INSIGHTS
FOR TEACHERS

Technology 2021

The Algorithmic Element of Computational Thinking

NMSSA
Wānangatia te Putanga Tauira
National Monitoring Study of Student Achievement

- **Providing students with opportunities to develop sequenced instructions engages them in algorithmic thinking.**
- **Recording algorithmic thinking clearly and concisely builds foundational coding skills.**

Important ideas for teachers about algorithmic thinking

- Algorithmic thinking is one of the six components of computational thinking. It involves developing step-by-step processes to solve problems. Computational thinking is broader than algorithmic thinking and also involves abstraction, decomposition, generalising, evaluation, and logic.
- Computational thinking can be described as the thought processes involved in defining a problem and expressing its solution in a way that enables it to be carried out. The solution is often a process, or set of instructions to be followed. Computational thinking can be carried out with or without computers.
- Computational thinking is different to programming. When you make a computational process happen on a computer, that's called programming. In order for the computer to understand the step-by-step instructions, they need to be written in code. The computer reads and processes the code, to carry out the instructions. Sound computational thinking prior to programming, leads to outcomes which are more likely to work well for the problem they're designed to solve.

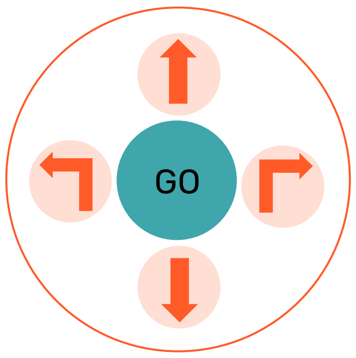
Evidence from NMSSA tasks

260 Year 4 students from 51 schools undertook a programming task called *Robot on the Move*.

The task involved working with a simple programmable robot, in a one-to-one interview situation with a trained Teacher Assessor. Students used an 8 x 3 grid, which showed iconic sculptures from around New Zealand, such as the 'sheep' in Omarama and the 'cow' in Morrinsville. The task had two components. Firstly, students were asked to program the robot to travel to three of the sculptures, from specified locations. The programs became increasingly complex. Secondly, students were asked to write some instructions for the Teacher Assessor to give to the robot to travel to the Crayfish in Kaikōura.

Before starting the task, students were asked if they had used "robots like these" before. They were also shown the buttons on the robot and given some time to experiment with them.

The robots were controlled using buttons, which each represented a command. Commands included forward, backward, 90° left turn, 90° right turn, and start the programme.



Had students worked with programmable robots before?

48% of Year 4 students reported that they had used "robots like these" before.

This is noteworthy, given the New Zealand Curriculum (NZC) expectation for students to "give, follow, and debug simple algorithms in computerised and non-computerised contexts" by the middle of Level 3 (CTDT, Progress outcome 2).

Could students program increasingly complex paths?

- **Most students could program a simple straight path, even those who had not used a programmable robot before.**

  Ninety percent of students who had not used a robot before successfully programmed the simplest straight path to the paua shell, and this percentage was slightly higher for students who reported that they had used a robot before (91%).

- **In general, as the paths increased in complexity, fewer students were able to successfully complete the program.**

  For example, of the students who had not used robots before, 90% were able to program the simple straight path to the paua, 73% were able to program the path to the sheep which involved one turn, and 42% were able to program the path to the cow which involved two turns and required the robot to move both forwards and backwards.

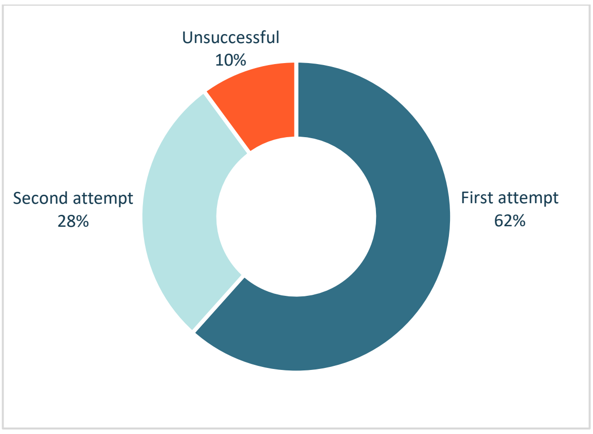
- **Students who had used programmable robots before were more likely to successfully program each of the paths than students who had not used the robots before.**

  For example, the path to the sheep was successfully coded by 90% of students who had used robots before, and 73% of students who had not used robots before. Additionally, students who had used robots before were more likely to complete the task on their first attempt.

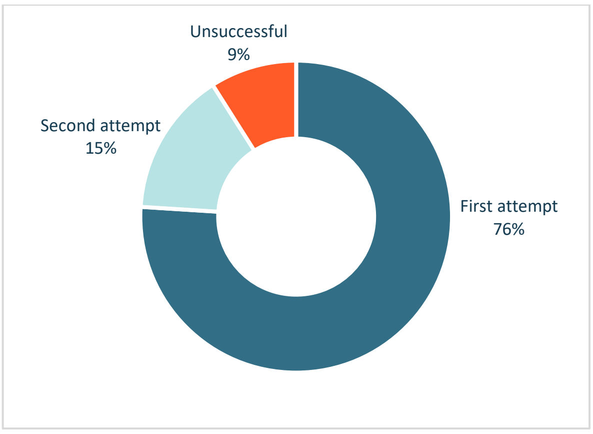
Paua Shell

Solution: forward 7

Not used robots before



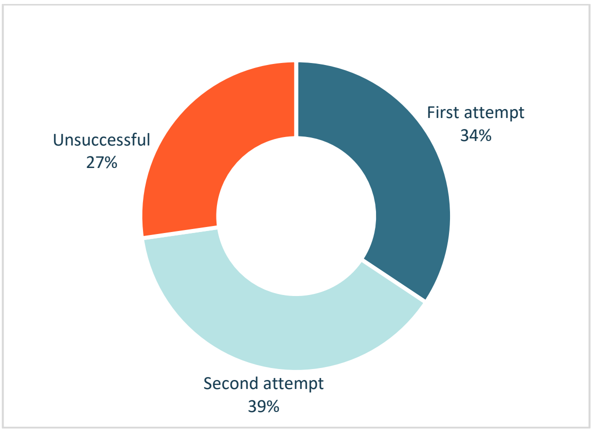
Used robots before



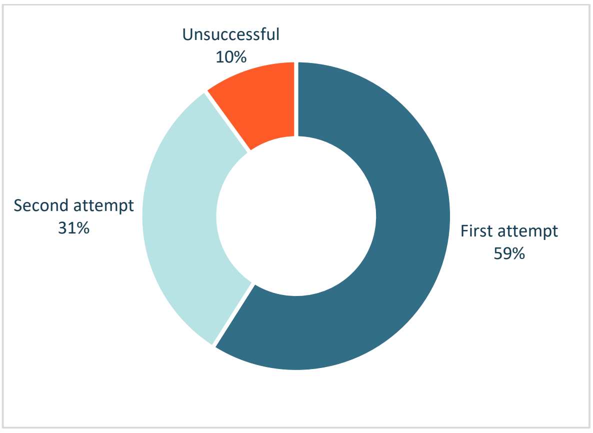
Sheep

Solution: forward 3, turn left, forward 1

Not used robots before

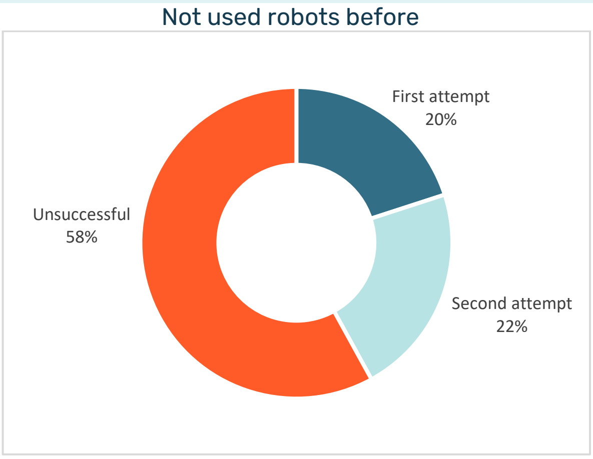


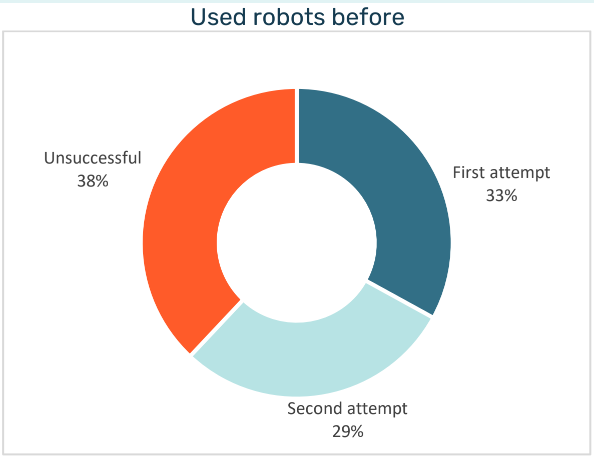
Used robots before



Cow **Solution:** back 1, turn right, forward 4, turn left, forward 1

◀ ↴ ▶ ▶ ▶ ▶ ⬏ ▶





How did students record sequenced instructions?

In addition to assessing students' ability to program a range of paths, the analysis also described how students recorded the sequence of instructions. Both factors were evaluated separately from each other. Students’ responses were able to be grouped into four increasingly sophisticated categories, which are described and exemplified overleaf.

The least sophisticated recordings included narrative descriptions and mapped paths, and were used by just under a third of students. About half of the students used words, arrows, or letters to record instructions, using one term for every instruction. Just over 10% of students were able to recognise repeated instructions and record these using numbers, and a further 10% could record instructions concisely, using repeats.

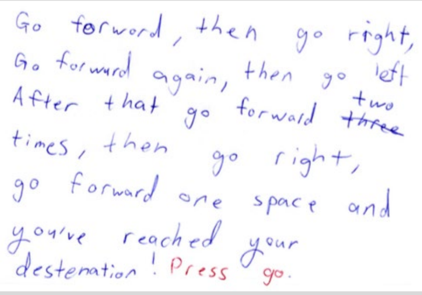
There was little difference in students’ ability to record sequenced instructions clearly and concisely, whether they had used a robot before or not. This may indicate that students who had used “robots like these” before hadn’t necessarily had experience with recording step-by-step instructions in written form.

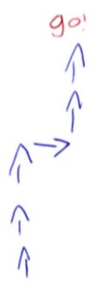
- Findings suggest that students who had used “robots like these” before hadn’t necessarily had experience with recording step-by-step instructions in written form. This is noteworthy because recording algorithmic thinking clearly and concisely builds foundational coding skills.

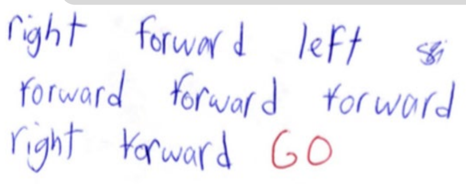
Algorithmic Thinking

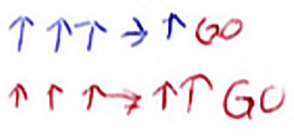
students' recording

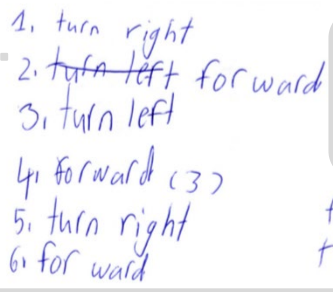
	Not used robots before	Used robots before
1. Records a sequence of instructions as **narrative text**, or a **mapped path**.	29%	27%
2. Records a sequence of instructions, using **one word, arrow or letter for every instruction**.	52%	51%
3. Records a sequence of instructions, recognising where the same instruction is carried out more than once, and **recording these as repeats**.	10%	15%
4. Records a sequence of instructions **concisely, using repeats**.	7%	10%

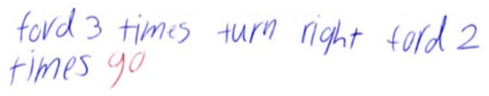

Go forward, then go right, Go forward again, then go left After that go forward ~~three~~ two times, then go right, go forward one space and you've reached your destenation! Press go.

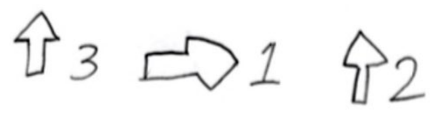


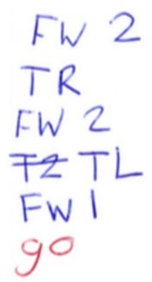

right forward left
forward forward forward
right forward GO




1. turn right
2. ~~turn left~~ forward
3. turn left
4. forward (3)
5. turn right
6. for ward


ford 3 times turn right ford 2 times go




FW 2
TR
FW 2
~~T2~~ TL
FW 1
go

Helpful resources for understanding and teaching algorithmic thinking

- The Kidbots section of the CSUnplugged website includes a selection of classroom activities designed to develop computational thinking, alongside other fundamental programming skills. The activities are suitable for students in Years 1 to 6 and include things to notice as students are working, and descriptions of the key important ideas involved.
- The webpage What is Computational Thinking? provides a clear description of computational thinking, written for teachers, and includes descriptions of all six elements of computational thinking. This information page is part of the CS Unplugged website.
- The SET article Computational thinking is more about humans than computers records an interview with Professor Tim Bell, who was involved in the introduction of digital technologies into the New Zealand Curriculum in 2018. Tim talks about why teachers need to consider computational thinking, and describes his long-standing work on teaching computational thinking effectively.