GoRobot!

Teacher Resources Year 7/8

Digital Ignition
Māpura Matihiko

Introduction

The Ministry of Education have revised the Technology learning area to strengthen the positioning of Digital Technologies in The New Zealand Curriculum. The goal of this change is to ensure that all learners have the opportunity to become digitally capable individuals. This change signals the need for greater focus on our students building their skills so they can be innovative creators of digital solutions, moving beyond solely being users and consumers of digital technologies.

What is technology about?

Technology is intervention by design. It uses intellectual and practical resources to create technological outcomes, which expand human possibilities by addressing needs and realising opportunities.

Design is characterised by innovation and adaptation and is at the heart of technological practice. It is informed by critical and creative thinking and specific design processes.

Why study technology?

With its focus on design thinking, technology education supports students to be innovative, reflective and critical in designing new models, products, software, systems and tools to benefit people while taking account of their impact on cultural, ethical, environmental and economic conditions.

The aim is for students to develop broad technological knowledge, practices and dispositions that will equip them to participate in society as informed citizens and provide
a platform for technology-related careers.

Strands

In **Technological Practice**, students examine the practice of others and undertake their own.

Students develop **Technological Knowledge** particular to technological enterprises and environments and in relation to how and why things work.

For the **Nature of Technology**, students develop an understanding of technology as a discipline and of how it differs from other disciplines. They learn to critique the impact of technology on societies and the environment and to explore how developments and outcomes are valued by different peoples in different times.

Learning pathways

Over the course of years 1–10, students learn in all five technological areas, developing their knowledge and skills in context. By offering a variety of contexts, teachers help their students to recognise links between technological areas.

Contents

Teacher Resource 1a (TR 1a)
Lets ESCape

Description

In this two-part activity, students will develop and practice the computational skills of algorithmic thinking and debugging as well as create algorithms containing functions, loops and conditionals.

Alignment to the New Zealand Curriculum

Technology Learning Area	
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.
Nature of Technology: Level 2	• Students will understand that technology both reflects and changes society and the environment and increases people's capability. • Students will understand that technological outcomes are developed through technological practice and have related physical and functional natures.
Nature of Technology: Level 3	• Students will understand that technological outcomes are recognisable as fit for purpose by the relationship between their physical and functional natures.

Mathematics and Statistics Learning Area	
Geometry and Measurement: Level 4	• Students will communicate and interpret locations and directions, using compass directions, distances, and grid references.

Social Sciences Learning Area	
Social Studies: Level 1	• Students will understand how the past is important to people. • Students will understand how the cultures of people in New Zealand are expressed in their daily lives.

TR 1a - Lesson Details

Learning Objectives

Students will be able to:

- 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 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 material ready:

[ ] Worksheet – Algorithms and debugging (1 per pair of students)

[ ] Paper and pens

Preparation

- Before the lesson, you will need to print enough copies of Worksheet – Algorithms and debugging for at least one copy per pair of students.
- Ensure that you are familiar with the definitions of relevant terms (see the glossary provided).

Time Allowance

- 15 minutes

TR 1a - Lesson Plan:

Introduction / Learning Hook

1. Ask students if they remember the story of Hatupatu and the bird-woman.
2. Elicit or retell the story. This can be found on page 4.
3. Tell students that they are going to debug an algorithm so Hatupatu can escape from the bird-woman's cave.

Running the Lesson

1. Give each student a copy of Worksheet – Algorithms and debugging.
2. Ask students to complete the Worksheet – Algorithms and debugging. In this activity, students must debug the algorithm to direct their character out of the labyrinth.
3. In pairs, ask the students to run through the process of debugging and circle any incorrect steps in the algorithms. The process of debugging is:
   a) Start at the beginning of the algorithm.
   b) Follow the algorithm step-by-step until you find the step(s) where the algorithm produces a result you didn't expect.
   c) Circle the incorrect steps and write down the correct steps in the space provided.
   d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.
4. After students have completed this activity, have students compare their sheets with a different pair to see if their algorithms align.

Conclusion

1. Ask students to discuss what happened when they found a bug in the algorithm.
2. 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.

Glossary	
Decomposition	Breaking a problem into smaller parts.
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	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(s). d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.

Hatupatu and the bird-woman

When Hatupatu was hunting one day for birds in the forest, he met a woman called Kurangaituku, who was spearing birds for herself. This woman had wings on her arms, and claws instead of fingers. Her lips were long and hard and pointed, like a bird's beak, and she was using them as a spear.

Kurangaituku captured Hatupatu. She took him home to her cave, and kept him prisoner there.

At dawn each day Kurangaituku went out to spear birds, but Hatupatu stayed at home. When she had gone he looked at all the possessions in her cave. There were pet birds and lizards, a taiaha, and piles of precious cloaks. Every day Hatupatu admired these treasures, wishing very much that he could escape and take them with him.

One morning he said to Kurangaituku, ‘When you go hunting today you had better go a long way, and travel over a thousand hills. When you get there, you will find birds for us.’

Kurangaituku agreed to this, and she went.

When Haptupatu thought that she was far enough away, he began to ransack her cave, stealing and destroying her treasures. But one of Kurangaituku’s little pet birds escaped. It flew away to get Kurangaituku. And as the little bird flew along he sang, ‘Kurangaituku, our home is ruined, our things are all destroyed’; he kept singing this and flew on and on.

After a long time, Kurangaituku heard him, and hurried back to her home.

When she got to the cave, no one was there. But the little bird showed her where Hatupatu had gone, and she ran on.

Hatupatu heard her behind him, and he thought, ‘I'm done for now.’ So he repeated a magic charm he knew; ‘O rock, open for me, open.’ Then the rock opened, and he hid inside it.

Kurangaituku came running past the rock, but she could not see him, and she ran on.

After her voice had died away in the distance, Hatupatu came out of the rock and ran on again.

Hatupatu came to the shores of Lake Rotorua. His home was on Mokoia Island in the middle of the lake. He dived in and swam under the water to the island, and there he was united with his parents, who had thought for a long time that he was dead.

The rock that Hatupatu split can still be seen today. It is a wāhi tapu (sacred place) to the Tūwharetoa, Te Arawa, and Raukawa iwi. Passers-by often place an offering of twigs, bracken and sometimes food there.

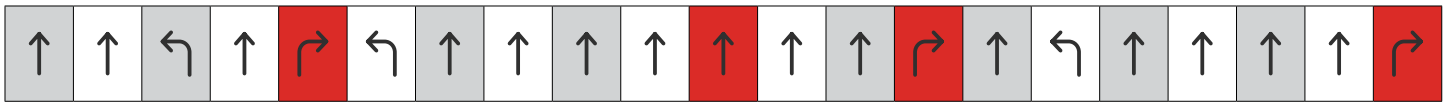
Adapted from:
http://teaohou.natlib.govt.nz/journals/teaohou/issue/Mao53TeA/c16.html

TR 1a - Lesson Resources: Relevant Links & Answers

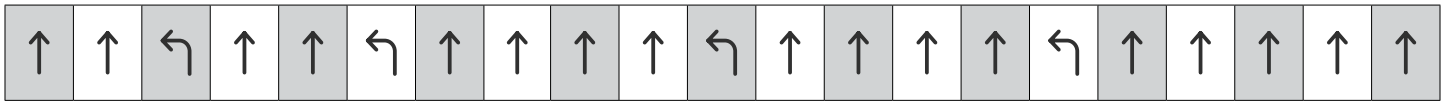
Useful Links	
Hatupatu and the bird-woman	• http://teaohou.natlib.govt.nz/journals/teaohou/issue/Mao53TeA/c16.html

Algorithms and Debugging - Worksheet Answers

Answer 1:



Answer 2:

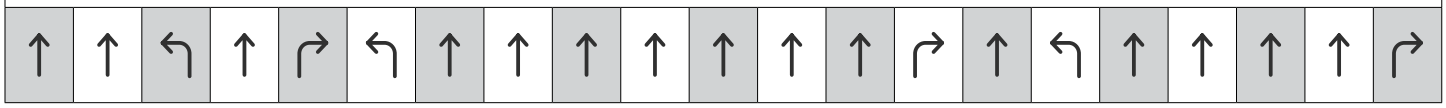


Student Worksheet: Algorithms and debugging

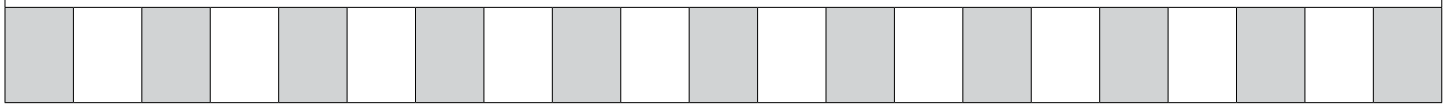
Help Hatupatu escape!

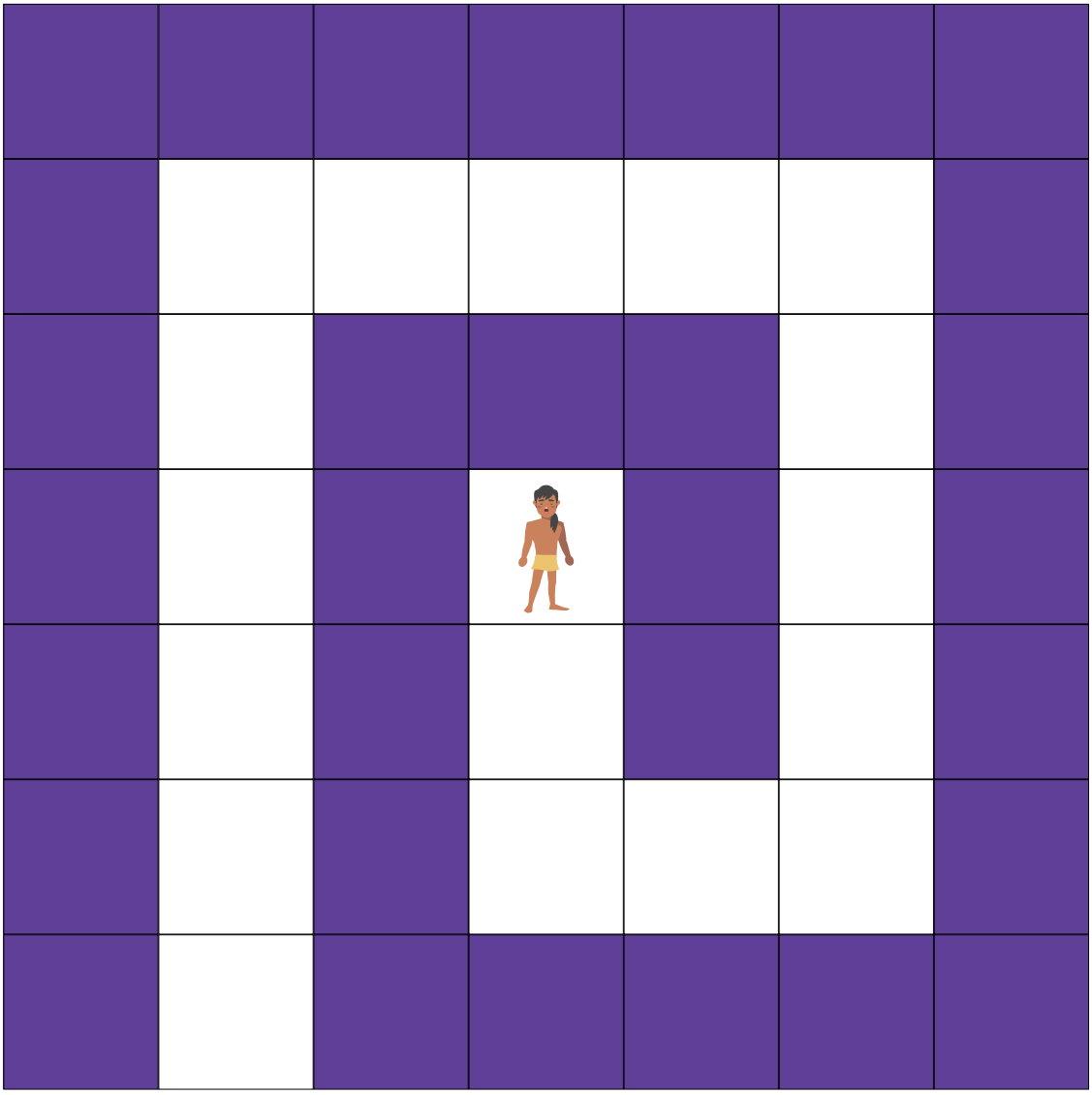
Hatupatu is trying to escape from the bird-woman's cave. We need to debug our algorithm so he can get out.

Circle the bugs in the algorithm below:



Write your correct algorithm below:





Teacher Resource 1b (TR 1b)
Digging into Algorithms

Description

In this two-part activity, students will develop and practice the computational skills of algorithmic thinking and debugging as well as create algorithms containing functions, loops and conditionals.

Alignment to the New Zealand Curriculum

Technology Learning Area	
Computational Thinking for Digital Technologies: 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 programming: 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.
Technological Practice: Level 3	• Students will undertake planning to identify the key stages and resources required to develop an outcome.
Nature of Technology : Level 2	• Students will understand that technology both reflects and changes society and the environment and increases people's capability. • Students will understand that technological outcomes are developed through technological practice and have related physical and functional natures.

Mathematics and Statistics Learning Area	
Geometry and Measurement: Level 4	• Students will communicate and interpret locations and directions, using compass directions, distances, and grid references.

TR 1b - Lesson Details

Learning Objectives

Students will be able to:

- 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 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.
- Understand that some algorithms are more efficient than others.
- Understand that conditionals let us run different code in different situations.
- Understand that loops let us repeat code again and again.

Materials

Ensure you have the following material ready:

[ ] Worksheet – Conditionals and loops (1 per pair of students)

[ ] Pens

[ ] Scissors

[ ] Glue

Time Allowance

- 15 minutes

Preparation

- Before the lesson, you will need to print enough copies of Worksheet – Conditionals and loops (1 & 2) for at least one copy per pair of students.
- Ensure that you are familiar with the definitions of relevant terms (see the glossary provided).
- Ensure access to visual audio tech to play the Japanese Hole Digging Contest video.

TR 1b - Lesson Plan

Introduction / Learning Hook

1. Show the Japanese Hole-Digging Contest video: https://www.youtube.com/watch?v=peVnXqa5n2o.
2. Tell students that while a few people like digging holes for fun, some people dig holes to find things.
3. Ask students what they already know about kāpia (kauri gum). Explain the history and significance of kāpia and Māori uses of kāpia.
   - Kāpia comes from kauri trees.
   - Kāpia is a golden gum that people use for jewellery and decorations.
   - In the 19th century kauri gum became a major export. Lots of people went to try and dig it up, but digging gum was a really hard way to make money.
   - New Zealand soldiers used the term 'Diggers' (from the gold and gum diggers) towards the end of World War I.
   - Māori used it to help their fires burn, as fuel for torches and they also chewed it.
4. Explain that digging up kāpia is really difficult and if we had an algorithm for a robot to do it, it would be much easier.

Running the Lesson

1. Put the students into pairs.
2. Give each pair a copy of the Worksheet – Conditionals and loops (1).
3. Tell students the words that they can use to direct Ririwai, and what each word does:
   (Move forward - Ririwai moves forward one square. Dig - Ririwai digs up the kāpia gum.)
4. Ask students to complete the worksheet by writing an algorithm for Ririwai to dig up all the kāpia.
5. After students have completed their algorithms, they must follow the algorithm they have written and ensure the algorithm gets all the kāpia and gets Ririwai to the final square.
6. If the algorithm does not achieve those outcomes, then the students have 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:
   a) Start at the beginning of the algorithm.
   b) Follow the algorithm step-by-step until you find the step/s where the algorithm produces a result you didn't expect.
   c) Correct the step(s).
   d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.
7. Ask the students to count the number of steps in their algorithms and write the number on their worksheets.

   Remind students that the conditionals block (IF/THEN/ELSE) let us run different code in different situations and that the loop block lets us repeat code again and again. These can help us write more efficient algorithms.
8. Give each pair of students a copy of the Worksheet – Conditionals and loops (2).
9. Ask students to cut out the blocks on the worksheet.
10. Ask students to create an efficient algorithm with the blocks, to solve the problem of Ririwai digging up all the kāpia. Students should try to create the most efficient algorithms possible (using the least number of blocks to collect all the kāpia).

11. After students have used the blocks to make their code, they should glue their new, more efficient algorithms next to their original algorithms, and write the number of blocks they used to create that algorithm (see below for the solution to the algorithm.)

12. After students have completed their algorithms, they must follow the algorithm the have written and ensure that the algorithm gets all the kāpia and gets Ririwai to the final square.

13. If the algorithm does not direct them to the square, then the students have 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:
   a) Start at the beginning of the algorithm.
   b) Follow the algorithm step-by-step until you find the step/s where the algorithm produces a result you did not expect.
   c) Correct the step(s).
   d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.

14. Ask students to count the number of blocks in their debugged algorithm.

15. Ask students to compare the number of blocks in their new algorithm with the number of steps they wrote in the previous activity.

Conclusion

1. Have a discussion with the students around efficiency, have students identify why it is better to write efficient algorithms. By comparing the two algorithms in the activity, it should be clear that writing more efficient algorithms require less coding and make code easier to read. Having loops allows programmers to repeat code without having to rewrite it and conditionals allow programs to execute code based on different conditions.

2. Ask students if they found any bugs in the algorithm. If they did, ask them why the bug occurred and how they found it (by going through the process of debugging).

3. As a group, work out how many blocks they would need to get all the kāpia if they could not use the conditional block, but could not use the loop block? What if they couldn't use the conditional block, but could use the loop block? (It would take 19 blocks if they could not use the loop block, and 26 if they couldn't use the conditional block, though an error may occur.) What if the swamp Ririwai was in 100 blocks long? Explain that when algorithms are written for computers, they also need to make them efficient. Every step in an algorithm takes time, so with more efficient algorithms, computers can do things much more quickly.

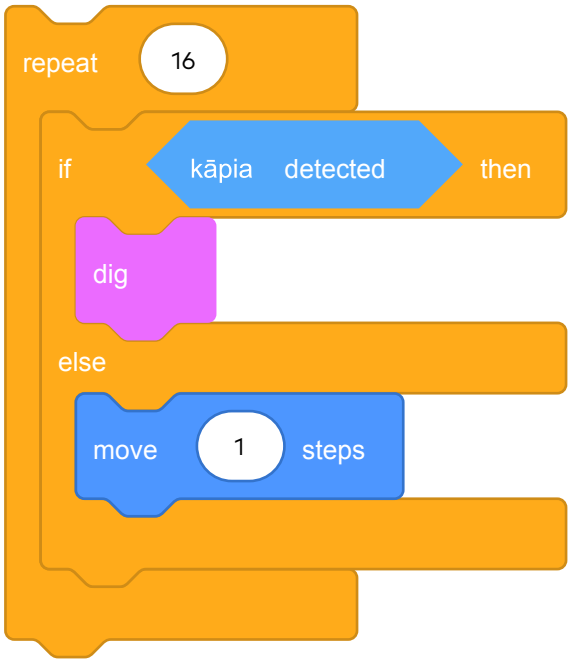
4. Ask students whether it was helpful to have someone else check their algorithm for them, and if it was, why that might be. Discuss the importance of collaboration in getting to the best possible outcome and problem solving.

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	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 did not expect. c) Correct the step(s). d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.
Conditionals	Lets us run different code in different situations.
Function	A group of instructions that has been given a name.
Loops	Let's repeat code again and again

TR 1b - Lesson Resources: Relevant Links & Answers

Useful Links	
Japanese Hole-Digging Contest	• https://www.youtube.com/watch?v=peVnXqa5n2o/

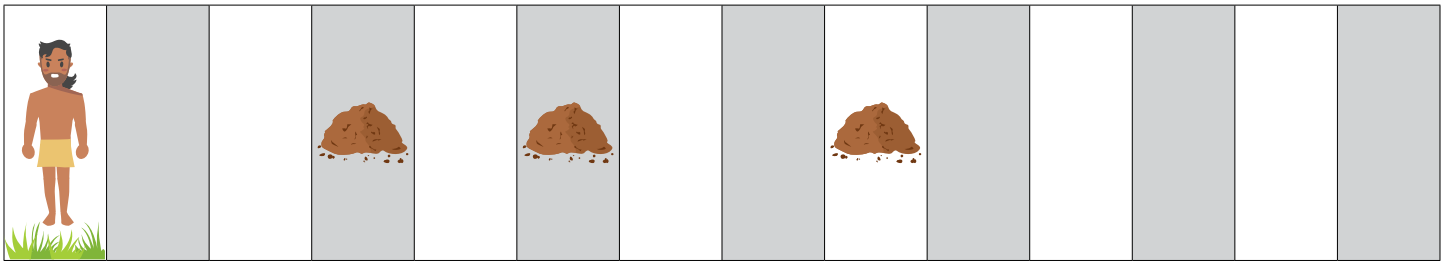
Conditionals and Loops - Worksheet Answers

Write your algorithm below:	Paste your blocks below:
1. Move Forward **2.** Move Forward **3.** Move Forward **4.** Dig **5.** Move Forward **6.** Move Forward **7.** Dig **8.** Move Forward **9.** Move Forward **10.** Move Forward **11.** Move Forward **12.** Dig **13.** Move Forward **14.** Move Forward **15.** Move Forward **16.** Move Forward	
Number of steps: 16	**Number of blocks: 5**

Student Worksheet: Conditionals and loops

Help Ririwai dig up kāpia (kauri gum)!

Note*: Don't forget to debug your code when you've finished.

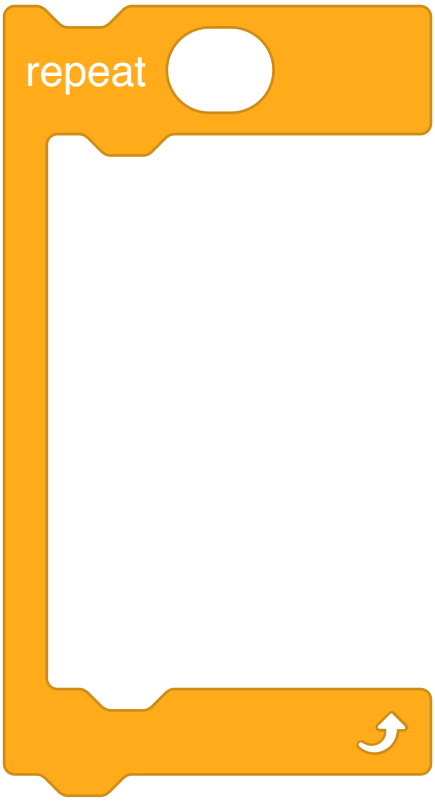


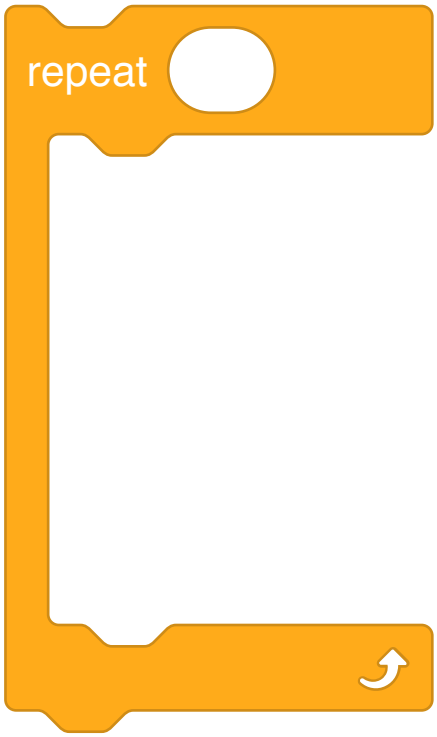
Write your algorithm below:	Paste your blocks below:
Number of steps:	**Number of blocks:**

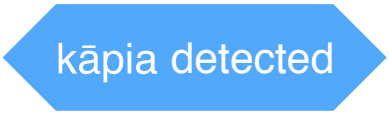
Student Worksheet: Conditionals and loops

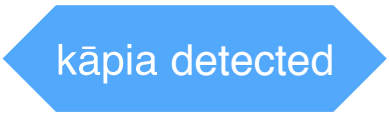
Cut the blocks out below to workout your algorithm, once you have workout your algorithm paste your answer in the right column on the worksheet.

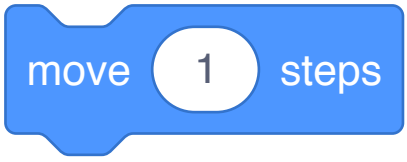
Note*: You may not need all the blocks for your algorithm.

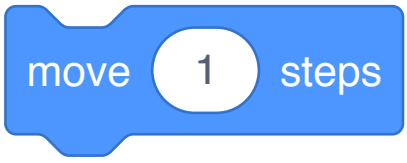


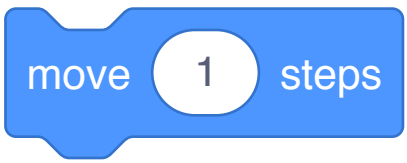








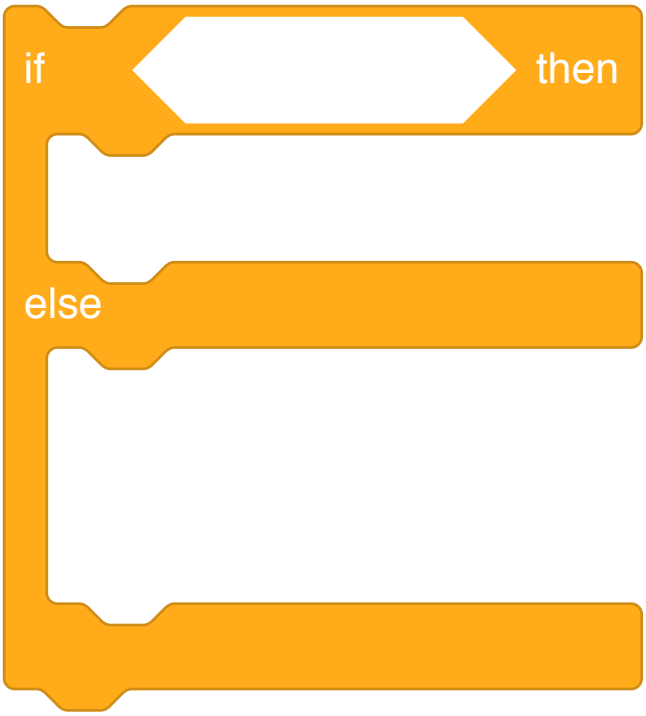


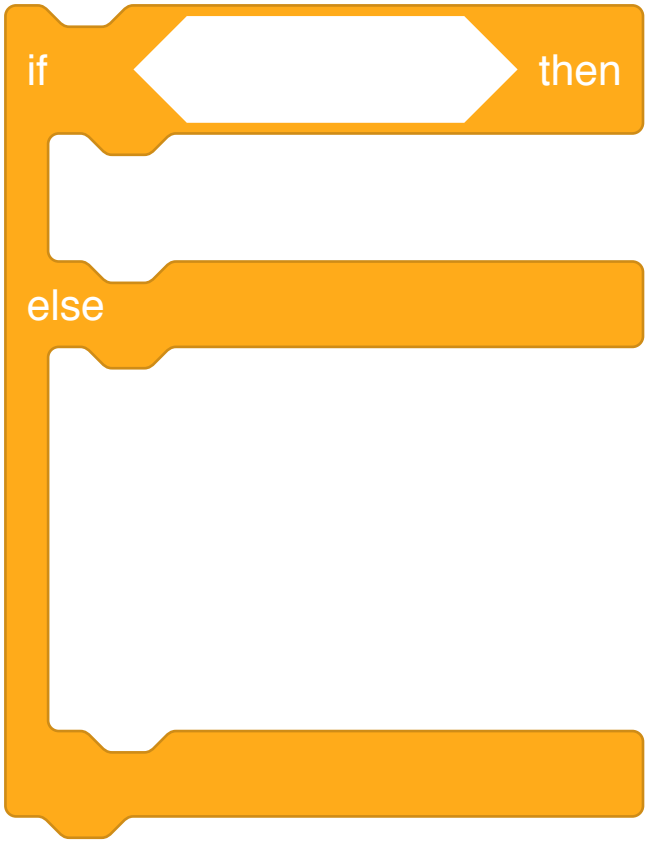


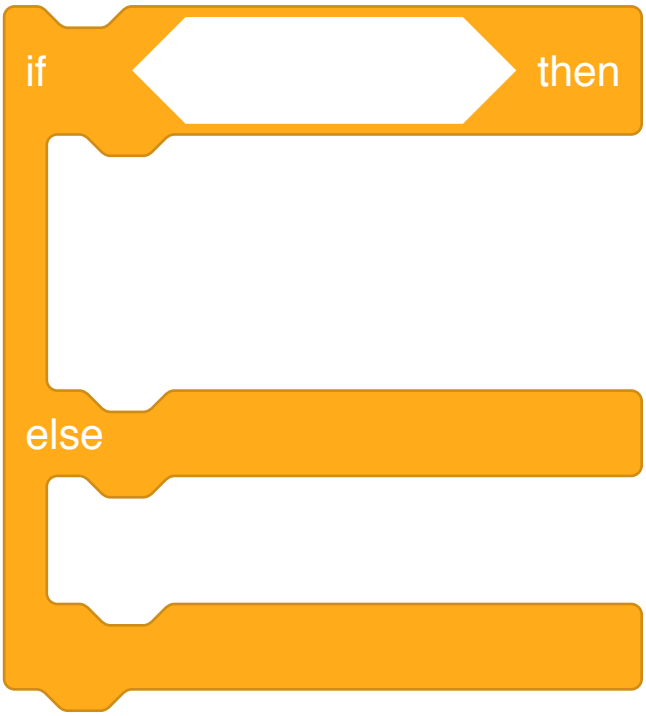
Student Worksheet: Conditionals and loops

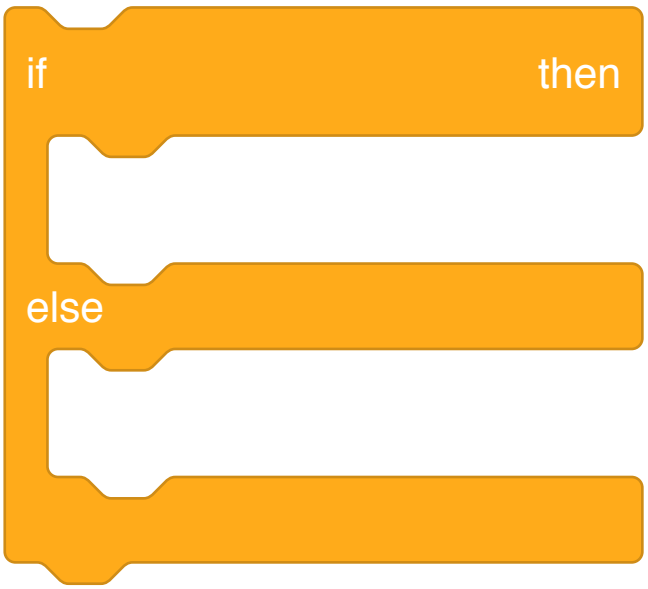
Cut the blocks out below to workout your algorithm, once you have workout your algorithm paste your answer in the right column on the worksheet.

Note*: You may not need all the blocks for your algorithm.









Teacher Resource 2 (TR 2)
Escape the Cave

Description

In this activity, students create a series of algorithms that include conditionals and loops and functions to help Hatupatu escape from the bird-woman's cave.

Alignment to the New Zealand Curriculum

Technology Learning Area	
Computational Thinking for Digital Technologies: 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 programming: 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.
Technological Practice: Level 3	• Students will undertake planning to identify the key stages and resources required to develop an outcome.
Nature of Technology : Level 2	• Students will understand that technological outcomes are developed through technological practice and have related physical and functional natures.

Social Sciences Learning Area	
Social Studies: Level 1	• Students will gain knowledge, skills, and experience to understand how the past is important to people. • Students will gain knowledge, skills, and experience to understand how the cultures of people in New Zealand are expressed in their daily lives.

Mathematics and Statistics Learning Area	
Geometry and Measurement: Level 4	• Students will communicate and interpret locations and directions, using compass directions, distances, and grid references.

TR 2 - Lesson Details

Learning Objectives

Students will be able to:

- 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 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.
- Understand that some algorithms are more efficient than others.
- Understand that conditionals let us run different code in different situations.
- Understand that a function is a group of instructions that has been given a name.
- Understand that loops let us repeat code again and again.

Materials

Ensure you have the following material ready:

[ ] Worksheet – Escape from the cave (1 per pair of students)

[ ] Paper

[ ] Pens

Time Allowance

- 60 minutes

Preparation

- This activity assumes that students have already completed the two activities - Year 7 Teacher Resources 1a and 1b and/or they are comfortable with block programming.
- Before the lesson, you will need to print enough copies of Worksheet – Escape from the cave for at least one copy per pair of students.
- Ensure that you are familiar with the definitions of relevant terms (see the glossary provided)

TR 2 - Lesson Plan

Introduction / Learning Hook

1. Ask students if they remember the story of Hatupatu and the bird-woman.
2. Elicit the story of Hatupatu and the bird-woman. The story can be found on page 16.
3. Tell students that they are going to create a number of algorithms so Hatupatu can escape from the bird-woman's cave.

Running the Lesson

1. Give each student a copy of Worksheet – Escape from the cave.
2. Remind students that inputs are the signals or data received by a digital device, and point out the input blocks on the worksheet. Explain to students that the input blocks work like the sensors on the mBot or the eyes of Hatupatu – they allow him to react when he 'sees' or detects certain things.
3. Highlight the different blocks that they can use.
   - Move forward
   - Turn left
   - Turn right
   - Pick up (Note: this picks up only one object at a time)
   - Attack
4. Have students read the runForward() function on the worksheet. See if they can determine what this function will achieve (move Hatupatu forward three squares).
5. Have students work out what the wholeLine() and runDiagonal() functions do.
6. As a class, discuss the wholeLine() and runDiagonal() functions, stressing that the wholeLine() function does not pick up everything thing in a line, but only one item from every square. If there are multiple items on one square, the wholeLine() function will only pick up one of them.
7. Ask the students why functions are important, then tell them:
   - Functions make code more manageable
   - Functions reduce the amount that needs to be written
   - Functions make code easier to understand.
8. Remind the students that an important and necessary part of creating algorithms is debugging.
9. Run through the process of debugging with the students. The process of debugging is:
   a) Start at the beginning of the algorithm.
   b) Follow the algorithm step-by-step until you find the step/s where the algorithm produces a result you didn't expect.
   c) Circle the incorrect steps and write down the correct steps in the space provided.
   d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.

10. Ask students to complete the Worksheet – Escape from the cave. In this activity, students will need to write different algorithms to achieve five different outcomes:

    - Visit every square on the map
    - Get the most gold
    - Attack all the lizards
    - Get out of the cave the fastest
    - Achieve any outcome that they like

    Note: When students use the 'pickUp' block, they will only pick up one item at a time, when students use the 'attack' block, they will attack only one lizard at a time.

11. Give students enough time to write all five algorithms and to count the number of steps in each algorithm.
12. After students have completed this activity as a class, test and debug some of the students' algorithms. The algorithms should be followed exactly to determine the path that the algorithm describes. It may help to have students draw the path on the worksheet in a different colour and tally up how many of each item they collected, as they follow each step of the algorithm.
13. Ask students to compare their algorithms, seeing which ones took the least number of steps.

Conclusion

1. Ask students to discuss what happened when they found a bug in the algorithm.
2. Ask students whether it was helpful to have someone else check their algorithm for them, and if it was, why that might be.
3. Discuss the importance of collaboration in getting to the best possible outcome and problem solving.
4. Ask the students if their algorithms were the same or different from other students'.
5. Ask students how they could determine which algorithms were more efficiently written.
   (By counting the number of steps in the algorithm).
6. Ask the students when they used conditionals, and how they were useful.
7. Ask the students why they used functions in their algorithms.
8. Ask the students what functions they created and why they created them.

Glossary	
Decomposition	Breaking a problem into smaller parts.
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	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(s). d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.
Conditionals	Lets us run different code in different situations.
Function	A group of instructions that has been given a name.
Loops	Let's repeat code again and again

Hatupatu and the bird-woman

When Hatupatu was hunting one day for birds in the forest, he met a woman called Kurangaituku, who was spearing birds for herself. This woman had wings on her arms, and claws instead of fingers. Her lips were long and hard and pointed, like a bird's beak, and she was using them as a spear.

Kurangaituku captured Hatupatu. She took him home to her cave, and kept him prisoner there.

At dawn each day Kurangaituku went out to spear birds, but Hatupatu stayed at home. When she had gone he looked at all the possessions in her cave. There were pet birds and lizards, a taiaha, and piles of precious cloaks. Every day Hatupatu admired these treasures, wishing very much that he could escape and take them with him.

One morning he said to Kurangaituku, 'When you go hunting today you had better go a long way, and travel over a thousand hills. When you get there, you will find birds for us.'

Kurangaituku agreed to this, and she went.

When Haptupatu thought that she was far enough away, he began to ransack her cave, stealing and destroying her treasures. But one of Kurangaituku's little pet birds escaped. It flew away to get Kurangaituku. And as the little bird flew along he sang, 'Kurangaituku, our home is ruined, our things are all destroyed'; he kept singing this and flew on and on.

After a long time, Kurangaituku heard him, and hurried back to her home.

When she got to the cave, no one was there. But the little bird showed her where Hatupatu had gone, and she ran on.

Hatupatu heard her behind him, and he thought, 'I'm done for now.' So he repeated a magic charm he knew; 'O rock, open for me, open.' Then the rock opened, and he hid inside it.

Kurangaituku came running past the rock, but she could not see him, and she ran on.

After her voice had died away in the distance, Hatupatu came out of the rock and ran on again.

Hatupatu came to the shores of Lake Rotorua. His home was on Mokoia Island in the middle of the lake. He dived in and swam under the water to the island, and there he was united with his parents, who had thought for a long time that he was dead.

The rock that Hatupatu split can still be seen today. It is a wāhi tapu (sacred place) to the Tūwharetoa, Te Arawa, and Raukawa iwi. Passers-by often place an offering of twigs, bracken and sometimes food there.

Adapted from:
http://teaohou.natlib.govt.nz/journals/teaohou/issue/Mao53TeA/c16.html

TR 1b - Lesson Resources: Relevant Links & Answers

Useful Links

Hatupatu and the bird-woman	• http://teaohou.natlib.govt.nz/journals/teaohou/issue/Mao53TeA/c16.html

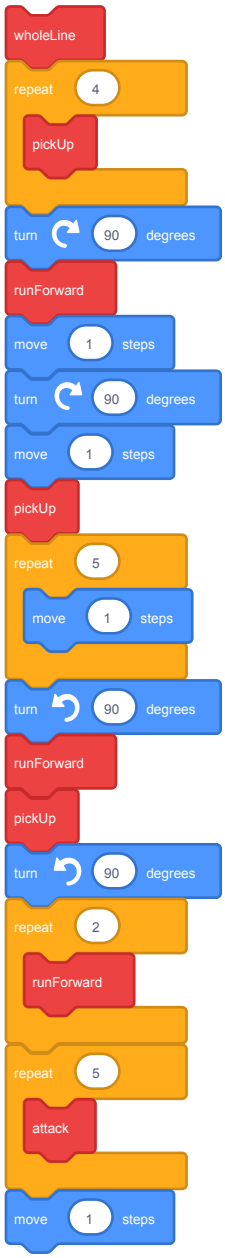
Escape from the cave - Worksheet Answers

1. Visit every square on the map.

Note: Students may not remember that they need to pickup the taiaha. Students should be reminded of this when they come to the debugging stage.

Number of blocks: 17

2. Get the most gold!



Number of blocks: 20

3. Attack the lizards!

Note: Students may not remember that they need to pickup the taiaha. Students should be reminded of this when they come to the debugging stage.

Number of blocks: 18

4. Get out of the cave the fastest!

Number of blocks: 10

Student Worksheet: Escape from the cave (1 of 8)

Help Hatupatu escape!

Write algorithms to help Hatupatu escape from the cave.

Note: A number on a square represents the number of objects on that square. For example, there are five lizards guarding the cave's exit.

Student Worksheet : Escape from the cave (2 of 8)

Rules

1. You need the taiaha before you can pass the lizards.
2. You have to defeat all the lizards on the square before you can pass.
3. Your algorithms have to get you to the exit.

Goals

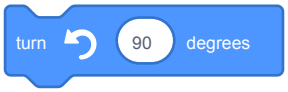
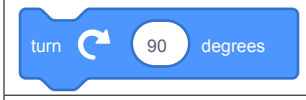
Using the blocks, write a different, efficient algorithm for each of the following on the worksheets

1. Visit every square on the map
2. Get the most gold
3. Attack all the lizards
4. Get out of the cave the fastest
5. Achieve any outcome you like

Blocks & Functions

Hatapaku has the following **input** blocks:

You may use the following **blocks**.

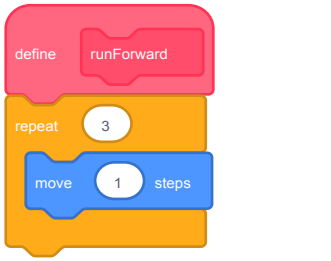
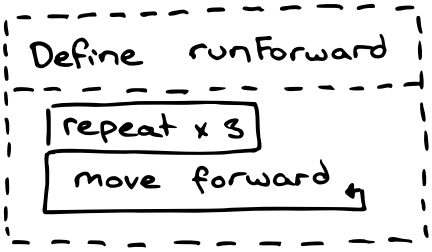
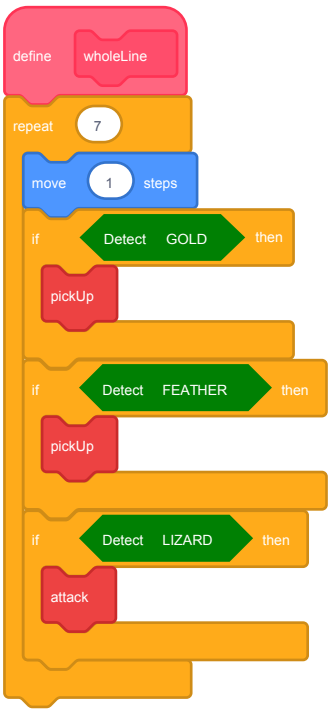
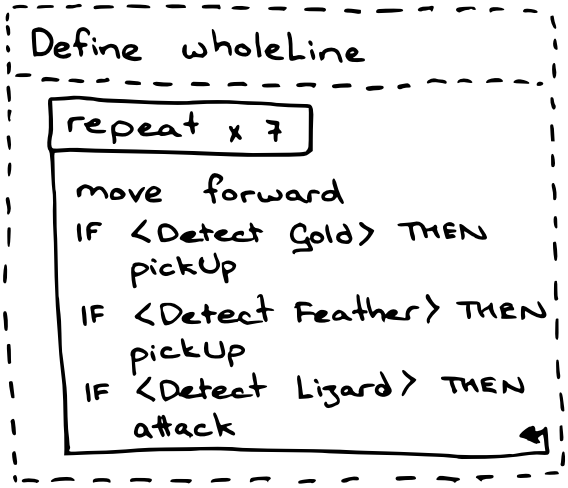
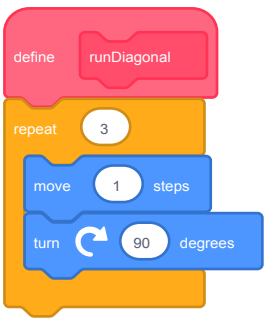
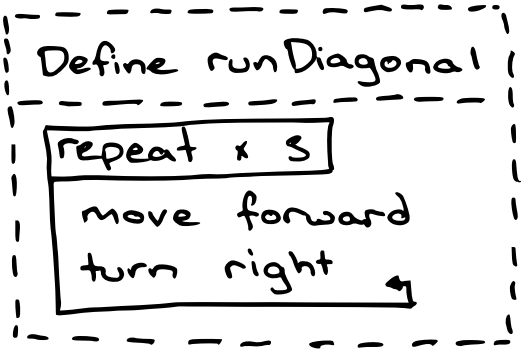
	Moves Hatupatu forward one square
	Turns Hatupatu to the left
	Turns Hatupatu to the right
	Hatupatu picks up one item
	Hatupatu attacks one lizard

Student Worksheet : Escape from the cave (3 of 8)

You may use the following **functions**.

Funtions & Examples

You can also define and name your own **functions**.

What they look like in Scratch:	Example of how to write them:
	
	
	

Student Worksheet : Escape from the cave (4 of 8)

1. Visit every square on the map.

Number of blocks:

Student Worksheet : Escape from the cave (5 of 8)

2. Get the most gold!

Number of blocks:

Student Worksheet : Escape from the cave (6 of 8)

3. Attack the lizards!

Number of blocks:

Student Worksheet : Escape from the cave (7 of 8)

4. Get out of the cave the fastest!

Number of blocks:

Student Worksheet : Escape from the cave (8 of 8)

5. Achieve any outcome you like.

Number of blocks:

Teacher Resource 3 (TR 3)
Functional Bracelets

Description

In this activity, students will create a “program” based on a series of functions while creating a bracelet for themselves and others.

Alignment to the New Zealand Curriculum

Technology Learning Area	
Computation 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.

The Arts Learning Area	
Visual Arts: Level 1	• Explore a variety of materials and tools and discover elements and selected principles.
Dance: Level 4	• Students will prepare and share dance movement individually and in pairs or groups

TR 3 - Lesson Details

Learning Objectives

Students will be able to:

- Understand that an algorithm is a clear, step-by-step set of instructions to solve a problem.
- Understand that a function is a group of instructions that have been given a name.
- Understand that functions make code more manageable and easier to read.

Materials

Ensure you have the following material ready:

[ ] Pieces of elastic bead cord (two pieces per student with spares)

[ ] Plastic beads

[ ] Spacer beads (longer beads)

[ ] One "special" bead per student

[ ] Masking tape

[ ] One premade bracelet

[ ] Worksheet – Functions (one per student)

[ ] Pens and paper

Preparation

- Before the lesson, you will need to print enough copies of Worksheet – Functions for every student.
- Ensure that you are familiar with the definitions of relevant terms (see the glossary provided).
- Prepare a bracelet using the materials that you can show to the students as an example.

Time Allowance

- 40 minutes

TR 3 - Lesson Plan:

Introduction / Learning Hook

1. Let students know that they are about to have a dance off competition. They will all get one opportunity to watch the dance and then they will have to repeat it immediately.
2. Show students the Floss Dance video once: https://www.youtube.com/watch?v=HPJM6ww5hNU
3. Ask students to stand up and dance the floss dance. You can do this as a group or take turns individually.
4. Ask the students if they think they would have done better had they been taught the dance step by step.
5. Remind students that a set of clear and simple, step-by-step instructions is called an algorithm and that they can write algorithms to teach anything. Ask students what a function is and how they could be used to teach this dance.
   (Function: A group of instructions that has been given a name so that they can be called on again.)
6. Break the dance down as a class to elicit the functions that could be used to teach it. Below is an example of possible functions, however, your class might come up with different functions:

   - Centre Start:

   a) Move your hip from neutral standing centre to the right.
   b) Keeping your arms parallel, simultaneously (with step 1) move your left and right hand to the left side of your body.
   c) Move your hip from the right to the left.
   d) Keeping your arms parallel, simultaneously (with step 3) move your left and right hand to the right side of your body.
   e) Repeat steps 1-4 until you've established a smooth rhythm.

   - Right Hip:

   a) Move your hip to the right side of neutral standing centre.
   b) Keeping your arms parallel, simultaneously (with step 1) move your left and right hand to the left side of your body.
   c) Move your hip to the left side of your body.
   d) Keeping your arms parallel, simultaneously (with step 3) move your left hand to the left side in front of your body, and your right hand to the right side behind your body.
   e) Move your hip to the right side of your body.
   f) Keeping your arms parallel, simultaneously (with step 5) move your left and right hand to the left side of your body.
   g) Move your arms and hips back to neutral.

- Right Hip:

a) Move your hip to the left side of neutral standing centre.

b) Keeping your arms parallel, simultaneously (with step 1) move your left and right hand to the right side of your body.

c) Move your hip to the right side of your body.

d) Keeping your arms parallel, simultaneously (with step 3) move your left hand to the left side in front of your body, and your right hand to the left side behind your body.

e) Move your hip to the left side of your body.

f) Keeping your arms parallel, simultaneously (with step 5) move your left and right hand to the right side of your body.

g) Move your arms and hips back to neutral.

7. Once you have written the functions, get everyone up to try to do the dance now that they know the instructions and go through the algorithm function by function rather than step by step.

8. Have a class discussion about the value of using functions.

9. Let students know that in the next activity they are going to be continuing to practice using functions but this time they will use them to write instructions to make jewellery.

Running the Lesson

1. Show students the premade bracelet.
2. Tell students that today they are going to make bracelets for themselves and for someone special to them.
3. Remind the students that an algorithm is a clear, step-by-step set of instructions to solve a problem.
4. Elicit the algorithm for putting a bead on the cord. E.g. 1) Move your right hand to the beads. 2) Use your middle finger and thumb to pinch a bead. 3) Take it out of the bowl. 4) Pick up the cord in your left hand. 5) Thread the bead on to the cord. 6) Let go of the bead with your right hand. Etc.
5. Tell students that they will have to "run" the algorithm for threading their beads on to their cords many times. Instead of writing down all the steps every time, you could write them down once (define the function) and then give the function a name. Then every time they wanted to show they were threading a bead, they could just use the name of the function.
6. Ask students what other functions they might need to make a bracelet, for instance, tying a knot to create a space between beads, putting a bead on in a different way (thread string through bead twice, tie knot in middle), putting on different types of beads.
7. Tell students that they are going to make their bracelets and write down the names of the different functions they needed for their "program" to make their bracelet.
8. Ensure that students understand they need to write down their program, because their partner will also need to follow their program to make a second bracelet.
9. Give each student a piece of elastic bead cord.
10. Use the masking tape to stick one end of the cord to the student's desk (this way the beads won't fall off when they thread them on the cord).
11. Tell the students that they will also receive one special bead that is different from the other beads. If the special bead has cultural significance, be sure to discuss its importance with the students.
12. Give students time to create their bracelets and write down the functions they used.
13. When all the students have finished, in pairs, get them to debug their programs.
14. When functions have been pair checked/debugged, ask students to swap their programs with their partner.
15. Students then recreate the bracelet using the programs they have been given.
16. Ask students to debug their programs.
17. Have students hand the replica bracelet back to the owner.
18. Have students talk about who the person is that they want to give their matching bracelet to.

Extension Activities

1. Have your students research beads. These could be beads that have special significance in their own culture or Māori culture. They could also research different processes for creating beads that are used around the world (for instance, the glass beads of Venice).

Conclusion

1. Have a discussion about algorithms, debugging and functions. You might like to ask:
   - What is an algorithm?
     A set of clear, step-by-step instructions to solve a problem.
   - Why did we just write down the functions and not every step in the algorithms?
     Because like in programming, once we have defined our functions (written down the words in them), it is easier to call them than it is to rewrite them.
   - What names did you give your functions?
     Thread bead, tie knot, add spacer etc.
   - Why did we need to debug?
     To make sure our bracelets looked the way they were supposed to.

2. Relate the idea of the structure of making their bracelets to a program.
   - Our "program" for making a bracelet is like a computer program. We have different functions that we call when we need them. Having our instructions written this way makes them more manageable and easier to read, just like in programming.

Glossary

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(s). d) Start at the beginning again and repeat these steps until the algorithm does what you were expecting.
Function	A group of instructions that has been given a name.

TR 3 - Lesson Resources: Relevant Links & Answers

Useful Links	
Floss Dance Video	• https://www.youtube.com/watch?v=HPJM6ww5hNU

Escape from the cave - Worksheet Answers

Write the different functions in your bracelet program in the space below:

1. Add bead
2. Add bead
3. Tie knot
4. Add spacer
5. Tie knot
6. Add bead
7. Special bead
8. Add bead
9. Tie knot
10. Add spacer
11. Tie knot
12. Add bead
13. Add bead
14. Tie ends

Student Worksheet : Functions

Write the different functions in your bracelet program in the space below:

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

Additional Resources

Resource & Link	Description
Curriculum Exemplars **Computational thinking** http://technology.tki.org.nz/Technology-in-the-NZC/CT-Progress-outcomes-exemplars-and-snapshots **Designing and developing digital outcomes** http://technology.tki.org.nz/Technology-in-the-NZC/DDDO-Progress-outcomes-exemplars-and-snapshots	The Ministry of Education (Te tahuhu o te Matauranga) provides exemplars of work in areas of technology for both computational thinking and designing and developing digital outcomes. Case studies, teaching snapshots and student showcases to exemplify current technology teaching are available on the Technology Online website.
GameFroot **Mihi Maker** http://make.gamefroot.com/#/activity/mihi-maker	Mihi Maker is a fun easy to use activity that combines coding, social studies and indigenous culture. Students can code their own mini-game and learn how to introduce themselves in Te Reo Māori.
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.
BBC **Bitesize Computing** https://www.bbc.com/education/subjects/z34k7ty	The BBC Bitesize Computing website provides lessons in computers in society, binary and data representation, hardware, software, networks, databases and programming.

Resource & Link	Description
ScratchJr **Website & Free App** https://www.scratchjr.org/	ScratchJr is a free app that is available on both Apple and Android devices. It includes many pre-programmed lessons that allows children to program their own interactive stories and games. By snapping together graphical programming blocks, children can make characters move, jump, dance, and sing. In the process, children learn to solve problems, design projects, and express themselves creatively on the computer. They also use math and language in a meaningful and motivating context, supporting the development of early-childhood numeracy and literacy.
Tutorials https://www.scratchjr.org/teach	The ScratchJr website offers a range of activities that gives you and students a quick way to learn how to do new things with ScratchJr.
Teacher Resources https://www.scratchjr.org/teach/curricula/animated-genres	ScratchJr offers curricula that introduces computer science concepts to children, while allowing them to practice critical thinking and problem-solving skills. Lessons cover topics like programming, expressing through technology, and user-centered design.
Scratch **Website** https://scratch.mit.edu/	The Scratch App and website has pre-programmed lessons that allows students to create digital stories, games, and animations. It also allows them to share their creations with others.
Video Tutorials https://scratch.mit.edu/help/videos/	The Scratch website has a series of video tutorials that take you through a full range of skills needed to use Scratch.
Usage Tutorials http://scratch.redware.com/videos	The Scratch website offers a range of tutorials created by users.
Code.org: **Hour of Code** https://code.org/learn	This resource provides an opportunity for differentiation in levels of difficulty and the option to switch between the easier (dragging & dropping blocks of JavaScript code) and more difficult (writing the code) versions. It provides a range of well-known sequential scenarios to learn how to code (e.g. by using Star Wars characters, Pacman, etc).
Google: **Made with Code** https://www.madewithcode.com/	This website contains a series of computer programming tutorials directed towards young female learners.