



PROGRESS OUTCOME 1

# Collecting pollen

## Annotation

Cheryl-Ann responds to the task of getting the bee to the hive by breaking the problem down into a set of precise instructions. She tests these instructions with a peer to make sure they work.

## Background

The class is studying bees as part of a school-wide science inquiry about insects. They have been investigating the way in which bees collect pollen from different plants and take it back to their hive.






## Task

The students each get a grid showing a bee, flowers and a hive. Their task is to create a set of instructions for the bee to follow in order to collect pollen from each flower and take it to the hive. As a class, they compile a list of “direction” words that they can refer to. They decide to use “up”, “down”, “left” and “right”.

The students then test their instructions by giving them to a peer to follow.

**This task is linked to the work the class is doing in science. It further develops their literacy skills in giving and following instructions, plus their understanding of the positional language used in the geometry strand of mathematics.**

Get the honey bee to its hive. Collect pollen from the flowers as you go.

## Student response

right up up up  
right right down down  
right right up  
up right right right up

**Mr Cusack:** *What did you notice when Hemi tested your instructions, Cheryl-Ann?*

**Cheryl-Ann:** *They worked. He went to each plant and got to the beehive.*



PROGRESS OUTCOME 1

# Teaching robots to dance

## Annotation

The students break down a familiar process into a set of precise and unambiguous instructions. They articulate these instructions, identify where they have made mistakes, and correct the instructions.

## Background

As part of health and physical education, students have been learning names of body parts, practising movement skills, and developing their awareness of space around their bodies. This has included learning the song “Heads, shoulders, knees, and toes” and its associated actions.

## Task

The teacher, Ms Sinclair, asks the students to teach “Heads, shoulders, knees, and toes” to a robot. She explains that computers can only understand specific, simple instructions. So to teach a robot to do something, the task has to be broken down. The students will have to think carefully about the song and its actions and break them down into clear steps.

To help the students with their instructions, Ms Sinclair offers to pretend to be the robot.

**Health and physical education provides a meaningful context for this computational thinking task. The task also provides an opportunity for students to work together to achieve a particular purpose.**

## Student response

- Ms Sinclair:** *What's the first thing you need to tell the robot to do?*
- Ani:** *You go "Heads, shoulders ..." (starts singing the song and doing the actions)*
- Ms Sinclair:** (in a pretend robot voice) *I don't understand.*  
(in her normal voice) *How would you explain the first thing you're doing?*
- Ani:** *First you do this.* (puts her hands on her head)
- Ms Sinclair:** *Can you describe exactly how you do that?*
- Ani:** *You put your hands on your head.*
- Ms Sinclair:** (puts her hands on her head)
- Oliver:** *And then you sing, "Heads".*
- Ms Sinclair:** (puts her hands on her head and sings "Heads")  
*That's right! What should I do next?*
- Petra:** *Now you do this.* (sings "shoulders" and puts her hands on her shoulders)
- Ms Sinclair:** (in pretend robot voice) *I don't understand.*  
(in her normal voice) *Could you explain exactly how I do that?*
- Petra:** *You put your hands on your shoulders, and then you sing "shoulders".*
- Ms Sinclair:** (puts her hands on her shoulders and sings "shoulders").  
*Great! Now what do I do next?*
- Hone:** *Now you put your hands on your knees and sing "knees".*  
  
(all students do this together)
- Hone:** *Then you put your hands on your feet and sing "and toes, knees and toes, knees and toes".*  
(teacher puts her hands on her feet and leaves them there the whole time while singing "and toes, knees and toes, knees and toes")
- Hone:** *No! You put your hands on your feet and sing "and toes", then on your knees and sing "knees", then back on your toes and sing "and toes".*  
(teacher does this)
- Ms Sinclair:** *Great! Now let's start from the beginning. Can you give me all the instructions, one by one, as we do the song?*



PROGRESS OUTCOME 2

# Getting to the hive

## Annotation

Harry responds to the task by breaking down the problem into a series of steps and creating an algorithm (a set of instructions). He then puts these instructions into a programming environment. In doing this, he shows he can:

- decompose a task into a set of instructions
- create a program that includes sequence and outputs
- test his program and use a simple debugging strategy to find and fix his mistakes.






## Background

The class is studying bees as part of a school-wide science inquiry about insects. They have been investigating the way in which bees collect pollen from different plants and take it back to their hive.

## Task

The students each get a grid showing a bee, flowers and a hive. Their task is to create a set of instructions for the bee to follow in order to collect pollen from each flower and take it to the hive. As a group, they compile a list of “direction” words that they can refer to. They decide to use “up”, “down”, “left” and “right”.

Get the honey bee to its hive. Collect pollen from the flowers as you go.

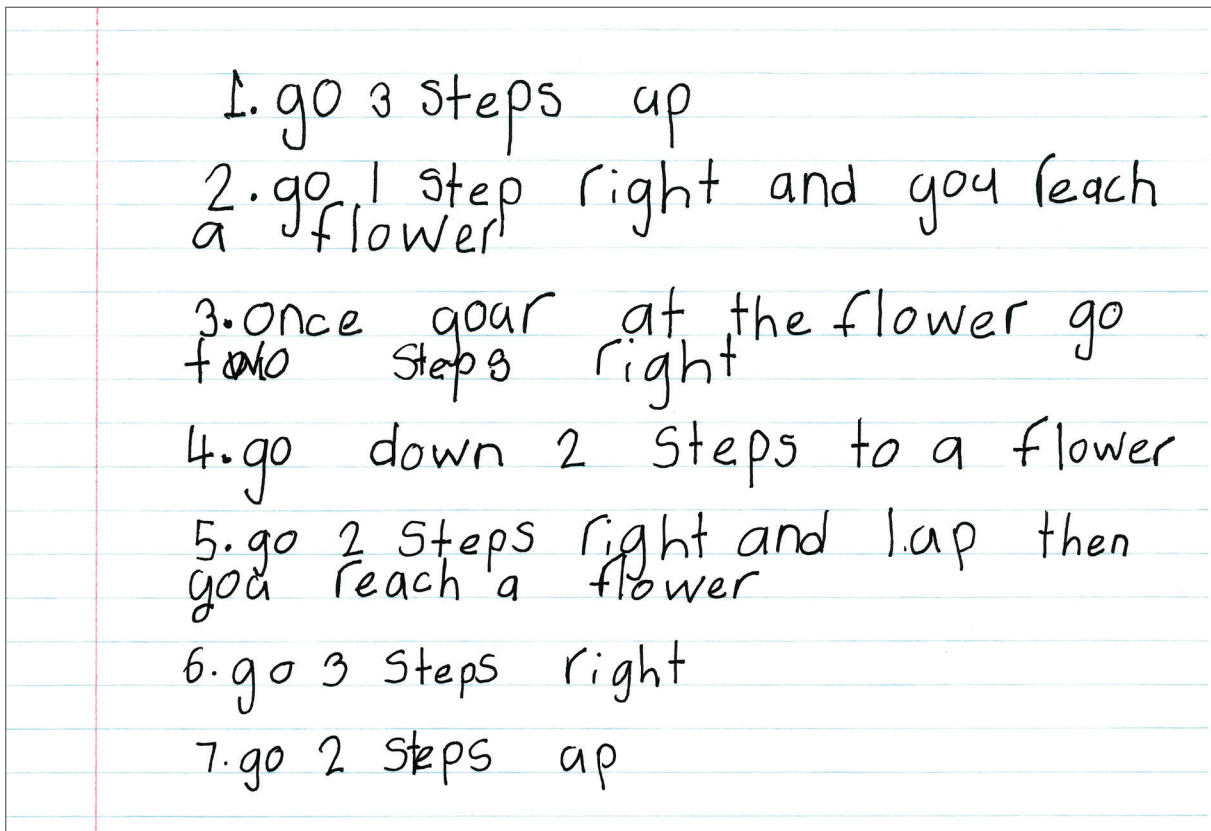
Mr Cusack puts a template of the grid into an introductory programming environment (in this case, ScratchJr). He asks the students to put in their step-by-step instructions (algorithms) and then test them to ensure that the sequence is correct.

As well as asking the students to put in the sequence, Mr Cusack suggests they make their programs more interesting for the user.

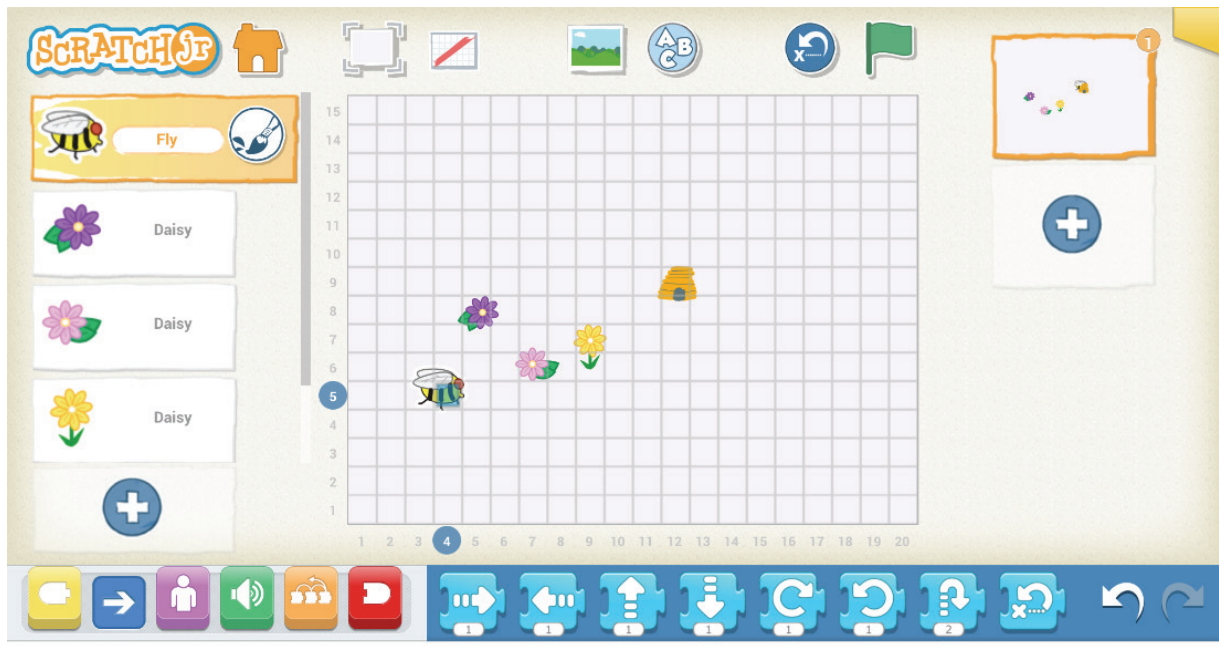
**The task builds on the work the students have been doing to develop their skills in giving and following oral and written instructions (literacy) and in creating simple maps to show position and direction (geometry strand in mathematics). (See Exemplar 1.)**

---

### Student response



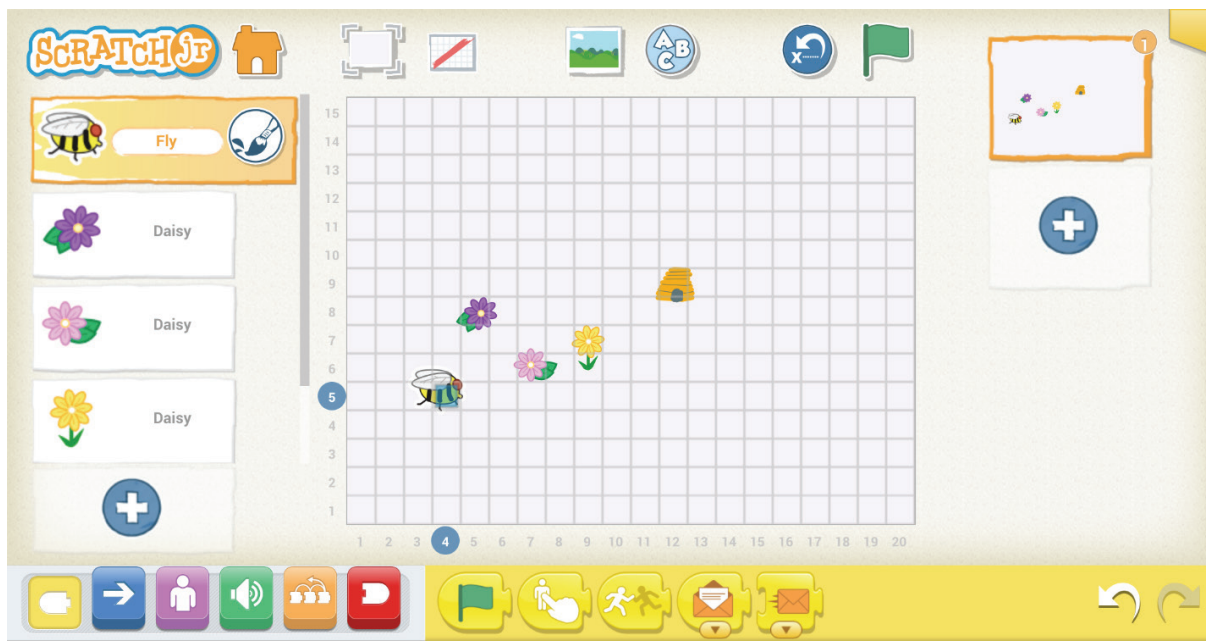
Although he writes his instructions correctly, Harry incorrectly uses “left” instead of “right” at the second step when he puts it into ScratchJr. After Harry fixes this error, his program looks like this.



**Mr Cusack:** *What happened when you put your instructions into the program? Did your bee get to the hive?*

**Harry:** *The first time I tried I didn't put it in properly, so it didn't work. I found a mistake and fixed it, and then my bee got to the hive.*

**Mr Cusack:** *Well done. How about adding some output blocks to make your program more interesting?*



**Mr Cusack:** *What did you add to your program?*

**Harry:** *You can tap the bee now to make it go, and it makes a noise when it gets to a flower.*

**Mr Cusack:** *How did you know when to put the sounds in?*

**Harry:** *I counted how many squares to a flower so it had to go there.*

Downloaded from <http://technology.tki.org.nz>

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab (<http://scratch.mit.edu>).

Copyright © Ministry of Education 2017, except for Scratch images

The program used in this exemplar is not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-197-5





PROGRESS OUTCOME 2

# How Māui slowed the sun

## Annotation

The students show that they can break down a task into smaller steps and create a set of instructions to accomplish that task. They think logically and predict where the “robot” will be on the grid after each instruction in their algorithm.

When they encounter problems in their instructions, they can explain what these problems are and work collaboratively to fix them.

They evaluate the success of their instructions by testing them, and they communicate effectively when they do this.

The students understand that there are multiple algorithms that could solve this same problem.

## Background

The students have been exploring different ways of giving instructions to achieve the same outcome. They have learned that computers need a clear and precise set of instructions because they can't think for themselves, and that these instructions are called algorithms.

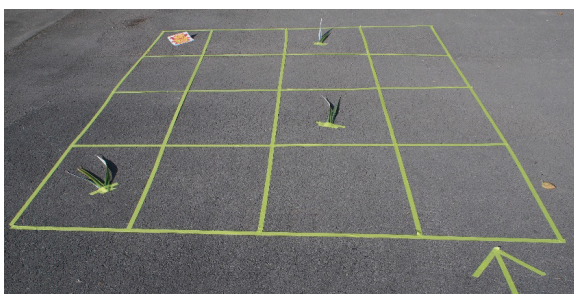
The students have been reading the legend “How Māui slowed the sun”, and Mrs Chen decides to use this story as the context for developing their computational thinking.

## Task

The students' task is to direct Māui (a “robot” student) around a grid to collect flax and take it to Tama-nui-te-rā, the sun. The grid has been marked out on the ground.

The students work in groups to write an algorithm to guide Māui around the grid, collecting all the flax before taking it to Tama-nui-te-rā. They are only allowed to give three different

instructions: “take a step forward”, “turn to your left”, and “turn to your right”.



The instructions must be written using these symbols:

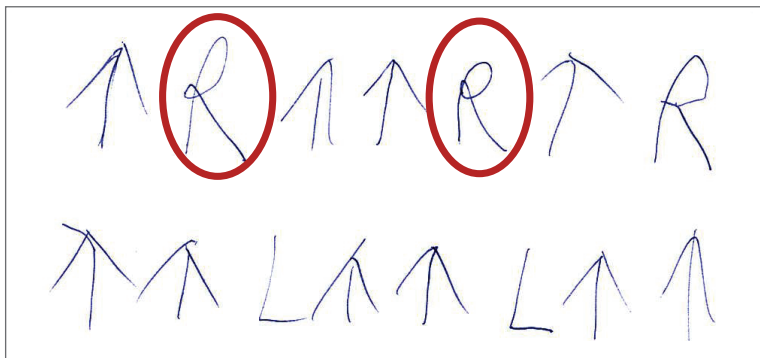
- “↑” (take a step forward)
- “L” (turn to your left)
- “R” (turn to your right).

Māui, the “robot”, follows these instructions, starting by standing on the green arrow and facing forwards.

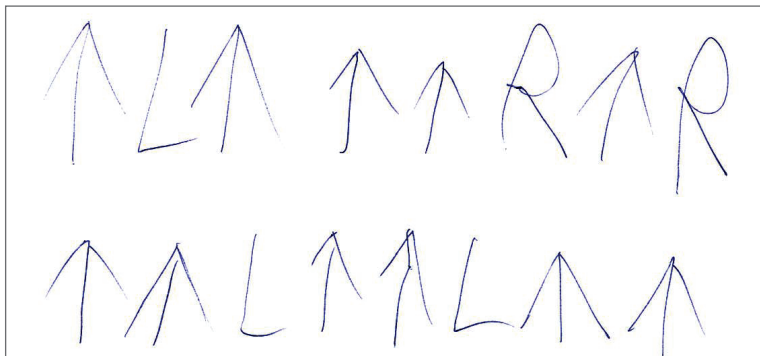
The group observes whether Māui successfully collects the flax and ends up on Tama-nui-te-rā.

**This task has been designed to further students’ capability in working as a group to achieve a specific outcome. It also deliberately introduces symbols as a means of communicating an algorithm.**

### Student response



There are two mistakes in one group’s first algorithm. They work together to fix these and write an algorithm that works.



**Mrs Chen:** *What happened when Māui tried to follow your instructions?*

**Simon:** *We got it wrong the first few times. The first mistake was that we made him turn right instead of left, but after we’d fixed that we found another bug, because we didn’t make him walk far enough to get to the flax. But it works now.*

**Mrs Chen:** *Is this the only way you could have solved this problem?*

**Mira:** *We could try going to the flax squares in a different order, but this way works well.*



PROGRESS OUTCOME 2

# Catching chickens

## Annotation

Stephanie's response shows that she can:

- decompose a problem into a set of step-by-step instructions (algorithm)
- create a program in a simple programming environment based on this algorithm
- test her program, using a simple debugging strategy to fix the errors she finds.

## Background

The students in Stephanie's learning group have been learning about algorithms and how they can use these to write computer programs. They have been using an introductory programming environment (ScratchJr) and experimenting with making characters move. They get frustrated when their programs do not work as expected.

The students have discussed the need to give very precise instructions in their programs. They have worked on ways to identify problems in their programs ("bugs") so they can fix these ("debugging"). They practised debugging by working through instructions, one step at a time, and describing what they think will happen.

## Task

Ms Santos provides students with a ScratchJr template, and asks them to create a program where a boy takes two escaped chickens back to the barn. She also provides a model of where the characters should be positioned in the scene after the program has run.

## Template



## Model of completed scene

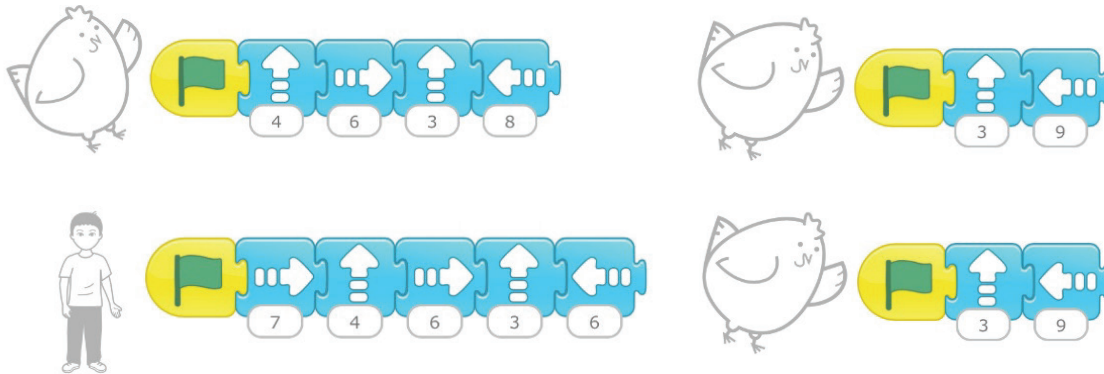


Before they start work in ScratchJr, students think through the series of steps (algorithm) they need to identify and then write their program.

**This task requires the students to think logically as they plan their algorithm. It also encourages them to persevere until they achieve the specific outcome modelled by Ms Santos.**

## Student response

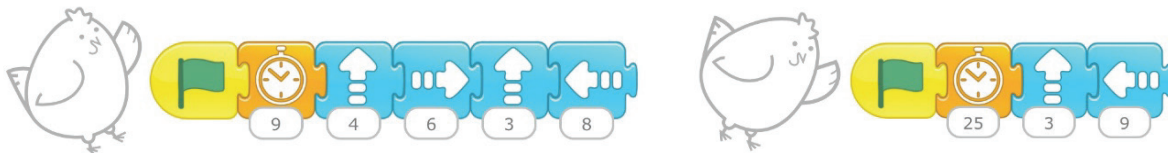
Stephanie plans specific instructions for the two chickens and the boy, then writes a program.



She tests her program, but the chickens move too soon and get to the barn before the boy.



Stephanie debugs her code by stepping through each of the instructions she has given. She realises there is a bug in the code for the chickens and edits her program to fix this.



Stephanie runs the program again, and it completes successfully.

**Ms Santos:** *What was the bug in your program?*

**Stephanie:** *I wanted the chickens to wait until the boy got to them before they started moving, but I'd told them to move when I pressed the green flag. I changed it so that now they wait until he gets to them before they start moving.*

Downloaded from <http://technology.tki.org.nz>

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab (<http://scratch.mit.edu>).

Copyright © Ministry of Education 2017, except for Scratch images

The program used in this exemplar is not officially endorsed by the Ministry of Education.

ISBN: 987-1-77669-199-9



# Dance moves

## Annotation

Maia's response to the task demonstrates that she:

- can develop a basic computer program in a block-based programming environment
- understands the need to be precise because computers cannot infer
- understands the difference between an algorithm and a program
- can identify patterns in algorithms and programs where iteration can be applied.

## Background

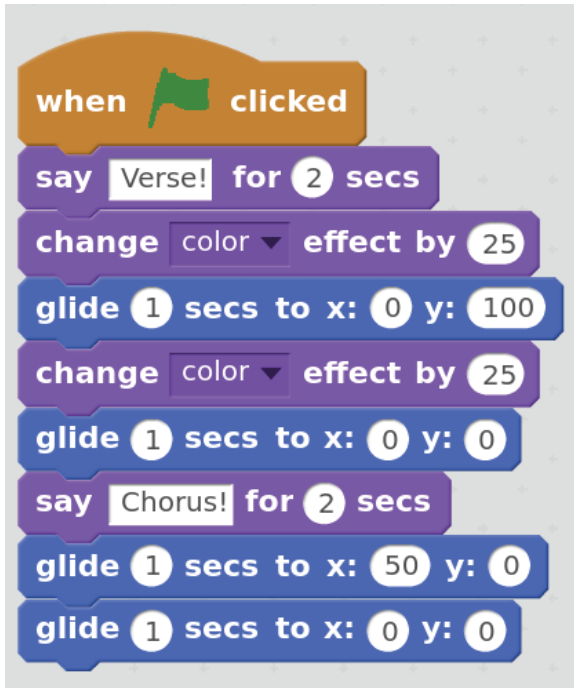
The students in Maia's learning environment have had some experience of working in a block-based programming environment (Scratch) and been introduced to the concept of iteration using repeat blocks as a way of simplifying algorithms and programs.

## Task

Using their new knowledge of iteration, the students are asked to use a block-based programming environment (Scratch) to create an animated character that moves to music. The animation needs to repeat the same sequence of movements in the verse of the song and a different sequence in the chorus.

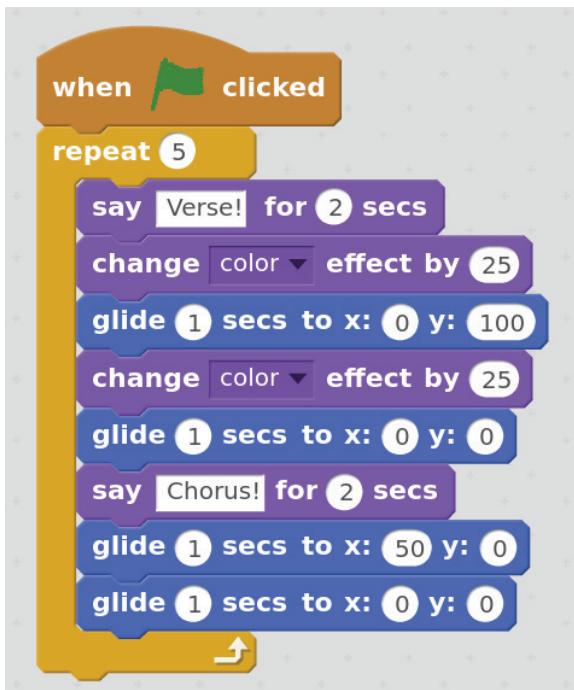
**Mrs Anderson takes the opportunity to link iteration in digital technology to the way her students have been writing repeated patterns in their music activities. The task also enables her to reinforce the students' emerging mathematical understanding of the x and y axes.**

## Student response



After creating her program, Maia finds that every time she wants her character to dance for a verse and then a chorus, she has to press the green flag again.

Using the knowledge she has gained about iteration, she modifies her program using the repeat block so that her character dances to the verse and chorus five times.



**Mrs Anderson:** *Can you explain what your character does?*

**Maia:** *In the verse, it jumps up and down and changes colour. Then for the chorus, it moves to the right and then to the left.*

**Mrs Anderson:** *How many times does it happen?*

**Maia:** *Five times!*

Downloaded from <http://technology.tki.org.nz>

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab (<http://scratch.mit.edu>).

Copyright © Ministry of Education 2017, except for Scratch images

The program used in this exemplar is not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-200-2



PROGRESS OUTCOME 3

# Climbing stairs

## Annotation

Wiremu can create an algorithm that is different to his teacher’s but that accomplishes the same task. He can implement his algorithm in a programming environment, and he can improve it by including iteration using a repeat block.

By doing this, Wiremu demonstrates that he:

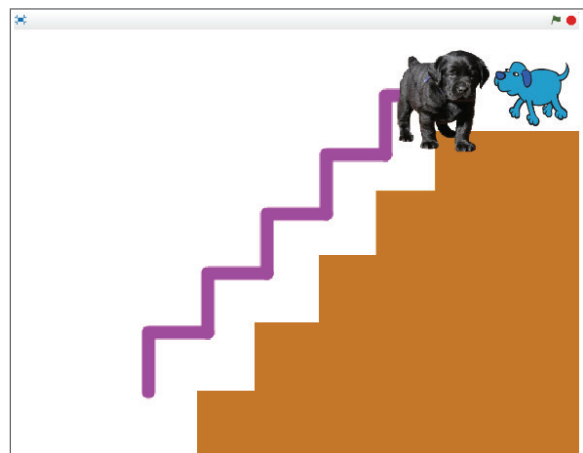
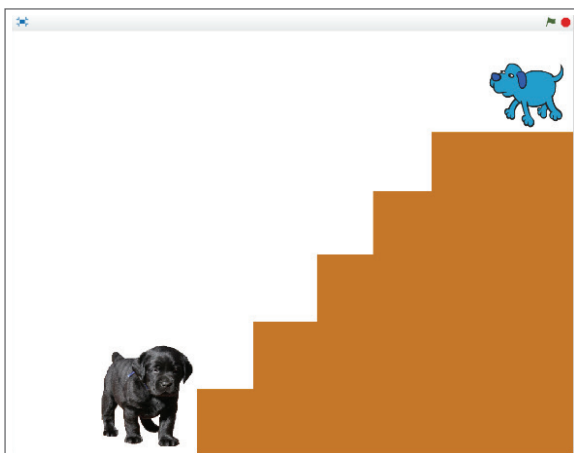
- understands that there can be more than one algorithm for solving the same problem
- can create an algorithm to develop a basic computer program in a block-based programming environment
- can identify patterns where he can apply iteration.

## Background

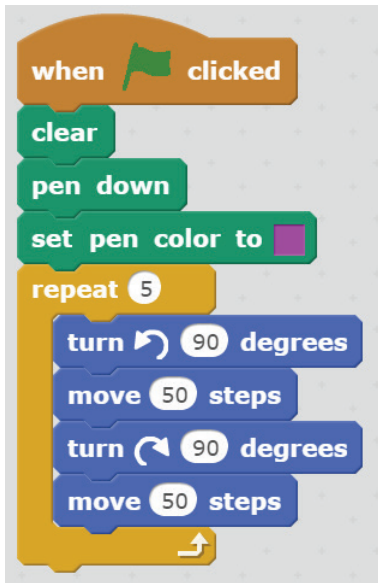
The students in Wiremu’s group have some experience in creating programs involving sequence and iteration (using repeat blocks) in a block-based programming environment (Scratch).

## Task

Mr Chandra asks the students to create a program to move a puppy to the top of some stairs with a line behind it showing its path.







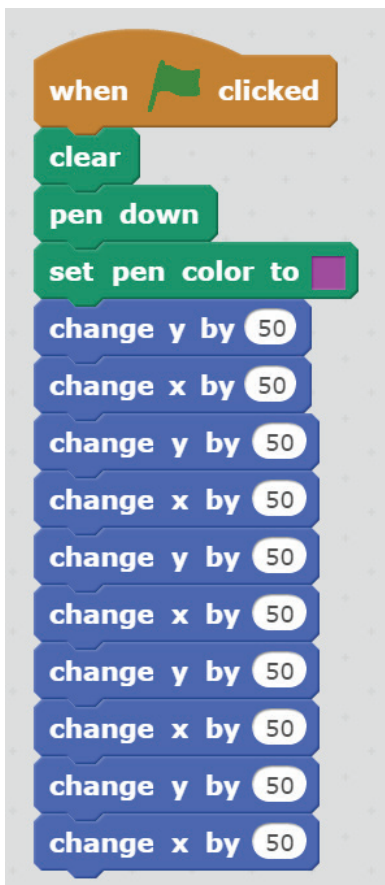
Mr Chandra shows the students this program.

He then asks them to create another program that solves the problem using a different algorithm.

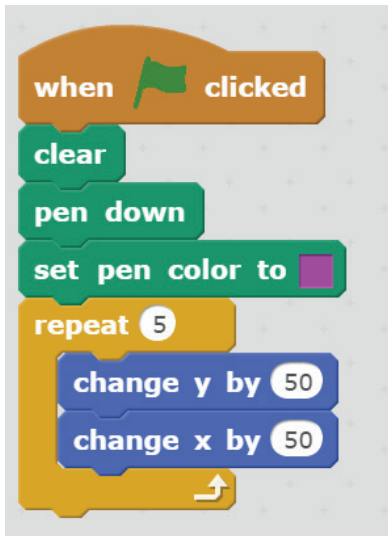
As well as highlighting how symbols are used to communicate instructions (including iteration), the task enables the students to use their mathematical understanding of right-angled turns and their emerging understanding of how x and y can be used to signal direction.

---

### Student response



Wiremu creates a program that follows a different algorithm to Mr Chandra's.



He notices there is a pattern in his code and improves his program by adding iteration using a repeat block.

Downloaded from <http://technology.tki.org.nz>

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab (<http://scratch.mit.edu>).

Copyright © Ministry of Education 2017, except for Scratch images

The program used in this exemplar is not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-201-9



# Coded messages

## Annotation

Annika's response shows that she understands the concept of using binary digits to represent types of information. She shows that she can work systematically to convert the numbers 1-26 to binary numbers (base 2) and use these binary numbers to represent text.

## Background

The class has been exploring how the binary number system works using just two digits, 1 and 0. Ms Richards has explained that computers store data using binary digits (bits), and that this is why computers are called digital devices.

## Task

The students are given a table to help them to convert letters to numbers and these numbers to binary. Ms Richards explains that each of their binary numbers should contain five bits.

Each student writes their name in binary and then writes a message in binary for another student to read.

**Ms Richards links this task to the students' mathematical understanding of how the base 2 number system works.**

## Student response

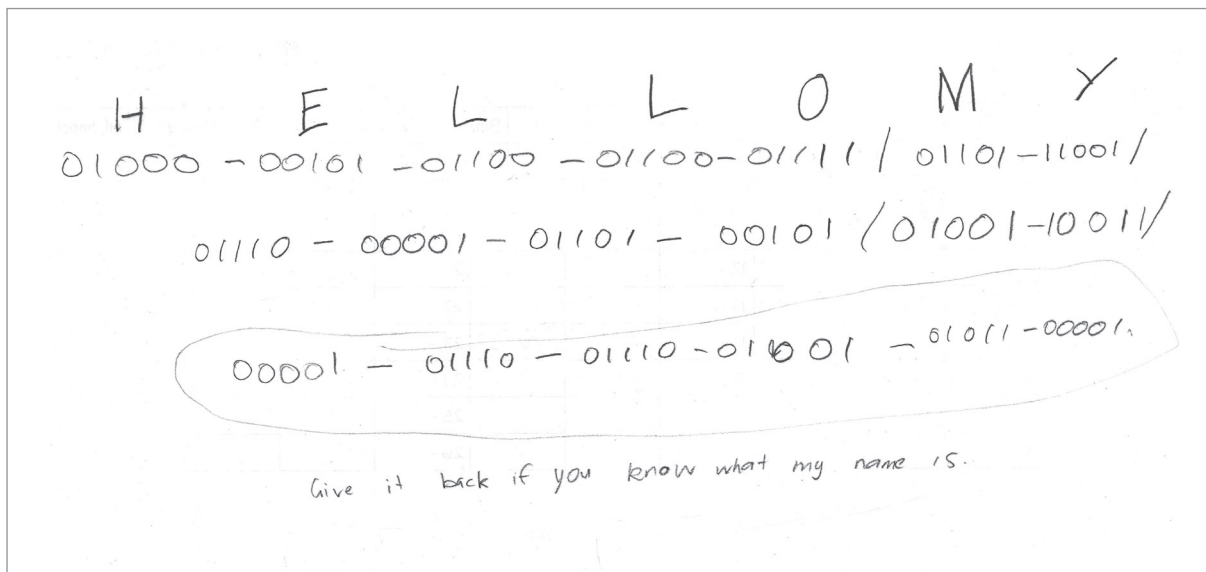
Annika completes the table, then writes her name using binary number representation.

Base 10	Binary	Alphabet	Base 10	Binary	Alphabet	Base 10	Binary	Alphabet
0	<del>00000</del>		9	01001	I	18	10010	R
1	00001	A	10	01010	J	19	10011	S
2	00010	B	11	01011	K	20	10010	T
3	00011	C	12	01100	L	21	10011	U
4	00100	D	13	01101	M	22	10100	V
5	00101	E	14	01110	N	23	10111	W
6	00110	F	15	01111	O	24	11000	X
7	00111	G	16	10000	P	25	11001	Y
8	01000	H	17	10001	Q	26	11010	Z

A N N I K A  
 1 14 14 9 11 1

00001 - 01110 - 01110 - 01001 - 01011 - 00001

Annika then writes a message which reads "HELLO MY NAME IS ANNIKA" and gives it to another student, who figures out what it says.



Ms Richards decides to explore Annika's understanding of bits.

**Ms Richards:** *We're using five bits to make our numbers, so how many numbers can we make?*

**Annika:** *If we kept going in the table, we could get to 32.*

**Ms Richards:** *How could we go further and make more numbers?*

**Annika:** *We would need to use more bits for our binary numbers.*



# Robotics challenges

## Annotation

Jamal and Roman are able to decompose a robotics problem into logical steps, write an algorithm for the problem, and create a program for a robot. They test whether the program functions as intended when downloaded to the robot and then refine their algorithm and program. In doing so, they show that they are able to use a block-based programming environment to create a program that uses:

- sequence (blocks in the correct order)
- inputs (readings from colour and ultrasonic sensors)
- selection and iteration through the use of repeat-until loops.

They also show that they can:

- refine an algorithm and program to be more precise
- understand connections between programming and algorithms by using an algorithm as code comments to guide and document programming.

## Background

The students have been learning to program robots using the LEGO MINDSTORMS® EV3 integrated development environment (IDE). They have practised writing algorithms to decompose problems. They have worked in teams to write, test and debug programs to solve different challenges, using input from various sensors to make decisions about how a robot's motors should move. They are learning the fundamental building blocks of programs (sequence, selection and iteration) so that they can combine these into larger programs to solve a series of problems.

## Task

The students are asked to work in pairs to write algorithms for problems that may arise when developing larger programs to solve multi-step robotics challenges.

They are then asked to test their algorithm by translating it to LEGO MINDSTORMS® EV3 code, using the algorithm as code comments and checking to see their program works when downloaded to the robot. The students are asked to refine their algorithm (and program) if bugs are found upon testing.

**This task provides an opportunity for students to use the discipline-specific language of computer science.**

## Student response

Jamal and Roman formulate their original algorithm for the problem.

### With your partner(s) write the algorithm for:

Moving forward until the robot detects white, then turning the robot until an object is detected (e.g. a can) 30 cm or closer and then moving the robot toward that object.

Repeat until colour sensor reads > 50:

Turn both motors on

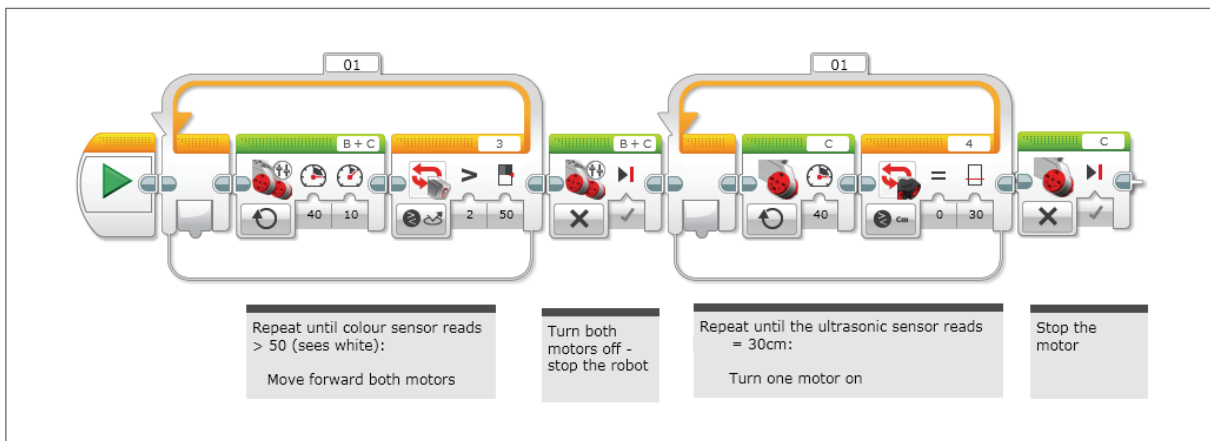
Turn both motors off

Repeat until the ultrasonic sensor reads = 30cm:

Turn one motor on

Stop the motor

They test their algorithm by translating it into a program that uses input from colour and ultrasonic sensors, selection through use of comparison operators (greater than and equal to), and iteration through a repeat-until loop. Their code comments are taken from their algorithm.



**Jamal:** *We tested the program, but the robot just spins around and around and never stops.*

**Mrs MacKenzie:** *What is the logic you are using to stop your repeat loop? Also, have you thought about how the robot will move toward the object once it detects it?*

**Roman:** *We are making our loop continue until it sees an object "equal to" 30 centimetres away. This won't detect anything closer than 30 centimetres. We have to fix our algorithm to look for everything less than 30 centimetres away and add a move-forward step.*

## With your partner(s) write the algorithm for:

Moving forward until the robot detects white, then turning the robot until an object is detected (e.g. a can) 30 cm or closer and then moving the robot toward that object.

Repeat until colour sensor reads > 50:

Turn both motors on

Turn both motors off

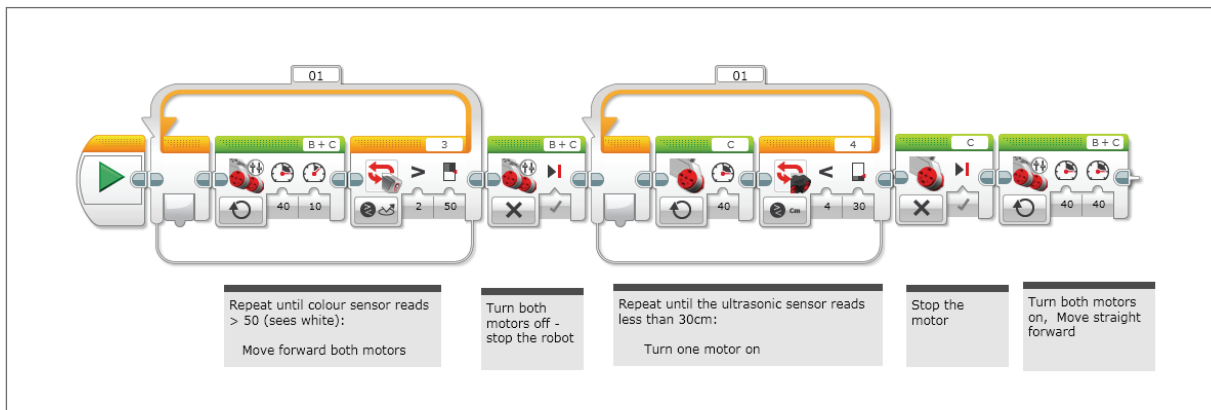
Repeat until the ultrasonic sensor reads < 30cm:

Turn one motor on

Stop the motor

Turn both motors on

When they retest their revised algorithm in the program, the robot moves as intended, showing that they have successfully debugged their program using a range of programming constructs.



Downloaded from <http://technology.tki.org.nz>

LEGO® and LEGO MINDSTORMS® are trademarks of the LEGO Group of companies, which does not sponsor, authorise or endorse this site.

Copyright © Ministry of Education 2017, except for LEGO images

The program used in this exemplar is not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-203-3





PROGRESS OUTCOME 4

# Parity bit magic

## Annotation

Jacob and Sarah recognise that it is important for computers to be able to detect if an error has occurred when data is stored or transmitted over a network such as the Internet.

They are able to demonstrate an error-correction technique using parity bits. By doing this, they:

- show their understanding of how a computer can add extra information (bits) to a file to detect errors
- show how a computer could first detect an error and then correct it.

In applying the techniques, Jacob and Sarah also demonstrate computational thinking by recognising patterns in data and by applying logical reasoning to create even parity and determine the specific place in data where an error has occurred.

## Background

The class has previously learned that computers store and transmit data using patterns of binary digits, or bits (represented by 0s and 1s). They have learned how numbers and text are represented using bits. They have discussed the importance of making sure that data is accurate when it is stored or transmitted and have explored some of the consequences that incorrect data storage or transmission could have in their daily lives. They have also discussed ways in which errors are detected by computers using examples from the check digits on bar codes. As a group, the students participated in a “parity magic trick” activity so that they could determine how a computer might use parity bits to detect and correct errors.

## Task

Mr Liu prepares a Google™ slideshow template and asks the students to work in pairs to respond to three questions to demonstrate their understanding of how to detect and fix computer errors using parity bits. They are required to:

- explain what can go wrong when a computer is saving or sending files
- explain why a computer needs to know if there is a problem and give examples of what can happen if it can't detect errors
- demonstrate how a computer uses parity bits to detect errors.

**The task requires the students to use their knowledge of discipline-specific language to communicate their understanding.**

## Student response

The following screenshots are from Jacob and Sarah's slideshow. The blue text shows their responses.

### What can go wrong when a computer is saving or sending files?

Lots of different things can go wrong when a computer stores a file or when we try to send a file over the internet. Here are some things that can go wrong.

- Sometimes there might be something wrong with the hardware, like a bad part of a disk, or a scratch on a DVD.
- Sometimes there might be a problem with the wifi at school because too many people are on it and the computer can't send everything in the file.
- Sometimes there might be a problem with a barcode, like it is scratched, when you buy at the store or check out a book at the library.
- Sometimes an eftpos card doesn't work anymore after it was near a magnet.

### Why does the computer need to know if there is a problem? What can happen if the computer can't detect errors?

The computer needs to know if there is a problem because otherwise our files would be wrong or maybe they won't open. A message we send might be garbled up and our friends would get the wrong message or couldn't read it. We might be charged the wrong price for something we buy. We might get in trouble at school for not returning a book that we never checked out!

Using parity bit cards, with your partner show how a computer uses parity bits to detect errors. Use photos and notes to explain your thinking.



A computer stores all our files in patterns of on's and off's - these are the bits. The black is off and the white is on. This shows the pattern for storing a pretend file.

Using parity bit cards, with your partner show how a computer uses parity bits to detect errors. Use photos and notes to explain your thinking.



The computer adds another on or off to each row and column to help it know if there are any mistakes in the files.

First the computer needs to make all the rows have an even number of black bits.

The computer uses the extra bits to decide if there has been an error in a bit.

We added a bit card to rows 1 and 3 to make sure all the black off bits were even numbers.

Using parity bit cards, with your partner show how a computer uses parity bits to detect errors. Use photos and notes to explain your thinking.



We did the same thing for the columns by adding an extra row to the bottom.

We added a bit card to columns 2 and 4 to make sure all the black off bits were even numbers.

Using parity bit cards, with your partner show how a computer uses parity bits to detect errors. Use photos and notes to explain your thinking.



We got someone to deliberately flip one of the parity bit cards when we weren't looking, to create an error. We figured out which one it was by looking at the parity bit cards.

We figured out that this is the bit that is flipped or wrong because the row has an odd number of black bits and the column has an odd number of black bits.

This is how a computer would detect an error using parity bits. It could flip the bit back the other way to fix the error.



PROGRESS OUTCOME 4

# Beat the Goalie

## Annotation

Kiri and Ben can analyse a computer application, identify the functional requirements for a game and create an algorithm for each functional requirement.

They develop a computer program that shows they can:

- use variables for representing and storing data (“timer” in the screenshot from Scratch)
- sequence (“wait 1 sec, change timer by -1”)
- use selection and iteration via a repeat block with conditional logic (“repeat until timer = 0”).

They are able to test and debug their code to ensure it functions correctly. Their response demonstrates their ability to apply logical reasoning to a programming problem.

## Background

The students in this learning space have had prior experience with developing programs in a block-based integrated development environment (IDE).

The task below is one part of an activity for developing the “Beat the Goalie” game in a block-based coding language. Other parts of the activity involve students writing, implementing and testing algorithms for all the functional requirements of the game.

## Task

Students are shown a game called “Beat the Goalie”. They are unable to view the code for the game, so can only see the final game in action. Based on what they can see, they are asked to develop a set of functional requirements that describe the game’s objectives and how a player would interact with the game.



Working in pairs, the students are then asked to develop an algorithm for the functional requirement “The countdown timer should count down from 30 and stop the game when time is up”.

They are then asked to create the programming code to implement and test their algorithm for the countdown timer.

Mr Stuart provides a template for the students to record their response to each part of the task.

## Student response

Kiri and Ben develop functional requirements for the game and an algorithm for the countdown timer.

### Functional Requirements for the Beat the Goalie Game

After playing the “Beat the Goalie Game” write down the informal instructions here:

1. Timer counts down from 30 to 0 seconds until game is over.
2. Use left and right arrow keys to move the goalie.
3. The ball will move from left to right automatically.
4. Use spacebar to shoot at the goal and then you get a point.
5. The goalie will say goal or save depends on if you score or not.
6. Score as many goals as you can in the time limit.

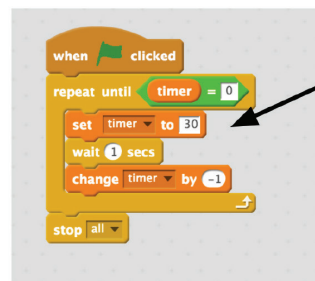
### Algorithm for a countdown timer

1. Make a timer variable
2. Set timer to 30
3. Start at 30 seconds
4. -1 every second until timer= 0
5. Stop the game

They test their algorithm by creating a program in Scratch that uses sequence, selection, iteration, and variables. They realise through testing that their program code is in the wrong sequence, preventing the program from working as they intended. They correct the bug in their logic and get the program to work correctly.

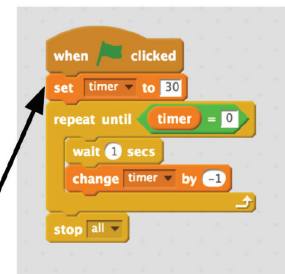
### Computer Program in Scratch for a Countdown Timer

Add a screenshot of your computer program here. If it did not work the first time, add that code and write why it didn't work.



The first code didn't work because we had the “set timer to 30” inside the repeat block. It just kept resetting every time it did the repeat and it never counted down.

We moved it out of the repeat so it only sets the time once at the start.



Downloaded from <http://technology.tki.org.nz>

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab (<http://scratch.mit.edu>).

Copyright © Ministry of Education 2017, except for Scratch images

The program used in this exemplar is not officially endorsed by the Ministry of Education.

ISBN: 987-1-77669-205-7



PROGRESS OUTCOME 4

# Comparing search algorithms

## Annotation

Anahera recognises that computers need to search through a lot of data and that therefore it is very important to have fast and accurate algorithms to perform these tasks. She demonstrates her understanding by describing important factors that people rely on when searching for information on the Internet and the consequences of a slow or inaccurate search algorithm.

Anahera recognises that some algorithms perform better than others for a given search. She demonstrates her computational thinking skills by comparing linear and binary search algorithms and drawing conclusions about their efficiency.

## Background

Anahera's class has been learning about algorithms and how they relate to computer programs. They have developed and tested different algorithms for basic robotics tasks and discussed different contexts in which people depend upon fast, accurate searching and retrieval of data – searching the Internet for information, the supermarket checkout screen for prices, and so on. Before performing the investigation in this activity, the students trialled linear and binary search techniques by looking for numbers hidden under paper cups arranged in random order (representing a linear search) and sorted order (representing a binary search).

## Task

After watching a video on how a Google™ search works, the students answer questions on important factors for humans when searching for information on the Internet.

They then investigate the differences between linear and binary search algorithms by trying to guess the position of a selected word in unsorted and sorted lists of 25, 50, and 100 words. Using a spreadsheet, Ms Young reveals the word after each guess, and the students record the number of guesses it takes to find each answer.

A	B	C	D	E	F	G	H	I
	<b>Unsorted List</b>			<b>Sorted List</b>				
1	aquatic		1					
2	acoustics		2					
3	complex		3					
4	muddle		4					
5	cooing		5					
6	cobweb		6					
7	merciful		7					
8	thoughtful		8					
9	permissible		9					
10	impulse		10					
11	long		11					
12	inconclusive		12					
13	spoon		13	alive				
14	plausible		14					
15	marry		15					
16	mark		16					
17	embarrassed		17	amuck				
18	nine		18	amusing				
19	analyse		19	analyse				
20			20	anger				
21			21					
22			22					

Search Words	How Many Guesses Unsorted	How Many Guesses Sorted
analyse	19	5
cave		
belligerent		

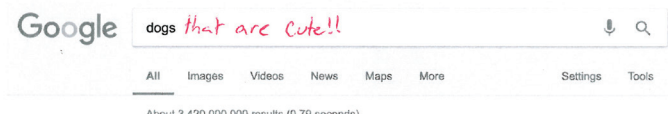
After the students complete the investigation, they answer questions about linear and binary search algorithms.

### Student response

After watching the video on Google searches, Anahera gives the following answers to Ms Young's questions.

### Search Algorithms

One thing that computers need to do every day is search through a lot of data! Google Search is something we all depend upon to give us search results from the billions of web pages of information stored around the world.



We have watched the [Google Searching Algorithms](#) video explaining the basics of their search algorithm works. Based on watching the video and your own experience with searching for information on the Internet, answer the following questions **in your own words**.

*What are some of the most important factors for humans when they are searching for some information on the Internet?*

*I think the most important factors are:*

- getting info-mation quickly because we don't have hours
- The wright info not something about cats if you said dogs
- not all over the page, easy to read in line.

*What would happen if Google used a different search algorithm that didn't meet the factors you listed above?*

- we would have to wait hours to find one thing.
- it would be inapropriate and not helpfull.
- We wouldnt use it
- would have lots of Bugs



Anahera records her results in the linear vs binary search activity on a template provided by Ms Young.

## Linear and Binary Search Algorithms

Two common search algorithms that are studied in computer science are Linear and Binary search. We are going to do an investigation into those two types of search algorithms, so you can determine which one would work the best in different situations.

We are going to place a guessing game using words that are in unsorted (random) and sorted (alphabetical order) lists. Pretend you are a computer using an algorithm to search for the word in each of the lists. Your teacher will reveal each word as per your instruction of where to search in the list.

Record how many guesses it took you to find the following words in the unsorted and sorted lists of **25 words**:

Search Words	How Many Guesses	How Many Guesses
	Unsorted	Sorted
grab	6	7
heavenly	14	6
greedy	4	4

Record how many guesses it took you to find the following words in the unsorted and sorted lists of **50 words**:

Search Words	How Many Guesses	How Many Guesses
	Unsorted	Sorted
dislike	45	4
ear	<del>22</del> 22	<del>6</del> 6
clever	14	5

Record how many guesses it took you to find the following words in the unsorted and sorted lists of **100 words**:

Search Words	How Many Guesses	How Many Guesses
	Unsorted	Sorted
cave	1 ☺ lucky	7
analyse	45	6
belligerent	67 ☹	4

Anahera gives the following answers to the questions on linear vs binary search algorithms.

## Linear and Binary Search Algorithms Comparison

Now that you have completed your investigation using the guessing game, answer the following questions about Linear and Binary Search Algorithms:

*Overall, which algorithm was the most efficient (took the least amount of guesses) to find the word being searched for? Explain why you think this is the case.*

Binary - because you find it in minutes. Its fast because you cut it in half each time.

*If there are 100 items to search through, what is the maximum number of searches that a linear search will have to perform? What is the maximum number of searches that a binary search will have to perform.*

linear = 100

Binary = 7

*Can you explain the situations where a computer would not be able to use a binary search algorithm and would have to use a linear search algorithm?*

~~When the~~ if there is a highscore list or a waiting list. because its not sorted



PROGRESS OUTCOME 5

# Programming in Swift Playgrounds

## Annotation

Dion is able to discuss the difference between a “for loop” and a “while loop” and can explain when different types of iteration control structures are best applied to solve a problem. He shows insight into the operation of a while loop and demonstrates an understanding of the use of functions in a program and how they make programming easier to read and more efficient.

Dion is able to independently develop, test, and debug a computer program in the Swift programming language. He can apply algorithmic thinking and can decompose problems to determine which programming structures are best for solving a problem that has several possible solutions.

His programming code demonstrates:

- an understanding of variables
- the use of appropriate naming conventions
- selection with conditional logic, including compound logical operators (&&)
- the use of functions
- iteration using a while loop.

## Background

Dion and his fellow students are learning to program in Swift using Swift Playgrounds (a free iOS-based environment for learning programming). They have been writing algorithms to solve open-ended, puzzle-based problems and applying the algorithms when writing, testing, and debugging their programs.

## Task

The students are asked to take on a challenge that is part of Swift Playgrounds. The challenge requires them to understand and apply:

- programming knowledge of variables and naming conventions
- differences between types of iteration control structures
- the appropriate use of these control structures
- selection through the use of logical comparison operators.

The task provides an opportunity to use the discipline-specific language and skills of a computer scientist.

**Challenge: Collect exactly three gems and toggle open four switches**

To collect the right number of gems and toggle open the right number of switches in this challenge, you'll need two separate variables. As your character handles the gems and switches, you'll increment the right variable so you stop at the right time.

Start by declaring one variable for the number of gems, and another for the number of switches. Use the guide below to name your variables.

**Naming a variable**

**Use camelCase:** Start with a lowercase letter on the first word, and capitalize each new word.

**Use a descriptive name:** Give the variable a name that tells you what it stores, like `numberOfGems`.

Increment each variable by 1 when you collect a gem or toggle a switch.

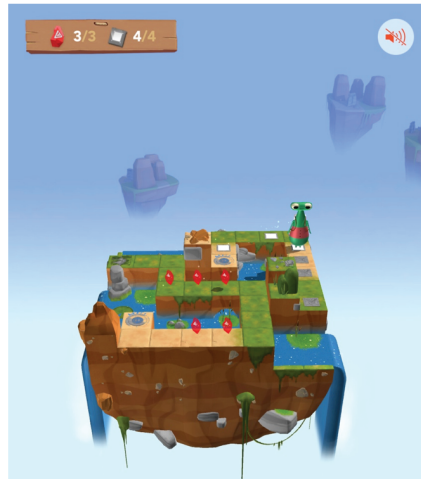
Use one of the comparison operators below to create a condition to tell your character when to stop. Determine what type of control structure would be best to use (for or while loop) to solve this problem.

**Comparison operators:**

**Less than operator:** `(a < b)` returns true if a is less than b.

**Greater than operator:** `(a > b)` returns true if a is greater than b.

**Equal to operator:** `(a == b)` returns true if a equals b.



## Student response

The following response is from Dion's post to the Google™ Classroom.

**Mrs Patel:** *Sometimes a for loop is better for solving a programming problem, and sometimes a while loop is better suited to the task. Describe the difference and when you would use one over the other.*

**Dion:** *A for loop means you repeat a pattern a certain amount of times. A while loop means you repeat a pattern while something is true. You use a while loop when you have a condition you have to reach, and you use a for loop when you know the amount of times you want to repeat your pattern, and it is unchanging. When using a while loop, your condition must be able to become false, otherwise your loop will be infinite.*

**Mrs Patel:** *What is the purpose of functions when writing a program?*

**Dion:** *Functions let you group a sequence of actions you may need to repeat multiple times, which makes your final code much shorter, tidier, and easy to read. It also means it's quicker to write code, by just writing the function rather than the whole sequence when you need to use it in your code.*

The following is a sample of Dion's code after he has tested, debugged and commented it. The code demonstrates the use of variables, iteration through while loops, predefined functions, selection through conditional logic and use of compound logical operators.

```
//create the variables to store the number of gems and switches
collected

var switchesToggled = 0
var switchesToggled = 0

//the character will move forward while it hasn't collected all the
gems and switches

while switchesToggled < 4 || gemsCollected<3 {

    moveForward()

    //the character will turn right if both the conditions are true
    if isBlocked && isBlockedLeft{
        turnRight()    }

    //the character will turn left if both conditions are true
    if isBlocked && isBlockedRight{
        turnLeft()    }

    //increase the gemsCollected count if both conditions are true
    if isOnGem && gemsCollected<3{
        collectGem()
        gemsCollected = gemsCollected+1    }

    //increase the switchesCollected count if both conditions are true
    if isOnClosedSwitch && switchesToggled<4{
        toggleSwitch()
        switchesToggled = switchesToggled+1
    }
}
```

Downloaded from <http://technology.tki.org.nz>

Swift Playgrounds and Swift are trademarks of Apple Inc., registered in the U.S and other countries.

Google is a registered trademark of Google Inc, used with permission.

Copyright © Ministry of Education 2017, except for student work copyright © student, and Swift Playgrounds image

The program used in this exemplar is not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-207-1



PROGRESS OUTCOME 5

# 24-bit colour calculations

## Annotation

Grace is able to use a spreadsheet to convert between 24-bit binary and decimal values. She can use logical reasoning to work out that a colour displayed by a computer is stored as a 24-bit digital value that can be calculated, and that this is what enables computers to display colours. She can calculate how many different colours can be displayed by a computer using 4, 8, 16, and 24 bits. She is able to describe how a computer uses bits (binary digits) and pixels to represent a digital image, and how colour is converted into digital data.

## Background

The students in Grace's learning space have been using their knowledge of binary representation to look at how pixels and colours are represented.

## Task

The students are asked to read and complete the tasks outlined in the following worksheets.

**The task provides an opportunity to explore the mathematical relationships between bits, binary, and powers of 2.**

## Student response

Grace

### Creating Colours from Bits

We have learned that each pixel on the screen is made up of three tiny red, green and blue lights. By changing the amount of light each red, green and blue light emits, we can mix millions of different colours, in fact more than the human eye can perceive.

The following activities will help you work out how a particular colour is represented in both decimal (for humans to understand) and binary (for the computer to understand).

Use an online RGB colour picker to find the red, green and blue values for a colour of your choice. Add a screen shot of the colour you have selected and write in the values below:

Red 250 Green 15 Blue 150

Open the **RGB Colour Calculator** spreadsheet and use the **Binary Calculator** to work out the binary representation for the red, green and blue values and write in the binary below:

Red 11111010 Green 00001111 Blue 10010110

Now use the **Colour Code Calculator** on the spreadsheet to work out the calculated number for the colour as well as the final binary number for the colour:

Red 250 X 65536 = 16384000

Green 15 X 256 = 3840

Blue 150 X 1 = 150

Total Colour Value 16,387,990

Colour Value in Binary 1111010000011110010110

The total colour value in binary for 16,387,990 is the same as the RGB in order.

**Ms Williams:** How did you use the spreadsheet to help you with your calculations in this part of the activity?

**Grace:** First I picked a shade of pink with the RGB colour code of 250, 15, 150. I changed the bits between 0 and 1 to calculate the binary values for the red, green, and blue colours. Next, I used the spreadsheet to help me calculate the total decimal and binary number that represents the colour.

**FLIP THE BITS BETWEEN 0 AND 1 TO CHANGE THE VALUES FOR RED, GREEN AND BLUE**

8 RED BITS								8 GREEN BITS								8 BLUE BITS								
1	1	1	1	1	0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1	1	0
128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	
128	64	32	16	8	0	2	0	0	0	0	0	0	8	4	2	1	128	0	0	16	0	4	2	0
<b>250</b>								<b>15</b>								<b>150</b>								

**RGB COLOUR CALCULATION USING BINARY**

250    15    150

11111010 00001111 10010110 11111010000011110010110

**CHANGE THE VALUES IN THE DECIMAL COLUMN BELOW TO CALCULATE THE BINARY AND DECIMAL COLOUR VALUE**

8 RED BITS								8 GREEN BITS								8 BLUE BITS							
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
8388608	4194304	2097152	1048576	524288	262144	131072	65536	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

**RGB COLOUR CALCULATION = RED \* 65536 + GREEN \* 256 + BLUE \* 1**

	BINARY	DECIMAL	x		
R	11111010	250	x	65536	16384000
G	00001111	15	x	256	3840
B	10010110	150	x	1	150
<b>TOTAL (COLOUR VALUE)</b>	<b>11111010000011110010110</b>				<b>16387990</b>



How many different colours can be displayed with 24-bits (**8 red bits, 8 green bits and 8 blue bits**)? Show your calculation below:

8 bits is one byte. 1 byte has 256 numbers that it can represent (0-255). So you have 256 of each colour.  
 $256 \times 256 \times 256 = 16,777,216$  can be displayed.

How many different colours can be displayed with 16-bits (**5 red bits, 6 green bits and 5 blue bits**)? Show your calculation below:

5 bits can represent 32 different numbers.  
6 bits can represent 64 different numbers.  
 $32 \times 64 \times 32 = 65,536$  different colours

How many different colours can you make from 8-bits (**3 red bits, 3 green bits and 2 blue bits**)? Show your calculation below:

3 bits can represent 8 different numbers  
2 bits can represent 4 different numbers  
 $8 \times 8 \times 4 = 256$  different colours

How many different colours can you make from 4-bits (**1 red bit, 2 green bits and 1 blue bit**)? Show your calculation below:

1 bit can represent 2 different numbers  
2 bits can represent 4 different numbers  
~~the~~  $2 \times 4 \times 2 = 16$  different ~~are~~ colours

Now that you have completed this investigation, describe in your own words how a computer would represent all the colours that make up a digital image and what happens when you have fewer bits to represent colours.



Digital images are made up of millions of pixels that can each have a ~~are~~ colour value. So the computer represents an image by storing the red, green and blue values for each pixel in the image. Each tiny pixel shows a different amount of red, green and blue light to mix different colours, like a colour mixer of light. When you have fewer bits there aren't as many different colours to display so the image quality isn't as good. It becomes blotchy.



PROGRESS OUTCOME 5

# Mixing colours

## Annotation

Pita is able to apply his knowledge about the digital representation of colours to develop a computer program that includes a calculation for a 24-bit colour with red, green, and blue values. He can use compound mathematical operations in a computer program and variables to represent and store data, including the results of calculations.

His program uses:

- sequence (steps in the correct order)
- inputs (RGB colour values, pen size)
- outputs (colour, pen-size, and text displays)
- selection through conditional logic (if-then-else)
- iteration (a forever loop).

He has also supported his program with comments that explain his code and calculations.

## Background

The students in Pita's group have been learning programming in a block-based programming environment. They have developed, tested, and debugged several programs that use the programming constructs required for the task below (sequence, selection, iteration, inputs/ outputs, conditional logic, and compound mathematical calculations). They have also developed a basic paint program that uses set colour choices.

In addition, the students have been learning about binary representation for colours using 24 bits (8 bits for red, 8 bits for green, and 8 bits for blue) and how colours are displayed on a screen using pixels. They have applied the calculation used by computers to determine colours.

## Task

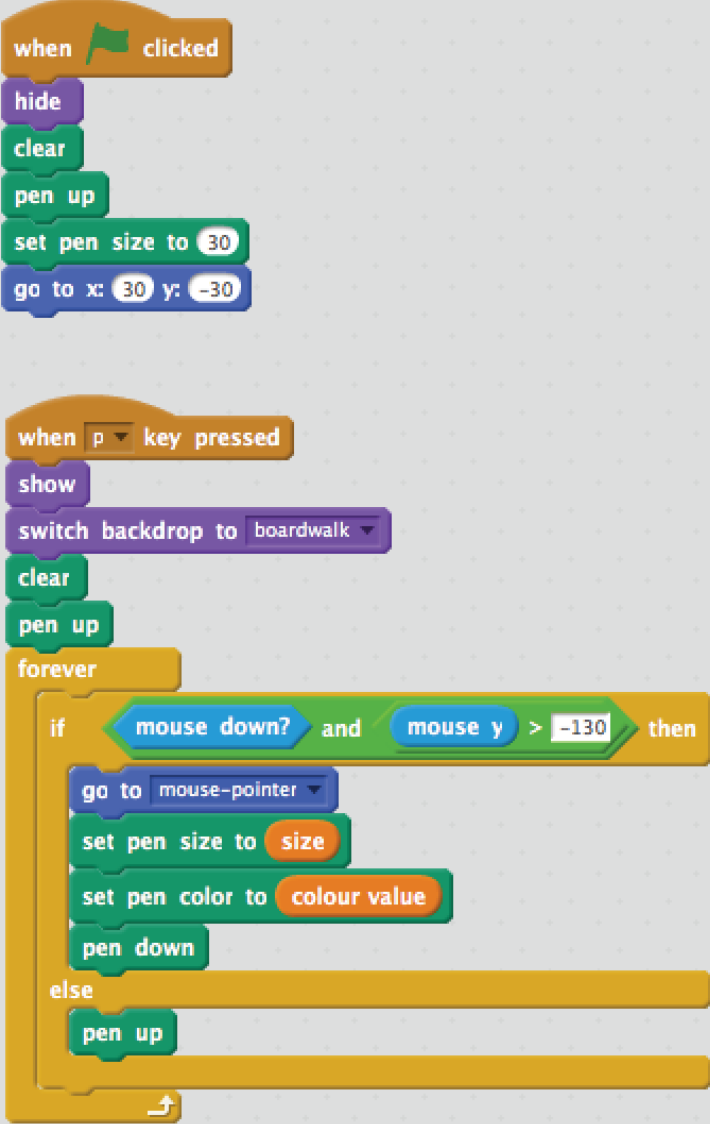
The students are asked to refine their basic paint program so that the user can create custom colours to paint with on the screen. The user should be able to input the RGB (red, green, and blue) values for a colour, and the program should then calculate the colour.

The program should include variables for red, green, and blue values (between 0 and 255), the colour value, and the size of the paint brush. It should also display a "swatch" of the custom colour that the user has mixed, to allow them to preview their colour.

**There are clear links to mathematics in the calculations that underpin colour values. The task is of particular interest to students studying photography and developing their ability to print a digital copy of an image that accurately captures the colours of the original.**

## Student response

Pita develops, tests, and debugs his program and adds relevant code comments. In the following code, he uses inputs from a user (pressing the P key; mouse clicks), sequence, and selection (if-then-else), and he demonstrates his understanding of the x and y coordinates of the mouse. He also uses conditional logic to determine whether the mouse is within the drawing area of the program's canvas.



The image shows a Scratch script with two main event triggers. The first is 'when green flag clicked', which performs a sequence of actions: hide, clear, pen up, set pen size to 30, and go to x: 30 y: -30. The second is 'when p key pressed', which performs a sequence: show, switch backdrop to boardwalk, clear, pen up, and then enters a 'forever' loop. Inside the loop, there is an 'if' statement: 'if mouse down? and mouse y > -130 then'. The 'then' block contains: go to mouse-pointer, set pen size to size, set pen color to colour value, and pen down. The 'else' block contains: pen up.

The computer constantly checks to see if the mouse is pressed and the mouse is in the painting area before drawing.

If that is true then the pen size changes to what the user has put into the variable and the colour changes to what the user has put for the r, g, b values.

In Pita's colour-swatch code below, he uses variables (to store data and the results of calculations) and iteration (a forever loop). He also applies compound mathematical operations within the program and includes code comments to document how his code works.

```
when clicked
hide
pen up
```

When the program starts (the green flag is clicked), the colour sprite is hidden and the pen is up so nothing will be drawn.

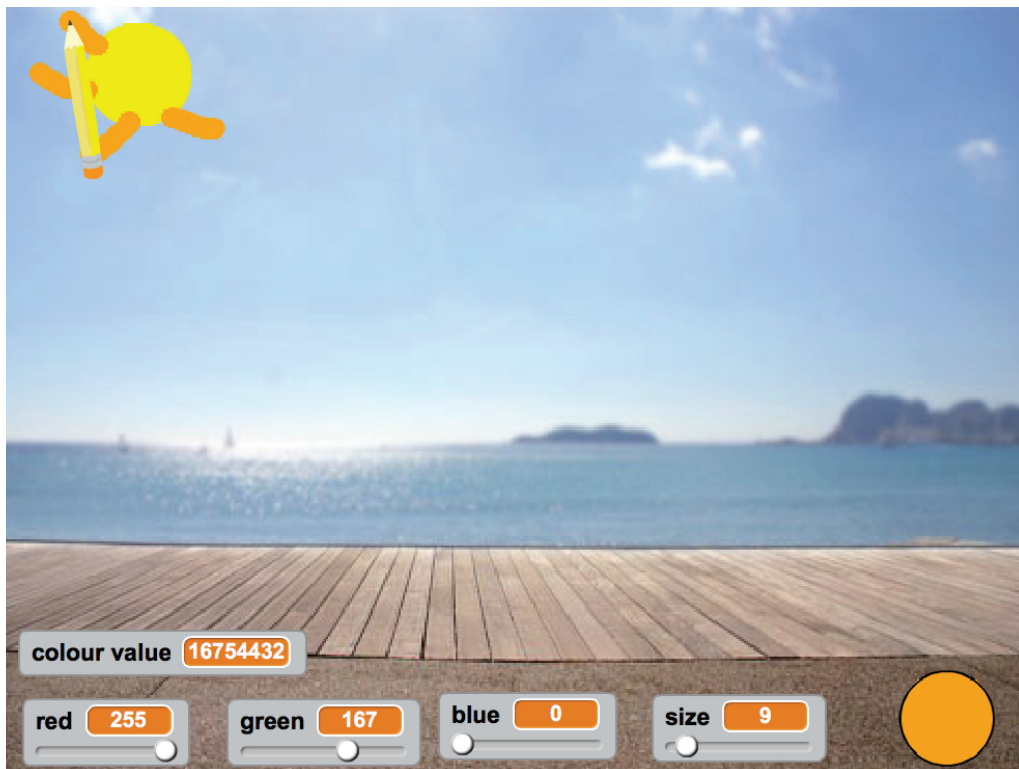
```
when space key pressed
show
set pen size to 43
forever
set colour value to 65536 * red + 256 * green + blue
set pen color to colour value
pen down
```

When the space key is pressed, the colour sprite is shown and the pen is down so it is ready to draw.

The colour value is constantly being calculated by finding the decimal value for the red, green and blue:

red value = red x 65536  
green value = green x 256  
blue value = blue x 1

These are added together to come up with the total colour value.



Downloaded from <http://technology.tki.org.nz>

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab (<http://scratch.mit.edu>).

Copyright © Ministry of Education 2017, except for Scratch images and images of deck and sea

The program used in this exemplar is not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-209-5



PROGRESS OUTCOME 5

# Pirate Game

## Annotation

Tua is able to use algorithmic thinking to develop two different algorithms to solve a specific problem. He uses logical reasoning to evaluate the relative efficiencies of the algorithms for program performance.

He demonstrates that he can operate within a text-based programming environment to create a program to implement an algorithm. His program reveals an understanding of:

- iteration (the step event repeats constantly)
- selection through conditional logic (if player\_coins <=0)
- global variables (player\_coins)
- derived values (room\_height and room\_width)
- random numbers
- functions and parameters (instance\_create).

Tua provides documentation for the program through the use of comments. He also demonstrates that he can apply iterative development and testing to identify a missing step in a program before moving on to the next part of the program.

## Background

The students in Tua's class are learning to program in a text-based programming environment. They have developed and tested several programs that use variables, functions, random numbers, and derived values. They have practised the decomposition strategy of breaking down a program into functional requirements, breaking each requirement down further into an algorithm, and then translating this into programming code. They have also explored iterative development by testing each part of their program before moving on to the next requirement.

## Task

The students are given a game called The Pirate Game in a text-based programming environment (GameMaker: Studio). The game has been deliberately "broken" by their teacher, Mr Walker - code is missing that is necessary for the game to run. The students play the game to determine its rules and what has been broken.

The students are asked to develop a set of functional requirements for the game that cover its objectives and how a player would interact with the game.

Next, they are asked to develop two algorithms for the missing code and to evaluate them in terms of their effect on the program's performance. They are required to program, test, and debug the algorithm that they decide is the most efficient in order to ensure the game functions as intended.

## Student response

After playing The Pirate Game, Tua develops the following set of functional requirements.

### Functional Requirements for the Pirate Game

- If the pirate touches the player it should destroy itself and subtract a coin.
- When all coins are lost the game should end.
- When a pirate reaches the edge of the screen it will respawn in a random location on the Y axis.
- Pirates will spawn in a random location once destroyed.
- WASD keys are used for player movement.
- Sound button turns music on and off (toggle).
- Player can choose a sprite.

He realises that the bug in the game is that the game never ends. This is because there is no program code to determine whether all the coins are lost and what should happen when they are. He develops two algorithms to fix the bug in the program.

### Algorithm 1 for testing coins

#### Pirate Object

Forever:

    If player coins less than or equal to zero:  
        End game  
    If pirate touches player:  
        Player coins decrease by one  
        Pirate object is destroyed

### Algorithm 2 for testing coins

#### Player Object

When player touches pirate:

    Player coins decrease by one

    If player coins less than or equal to zero:  
        End game

    pirate object is destroyed

Mr Walker and Tua discuss the relative efficiencies of Tua's algorithms.

**Mr Walker:** *Which algorithm would be better in terms of the use of computing resources and game play?*

**Tua:** *The first algorithm would make the game laggy because it is constantly testing the coins and collisions on every pirate object. The pirates respawn, so that would mean even more checking. I think my second algorithm is better because it focuses on the player collision event and there is only one player object. And it only checks the number of coins when the player collides with a pirate. This is better because the calculations and decisions happen only when they need to, not all the time.*

Tua implements his second algorithm by writing code in GML. When testing his code, he finds there is still a problem – pirates are not respawning in the game, so the game play cannot continue after a pirate collides with the player.

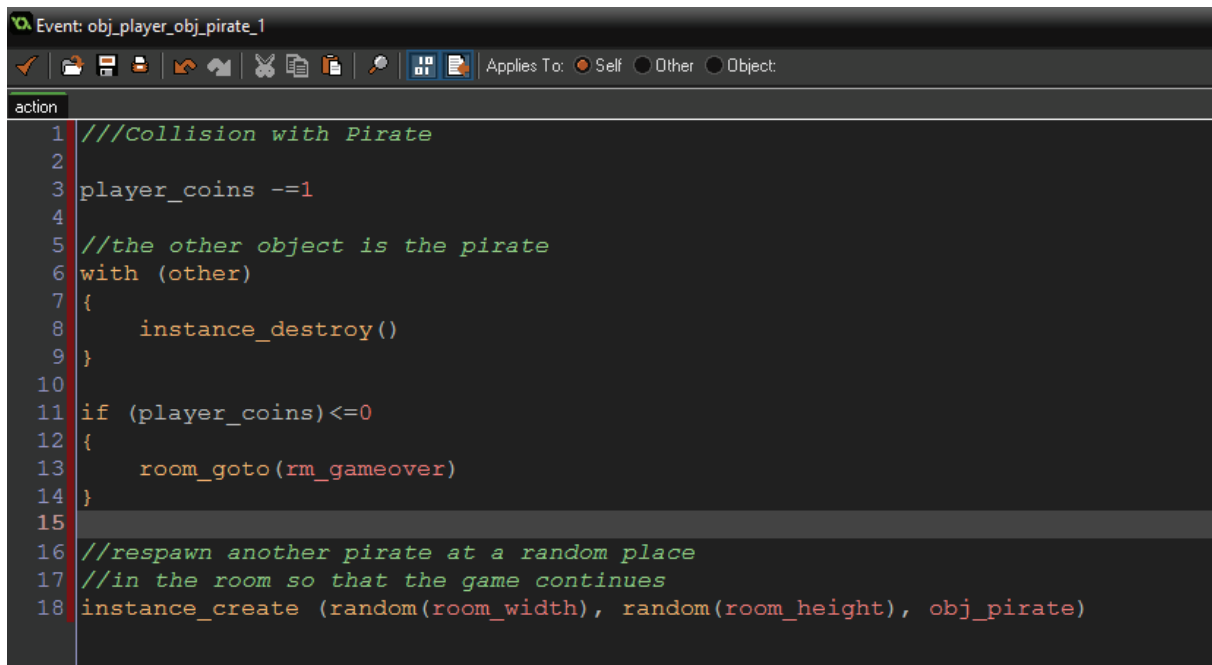
Mr Walker and Tua discuss the bug in his program.

**Tua:** *The game stops after one collision with a pirate. The pirate doesn't respawn.*

**Mr Walker:** *What should happen next to continue game play?*

**Tua:** *I need to make a pirate respawn in the room at a random location.*

Tua adds the function to create another pirate at a random location within the game screen. He then tests his final version to ensure that it allows the game play to continue.



```
Event: obj_player_obj_pirate_1
action
1  ///Collision with Pirate
2
3  player_coins -=1
4
5  //the other object is the pirate
6  with (other)
7  {
8      instance_destroy()
9  }
10
11 if (player_coins)<=0
12 {
13     room_goto (rm_gameover)
14 }
15
16 //respawn another pirate at a random place
17 //in the room so that the game continues
18 instance_create (random(room_width), random(room_height), obj_pirate)
```





PROGRESS OUTCOME 6

# Linear versus binary searches

## Context

Sharee has been investigating the impact of using different algorithms for searching and sorting data. She has experimented with two algorithms to determine which one works better for searching different data sets.



## Insight 1: Applications of searching

Most people don't realise how much they rely on computers use of fast and accurate search algorithms for all kinds of information. So I researched and brainstormed the different types of everyday computer searches. We take it for granted that a web search will provide instant results and that our groceries will scan quickly at the supermarket checkout. In medicine, doctors and researchers need to search databases of diseases or genetic information. The police need quick access to information about offenders and public safety. Because of the computer's ability to quickly search large amounts of data for answers to problems, artificial intelligence is developing rapidly.



## Insight 2: Different algorithms for searching data

During my research, I discovered many different search algorithms. Some are kept confidential, like the Google™ search algorithm, to maintain their competitive edge over other businesses. However, linear and binary searches are two basic search algorithms that most computer science students are familiar with.

To understand how they work, I tried each one with a friend. First, I printed out fifty random numbers and put them face down in random order. I then asked my friend to find a specific number using a linear search. That meant he had to start with the first number and look at each one in order until he found the number. We worked out that it could take fifty attempts to find the correct number, if it was the last one.

I then put the numbers in numerical order and asked him to find the number using a binary search by cutting the data set in half each time. He started in the middle and identified whether the specific number was higher or lower than the middle number and therefore which half of the data set it was in. We worked out that using this approach would take up to six attempts to find the number, because after cutting the data set in half six times you have only one piece of data left.



### Insight 3: Determining best, average and worst cases

After doing the experiment with my friend, I used a spreadsheet to look at the best, average, and worst cases when searching data using these two algorithms. I also researched this on the internet to see if my conclusion was correct.

I learnt that the worst case for a linear search is that you must look through every piece of data, whereas a binary search cuts the data in half each time, so at worst you have to look through the data as many times as you can cut it in half (i.e., until there's one item left).

On average, that means it's much quicker for a computer to perform a binary search. For example, if I have 1024 items of data to search through, it will take me at most 10 searches:

1)  $1024/2 = 512$

2)  $512/2 = 256$

3)  $256/2 = 128$

...

9)  $4/2 = 2$

10)  $2/2 = 1$

From my research, I learnt that this is a log base 2 function, so I used the log base 2 function in my spreadsheet to work out the worst case. I also read that the average and the worst case for a binary search are just about the same.

I tested this by downloading and running a binary search program 20 times for each data set and put the results into a spreadsheet. I saw that the average and worst cases were almost exactly the same. So I used the same spreadsheet formula for the average as for the worst case scenario (log base 2).

The table below shows my conclusions:

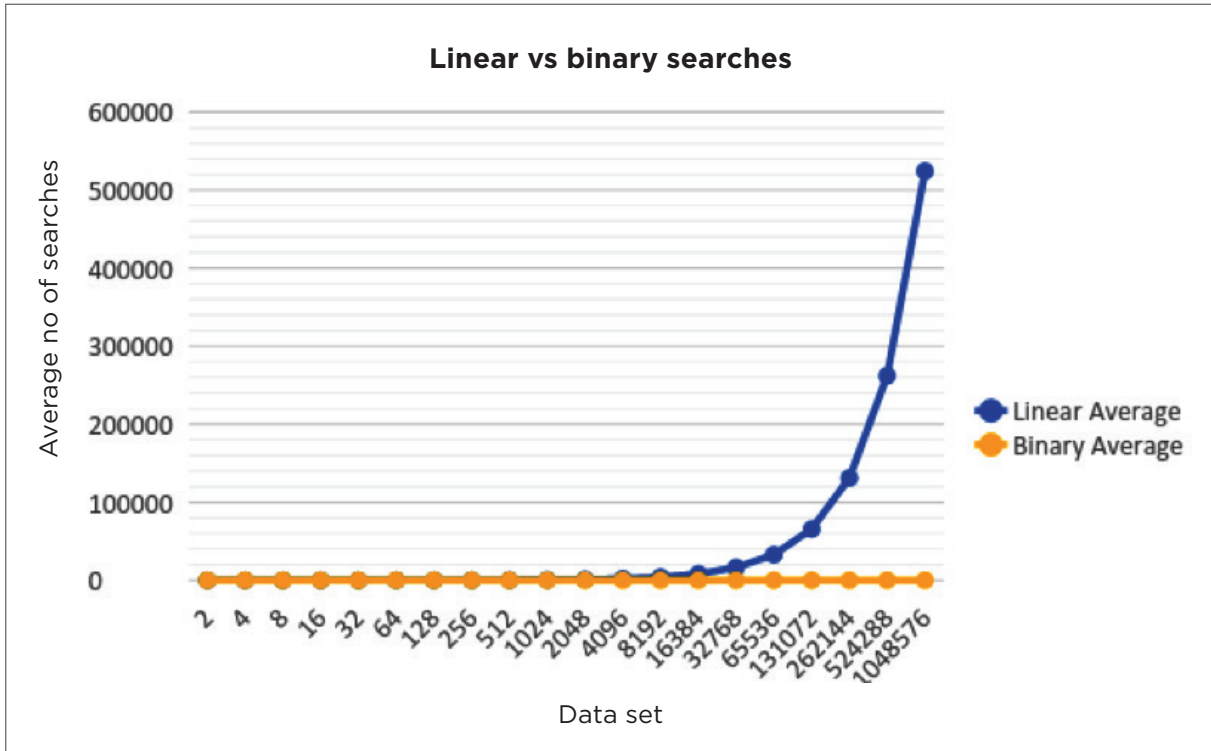
Linear Search			
Data set	Best	Worst	Average
2	1	1	1
4	1	3	2
8	1	7	4
16	1	15	8
32	1	31	16
64	1	63	32
128	1	127	64
256	1	255	128
512	1	511	256
1024	1	1023	512

Binary Search			
Data set	Best	Worst	Average
2	1	1	1
4	1	2	2
8	1	3	3
16	1	4	4
32	1	5	5
64	1	6	6
128	1	7	7
256	1	8	8
512	1	9	9
1024	1	10	10



#### Insight 4: Cost changes as problem size increases

I then added further data sets to the spreadsheet and made a graph comparing the two searches. I concluded that using a linear search with large amounts of data is much slower than a binary search because each time the data set doubles, the average number of searches for a linear search doubles, whereas it only adds one more search for a binary search.





PROGRESS OUTCOME 6

# Why compression matters

## Context

Sarah has been investigating the concept of compression coding, by looking at the different ways images can be compressed and the trade-offs of using different methods of compression in relation to the size and quality of images. She researches the topic by doing several web searches and produces a report with her findings.



### Insight 1: Reasons for compressing files

I realised that data compression is extremely useful as it reduces the amount of space needed to store files. Computers and storage devices have limited space, so using smaller files allows you to store more files. Compressed files also download more quickly, which saves time – and money, because you pay less for electricity and bandwidth. Compression is also important in terms of HCI (human-computer interaction), because if images or videos are not compressed they take longer to download and, rather than waiting, many users will move on to another website.



### Insight 2: Representing images in an uncompressed form

Most people don't realise that a computer image is made up of tiny dots of colour, called pixels (short for "picture elements"). Each pixel has components of red, green and blue light and is represented by a binary number. The more bits used to represent each component, the greater the depth of colour in your image. For example, if red is represented by 8 bits, green by 8 bits and blue by 8 bits, 16,777,216 different colours can be represented, as shown below:

8 bits corresponds to 256 possible numbers in binary

red x green x blue =  $256 \times 256 \times 256 = 16,777,216$  possible colour combinations



### Insight 3: Lossy compression and human perception

There are many different formats for file compression such as JPEG (image compression), MP3 (audio compression) and MPEG (audio and video compression). These formats use lossy compression, in which some of the original data is removed when the file is compressed. The benefit is a smaller file that downloads more quickly, but it does mean that you cannot convert the file back to its original full quality.

We use lossy compression because most image and sound files have more data than the human eye or ear can perceive and if data is removed it isn't usually noticeable. If we compress a file too much, people will start to notice - for example, an image may look pixelated or blurry. Sometimes it's not appropriate to compress an image because the quality is extremely important, such as in a medical scan.



### Insight 4: Lossless compression

Lossless compression compresses a file without removing any data. Instead a computer uses an algorithm to map the repeating patterns or information in the file (such as colours for an image or letters in a document). Most people use zip compression for general files. For images, lossless compression methods include PNG (Portable Network Graphics) and RLE (Run Length Encoding). Because lossless compression works by storing a map of repeating patterns of data, it is most effective when there is a lot of repeated data.

Below is an example of how lossless compression could work for a simple black and white image using RLE. The image on the left could be exactly replicated by a computer using the encoding map on the right. The computer has all the information it needs to recreate the image pixel by pixel (how many bits are black and how many bits are white).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	
1																								Characters for encoding				
2	1																							0	1	18	1	
3	2																							1	1	16	1	1
4	3																							2	1	14	1	2
5	4																							3	1	12	1	3
6	5																							4	1	10	1	4
7	6																							5	1	8	1	5
8	7																							6	1	6	1	6
9	8																							7	1	4	1	7
10	9																							8	1	2	1	8
11	10																							9	2	0		
12	11																							9	2	0		
13	12																							8	1	2	1	8
14	13																							7	1	4	1	7
15	14																							6	1	6	1	6
16	15																							5	1	8	1	5
17	16																							4	1	10	1	4
18	17																							3	1	12	1	3
19	18																							2	1	14	1	2
20	19																							1	1	16	1	1
21	20																							0	1	18	1	



PROGRESS OUTCOME 6

# Designing a virtual golf game

## Context

Physical education teacher Ms Henare has asked her year 11 students to create a computer game to help students work on a particular sports skill.

In consultation with Ms Henare and some year 9 students, Leilani has decided that a virtual golf game will be fun for students to use and help them with their golf swing.



## Insight 1: Planning

I decided to use Scratch to develop the game. First I sketched out a plan using a flowchart, which outlined the problem and the specific tasks the program should perform. My aim was to design a game that is engaging, performs the tasks, functions as intended and is easy for the user to play.

My plan included creating a pseudocode that sets out the algorithm, breaking down the tasks into their necessary steps and identifying the sprites (characters) and components I would need to create for the game.



## Insight 2: Setting out and commenting on the program code

I discussed my pseudocode with Ms Henare and began to develop the images, sprites and program code, as per my flowchart.

I had to plan how to make the game become increasingly more difficult (for example, by adding different levels). I also wanted to use a timer and keep a score (which would be a variable). I thought about the blocks I could use, such as actions to control the sprite movement (hitting the arrow keys and spacebar), and the variable for keeping the score. I sketched out my ideas and the bits of code and discussed these with Ms Henare, who was happy with my progress.

Next I started to develop the program. I annotated my code as I went, using the comment feature as I worked through the steps – the comments explain my design and implementation decisions. I also identified my variables and their names. This is a good way of reminding myself what the code is supposed to do (its function and behaviour). It's also useful if you have to explain the code to another coder and your reasons for programming it that way.



### Insight 3: Testing and debugging the program

I worked through my blocks, testing the project to see if it ran as it should. I found a few errors, such as when you clicked the green flag, an animation of a golf club should have swung back and forth once, then stop. But it didn't stop – instead, the animation started all over again. I had to work through the blocks to figure out why this happened. I fixed the error and added a comment to the block for future reference.



### Insight 4: Meeting end-user requirements

I asked a group of year 9 students to test the game for me. After discussing their feedback, we decided I needed to add another level for the more sophisticated golfer, as the game was too easy for some users. One error they found was that the “# of hits” display increased by more than 1 each time the golf club hit the ball. I had to work through my blocks to figure out how to fix this.



### Insight 5: Testing the final outcome

After I'd finished my debugging, I asked Ms Henare and a group of year 9 students to test the game again. It functioned as intended, and they were happy with the outcome.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/>

[Technology/Digital-technologies](#)

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab (<http://scratch.mit.edu>).

Copyright © Ministry of Education 2018, except for student work copyright © student  
The program used in this exemplar is not officially endorsed by the Ministry of Education.  
ISBN: 978-1-77669-238-5



PROGRESS OUTCOME 7

# Solving unsolvable problems

## Context

The students have been investigating possible answers to the question “How could you solve an unsolvable problem?” An algorithm is a set of instructions for solving a problem, so students were asked to consider whether people could solve all their problems using computer-based algorithms.

Kalani has decided to investigate how well algorithms can solve the complex problem of efficient international parcel delivery.



## Insight 1: Complex problems

I did some research and discovered that because of their complexity, some problems are intractable. In an ideal world, computers would always be able to use an algorithm to find the best solution to any problem within moments. However, in reality many problems are so complex that there are billions of possible solutions and a computer cannot solve these problems in a useful amount of time.

From my research, I determined that the complexity of a problem is often described using the Big O notation. This describes the performance of the algorithm in the worst-case scenario by using an expression of  $n$ , where  $n$  can be the number of objects within a problem or the number of steps a computer must take to solve the problem.



## Insight 2: Problem-solving mechanisms

Computers do not approach problems in the same way as people. Because a computer doesn't have the ability to be 'spatially aware', it cannot approximate a solution in the same way we can. To solve a problem, a computer has to 'brute-force' its way through the problem by calculating every possible candidate for a solution.

'Brute-force searching' a problem will deliver the best solution, but it usually takes impractical amounts of time to do so. For example, in an algorithm set up to efficiently manage an international delivery system, for every new city added to the list of delivery addresses, the number of solutions increases by another order of magnitude, as there are  $n!$  ( $1 \times 2 \times 3 \times \dots \times n$ ) solutions for a problem of size  $n$  (where  $n$  = the number of addresses in the system).





### Insight 3: Application and impacts

I decided to see whether algorithms could improve efficiencies in the transport industry and have a positive impact by reducing emissions. For example, logistics and distribution companies regularly need to improve the efficiency of the algorithms they use for their deliveries. A multinational courier company delivering hundreds of thousands of mail items every day must manage its fleet of courier vans and aeroplanes so mail is delivered as quickly and cheaply as possible to the correct addresses. To do this, the routes the couriers take must be organised to minimise the total distance travelled.

The less distance a courier travels, the lower the cost of fuel and wages and the more profit the company will make from its deliveries. An added benefit is that fewer fuel emissions will cause less harm to the environment. Any cost savings will compound over time into major increases in profit, putting the company ahead of its competitors. However, the courier company can't wait for a computer to take months or years to find the best route. At the same time, relying on human interpretation to find the best route is unlikely to be as beneficial as using a computer.

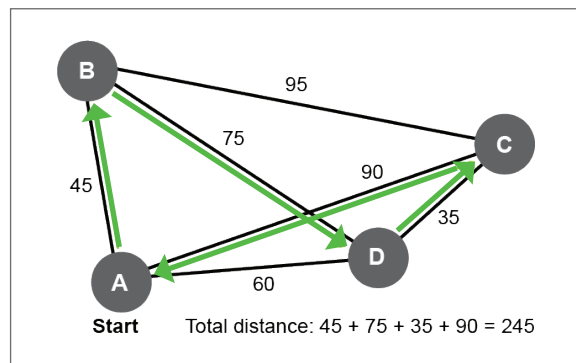


### Insight 4: Solving the unsolvable

I researched this concept and learnt about heuristics, which in computer science are techniques designed to help computers solve problems efficiently and quickly. The “nearest neighbour” heuristic is commonly used to find a solution to complex problems. Nearest neighbour is a “greedy” algorithm, where the algorithm attempts to make the best possible choice at each stage, rather than looking for an overall best solution.

In the example below, a company has to send a courier from the depot (A) to three different houses (B, C and D) to deliver parcels and then return to the depot. Using the nearest neighbour heuristic, the total distance travelled is 245 units. This route appears to be efficient, but we don't know whether this is the optimal route because the other routes have not been calculated.

The nearest neighbour heuristic is one of the fastest algorithms for solving intractable problems like this, because at each step it simply calculates the distance from a point to all other points and chooses the point the minimum distance away.



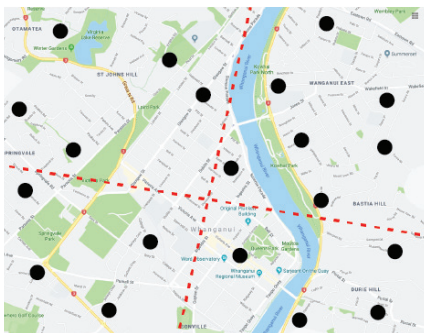


## Insight 5: The wider implications

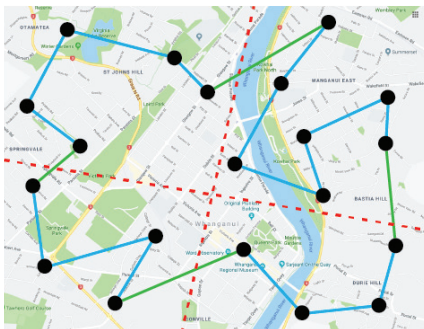
In real-world cases, typically more than one heuristic is used on a problem to ensure the best possible solution is found within a short amount of time.



For example, a company needs to deliver to the 20 addresses shown on this map. If it were to brute-force this problem, approximately  $2.4 \times 10^{18}$  obtained solutions would need to be calculated. This would take an impractical amount of time, and so the company needs to use a better way to find a solution.



In this image, the company has used the divide and conquer heuristic to divide the 20 addresses into smaller problems of more manageable size. The company can then brute-force these individual sections and put them back together. However, with a larger problem, brute-forcing could again run into time issues. In that case, the company could use other heuristics to find an efficient solution for each small sub-problem.



Once all the small solutions are joined together, the company has a final solution to its initial problem. This efficient and optimal problem-solving approach enables the company to deliver their parcels more quickly, decreasing their running costs and increasing profits.

This method of problem solving has implications for many kinds of organisations. It helps them reduce costs and solve 'unsolvable' problems, as well as benefitting the environment.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/>  
[Technology/Digital-technologies](#)

Google Maps™ is a registered trademark of Google Inc., used with permission.  
Copyright © Ministry of Education 2018, except for student work copyright © student  
ISBN: 978-1-77669-239-2



PROGRESS OUTCOME 7

# Looking at cybersecurity

## Context

Hamiora has been investigating the concept of encryption and its impact on society. He has learnt that encryption is an important aspect of computer science as it impacts upon all our online activities, including online shopping, banking and financial transactions.



### Insight 1: The concept of encryption

Most people take cybersecurity for granted, yet few understand exactly how encryption works. I discovered that encryption is the conversion of data into a form that is unreadable to anyone except the intended recipients. This is important in our society because we all have private information that we don't want the world to see. This includes the things we expect to remain private, like our online searches and the messages we send our friends, but also the things we expect to remain secure, like our banking information when we use internet banking or the credentials for our Wi-Fi networks. Unfortunately, there are malicious people who break into individuals' systems for personal gain. Encryption helps protect against this.



### Insight 2: An example of encryption

A simple example of encryption is an XOR cipher used on text, which is stored as binary using the ASCII encoding standard. This relies on a constant binary 'key' that is applied to the input. For simplicity, the binary key is exactly 8 bits long (the length of a byte).

The example below uses the string "Testing testing" (but in binary) as the input and the binary 00010111 as the key.

Input string:	Testing testing
Key:	00010111
Output of XOR cipher (gibberish):	Crdc-yp7crdc-yp

The XOR cipher works by applying the XOR binary operator to each bit of each byte. The binary operator takes two bits and returns true (represented as 1) if the two inputs are different, or false (represented as 0) if the two inputs are the same. One of the two bits is from the key, while the other is from the input string.



### Insight 3: Key problems and issues

A major problem with using an XOR cipher on text is that it becomes vulnerable to frequency analysis. This is a technique that uses the known frequency of each letter in a language to guess which letter corresponds to each encrypted character. For example, since letters like e and t are far more common than z and q in the English language, it's highly probable that two of the characters that appear most commonly in an encrypted message are e or t. Using this information, a hacker can guess the code and decrypt the message, thus "cracking" the cipher.



### Insight 4: The key distribution problem

Two forms of encryption ensure that only the intended recipient can read a message: symmetric cryptography and asymmetric cryptography. Symmetric cryptography relies on the sender and receiver sharing or exchanging a key securely. Most real-world systems use asymmetric cryptography, or public-key cryptography, which uses two keys instead of one. The first key (the "public key") encrypts a message and can be shared widely. The other key (the private key) decrypts the message and is only known to the recipient of the message.

With symmetric cryptography, the one key must be kept absolutely secret between the two parties. However, in asymmetric cryptography the public key used to encrypt messages can be shared widely without fear of it being used to decrypt messages, because decryption requires the private key.



### Insight 5: How asymmetric cryptography is used

A popular form of asymmetric cryptography is RSA (named after its inventors: Rivest, Shamir and Adleman), which uses a public key and a private key. RSA is used on the internet to securely establish a connection between computers and websites. As RSA uses asymmetric cryptography, the website and the computer can send their public keys to each other, without having to worry about communicating secretly.

RSA encryption is used in many everyday internet tasks, including internet banking, checking email and browsing online. It uses what is known as a 'trapdoor calculation', where the key is created by multiplying two very large prime numbers. To crack the key, you would need to factorise the resulting number, which is an intractable problem that would take many years to solve.



PROGRESS OUTCOME 7

# Ordering smoothies

## Context

Jordan's class has been learning how to develop a computer program that uses a GUI (graphical user interface) and responds to users' inputs from button clicks, text input, or selections from drop-down menus or radio buttons.

The students have been given a scenario of developing an app for ordering smoothies from the school canteen. They have had to think about the design of the GUI and the program, and the testing needed to ensure the program functions accurately.



## Insight 1: Designing the GUI

We asked the school canteen about the process when students and teachers order smoothies. They gave us the following information about smoothie choices:

- three dairy choices (customers can only pick one)
- three size choices (customers can only pick one)
- six mixture choices (customers can only pick one).

I decided that the GUI would need three radio buttons for the dairy choices, three radio buttons for size choices and six radio buttons for the mixture. There would also need to be a button for clearing the order, a button to submit the order and a text box to display the order on the screen. I sketched out the layout and realised that there was no place to enter a name for the order, so I added another text box for input.



## Insight 2: Decisions about variables and collections

Looking at my design and thinking about how the program should function, I decided that I would need integer variables to store the choices for dairy, size, and mixture, and a string variable to store the name. I decided to store the different options for each choice in arrays so it would be easy to build the order. I could have built the order by hard-coding the output with a series of switch statements, but I thought this would take longer to code, which could introduce more errors and would be harder to update if the choices changed. Using arrays, I could loop through them to find the choices and build the order.



### Insight 3: Documenting and testing

After I had used a form design tool to build my GUI and initialised my variables and collections, I decided to program and test each event to make sure it worked as I expected. As each event should work in a similar way, I thought it would be more efficient to have one functioning correctly before programming the rest. I made sure all my variable names were clear and used the camelCase convention we had been taught in class. I then wrote some pseudocode for the first radio button click event and used that for my code comments.

I began testing by selecting a dairy choice and checking whether the correct choice displayed in the order output on the screen. At first this didn't work as I expected, but then I remembered that arrays are 0-indexed and I had set my choices to start with 1. Finding this out early on enabled me to get the codes to work correctly for the other radio button click events.



### Insight 4: Testing and debugging the order output

My order output had the correct information, but it wasn't readable because all the choices were joined together, such as in "Charlietrimlargebananarama". I had used string concatenation to build the output, but I had forgotten to add in carriage returns and newline commands. I added these commands and tested that the output was easier to read.



PROGRESS OUTCOME 8

# Can algorithms be biased?

**Context**

Erana has been working on a project in her digital technologies class. Through it she has developed a strong interest in programming and algorithms.

 **Insight 1: A digital concept in practice**

This year I began researching the concept of the social impacts of algorithms after reading articles about how algorithms affect human behaviour. Headlines included “Algorithmic Discrimination”, “Algorithms for Design” and “Algorithmic Personality Detection”. Just reading the headlines was disturbing, so I wanted to find out more. I decided to investigate policing, as it would be scary if computers started policing ... or would it?

 **Insight 2: Heat-map algorithms**

I did some research into how ‘heat-map’ algorithms work. In policing, a heat-map algorithm tells the police where to look for criminals, based on the locations in which people have been arrested in the past. The police claim it helps allocate resources more effectively. But I discovered that data is only useful for predictive processing if you are working with a large sample or big data.

The data that police collect is like a tally: they count all the arrests they’ve made and display the most common locations. While most algorithms might appear to be just programs, some are designed to find trends. This is how an algorithm that identifies dominant trends could be discriminatory.

To test this, I conducted some experiments on a small scale. I looked at Google™ predictive searching and how it works. I then wrote a simple program that searched for common words to show how this might work on a large scale.



### Insight 3: The key problem

After I tested my simple program to show how data can be gathered and analysed, I looked at how this could be improved. I examined the concept of “sample size disparity” and discovered that if you want to make a prediction, you need to be careful about how your algorithm treats outlying data.

For example, if a disproportionate number of people in a specific ethnic group behave in a particular way, an algorithm may connect all people from that group with that behaviour. This is like saying everyone from the “rough side of town” is a criminal.

To test this, I built a data set of different ethnicities and included their street addresses and their favourite fruit. I put in a disproportionate number of English people liking bananas, and when I created a simple query, my program predicted that English people ate bananas.



### Insight 4: Applying algorithmic predictions in society

I wanted to use the results from my research and experiments to find out the wider applications. As a society, we want to reduce crime and better resource police to deal with crime. It would be great if we could predict where crimes are likely to occur, but I found that algorithms are not unbiased lines of code – an algorithm designed to pick up statistical patterns in data will report back what it is exposed to.

This means that if data includes a social bias against a minority, an algorithm is likely to incorporate this bias. This can lead to unfair or bad decisions by police or the government for members of the minority.



### Insight 5: Critical evaluation of the concept and its future use

I was amazed at what I found out from my project. Nobody can agree on how an algorithm can be ‘fair’. I thought algorithms were immune from bias, but they may not be because they try to make meaning of data and will predict outcomes as they process it.

There are many problems associated with this and no easy solutions, particularly because it’s difficult to know how some algorithms are used. Many companies keep their algorithms secret in case they are accused of being biased or to prevent their algorithm being stolen. I think that algorithms need to be carefully developed to consider the range of social conditions that may influence their predictions.

We all want less crime and to be able to prevent crime, but we don’t want to unfairly target specific groups. As computer science progresses, I think we will continue to use algorithms in fighting crime. If we keep checking the results and keep the public informed, the algorithms will become more refined.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/Technology/Digital-technologies>

Google™ is a registered trademark of Google Inc., used with permission.

Copyright © Ministry of Education 2018, except for student work copyright © student  
ISBN: 978-1-77669-242-2







PROGRESS OUTCOME 8

# Communicating online successfully

## Context

Ravi's digital technologies class has been learning about the use of data on the internet at home and school. He has developed a real interest in how data is organised, manipulated and used in computers.



### Insight 1: Key concepts

A key aspect of society today is communication. Even the most basic (short distance) and complex (long distance) aspects of communication can be done using technology. People rely on technology for communication and expect it to function well, so they don't tend to think about the mechanisms that enable this.

I started researching the concept of network protocols, as I like using the internet and wanted to find out more. Transferring files around the world is a complex and difficult task, so it's not surprising that data can be corrupted, delayed or lost in transit. To address this problem, transport protocols are created to send intact data from one location to another.

On a conceptual level, a protocol is an established process for accomplishing a task. When applied to networking, a protocol is a formal set of rules that ensures a successful communication between two parties. The presence of protocols across a network enables data to be transferred efficiently and means that users don't have to worry about the reliability of online communication.



### Insight 2: How transport protocols work

I posed the question "Can a protocol ever reach 100 percent efficiency?" To test different transport protocols and their efficiency in transferring data across a network, I conducted an experiment with Skype. Skype uses the protocols TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). By carrying out a long-distance video call and chat (messages) with my family back in Sri Lanka, I could see the application of these two different transport protocols and their results.

Skype's use of two protocols means that data being sent from New Zealand to Sri Lanka can travel through different routers. When data is sent, large amounts of data are broken up into transportable segments called packets, which later combine to form the original set of data. Data packets can be traced by an application called Wireshark. Using this application allowed me to see the packets that I was sending through each router and enabled me to identify one of the causes of the fluctuating quality in our Skype calls.



### Insight 3: Network protocols in today's world

TCP has built-in calculating algorithms so that it can obtain the most suitable window size for efficient transport of a file. This size is chosen once the first couple of packets are received so an efficient data transfer rate is achieved early in the transfer. This mechanism also helps to control congestion by constantly adapting to the fluctuating network. Due to all these safety mechanisms, TCP is known for prioritising accurate data transmission over speed, while UDP prioritises speed over accuracy.

Our school network is large due to our BYOD policy. It uses both TCP and UDP transport protocols for several applications run by the school. The centre of our school's online learning experience is our e-learning website. It contains subject resources, external links and daily updates on school notices. It's accessible in and out of school and is linked to the school database from which it retrieves information.

The website, however, is run by an external company, so the servers for e-learning aren't located in the school. The company uses TCP to manage the website because all the information must be displayed accurately. Because data needs to be received intact and doesn't require simultaneous delivery, TCP is ideal.



### Insight 4: How network protocols affect me and others

My family in Sri Lanka and I often experience poor video reception when talking to one another on Skype. I think this is because bandwidth capacity is much smaller in Sri Lanka than in New Zealand. Using Wireshark, I could see statistics for the data packets sent and received – the amount of data being sent was much greater than the amount I was receiving. I concluded that some of the data was getting lost due to poor bandwidth and not being recovered due to UDP protocols.

The expression “You can't build a strong building on a weak foundation” applies to network protocols. I think Sri Lanka's network layer with its poor bandwidth is a weak foundation. It's almost impossible for a protocol to be 100 percent efficient when built on top of this layer. The most an efficient protocol can do is reduce the impact of the weak foundation.



### Insight 5: Wider impacts and scenarios

UDP has speed but lacks reliability; TCP has reliability but lacks speed. For our school's e-learning website, it's clear that TCP is better; for video calls, UDP is better. Both protocols have their pros and cons. When used together, they can create reliable and speedy data transfers.

Skype uses TCP for instant messaging to establish trustworthy communication and UDP in video and voice calls for speed and synchronisation. As a result, Skype incorporates the advantages of both protocols, maximising their usefulness and helping to make Skype one of the most popular long-distance communication applications.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/>

[Technology/Digital-technologies](#)

Skype® is a registered trademark of Microsoft Corporation.

Wireshark® is a registered trademark of Wireshark Foundation.

Copyright © Ministry of Education 2018, except for student work copyright © student

The programs used in this snapshot are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-243-9





PROGRESS OUTCOME 8

# An Agile approach to game design

## Context

Lily's class has been working on a collaborative project that involves using a software engineering approach to design a game in Unity® software. She was learning a lot from developing the game and wanted to learn more about the ways to make good software.



### Insight 1: Key aspects in developing software

I did some research into software engineering to understand why it is so important. I learnt that for a software project to be successful, it must be delivered on time, within budget and with few or no errors. Failure in any one of these three requirements can cause failures in the others and can compound into the overall failure of the project. Meeting these three requirements is often difficult in reality, due to the complexity and sheer size of many projects. When a project fails, it can be very costly, and not just in monetary terms.

For example, NASA's Mars Climate Orbiter was sent into space in 1998 to orbit Mars and observe its climate. Before the satellite managed to achieve a stable orbit, it burnt up in the Martian atmosphere. On review, it was discovered that within the software controlling the satellite's thrusters, the force required to be exerted had been calculated using English pounds rather than the industry-standard newtons.



### Insight 2: Wider implications of the NASA debacle

The cost of the Climate Orbiter's software failure was far greater than the \$125 million dollars it cost to build. Although NASA had a working blueprint for the satellite, producing the satellite a second time with the error corrected would be pointless, as technology had developed considerably since the initial development.

Therefore, development of a new satellite would have had to begin from scratch, which would take many years and a lot more money. Also, a costly failure such as the Mars Climate Orbiter can cause a government to lose confidence, cut funding and reduce support for similar projects in the future.



### Insight 3: Comparing different project methodologies

I wanted my game development to be successful, so I decided to take on board what I had learnt from the software disasters I had researched and find a good strategy for my game design project.

Projects are usually described as a series of stages, such as analysis, design, and implementation. I was currently using a basic Waterfall process to develop outcomes – this is a linear, sequential method of working through a project with five stages: analysis, design, implementation, testing and release. Each stage is visited once and completed before moving on to the next stage.

I looked at the Agile methodology and compared its more flexible, iterative approach with the Waterfall process.



### Insight 4: Using the Agile approach

I decided to use an iterative Agile approach with my team to deliver the game. We wanted to make sure we delivered the project on time and within cost (we agreed that the ‘cost’ would be the impact of the project on our other subjects). It was also important that the game would be playable, that it would work as intended without any major errors.

We decided to focus on short iterations – rather than conducting the whole project from start to finish, we would make small features in short periods of time. In this way, the results of each iteration could be used to adjust the overall project plan. Each iteration was like a short project in itself, which was awesome, as it meant we could work together but be assessed separately. It also allowed us to adjust goals and measure progress regularly.



### Insight 5: The Scrum process

We had a large team on this project – nearly half the class! We decided to use a Scrum process and set up team responsibilities accordingly. I was the Product Owner – my role was to prioritise and define the features of the project. We also had a Scrum Master, who had to ensure that the team was fully functional and productive and followed our agreed process. We had a large Development Team, which selected the sprint backlog, organised themselves and demonstrated work results to me as the product owner.

As we worked through the process, we found out which members of the team could handle particular jobs and so we were able to better allocate tasks. Also we made sure we were always working towards a realistic goal.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/Technology/Digital-technologies>

Unity® is a registered trademark of Unity Technologies.

Scrum was developed by Ken Schwaber and Jeff Sutherland.

Copyright © Ministry of Education 2018, except for student work copyright © student

The program and processes used in this snapshot are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-244-6



PROGRESS OUTCOME 1

# Vacuum cleaners

## Annotation

Leo and Malachi's responses show that they:

- can identify digital devices and explain what makes them digital
- know that digital devices are created by people for particular purposes
- have some understanding of the role digital devices play in our everyday lives.

## Background

The students have been learning to identify digital devices that they use in their everyday lives. They understand that all these digital devices have been designed and made by humans for a particular purpose.

## Task

As a starter, students play a memory game in pairs. They match the digital version of a device with a version that is not digital. For example, they pair an email with a letter in an envelope, and a smartphone with a landline phone.

Following the game, the class discusses the difference it makes when a device is digital. They also identify other digital devices they can think of.

In pairs, students choose a pair of devices, digital and non-digital, either from the memory cards or of their own choice. They are asked to explain the differences between the devices, why one is digital and the other is not, and what the impact of these differences has on everyday lives.

Students can take photos of devices, draw them, or find images on the Internet.

## Student response

Leo and Malachi work with Ms Wilson to find images of a vacuum cleaner and a robot vacuum cleaner. They discuss the differences between the digital and non-digital devices and present their findings to the class.



**Leo:** *This vacuum cleaner doesn't have a memory. You can't make it do anything different, or work on its own. When you use it you can't be anywhere else. It makes your back hurt.*



**Malachi:** *The robot vacuum cleaner follows the instructions it's been given. These instructions are stored in its memory and can be changed to do something different. You can go away and do other jobs or relax.*



PROGRESS OUTCOME 1

# Turning on the TV

## Annotation

Lindy shows that she:

- can identify the inputs and outputs of a system
- understands that when we give a digital device an input, the device processes it and produces an output.

## Background

Lindy has been learning that digital devices receive inputs and produce outputs.

She has previously learnt that if she wants to use an image from the Internet, she needs to make sure the person who created the image allows other people to use it without charge.

## Task

The students are given a brief to create a flowchart that shows the sequence:

- giving input to a digital device
- the device processing the input
- the output of the device.

They work with Mr Hughes to each select an image of a digital device to work with, and find suitable images of an input, process and output for the device.

They then import the images into the Book Creator application on an iPad. They use the app to make their flowchart and briefly describe what is happening at each stage.

**The task gives the students the opportunity to practise creating meaningful texts, using a particular layout and visual language features to communicate their understanding.**

## Student response

### TV flowchart



We use the remote.



The TV turns on.



We can watch the TV.

**Mr Hughes:** *Can you tell me which bit of the television system is the input?*

**Lindy:** *The remote is the input – it's how I switch it on.*

**Mr Hughes:** *Why is the remote the input?*

**Lindy:** *Because it's how I tell the television what to do.*

**Mr Hughes:** *Can you think of another digital device that has a different kind of input?*

**Lindy:** *Our microwave at home has a keypad.*

**Mr Hughes:** *So now you've used the remote to switch the television on, could you use the remote in another way?*

**Lindy:** *I could turn the volume up or change the channel.*

**Mr Hughes:** *So which bit of your sequence is the output?*

**Lindy:** *It's what comes out of the television.*

Downloaded from <http://technology.tki.org.nz>

iPad is a trademark of Apple Inc., registered in the United States and other countries.

Book Creator is a trademark of Red Jumper Limited.

Images sourced from <https://pixabay.com/> and used under Creative Commons CCO.

Copyright © Ministry of Education 2017, except for TV flowchart photographs

The program and resource used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-212-5





PROGRESS OUTCOME 1

# Animating a song

## Annotation

Shelby, Kaia and Nīkau show that they can select and use appropriate applications to create, manipulate, retrieve and share digital content to meet a specified brief.

## Background

The students have been participating in a project on digital storytelling. As part of their inquiry, they have learnt how to use the I Can Animate *Lite* and iMovie applications.

## Task

The students are given a brief. This includes working collaboratively to design and develop a stop-motion video that uses claymation to portray “The Duck Song”, a popular song for young children. Over a series of sessions, they are required to:

- create the clay characters for the animation and make props for the setting
- take photos and import them into a video-editing program
- edit the video
- sing and record the song as an audio file to be imported into the video.

**The task is an opportunity for the students to practise creating visual texts and performing music, using their skills and imagination in ways that evoke a response in their audience.**

## Student response

**Mr Wong:** *What did you learn from watching other stop-motion videos before you made yours?*

**Shelby:** *We noticed that in one of the videos, a park bench had moved, and you could tell. So we made sure to keep things lined up. We used the tool in I Can Animate to see where the last picture was so we could line everything up.*

**Mr Wong:** *How did you work as a team to develop your stop-motion video?*

**Kaia:** *Shelby took the pictures, Nīkau moved the characters and I moved the light. Shelby sang the duck parts, Nīkau sang the shopkeeper parts and we all sang the chorus together. We all made the clay characters and the set.*

**Mr Wong:** *Did you change anything about your story as you created your video?*

**Nīkau:** *We had to cut it down a lot because the original song was three minutes long, and that would be too many pictures. But it still had to make sense. We cut out some of the song, and we still took 830 pictures!*

**Mr Wong:** *What apps did you use?*

**Shelby:** *We used I Can Animate to take all the pictures, and it helped us to line everything up. That's why we didn't use just the camera on the iPad. We put the pictures in iMovie so we could add our singing for the voice-over and the credits.*

**Mr Wong:** *You've been working on this project for over two weeks now – how were you able to find your work at the start of each session?*

**Kaia:** *We saved our work and called it "Duck Song" so we could find it again, and we made sure we used the same iPad each time.*



Downloaded from <http://technology.tki.org.nz>

I Can Animate *Lite* is a product of Kudlian Software Limited ([http://www.kudlian.net/kudlian\\_Software/Terms.html](http://www.kudlian.net/kudlian_Software/Terms.html)).

The Duck Song is copyright to Bryant Ogeden (<http://www.songdrops.com/>).

iMovie and iPad are trademarks of Apple Inc., registered in the United States and other countries.

Copyright © Ministry of Education 2017, except for images copyright © the students  
The programs and resources used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-213-2



PROGRESS OUTCOME 1

# Bottle-cap music

## Annotation

Priya and Angus show that they:

- can identify the inputs and outputs of a system
- understand that when we give a digital device an input, the device processes it and produces an output
- can use an application to create original content.

## Background

The students have been exploring how a Makey Makey® controller can act as an interface between a conductive material, digital devices and software. This has strengthened their understanding of inputs and outputs. It has also supported them to build on their science learning about circuits.

## Task

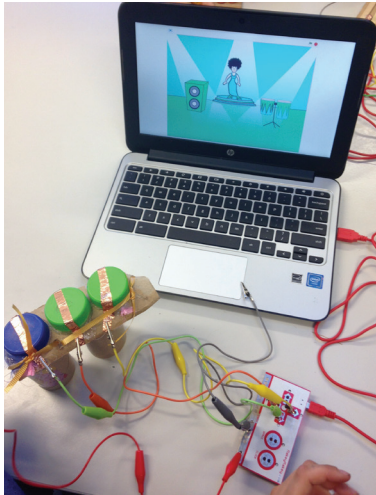
The students are given a brief to design and create a musical instrument out of everyday objects. They are to construct a Makey Makey controller circuit along with a program in Scratch Junior that will process the Makey Makey’s commands to produce a sound. The musical instrument will provide the input and the sound will be the output.

The students are shown how to use an input part of the Scratch program (an event block) and an output part of the program (a sound block) in Scratch – for example:



## Student response

Priya and Angus use cardboard and bottle caps to make their instrument, and they connect it to the computer through a Makey Makey controller. When the bottle caps are touched, they hear the corresponding sounds they have programmed in Scratch.



**Mrs Kendall:** *What are your inputs?*

**Angus:** *The bottle caps.*

**Mrs Kendall:** *What are your outputs?*

**Priya:** *The sounds.*

**Mrs Kendall:** *What is the digital device that turns your inputs into outputs?*

**Priya:** *The Makey Makey. But the Scratch program we put into the computer helps too.*

Downloaded from <http://technology.tki.org.nz>

Scratch and Scratch Junior are developed by the Lifelong Kindergarten Group at the MIT Media Lab (<http://scratch.mit.edu>).

Makey Makey® is a registered trademark of Makey Makey LLC.

Copyright © Ministry of Education 2017, except for scratch image

The program and resource used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-214-9



PROGRESS OUTCOME 2

# Video of a system

## Annotation

In writing their script and filming, editing and exporting their film, the students show that, when given criteria and tools, they:

- can independently make decisions about creating, manipulating, storing, retrieving and sharing content that has been developed for a specific purpose
- understand the particular roles of components in an input-process-output system and how they work together
- understand that inputs are transformed into outputs within a system and the “control” role that humans have in this
- are able to use a software application and a range of file types to meet a specific purpose.

## Background

The students have been taking multimedia classes once a week during their Special Interests time. In these classes, they have been working on script writing, filming with iPad mobile digital devices and digital cameras, and editing film using different software applications.

In their technology class, they are learning about computers and how digital devices contain a central processing unit (CPU) that carries out instructions from computer programs and processes data.

## Task

In groups of 4-5, students are asked to create a video that shows the role of components in an input-process-output system and how people can interact with these. They are able to choose the type of video they will make (e.g., filmed people with props, stop-motion), how they film it and how they edit it.

**The task gives the students the opportunity to practise working collaboratively to achieve a specific outcome.**

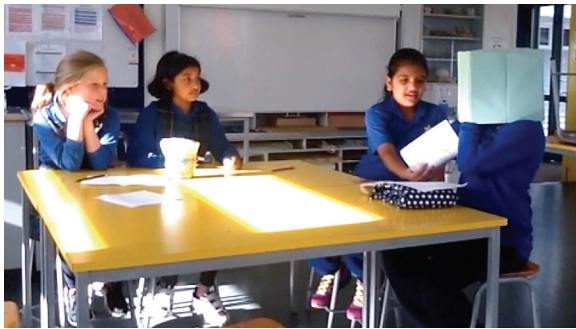
## Student response

Together a group of five students write a script for their video. Rhiannon, Ruby, Shakira and Zoe act out the script, and Ben films it using an iPad. They work together to edit their video clips and add sound effects using the application iMovie.

In the film, Shakira, Ruby, and Rhiannon give a computer (played by Shakira with a box on her head) inputs to process and receive outputs from it.



Ruby writes a message and says, "Give the computer this as an input." Rhiannon puts it into the "computer" by handing it to Computer-Shakira.



A dial-up Internet sound effect plays while Computer-Shakira takes the piece of paper and pushes it into the box.



Computer-Shakira says, "Scanning ... scanning ... processing ... processing. The message says, 'Ruby is cool.'"

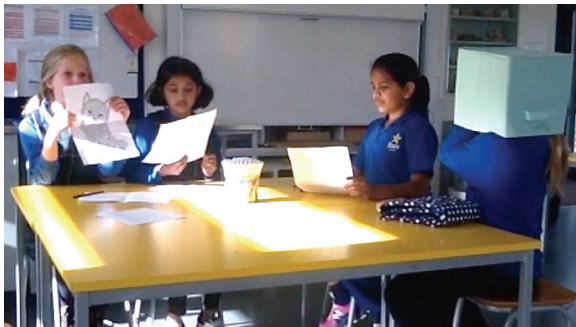
The students gasp in amazement.



Zoe asks Ruby to find some colouring-in pages online. She wants a picture of a cat. Ruby says, "I'll use the keyboard to input that. C-A-T, enter."



Computer-Shakira beeps and says, “Processing ... sending to output. Here is your picture of a cat. Have a nice day.” She pulls a piece of paper out and hands it to Zoe.



Ruby and Rhiannon also each search for a picture and receive an image from Computer-Shakira.

The project containing the students’ film can only be viewed in iMovie, so when they complete their video they export it as a MOV file. This enables them to share it by email with Ms Ramirez and to import it onto the school’s computers.

Downloaded from <http://technology.tki.org.nz>

iMovie and iPad are trademarks of Apple Inc., registered in the United States and other countries.

Copyright © Ministry of Education 2017, except for photographs copyright © the students

The program and resources used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-215-6





PROGRESS OUTCOME 2

# Scavenger hunt

## Annotation

By creating their QR-code scavenger hunt, Samuel and Cody show that they:

- can use a particular application (Qrafter) to create, store and test content within a digital system in order to meet a specific purpose
- understand the particular roles of components in an input-process-output system and how they work together
- understand that the same thing can act as an input or an output in different situations
- are aware of intellectual property when retrieving and using online content.

## Background

The students have been learning about data representation using binary numbers. They know that we can represent different types of data, such as text and images, using binary, and they understand how error-detection and correction algorithms can be used to detect and correct errors in data. They have practised using an error-detection algorithm to check whether barcodes are correct. In this process, they have seen that it is possible to encode binary data using the black and white strips in a barcode. They understand that computers can obtain information from these barcodes.

The students have been creating QR codes (two-dimensional barcodes) using the Qrafter app, which can link to URLs, display text and show a position on a map. In mathematics, they have been learning about compass directions and how to give directions using “north”, “east”, “south” and “west”.

## Task

Students are given a brief to work in pairs to create a scavenger hunt around their school for other pairs to complete. They are to choose three locations and place a QR code at each of them, as well as at the start position (their classroom). When each QR code is scanned, it should give a hint about where the next code can be found. The last code in the scavenger hunt should link to a free-to-use online image, which indicates the end of the hunt.

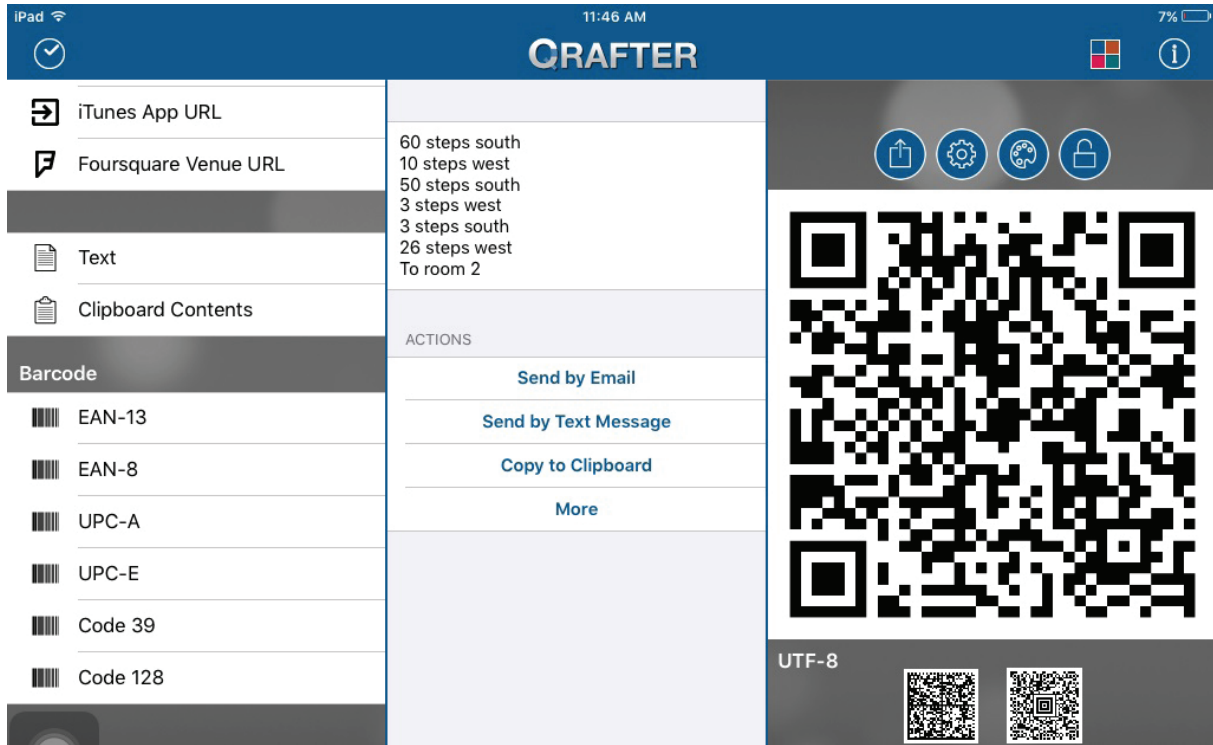
The students then follow each other’s scavenger hunts. They take a screenshot of the final image to show that they made it to the end.

**Ms Ryan takes the opportunity to link using symbols and images to an investigation by the students into environmental print in their local community.**

## Student response

Samuel and Cody create four QR codes, save them to the iPad camera roll and email them to Ms Ryan to be printed out.

Their first QR code gives a set of navigational directions to follow:



**Ms Ryan:** *What are the inputs, processes and outputs this system has used?*

**Samuel:** *When we wrote our instructions into the Crafter software, that was the input. The app processed the instructions and turned them into a QR code, which is the output.*

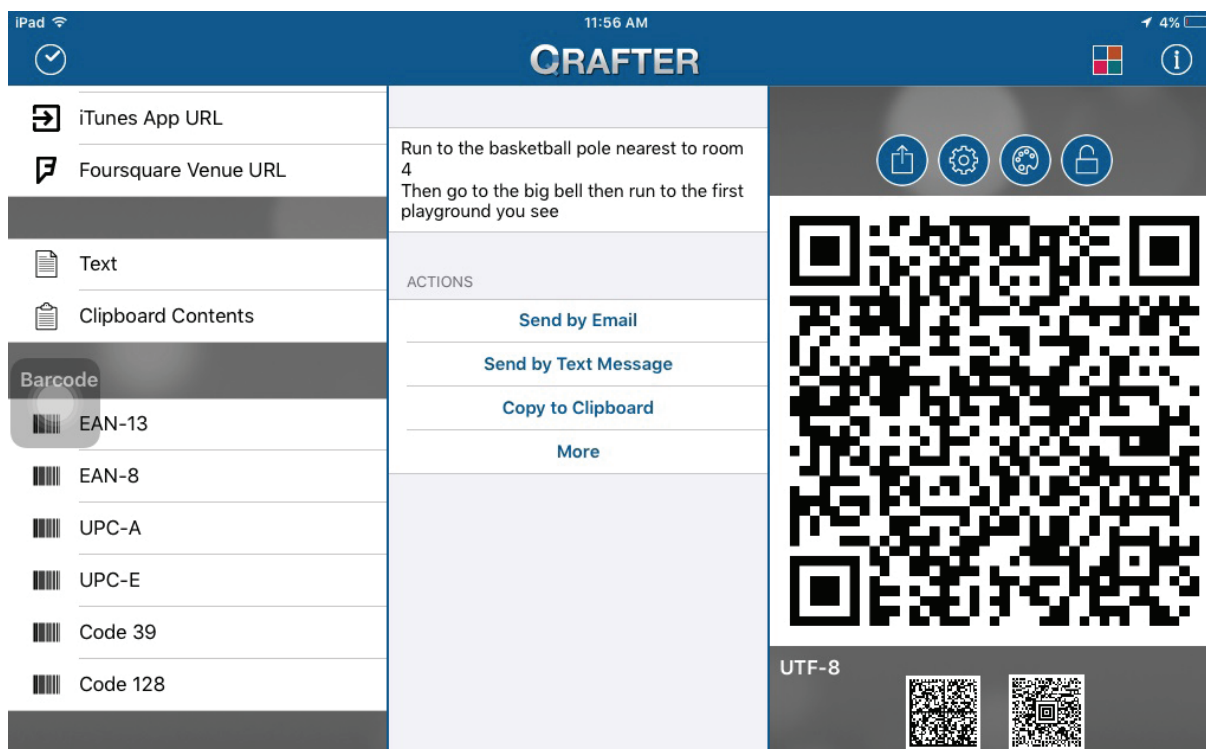
Cody doesn't initially agree with Samuel, but he understands after Samuel explains it again.

**Cody:** *No, the QR code is the input. We scan it, so it's the input, and then the app processes it.*

**Samuel:** *That's when we're scanning them though. First we're making them, so the code is the output at the moment. We scan them later.*

**Cody:** *Oh, OK.*

Their second QR code gives a set of directions using school landmarks:



Their third QR code gives a map location.

Their fourth and final QR code links to an online image, which they have confirmed as free-to-use and which signals the end of the scavenger hunt.

Before and after printing their QR codes, the students scan them to test that they give the expected output.

**Ms Ryan:** *Can you explain what you're doing?*

**Cody:** *We want to check the codes work and we have them in the right order, so we're scanning them before we put them outside. This time the codes are the input.*

**Samuel:** *Now the app turns the code back into the instructions so someone can follow them.*

All the students attach their first QR code to the whiteboard and then place their other codes around the school. They then scan another pair's code on the board to start a scavenger hunt.

Downloaded from <http://technology.tki.org.nz>  
 iMovie and iPad are trademarks of Apple Inc., registered in the United States and other countries.  
 Crafter is a product created by Kerem Erkan.  
 Copyright © Ministry of Education 2017, except for Crafter images  
 The programs used in this exemplar are not officially endorsed by the Ministry of Education.  
 ISBN: 978-1-77669-216-3



PROGRESS OUTCOME 2

# Superhero robot

## Annotation

Through developing and trialling their robot, Caitlin and Mereana show that they can:

- draw on their knowledge of how components function and work together to transform inputs into outputs
- connect input sensors (colour and ultrasonic) to their processing device (NXT brick)
- control the output devices (motors) by writing and downloading programs
- test that their system works
- identify malfunctions and failures within the system and fix errors.

## Background

The students have been learning about robotics and the input-process-output cycle within a basic digital system.

## Task

The students are given a brief by their teacher. It includes working collaboratively in small groups to develop, build, and test a Superhero Robot for a search and rescue competition. They are instructed to take photos at key stages of the development process and to upload these to the class blog. After finalising their building and testing, each group must create a poster that communicates the attributes (superpowers) of their robot.

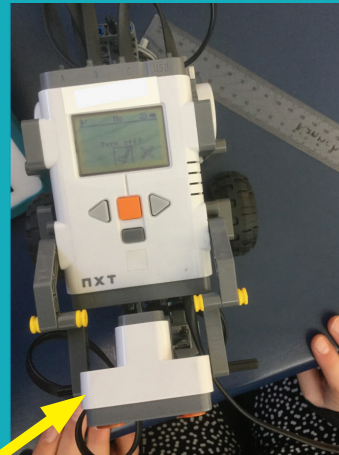
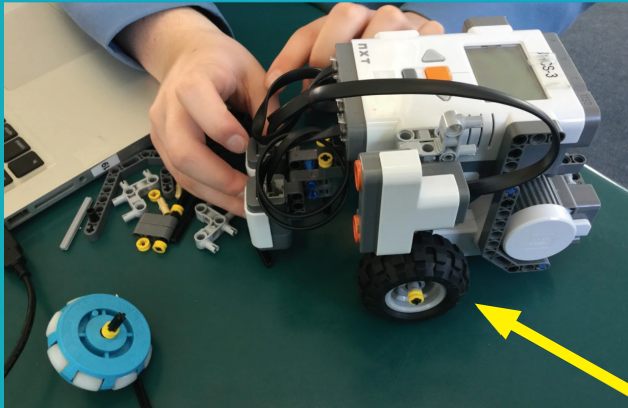
In the class blog and the poster, the group should identify and describe:

- the inputs into the system and why and how they used them
- the processing part of the system and its role in the system
- the outputs of the system and why and how they used them
- how they controlled the system, including examples of how they controlled what the robot did using inputs
- the types of testing they used to make sure the robot system worked
- the malfunctions and failures that occurred and what they had to refine to overcome them.

## Student response

Caitlin and Mereana use LEGO MINDSTORMS® to develop, build and test a Superhero Robot for the search and rescue competition. The following images and notes are taken from their contribution to the class blog.

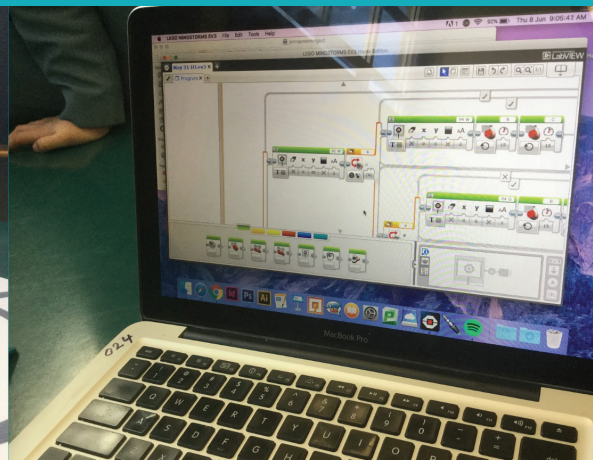
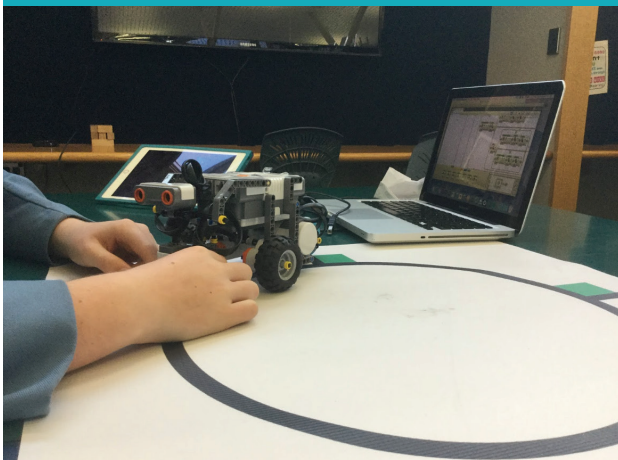
### Building our robot - connecting colour and ultrasonic input sensors



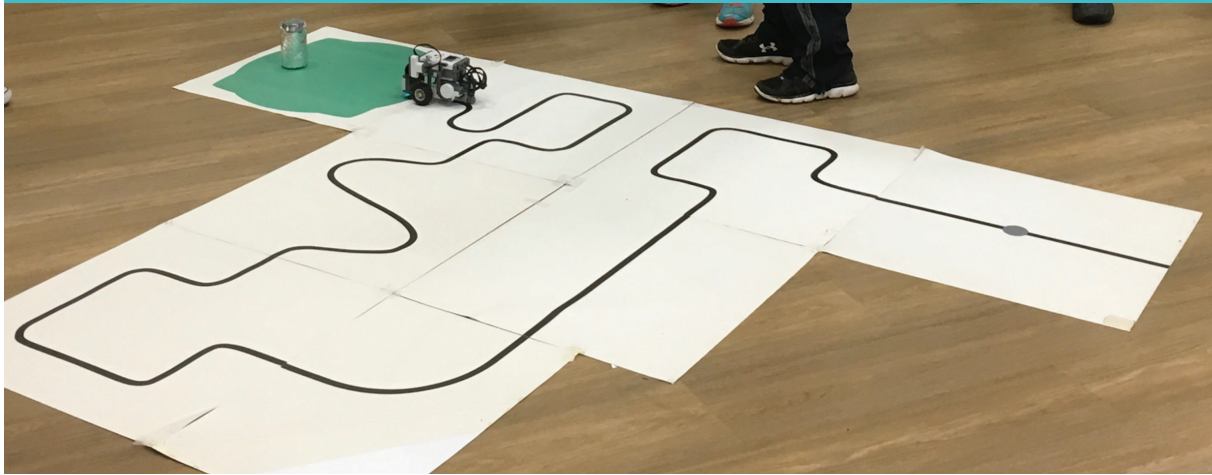
We moved the ultrasonic sensor to the side because the robot was too long to turn well and was 5 cm over the length allowed in the rules.

### Programming and testing our robot to follow lines

The robot reads the colour from the input sensors and sends output to the motors. We changed the power levels in our program to control the motors.



## Back on track - success at RoboCup :-)



Caitlin and Mereana create a poster about their Superhero Robot.

# Superhero Robot

## Inputs:

Inputs that we used are colour sensor to follow the black line, ultrasonic sensor to make it get the can out of the green by telling it that when it is less than 30 cm go really fast to rescue it out of the green, the buttons on the brick to tell it when to play the program and when to stop the program. We told the motors what to do by putting numbers into the program and telling it what speed to go and how fast.

## Process:

Our process was the program telling the output what to do from what we told it from the inputs to do. It will follow the input and come out in the output.

## Output:

One of the outputs is when it follows the black line and tells it to switch when it sees black. From the process the output will always do what it is told but will change when the process has told it to switch or make a decision. When the process has told the ultrasonic sensor to charge at the can the output will do that but it will search for the can first.

## Testing:

The main things that we tested were what colour it was reading so it can move forward carefully and also how close the can was from the robot.

## Transform:



Our process was the program telling the output what to do from what we told it from the inputs to do. It will follow the input and come out in the output.

## Our robot... Wall-E:



Downloaded from <http://technology.tki.org.nz>

LEGO® and LEGO MINDSTORMS® are trademarks of the LEGO Group of companies, which does not sponsor, authorise or endorse this site.

Copyright © Ministry of Education 2017, except for student work and photographs copyright © the students

The program and resource used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-217-0



PROGRESS OUTCOME 2

# Digital debate

## Annotation

During a structured debate, Charlie, Edward and Nico show that they can:

- describe a range of real-world applications of robotic systems, including in social-services and manufacturing settings
- explain how some robotic devices have changed over time to improve people's lives
- construct an argument for the positive impact of robotics on humans and society.

## Background

The students have been learning about robotics through the practical, hands-on development and building of robots in preparation for a robotics competition. They have watched online talks and videos about robotics and read articles about robotic systems and how they have changed over time. In other learning areas, they have been developing and using debating skills in a range of topics.

## Task

The students are organised into teams for a debate, each team arguing either for or against the statement "Robotics will be the downfall of mankind".

To construct and support their arguments, each team is required to review the robotics material already discussed in class, carry out further research and select examples as evidence.



## Student response

The following image is from Charlie, Edward and Nico's research notes on robotics. Their team is arguing against the statement "Robotics will be the downfall of mankind".

**How Robots Have Changed Over Time**

The first robots were only used for manual labour. Now they are used for detecting cancer, driving cars and deactivating bombs.

**Examples of Robots in Our Daily Lives**

Robots are used for things that humans can't do or find dangerous, like deactivating bombs, which are too dangerous for humans, and doing more precise surgery so it is safer.

**How a Real World Robot Uses Input Sensors and Controls Output**

An example of this is the machine that takes blood. This uses ultra sound X-ray sensors to find the veins, and when it does it injects the needle. This makes this process more accurate than when humans do it.

**Charlie:** *I'll collect information on health robots, Edward can do transport and manufacturing and Nico can do robots in war.*

**Edward:** *We'll have to show how these robots have made life better for people. Let's find more examples of successes in health, transport and war to make our argument stronger.*

**Nico:** *Let's do a robot that's really changed things, like self-driving cars, and how they will make roads safer. The others will probably say it's going to be more dangerous.*



PROGRESS OUTCOME 3

# Reflecting on a game design

## Annotation

Within the context of game design, Luka is able to show his understanding and application of:

- the role of operating systems in managing personal computer hardware, security and application software
- file management conventions
- a range of storage devices
- selecting software and file types for a particular purpose.

## Background

The students have learnt about the key features of operating systems, file management conventions and the importance of operating systems in managing their digital devices and applications. They have developed a class set of conventions for file and folder management that have been applied in all their learning contexts.

The students have recently completed a unit of work on game design that included developing a game idea, images and the actual game.

## Task

The students are given the following task sheet that requires them to reflect on how they used their knowledge of operating systems, file management, and file types in their recently completed game project.

The task provides the students with an opportunity to follow a particular text structure to clearly communicate their knowledge and understanding.

## Operating Systems and File Management in Game Design Assessment Report

Create a short report to show your knowledge of operating systems, file management and file types by showing their importance in your game design project. Your report should be well-organised, with appropriately styled headings, fonts and headers/footers. It should include brief written descriptions and screen captures of your use of OS key features, file management and file types when creating your game.

### Operating Systems

- Describe (and show screen captures of) the key features of operating systems that you accessed to complete your game design outcome

### File Management

- Describe how you structured and named your files and folders, with reference to file-naming conventions that are important
- Explain the purpose for your file and folder management system and use of particular storage devices

### File Types

- Give an example of and explain why you used certain file types when developing one part of your computer game (for example, the sprites or logo)
- Explain why you saved your files in a particular format, with the reasons why the file type suited the purpose of the outcome

## Student response

Luka's response to the task sheet is shown below.

# OPERATING SYSTEMS (OS) and File Management For Game Design

## OS with Game Design

The OS is what controls and manages all of your device's applications, software and hardware. Games are made to run on a specific OS. For example, they might run on IOS, Android, Xbox or PlayStation. Every application and every game we use is coded to work with a specific operating system. If I am going to be a game designer in the future, then I will need to know which OS my game is targeted for. The OS is what knows about the particular hardware that the game is being run on and manages that. For example, if there is a special controller, or if there is a touch or swipe, or if there is a keyboard to control the game.

When I was designing my game, I used the OS to:

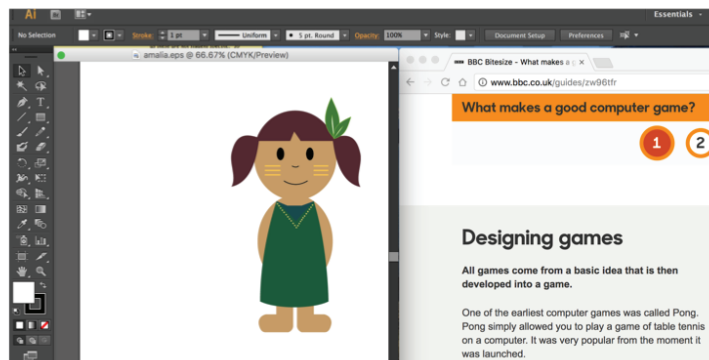
- help me launch my design and programming software
- switch back and forth between apps (multi-task)
- print off my work
- log on to my computer
- manage my files
- share my work with my teacher and friends.



The OS uses a **GUI (Graphical User Interface)**. This focuses on making it easier for the user to perform tasks (e.g. opening up an application) without

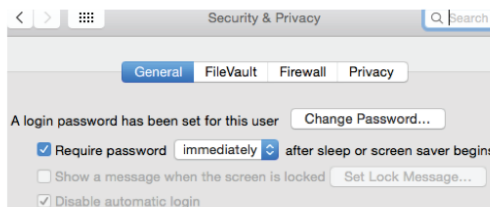
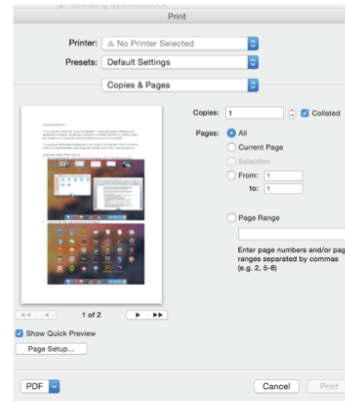
having the need to memorize an interminable amount of commands by simply clicking on icons and buttons. I can use the dock to launch my programming software (GameMaker: Studio) instead of memorizing the commands to launch it.

My OS (OS) can **multitask** between applications and desktops, which means that multiple applications and desktops can be open and available at the same time. It means you don't have to close an app to open another. I can use the Google Chrome browser and my design software at the same time and easily swipe back and forth.



## OPERATING SYSTEMS (OS) and File Management For Game Design

When I want to print my designs for my game, I use the OS because it **manages printing and all of the computer's hardware** to devices such as: screen, mouse, keyboard...etc. This means that I am able to print things (e.g. documents, pdfs, pngs ...etc.) straight from my computer.



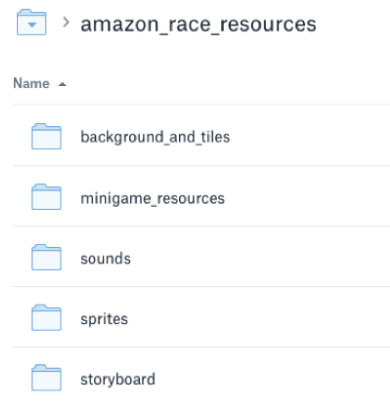
The OS is what manages the **security of my computer** (logons, installation of applications, airdrop file transfer, security through Bluetooth). This means that before something, such as a file, is installed, downloaded or transferred it

asks for your personal password or asks you to accept. When I airdrop files to my partner that I am designing my game with, the OS asks you to “accept” files so that random hackers can’t send you files on airdrop.

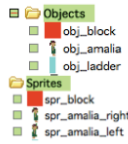
### FILE MANAGEMENT with Game Design

Another key feature of the OS is that it allows you to **manage files and folders**. This means that you can: name, rename, lock, colour, code, tag, paste, copy or delete any files and folders. The OS can help organize files by changing how we view the folders (i.e. showing items as icons, in a list or with Cover Flow). The OS can also help arrange files by date, name, kind, application...etc.

I organised my work in folders on DropBox so I could share my work with my teacher and partner easily. Also, DropBox is in the cloud, so it backs up my work automatically and it is safer than only having one copy on my laptop, especially for an important project.



## OPERATING SYSTEMS (OS) and File Management For Game Design



When naming my folders and files I don't name them 'untitled' or 'untitled1.doc' as then they would be hard to find, and I would have to open and check everything to find what it is I want; instead I named them strategically. For example, for my game design when naming my sprites, I would put 'spr' at the start and 'obj' for my objects. This way it made it less demanding to identify what type of file something is and divide it into categories.

I also don't use any caps or gaps to help get me into the habit of having good naming conventions. It is especially important to get into the habit of these good naming conventions for a subject such as Digital Technology itself. This is because gaps can break computer programs and uppercase letters could be read and interpreted differently to lowercase letters in other computer programs and/or web pages. Therefore, for when programming my game in GameMaker: Studio for my game design project, I did not use any gaps or caps in the names of my objects, sprites, tiles, sounds and backgrounds.

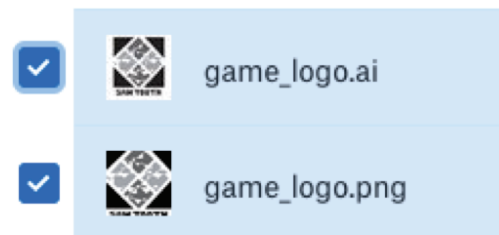
### FILE TYPES

Another thing related to file management is knowing about file types. The OS recognises a lot of different file types and will automatically start an application when the file is clicked. For example, when I double click a .docx file, it opens in Microsoft Word.

When creating my sprites and resources for my game, I made them in Adobe Illustrator and therefore saved them as .ai files. This meant that I could always go back to my creations and change/edit or add features to them in Adobe Illustrator whenever I wanted.

After creating my sprites for my game in Adobe Illustrator I would save them as a .png as that file type format supported transparency, which was desirable, as I would not want a block of white around one of my objects during my game.

The .jpeg file format did not support transparency and is more suited for photographs. Therefore, I did not use this file type to create my sprites, as it would mean that there would be a block of white around the objects of my game.



Downloaded from <http://technology.tki.org.nz>

IOS is a trademark or registered trademark of Cisco in the United States and other countries and is used under licence.

OS X and Cover Flow are trademarks of Apple Inc., registered in the United States and other countries.

Google Chrome™ is a registered trademark of Google Inc., used with permission.

Adobe Illustrator® is a registered trademark of Adobe Systems Incorporated in the United States and/or other countries.

PlayStation is a registered trademark of Sony Interactive Entertainment Europe Limited.

Bluetooth® word mark and logos are registered trademarks owned by the Bluetooth SIG, Inc.

Android is an open-source operating system developed by Google.

Xbox is a registered trademark of Microsoft Corporation. Microsoft Word is part of the Windows Office suite. This publication is not affiliated with, or authorised, sponsored or approved by Microsoft Corporation.

Dropbox is a trademark owned and copyrighted by Dropbox, Inc.

GameMaker: Studio is a product created by Yoyo Games.

Copyright © Ministry of Education 2017, except for student work copyright © student

The programs and resources used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-219-4





# Our changing digital society

## Annotation

In working collaboratively to present their research, Miyeon and Kade demonstrate that they can:

- design, develop, store and evaluate digital content that meets its purpose
- use selected software to develop and combine digital content
- gather, analyse and present data in a meaningful way
- understand the relationship between operating systems and digital devices and the influence and impact they have on humans and society.

## Background

The students have learned how to use Google™ applications to develop, combine, manipulate and present content. They know how to create Google Forms to gather data. They have been learning about the features of operating systems for specific devices, as well as watching videos and taking part in class discussions about the changes in digital devices and operating systems over time. The class has brainstormed and discussed changes in user interfaces when interacting with digital devices, including changes in both the hardware and user interface of their operating systems.

In mathematics, the students are conducting statistical investigations, learning how to pose investigative questions, collect data, and select the most appropriate data display for their results.

## Task

The students are given a brief to guide their investigation into “Our changing digital society”. They are to work in pairs or small groups and can present their findings in a format of their choice. The final outcome will be presented in their social studies class and used as evidence of understanding for an inquiry into the impact of new technology and ideas on culture and heritage.



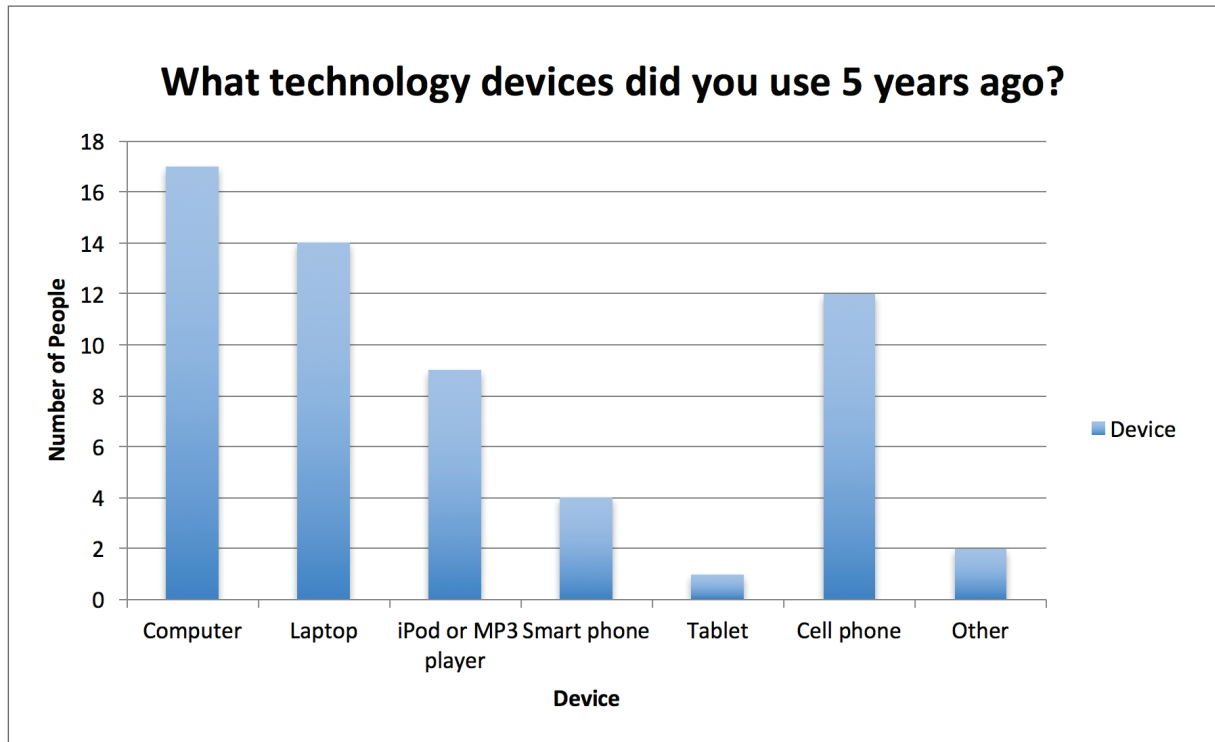
## Our Changing Digital Society - Operating Systems and Digital Devices

This is a team project. You will work collaboratively with your partner(s) to:

1. Research changes in operating systems and digital devices over time and the impacts on our society.
2. Plan and design a survey, identifying what questions you will ask. Think about the design of the question type and how it will help you to analyse your data (e.g. multi-choice, tick box, scale). You should have a range of questions that help you to determine your respondents':
  - a. use of various devices (e.g. laptop, desktop, tablet, mobile) and the OS they run on
  - b. preferred methods of interaction with the devices (e.g. touch, swipe, keyboard, mouse, voice)
  - c. use of security features that are available on the OS/device
  - d. changes to the interaction with and use of devices over time
3. Create the survey in Google Forms. The survey should be given to a range of people across different age groups (e.g. grandparents, parents, teachers, aunts, uncles, brothers, sisters, cousins, friends).
4. In Google Sheets, analyse your results from the survey (using the various formulae we have learned in class such as average, countif, sumif).
5. On another worksheet tab from your own survey data, analyse the changes in operating system use over time from the w3schools website. Note: Use January data only for the past 10 years of statistics.
6. Create a slideshow, poster, report or movie based on our class discussions, your own research and your survey results. Include the following content:
  - a. Your own experiences:
    - i. How have operating systems and devices changed since you started using computers?
    - ii. What operating system & device do you prefer to use primarily and why?
  - b. Survey results:
    - i. Use tables and graphs to show the results of your survey and the w3schools data.
    - ii. Summarise the findings.
  - c. Conclusions:
    - i. How have changes to operating systems and devices changed the way we work, communicate, study, relax and also entertain?
    - ii. How do you think these will continue to change with the introduction of new technology, in both positive and negative ways?

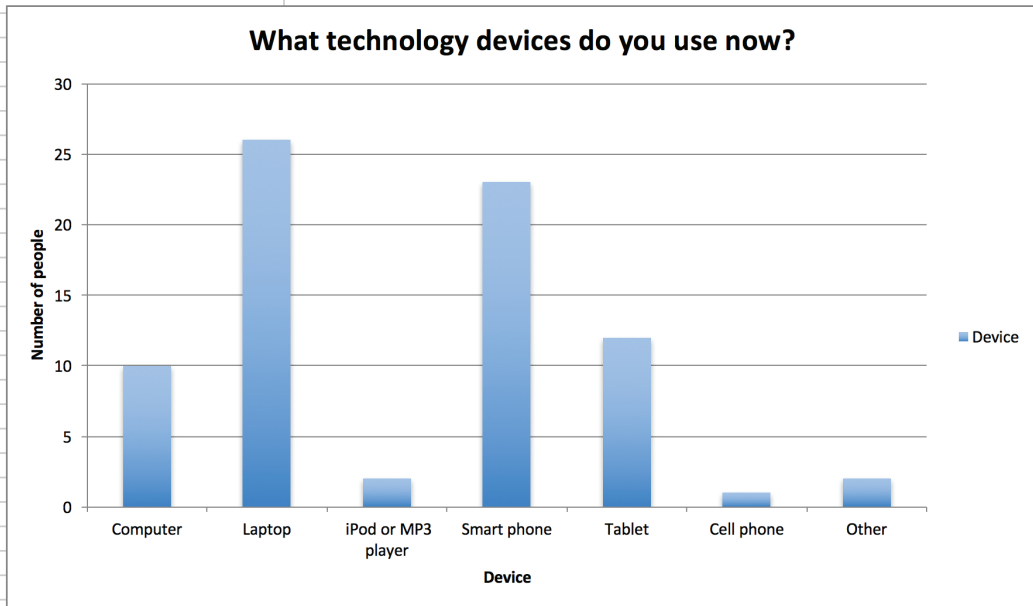
## Student response

Miyeon and Kade design and develop an on-line survey, manipulate and analyse the data using a spreadsheet, and present their findings online. The following response shows excerpts from their presentation.



=COUNTIF(E3:E24,"\*Tablet (e.g. ipad, Galaxy etc.)\*")

	Laptop
	Laptop, Cell phone
<b>What technology devices do you use now?</b>	<b>Device</b>
Computer	10
Laptop	26
iPod or MP3 player	2
Smart phone	23
Tablet	12
Cell phone	1
Other	2





- Mr Pearson:** *I see you've performed a COUNTIF calculation in your spreadsheet. Can you tell me why you used this formula?*
- Miyeon:** *We wanted to calculate how many different devices people use, and the quickest way to do this is to use a COUNTIF. You can see how the formula calculates how many times a certain word (like "laptop") is used in the column and gives the answer. This is a reliable way of quickly counting the data entries.*
- Kade:** *We can also use this formula again for other calculations -- we just need to change the name of the device. It's better than manually counting everything, and if we want to add more data to the spreadsheet the calculation will update.*
- Mr Pearson:** *You've used graphs to present your survey data. Why have you done this?*
- Miyeon:** *We used graphs because they're easy to read, and they show the findings of the survey very clearly. We can also copy the graphs to other documents. For example, we can copy the graphs into our infographic about "Our changing society".*
- Mr Pearson:** *So what does your survey data tell you?*
- Kade:** *The graphs tell us that some of the changes between five years ago and now are that computers have decreased in popularity, and laptops have gone up in popularity. The use of cellphones has decreased. People are using smartphones instead.*
- Miyeon:** *They also tell us that people like having more functions on their phones, and that the newer technologies (with more functionality) have had the greatest increase in usage. We can also see from the graphs that five years ago, people liked to control their devices using a keyboard and a mouse. Most people didn't know what a track pad, touch, or voice control were. Today most people know what these control methods are.*
- Kade:** *Yes, I'd say that even though the keyboard and mouse have remained popular, touch and swipe controls are very popular, and most devices use these. Voice control isn't as popular. I think that's because it isn't as easy to use, and it can be unreliable.*

## Conclusion



Operating systems and digital devices have affected the way we work, study, communicate and spend our free time greatly over the past ten years. Having cameras and sound built in to devices for user interaction is making communication easier, more personal, face-to-face and faster through the internet. It is easier to communicate with people all around the world making the world seem smaller.

We believe that operating systems will continue to become faster and make it easier to work our devices. This will affect our lives by making them revolve more around our devices, meaning we spend less time face-to-face with our family and doing other activities such as sport and art. An example of this is with the teenage generation. They are spending more time on their devices watching

movies, playing games, doing homework and talking to friends instead of playing outside and doing non-digital activities. We believe that this problem will become huge and through all generations with less people going outside and doing things.



We think devices will become smaller, touch and voice controlled and also controlled by other devices over the internet (Internet of Things). This means less human error as computers do more of the work. The negative side of this type of

technology is that we would become more reliant on the devices and if they broke down we would not know what to do to help ourselves. We could get less exercise and less appreciation for natural things. Security is becoming an even greater issue because everything is becoming linked together. Our operating systems and devices keep adding more and more security features, such as longer passcodes, finger print recognition and even face recognition.

Downloaded from <http://technology.tki.org.nz>

Google™ and Google Forms™ are registered trademarks of Google Inc., used with permission. Images in students' conclusion sourced from <https://pixabay.com/> and used under Creative Commons CCO.

iPod is a trademark of Apple Inc., registered in the United States and other countries.

Copyright © Ministry of Education 2017, except for student work copyright © the students and images in the Conclusion

The programs and resources used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-220-0



PROGRESS OUTCOME 3

# Designing a logo and business card

## Annotation

Aria uses an iterative process and demonstrates that she is able to:

- select appropriate software for each stage in the development process
- successfully combine digital content to create a final outcome
- apply design elements to enhance the readability and usability of the final outcome
- apply different colour modes and resolution to an image and explain their impact on the quality or usability of the final outcome
- apply efficient file management procedures
- differentiate between linked and embedded content and select the appropriate mode for her outcome.

## Background

The class has learned how to use vector image creation software and layout software. They have looked at case studies in logo development and considered ethical issues in logo design. They have also explored the key principles of what makes an effective logo and researched and evaluated logos with reference to these principles.

## Task

The students are given a brief to create a business name based on their own areas of interest and to design and develop a logo and business card for the business. They are asked to record their development process in a digital format (for example, audio recording, screen capture or text and images).

## Student response

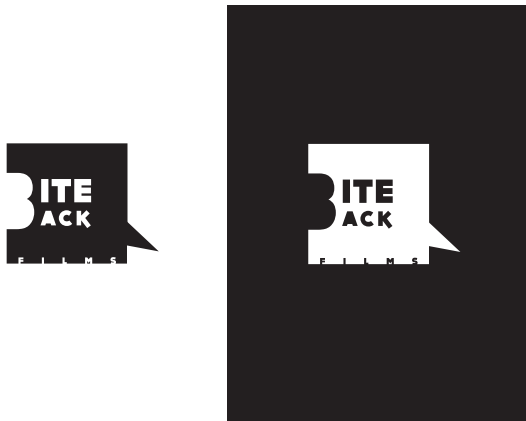
The following pages are from Aria's portfolio, which includes her sketchbook and digital submissions of her work.



Using my analysis of other logos, I was able to come up with some ideas for my logo. My analysis told me what worked and did not work for different companies, what graphics to use and the different ways to use text and images.

I came up with the idea of cutting text from an object or image to create contrast. I also wanted to use a cool font which would be the focal point of my logo.

I had to think about the readability of the text – for example, the word ‘films’ – the word needed to stand out. My friends also helped me come up with a business name. I found it hard to think of a good name and we brainstormed together to find the best one.



This is an example of how I tested the colours – you can see I tried the reverse (white on black compared to black on white). After showing my friends my two designs, we decided the white on black gives more contrast and is more effective.



I have also used **contrast** throughout my logo design to ensure that the viewer can easily differentiate the pieces of information and images. This allows the text to stand out more vividly.

I chose to use white on black (rather than black on white) as I feel that:

- a) the images and text stand out more boldly if it is white on black, and
- b) there are so many white business cards out there (I know my Dad's department's business cards are all black text on white) – I wanted to ensure that my business card was different to all the others – that it stands out.

I also used large amounts of **blank space** on the back and front (in particular, the back) to make sure that the viewer's eyes were drawn to a particular piece of information and were not distracted by different coloured, detailed patterns.



Throughout my design, I also incorporated  **Kerning**, which I used to make sure that the fonts I used were not overly squished together (the font that I chose to use as the main text for the logo, and consequently, the business card, Dock 11, was quite squished so I used kerning to spread the letters apart so that it was more easily readable).

I also made sure I had a **two font** maximum so that the final business card and logo did not look too busy, yet still showed the difference which was used to create **visual hierarchy**. My two fonts were Helvetica Light and Dock 11.

Finally, I used **leading** to make sure that similar information was grouped together and unimportant information was not unnecessarily highlighted.

I used an **image/icon** similar to that used on Google Maps (which I created myself using the pen and shape tools). I used this as it is a symbol that is well known and recognised, meaning that it would be easily seen and recognised as the location of the business.

I used clear and strong right alignment to give the viewer a line to follow to view the information.

This is further accentuated by the logo's point, which points towards the information which, again, gives a definite line that the user follows to see the information.

This visual hierarchy ensures that the most important things will be viewed first, followed by those that are slightly less important.

- Mrs Cho:** *Can you explain some of the tools you used to develop your logo?*
- Aria:** *I used the shape and pen tools to create the speech bubble and merged them into one shape using the shape modes. I used a font with a clear “B” shape (Bebas Neue) and widened it with the transform tool to make it stand out. I also used the pathfinders to correctly cut the “B” outline from the speech bubble, and I used the kerning tools to get the effect I wanted for the “films” across the bottom. Throughout, I used multiple guides to centre and align text accurately with the speech bubble.*
- Mrs Cho:** *Can you explain what file-type considerations you had to think about when you were creating your logo?*
- Aria:** *I made it a vector graphic, so that it would be scalable for various uses, such as a business card, a website or an animation. Logos need to scale easily because they can be used on a business card or a billboard. I could then export the graphic for different uses and keep it high resolution for print and lower resolution for web, to keep loading times down.*
- Mrs Cho:** *What colour mode did you use for your logo creation?*
- Aria:** *Since I was designing for print, I set up my document in CMYK mode. RGB is for images that are going to be viewed on a screen, because computer screens use red, green, and blue lights to create each pixel. Printers use cyan, magenta, yellow, and black ink to create colours. It’s important to get the colour mode correct or it will look different to what you intended.*
- Mrs Cho:** *What was the most important aspect of your file and folder management when working on your logo and business card?*
- Aria:** *It was particularly important to have good folder structure because if a link is broken in a project (for example, if I moved or deleted the logo), the program will replace the image with a low-resolution copy, and this means it will be pixilated and not at the original vector standard.*
- Mrs Cho:** *Did you link or embed your logo and other graphics when using them in your business card?*
- Aria:** *I linked the file because it was better and more efficient. I could then alter or edit the original image in Illustrator, and the changes would flow through to the linked images in the InDesign document. Whereas with embedded files, they’re not editable and don’t update efficiently. You have to copy and paste again if you don’t link the files.*

Downloaded from <http://technology.tki.org.nz>

Adobe Illustrator® and Adobe InDesign® are registered trademarks of Adobe Systems, incorporated in the United States and/or other countries.

Google Maps™ is a registered trademark of Google Inc., used with permission.

Copyright © Ministry of Education 2017, except for student work copyright © the student

The programs and resources used in this exemplar are not officially endorsed by the

Ministry of Education.

ISBN: 978-1-77669-221-7





PROGRESS OUTCOME 3

# Designing a road safety game

## Annotation

Through an iterative process of designing and developing a board game, the students show that they can:

- critically evaluate outcomes to determine fitness for purpose for the intended audience
- create original digital media content by selecting and using software tools and techniques
- test and evaluate digital content to determine whether it meets its intended purpose
- store and share digital content using a range of cloud-based and local storage devices.

## Background

The students have learned how to design and create vector graphics for use in print and online contexts. They have also learned how to develop concepts by researching existing ideas, analysing their features, and evaluating their content to determine whether they will fit the intended purpose. The class has also been working on the skills they need to work collaboratively to achieve specific outcomes.

## Task

The students are asked to work in groups of three to design and develop a board game, using a range of digital-media tools and techniques. The game will be entered into the New Zealand Transport Agency's game design competition. In the competition brief, students are required to work collaboratively to make decisions about game design, goals, rules, and the design and development of game assets (logo, game pieces and game board). The students are then required to work independently to design and develop a specific game asset.

Ms Dunlop uses this task to observe and feed back on how each group works collaboratively to develop an outcome within a set time frame.

## Student response

The following discussion between Ms Dunlop and one of the student groups covers their use of digital tools to develop and document a road-safety game for the competition.

**Ms Dunlop:** *How did you work as a team to develop your game and the individual elements of the game?*

**Ella:** *We all discussed the general design of the board game, including the theme, how it would be played, the rules of the game and the design of the specific game pieces. Once we had decided on those main ideas, we were able to divide the tasks. Each person was responsible for developing a specific game piece – like the board, the logo, the cards or the characters.*

**Crystal:** *I co-ordinated the overall design and development of the game pieces – I set up a Google Drive™ folder for everyone to put their designs in, and I checked in on everyone to make sure they were on track with their work. I was also in charge of designing and developing the logo, which I shared with the rest of the team so that the same look was applied across all aspects of the game.*

**Kahurangi:** *I developed the game cards using the agreed theme of the game. We wanted to make sure we used the same logo (Prickle in a Pickle) and colour scheme throughout the game, so I looked after this aspect. Once Crystal had created the logo, I could get this from the shared drive and use it on my cards.*

**Ella:** *We had regular meetings around the project and discussed ideas, how the game was developing, any issues and how we were going time-wise.*

**Ms Dunlop:** *What software did you use during the development of your game? Why did you select it?*

**Kahurangi:** *We used Illustrator to create the logo, characters and other graphics such as the board because it's a vector graphics software and has the tools we needed, such as the pen tool and the use of layers. We could export the graphics and make them scalable. It was easier to all use the same software as we were sharing work (like Crystal's logo). I set up a template for the cards – I saved the file as an .ait file – this way I could then use the template to design all of the cards. It's really easy to set up and save your work as a template in Illustrator.*

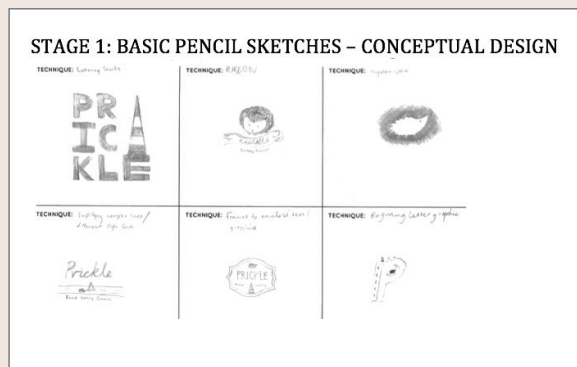
**Crystal:** *I used Google Docs for recording the design ideas we had, including ideas we got from our research on the internet. Google Docs was easy to use because it has more layout and font features for styling a document, and the others could ask questions or put comments in the document. For example, I inserted the logo design in Google Docs and shared it with the others and asked them to make comments. It was much easier to do it this way.*

**Ella:** I used Illustrator for developing my game pieces – I then exported them and put them in a Google Doc for the others to comment on. I also put the final outcome in the Google Drive for the others to access. Illustrator gives you options for exporting your images, depending on what you want to do with them. I exported the games pieces as jpg files – I also saved them as PDF files to drop into the drive for the others to see and comment on.

**Ms Dunlop:** Did you just start designing your game, or did you do some research first?

**Kahurangi:** We researched games for children and games about safety. We brainstormed different ideas and then created concept sketches for the characters, the logo and other assets. We did colour and font research too. All of these ideas we put in the Google Drive so that we could comment on them.

**Ella:** I found I was able to work at home on this project as everything I needed was in the Google Drive. If I had an idea about a game piece, I could do it in Illustrator, then export it and put the image in a Google Doc for the others to see. It saved having to be at school to get their feedback.



**Ms Dunlop:** *How did you make sure your game worked and was fit for purpose for the intended audience?*

**Crystal:** *We tested the game on five primary school children (boys and girls). They all enjoyed the game and told us a few things we could do to make it better. They suggested we add more characters to the game so that more people could play. We thought that was a good idea, and it's something that's easy to fix.*

**Ella:** *Our research showed that a good game has some kind of reward system, so we incorporated that idea into our game. We made it so that if the player answered a question correctly, they earned a token.*

**Kahurangi:** *We also trial-printed aspects of the game to make sure the colours were right and came out as we expected. Like when I printed the cards, I had to make sure they were all the same size, the logo was clear and the text was easy to read.*

Downloaded from <http://technology.tki.org.nz>

Google Docs™ and Google Drive™ are registered trademarks of Google Inc., used with permission.

Adobe Illustrator® is a registered trademark of Adobe Systems, incorporated in the United States and/or other countries.

Copyright © Ministry of Education 2017, except for students' work copyright © the students

The programs and resources used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-222-4.





PROGRESS OUTCOME 4

# Keeping track of club members

## Context

Mr Sims, a teacher and the secretary of the college's golf club, needs a way of organising membership details, subscriptions, and performance rankings for the club.

He has asked Nathan, one of the students, to help set up a system to manage this. Mr Sims has provided him with the necessary data and identified the information the club requires. This includes a term-by-term report showing all club members' first and last names, dates of birth, and total game scores for the term. An additional requirement is a separate report for each age group (juniors are under the age of 14, intermediates between 14 and 15, and seniors 16 and over).



## Insight 1: Design decisions and stakeholder feedback

As I sketched out the design of the database, I had to consider the most efficient method of structuring the fields to make it easy to query data and extract information. I also had to think about the easiest way for a user to enter data into the database and the best software to use – I chose Filemaker Pro because Mr Sims has access to this.

Following discussion, we agreed that a form would be the best and easiest way of entering data. I added this to my plan, along with a method for creating the age group reports.



## Insight 2: Developing and testing the outcome

As I set up the fields, I was careful to split the first and last name of each member and the different elements of the addresses. I also created a drop-down list to make it easier and faster for Mr Sims to enter the title of each club member. I added a validation rule to the phone number to ensure data consistency, and I included a warning message as a prompt for invalid numbers. I also added an ID field so that each member will have a unique number.

I then developed the form for entering data, which I tested with Mr Sims – I wanted to make sure it was easy for him to use and functioned as intended. On the basis of his feedback, I resized some of the fields.



### Insight 3: Creating the age group reports

I entered data for several members from each age group and then performed separate queries to extract the age-group data and populate separate reports. I showed these to Mr Sims. He confirmed they were satisfactory, and I then saved the queries so that he could access and rerun them in the future.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/Technology/Digital-technologies>

FileMaker is a trademark of FileMaker Inc., registered in the United States and other countries.  
Copyright © Ministry of Education 2018, except for student work copyright © student  
The program used in this exemplar is not officially endorsed by the Ministry of Education.  
ISBN: 978-1-77669-245-3







PROGRESS OUTCOME 4

# Keeping a watch on your fitness

## Context

The kura holds an annual wearable arts competition as part of its arts and cultural curriculum. Students must work in teams of two or more and complete their entries within one term. Their outcomes are judged by year groups before being displayed at the wearable arts evening. Local businesses and charitable groups sponsor prizes.

One category requires students from two or more different specialist subjects to collaborate on the design and development of an outcome. Jade and Aniwaha have worked together on an entry in this category.



## Insight 1: Design and construction decisions

My friend, Aniwaha, is very skilled at fabric construction, and I'm good at building and programming circuits. We decided to together make a 'health watch' that would remind its wearer to complete activities for improving their health and well-being.

We came up with the idea of a wristband with LEDs. The LEDs would display a sequence that would prompt the user to exercise and that they could change to show that they had completed the exercise.

As the electronics expert, I had to understand the components and their functionality to ensure that they worked well in the finished outcome. I knew that the components had to be as compact as possible, so that our wristband would be lightweight and comfortable for the wearer.

We decided to sew an electronics board and battery holder into a wristband. We selected ultra-bright LEDs that could be chained together.

We sketched out our initial design ideas (including possible symbols for the user interface) and got feedback from potential end users, which helped us develop specific criteria.



## Insight 2: Installing and testing components

I investigated how to sew the components on and connect them using conductive thread, while avoiding any short circuits. Building the circuit proved to be the biggest challenge – securing the components using the thread was a much more complex task than I had expected, but working with Aniwha meant we could problem-solve together and find ways to make everything work.

Once the components were sewn on and connected, we tested the LEDs to ensure they were visible on the fabric we'd chosen. The LEDs allowed for multiple colours, so we plugged the electronics board into my computer and used an open source microcontroller to modify the code to select the most appropriate colours.



## Insight 3: Meeting end-user requirements

We then trialled the wristband with end users. This raised a concern with the functionality and design of the user interface, so we adjusted the symbols on the wristband to make sense to the wearer – so that, for example, they didn't put it on the wrong way round.

Finally, we tested the connectivity of the circuits under different physical situations. We used a range of students completing different fitness activities: skipping, a beep test, gymnastics, and cheerleading. The wristband fitted our chosen testers well and stood up to the activities.



PROGRESS OUTCOME 4

# A computer for the shearing shed

## Context

The school farm has a shearing shed on the grounds. The farm is constrained by tight budgets but wants to have a working computer system in the shed with basic functionality for recording farming data.

In consultation with the farm manager and the school network manager, Zoe has been given access to some computer equipment no longer needed by the school to develop a system.



### Insight 1: Design decisions

I sketched out a plan of the components, hardware, and software I would need to create a working computer system and discussed this with the network manager and farm manager. I then checked the available components to ensure interoperability and safe use and was able to confirm that I had all the necessary components.



### Insight 2: Installation and testing

I configured the computer system hardware by installing and testing the internal components. The computer didn't respond, so I knew there was an error and had to troubleshoot to discover that the problem was the RAM seating and the internal cable connections. I attached the other computer peripherals (mouse, keyboard, and screen) and made sure all components initially functioned.

Following consultation with the farm manager and considering the budget, I selected Linux® as the operating system because it's open source with no cost for using it. I explained to the farm manager how Linux compares to proprietary software. I downloaded and installed the most recent version and ensured all components were recognised by the system.



### Insight 3: Meeting end-user requirements

After discussing storage requirements with the farm manager, I determined that he needed some spreadsheeting software. I installed Open Office, which again is non-proprietary and has no cost associated with its use.

The farm manager tested the system. It met his requirements, but he was concerned about security when students were in the shed. I installed a BIOS password and got the manager to set a password.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/Technology/Digital-technologies>

Linux® is the registered trademark of Linus Torvalds in the US and other countries.  
Apache OpenOffice™ is a registered trademark of The Apache Software Foundation.  
Copyright © Ministry of Education 2018, except for student work copyright © student  
The programs used in this exemplar are not officially endorsed by the Ministry of Education.  
ISBN: 978-1-77669-247-7





PROGRESS OUTCOME 4

# Programming pizzas

## Context

Students in a year 11 class have been investigating systems in local businesses. Their aim has been to identify a need for increased efficiency that a design intervention could meet. They would then develop a basic digital program to respond to this need.

Kyle has worked with a partner to design a program for a new local pizza business.



## Insight 1: Design decisions

To help us understand the task, we completed a case study of the business. We wanted to know what it needed to function more efficiently so that we could apply our knowledge about systems thinking to program a digital solution.

We decided that the most logical solution was to design a program to track the volume of pizza toppings used and automatically produce a regular list of what needs to be ordered when levels fall below a set amount.



## Insight 2: Construction and testing

We created the variables from the data the manager provided for each of the toppings, including how long each ingredient remained fresh compared with the amount that was typically used on particular days of the week. This meant we could use test data for developing our code.

We included actions (purchase of the toppings), conditions (logging the use of toppings from sales data), and conditional control loops that adjusted the data accordingly.

At this stage, we looked at the problem again and came up with some simple test code to check our solution. This helped us identify some further issues that needed to be addressed. After addressing these in our program plan, we started writing the code.



### Insight 3: Meeting end-user requirements

Each time we made changes, we discussed them with the manager. This proved vital to the process of developing a successful program. We found we needed help with debugging, as at times it was overwhelming.

During testing the manager decided that the program needed to include sauces. But adding the tracking of sauces was problematic, because they were in liquid form and the code was set up to track volumes and weights. If we'd had more time, we could have done this, but our deadline didn't allow it.

The final program was successful in that the store manager could see at a glance what was being used and needed reordering and what was close to its use-by date to create specials.

We also tested the final outcome using data from our school food technology rooms.

We found that the program was transferable and could be used for their tracking systems too.



PROGRESS OUTCOME 4

# Designing a portfolio website

## Context

Amber has been designing and coding a website to showcase her visual arts portfolio. During the process, she had to make decisions regarding the website's design, layout, image file sizes, and file types, all of which have an impact on its usability and functionality.



## Insight 1: Preparing images

Because I'm developing a website to showcase my visual arts portfolio, the quality of the imagery is an important consideration. I decided to use one of the school's high-quality digital cameras and shoot my artwork in the photography room to ensure consistent lighting.

However, I couldn't put the raw images straight onto my website, as the large file size would make the download time too slow (when researching human-computer interaction, we learned that most users won't wait longer than a few seconds for a page to load). So I used GIMP to scale the images to fit my design and exported them as optimised jpegs, which testing had proven would give me the best quality for a reduced file size.



## Insight 2: Website design

When developing my design, I researched websites of photographers and artists to see what works best when displaying portfolios. I learned that most portfolio websites have smaller thumbnails of images for the user to browse and navigate for a larger view.

From my research I realised it's important to use a neutral colour scheme without background patterns so that the focus is on the artwork rather than the website graphics. I also wanted only a small amount of text so that the artwork could be as large as possible.

Before starting to code my design, I got feedback on it from my visual arts teacher and classmates, because they understand design aesthetics and often look through art-focused websites. I also asked my digital technology teacher about whether my design would be able to be coded effectively. I integrated their feedback into my final design.



### Insight 3: Development and improvement

From our practice in web coding, I realised that it would be more efficient to first plan out my page structure on paper, thinking about HTML tags I would need. I coded the structure and then I applied some basic Cascading Style Sheets (CSS) to see how it would look. (It's easy to make changes to the CSS and look at them in the browser for quick testing.)

Once I had worked out the structure, I started adding some content and the colour schemes and fonts. I used comments in my HTML and CSS to help me with my testing because you can comment out parts and easily go back and forth between choices. My classmates and my teacher helped me to make decisions about size and font choice that matched my design aesthetic and made the information easy to read.



### Insight 4: Browser testing

All along, I had been testing my website on my laptop in both Chrome and Safari as they are the most common browsers for macOS. My teacher had set up a server so that we could upload our websites and have others test them out. Having my classmates and parents test my site on different computers and browsers showed me things I could improve if I ever wanted to use it professionally. I decided to make sure it worked on modern browsers but didn't waste time checking whether it worked on older ones, because it was currently only for personal use.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/Technology/Digital-technologies>

Google Chrome™ is a registered trademark of Google Inc., used with permission.

Safari is a web browser developed by Apple Inc., registered in the United States and other countries.

macOS is a trademark of Apple Inc., registered in the United States and other countries.

Copyright © Ministry of Education 2018, except for student work copyright © student

The programs used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-249-1







PROGRESS OUTCOME 5

# A bowls club database

## Context

The secretary of the neighbourhood bowls club is Jack's grandfather. He needs a way of organising membership details, subscriptions, and performance rankings for the club. He has approached Jack to help set up a system to manage this and provided him with the necessary data.

In consultation with his grandfather, Jack has identified the information the club needs and the end-user requirements. These include producing a report showing all club members' first and last names and whether their membership fees are outstanding.

The club also wants a tally of each member's scores, so that they can award certificates to high-scoring players.



## Insight 1: Design decisions and stakeholder consultation

I sketched out a plan for the database structure using two tables, one for membership details and one for game scores. I had to consider the most efficient method of structuring the fields and tables to avoid duplicating data. I also had to think about how the user might want to enter, find, and extract data from the database and the most efficient way to do this. Finally, I considered how the tables might link to each other.

I consulted grandad about my plan, including the layout and linking of the two tables, the fields and data types, and the validation rules I intended to use. He was happy for me to proceed. We also discussed designing a form for him to use when entering data. We decided I should add buttons to the form for entering a new record, printing, and closing the database.



## Insight 2: Developing the outcome

I set up a folder to store the database and proceeded to set up the two tables.

I added the following validation rules:

- telephone number: limited to ten characters with a warning message as a prompt for invalid numbers
- email address: ensuring that the @ symbol is included and a warning if it isn't
- date of birth: ensuring that a 4-digit year and viable date are entered.

In the game scores table, I added a field for total score and created a calculation that would add up the scores to keep a running tally for each member.



## Insight 3: Testing the basic outcome

Once I had set up the tables, I entered a couple of members' details and tested the validations for the date of birth, email, and telephone number fields.

To determine that my tables were linked correctly, I checked that the membership ID in the game scores table brought up the correct member details from the membership details table. I also tested my calculation field to see if the scores were adding correctly.

Everything worked well, so I continued to add the rest of the records.



## Insight 4: Developing the data-entry form

I developed a form for entering data, resized some of the fields so that the data displayed accurately, and added buttons to the form.

I then had grandad test the form to make sure it functioned as intended and was easy for him to use. He wasn't happy with the placement of the buttons because he had to scroll down to find them, so I moved them to the side of the form.



## Insight 5: Creating the report and certificate

I performed a query to show all members with fees outstanding and their membership details and used the customised design tools to produce a report containing all the required fields.

I performed a second query to extract the data for the certificates and populated a list for grandad. He was able to use this data to create a merge that allowed him to easily print out all the certificates.



PROGRESS OUTCOME 5

# Irrigating the orchard

## Context

Katie's aunt owns a citrus orchard in Northland and has issues with irrigation. Her concerns include knowing how much water is in the dam and where water is going in the orchard, ensuring she only accesses a certain amount of water each day, and making sure there is always enough residual water for the volunteer fire brigade.

She has asked Katie to help her find a way to manage her water use through both an automatic and a manual interface.



## Insight 1: Investigating the issues

I went on a site visit with Auntie Jane to see the orchard and how it's laid out. After she had gone through the requirements for the project, I was able to start thinking about a range of considerations. These included:

- the systems that were currently set up – I sketched out the existing watering system, which was basically a timer on a pump sending water to irrigation hoses
- the electronics required for the size of the orchard
- how to maintain awareness of water levels in the dam
- the requirements for fire fighting
- how much water can be taken off the dam
- reliability concerns.

I then started thinking about how I could respond to each consideration.



## Insight 2: Planning the outcome

I drew a number of interface sketches to help determine component selection and showed my ideas to my teacher and an electronics engineer to confirm my modelling process. I then carried out virtual testing of the components that I could possibly use.

The orchard is large and on a terrace so I needed to consider requirements such as longer watering times on the upper terraces and shorter times on the lower terraces. There was also a requirement that watering takes place until soil moisture sensors are at a certain resistance.

Using RF transceivers like the Dorji ASK modules, HC-12 short-range modules or LoRa long-range modules that enable remote data packet exchange would allow for a more 'internet of things' approach.

Components and sensors I determined I would need to test and configure included:

- a DC solenoid valve component such as the H-Bridge L9110H
- a decoupling capacitor to stabilise power supplies under high current loads
- analogue (resistive) moisture sensors to measure what level the dam is at
- digital temperature sensors such as the DS18B20
- an I2C-based sensor such as the BME280 digital sensor
- user Interface controls and feedback
- some SPST-type tactile switches, or a keypad (if many input switches are needed, a shift register may be appropriate)
- an LCD/OLED display via parallel or I2C communication
- a real-time clock with a secondary power supply.



## Insight 3: Testing components and improvement

I tested the components to make sure that the system I was planning would work at a conceptual level and could be iteratively developed. I worked out the amount of current that would be needed and how to develop the required checks and balances to ensure the system was accurate and reliable. I also tested the input interface on expected sensor inputs, automated timing events, and manual interrupts.

I made a small-scale model to test the concept using an online circuit simulator. I started thinking about how to control the flow to the irrigation lines. I tested the relay to ensure that if the water level was low, the pump would not be turned on.

I discovered that a power cut in the area would mean that the time would be lost and reset to 1 January 1970. In my third version, I set up a clock module with battery backup using an RTC (real-time clock) to make sure it would function reliably in the future.



#### Insight 4: Meeting end-user requirements

I tested the orchard control system and made sure that it worked for Auntie Jane, who was able to control the timing and manual settings via her cellphone using Serial Bluetooth. This allowed me to shift the focus of interface design from the electronics system to an iOS or Android app interface. Using MIT App Inventor allowed for quick prototyping of Bluetooth-enabled apps without the need for high levels of software coding knowledge.

Auntie Jane now could set the upper and lower limits of the dam-filling system and control the times when the pumps turned on. She checked to see if water was coming out in the rows she wanted to irrigate.



#### Insight 5: The choice of components and subsystems

I decided to use an RTC instead of moisture sensing to automate the timing of the irrigation system because the soil moisture levels varied across the orchard rows due to the contours of the land.

I decided not to add multiple sensors as a trade-off between accuracy and dealing with multiple data readings. After analysing the maximum water capacity of the dam, I modified the timing of the valve to avoid over-filling and over-emptying.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/>

[Technology/Digital-technologies](#)

DORJI™ is a trademark and brand of Shenzhen Youtu Technology Co. Ltd.

HC-12\* is a trademark of ARDUINO®.

LORA™ is a trademark of Semtech Corporation.

DS18B20™ is a trademark of Maxim Integrated Products, Inc.

BME280™ is a trademark of Robert Bosch GMBH.

Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc.

iOS™ is a mobile operating system created and developed by Apple Inc.

Android™ is a mobile operating system developed by Google™.

Copyright © Ministry of Education 2018, except for student work copyright © student

ISBN: 978-1-77669-251-4



PROGRESS OUTCOME 5

# Expanding the farm computer network

## Context

The school farm has been operating with a very basic computer in the shearing shed (see Snapshot 3). The farm manager now wishes to build a new computer network with:

- the capacity to print out data in a room not accessible to students
- a connection to the Internet
- two more computers with student access to the Internet
- managed access for selected student-owned devices.

The computer is in working order with a Linux®-based operating system and Open Office software. The farm has a networked connection.

In consultation with the farm manager and the school network manager, Georgia has been given access to computer equipment no longer needed by the school.



## Insight 1: Designing the solution

I sketched out a plan of the components, hardware, and software I would need to create a working network and how the devices would need to connect with each other. I discussed this with the network manager and farm manager.

I identified the need for a Wi-Fi-capable router connected to a switch that would connect the main computer with the other two student machines.

As students would also be accessing the network from their own devices, I needed to consider safe use. I researched the parts and components I would need. With guidance from the network manager, I plotted a simple network architecture diagram with possible subnet and gateway arrangements.



## Insight 2: Configuring the network components

I decided that the best way of allowing students to connect to the Internet would be through configuring access control on a Wi-Fi-capable router. I configured access to the network for the three computers on an Ethernet 4-port, Wi-Fi-enabled router. I set the name field and configured the speed duplex to '100 Mbps full duplex'. I set the layer 2 switching on and enabled lock access.

I also configured QoS (Quality of Service) priority for the port settings in the QoS list. I configured the IP addresses of the machines and pinged them to confirm connectivity. (I'd set up a DHCP server as I'd decided to use dynamic rather than static addressing.)



## Insight 3: Configuring peripherals and student devices

I researched printing options and decided that I would need to set up a shared printer to allow printing from the farm manager's desktop computer.

For students using their own devices, I configured the router to allow 'all connections', because the school network connection already filters the Internet and requires a sign-in.



## Insight 4: Testing

From feedback from the boarding hostel supervisor, I discovered that I needed to block boarders from bypassing the hostel network and connecting to the Internet via the farm network when they were not supposed to. I decided to record the MAC (media access control) addresses and block all connections outside normal class hours other than those selected to access the farm network manager's computer.

I had a problem connecting one of the student computers to the Internet when I had the network set up – the message was that there was a DNS error. I pinged the network and could not find the host, so I checked the host's file and the DNS settings. I flushed the DNS cache, and the problem was resolved. I determined that I must have had a bad result cached.



## Insight 5: Confirming fitness-for-purpose

Once the network was up and running and tested, I decided to review the network to ensure that, once I left school, it would continue to work and be able to be maintained.

In discussions with the network manager, I explained that I'd used a 4-port Wi-Fi router because it was the only one I could access that could be configured and restricted. If the network grows, it can easily be replaced and more computers added and configured quickly.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/Technology/Digital-technologies>

Linux® is the registered trademark of Linus Torvalds in the US and other countries.  
Apache OpenOffice™ is a registered trademark of The Apache Software Foundation.  
Copyright © Ministry of Education 2018, except for student work copyright © student  
The programs and resources used in this exemplar are not officially endorsed by the Ministry of Education.  
ISBN: 978-1-77669-252-1





PROGRESS OUTCOME 5

# Sandwich costings

## Context

The students in a year 12 class have been investigating the systems involved in the operation of a local small business, club, or organisation. They were asked to critically analyse existing systems and identify a need that could be met through some form of advanced digital design intervention. They then designed and developed a program (as an outcome) to respond to this need.

James has worked with a partner, Isaac, to develop a program to help the school canteen manager determine accurate costings for the canteen's sandwich range.



## Insight 1: Design decisions

Our discussions with the canteen manager showed up a need for accurate costing of a new sandwich menu with more flexible options. The canteen manager wanted to know how much each sandwich would cost taking account of different combinations of bread, wrap, fillings, salads, and sauces. He also needed to understand how much the final price should vary as the costs of the raw ingredients fluctuated.

Isaac and I created a program plan. We sketched a simple flow chart for the canteen's ordering system to help us understand the possible data structure. We also generated some pseudocode to identify the modules needed for our code.



## Insight 2: Developing the data structure

We developed the data structure to contain the required information that we'd identified through our planning. We then designed and tested the UI (user interface) to ensure we could collect the data we would need. Finally, we checked that our solution was flexible enough to include the variable costs and measurements of ingredients.

We had to ensure that our data structure would allow us to provide an average cost per sandwich, and that this could vary according to purchase costs and expected profit margins.





### **Insight 3: Iterative development and improvement**

As our design developed, we found we had to use modules with parameter passing and a GUI (graphical user interface) widget or buttons to make the program more accessible for the canteen manager.

We found a way to make the cost input more efficient, by exporting CSV data from the canteen manager's ordering system and importing it into the database for the program.



### **Insight 4: Meeting end-user requirements**

Isaac and I regularly consulted with the canteen manager. We developed the program and used a set of test cases to check its functionality.

During testing, the manager added that it would be useful to know the five top-selling combinations to help with stocktaking and reordering. We were able to identify top sellers and save them in rank order.



### **Insight 5: Testing the final outcome**

The canteen manager tested the system and found that it helped him identify when to pass on savings to customers as specials. He explained that knowing the best-selling combinations would allow him to reduce prices further. He was very pleased with the final program and said it had made his operations much clearer and easier to understand.



PROGRESS OUTCOME 5

# A kapa haka website

## Context

Kiri and Jasmine have been developing a website for the school's kapa haka group. They decided to share the tasks and each focus on their strengths in digital media to produce a better quality outcome. Kiri has worked on the coding of the website and the information structure, while Jasmine has focused on its appearance (layout and colour scheme) and creating and editing images, videos and sound. Working as a team has allowed them to address the wide range of stakeholders and considerations more effectively.



### Insight 1: Research on content (Kiri and Jasmine)

Before we could work on the design, we needed to interview the student and teacher leaders of the kapa haka group to learn what types of information and media they wanted the website to contain.

The discussion also revealed the need for English and te reo Māori versions of the site. We decided to develop our translations with students from the te reo Māori language class rather than rely on an automated translator, which still can't produce accurate translations for te reo Māori.



### Insight 2: Project management (Kiri and Jasmine)

Because we were working as a team, it was especially important to develop a plan for our outcome and find the best way to keep track of our tasks and progress. We decided to use Trello™ for project management because it is a free online tool that we could both access easily, even at home or after school. We could also allow our teacher and other stakeholders to log on and comment on questions we had.

We used Trello to set up key tasks in the order that they needed to be accomplished, to track our progress with them, and to reorder them when we struck obstacles. We also agreed to set up a Google™ Team Drive to keep version-controlled folders of our work, provide easy access to files and folders, and ensure we were backing up at all times.



### Insight 3: Design research (Kiri and Jasmine)

When developing the website markup and styling, we researched other schools' websites as well as national kapa haka competition websites. We had to ensure the site's look and feel would complement our school website and the kapa haka group's logo.

As we developed the design options, we worked together to ensure that they could be coded and the content easily displayed. Once we had narrowed the design options down to two, we presented them to the principal and the leaders of the kapa haka group for their feedback.



### Insight 4: Coding decisions (Kiri)

I decided against making two completely different versions of the site because the duplication of work when updating would be considerable and introduce a greater chance of errors. Instead I achieved two versions of the site using dynamic CSS (cascading style sheets) and JavaScript. I also decided to use a CSS Flexbox Grid plugin to ensure that the website was scalable for a range of devices.



### Insight 5: Video and photography (Jasmine)

I scheduled video and photography shoots according to the kapa haka group's performance schedule and practice times. I needed to ensure that I had permission to use the photos and video from the members of the group and discovered that the school office keeps a record of these.

Because videos of the performances would be a key element of the website, I focused on the lighting and sound equipment to ensure we achieved recordings of a high quality without too much background noise. I tested the recording set-up in kapa haka practice sessions. I then used this footage to test the editing features of the software, exporting files in different sizes and qualities to use in our website testing.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/Technology/Digital-technologies>

Trello is a trademark of Trello, Inc.

Google™ Team Drive is a registered trademark of Google Inc., used with permission.

Java is a registered trademark of Oracle and/or its affiliates.

Copyright © Ministry of Education 2018, except for student work copyright © student

The programs used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-254-5





PROGRESS OUTCOME 6

# Selling cellphone cases

## Context

A Young Enterprise group called Caseman has designed and made iPhone cases with a 3D printer. They have approached Hemi to help them create a system for producing invoices.

In consultation with Caseman's CEO (John Smith), Hemi has identified the information each invoice should show and that a database will be required to meet end-user requirements.



## Insight 1: Design decisions based on client needs

During discussion with John, I identified that the database would require a number of tables (e.g., customers, items, and invoices) to allow the group to keep track of each invoice. I had to consider how to enter the data, the information the database needed to produce, and who would be using this information.

I explained to John that using validation and drop-down lists based on already available information would minimise data-entry mistakes. John wanted to allow other members of the team to access information, but to avoid the duplication of orders (a problem that had already occurred), he requested that the database be set up so that only he could add new invoices. He agreed to the use of permissions and passwords to make this possible.

John also wanted to be able to create emails to offer new products or special offers to previous customers and to offer the specials to customers who had paid in full. I explained that this could be done using advanced queries and sending the results to his email program (Apple Mail).



## Insight 2: Developing the outcome

I started by identifying the information I needed for the tables. My solution could not be in the form of a flat file because a customer might want to order several different cases for family and friends. I decided to set up a relational database.

The main table for the database became the invoices table, where a record of each invoice is stored. It includes a primary key field (pk\_InvoiceID) that is unique and increases by one for each new invoice. I also included an items table, in which each item appears once with a specific price attached to it. If John needs to update pricing in his invoices, he only needs to change it in that table.

Because each invoice can have more than one item and any item can appear on many invoices, I needed to create a many-to-many relationship. I did this by creating a simple join table.

I linked the 'Invoice\_Item' field to the 'Item\_Name' field in items so that each line in the invoice would only be able to select an item available in the items list. 'Item\_Cost' is retrieved from the items list and populated with a simple script. Although not strictly necessary for creating invoices, I added a customer table to make it easy to create invoices for previous customers and to support the email function that that John had requested.



## Insight 3: Testing using data

Once I had designed the main invoice table layout, I tried adding some data to the system. I immediately realised that as well as looking up previous customers, John would need to add new ones. I included that option in the layout and tested it by adding several different records.

My display for the line items worked well, but I needed to add a delete button and a simple script to give the correct value for the price for multiple orders of the same item.

I also realised the need for a total, which I created with a summary field. I tested the invoicing system by selecting a name from the existing customer list, which updates every time a new customer is added.



## Insight 4: Data security measures

I then considered data security and privacy in the design.

To allow other members of the team to look at invoices but not create new ones, I set up two password-protected accounts. The one for 'staff' allows read-only access. The other account is specifically for John. It allows him to create and delete invoices but not to change layouts or scripts. Only I can do that as the designer, which protects the intellectual property that I have asserted over the design of this database.

Because both of these accounts use passwords, all the customer information and database details are kept private and secure.



## Insight 5: Setting up the auto-email

John was very pleased with the final design of the database and the ability to keep track of invoices and find customers quickly. I started work on the option to create a simple email that would send a promotional message to customers who meet certain criteria. This was much more complicated than I'd expected because the same customer can have more than one invoice.

I set up a field to count the total number of invoices a customer has in the system. I then created a new 'Paid' field for each invoice, which John selects when an invoice is paid. To complete the query, I created a calculation that compares the number of invoices a customer has against the number of 'Paid' entries. If they match, the field status is marked as 'True' and the customer is eligible to receive an email.

To generate the email that John wanted, I created a simple data-entry form for adding a message title and message content. I then created a script that would carry out the query and produce an email that merged customer information with the message.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/>

[Technology/Digital-technologies](#)

iPhone is a trademark of Apple Inc., registered in the United States and other countries.  
Apple Mail is a trademark of Apple Inc., registered in the United States and other countries.  
Copyright © Ministry of Education 2018, except for student work copyright © student  
The programs used in this exemplar are not officially endorsed by the Ministry of Education.  
ISBN: 978-1-77669-255-2





PROGRESS OUTCOME 6

# Designing a speaker box

## Context

The school's integrated course encourages specialist students to combine their expertise to develop multi-disciplinary outcomes. An end-of-year occasion attended by whānau, local business, and council members celebrates these outcomes.

Jarrold and Krish have been working across two specialist technological contexts: digital and resistant materials, and construction. During the early scoping aspect of their project, they surveyed the school's three senior year groups. The most popular request was for a small Bluetooth® speaker and amplifier for linking with students' phones.

Jarrold explored the idea further with some of the students and decided that the device could include an audio-reactive LED matrix. The matrix would run a number of visualisation modes to react to the music via a microphone inside the speaker box.



## Insight 1: Design and construction decisions

As the digital expert, I began to design and model a suitable electronic system. I met regularly with Krish, my construction partner, to ensure that we both understood the specifications and requirements. For example, we clarified the speaker-box measurements to make sure that the box could house an audio Bluetooth® amplifier board, an open source microcontroller (to control the LED matrix), a power supply and speakers, and a small, single-supply electret microphone circuit. We knew that to be fit-for-purpose, the outcome needed to look good and work well.

For the amplifier board, I sketched some circuit diagrams to show how I planned to build and control the amplifier, and I researched the components I might need and which would be suitable.



## Insight 2: Trialling components and construction

Firstly, I had to gather electronic components – the Bluetooth® module, LCD Bluetooth® module, and lithium polymer battery speaker. The amplifier components included left- and right-channel amp modules, a transformer, a switch and hardware materials.

Getting the correct speakers for the amp was a challenge. We learned that the load handled by an amp (expressed in Ohms) has to match the load of the speaker, so, for example, you need to use an 8 Ohm speaker with an amp rated at 8 Ohms.

I worked closely with Krish as he measured and cut out the various parts of the speaker box, such as the top panel for the push-button controls. As we added components, we tested all parts to make sure they were still functional (so that we didn't fry anything and everything was put together correctly before being powered up). We also investigated the potential draw on the battery.



## Insight 3: Configuring the network components

Potential challenges centred on aesthetic and functional factors. For example, the LED display was an important part of the complete unit and needed to look good. But I had to make sure it was earthed properly, as otherwise it might malfunction.

I sketched out a step-by-step plan as I worked through the best way to lay out the components inside the speaker. I also found I had to shuffle the components around to insert the LED baffle in the enclosure.

Once I had determined the layout, I had to connect the components from the JST PCB connectors.



## Insight 4: Testing the final outcome

Once we had installed everything, we tested the speaker with a range of phones. For our first test, we turned on the speaker and connected my phone to Bluetooth®. I wanted to test the sensitivity of the microphone because the LED panel responds to this sound source. I found I had to adjust the MIC\_HIGH and MIC\_LOW parameters in the code to improve how the panel would read and react to the sound.

We then tested the sound source with Krish's phone to make sure the settings worked on a different make of phone. Finally, some senior students tested the speaker box in a variety of settings. They reported that it worked well and looked great!

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/Technology/Digital-technologies>

Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. Copyright © Ministry of Education 2018, except for student work copyright © student. The program used in this exemplar is not officially endorsed by the Ministry of Education. ISBN: 978-1-77669-256-9





PROGRESS OUTCOME 6

# Farm connections: Further extending the network

## Context

The school farm has been operating with an established computer network in the shearing shed (see Snapshot 8). The farm manager now wishes to expand data collection and analysis across the farm to monitor soil moisture, humidity levels and water trough levels using recently purchased Wi-Fi-enabled technologies. The redesigned network must also further improve student safety and connectivity while ensuring that students are not able to 'hack' the system.

In consultation with the farm manager and the school network manager, Callum has been given access to computer equipment no longer needed by the school.



### Insight 1: Planning the outcome

This was a large job, so, in discussion with my teacher, we agreed that I would enhance the existing network to take on the Wi-Fi technologies and another student would develop the monitoring solution.

I sketched out a plan of components, hardware and software for a working network and how the devices would connect with each other. I discussed the plan with the network manager and farm manager. We identified that the network needed to recognise and communicate with a range of devices and that these would require a reliable way to connect to the network.



### Insight 2: Developing the wider network

I set up wireless devices to communicate with the network and configured the IP addresses assigned by the DHCP (Dynamic Host Configuration Protocol). I then found that the network topology was inadequate for this setup because the devices could not connect with a reliable signal.

I researched my options and decided that a different network topology was required because of the farm's unique environment. I carried out a site survey and began mapping, using Google Earth™ to place mesh nodes above water and away from animal interference.



### Insight 3: Refining my understanding

I realised that in order to have this large and complex network function reliably, I needed to consider the layout as a whole. I researched the OSI (Open Systems Interconnection) model and found that its goal is the interoperability of diverse communication systems with standard protocols. This model allowed me to structure my network within a seven-layer system so that others could easily track my process and setup.

I developed a schematic that showed the different parts of my project against these seven layers – that is, the physical, data link, network, transport, session, presentation, and application layers. I decided to set up a VLAN (virtual local area network) to manage this connectivity using Ethernet bridging.



### Insight 4: Implementing a mesh network

Now it was time to set up the mesh network that would communicate via different sensors situated around the farm. I set the sensors up and configured their settings to ensure connectivity. One sensor would not establish a connection, so I manually assigned an IP address to it. This worked, so I reset it to 'dynamic' and restarted the server, which resolved the issue.



### Insight 5: Testing the functionality

I decided to configure ICMP (Internet Control Message Protocol) on this network because it allows computers on a network to share error and status information. I wanted ICMP for network discovery and mapping and decided there was little risk of cascading ping floods, so I changed the configuration setting to allow for this.

I used a variety of tools and techniques to make sure that the network was functioning as it should. I ensured that all the IoT (Internet of things) clients were sending data to the server, and I checked aerials, power and water tightness on devices that were not sending data.

All mesh nodes were operational and communicating. I completed some simple diagnostics, including powering components off and on to make sure that the network would still function if one of the nodes went down.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/Technology/Digital-technologies>

Google Earth™ is a registered trademark of Google Inc., used with permission.  
Copyright © Ministry of Education 2018, except for student work copyright © student  
The programs and resources used in this exemplar are not officially endorsed by the  
Ministry of Education.  
ISBN: 978-1-77669-257-6





PROGRESS OUTCOME 6

# Designing a flexible booking system

## Context

The school catering department holds an annual fundraiser in support of local charities. The department has asked Abby's digital technology class to create a program to manage the sale of tickets and collection of revenue and to produce PDF tickets electronically. The program needs to produce a spreadsheet giving the numbers of meals required for one lunchtime and two evening sittings, customers' specific dietary requirements, and their table and seat numbers.



### Insight 1: Design decisions

In planning the program, I used a flowchart and pseudocode to include all of the required functionality. As there were many variables, I decided the best solution was a program that included a GUI (graphical user interface) easy for the person taking bookings to use and that would generate the information needed by the caterers.

My aim was to make the user interface as simple and inviting as possible to ensure a smooth booking process. Inputting the data was going to be tricky to organise because it contained multiple variables and input events. I discussed my initial plans with the catering manager to check if I'd overlooked anything.



### Insight 2: Writing the program

As I began writing the code, I took account of the caterer's wish to be able to reuse the program for subsequent events. This meant I needed to use a modular design to add flexibility. I had to battle early on with the reset facility because an incomplete booking could give an incorrect remaining number of seats. I corrected this after finding an error missing from the reset code that would have stopped the program from running.



### Insight 3: Refining and trialling the program

As the fundraiser is for charity, I wanted the program to give the person buying the seats a couple of opportunities to confirm they were happy with their purchase. I also decided to add a countdown of remaining seats on the GUI to encourage buyers to tell their friends they were selling out. I tested the program with classmates to ensure that seats couldn't be double-booked and that it was easy to use.

In the final testing session, the catering manager noticed that the PDF tickets didn't include the seat number. I had missed this error when running my planned test cases, which showed that it's important to test a system with its end-users. I was happy that I could fix this error after the catering manager discovered it.



### Insight 4: Evaluating the final outcome

Our debugging had fixed all the errors, so the final program worked efficiently.

Being able to reuse the program means other groups in the school can also use it. The school production team has asked for my help to set it up for their performances. In the next version I'll include a function to check if there are enough tickets left each time a purchase is made, and to provide feedback if there are not. To encourage ticket sales, I'll also add the ability to automatically email a link to event information on the school website to a buyer's friends when the buyer is making a booking.



PROGRESS OUTCOME **6**

# Monitoring wildlife

## Context

For his digital technology project, Finn has elected to design and build a system to help a local ecosanctuary monitor wildlife. The system needs a web-based interface so that staff and volunteers can remotely monitor the enclosures and a graphical function to make the best use of the data.



## Insight 1: Research and end-user consultation

In order to understand the requirements and purpose of the project, I met with a range of end-users and stakeholders, including the primary manager of the site, the volunteers who help manage the wildlife, and an electronics expert who understands electronic systems in outdoor environments. I also needed to visit the site to understand the physical requirements and constraints.

From my research, I realised that my system had to be accessible for people such as the volunteers at the ecosanctuary, so it needed to have a user-friendly interface and be easy to maintain.

I could see I would need to integrate specialised knowledge from a range of digital technologies, including electronics, programming, databases, and web coding, in the development of my outcome. For example, I needed to develop a way to graph the data collected by the system. Graphing shows visible trends in data over time and would help the ecosanctuary staff to analyse the data collected.

I researched some existing graphing-interface solutions, using GitHub and YouTube, to see how they were designed. I decided to use HTML5 to create the graphs because it's up-to-date, fast, and secure. I wanted to build something based on current technology to future-proof the system as much as possible.



## Insight 2: Key design decisions

I decided to use online project-management and code repositories to manage the project and version control.

When developing the hardware and software for the website's front and back ends, I considered:

- which devices and alert representations would best suit the volunteers
- how to develop a system that would suit a volunteer organisation with limited funding
- how to set up the system so that it could be maintained and developed in the future without incurring major costs.

I decided that my platform would be released as open source and that I would only use open-source software to develop the outcome. This was both an ethical and a cost-saving decision. I also decided that all of the components should have a modular structure to make it easy to add new alerts or monitoring in the future.



### Insight 3: Building and testing the system

This system is for use in monitoring outdoor areas, so my solutions had to be weatherproof. I tested different casing options to find one that would be both inexpensive and watertight.

I had to work out how to transfer data from the sensors to the central system in an outdoor environment in which network wires are impractical because of the distances involved. I ran a series of tests for radio range and frequency propagation over terrain to determine the impact that weather or radio interference would have on reception.

I configured the software to trigger the alert system if a gate was triggered and had been open too long. Data was sent via Picaxe using a RADIUS (Remote Authentication Dial-In User Service) server that I also configured.

To determine if gates were open or closed, I first tried a Hall Effect Sensor. However, a pin broke off the HES I was using, which raised concerns over the delicacy of the device and how to secure it to a gate.

I decided that a more robust solution was a magnetic reed switch. I drilled the magnet and embedded it in the gate, and used the same poking-through-wood approach for the sensor. The Picaxe was protected in a plastic container attached to the side of the fence post, making it both decent looking and weather resistant.

I knew that the application needed to be user-friendly, so it was important to get the interface to work well on mobile devices to provide quick and easy access to data. After some research, I chose a system that uses two cascading style sheets (style.CSS and mobile.CSS) for the different screen sizes of computers and cellphones.

This was far easier than having two different websites and meant I could manage them both from one codebase. Also, the ability to download data makes it easier to analyse (for example, in custom programs). To allow this, I created a PHP script that automatically gets the relevant data from MySQL and puts it in a CSV file.

Downloaded from <http://technology.tki.org.nz> or <http://seniorsecondary.tki.org.nz/Technology/Digital-technologies>

GitHub® is an exclusive trademark registered in the United States by GitHub, Inc.

YouTube™ is a registered trademark of Google Inc., used with permission.

MySQL™ is a trademark of Oracle and/or its affiliates.

Copyright © Ministry of Education 2018, except for student work copyright © student

The programs used in this exemplar are not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-259-0

