcome. Revisit planning to include reviews of progress and identify implications for subsequent decision making

Level 4: Undertake planning that includes reviewing the effectiveness of past actions and resourcing, exploring implications for future actions and accessing of resources, and consideration of stakeholder feedback to enable the development of an outcome

Level 5: Use planning tools to support and justify planning decisions that will see the development of an outcome through to completion

Key Competencies:

- Thinking
- Managing Self

Vocabulary List:

- Engineering Journal
- Evidence
- Timeline

Items and equipment you need:

- Device
- Internet access
- Calendar
- Journal

We can now:

- Create an engineering journal
- Plan an overall project timeline

Step 2: The Engineering journal

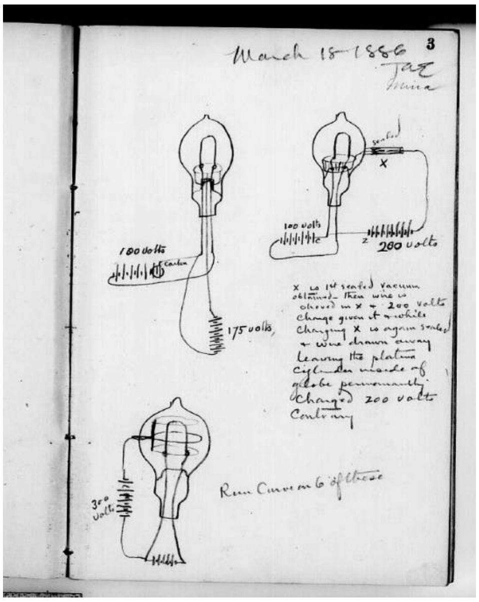
Pre-activities:

- **Review an engineering journal**
- **Discuss how you would keep track of group work**
- **Talk to an engineer about how they keep track of their work**

The first three steps should be read and reviewed and delivered together.

An engineering journal is a one stop shop for students work. It shows their thinking, failures, testing and progress. An engineering journal can be a combined work or individual. The preference is on a combined journal with each person contributing.

The engineering journal can be electronic (scan in any drawing, sketches, hand written notes and use Microsoft Notebook, a Word Document, Padlet) or physical (print out photos, screenshots and use a bound paper book).



The engineering journal should be linear. This means that there aren't sections for Scrum meetings, planning, and research. Instead students record what is happening on the day in the next available page. They should never rip a page out, simply draw a line through anything that is an error and carry on.

What should go in the Engineering Journal?

1) Make sure that a screenshot or photo is added to the engineering journal each day (and is DATED) after each Sprint meeting. Make sure that the answers to the questions are recorded at each meeting and noted in the Engineering journal.
2) Evidence of each task should be captured in the Engineering journal – could be photos, screenshots, links, video (if online), notes, sketches, printouts
3) The overall calendar – see Step 3 for more information

Curriculum Links:
Level 1: Outline a general plan to support that development of an outcome, identifying appropriate steps and resources

Level 2: Develop a plan that identifies the key stages and the resources required to complete an outcome

Level 3: Undertake planning to identify the key stages and resources required to develop an outcome. Revisit planning to include reviews of progress and identify implications for subsequent decision making

Level 4: Undertake planning that includes reviewing the effectiveness of past actions and resourcing, exploring implications for future actions and accessing of resources, and consideration of stakeholder feedback to enable the development of an outcome

Level 5: Use planning tools to support and justify planning decisions that will see the development of an outcome through to completion

Key Competencies:

- Thinking
- Managing Self

Vocabulary List:
- Design Process
- Engineer
- Brainstorm

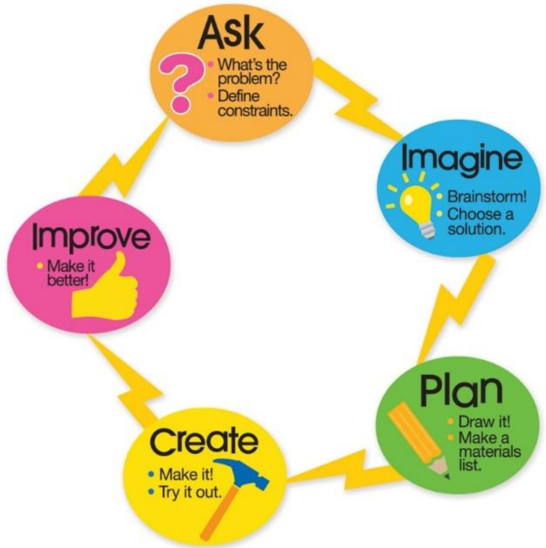
Items and equipment you need:
- Device
- Internet access
- Paper
- Pen/Pencil

We can now:
- Apply the Design Process
- Explain the design process

Step 3: The design process

Pre-activities:
- **Find different examples of the design process**
- **Visit an engineer and discuss how they use the design process at work**
- **Use the design process to create a way to allow an egg to be dropped from a height without breaking**
- **Carry out the marshmallow challenge**

The first three steps should be read and reviewed and delivered together.



The design process is a way of thinking that people like engineers and programmers use to solve a problem. Everything in our lives can be defined as a problem, whether it's making life better or fixing an issue. The design process gives guidelines so that we can move from a problem to a solution. The design process is a circle because we can always make things better.

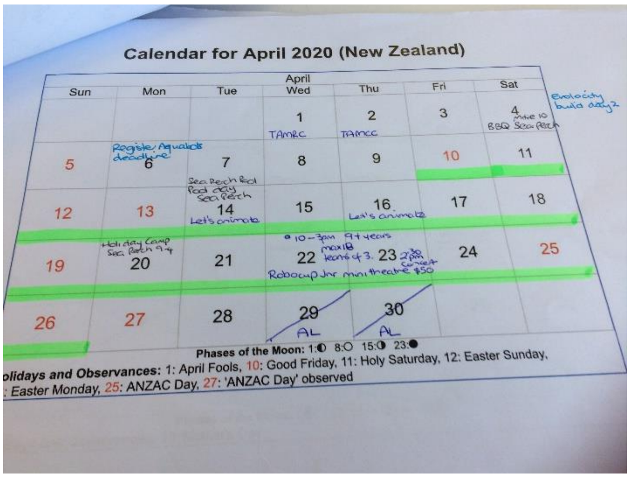
Designing the robot

Ask is the first step and in this, we begin to understand what we are aiming for.
1) Hand out the rules to each team
2) Have them summarise them and stick them into their engineering journal
3) Brainstorm all the tasks that you think you will need in preparing your entry

Here is a list to get them started:
- Design a robot

- Build the robot
- Code a line follower
- Code turning on green
- Code avoiding obstacles
- Code Find silver
- Code search pattern
- Build an attachment
- Design an attachment
- Alter code as needed
- Test

4) Take a large calendar and lay it out on the floor so that students can see the whole time from now till the competition, including the workshops and build days. Have them layout the items from their brainstorm on the calendar. Which items need to be done first, which can be done together, which cannot be done till something else is. Link these items together – this calendar will be the general overview and guide for how this will be done. Put a copy of this into the engineering journal and all big tasks into their SCRUM board backlog



Curriculum Links:
Level 2: In authentic contexts and taking account of end-users, students give, follow, and debug simple algorithms in computerised and non-computerised contexts. They use these algorithms to create simple programs involving outputs and sequencing (putting instructions one after the other) in age-appropriate programming environments.

Key Competencies:
- Using language, symbols and text

Vocabulary List:
- Programme
- Code
- Algorithm
- Line following
- Obstacle avoidance

Items and equipment you need:
- Device
- Internet access
- Robots
- Masking Tape

We can now:
- State the limitations of the robotic kit
- Select a robot
- State the parts of the robot
- Design a robot capable of rescue
- Build a robot capable of rescue

Step 4: Designing and Building a Robot

Pre-activities:
- View a variety of different robots either in person or online
- Explore the robots and identify its different parts

Your robot design should include:
- Two colour sensors
- Ultrasonic sensor
- Two motors (one to control left and right)
- Brain

Other key things to consider are:
1. Access for programming the brain
2. Access for charging/changing batteries
3. Access to turning on/off and pressing play
4. What will you attach to move/lift the victim
5. Ultrasonic sensor should be at the front and able to "see" the 1L bottle
6. Colour sensors need to be closer together than the width of the line
7. Colour sensors need to be close enough to the ground to ensure that reflected light is accurate
8. Can you go over the judder bars?
9. Can you go over the ramp?

You can use a design from online such as:
- Lego Mindstorms (Robot Riley, Sirius etc)
- Arduinos

Extension for older students:
- Try to use angles and wheel rotations to move between objects rather than time and speed
- Compare the difference in accuracy using different speeds. Record your results
- Compare the difference in accuracy using time vs distance. Record your results

Review and Reflection Prompts

Encourage students to regularly review what they have learnt, how they worked as a team and what they, personally can do to make things better, what they would do differently next time. This is a key step in the design process.

- How did you work well as a team – what did you do?
- What could **you** do to make things better next time?
- What went wrong? – Why?
- What went well? – Why?
- What can I do now (WALT)?
- Which image captures the feeling from this step?
- What can I add to our engineering journal?
- Anything else to share or discuss at this stage?

Curriculum Links:
Level 1: Give and follow instructions for movement that involve distances, directions, half or quarter turns.

Level 2: Create and use appropriate units and devices to measure length, turn and time

Level 3: Use linear scales and whole numbers of metric units for length, angle and time.

Level 4: Use appropriate scales, devices and metric lengths for length, angle and time

Level 6: measure at a level of precision appropriate for the task

Key Competencies:
- Using language, symbols and text

Vocabulary List:
- Sensor
- Pseudocode
- Reflected light intensity

Items and equipment you need:
- Paper, pencil/pen
- Robot
- Laptop

We can now:
- Create line following code
- Explain why the robot code works
- Explain what the code does
- Troubleshoot the code

Step 5: Line Following Code

Pre-activities:
- **Learn about how colour sensors work**
https://youtu.be/if1yk4WiaiQ
- **Learn how colour sensors work for Arduino**
https://youtu.be/MwdANEcTiPY

The basics of the Rescue code are Line Following. Line following code means that the robot will follow a black line no matter where the line leads.

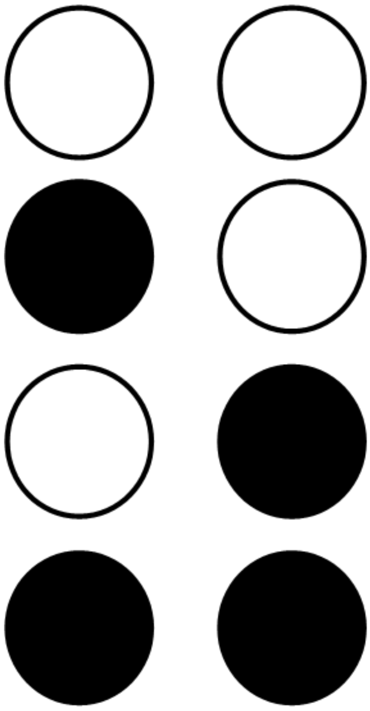
We will be using reflected light intensity and the Lego Mindstorm coding language to work through the examples. However, the theory can easily be applied to a different coding language/platform with the light options available. You cannot run this code effectively using "black" and "white" colours. You can also use PID line following – we are not covering this as it would be considered extension.

Reflected light intensity is a measure of how much light is being read by the sensor. The sensor sends a beam of red light onto the surface and depending on how much light is absorbed by the surface will depend how much light is reflected back. Each colour has a different reading and the reading will change depending on the light in the room and where the light is, e.g. a very bright room will have different reading to a dark room.

The theory behind the line following code is below.

There are two colour sensors. When they pass over the line there are four options that can be shown.

LEFT SENSOR RIGHT SENSOR



1. The first option is seeing white under both sensors. This means that the line is directly under the sensors. In this case we want to drive forward
2. The second option is black under the left sensor and white under the right sensor. This means that the robot is heading to the right and needs to turn back to the left to come back onto the line.
3. The third option is black under the right sensor and white under the left sensor. This is the opposite of the option above and we need to turn right to come back onto the line
4. The fourth option is that the robot sees black under both sensors. This usually means that it is at a crossroad. At this stage we just want to stop the robot. Later we will add the turning here.

Begin by exploring the pseudocode with your students.
The code needs to occur over and over so should be in a forever loop

```
Forever
        Check if the left sensor has white IF YES
                Then Check if the right sensor has white IF YES
                        GO FORWARDS
                        IF NO Turn Right
                IF NO
                        Then Check if the right sensor has white IF YES
                        Turn Left
```

IF NO

Then STOP

Have students test this pseudo code themselves using a black line and their own eyes

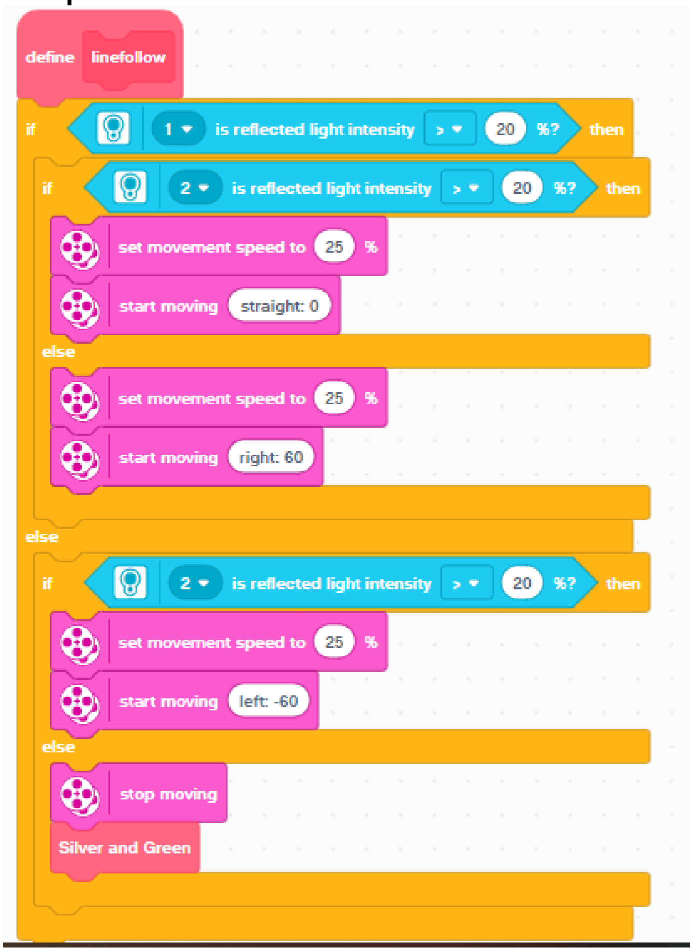
How to find reflected light intensity:

Place the robot over black and get a reading from the colour sensor. This will tell you what black is. You should allow a little leeway and select a number bigger than this to be the > than value

Next step is to create the code in your programming language.
Students should use their pseudocode and the readings they have taken to create their code attached to when started code.

If unsure about what to do here is a basic code – for now ignore the block that says silver/green

Sample Code



Extension for older students:

- Use PID to create a line following code
- Explain what PID is and how it works

Review and Reflection Prompts

Encourage students to regularly review what they have learnt, how they worked as a team and what they, personally can do to make things better, what they would do differently next time. This is a key step in the design process.

- How did you work well as a team – what did you do?
- What could **you** do to make things better next time?
- What went wrong? – Why?
- What went well? – Why?
- What can I do now (WALT)?
- Which image captures the feeling from this step?
- What can I add to our engineering journal?
- Anything else to share or discuss at this stage?

Curriculum Links:

Level 1: Give and follow instructions for movement that involve distances, directions, half or quarter turns.

Level 2: Create and use appropriate units and devices to measure length, turn and time

Level 3: Use linear scales and whole numbers of metric units for length, angle and time.

Level 4: Use appropriate scales, devices and metric lengths for length, angle and time

Level 6: measure at a level of precision appropriate for the task

Computational Thinking Progress Outcome 1

Computational Thinking Progress Outcome 2

Computational Thinking Progress Outcome 3

Computational Thinking Progress Outcome 4

Key Competencies:

- Thinking
- Using text, symbols and language

Vocabulary List:

- Ultrasonic
- Myblock/Function
- Obstacle
- Avoidance

Items and equipment you need:

- Pencil
- Paper
- Robot
- Laptop

We can now:

- Explain how an ultrasonic sensor works
- Create pseudocode that shows how the ultrasonic sensor will work
- Create code to use the ultrasonic sensor

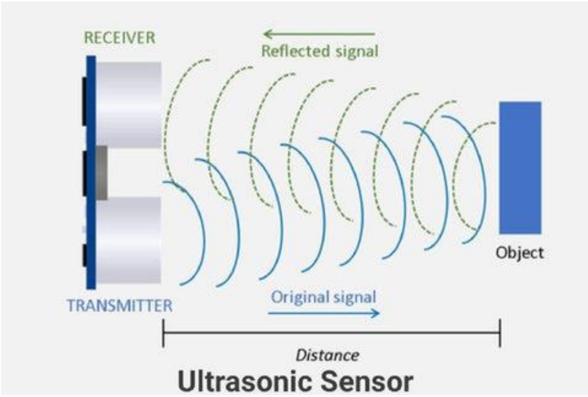
Step 6: Avoiding Obstacles

Pre-activities:

- **Learn how an ultrasonic sensor works**

a) https://youtu.be/2ojWO1QNprw

An ultrasonic sensor works by sending out a sound wave. When the wave hits the object, it bounces back and the sensor measures the time taken and uses that to work out the distance.



There is a moment between when the obstacle is identified and when it moves into the new code.

Avoiding obstacles requires identifying an obstacle in front of the robot and then going around the obstacle.

It's easiest to do this by creating a set of code that makes the robot go around the 1L bottle first. Create this code using forwards, turns as needed, ensure that the robot does not leave the square. (no example code is shown for this).

After you have created the "go around" code then you need to create the code that identifies an obstacle.

Start with some pseudocode:

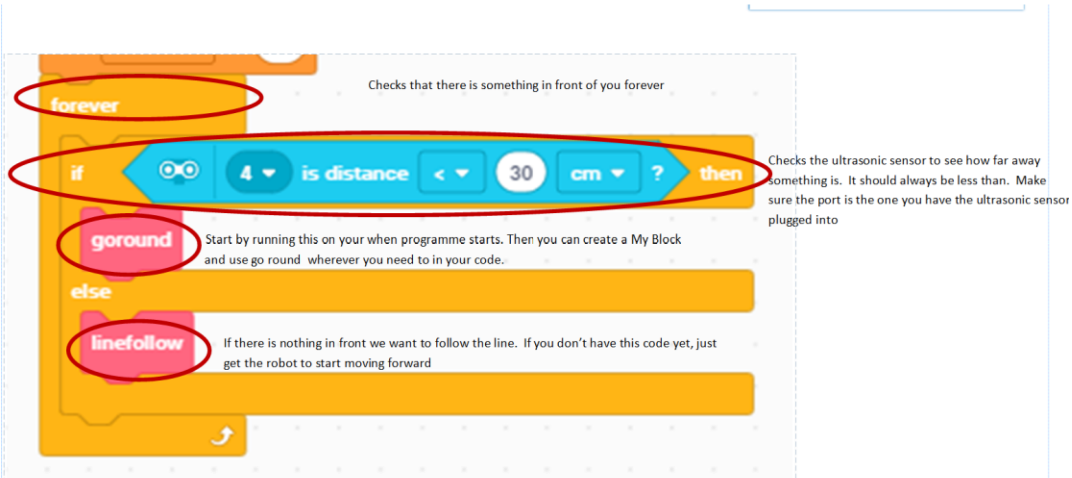
Forever
If there is something >= 15cm in front IF YES
    Then GO around it
IF NO
    Then GO forward a little bit

Finding how far away an obstacle is.
This is a bit of a trial and error to find how far away you need to turn. When testing the code, you will need to adjust the code you have already created to go around the bottle.

Finishing off
Combining the line follower and obstacle avoidance gives the main part of the Rescue code.

Start by creating the line follower as a function or a my block (if using Lego Programming). Make sure to only include the IF statements NOT The Forever.

Next add the MyBlock or Function into the part of the obstacle avoidance code where you had the go forward a little bit



Test your code, especially over the judder bars and the ramp.

Extension for older students:
- Ensure your robot is taking the most efficient route around the bottle

Review and Reflection Prompts

Encourage students to regularly review what they have learnt, how they worked as a team and what they, personally can do to make things better, what they would do differently next time. This is a key step in the design process.
- How did you work well as a team – what did you do?
- What could **you** do to make things better next time?
- What went wrong? – Why?
- What went well? – Why?
- What can I do now (WALT)?
- Which image captures the feeling from this step?
- What can I add to our engineering journal?
- Anything else to share or discuss at this stage?

Step 7: Turning Corners on Green

Pre Activities:

- Explore Self driving cars and how they work
- https://youtu.be/PRg5RNU_JLk?si=VVQfKxPhr21zLPOz
- https://youtu.be/G2OU_IzsMdE?si=eWTgbj0yUzMOgqv3

Curriculum Links:
Level 1: Give and follow instructions for movement that involve distances, directions, half or quarter turns.

Level 2: Create and use appropriate units and devices to measure length, turn and time

Level 3: Use linear scales and whole numbers of metric units for length, angle and time.

Level 4: Use appropriate scales, devices and metric lengths for length, angle and time

Level 6: measure at a level of precision appropriate for the task

Computational Thinking Progress Outcome 1

Computational Thinking Progress Outcome 2

Computational Thinking Progress Outcome 3

Computational Thinking Progress Outcome 4

Key Competencies:

- Thinking
- Using text, symbols and language

Vocabulary List:

- Myblock/function
- Reflected light intensity

Items and equipment you need:

- Pencil
- Paper
- Robot
- Laptop

We can now:

- Create pseudocode to turn on the green corner
- Create code that turns the robot the correct way on the green corner

Turning the correct way at a junction or intersection relies on identifying which sensor has the green square under it.

Start by putting the robot over the green square and noting the reading from the sensor. Remember to give it a bit of a leeway and use this as the less than value for green.

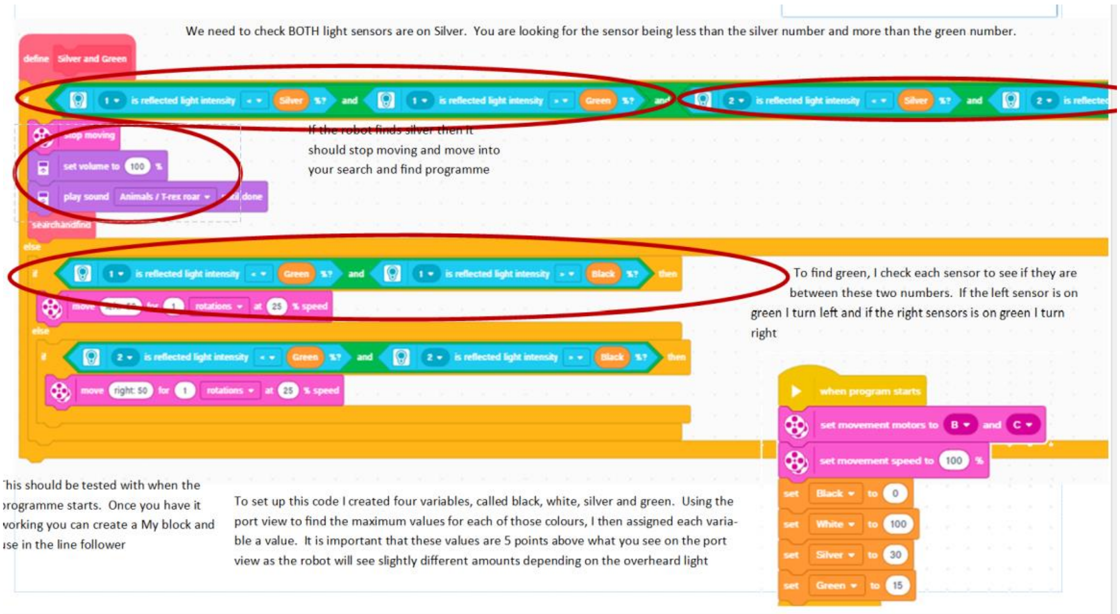
We want the robot to turn when it sees a value greater than black and less than green on one side and black on the other side.
The robot may need to move back before turning and/or move forward after turning to be in the best position to move

Pseudocode

If left sensor is less than green and greater than black
IF YES
THEN move back a little bit, turn 90 degree and move forward a little

Test the code to ensure that it works by placing it on the green square.

Repeat this process for the right sensor.
To add the two sets of code into the larger main rescue code create a function or myblock for turning right and a function for turning left. Then add these into the STOP section.



Extension for older students:

- Use PID for turning on green

Review and Reflection Prompts

Encourage students to regularly review what they have learnt, how they worked as a team and what they, personally can do to make things better, what they would do differently next time. This is a key step in the design process.

- How did you work well as a team – what did you do?
- What could **you** do to make things better next time?
- What went wrong? – Why?
- What went well? – Why?
- What can I do now (WALT)?
- Which image captures the feeling from this step?
- What can I add to our engineering journal?
- Anything else to share or discuss at this stage?

Curriculum Links:
Level 1: Give and follow instructions for movement that involve distances, directions, half or quarter turns.

Level 2: Create and use appropriate units and devices to measure length, turn and time

Level 3: Use linear scales and whole numbers of metric units for length, angle and time.

Level 4: Use appropriate scales, devices and metric lengths for length, angle and time

Level 6: measure at a level of precision appropriate for the task

Computational Thinking Progress Outcome 1

Computational Thinking Progress Outcome 2

Computational Thinking Progress Outcome 3

Computational Thinking Progress Outcome 4

Key Competencies:
- Thinking
- Using text, symbols and language

Vocabulary List:
- Myblock/function
- Reflected light intensity

Items and equipment you need:
- Pencil
- Paper
- Robot
- Laptop

We can now:
- Create pseudocode to locate the silver strip
- Create code that stops the robot on the silver strip

Step 8: Finding Silver

Once you have a line follower that avoids obstacles and can turn on the green squares correctly you have the last challenge. The last challenge requires you to no longer avoid obstacles, but instead to locate them.

The trigger for swapping from avoiding an obstacle to locating an obstacle (can) is to cross and identify the silver strip.

This is similar to what you have created for turning on green. The difference is your robot now needs to differentiate between silver, green and black.

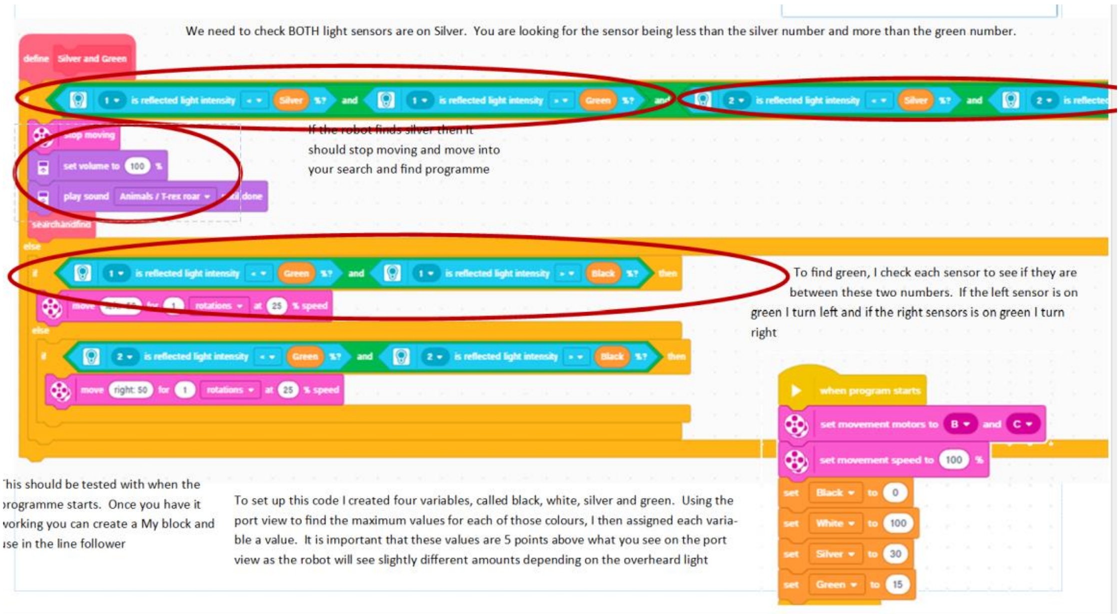
Pseudocode

If left sensor OR Right sensor in the range for silver and Isn't green or black
IF YES
THEN Play a sound (eventually this will then do find the can)
Else
Find Green

Test the code to ensure that it works by placing it on the silver strip

To add this set of code into the larger main rescue code either alter your find green code to include your silver. OR create a function or myblock to locate silver.

Extension for older students:
- Explore more efficient ways of creating this code
- Does PID work with finding silver
- Can you go faster when locating the colours and line following



Review and Reflection Prompts

Encourage students to regularly review what they have learnt, how they worked as a team and what they, personally can do to make things better, what they would do differently next time. This is a key step in the design process.

- How did you work well as a team – what did you do?
- What could **you** do to make things better next time?
- What went wrong? – Why?
- What went well? – Why?
- What can I do now (WALT)?
- Which image captures the feeling from this step?
- What can I add to our engineering journal?
- Anything else to share or discuss at this stage?

Step 9: Search Pattern

To locate the can and move it you need 2 sections.

The first is what to do when the item is located?
Should you move faster toward it and push it off?
Move forwards until I see it 5cm away?
Turn?
Pick it up?

Create pseudocode that runs through what you want to happen.
Next create and test code that enacts your pseudocode.

The second set of code is for a Search Pattern to find the can.

A search pattern can be a grid search or the easiest option which is to drive to the middle of the green area and then rotate slowly until the can is seen.
Create pseudocode and test.
Then create code.
Test your code to ensure it works before adding into the main section.
You may need to adjust your search pattern based on your silver selection response.

Extension for older students:
-Rather than keep running, make your code stop pushing once the can is off the green
-Rather than pushing, pick the can up and put it out of the green
-Or place the can on an elevated space next to the green

Curriculum Links:
Level 1: Give and follow instructions for movement that involve distances, directions, half or quarter turns.

Level 2: Create and use appropriate units and devices to measure length, turn and time

Level 3: Use linear scales and whole numbers of metric units for length, angle and time.

Level 4: Use appropriate scales, devices and metric lengths for length, angle and time

Level 6: measure at a level of precision appropriate for the task

Computational Thinking Progress Outcome 1

Computational Thinking Progress Outcome 2

Computational Thinking Progress Outcome 3

Computational Thinking Progress Outcome 4

Key Competencies:
- Thinking
- Using text, symbols and language

Vocabulary List:
- Myblock/function
- Reflected light intensity

Items and equipment you need:
- Pencil
- Paper
- Robot
- Laptop

We can now:
- Create pseudocode to search and locate the can
- Create code that searches and locates the can

Review and Reflection Prompts

Encourage students to regularly review what they have learnt, how they worked as a team and what they, personally can do to make things better, what they would do differently next time. This is a key step in the design process.

- How did you work well as a team – what did you do?
- What could **you** do to make things better next time?
- What went wrong? – Why?
- What went well? – Why?
- What can I do now (WALT)?
- Which image captures the feeling from this step?
- What can I add to our engineering journal?
- Anything else to share or discuss at this stage?

Curriculum Links:
Level 1: Is becoming reflective about the production of own texts

Level 2: Construct text that demonstrates a growing awareness of audience and purpose through appropriate choice of context, language and text form.

Level 3: Use oral, written and visual language to create meaning and effect and engage interest

Level 4: Is reflective about the production of own texts, monitors and self-evaluates progress, articulate learning with confidence

Level 5: Achieves a sense of coherence and wholeness when constructing texts

Level 6 - 8: Uses a wide range of oral, written and visual language features fluently and with control to create meaning and effect and sustain interest

Organises and develops ideas and information for a particular purpose or effect, using the characteristics and conventions of a range of text forms with control

Key Competencies:
- Using language, symbols and text
- Relating to others

Vocabulary List:
- Trial
- Progress
- Teamwork
- Decision making
- Learning Journey

Items and equipment you need:
- Engineering journal

We can now:
- Discuss our learning journey
- Answer questions about our programming and performance
- Determine who will answer specific types of questions

Step 10: Interview

Pre-activities:

- Perform 30 second impromptu speeches on a range of topics
- Watch inspiring speakers, what do they do?
- Watch awful presentations, what do they do?
- Have students sit in the "hot seat" for 2mins and tell a story to a small group, receive and give critical feedback
- Film students presenting and have them analyse their presentation
- Have students write 30 second speeches and have someone else present them. Both parties should give feedback

The interview
The judges are interested in learning how your robot and how you worked together. Your engineering journal will be of great help here.

Judges are looking for:
1) Did you build this yourselves, was it a constructed robot or a commercial one?
2) Did you make a stable robot that won't tip over?
3) What problems did you have when you started to programme and how did you solve them?
4) Engineering journal showing your trials, progress, failures etc.
5) Can you explain what the programme does, can you tell me what happens if you change x?
6) Did you use "if statements:", "repeats", "sensing"?
7) How have you used timers, sensors etc.?
8) How have you used any unusual technologies?
9) What did you learn from your experience?
10) How did you work as a team?

Prior to the competition decide who will answer which kinds of questions, such as who will talk about the programming, who will talk about the engineering journal, who will talk about team work, who will talk about the line following, turning on green, obstacle avoidance, can collection etc.

Discuss and summarise your learning journey. Start with how you came up with your design, how you added the sensors. How you programmed the robot and what challenges that bought. How you used SCRUM to manage your time.

Extension for older students:
- Deliver a presentation on your learning journey to the school
- Mentor a younger team and help them prepare their interview
- Score each other using the marking schedules

Review and Reflection Prompts

Encourage students to regularly review what they have learnt, how they worked as a team and what they, personally can do to make things better, what they would do differently next time. This is a key step in the design process.
- How did you work well as a team – what did you do?
- What could **you** do to make things better next time?
- What went wrong? – Why?
- What went well? – Why?
- What can I do now (WALT)?
- Which image captures the feeling from this step?
- What can I add to our engineering journal?
- Anything else to share or discuss at this stage?

Credits and Acknowledgements

Step 1
Image 1 - https://hygger.io/blog/how-to-use-scrum-task-board/
Image 2 - https://www.visual-paradigm.com/scrum/10-basic-scrum-rules/
Image 3 - https://www.visual-paradigm.com/scrum/why-fixed-length-of-sprints-in-scrum/

Step 2
Image 1 - https://www.sciencebuddies.org/science-fair-projects/engineering-design-process/engineering-design-notebook
Image 2 – J Cathro

Step 3
Image 1 - https://www.dowlingmagnets.com/blog/tag/engineering-magnets/

Step 5
Image 1 - https://le-www-live-s.legocdn.com/sc/media/images/resource-site/files/ev3_chromebook_userguide_us_color_sensor-169c6ea887da48723fa61f7a353b3f87.pdf

Step 6
Image 1 - https://robocraze.com/blogs/post/what-is-ultrasonic-sensor

Step 7
Image 1 – https://www.eventfinda.co.nz/2019/robocup-jnr-theatre-te-aka-mauri/rotorua
Image 2 - https://docs.microsoft.com/en-us/archive/blogs/developingfordynamicsgp/robocup-junior-australia-2014-wa-state-championships
Image 3 - https://docs.microsoft.com/en-us/archive/blogs/developingfordynamicsgp/robocup-junior-australia-2014-wa-state-championships

Step 9
Image 1 – J Cathro
Image 2 – J Cathro

Step 11
Image 1 – J Cathro