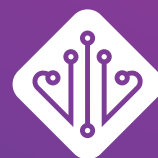


Teacher Resources

Year 4



Digital Ignition
Māpura Matihiko

Contents

01	Teacher Resource 1 (TR 1) Make an Algorithm	1
02	Teacher Resource 2 (TR 2) Let's Make a Hāngi (Me Mahi Hāngi Tātau)	6
03	Teacher Resource 3 (TR 3) Sharing Information with ScratchJr	11
04	Additional Resources	16

Teacher Resource 1 (TR 1)

Make an Algorithm

Description

Making an algorithm gives students practice at writing algorithms in a real-world setting – in this case, a dance!

Curriculum Alignment

New Zealand Arts Curriculum

Dance:

Level 1 & 2

- Demonstrate an awareness of dance in their lives and in their communities.
- Explore movement with a developing awareness of the dance elements of body, space, time, energy, and relationship.
- Share dance movement through informal presentation and share their thoughts and feelings in response to their own and others' dances.
- Identify and describe dance in their lives and in their communities.

New Zealand Technology Curriculum: Digital Technologies

Computational Thinking for Digital Technologies:

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).

TR 1 - Workshop Details

Learning Objectives

Students will be able to:

- Break a problem into smaller parts (decomposition).
- Write an algorithm (a set of clear and simple step-by-step instructions to solve a problem).
- Identify when an algorithm produces an unexpected result (a bug).
- Understand that a bug is a normal part of creating an algorithm.
- Understand that debugging is how we find and fix a bug in an algorithm.
- Complete the process of debugging:
 - a) Start at the beginning of the algorithm.
 - b) Follow the algorithm step-by-step until you find a step that produces a result you didn't expect.
 - c) Correct the step.
 - d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.

Materials

Ensure you have the following materials ready for your class:

- Whiteboard
- Marker
- Computer (optional)
- Speakers (optional)
- Paper and pens

Preparation

- Ensure that there is enough space for five groups of students to perform dances without running into tables and chairs or each other.
- Ensure that you are familiar with the definitions of relevant terms (see the glossary provided).

Time Allowance

- 60 minutes

TR 1 - Lesson Plan

Introduction / Learning Hook

1. Perform a dance / play a video of a dance such as the Ka Panapana. Alternatively, students may get up and perform the dance as a group. You may like to choose a traditional dance from a specific culture to extend the activity.

Link to Ka Panapana: https://www.youtube.com/watch?v=an_4rfU8y8k

2. Discuss with the students the cultural significance of dance, and ask them to consider why it might be important to pass on and teach people dances.
3. Tell students that today we're going to learn how to write instructions (an algorithm) to teach other people the moves to specific dances.

Running the Lesson

1. Remind students of the definition of algorithm, decomposition, bugs and debugging.
2. Break the students into five groups.
3. Ask each group to choose a dance that they know the moves for, which they will write an algorithm to describe. (The algorithm and links for the Macarena have been included below as an example).
4. Tell students that they should break the dance into different parts (decomposition) before writing their algorithms.
5. Have each group write down the algorithm (clear and simple step-by-step instructions) that describes how to perform their dances.
6. Once students have written out their algorithms, remind them that debugging is finding and fixing problems in their algorithms.

Get students to go through the process of testing and debugging their dance algorithms:

- a) Start at the beginning of the algorithm, acting out each step of their dance as they go.
 - b) Follow the algorithm step-by-step until they find a step where something happens that they didn't expect.
 - c) Fix the bug.
 - d) Start at the beginning again and repeat these steps until the algorithm does what they were expecting.
7. Once students have debugged their algorithms, ask them to switch algorithms with another group.
 8. The new group should follow the algorithm and act out the dance. They must try to guess what the dance is from the algorithm that they have been given.
 9. **Optional:** Get each group to learn and perform the dances they have been given.
 10. **Optional:** Ask students to discuss the different dances and what they mean.

Conclusion

1. Have a discussion about algorithms, and the importance of giving clear and simple instructions.

You might like to ask:

- What would happen if your dance steps were out of order?
- What happened if your dance steps were unclear?
- Were there any instructions that you found difficult to follow?
- Could you teach a different dance using algorithms?
- Why is using an algorithm better than just showing someone the dance and asking them to do it?
- What are some other times that you could use the process of debugging? (For instance, if you make a mistake in your homework, you could use the debugging process to find and fix the mistake).

Extension Activities

- Have your students research a dance that is common in their own culture or Māori culture. Have students write up the algorithm for that dance and explain the significance and history of the dance.
- Run an activity where students create the algorithm for the order to put your clothes on when getting dressed. This can be done using cultural dress, for example, the kākahu Māori, and can be extended into a research project on traditional outfits for different cultures.
- Have your students create an algorithm for a simple poi performance.

Example to poi performance: <https://www.youtube.com/watch?v=PEcjtEJ3Si0>.

Glossary

Decomposition

Breaking a problem into smaller parts.

Algorithm

A set of clear and simple, step-by-step instructions to solve a problem.

Debugging

A process of how to find and fix a bug in an algorithm.

This process is:

- a) Start at the beginning of the algorithm.
- b) Follow the algorithm step-by-step until you find a step that produces a result you didn't expect.
- c) Correct the step.
- d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.

TR 1 - Example & Relevant Links

Example

Dance Algorithm - The Macarena:

1. Put right arm forward, palm down.
2. Put left arm forward, palm down.
3. Turn right arm over.
4. Turn left arm over.
5. Put right hand on left shoulder.
6. Put left hand on right shoulder.
7. Put right hand behind head.
8. Put left hand behind head.
9. Put right hand on left hip.
10. Put left hand on right hip.
11. Put right hand on bottom.
12. Put left hand on bottom.
13. Sing "Hey, Macarena!"
14. Turn to the left.

Useful Links

Macarena Performance

- <https://www.youtube.com/watch?v=JRRUtp9QOU>

Teacher Resource 2 (TR 2)

Let's Make a Hāngi (Me Mahi Hāngi Tātau)

Description

In this activity, students will practice writing and debugging algorithms, to direct someone to collect vegetables for a hāngi. Students will also discuss what a hāngi is, and its cultural significance.

Curriculum Alignment

New Zealand Technology Curriculum: Digital Technologies

Computational Thinking for Digital Technologies:

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).

Computational Thinking for Digital Technologies:

Progress outcome 2

- They use these algorithms to create simple programs involving outputs and sequencing (putting instructions one after the other) in age-appropriate programming environments.

TR 2 - Workshop Details

Learning Objectives

Students will be able to:

- Recognise the Māori names for core vegetables in the hāngi.
- Break a problem into smaller parts (decomposition).
- Write an algorithm (a set of clear and simple step-by-step instructions to solve a problem).
- Identify when an algorithm produces an unexpected result (a bug).
- Understand that a bug is a normal part of writing algorithms.
- Understand that debugging is how we find and fix a bug in an algorithm.
- Complete the process of debugging:
 - a) Start at the beginning of the algorithm.
 - b) Follow the algorithm step-by-step until you find a step that produces a result you didn't expect.
 - c) Correct the step.
 - d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.

Materials

Ensure you have the following materials ready for each student:

- [] 1x Choose Your Ingredients worksheet
- [] 1-3x Collect the Food worksheets
Note: as discussed below, there are three difficulty levels provided.
- [] 1-3x It's Time to Debug! worksheet(s)
Note: each student will require 1x It's Time to Debug! worksheet for each Collect the Food worksheet they complete.

Preparation

- Print all appropriate worksheets.
- Ensure that you are familiar with the definitions of relevant terms (see the glossary provided).
- You may find it useful to read some background information on hāngi before introducing this lesson to your students. For your convenience, we have provided some links for information on hāngi in the Lesson Resources below.
- **Optional:** Arrange for access to computers connected to the internet for students to play the interactive Let's make a hāngi game (link in Lesson Plan).

Time Allowance

- 30 - 60 minutes

TR 2 - Lesson Plan

Introduction / Learning Hook

1. Ask students what they already know about hāngi. You may like to start with asking them who has had a hāngi before and for what occasion. Some students may have already helped in the preparation of a hāngi.
2. As a group, consider the types of food that are cooked in hāngi. Take this opportunity to teach or remind students the Māori names for hāngi ingredients.
3. Further discussion could include the important Māori cultural beliefs and practices that are associated with food and hāngi.
4. Discuss the process of preparing a hāngi with students, getting students to identify (or explaining) some of the main steps to prepare a hāngi.

Consider:

- The kind of preparation required.
 - The importance of the sequence of events in the preparation of a hāngi.
 - The way that everyone joins in to help with the preparations.
 - The types of occasions where hāngi are prepared.
5. **Optional:** get students to play the interactive computer game 'Let's make a hāngi' to familiarise themselves with the process and steps required to make a hāngi.
 6. Explain that breaking the process of cooking a hāngi into smaller parts is an example of decomposition and remind students of the definition of decomposition.

Running the Lesson

1. Give each student a copy of the Choose Your Ingredients worksheet.
2. Ask students to complete the Choose Your Ingredients activity (drawing a line to match hāngi ingredients with their pictures).
3. When students have completed the activity, go through the answers with them. Ensure they can recognise each ingredient and remind them of the Māori names.
4. Give each student a copy of one or more of the Collect the Food worksheets. There are three difficulty levels provided. These worksheets have been designed to work together or as standalone, so students can either complete all three worksheets sequentially in a lesson, or students can be given the most appropriate worksheets for their learning level.
 - **Collect the Food - Part 1** is the most basic worksheet and allows students to practise creating and debugging algorithms.
 - **Collect the Food - Part 2** is a slightly more difficult worksheet that allows students to consider using different algorithms to solve a problem.
 - **Collect the Food - Part 3** is the most difficult worksheet and allows students to consider different algorithms and their efficiency.
5. Ask students to complete the Collect the Food worksheet(s). In this activity, students must write an algorithm to direct their character around the garden to collect the ingredients for the hāngi. Students should try to create the most efficient algorithms as possible (using the least number of steps to collect all the ingredients).

Note: They are only allowed to give three different instructions: "Forwards", "Turn left", and "Turn right".

The instructions must be written using these symbols:



6. Allow students time to complete the worksheet(s).
7. After students have completed this activity, have students swap their sheet(s) with another student to test and debug each other's algorithms. Each student must exactly follow the algorithm provided to them and determine the path that the algorithm describes. It may help them to draw the path on the worksheet as they follow each step of the algorithm. They must determine whether the algorithm gets them to the final square.
8. If the algorithm does not direct them to the final square, then the student has found a bug! In this case, the two students must go through the process of debugging together to correct the algorithm.

The process of debugging is:

- Start at the beginning of the algorithm.
 - Follow the algorithm step-by-step until you find a step that produces a result you didn't expect.
 - Correct the step.
 - Start at the beginning again and repeat these steps until the algorithm does what you were expecting.
9. Ask students to share how many steps are in their finished algorithms. Find the algorithm with the least steps (the most efficient algorithm).

Conclusion

1. Ask students to discuss what happened when they found a bug in the algorithm.
 2. Have a discussion with the students around efficiency, getting students to identify why it is better to write efficient algorithms.
 3. Tell students that if each square represented ten metres, they might have to walk an extra kilometre or even more if their algorithm wasn't efficient. If they made a hāngi every weekend they'd have to walk lots of unnecessary kilometres every year! Explain that when people write algorithms for computers, they also need to try to make them efficient. Every step in an algorithm takes time, so with more efficient algorithms, computers can do things much more quickly.
 4. Ask students whether it was helpful to have someone else check their algorithm for them, and if it was, why that might be. Discuss the importance of collaboration in getting to the best possible outcome and problem solving.
-

TR 2 - Lesson Resources: Relevant Links

Useful Links	
General information about hāngi	<ul style="list-style-type: none">• https://www.newzealand.com/au/feature/maori-hangi/• http://assessment.tki.org.nz/Assessment-tools-resources/Assessment-tool-selector/Browse-assessment-tools• http://www.maori.cl/Hangi.htm
Images of hāngi	<ul style="list-style-type: none">• https://www.newzealand.com/au/feature/maori-hangi/• http://www.nzhistory.net.nz/media/photo/the-hangi
Instructions on how to prepare a hāngi	<ul style="list-style-type: none">• http://www.nzmaths.co.nz/node/474
Let's make a hāngi: An interactive game on hāngi preparation	<ul style="list-style-type: none">• http://wicked.org.nz/Interactives/Maori-themed-interactives-in-English/Hangi

Student Resources - Glossary






Let's learn some new words!

Glossary	
Algorithm	A set of clear and simple, step-by-step instructions to solve a problem.
A Bug	When an algorithm produces an unexpected result.
Debugging	<p>A process of how to find and fix a bug in an algorithm.</p> <p>This process is:</p> <ol style="list-style-type: none">Start at the beginning of the algorithm.Follow the algorithm step-by-step until you find a step that produces a result you didn't expect.Correct the step.Start at the beginning again and repeat these steps until the algorithm does what you were expecting.



Student Worksheet: Choose Your Ingredients

Match the hāngi ingredients to their pictures.

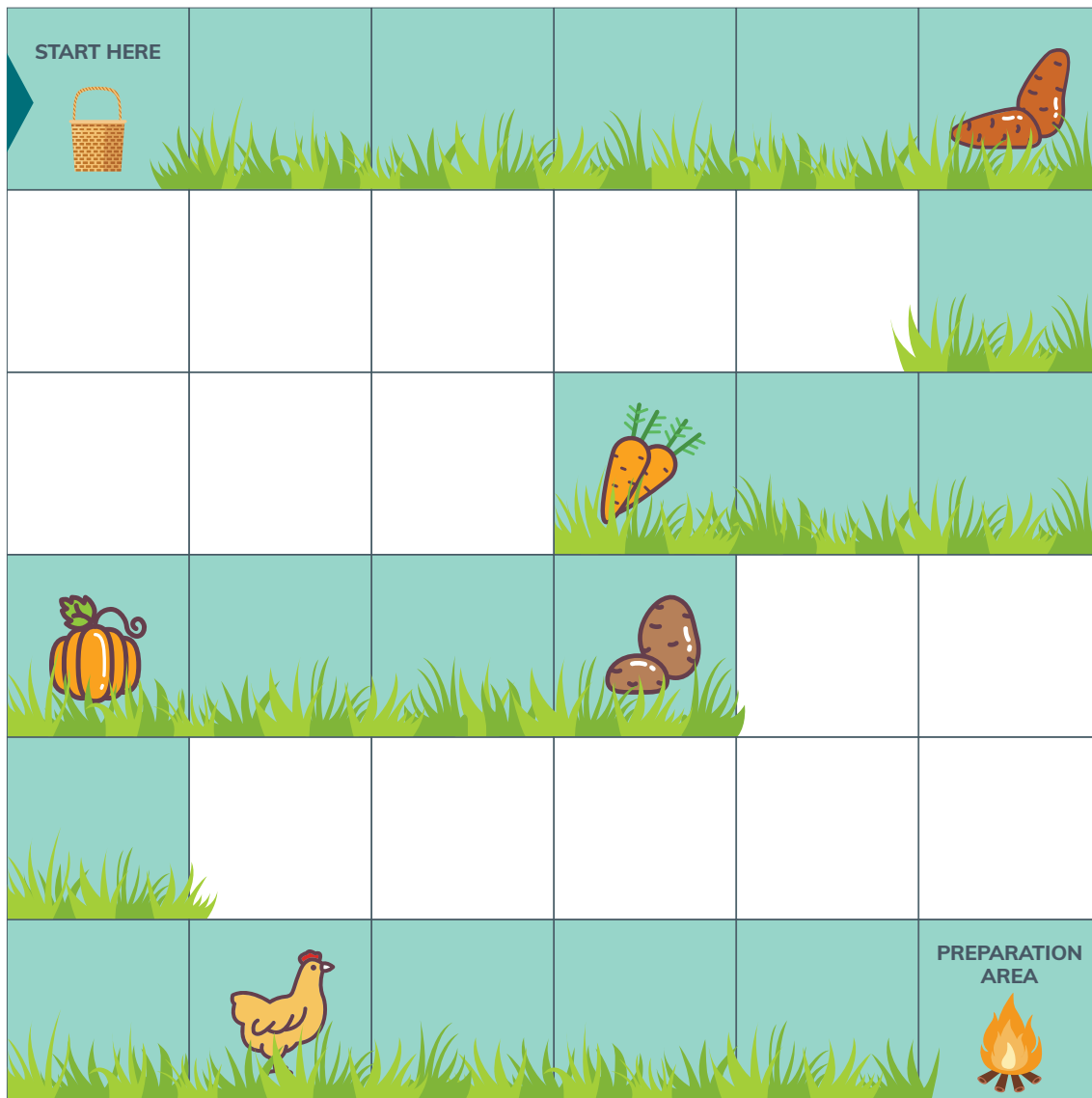
Te heihei (Chicken)	•	•	
Rīwai (Potatoes)	•	•	
Paukena (Pumpkin)	•	•	
Kumara (Sweet Potatoes)	•	•	
Kāreti (Carrots)	•	•	



Student Worksheet: Collect the Food - Part 1

Create an algorithm to get to the preparation area.

Use the symbols on the right.



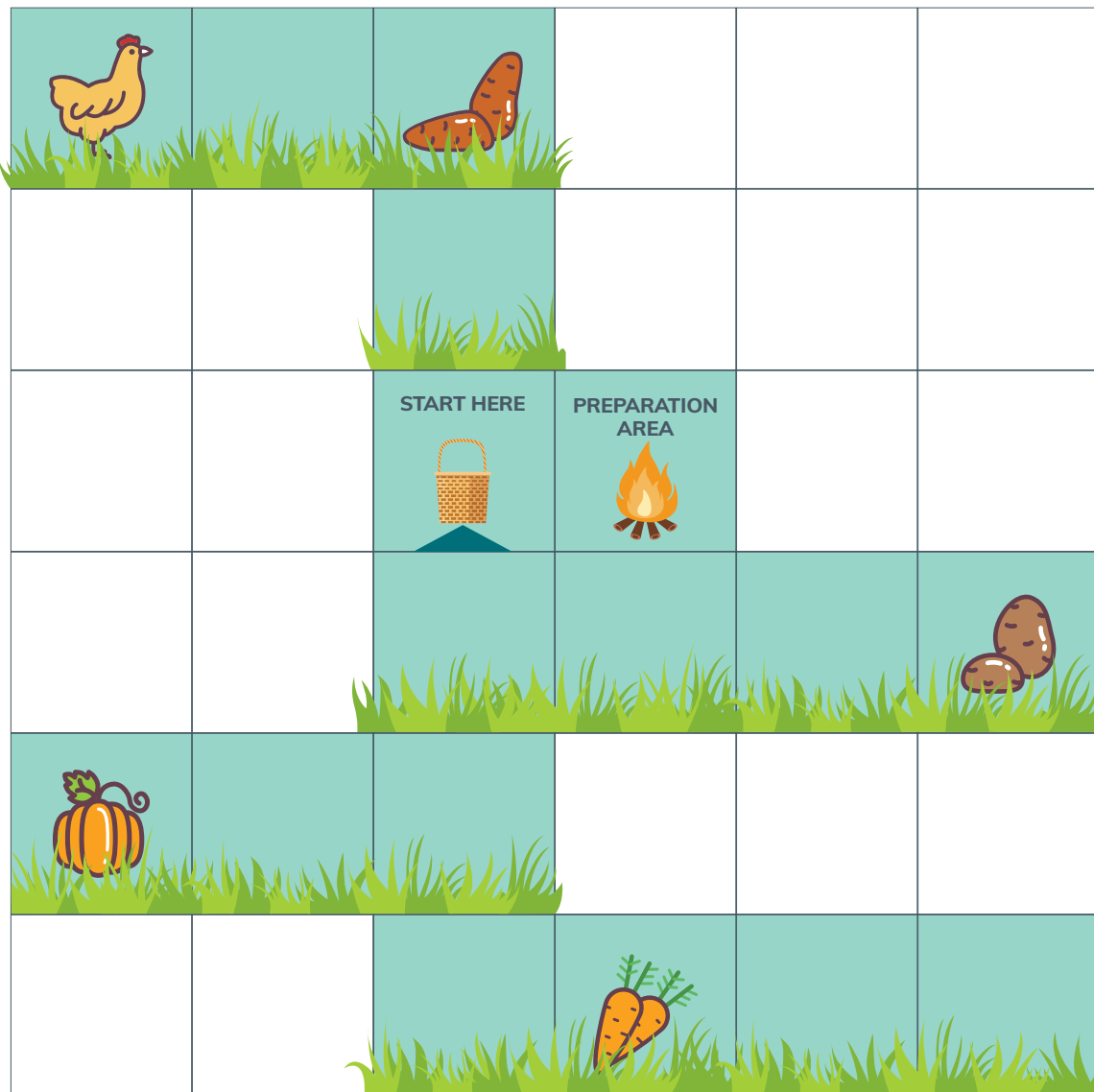
Use the space below to write out your algorithm.

Hint: Turning left or right does not move you to a new square.

Student Worksheet: Collect the Food - Part 2

Create an algorithm to get to the preparation area.

Use the symbols on the right.



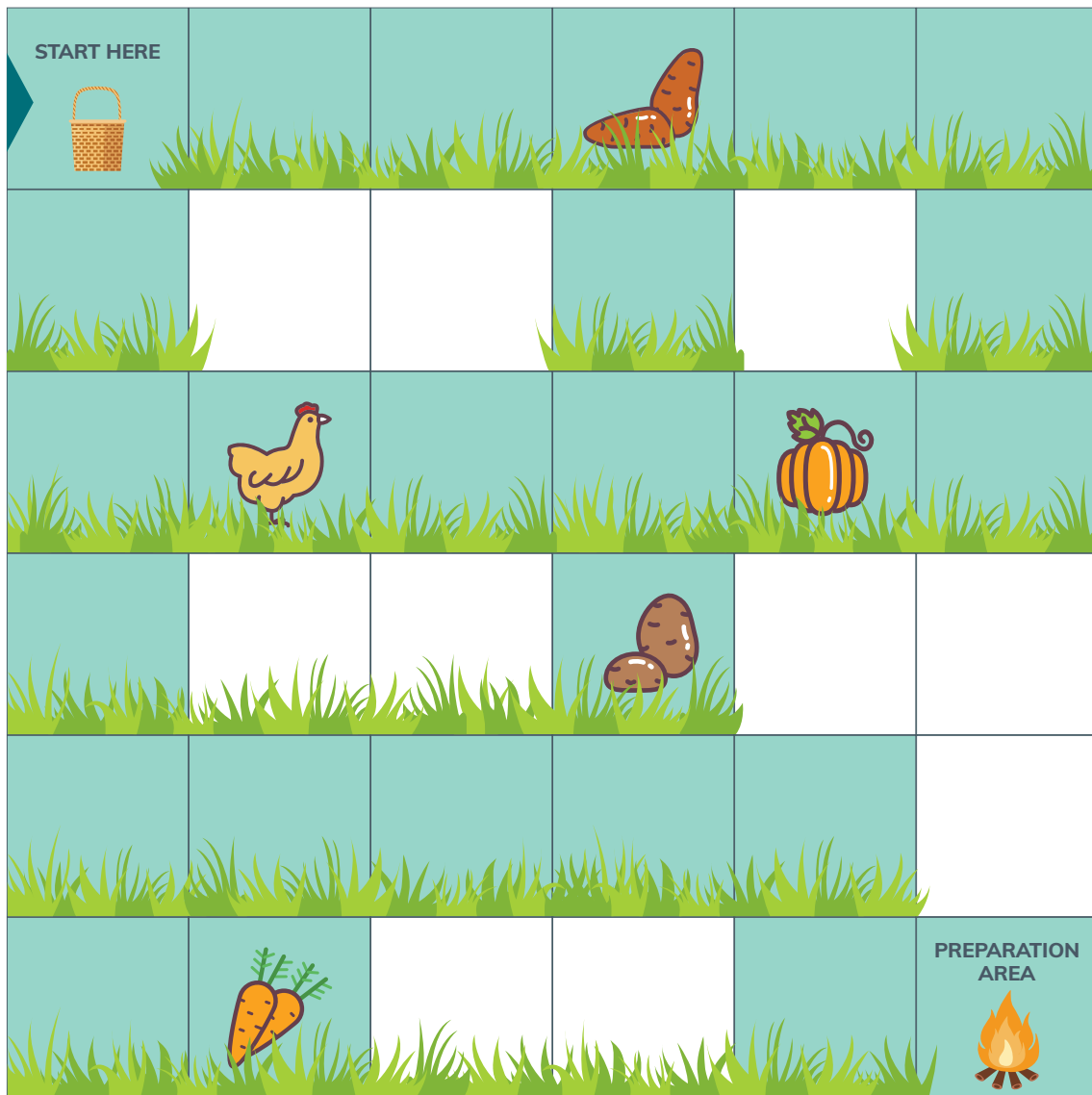
Use the space below to write out your algorithm.

Hint: Turning left or right does not move you to a new square.

Student Worksheet: Collect the Food - Part 3

Create an algorithm to get to the preparation area.

Use the symbols on the right.



Use the space below to write out your algorithm.

Hint: Turning left or right does not move you to a new square.

Student Worksheet: It's Time to Debug!

Rewrite the debugged algorithm below.

Count the steps in your algorithm.



Teacher Resource 3 (TR 3)

Sharing Information with ScratchJr

Description

Students will practise creating programs using ScratchJr that raise awareness of issues within their communities.

Curriculum Alignment

New Zealand Health and Physical Education Curriculum

Health Communities and Environments:

Levels 1 & 2

- Take individual and collective action to contribute to environments that can be enjoyed by all.
- Contribute to and use simple guidelines and practices that promote physically and socially healthy classrooms, schools, and local environments.

New Zealand Technology Curriculum: Digital Technologies

Computational Thinking for Digital Technologies:

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).

Computational Thinking for Digital Technologies:

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.

Computational Thinking for Digital Technologies:

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.

TR 2 - Workshop Details

Learning Objectives

Students will be able to:

- Break a problem into smaller parts (decomposition).
- Write an algorithm (a set of clear and simple step-by-step instructions to solve a problem).
- Understand that computers cannot think and can only follow the instructions that they've been given.
- Identify when an algorithm produces an unexpected result (a bug).
- Understand that a bug is a normal part of humans trying to get computers and digital devices to do what they want.
- Understand that debugging is how we find and fix a bug in an algorithm.
- Complete the process of debugging:
 - a) Start at the beginning of the algorithm.
 - b) Follow the algorithm step-by-step until you find a step that produces a result you didn't expect.
 - c) Correct the step.
 - d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.

Materials

Ensure you have the following materials ready for your class:

- [] 1x iPads/Android tablet with ScratchJr for each student/group of students (Preferred)
- [] 1x Computer with Scratch for each group of three students (Alternative)

Preparation

- Teachers should be familiar with the ScratchJr software before using it in this lesson.
- Information about interface can be found here: <https://www.scratchjr.org/learn/interface>
- Basic tutorials can be found here: <https://www.scratchjr.org/teach/activities>.

Time Allowance

- 45 - 90 minutes

TR 3 - Lesson Plan

Introduction / Learning Hook

1. Show examples of Earth Day Scratch projects from here: <https://scratch.mit.edu/studios/4982024/>
2. Discuss with the students why these people made their Scratch Projects.

Running the Lesson

1. Identify issues within the community that the students would like to raise awareness about, such as putting rubbish in the bin, saving energy, recycling etc.
2. As a research project, ask students to research facts about their issue.
3. If needed, remind students of how to use ScratchJr.
4. Remind students that an algorithm is a set of clear, step-by-step instructions to solve a problem. Computers cannot think and can only follow the steps that they've been given.
5. Arrange students into groups and have them decide on the project that they are going to create.
6. Ask students to plan how they would like to share their message. They might consider the number of characters and what they want the characters to do/say, or the number of pieces of information that they would like to share, and how they will convey each one.
7. Explain to students that by thinking about and planning different elements of their project they have just decomposed their problem into smaller more manageable parts.
8. Decomposition is breaking a problem down into smaller parts.
9. Tell students that we use decomposition to make our problems more manageable. This also makes it easier to program.
10. Give students enough time to create a ScratchJr project.
11. The complexity of the projects that the students create will depend on the time available for this lesson and the learning level of the students. Here are examples of possible short, medium and long projects:
 - Short: A single character uses the "say" block to display a speech bubble with a message.
 - Medium: Use the "move" and "say" blocks to have multiple characters deliver a message.
 - Long: Multiple characters interact using the "envelope" blocks in multiple "scenes".
12. If students find a bug, they should debug their algorithms. The way that they should do this is by running through their algorithms, step-by-step, until they find the bug, fix the bug and then testing again.

Once students have completed their projects, get them to present them or share them with each other.

Note: This activity could also be run with Scratch for more advanced classes, or classes without tablets. Tutorials and lesson plans to familiarise your students with the commands for Scratch can be found here: <https://scratch.mit.edu/help/videos>. All of the functionality that students have learnt to use in the ScratchJr workshop is available in Scratch.

Note: If students are sharing devices, they can reprise the programming roles that were used in the workshop. Roles should be rotated so that all students get a chance to practice all roles. These roles ensure that everyone has a turn at the device, and that students also develop their communication skills. The roles are:

The Brain:

- You are the strategist. You design the program and figure out what goes into the program next.
- You communicate this to your 'Hands'.

The Hands:

- You're the hands of the Brain, you do what the Brain says.
- You are responsible for using the iPad to tell the characters what to do.

The Eyes & Ears (x 2 if needed):

- You listen and watch, making sure the Hands understands what the Brain wants, and is doing as it says.
- You check for mistakes.

Conclusion

1. Ask the students why you asked them to break down their stories into smaller parts. Elicit from the students that they broke down their stories into smaller pieces because it made it easier to program. Breaking a problem down into parts is called decomposition.
2. Ask the students what would have happened if they used their blocks in the wrong order. Elicit from the students that an algorithm is a set of simple, step-by-step instructions to solve a problem. Link the students' responses back to the concept that computers cannot think and can only follow the steps that they've been given. This means that they follow algorithms exactly as written, even if there are mistakes in them.
3. Ask the students what they did when their code did not run in the way that they expected it to. Have a discussion about debugging. Elicit from the students how they approached debugging and emphasise that debugging is a necessary and important part of writing algorithms.

Extension Student Activities

1. Have your students create a different ScratchJr projects that give information to the people looking at the presentations. For example, what to do in an earthquake or how to catch a fish.
2. Have your students use ScratchJr to tell a story that is important to them, their family/friends or their whānau.
3. Give students time to create their own characters using the photo and drawing tools in ScratchJr and to use them in their projects.

Glossary

Decomposition

Breaking a problem into smaller parts.

Algorithm

A set of clear and simple, step-by-step instructions to solve a problem.

Debugging

A process of how to find and fix a bug in an algorithm.

This process is:

- a) Start at the beginning of the algorithm.
- b) Follow the algorithm step-by-step until you find a step that produces a result you didn't expect.
- c) Correct the step.
- d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.

TR 3 - Lesson Resources: Relevant Links

Useful Links	
General information about ScratchJr	<ul style="list-style-type: none">• https://en.scratch-wiki.info/wiki/ScratchJr
ScratchJr Interface	<ul style="list-style-type: none">• https://www.scratchjr.org/learn/interface
ScratchJr basic tutorials	<ul style="list-style-type: none">• https://www.scratchjr.org/teach/activities
Scratch tutorials and lesson plans	<ul style="list-style-type: none">• https://scratch.mit.edu/help/videos/

Additional Resources

Resource & Link	Description
CS Unplugged: Searching Algorithms https://www.csunplugged.org/en/topics/searching-algorithms/	This site contains lessons on sequential and binary search which can be run in a classroom without computers or other digital devices.
Additional Resources https://www.csunplugged.org/en/	CS Unplugged is a website with a wealth of resources, beyond what we've listed above, that you may want to explore with your students.
Code.org: Unplugged https://code.org/curriculum/unplugged	Code.org has compiled a list of lessons that teach the fundamentals of computer science, with or without access to computers. These lessons can be used as a stand-alone course or as complementary lessons for any computer science course.
Courses https://studio.code.org/courses?view=teacher	Code.org offers courses for students in grades K-12 and professional learning for teachers. You can sign up for free and complete the courses at your own pace. It also includes online Teacher Communities so that you can take the journey with like-minded teachers.
For Students https://studio.code.org/projects/public	The projects tab of the Code.org site allows students to write a new game for others to play, or to play games made by students all over the world.
Hour of Code https://ro.code.org/learn	Code.org's Hour of Code initiative provides a number of interactive coding tutorials for students of all ages. These tutorials build interactivity through the use of platforms and characters that students will be familiar with and interested in (such as Minecraft, Star Wars, Moana and Frozen.) Tutorials come in a number of languages, including Te Reo Māori, and some tutorials also deliver cultural content (such as Gamefroot's Mihi Maker).
Disney Moana: Wayfinding with Code https://partners.disney.com/hour-of-code/wayfinding-with-code	Disney has created a Moana-themed, gamified coding tutorial to introduce students to the fundamentals of computer programming. These tutorials can be used as part of an Hour of Code initiative, and are available in over 180 languages, including Samoan Polynesian.

Resource & Link	Description
<p>GameFroot</p> <p>Mihi Maker</p> <p>http://make.gamefroot.com/#/activity/mihi-maker</p>	<p>As part of the Hour of Code, Gamefroot has created an interactive tutorial that both teaches students the fundamentals of coding, but also walks them through the creation of a pēpeha. This tutorial incorporates Te Reo Māori throughout.</p>
<p>Curriculum Examples</p> <p>Computational Thinking</p> <p>http://technology.tki.org.nz/Technology-in-the-NZC/CT-Progress-outcomes-exemplars-and-snapshots</p> <p>Designing and Developing Digital Outcomes</p> <p>http://technology.tki.org.nz/Technology-in-the-NZC/DDDO-Progress-outcomes-exemplars-and-snapshots</p>	<p>The Ministry of Education (Te tāhuhu o te Mātauranga) provides exemplars of work in areas of technology for both computational thinking and designing and developing digital outcomes.</p>
<p>Technology</p> <p>http://assessment.tki.org.nz/Assessment-tools-resources/Assessment-tool-selector/Browse-assessment-tools</p>	<p>The Ministry of Education (Te tāhuhu o te Mātauranga) provides of work in areas of technology at various curriculum. Case studies, teaching snapshots and student showcases available to exemplify current technology teaching are available on the Technology Online website.</p>
<p>BBC</p> <p>Bitesize Computing</p> <p>https://www.bbc.com/education/subjects/z34k7ty</p>	<p>The BBC Bitsize Computing website provides lessons in computers in society, binary and data representation, hardware, software, networks, databases and programming.</p>