A programming journey

through the Digital Technologies Hub Scope and sequence

Leanne Robertson
Program Director, DT Hub
Leanne.Robertson@esa.edu.au

DIGITAL TECHNOLOGIES HUB

DIGITAL TECHNOLOGIES HUB

Home Teachers School Leaders Students Families

Search

www.dthub.edu.au

WELCOME TO THE DIGITAL TECHNOLOGIES HUB

Unpack the Digital Technologies Curriculum one step at a time. Find great lesson ideas linked to the curriculum, explore strategies and advice from Australian primary and secondary schools and more.

LEARN MORE ABOUT DIGITAL TECHNOLOGIES

TEACHERS

STUDENTS

CSER Digital Technologies Education

THE UNIVERSITY of ADELAIDE

CSER Digital Technologies Education

CSER Digital Technologies Education

About Us

Available MOOCs

Lending Library

Professional Learning

PL-in-a-Box!

Research

Resources

FAQs

Newsletter Subscription

Digital Technologies Education Programs

We run a range of Digital Technologies Programs for Australian teachers, including our free, online CSER MOOC courses, free professional learning events, and our National Lending Library.

Available MOOCs

Lending Library

Professional Learning

Resources

Newsletter Subscription

Stay up-to-date by subscribing to our eNewsletter.

Name

Email

Subscribe

CSER Adelaide Retweeted

Tina Photakis
@tina_p

. @SATeachLearn @TRB_SA @moons064 @cserAdelaide @DECDTeacherLead @DECD_ECT @Edufolios @EduTweetOz @EdTechSA

1h

CSER Adelaide Retweeted

Anne McIntosh
@mrsamcintosh

STEM challenge session @Ozobot 100cm challenge. Thanks @cserAdelaide for the loan of kit. Students have loved it. #engaged #STEMeducation

CSER F-6 Digital Technologies: Foundations

Digital Technologies involves learning about how we can create new technologies, as well as use them. This course will explain the fundamentals of digital technology and computational thinking specifically addressing the content descriptors and achievement standards of the Australian Curriculum: Digital Technologies (Foundations to 6).

Join us to learn about how digital technology can be integrated into your classroom, exploring example lesson plans, and helping form a community designed to share resources and support!

If it is your first time here, please enrol in the course by clicking the "Register" button below. If you have already registered, you may need to Log-in again by clicking on "Login" in the top, right-hand corner.

Professional Learning Certificate available (mapped to AITSL standards)
Free and open to all

This project receives funding from the Australian Government Department of Education and Training. In addition, the development of this course is supported by:

Google

Register

Spark your child's curiosity

Search the STARportal for activities

5000 Start your search

Activities Near You Now

Available Nationally

STEM Pack 7 - Robotics

Each STEM Career Pack provides teachers with a sequential set of tasks and information, including...

FREE
IN CLASS

Available Nationally

STEM Pack 6 - Art Conservation

Each STEM Career Pack provides teachers with a sequential set of tasks and information, including...

FREE
IN CLASS

Available Nationally

STEM Pack 5 - 3D Printing

Each STEM Career Pack provides teachers with a sequential set of tasks and information, including...

FREE
IN CLASS

Available Nationally

STEM Pack 4 - Bringing Engineering to Life

The aim of each STEM Careers Pack is twofold: first, to make students aware of the wide variety of STEM careers there...

FREE
IN CLASS

See More

Australian Computing Academy

Helping teachers implement the Australian Curriculum: Digital Technologies

Unpack the curriculum

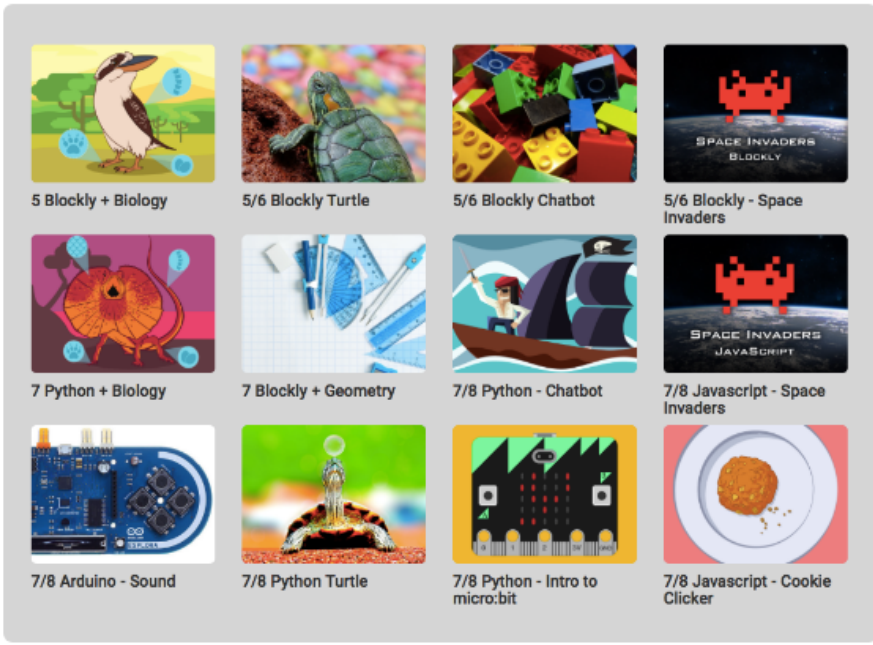
Australian Digital Technologies Challenges

New! Free online access for Years 5 - 8

The *Australian DT Challenges* are free in-classroom activities designed to address the most technically challenging aspects of the Year 5-6 and 7-8 bands of Australian Curriculum: Digital Technologies.

Each Challenge provides online and unplugged learning resources; engaging, authentic, real-world problems; modular lesson plans; and online training and support for teachers.

Years 5 - 8 students have free access to a learning platform that enables self-paced learning with immediate, intelligent feedback.



View all of the Challenges

DIGITAL TECHNOLOGIES HUB

What's the difference between ICT Capability and Digital Technologies?

http://bit.ly/ICTvsDT

Information Communication Technology (ICT) Capability

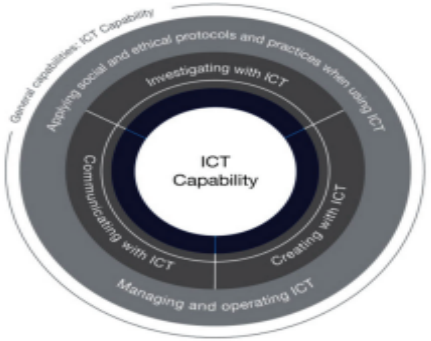
A general capability taught within all curriculum areas for students in years F–10.

Develops skills and understandings in managing and operating ICT to investigate, create and communicate.

Incorporates digital citizenship when considering the ethical and social impacts of using technologies.

Is explicitly planned and taught in all subject areas.

ICT supports students to be effective users of technology.



Digital Technologies

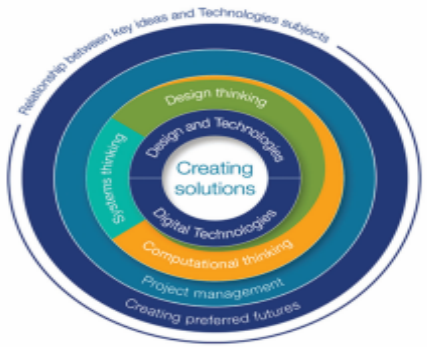
A new subject for F–10 (optional in 9-10) students with new and unique skills and content.

Develops knowledge, understandings and skills of the underlying concepts of information systems, data and computer science.

Encourages students to design and create digital solutions that solve problems taking their preferred futures into consideration.

Must be assessed and reported at least once every two years.

Digital Technologies build on and extend ICT, moving students from technology consumers to creators.



Use ICT

- Presentation tools
- Locate information
- Spreadsheets and graphing
- Digital publishing
- Interpret timelines
- Analyse and visualise data
- Ownership and use
- Managing files
- Cyber safety
- Mapping and geospatial tools
- Online communication
- Digital music / multimedia

Create solutions and learn about Digital Technologies

- Digital systems (networks)
- Robotics and automation
- Coding and programming
- Computational thinking
- User interface design
- Storing and transmitting data (binary numbers)
- Pattern recognition
- Algorithms
- Programming boards
- Data collection

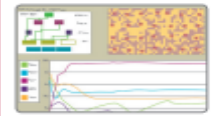
Examples of ICT in action

Use digital concept mapping tools to plan and select research tasks.

Use presentation software to present findings of an inquiry that includes text, images and video.

Use video to analyse a sports performance to provide coaching tips.

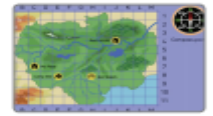
Use a computer simulation or game to test predictions and collect data.



Use a search engine effectively as a research tool.

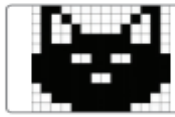
Use spreadsheet functions to create tables, record, sort, calculate and present data to identify trends.

Use an online game that has a grid map system to learn about directions.



Examples of Digital Technologies in action

Create and code an image using black and white squares. Invite a classmate to decode and recreate the image.



Compare a transport network and computer network to explore ideas about pathways, reliability, protocols and security.

Create an interactive story with user-input using a familiar programming language.

Create your own simulation using a visual or text-based programming language.

Explore ways to securely transmit data through techniques of encryption and decryption.

Create network diagrams to identify relationships between different sources of data (eg friends on social media) and analyse this data.

Design your own maze and use an app to program a robot to go through it.

Assessment of ICT Capabilities

Identify the impacts of ICT in society: identify how they use ICT in multiple ways on multiple devices.

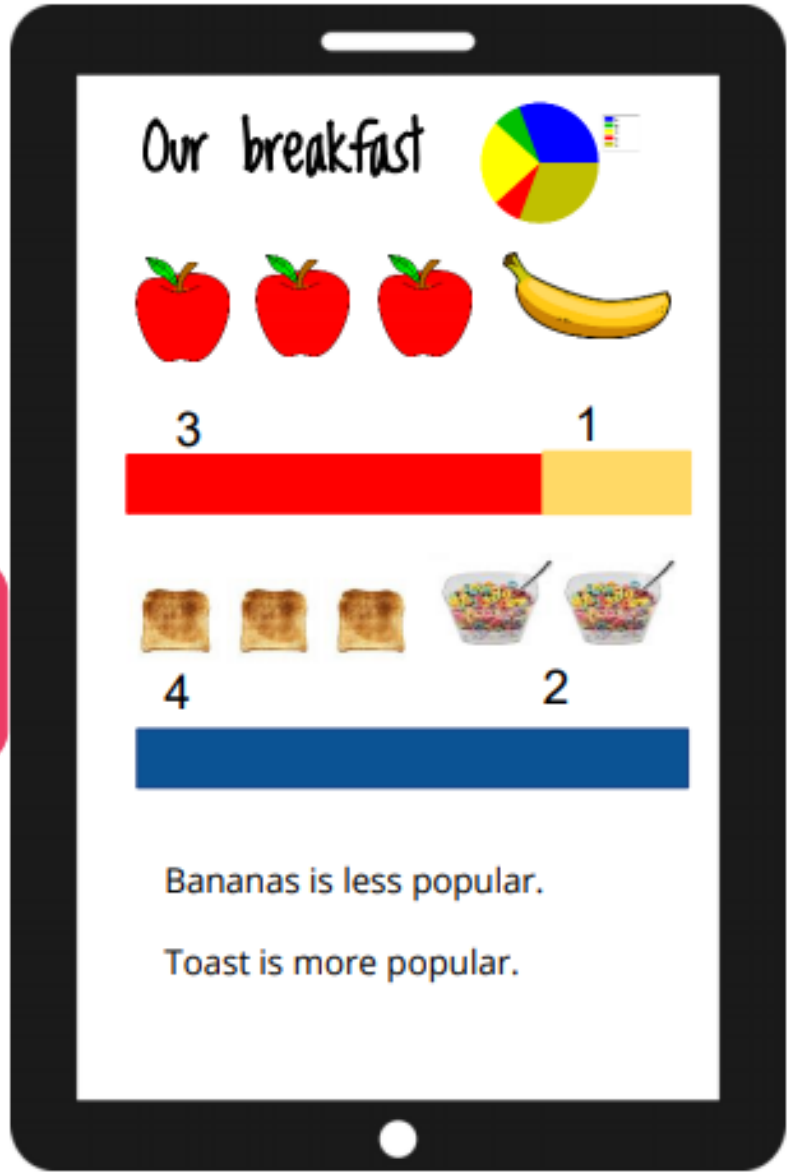
Understand ICT systems: identify common consumer ICT systems with input and output functions.

Generate ideas, plans and processes: use ICT to prepare simple plans to find solutions or answers to questions.

Generate solutions to challenges and learning area tasks: use ICT as a creative tool to generate simple solutions, modifications or data representations for personal or school purposes.

Select and use hardware and software: identify and safely operate ICT systems to complete relevant simple specified tasks and seek help when encountering a problem?

Manage digital data: Save and retrieve digital data with support



Assessment of Digital Technologies

Recognise and explore digital systems (hardware and software components) for a purpose (ACTDIK001).

Recognise and explore patterns in data and represent data as pictures, symbols and diagrams (ACTDIK002)

Collect, explore and sort data, and use digital systems to present the data creatively (ACTDIP003)

Create and organise ideas and information using information systems independently and with others, and share these with known people in safe online environments (ACTDIP006)

Note: This example of some ways to explore assessment, based on a Year 1 year level: drawing on the Level 1 ICT Capabilities examples and Band F-2 Digital Technologies.

Rebecca Vivian & Katrina Falkner, Computer Science Education Research (CSER) Group, The University of Adelaide

FILTER SCOPE AND SEQUENCE BY

YEAR LEVELS

F-2	3-4	5-6	7-8	9-10

This resource provides a possible set of sequenced topics that could be used in teaching the Australian Curriculum Digital Technologies curriculum to address the content descriptions of the curriculum.

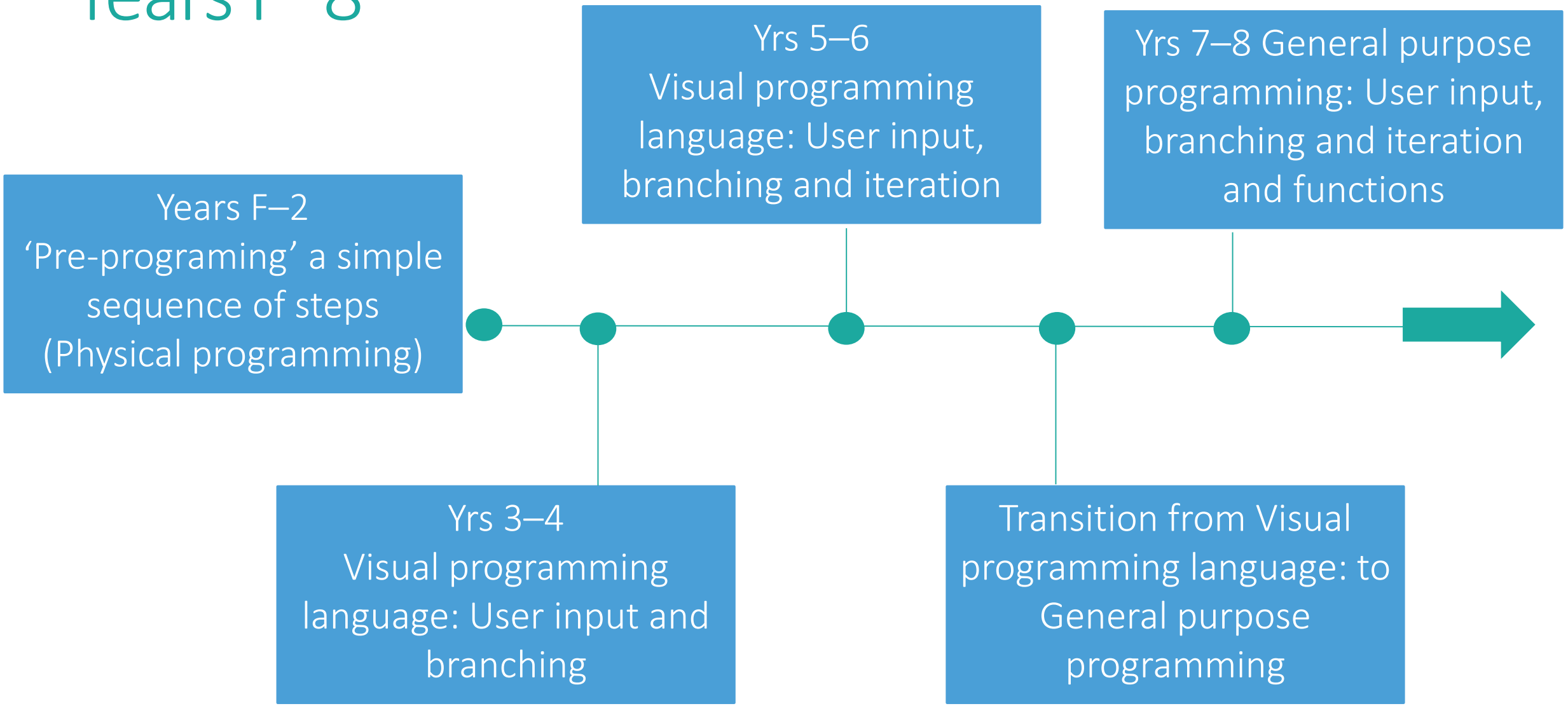
Units are organised under relevant topics for each band with an overview and a visual map of the content descriptors and key elements. Each unit is organised into a sequence of four key elements with a summary of the key focus and what to teach, targeted supporting resources, assessment advice and support for differentiation through the provision of a matrix based on the SOLO taxonomy.

Note: the suggested supporting resources are presented as a sample of targeted resources rather than an exhaustive list. Teachers can adapt the sequences adding their own relevant learning activities as required.

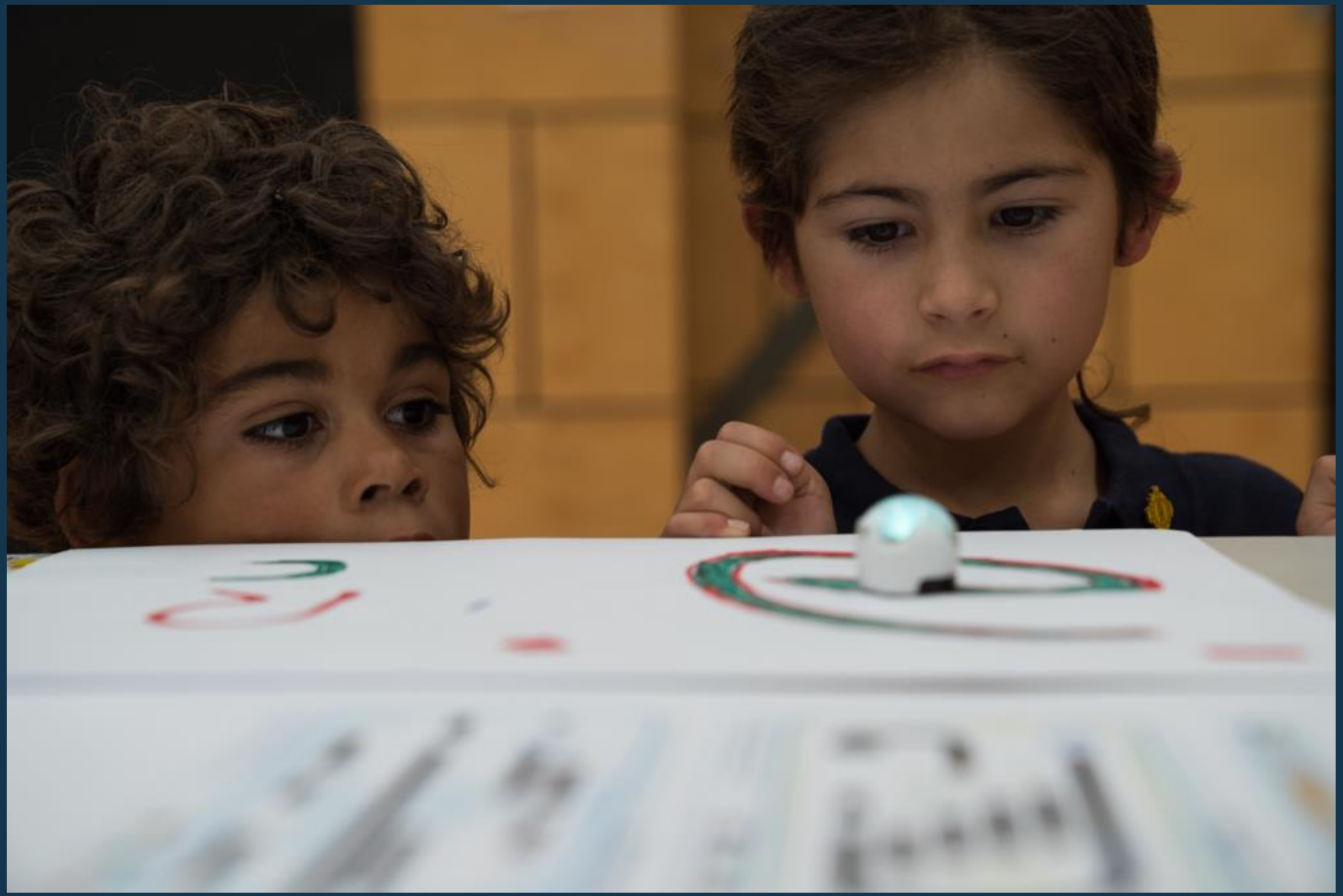
To view the topics and supporting units, select a year level and then a topic of interest.

F-10 Overview

Programming (Producing and implementing) Years F–8



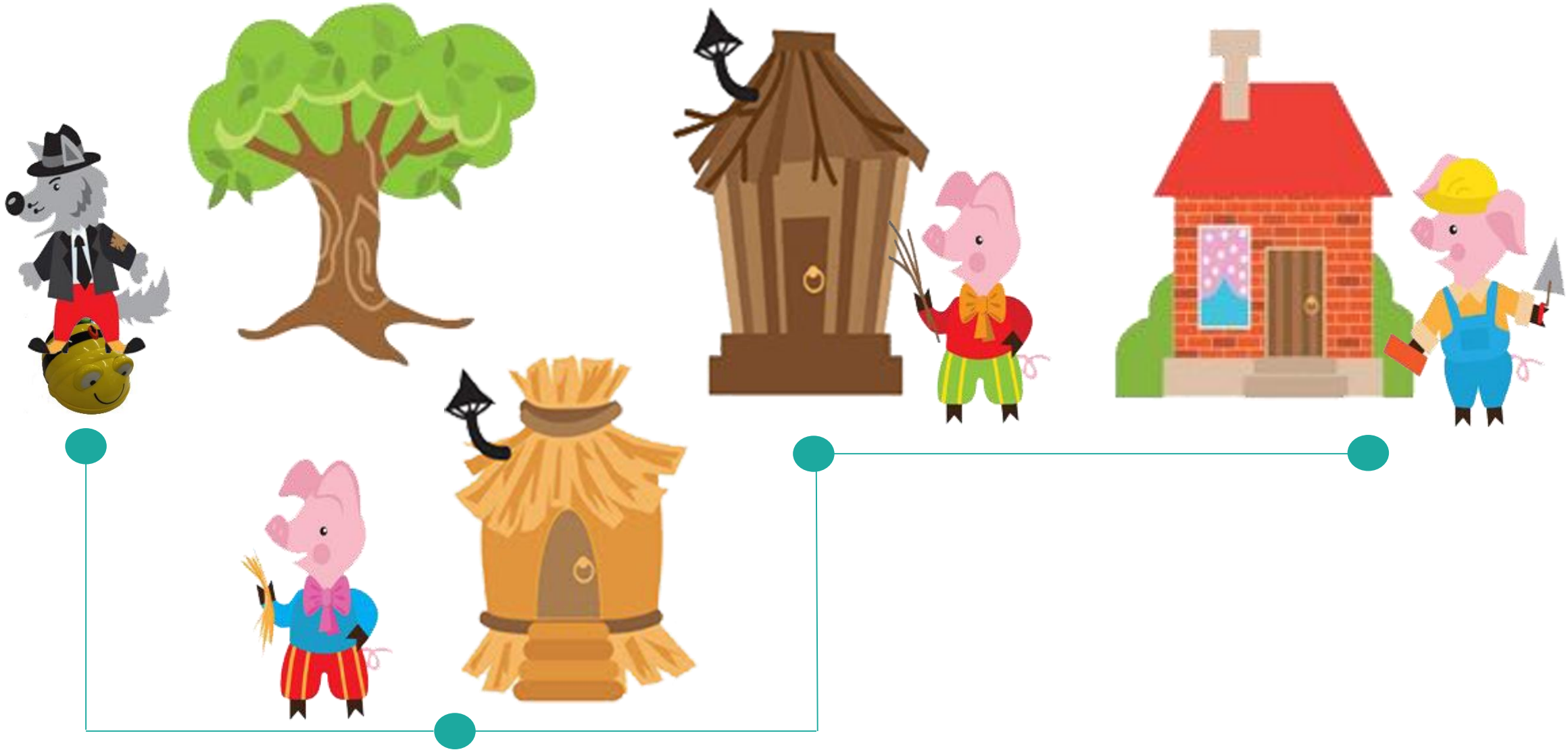
Eve
Year 1–2



Retell: The Three Little Pigs using Bee Bot

Physical programming

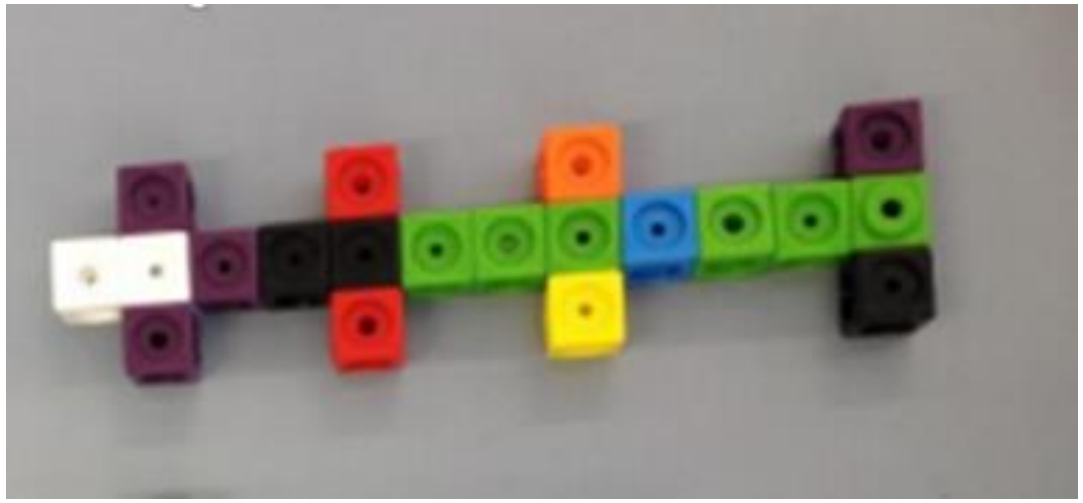
Program Bee Bot: Create a story map

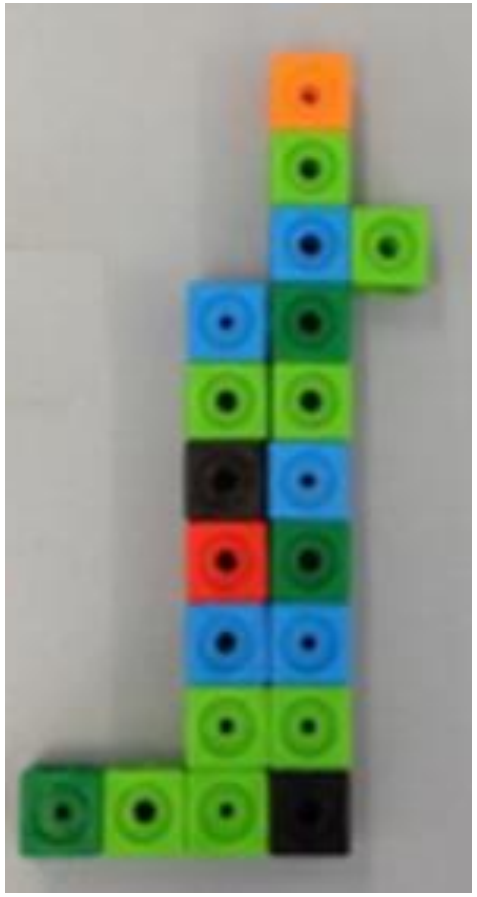


Computational thinking

Abstraction:	pull out the important parts of the story
Algorithm:	sequence steps in the correct order
Pattern recognition:	recognised patterns (parts of the story that are repeated)
Decomposition:	decomposed the problem into smaller parts
Evaluation:	did your directions work (end up at the right house in the correct order)?

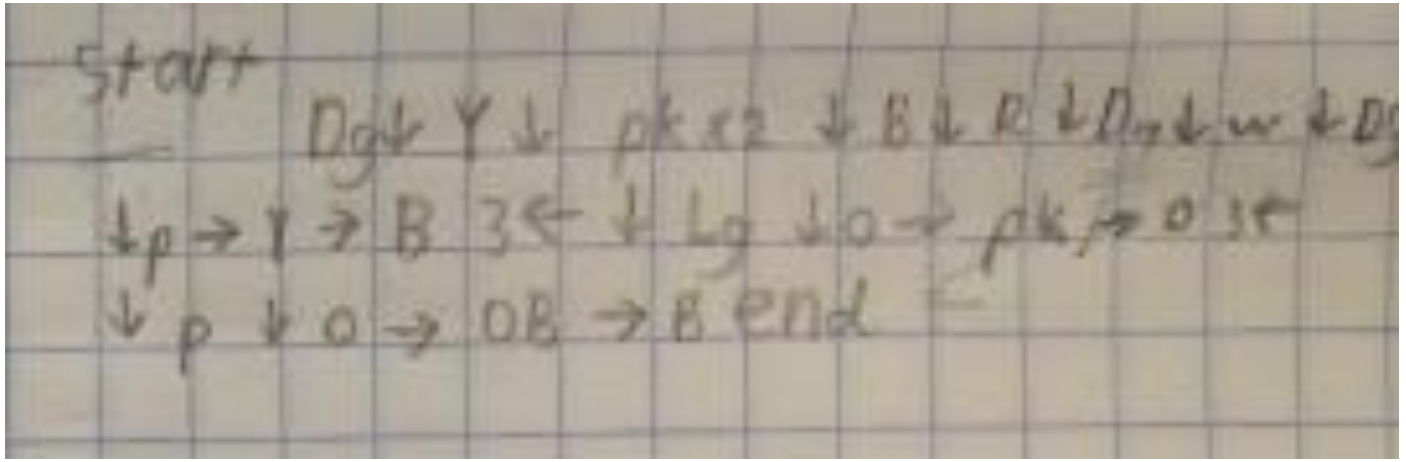
Create a program to build a unifix model





Colour key

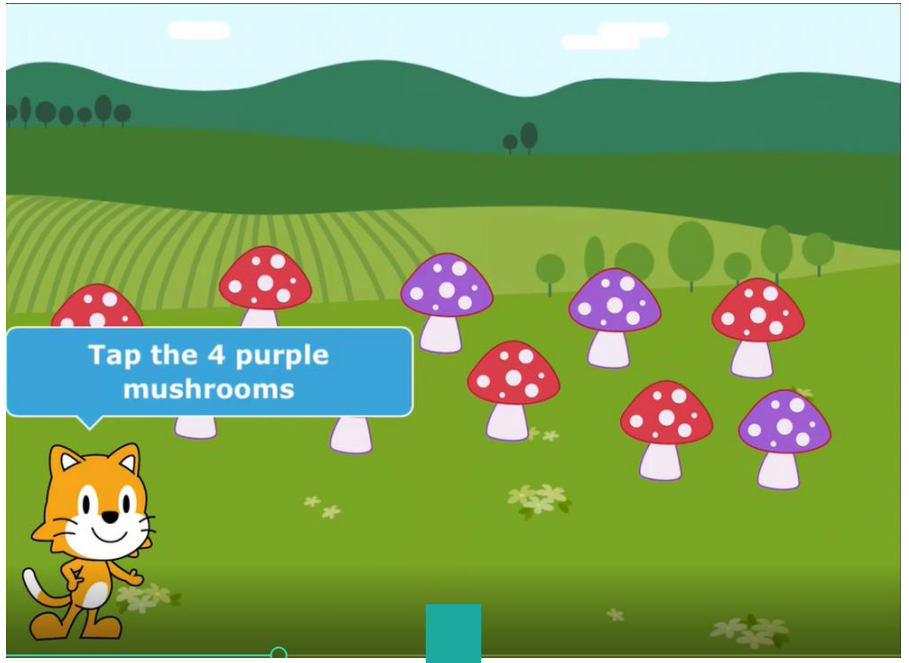
Coding language

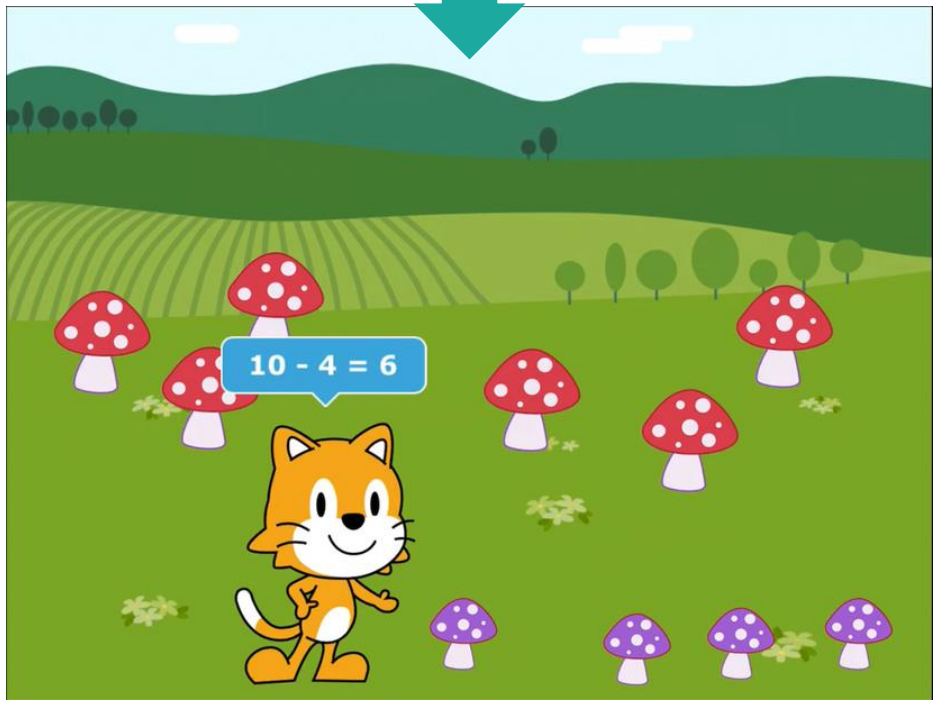


Credit Jackie Tither

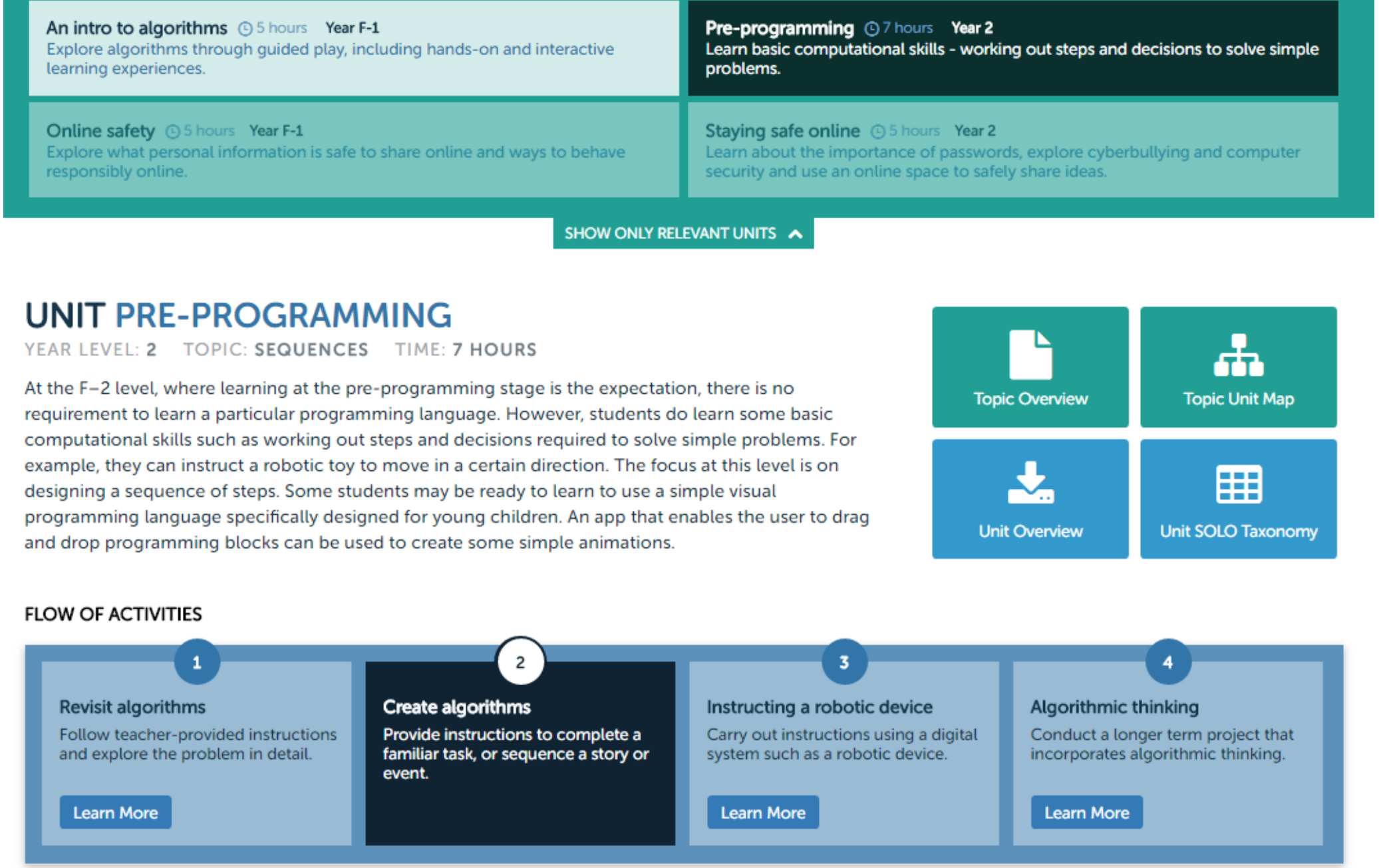
Intro to visual programming

(not a requirement at Yr 2)





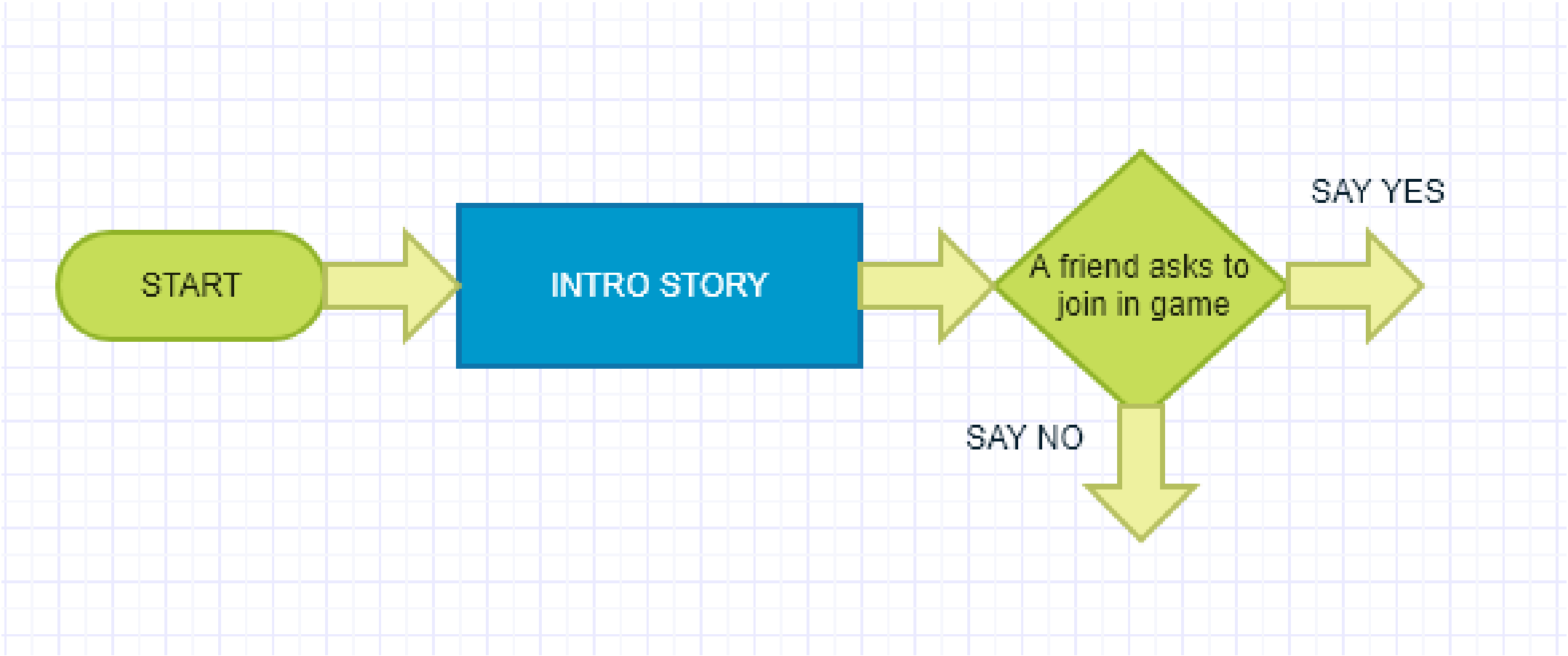
Credit Bev Babbage



http://bit.ly/DT_F-2

Jessie
Year 4

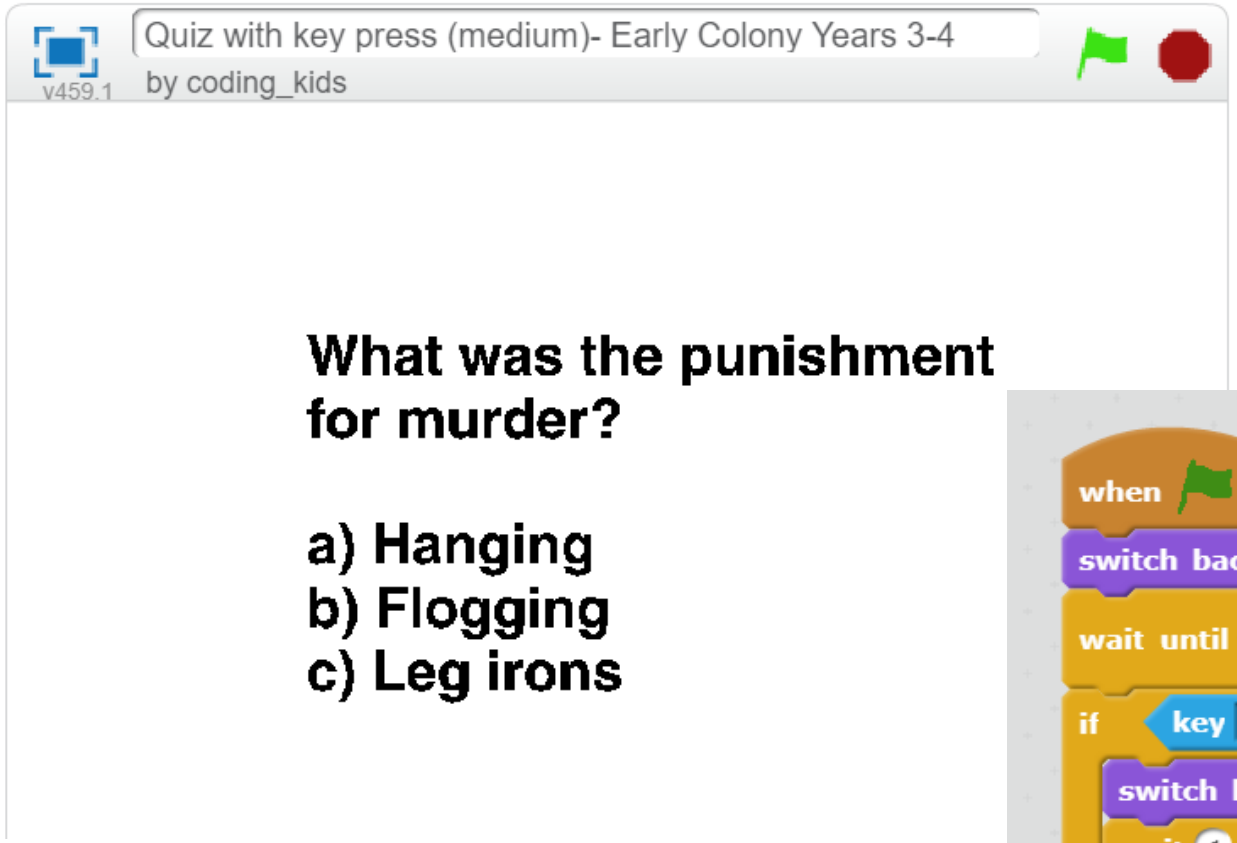
Branching: (decision making)
Choose your own adventure story



Design process

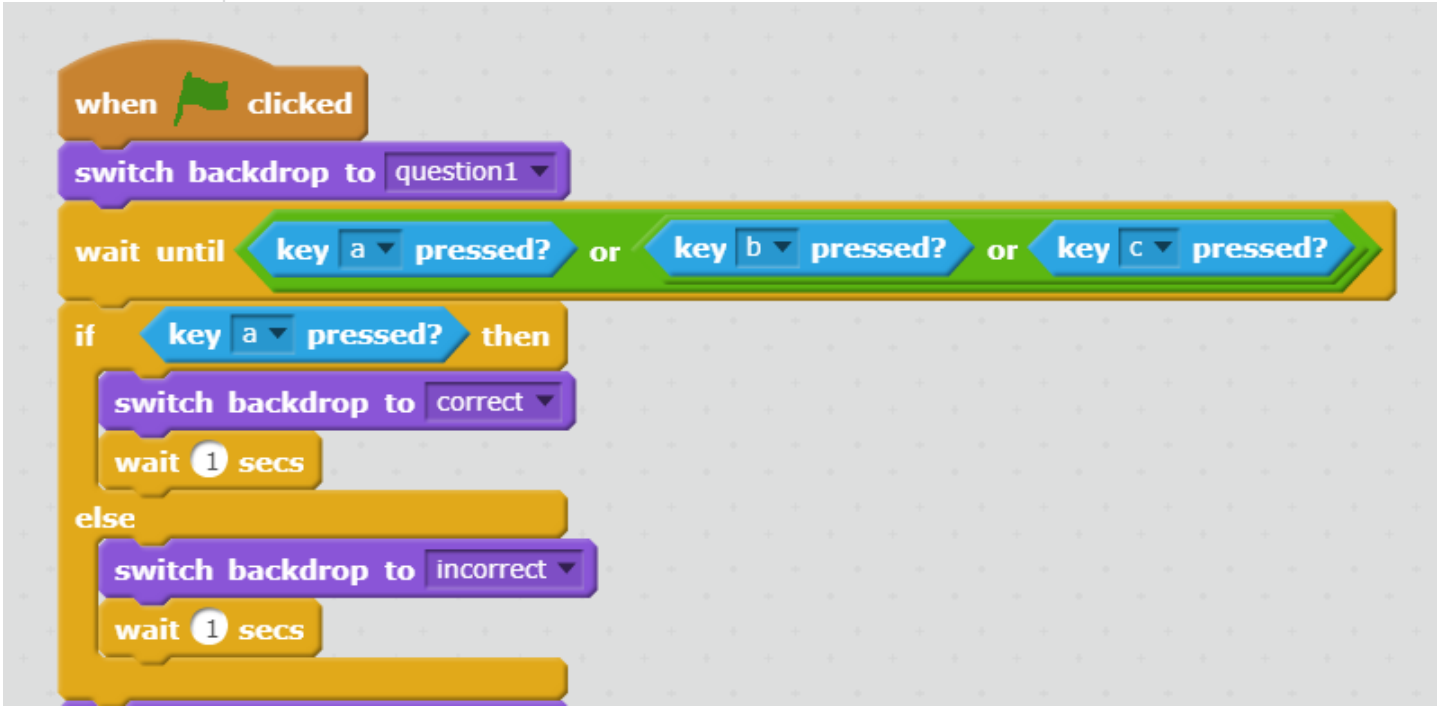
Australian Curriculum Processes and production skills strand	Example: choose your own adventure
Investigating and defining	define the problem (students not being nice to others)
Generating and designing	How will it function? - Create a flow chart to describe the flow of events (sequence of steps) What data is required?
Producing and implementing	implement the solution using a visual programming language
Evaluation	gain feedback in the evaluation phase

Quiz: remixing



User input Keystrokes a, b or c

Branching: If – then – else

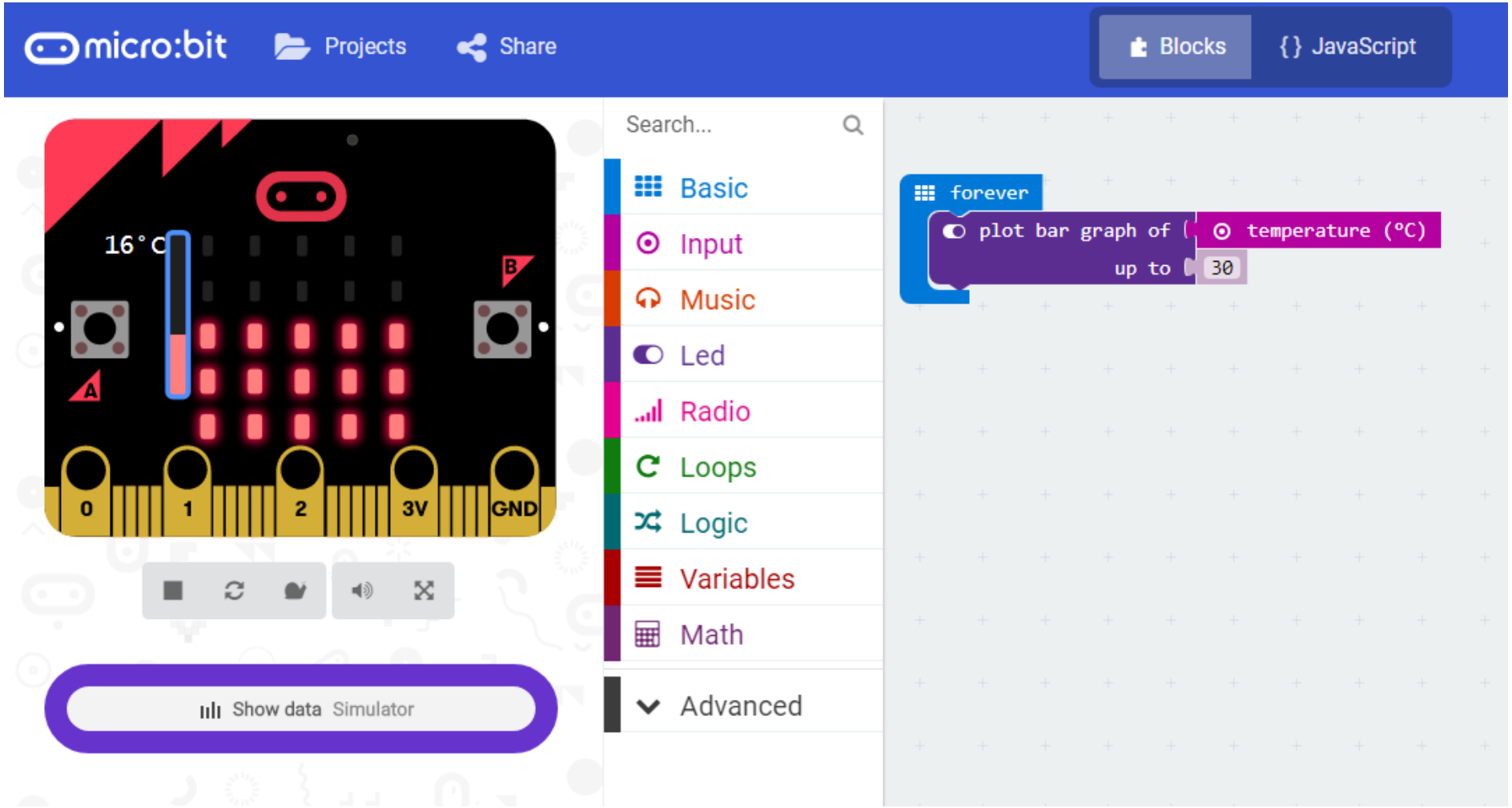


Computational thinking (CT)

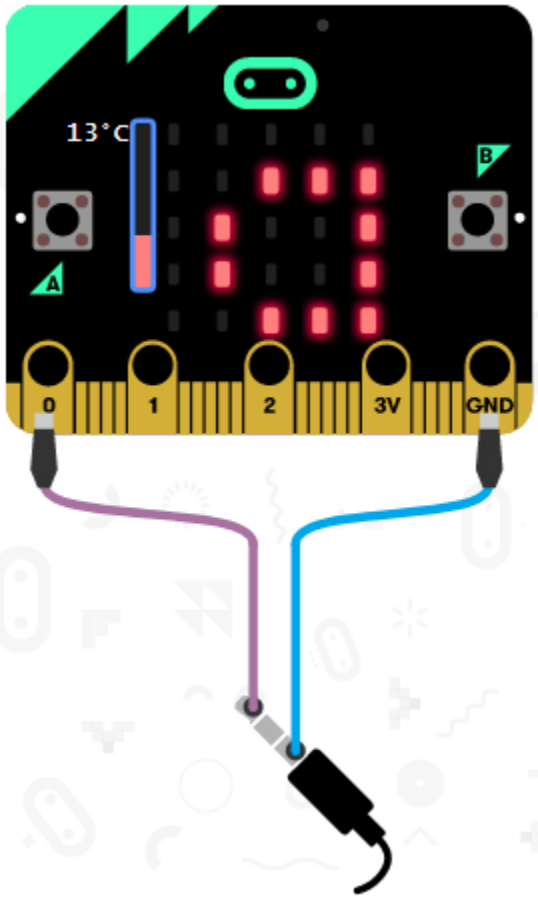
Aspect of CT	Choose your own adventure	Design a quiz (Colonial Australia)
Decomposition	Key stages of the story	How many questions? Question style
Abstraction	Which behaviours can be addressed?	Questions of interest
Algorithm design	User interaction What is the flow, what are the decisions and where do they take the user?	What is user input? What feedback does the user get? What is the flow of questions?
Pattern recognition	Repeat code	Repeat code
Evaluation	Is the user able to choose their own adventure?	Does the correct/incorrect feedback work as expected?

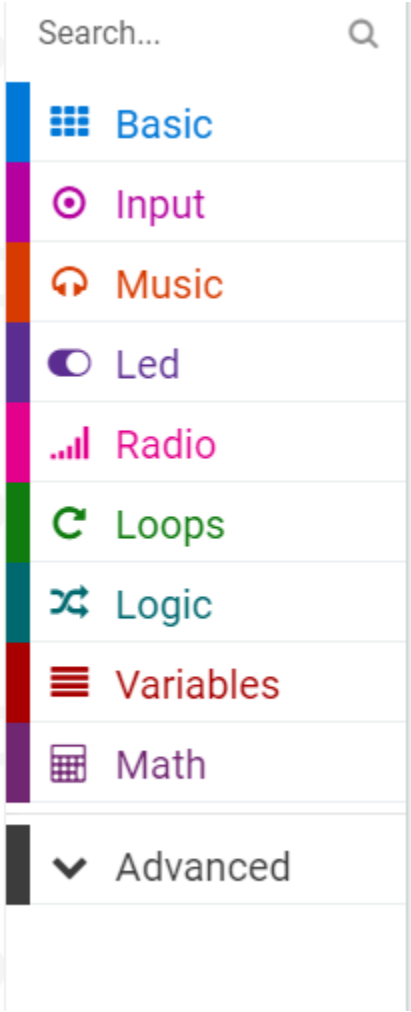
BBC micro:bit

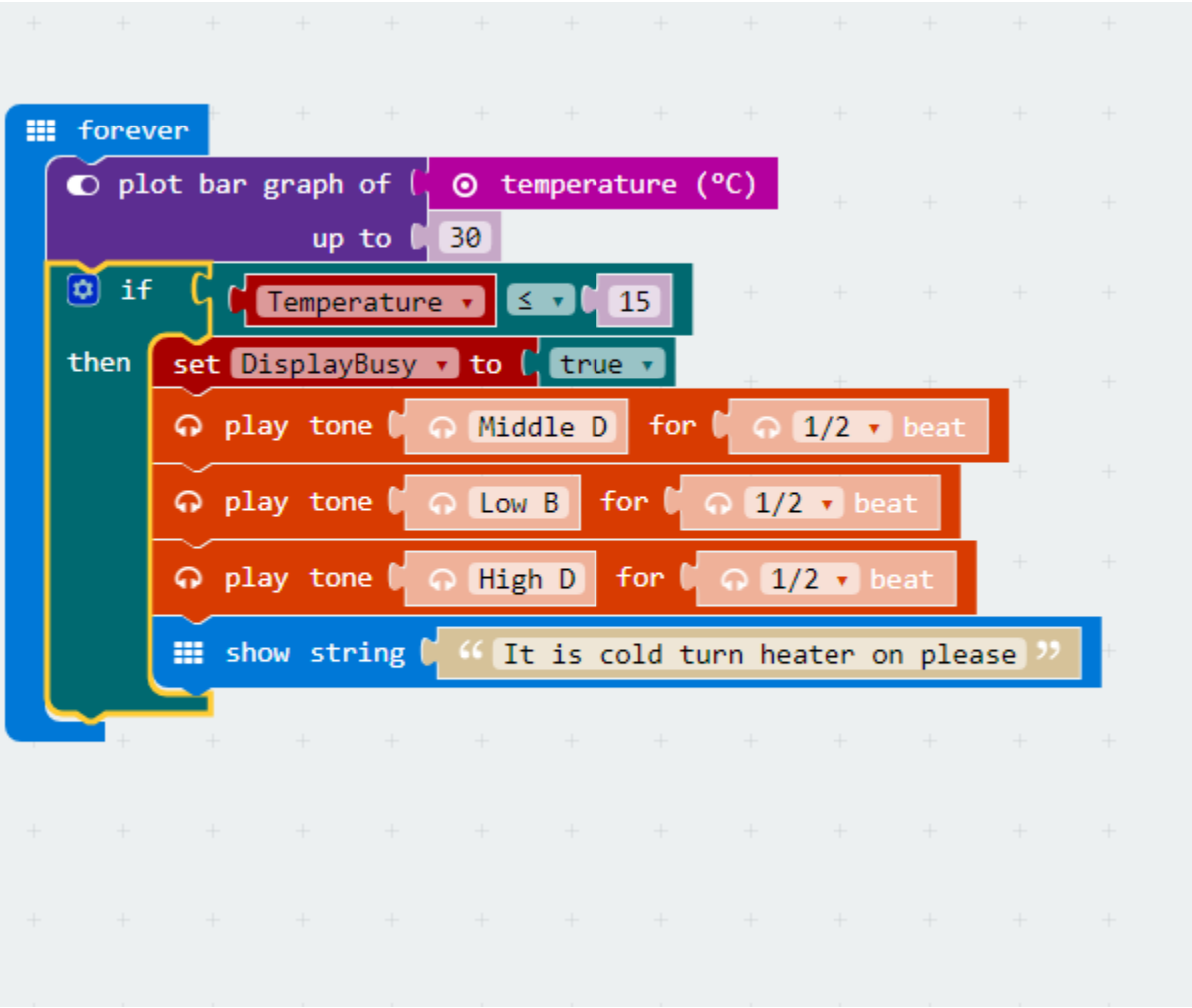
Make your own temperature sensor



BBC micro:bit







Intro to programming 8 hours Year 3
Follow the problem solving process to design and create a digital solution.

Programming project 12 hours Year 4
Develop an understanding of computer programming as a series of instructions

Communicate ideas and information 5-7 hours Year 3
Learn how information systems can be used by students and others in their community.

Apply protocols 7-8 hours Year 4
Develop a school ICT agreement and collaborate with others to complete an online task, using agreed protocols.

SHOW ONLY RELEVANT UNITS

UNIT PROGRAMMING PROJECTS

YEAR LEVEL: 4 TOPIC: DIGITAL SOLUTIONS TIME: 12 HOURS

Students should develop an understanding of computer programming as a series of instructions that can change depending on different user inputs or conditions. The focus is on how digital systems follow instructional pathways and how these can be described using flow charts or through the use of visual programming languages. These pathways can be hand drawn, displayed graphically, using cards or manipulated digitally using block-based programming languages.

Topic Overview | Topic Unit Map

Unit Overview | Unit SOLO Taxonomy

FLOW OF ACTIVITIES

1 **Define a problem**
Define a problem drawing on computational thinking and draw some conclusions about its features or needs.
Learn More

2 **Create a storyboard**
Create a storyboard or flow chart to record relationships between the content and processes.
Learn More

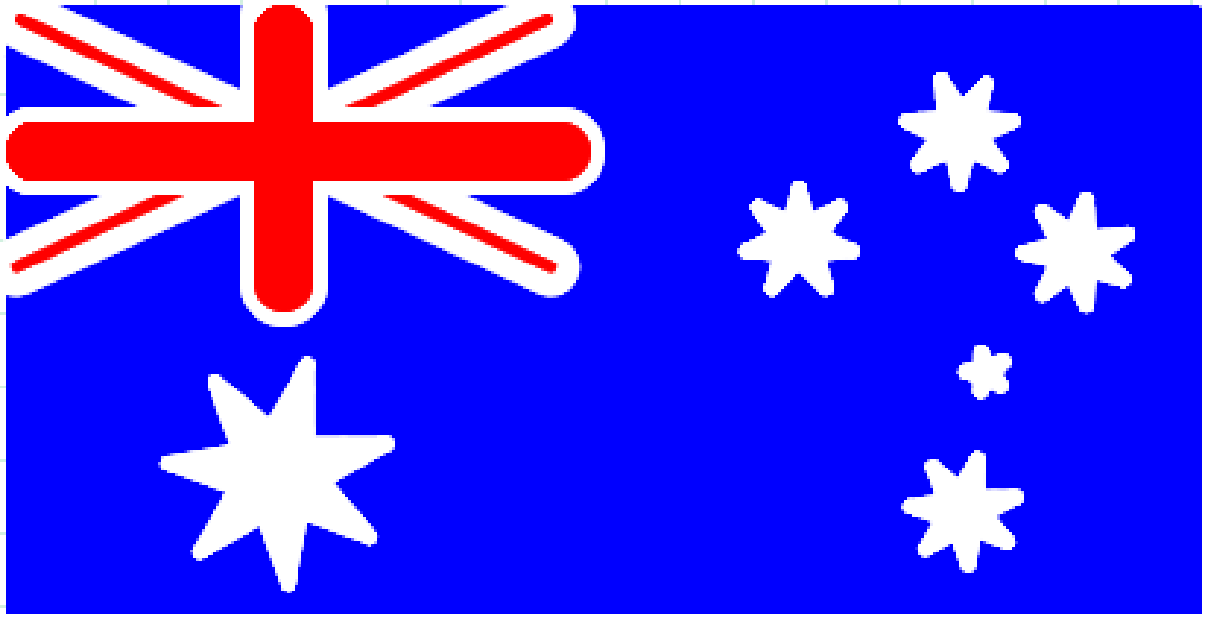
3 **Implement a solution**
Use a visual programming language as part of the digital solution.
Learn More

4 **Evaluate it**
Evaluate how well the solution met the desired outcome.
Learn More

http://bit.ly/DT_Yr3-4

Lexi
Year 5–6

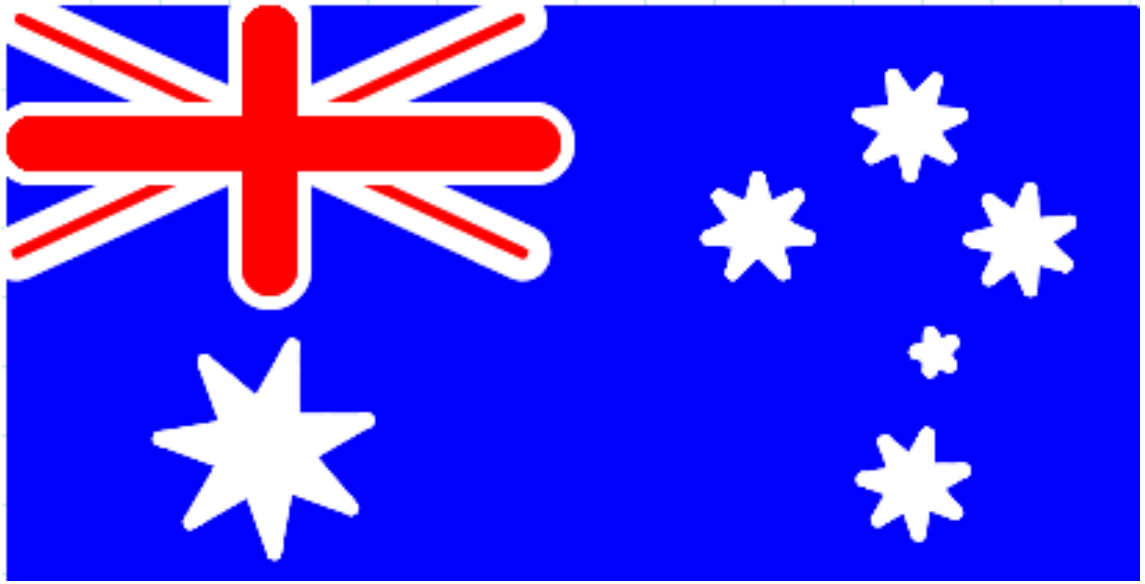
Repeat loops (repetition)
Designing a flag and create a program



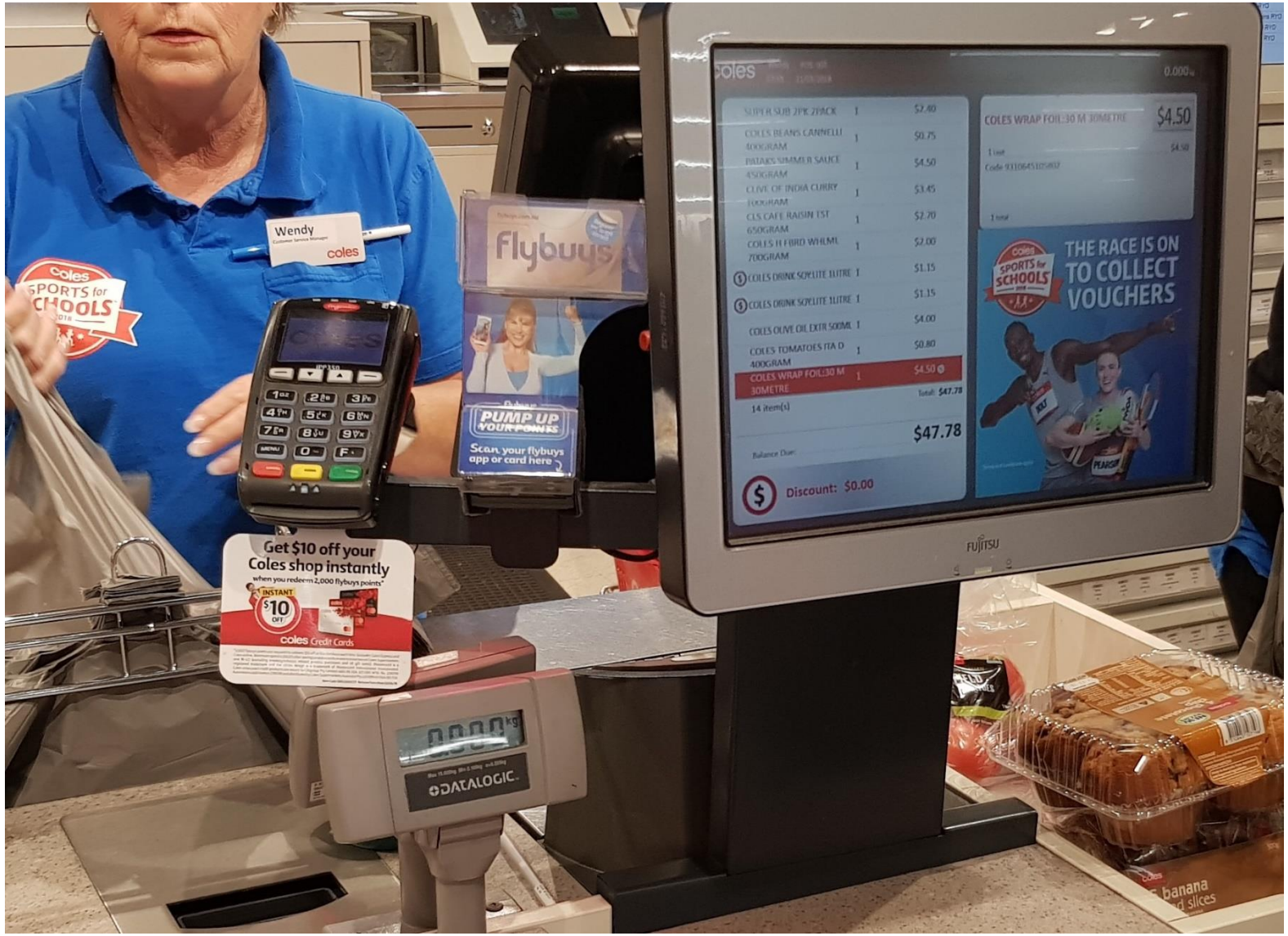
Repeat loops (repetition)

```
pen white, 5
for [1..7]
    fd 25
    rt 150
    fd 25
    lt 100
fill white
pu()
```

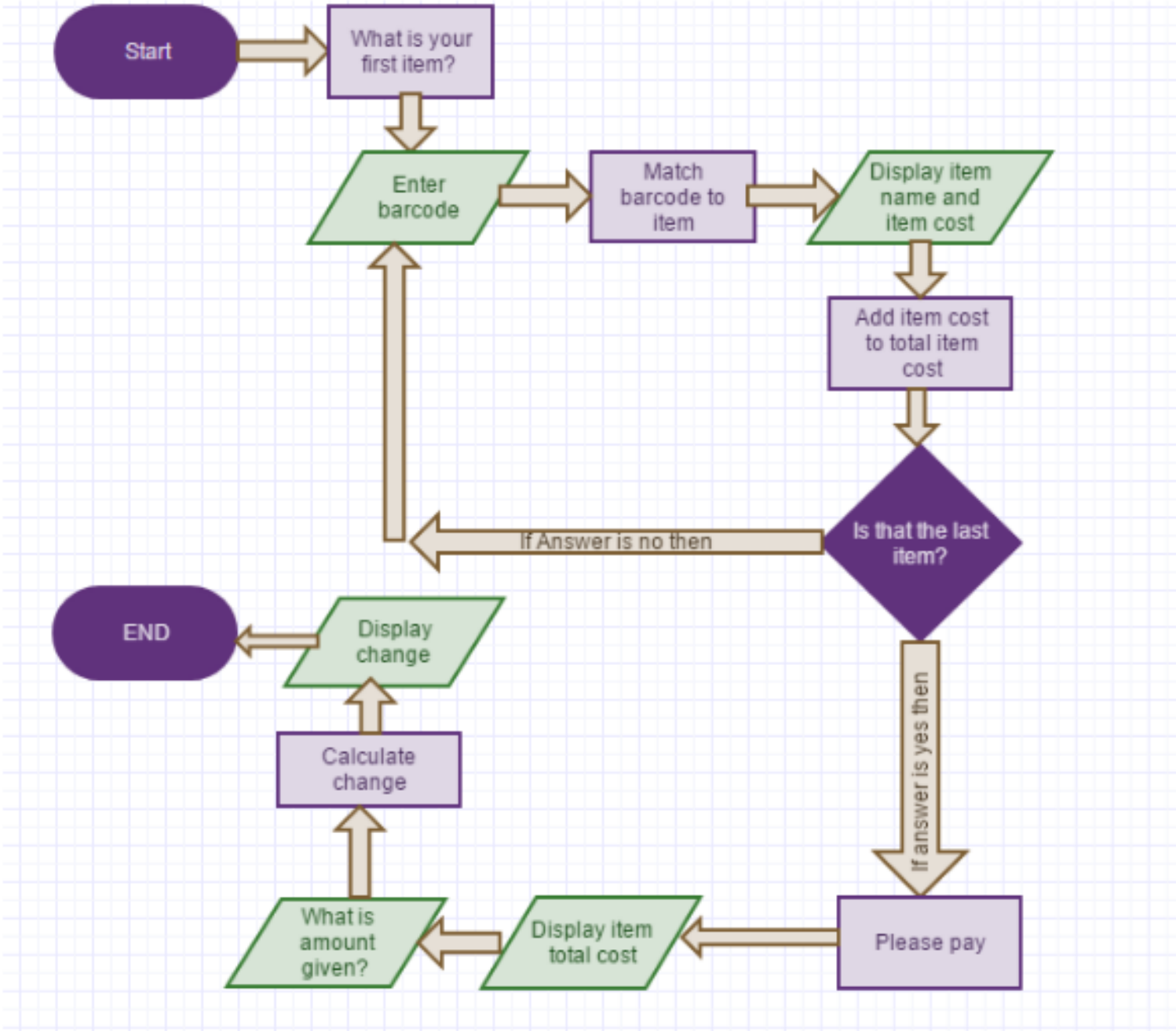
```
pen white, 5
for [1..7]
    fd 12
    rt 150
    fd 12
    lt 100
fill white
pu()
```



Model how a supermarket checkout works



Create an algorithm Flow chart (Gliffy)

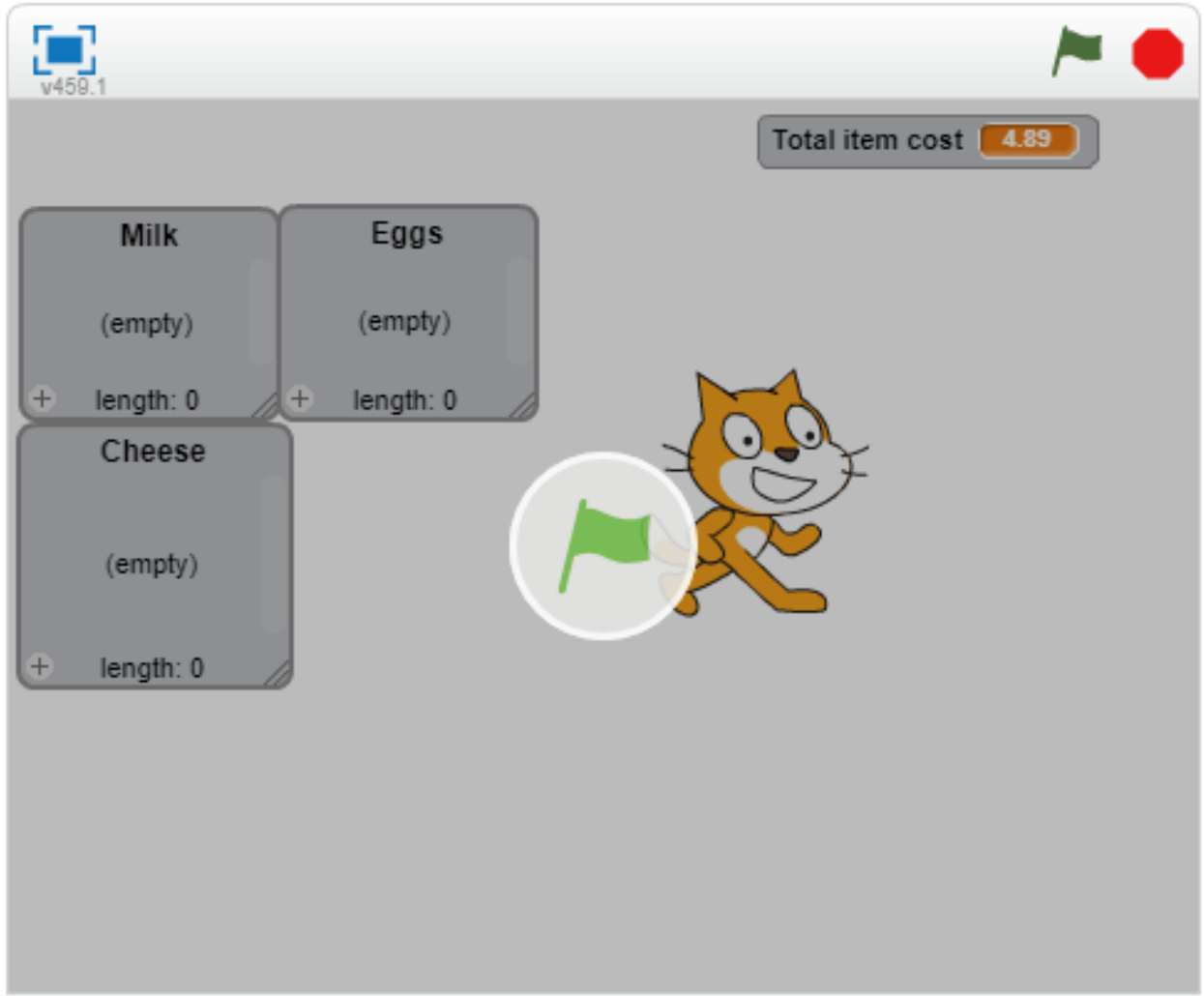


Checkout

by Fargo123

1 scripts
1 sprites

See inside



Instructions

Barcodes of items are

5000181028133
077782008708
9310625012342

Notes and Credits

I've been playing around with linking barcodes or real items to replicate purchasing

shopping grocery checkout

Shared: 23 May 2017

Modified: 14 Aug 2017

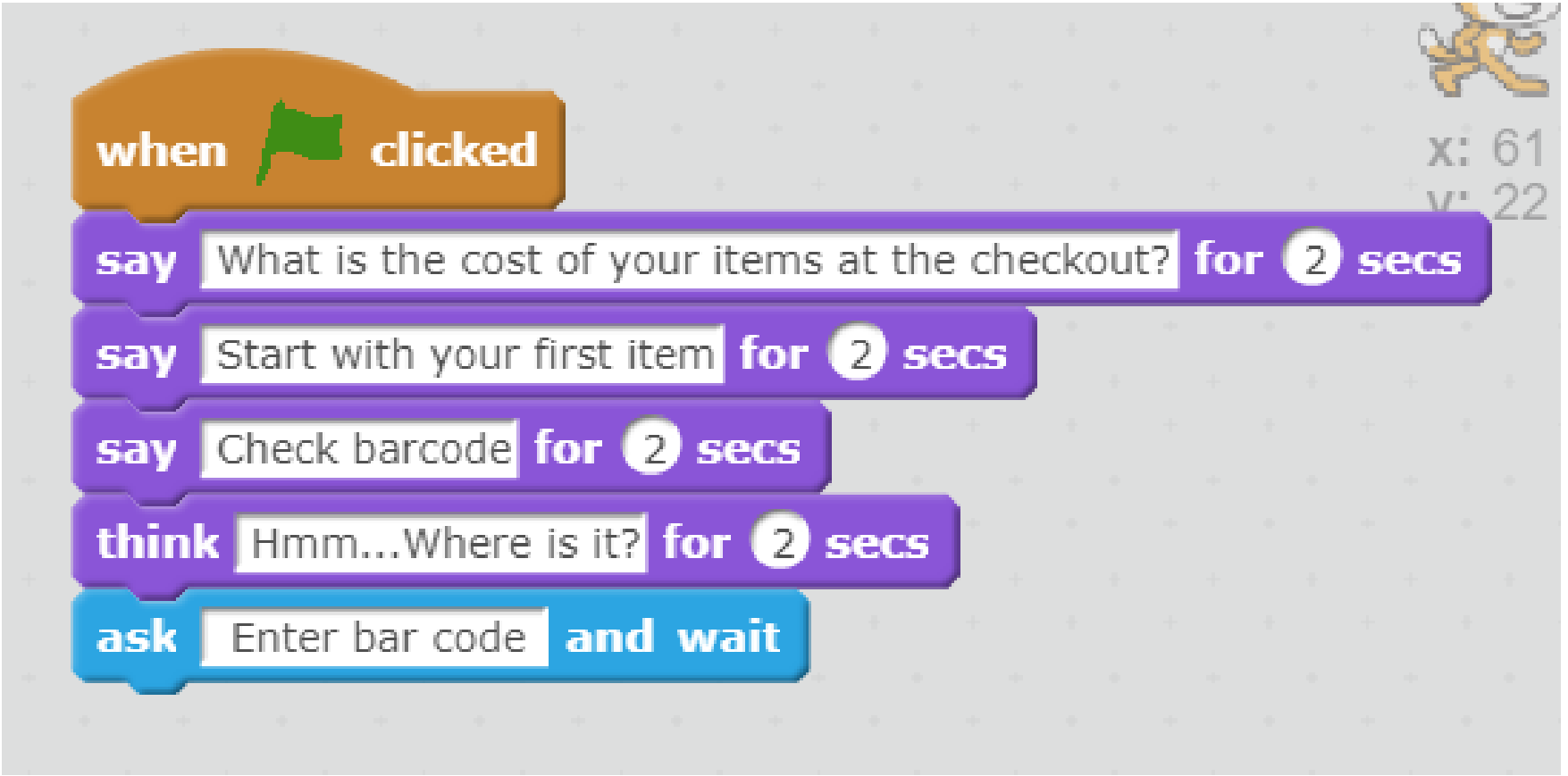
0 0 20 1

Build the program in stages

User input:

Use purple ‘Say’ blocks to print text onscreen

Use blue ‘Ask’ block for user input



Build the program in stages

Branching:

Test ‘If –then’ block

If correct bar code is entered

A series of actions occur onscreen

Build the program in stages

Branching:

Now it works for one item add some more

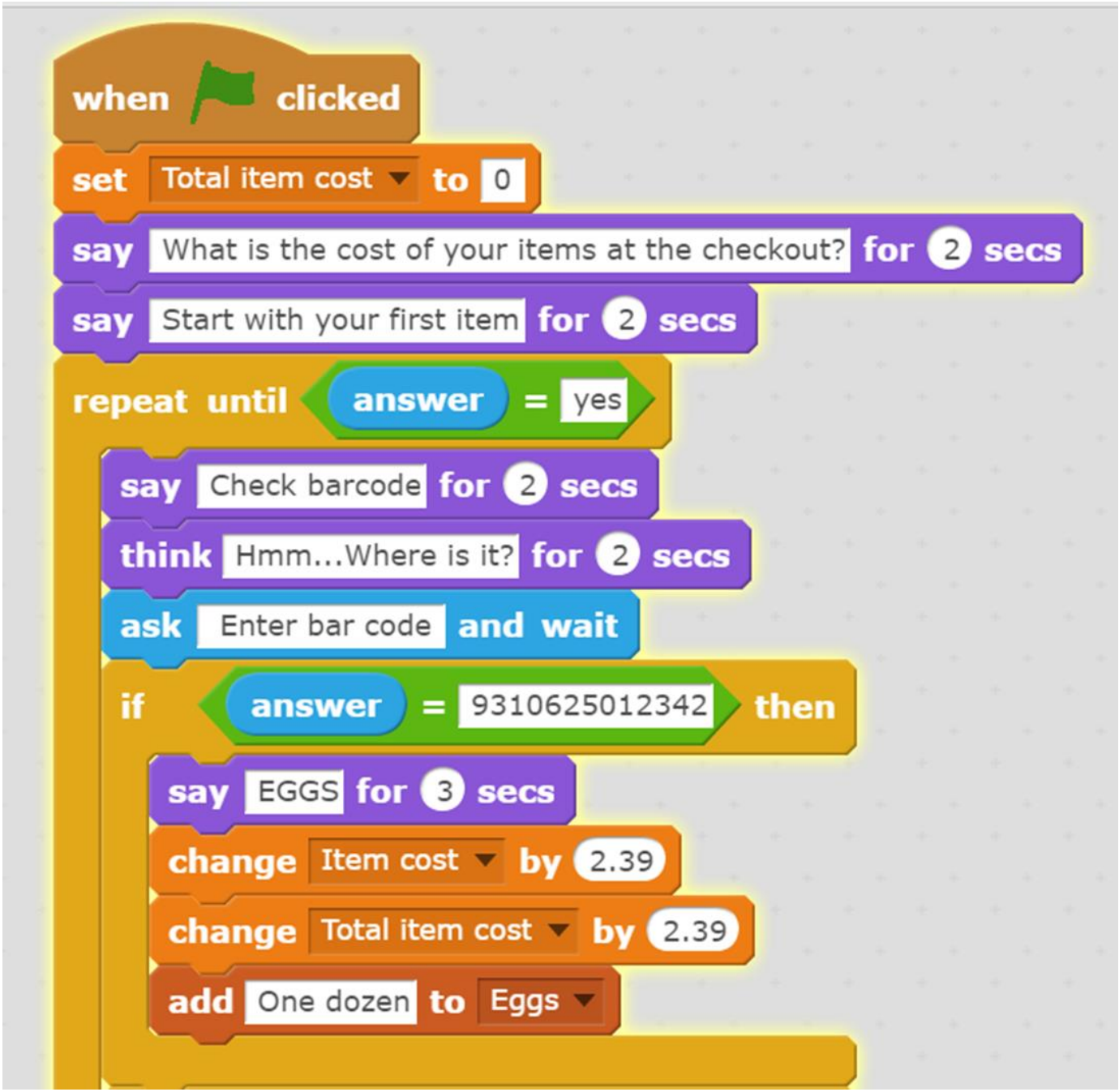
Reuse the ‘If –then’ block sequence for other items

Build the program in stages

Iteration:
Add in **repeat until** a certain condition is met.

In this case if answer is yes to ‘Is this your last item?’

A series of actions occur onscreen

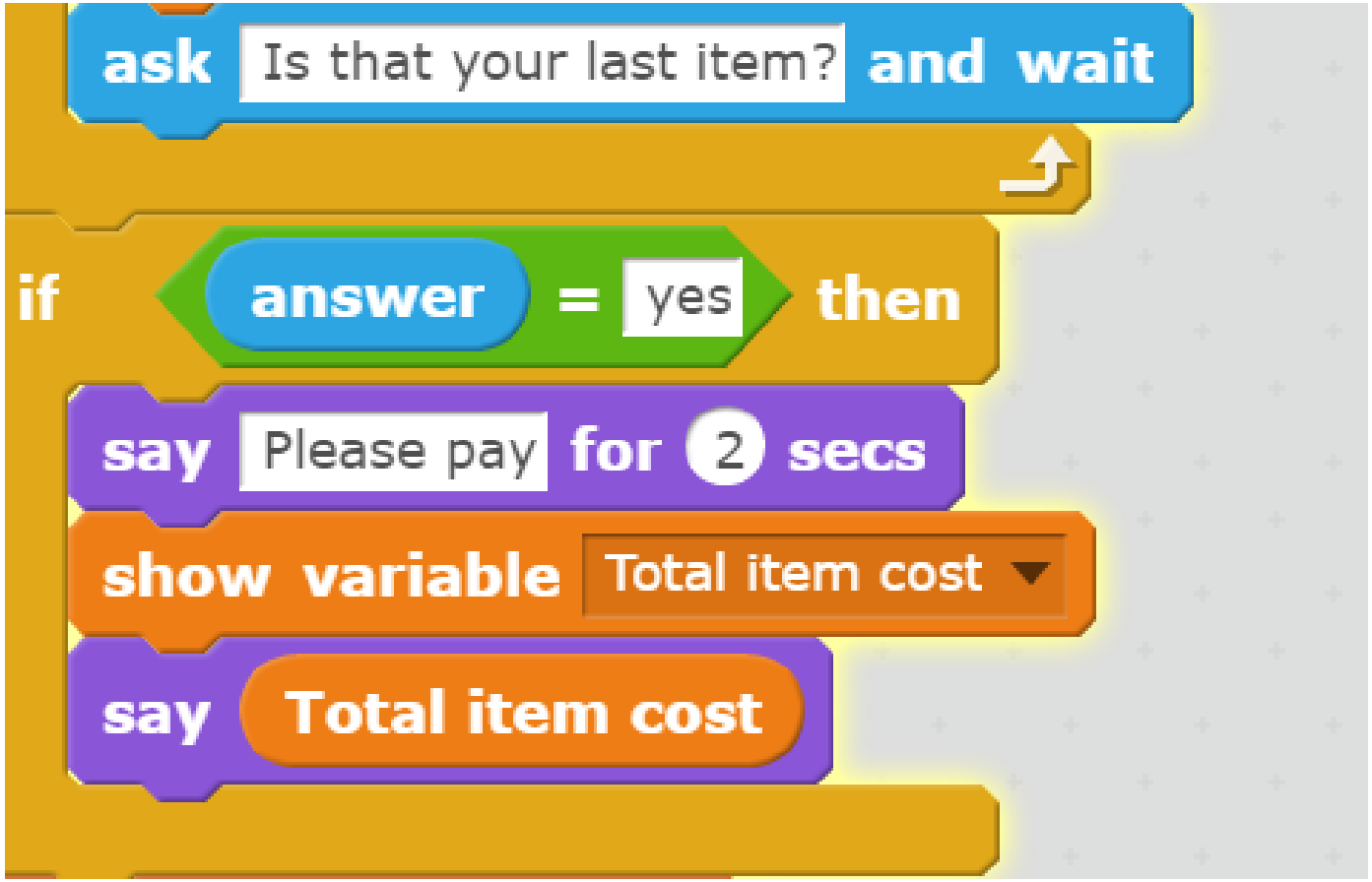


Build the program in stages

Iteration:

Add in repeat until a certain condition
In this case if answer is yes to:
Q: Is this your last item?

A series of actions occur onscreen



STEM project

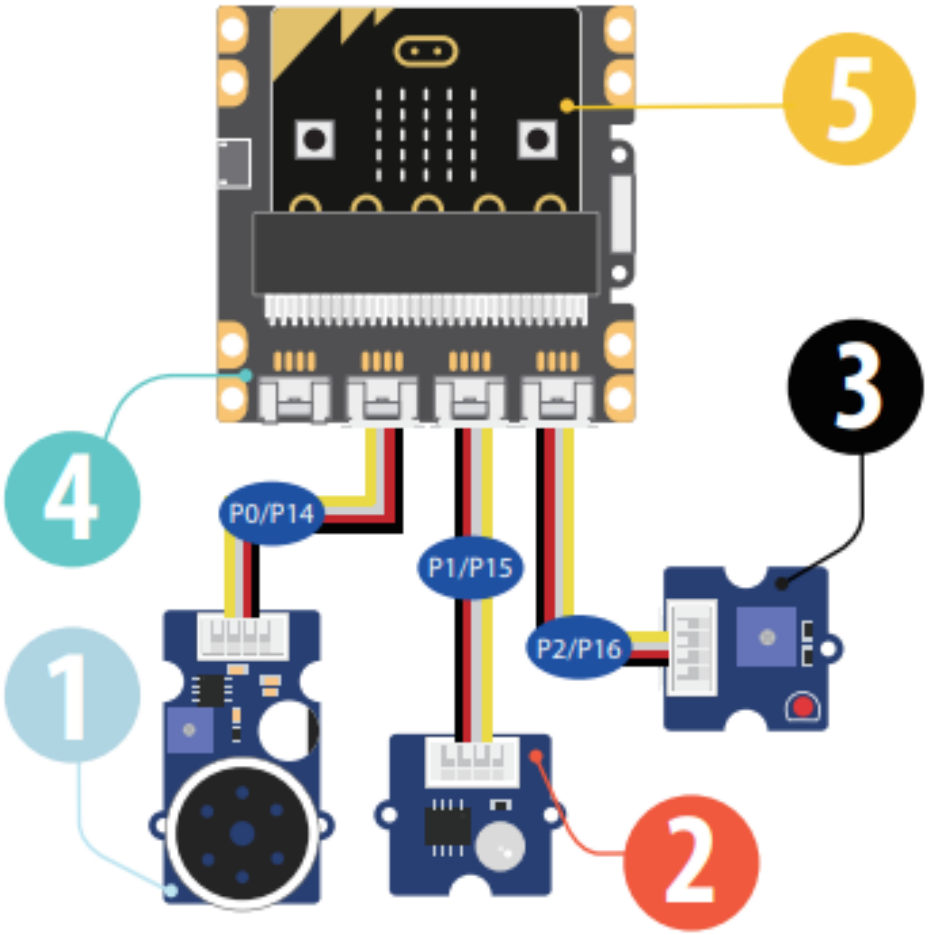
Grove Kit and BBC micro:bit

This set up shows how the set up Lexi used to connect:

- (1) speaker
- (2) light sensor
- (3) Red LED

This project can be a solution to guarding a prized procession.

Its an alarm system



STEM project

Lego Mindstorms robotics

Design a robot to complete a task

UNIT PROBLEM-SOLVING PROCESSES

YEAR LEVEL: 5 TOPIC: CREATING DIGITAL SOLUTIONS TIME: 16 HOURS

When students are set the task of solving a problem that requires a digital solution, they usually start by investigating and defining the problem. They draw on computational thinking, a problem-solving approach that involves activities such as organising data logically, breaking down problems into components, and designing and using algorithms and models to show how the solution will be developed and how it will appear. As part of designing their solution, students generate ideas and consider the user of their digital system. During the producing and implementing process students typically create their own solution using a visual programming language. Once a digital solution has been created it is important to evaluate it against relevant criteria, such as: Did it entertain the users (if a game)? Can updated data be added so the solution can be used in the future? (Future needs). Note: Sometimes when students are creating digital solutions they might return to a process they have already completed in order to make adjustments; however, typically at this level, students engage in each of these processes in the above-mentioned order.

Programming is the way we communicate algorithms to a digital system, such as a laptop or notebook, so that the system understands the instructions. Digital systems need precise instructions as they are unable to understand instructions that include superfluous details. We use programming languages to code the instructions. There are many different visual programming languages but all have common programming statements and use a common approach to creating a program and running it to see if it works as intended.

Topic Overview | Topic Unit Map | Unit Overview | Unit SOLO Taxonomy

FLOW OF ACTIVITIES

1. **Define the task**
   Analyse the problem and identify the functional requirements of the solution.
   Learn More
2. **Design a digital solution**
   Represent how the solution will be created and what the solution will look like
   Learn More
3. **Visual programming solution**
   Use a visual programming language to implement a digital solution.
   Learn More
4. **A digital solution**
   Design a digital solution in which data is input and output by digital systems.
   Learn More

http://bit.ly/DT_Yr5-6

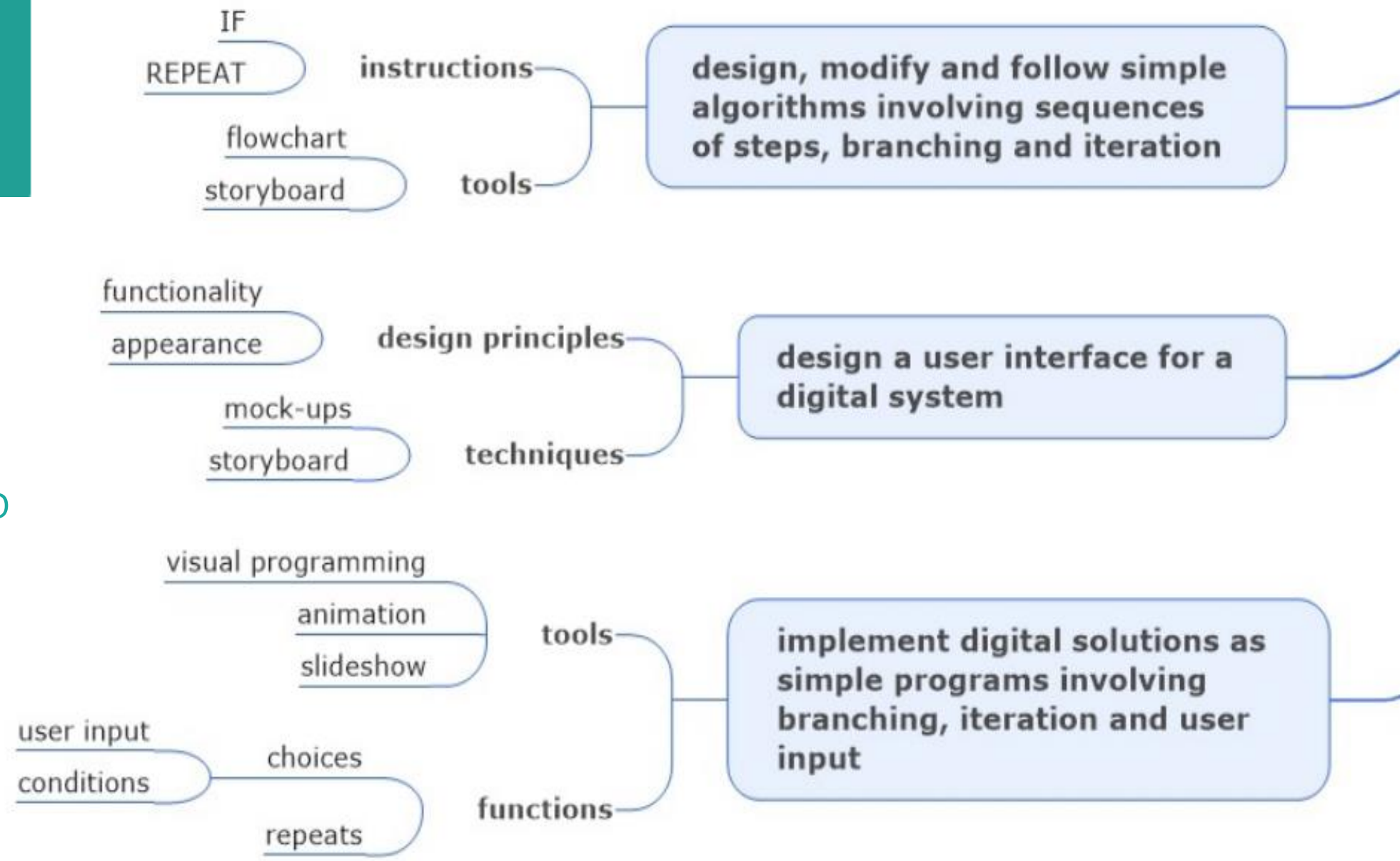
SOLO Taxonomy

SOLO stands for the Structure of the Observed Learning Outcome.

We are creating an online game				
SOLO LEVEL	**One**	**Many**	**Relate**	**Extend**
SOLO VERB	***Identify isolated skills***	***Describe and combine serial skills***	***Integrate skills***	***Evaluate skills***
DECLARATIVE KNOWLEDGE Knowing about (talking or writing about) the programming code Creating a digital solution using visual programming language Success criteria	I can **DEFINE** a problem identifying functional and data requirements I can **IDENTIFY** **…** the use of isolated visual programming skills when programming For example, the use of: ☐ an if/then statement ☐ loops or repetition ☐ user input	I can **DESCRIBE** … the use of isolated and combined visual programming skills when programming For example, the use of loops when: ☐ incorporating repeat instructions ☐ allowing for varied user input ☐ selecting options (for example, in a quiz)	AND I can **EXPLAIN** my programming choices – when programming a digital solution such as an animation, quiz, choose your own adventure story or controlling a robotic device	AND I can **EVALUATE** the effectiveness of my digital solution in meeting its functional requirements for: ☐ meeting its intended purpose ☐ user input.
FUNCTIONING KNOWLEDGE Knowing how to … Creating a digital solution using visual programming language Success criteria	I can interpret an algorithm presented as a flow chart I can use a visual programming language **IF** I copy programming examples created by someone else	I can create an algorithm that I use to plan out a program for a digital solution. I can create a paper prototype of my design to show screen transitions I can independently program a digital solution using a visual programming language	I can independently and confidently create a digital solution using a visual programming language AND I can debug as I build (correct my own code)	AND I can seek and act on feedback to improve the effectiveness of my programming choices as I go.

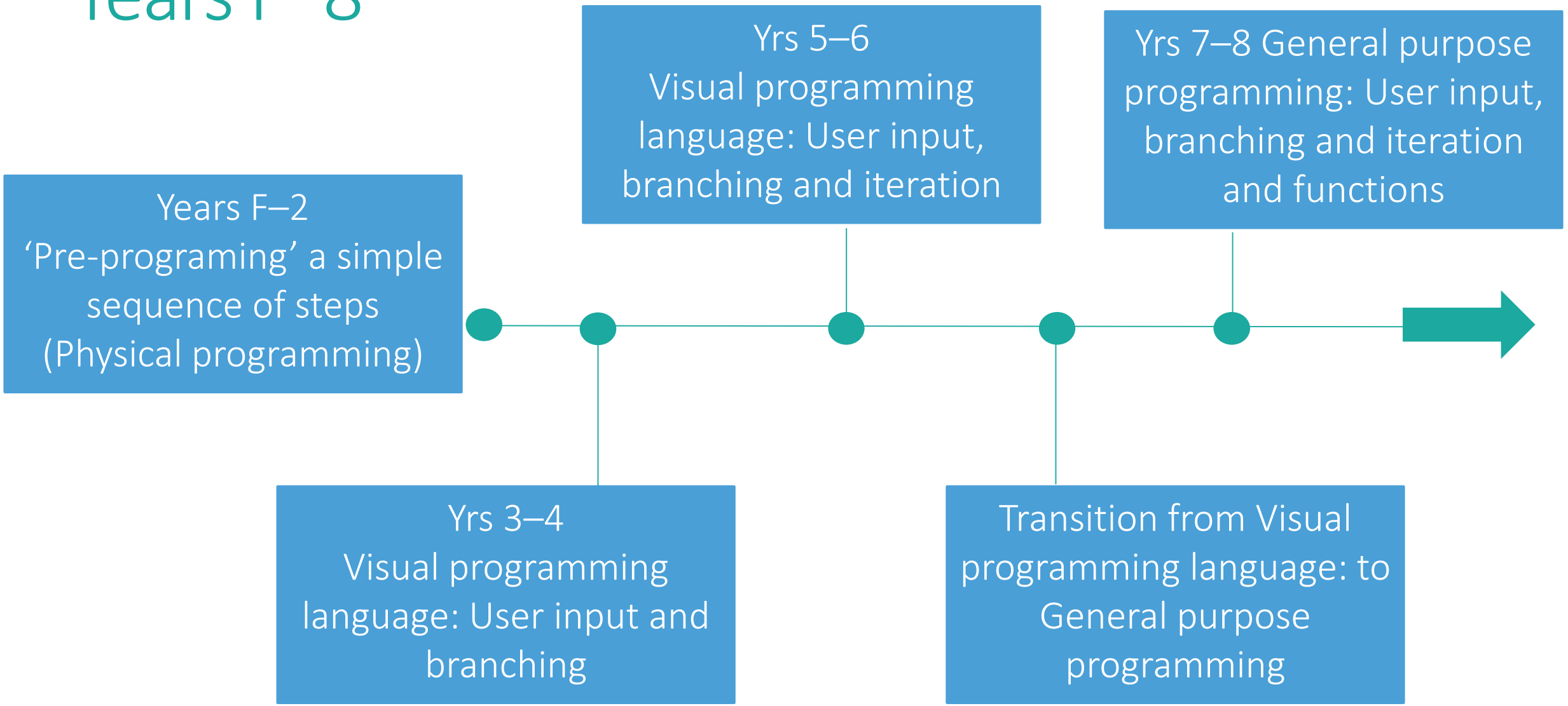
Topic Unit Map

A visual representation of the underpinning concepts related to the content descriptions



Created by Paula Christophersen

Programming (Producing and implementing) Years F–8



Looking into the Scope and sequence

Spend time as a group looking at the Scope and sequence.

Use the sheet provided to guide your exploration.

Lesson ideas

Lessons are currently being updated to reflect inclusive teaching practices

DT+ HASS Geography

Create a computer program to learn a traditional Aboriginal or Torres Strait Islander language.

Curriculum links | Assessment

Learning Sequence

- Suggested steps
- Discussion
- Why is this relevant

Suggested steps

1. Discuss and list ways you might learn a new language.
2. View the video Learn some Warrgamay words.
3. Discuss this approach to learning some words and phrases from a traditional Aboriginal language. From viewing the video, what words do the students now know? What does the video tell us about the animals and lands of the Warrgamay people?
4. Compare the video with this quiz, created in Scratch: Warrgamay animals.
5. Discuss how the quiz works and what programming blocks your students expect would have been used.

Lesson ideas

Lessons are currently being updated to reflect inclusive teaching practices

A level approach has been taken to differentiate the lesson

8. **Instructions (with Differentiation)**

The Digital Technologies curriculum differs from the old ICT curriculum in that there is an emphasis on students' thinking processes. Therefore this task has been divided into three levels where, first, students need to demonstrate they understand the logic and decision-making used to make a quiz game. Then they modify the example quiz to demonstrate understanding of Scratch. The actual creation of a new quiz is saved for last in Level Three. This allows you to differentate the task depending on students' understanding of both computational thinking and Scratch (or similar programs). Proficient students could start on Level Two or even Level Three where as struggling students practice pieces of the task at Level One.

Level One

Level Two

Level Three

Connect with the Digital Technologies Hub


DT Hub Newsletter


@DigiTechHub


DigitalTechnologiesHub


Digital Technologies Hub