CreateBase

Educator Guide

Send It

Send it Project •

Project Overview

"Dinosaur Game" - by Google

Background

In a jumping game, a character is endlessly running at an infinite amount of obstacles. The player must decide when the character jumps to avoid running into an obstacle for as long as possible.

Synopsis

In this project, users will be aiming to automate our version of this game by creating an "AI" (Artificial intelligence) that can exceed human capabilities and achieve as high of a score as possible. This AI will be controlling a robot to deliver a package as fast as possible, automatically jumping over any obstacles that get in its way.

Themes of this Project include **automation** and **algorithms**, including thinking about potential use-cases and moral dilemmas.

Sub-themes also include congestion and the delivery of goods using autonomous robots.

This project is divided into six lesson plans

01. Let learners **Imagine** the situation by playing and discussing a situation video. Learners will work in groups to explore the Project theme to accurately **Define** the situation.

02. Facilitate a class discussion around the topics and questions that your learners previously covered in **Define**. This lesson will end with an explicit definition of the problem and the tools available to solve it.

03. Learners will get a chance to **Research** the tools available on our platform that they will use to construct their solution. This lesson will end with a session where learners will **Plan** how they will build their solutions.

04. Learners will use our platform to **Create** and test their solutions to the problem inside our simulated environment.

05. Learners will use our platform to **Improve** upon their previous solutions, applying the skills they have learnt and the knowledge they have gained to solve more advanced problems.

06. Learners will continue using our platform to **Improve** upon their solutions, before taking the time to **Review** their entire work on the Project.

Project Overview

Subject Areas

- Technology
- Engineering
- Computer Science
- Automation
- Artificial Intelligence (AI)
- Ethics

Learning Outcomes

In this Project, **learners** will:

1. Learn how to formalise the logic occurring in their head into logical code to allow them to automate a task they would normally do manually.
2. Learn how to use flow control/ branching (If, else, else if) to write code that can make decisions.
3. Learn how to use comparison blocks (<, >, <=, >, ==) to inform decisions making by comparing two different values.
4. Learn how to read and act on sensor data returned by a robotic system (distance to next obstacle, height of next obstacle, obstacle type, obstacle velocity etc).

Equipment List

Learners require:

- Access to our digital platform through a laptop, PC or tablet (no account needed at this stage)
- Learning journal (included in lesson plan)

Educators require:

- Situation video (included in lesson plan)
- Printed Assessment worksheets (file included in lesson plan)
- Easy access to help sheets, either printed or digitally (files included in lesson plan)

Curriculum Standards

Curriculum standards that this Project aligns with.

Covered | Moderately covered

Achievement objectives (tasks)					
Technological practice	Level 1	Level 2	Level 3	Level 4	Level 5
Planning for practice	Outline a general plan to support the development of an outcome, identifying appropriate steps and resources.	Develop a plan that identifies the key stages and the resources required to complete an outcome.	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.	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.	Analyse their own and others' planning practices to inform the selection and use of planning tools. Use these to support and justify planning decisions (including those relating to the management of resources) that will see the development of an outcome through to completion.
Brief development	Describe the outcome they are developing and identify the attributes it should have, taking account of the need or opportunity and the resources available.	Explain the outcome they are developing and describe the attributes it should have, taking account of the need or opportunity and the resources available.	Describe the nature of an intended outcome, explaining how it addresses the need or opportunity. Describe the key attributes that enable development and evaluation of an outcome.	Justify the nature of an intended outcome in relation to the need or opportunity. Describe the key attributes identified in stakeholder feedback, which will inform the development of an outcome and its evaluation.	Justify the nature of an intended outcome in relation to the need or opportunity. Describe specifications that reflect key stakeholder feedback and that will inform the development of an outcome and its evaluation.
Outcome development and evaluation	Investigate a context to communicate potential outcomes. Evaluate these against attributes; select and develop an outcome in keeping with the identified attributes.	Investigate a context to develop ideas for potential outcomes. Evaluate these against the identified attributes, select, and develop an outcome. Evaluate the outcome in terms of the need or opportunity.	Investigate a context to develop ideas for potential outcomes. Trial and evaluate these against key attributes to select and develop an outcome to address the need or opportunity. Evaluate this outcome against the key attributes and how it addresses the need or opportunity.	Investigate a context to develop ideas for feasible outcomes. Undertake functional modelling that takes account of stakeholder feedback in order to select and develop the outcome that best addresses the key attributes. Incorporating stakeholder feedback, evaluate the outcome's fitness for purpose in terms of how well it addresses the need or opportunity	Analyse their own and others' outcomes to inform the development of ideas for feasible outcomes. Undertake ongoing functional modelling and evaluation that takes account of key stakeholder feedback and trialling in the physical and social environments. Use the information gained to select and develop the outcome that best addresses the specifications. Evaluate the final outcome's fitness for purpose against the brief.

Covered Moderately covered

Achievement objectives (tasks)					
Technological knowledge	Level 1	Level 2	Level 3	Level 4	Level 5
Technological modelling	Understand that functional models are used to represent reality and test design concepts and that prototypes are used to test technological outcomes.	Understand that functional models are used to explore, test, and evaluate design concepts for potential outcomes and that prototyping is used to test a technological outcome for fitness of purpose.	Understand that different forms of functional modelling are used to inform decision making in the development of technological possibilities and that prototypes can be used to evaluate the fitness of technological outcomes for further development.	Understand how different forms of functional modelling are used to explore possibilities and to justify decision making and how prototyping can be used to justify refinement of technological outcomes.	Understand how evidence, reasoning, and decision making in functional modelling contribute to the development of design concepts and how prototyping can be used to justify ongoing refinement of outcomes.
Technological products	Understand that technological products are made from materials that have performance properties.	Understand that there is a relationship between a material used and its performance properties in a technological product.	Understand the relationship between the materials used and their performance properties in technological products.	Understand that materials can be formed, manipulated, and/or transformed to enhance the fitness for the purpose of a technological product.	Understand how materials are selected, based on desired performance criteria.
Technological systems	Understand that technological systems have inputs, controlled transformations, and outputs.	Understand that there are relationships between the inputs, controlled transformations, and outputs occurring within simple technological systems.	Understand that technological systems are represented by symbolic language tools and understand the role played by the "black box" in technological systems.	Understand how technological systems employ control to allow for the transformation of inputs to outputs.	Understand the properties of subsystems within technological systems.
Nature of technology					
Characteristics of technology	Understand that technology is purposeful intervention through design.	Understand that technology both reflects and changes society and the environment and increases people's capability.	Understand how society and environments impact on and are influenced by technology in historical and contemporary contexts and that technological knowledge is validated by successful function.	Understand how technological development expands human possibilities and how technology draws on knowledge from a wide range of disciplines.	Understand how people's perceptions and acceptance of technology impact on technological developments and how and why technological knowledge becomes codified.
Characteristics of technological outcomes	Understand that technological outcomes are products or systems developed by people and have a physical nature and a functional nature.	Understand that technological outcomes are developed through technological practice and have related physical and functional natures.	Understand that technological outcomes are recognisable as fit for purpose by the relationship between their physical and functional natures.	Understand that technological outcomes can be interpreted in terms of how they might be used and by whom and that each has a proper function as well as possible alternative functions.	Understand that technological outcomes are fit for purpose in terms of time and context. Understand the concept of malfunction and how "failure" can inform future outcomes.

Curriculum Standards

Curriculum standards that this Project aligns with.

Covered | Moderately covered

	Progress outcomes (assessment)				
	Level 1	Level 2	Level 3	Level 4	Level 5
Computational thinking for digital technologies	In authentic contexts and taking account of end users, learners 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).	In authentic contexts and taking account of end users, learners 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.	In authentic contexts and taking account of end-users, learners 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, sequence and iteration (repeating part of the algorithm with a loop). They understand that digital devices store data using just two states represented by binary digits (bits).	In authentic contexts and taking account of end-users, learners 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.	In authentic contexts and taking account of end users, learners independently decompose problems into algorithms. They use these algorithms to create programs with inputs, outputs, sequence, selection using comparative and logical operators and variables of different data types, and iteration. They determine when to use different types of control structures. Learners document their programs, using an organised approach for testing and debugging. They understand how computers store more complex types of data using binary digits, and they develop programs considering human-computer interaction (HCI) heuristics.
Designing and developing digital technologies	In authentic contexts and taking account of end users, learners participate in teacher-led activities to develop, manipulate, store, retrieve and share digital content in order to meet technological challenges. In doing so, they identify digital devices and their purposes and understand that humans make them. They know how to use some applications, they can identify the inputs and outputs of a system, and they understand that digital devices store content, which can be retrieved later.		In authentic contexts and taking account of end-users, learners make decisions about creating, manipulating, storing, retrieving, sharing and testing digital content for a specific purpose, given particular parameters, tools, and techniques. They understand that digital devices impact on humans and society and that both the devices and their impact change over time. Learners identify the specific role of components in a simple input-process-output system and how they work together, and they recognise the "control role" that humans have in the system. They can select from an increasing range of applications and file types to develop outcomes for particular purposes.	In authentic contexts, learners follow a defined process to design, develop, store, test and evaluate digital content to address given contexts or issues, taking into account immediate social, ethical and end-user considerations. They identify the key features of selected software and choose the most appropriate software and file types to develop and combine digital content. Learners understand the role of operating systems in managing digital devices, security, and application software and are able to apply file management conventions using a range of storage devices. They understand that with storing data comes responsibility for ensuring security and privacy.	

Project Overview

Adapting the lessons

The times given for each step of the Creation Process are just for guidance. Depending on how much time you have available and how in-depth you want to discuss each topic, you could spend up to twice as long on each step of the creation process.

We recommend using the times as a minimum estimate to help you with your own planning. You are free and encouraged to adjust, skip and/or repeat steps depending on the engagement and aptitude of your class and your desired goals for the lesson.

Online learning

These lesson plans can be used for online learning.

If your school runs structured lessons through video meeting platforms, we recommend using breakout rooms for class discussions. All other tasks can be completed in the lesson as per the lesson plan or as homework activities.

If an individual learner needs additional help or guidance, you can ask them to temporarily share their screen (potentially in a breakout room) so that you can see what they are doing and guide them more closely.

Suppose your school is not running regular lessons or has poor turnout. In that case, you can instead skip over the class discussion parts of the lesson plans and assign learners individual tasks (e.g. complete the Research step by DD/MM/YYYY).

We recommend you use the learning journal to set completion targets for your class and have them share with you their progress by making their learning journals available to you through email or google classroom.

Overview

Imagine (15 mins)

Introduce and discuss the Project with your learners with a video that showcases the situation.

Define Part I (30 mins)

Let your learners work in groups to explore the Project theme and define the situation accurately.

Lesson Resources

Situation cutscene

Presentation slides for Define

Glossary

1. **Artificial intelligence (AI):** Broadly speaking, artificial intelligence (AI) is all about creating programs and machines that can carry out human behaviours like critical thinking and decision making. Generally, the quality of an AI is based on the complexity and the accuracy of the tasks it can complete.

   E.g. AI can be as simple as an automatic light switch or as complex as responsive conversation software like Siri.

2. **Automation:** Automation is all about using technology to carry out tasks that a human would have traditionally carried out. Examples include robots in car manufacturing assembly lines and self-driving trains. Humans are good at critical thinking, whereas most technology is great at repeatedly performing the exact same action. This is why routine tasks (tasks that involve performing the same action repeatedly) are usually the first to be automated as little or no critical thinking is needed.

   E.g. robots in car manufacturing assembly lines and self-driving trains.

3. **Ethics:** Ethics are all about how a person determines which actions are good and bad. What is the reasoning behind their decision? There is not always a right and a wrong answer regarding ethics, as different people place different amounts of value on different concepts. Ethics are important as they can help us understand why other people make and justify their decisions and understand ourselves.

   E.g. "Would you lie to convict a criminal that you know is guilty?" Whether or not a person values truth or justice more is likely to influence their answer to this question.

4. **Congestion:** Congestion occurs when a location is so crowded that it becomes difficult to move. This normally happens when a large amount of something is trying to move through a small opening.

   E.g. Cars on a motorway or water down a drain.

Lesson 1 •

Imagine 15 mins

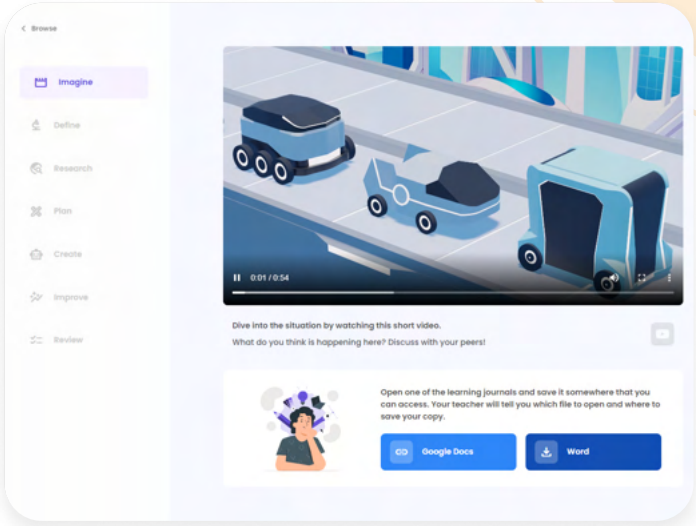
1

Follow these steps

1. Bring all of your learners to the front of the class if possible.

2. Welcome the class and tell them that they will be completing a **Project** over the following six lessons where they will get to plan, build, test, and improve a solution to a problem.

3. Tell the class that we will begin by playing a cutscene of the situation to set the scene, then play the following video:



https://app.createbase.co.nz/send-it

4. Discuss with the class what happened in the video.

   Discussion points could be based on:

   - **What the main character is doing**
   - **The problem he encounters**
   - **What he is trying to get the robot to do**

If they understand the problem, you might want to ask learners what they would do in the same situation.

Tips for Educators

For your reference, here is a quick explanation of what is happening in the video:

- On route for package delivery, there are major delays during transit.
- To ensure that the package arrives in time, you resort to sending out your delivery robot to beat the traffic.
- In a rush, you must code the robot to navigate the obstacles before it gets hit and loses the package.

Helpful Examples

If you are finding it difficult to get the kids to talk about it, replay the video to the class with pauses and ask questions about what is happening in specific scenes.

Here are some example discussion points to get you started:

0:00-0:05 seconds

Q "What do you think is happening here?"

A Cars stuck in traffic.

Q "What can cause a traffic jam?"

A An accident, lots of cars on the road at once, cars changing lanes too often, traffic light malfunction etc.

0:14 seconds

Q "What information can we gather from the screen in the character's car?"

A Package to deliver but delays due to traffic mean that we can't deliver it on time.

0:32-0:39 seconds

Q "What is this thing in the back of the character's car?"

A A humanoid robot/delivery bot.

Helpful Examples

0:40-0:46 seconds

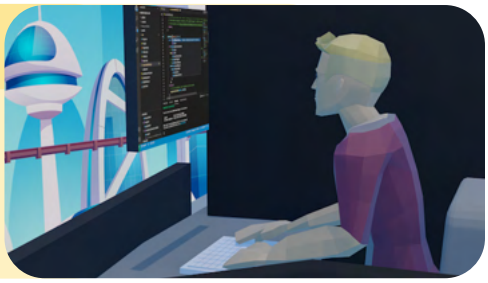
Q "What is the character doing?"

A Putting the package inside the robot.

0:47-0:49 seconds

Q "What is the character doing now?"

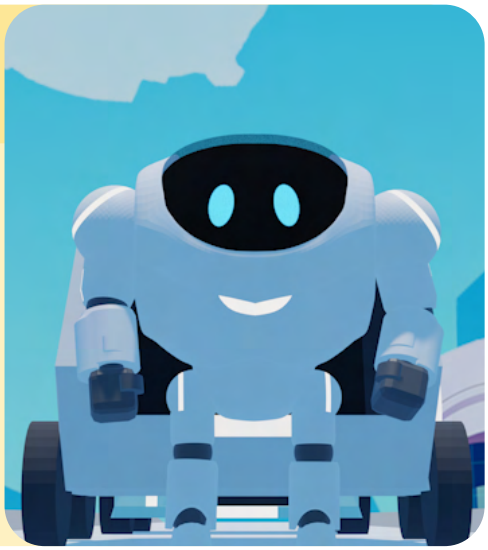
A Writing some code and uploading it to their robot.



Summarise the video

Q What has happened?

A The human character is stuck in traffic, so they can't deliver a package in time. Luckily, they have a humanoid robot that they can program to deliver their parcel.



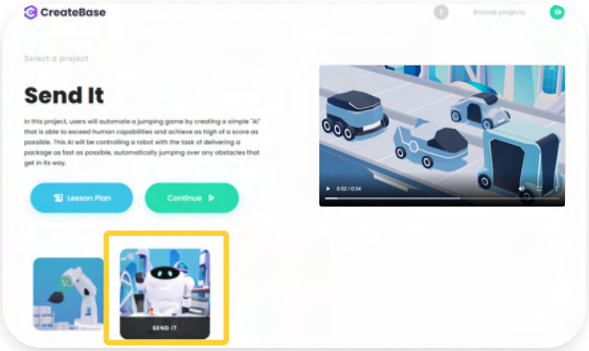
Tips for Educators

Feel free to move on to Define once you are satisfied that your class understands this project's background.

Define Part I 30 mins

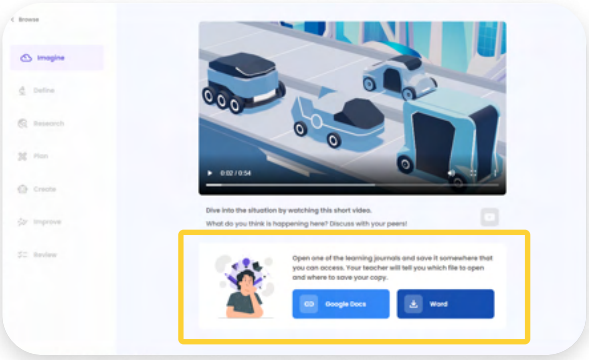
Now that you have watched the video with your class and are happy that they understand what has happened, your learners will have an opportunity to explore some of the broader topics that could stem from an autonomous delivery robot.

1. To get started, create small groups of 3-4 learners. And send them to their computers. Learners should be seated next to their group members.

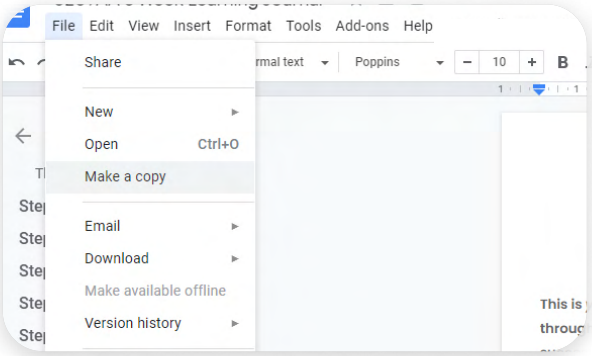


2. Tell your learners to go to https://app.createbase.co.nz/send-it

3. Ask them to rewatch the Situation video and tell them to download either the Google Docs or Microsoft Word **learning journal**.



If you want your learners to work from Google Docs, they must choose the Google Docs option and then click ***File*** -> ***Make a copy*** and save it somewhere that they can access.



4. Once downloaded, the learners should fill out their names and then move to the Define section on the website.

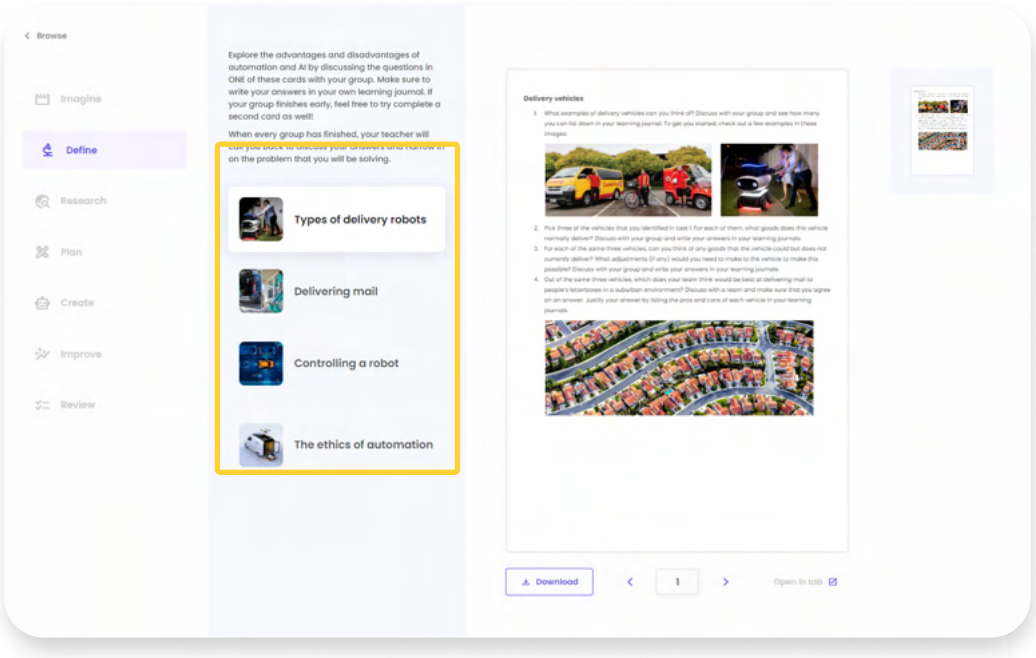
Tips for Educators

To reduce confusion, it is recommended that you tell your learners exactly where to save their files. If you want to access these files later to check on their progress, ask them to save them in a location that you can also access.

Define Part I 30 mins

5. Once in the Define section, **as a group**, tell them to select any one of the sections and discuss the questions. For each question, each learner needs to write an answer in their learning journal. Wander the room while this is happening and ask learners questions to help them start thinking.

   If a group finishes early, let them answer a second section.



6. With 5-10 minutes to go, remind the class that they must have answers for every question in at least one card in their individual learning journals by the end of the lesson.

7. At the end of the lesson, make sure that each learner has their copy of the **learning journal** saved somewhere that they can easily access at the start of the next class.

Homework

For homework, inform each learner that if they haven't already done so, they must have answers for every question in at least one of the Define cards written out in their individual learning journals by the start of the next lesson.

Overview

Define Part II (45 mins)

Facilitate a class discussion around the topics and questions that your learners covered as groups in Define Part I. learners will get a chance to share their opinions about the topics that they covered and also hear the views of others. End the lesson by explicitly stating the problem and the tools that learners can use to solve it.

Glossary

1. **Artificial intelligence (AI):** Broadly speaking, artificial intelligence (AI) is all about creating programs and machines that can carry out human behaviours like critical thinking and decision making. Generally, the quality of an AI is based on the complexity and the accuracy of the tasks it can complete.

   E.g. AI can be as simple as an automatic light switch or as complex as responsive conversation software like Siri.

2. **Automation:** Automation is all about using technology to carry out tasks that a human would have traditionally carried out. Examples include robots in car manufacturing assembly lines and self-driving trains. Humans are good at critical thinking, whereas most technology is great at repeatedly performing the exact same action. This is why routine tasks (tasks that involve performing the same action repeatedly) are usually the first to be automated as little or no critical thinking is needed.

   E.g. robots in car manufacturing assembly lines and self-driving trains.

Lesson Resources

Presentation slides for Define

3. **Ethics:** Ethics are all about how a person determines which actions are good and bad. What is the reasoning behind their decision? There is not always a right and a wrong answer regarding ethics, as different people place different amounts of value on different concepts. Ethics are important as they can help us understand why other people make and justify their decisions and understand ourselves.

   E.g. "Would you lie to convict a criminal that you know is guilty?" Whether or not a person values truth or justice more is likely to influence their answer to this question.

4. **Congestion:** Congestion occurs when a location is so crowded that it becomes difficult to move. This normally happens when a large amount of something is trying to move through a small opening.

   E.g. Cars on a motorway or water down a drain.

Lesson 2 •

Define Part II 45 mins

2

Now that your learners have had a chance to discuss some of the broader topics that could stem from an autonomous delivery robot in groups, discuss the same or similar topics as a class.

If each group in **Define Part I** chose different topics, you should now have a range of "experts" in each topic that you can call upon for their opinions.

The direction of this discussion should be dynamic and driven by learners' curiosity. Make sure to ask follow up questions, and if the learners lead you away from the prompts below, then **don't be afraid to go off-script!**

1. Introduce your class to today's lesson, saying that now we have a bunch of subject matter experts, we will be having a class discussion on the Define themes. It is important that everyone shares their knowledge and opinions to fully understand our problem from different perspectives.

> "Now we have a bunch of subject matter experts, we will be having a class discussion on the Define themes."

2. You may wish to have each learner open their **learning journals** for reference or instead stay off their device for the entire lesson, depending on how easily you think they will be distracted.

3. You now want to discuss as a whole class the themes raised in some of the Define cards. Try and get every learner thinking and participating, even if they were not in a group that discussed the particular topic you are currently on.

Tips for Educators

To get started, try using the following structure (with examples) for the discussion on the next pages:

Delivery Robots

1. Ask learners to provide examples of delivery robots, vehicles and other devices. Examples include bicycles, quad-bikes, self-driving cars, aircraft, small drones, line-following vehicles, humanoid delivery robots, transportation tubes and the internet (delivers information).

2. For each of these robots/vehicles, you could ask to follow up questions along the lines of:

> "What types of goods does this device currently deliver? Can you think of any other goods that it could be used for (that it is currently not being used to deliver)?"

If multiple suggestions are similar, ask the learners to compare the devices by listing their pros and cons.

> "Compared to a self-driving truck, would a drone be better or worse for delivering mail in a city CBD? Why? - a drone could get to the building faster, but where will it drop the package off?"

3. Now that learners have a good variety of examples of delivery robots, you can now ask them to start thinking about the environment they operate in and how this might affect the design of each robot and which one will be optimal in each situation.

Example questions might include (where X could be anything, like mail):

- "What types of obstacles might X delivery robot encounter?"
- "How would robot X detect and avoid these obstacles?"
- "When the robot does detect one of these obstacles, what should it do?"
- "What design features, like sensors, transportation mode and shape, would the robot need if it was instead used in environment Y or to deliver item Z?"

Artificial intelligence (AI)

1. With learners understanding how the environment affects a robot's design, you can now ask them how these robots should be controlled.

Ask questions that make learners think about human-controlled delivery robots vs AI-controlled robots.

"If I told you to run in front of a car, would you do it? No, you would decide that my instruction was bad and ignore it. What would happen if I programmed a robot to do the same thing?"

A big difference between a robot and a human is that a human can reject instructions and make its own decisions. A robot does exactly what it is told to do, every time. This can be good (e.g. quality control in a manufacturing line) and bad (if something goes wrong, the robot can't change its action unless it has been programmed to).

2. Explain what an AI is using the glossary as assistance.

"An AI bridges the gap between a human and a robot by allowing them to "think" and make their own decisions."

3. What are the pros and cons of controlling a vehicle using a person compared to an AI?

Things to consider include:

- **Cost** (humans have to be paid, AI need to be bought and maintained: more upfront cost but less cost over time)
- **Size** (an AI's computer takes up less space in a vehicle than a human, so the vehicle can be smaller and/or have more space for goods)
- **Risk** (the cost of human life is much more than an AI, so AI might be better for high-risk scenarios)
- **Accuracy & precision** (would an AI or a human be more accurate? why?)

Sensing Sensors

It is important for learners to understand what a sensor is and be able to draw parallels between man-made sensors like thermometers and natural sensors like human eyes. This will be useful during the Plan step. Students will identify the information they are using to make decisions when manually playing the game and convert that into an algorithm.

To get started, try having a class discussion using the following questions:

1. "How can robots react to a changing environment?"

This is usually done using sensor information to drive decisions made in the robot's code.

2. "How is this similar to how humans operate?"

Humans use their sense of Touch + Heat (sensors) to tell how hot an object is. When your fingers sense a very hot object, the brain (controller) tells your muscles to immediately remove your hand away from that object.

3. "What is a sensor?"

A sensor is a device that measures a property of its surroundings.

4. "What are some examples of sensors?"

Thermometer (measures temperature), Fire alarms (detects smoke), speedometer (measures speed), odometer (measures distance).

Ethics & Automation

1. Explain the concept of automation to the class:
   The dictionary defines automation as “the technique of making an apparatus, a process, or a system operate automatically.”

2. Ask the class if they can explain what ethics means. If not, explain the concept to the class: The dictionary defines ethics as “moral principles that govern a person's behaviour or the conducting of an activity.”

Reminder: Ethics are how a person decides which actions are good and which actions are bad.

3. Have a conversation with the class about the ethics of automation.

   Example prompts include:

“If an automated robot causes some damage, who is responsible? The robot (it only did what it was programmed to do)? The programmer? The company who sold the robot?”

“If we replace a delivery driver with an autonomous robot, what happens to the driver? They may lose their job and their income. Is this worth it if it improves delivery times? What support systems can we put in place to help support this person who has lost their job due to automation?”

Ethics & Automation

Optional: discuss the trolley problem with learners.

Situation

A train is moving down the tracks and is going to run over five people. You have a lever that you could pull to divert the train to another track, which would result in it only running over one person. Would you pull a lever to kill one person but save five?

These thought experiments can be fun activities to get learners to think critically and explore different outcomes about a hypothetical situation.

- What should an AI do in the same situation?
- What if you were a doctor with five patients in need of life-saving transplants. A healthy patient walks into your practice. Would you kill one patient to save five?
- Are there any inconsistencies in these answers? If so, why do we feel different about each situation? (maybe it is due to directly vs indirectly causing harm) One moral solution is to prioritise not killing people first and saving people second.
- If AI becomes common, there will inevitably become a situation where an AI needs to decide between two or more groups of people to harm. How do you think AI should make this decision? What variables will they need to take into account? For example, should the AI value the safety of the car's occupants more, or less, than that of potential victims outside the vehicle?

Define Part II 45 mins

4. Bring the discussion to a close by linking it back to the situation video.

5. State the problem that the learners will be solving for the remainder of the Project:

> "In this Project, your task is to create a basic AI that will tell the humanoid delivery robot in the situation cutscene what action to take when it approaches an obstacle."

6. State the functionalities of the robotic system:

> "Your robot has a sensor that detects how far away the nearest object is in front of it. You can also give the robot commands to both jump and slide."

Homework

Homework is optional for this lesson, but it may include the learners drafting their algorithms at home. If you want to make sure this is completed, tell the learners to write down or draw their thinking process for controlling the robot, ready to present during the next lesson.

Overview

Research (25 mins)

Learners will get a chance to research the tools available on our platform that they will need to use to construct their solution.

Plan (20 mins)

This lesson will end with a session where learners will plan how to construct their solutions, either as a class, in groups or as individuals.

Lesson Resources

Presentation slides for Define

Glossary

1. **Algorithm:**
   An algorithm is a sequence of rules or instructions. Computers usually follow algorithms to perform a calculation or make a decision. Good algorithms are well-defined, meaning that they have a rule or an instruction for every relevant possibility.

2. **Comparison:**
   In programming, a comparison is a statement where we compare two different values with an operator. The operator determines what characteristic we are comparing and is usually a mathematical symbol like equals, less than or greater than. The statement that we write will always be either TRUE or FALSE. For example, we might write a statement that says, "20 is less than 10." This statement would be FALSE. We might instead write "20 minus 10 equals 10." This statement would be TRUE. Comparisons are important in coding because we can use the TRUE/FALSE result to decide which branch of code we should run next.

3. **Loops:**
   In programming, a loop refers to a piece of code that is run multiple times. The number of times that the code repeats itself depends on how the loop is written.

4. **Flow control/branching:**
   In programming, the flow of code refers to the order in which code is run, and actions are performed.

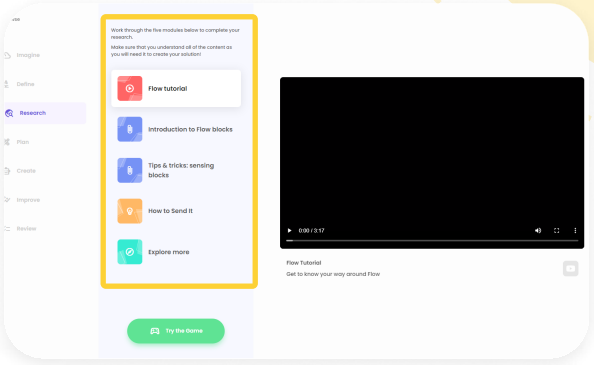
   Think about it like walking down a tile path, where each tile represents a line of code. When you stand on a tile, that line of code is run. By looking ahead at the upcoming tiles, you can see the order that each line of code will be run. If your path splits into two, you can no longer tell exactly which code will be run in advance until you decide which path to walk down. Flow control is all about how we write our code to make these decisions. Branching is the simplest form of flow control, where the code "path" splits into multiple paths.

Lesson 3 •

Research 25 mins

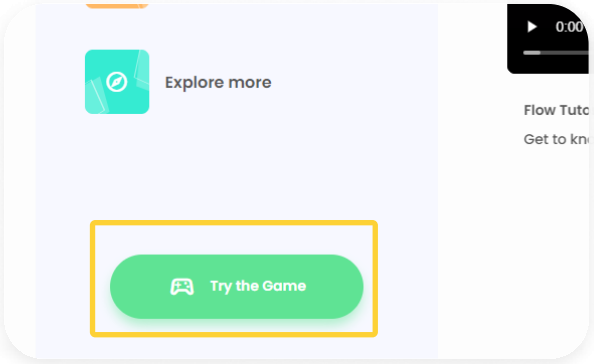
3

1. Tell the learners to return to their computers and read all of the content under the **Research** section of the platform.

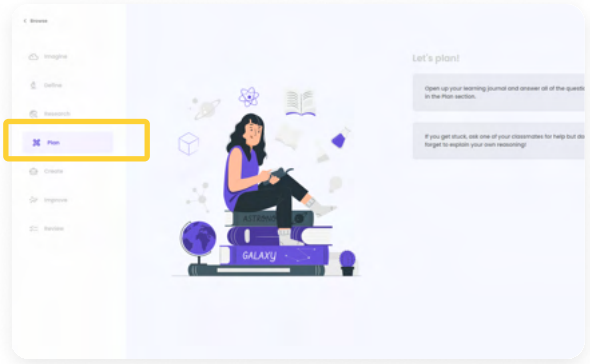


2. If a learner finishes early, tell them to try and complete the “game” manually. Tell them to think about what information they are processing when playing the game and how they are using this information to make decisions.

   How could the robot gather this information and use it to make its own decisions?



3. Once you are confident that every learner has read all of the Research content and has had a chance to play the “game”, ask them to move to the **Plan** section on the platform.



Plan 20 mins

1. Instruct learners to return to their **learning journal** and type their answers to the **Plan** questions. Give the learners a maximum of a few minutes.

2. Bring the learners away from their computers back to the front of the class when they have all finished.

3. Tell the class:

> "Now that we know all of the tools at our disposal and the basics of operating the robotic system, we will now plan how we will utilise these tools to solve the problem."

4. Start by asking learners to talk about how they played the game.

    1. What sensors were they using (eyes), and what information were they collecting (distance between the robot and the incoming cars)?

    2. What decision did you have to make (if the car is within some distance x, press the spacebar, otherwise wait)?

    3. Explain that your brain is using an if statement to decide how to control the robot. We can make a basic AI that does the same thing.

5. Walk through the process of jumping over a few obstacles with the learners, linking information and decisions back to the code blocks that they learnt in **Research**.

6. Conclude the lesson by telling the class to draft their algorithm at home, as starting from the next lesson; they will be coding and testing their solutions!

Homework

Homework is optional for this lesson, but it may include the learners drafting their algorithms at home. If you want to make sure this is completed, tell the learners to write down or draw their thinking process for controlling the robot, ready to present during the next lesson.

4

Overview

Create (45 mins)

Your learners will use our platform to create and test their solutions in our simulation. Your role as an educator will be to guide struggling learners while prompting excelling learners with questions to help them identify areas of improvement.

Glossary

1. **Algorithm:**
   An algorithm is a sequence of rules or instructions. Computers usually follow algorithms to perform a calculation or make a decision. Good algorithms are well-defined, meaning that they have a rule or an instruction for every relevant possibility.

2. **Comparison:**
   In programming, a comparison is a statement where we compare two different values with an operator. The operator determines what characteristic we are comparing and is usually a mathematical symbol like equals, less than or greater than. The statement that we write will always be either TRUE or FALSE. For example, we might write a statement that says, "20 is less than 10." This statement would be FALSE. We might instead write "20 minus 10 equals 10." This statement would be TRUE. Comparisons are important in coding because we can use the TRUE/FALSE result to decide which branch of code we should run next.

Lesson Resources

Answer document

3. **Loops:**
   In programming, a loop refers to a piece of code that is run multiple times. The number of times that the code repeats itself depends on how the loop is written.

4. **Flow control/branching:**
   In programming, the flow of code refers to the order in which code is run, and actions are performed.

   Think about it like walking down a tile path, where each tile represents a line of code. When you stand on a tile, that line of code is run. By looking ahead at the upcoming tiles, you can see the order that each line of code will be run. If your path splits into two, you can no longer tell exactly which code will be run in advance until you decide which path to walk down. Flow control is all about how we write our code to make these decisions. Branching is the simplest form of flow control, where the code "path" splits into multiple paths.

Create 45 mins

1. Welcome the class back to the Run it Down Project. If there has been a significant delay between the current and previous lesson, then you might want to give a quick recap to your learners of what you covered in your class in the last lesson.

2. Tell the class that today they will be building and testing their solutions. If you requested your learners to create a plan for their algorithm between lessons as homework, then perhaps ask a few learners to share with the rest of the class.

3. Briefly explain to the class how the Create step will work:

Build
Learners will implement their solution by starting with individual elements and building up to the final solution.

Test
Learners should constantly be testing their code to see how well their individual components or final solution solves the problem.

Iterate
Learners will continue building and testing until the problem has been completely solved.

Share
When learners think that they finished, encourage them to share their solution with other classmates who have finished and/or an educator to get their feedback. Make sure that the learners explain how they arrived at their solution. If necessary, they might need to start iterating again, either because a problem was identified or an educator/classmate gave them an idea for how to improve their solution.

4. Tell your learners to return to https://app.createbase.co.nz/ and start the Create step.

> **Tips for Educators**
>
> Attached as a lesson resource is an answer sheet with explanations for the educator's viewing only. You can use the help sheet to assist struggling learners and give additional prompts to excelling learners to make them think more about their solution.

5. If any learners finish early and you do not see any more ways for them to work on their solution in the **Create** step, then either ask them to help the other classmates or let them move onto the Improve step early.

6. At the end of the lesson, tell the class to wrap up their work by taking a screenshot of their final code and inserting it into their **learning journal** under the **Create** step. They should do this regardless of if they have a working solution or not.

7. Ask any learners who have yet to finish the **Create** step to visit the platform from home and try to get a working solution before the next class.

Overview

Improve Part I (45 mins)

Your learners will get a chance to apply the skills they have learnt and the knowledge they have gained to solve more advanced problems. We might adjust the problem parameters or solution constraints to enable new solutions.

Glossary

1. **Algorithm:**
   An algorithm is a sequence of rules or instructions. Computers usually follow algorithms to perform a calculation or make a decision. Good algorithms are well-defined, meaning that they have a rule or an instruction for every relevant possibility.

2. **Comparison:**
   In programming, a comparison is a statement where we compare two different values with an operator. The operator determines what characteristic we are comparing and is usually a mathematical symbol like equals, less than or greater than. The statement that we write will always be either TRUE or FALSE. For example, we might write a statement that says, "20 is less than 10." This statement would be FALSE. We might instead write "20 minus 10 equals 10." This statement would be TRUE. Comparisons are important in coding because we can use the TRUE/FALSE result to decide which branch of code we should run next.

Lesson Resources

Answer document

3. **Loops:**
   In programming, a loop refers to a piece of code that is run multiple times. The number of times that the code repeats itself depends on how the loop is written.

4. **Flow control/branching:**
   In programming, the flow of code refers to the order in which code is run, and actions are performed.

   Think about it like walking down a tile path, where each tile represents a line of code. When you stand on a tile, that line of code is run. By looking ahead at the upcoming tiles, you can see the order that each line of code will be run. If your path splits into two, you can no longer tell exactly which code will be run in advance until you decide which path to walk down. Flow control is all about how we write our code to make these decisions. Branching is the simplest form of flow control, where the code "path" splits into multiple paths.

Lesson 5 •

Improve Part I 45 mins

5

1. To start the lesson, confirm the number of learners who have completed the **Create** step. Tell all of the learners who have a working solution that their next task is to move on to the **Improve** step.

    a. Explain that during the **Create** step, learners had to successfully deliver the package 1,000m, jumping over cars along the way. In the **Improve** step, they must now try and deliver the package with the highest possible score.

    b. To increase their score, learners can enable **modifiers** in the setting menu. Each **modifier** adds a new obstacle to overcome; however, they will increase their final score if they can successfully deliver the package.

> Each modifier will require changes to the learner's previous solution, and learners can have as many modifiers active at once as they can handle.

    a. Recommend to the kids that they add **modifiers** one at a time, ensuring they have a working solution before adding an additional modifier.

2. If any learners have yet to complete the **Create** step, make sure they complete the **Create** step first and post a screenshot of their now working solution in their **learning journal**.

Every learner working on **Improve** should be able to show and explain to you a working solution for **Create** (as long as their answer works, it is okay, and even encouraged, if it is different from the example solution).

Improve Part I 45 mins

You should prioritise spending your time to help these learners complete the **Create** step so that they also can move on to **Improve**.

3. Let the kids who have finished the **Create** step try to complete the **Improve** step with minimal assistance. The **Improve** step helps reinforce the learners' learning, so be careful not to give the learners the exact answer yourself. Instead, ask questions to struggling learners about their code to help them realise the solutions themselves.

   An example question would be asking them to step through it one by one with you and explain their reasoning for adding each block.

"Can you show me how you came up with this solution?"

"What does this block do? Why did you place it here?"

4. If any learners can deliver the package with the highest possible score (every modifier enabled at once), ask them to help their struggling peers.

5. At the end of the **Improve** step, tell the class to wrap up their work by taking a screenshot of their final code and inserting it into their learning journal. They should do this regardless of if they have a working solution or not. At the end of this lesson, it is expected that most learners will not have achieved a fully complete solution to Improve.

Overview

Improve Part II (30 mins)

Learners will have the opportunity to continue using our platform to Improve upon their solutions.

Review (15 mins)

Learners will then review their work. Options for review include having members of the class share their unique solutions and the decision-making that got them there and/or conducting self-assessments using their learning journal.

Glossary

1. **Algorithm:**
   An algorithm is a sequence of rules or instructions. Computers usually follow algorithms to perform a calculation or make a decision. Good algorithms are well-defined, meaning that they have a rule or an instruction for every relevant possibility.

2. **Comparison:**
   In programming, a comparison is a statement where we compare two different values with an operator. The operator determines what characteristic we are comparing and is usually a mathematical symbol like equals, less than or greater than. The statement that we write will always be either TRUE or FALSE. For example, we might write a statement that says, "20 is less than 10." This statement would be FALSE. We might instead write "20 minus 10 equals 10." This statement would be TRUE. Comparisons are important in coding because we can use the TRUE/FALSE result to decide which branch of code we should run next.

Lesson Resources

Answer document

3. **Loops:**
   In programming, a loop refers to a piece of code that is run multiple times. The number of times that the code repeats itself depends on how the loop is written.

4. **Flow control/branching:**
   In programming, the flow of code refers to the order in which code is run, and actions are performed.

   Think about it like walking down a tile path, where each tile represents a line of code. When you stand on a tile, that line of code is run. By looking ahead at the upcoming tiles, you can see the order that each line of code will be run. If your path splits into two, you can no longer tell exactly which code will be run in advance until you decide which path to walk down. Flow control is all about how we write our code to make these decisions. Branching is the simplest form of flow control, where the code "path" splits into multiple paths.

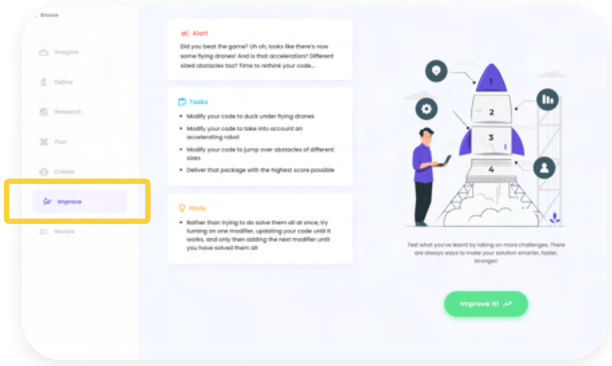
Lesson 6 •

Improve Part II 30 mins

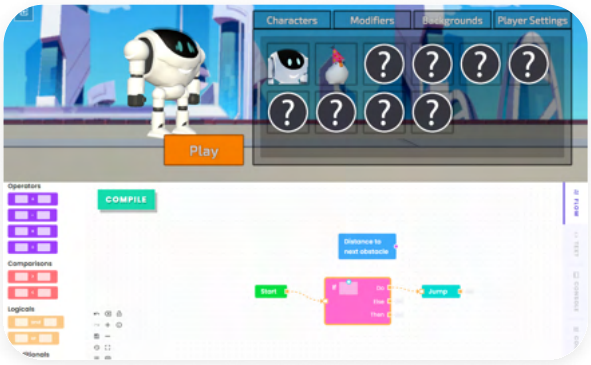
6

1. At this stage, every learner should have completed the **Create** step and be working on **Improve**. Some learners may have even completed all of the tasks in **Improve** already.

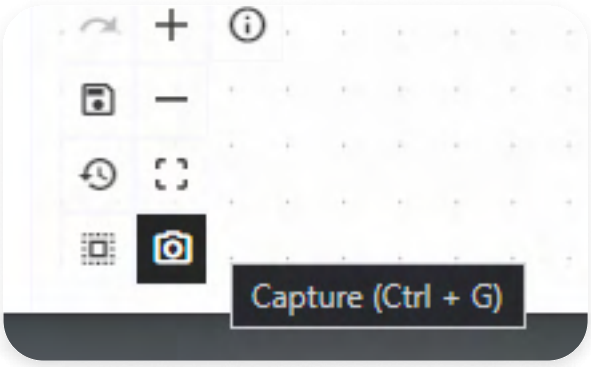
2. You should spend the first half of this lesson letting learners continue working on the **Improve** steps. You may want to instruct any learners who have already completed the **Improve** step to assist their classmates, try and optimise their code solution (for example, by minimising the number of blocks that they use) or just play the game in the **Research** step.



3. At the end of the **Improve** step, ideally, each learner has a solution that works for at least one of the modifiers: it is not expected that every learner will complete it entirely.



4. When you feel like the learners have progressed far enough into **Improve**, ask them to take a screenshot of their best solution or use the “capture” button (Ctrl + G) and insert it into their **learning journal**.



Lesson 6 •

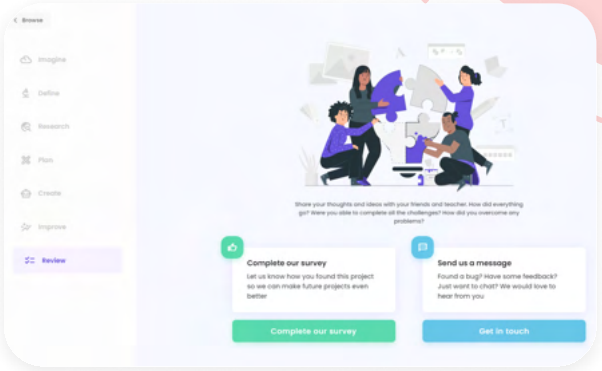
Review 15 mins

6

The final part of this Project is the **Review** step.

As part of the Review process, we provide two suggested options:

Share and **Reflect**. Feel free to do one or the other or both, depending on the time that you have available.



Share

1. Encourage your learners to talk to the person next to them about their final solutions and their decisions at each step to their classmates.

   This helps learners remember what they have done and get new ideas by listening to how others solved the problem differently, which may help with learner reflection in the next task.

2. During this process, walk around the classroom and ask learners who do not have much to say prompts to get them talking.

Reflect

1. Ask your learners to complete the Reflection section of the learning journal.

2. When every learner has completed their learning journal, you could ask some learners to share its contents and ask them questions as they do so.

3. **Optional**: At the end of the lesson, ask each learner to submit their completed learning journals somewhere where you have access in case you want to review them.