# Programming in Stage 2

Leanne Robertson, Program Director DT Hub
Martin Richards, Content Manager, DT Hub

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

DIGITAL TECHNOLOGIES HUB

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

http://bit.ly/ICTvsDT

## Information Communication Technology (ICT) Capability

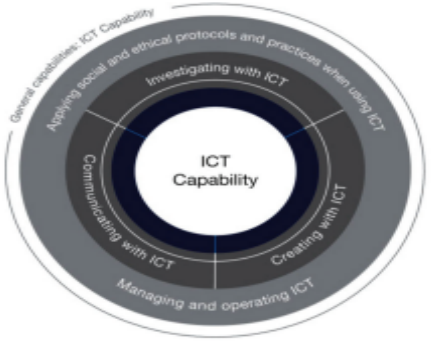
A general capability taught within all curriculum areas for students in years F–10.

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

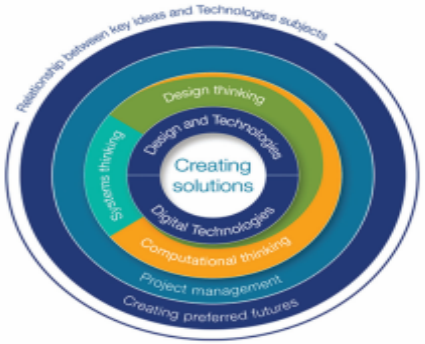
A new subject for F–10 (optional in 9-10) students with new and unique skills and content.

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.

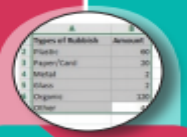
**Digital Technologies build on and extend ICT, moving students from technology consumers to creators.**






## Use ICT

- Presentation tools
- Locate information
- Digital publishing
- Interpret timelines
- Ownership and use
- Managing files
- Mapping and geospatial tools
- Online communication
- Digital music / multimedia



- Spreadsheets and graphing

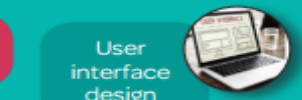
- Analyse and visualise data


- Cyber safety

## 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

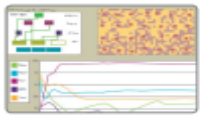
- Data collection

## 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.

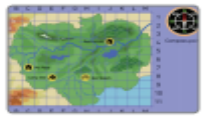


Use a computer simulation or game to test predictions and collect data.

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.

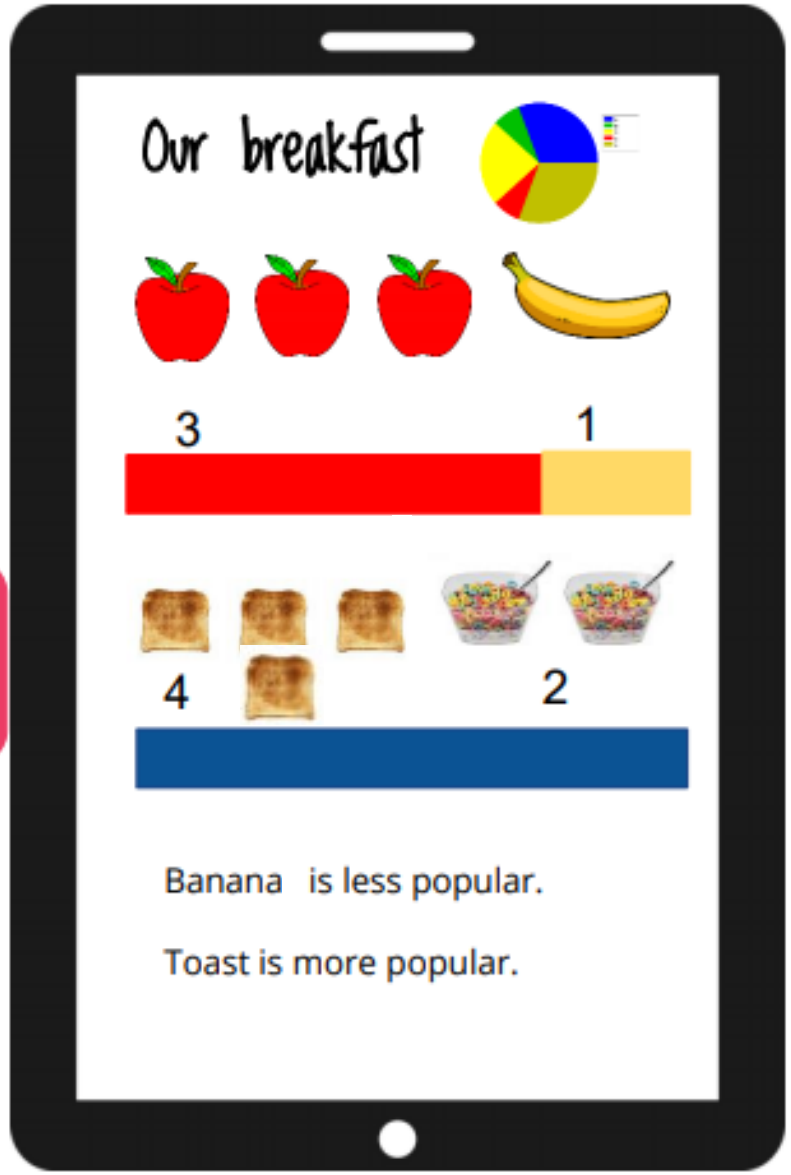
**Understand ICT systems:** identify common consumer ICT systems with input and output functions.

**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)**

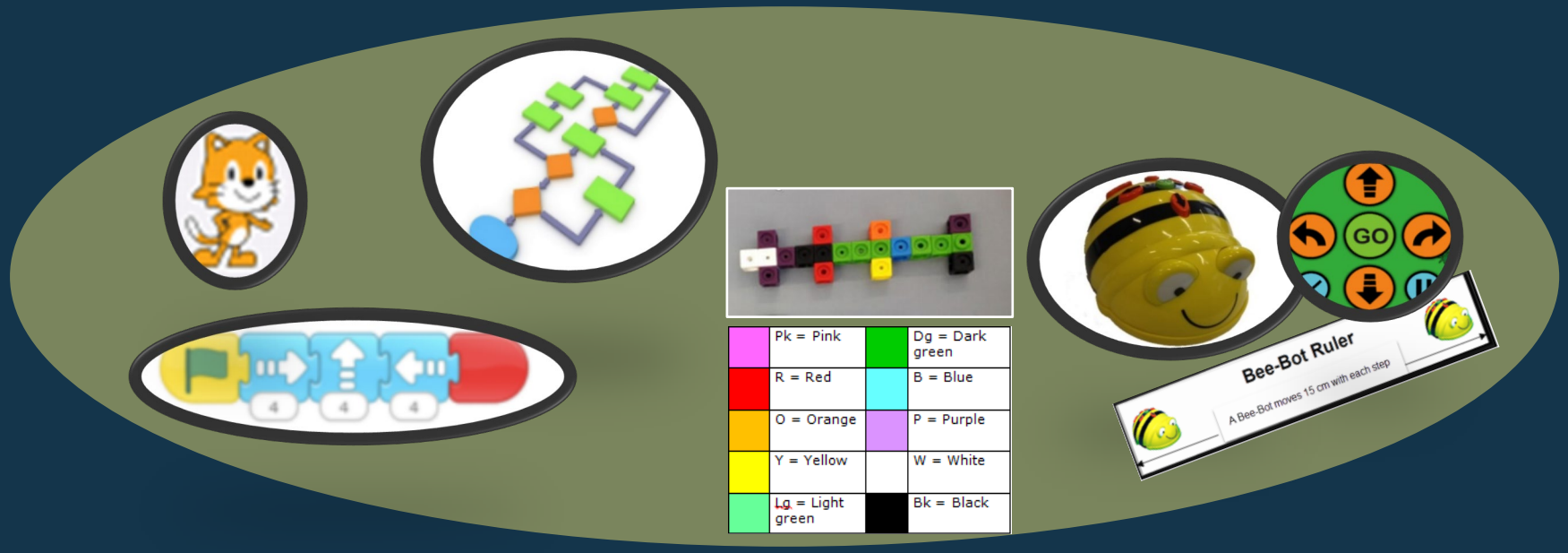
**Collect, explore and sort data, and use digital systems to present the data creatively (ACTDIP003)**

**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




# Algorithms and programming

A sequence of steps that includes user input and branching

# Digital Technologies

## **Focus question:** How are algorithms used to develop digital systems?

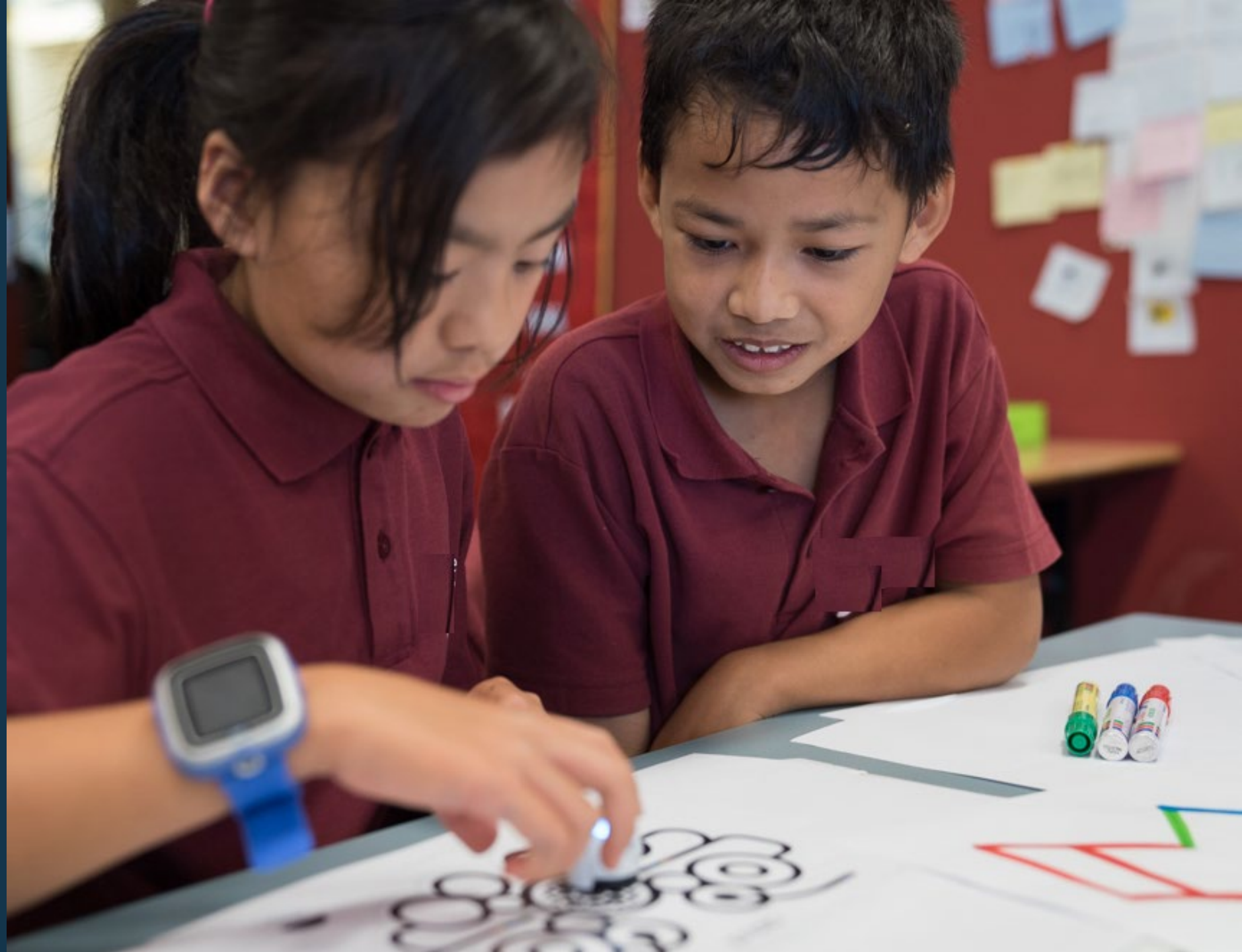
Stage 2
selects and uses materials, tools and equipment to develop solutions for a need or opportunity ST2-2DP-T

Stage 2
defines problems, describes and follows algorithms to develop solutions ST2-3DP-T

Stage 2
describes how digital systems represent and transmit data ST2-11DI-T

describe and follow a sequence of steps and decisions (algorithms) to solve defined problems involving **branching** and **user input**

design and produce digital solutions using a visual programming language

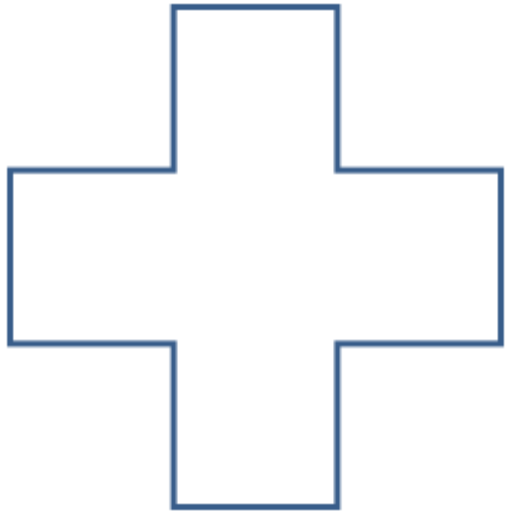


# Probot: Integrate with Mathematics

## Angles and 2D shapes

**To get started you need to know**

| | |
|---|---|
| **moves Pro-Bot fwd 25 cm** | **moves Pro-Bot bwd 25 cm** |
| **rotates Pro-Bot 90° right** | **rotates Pro-Bot 90° left** |

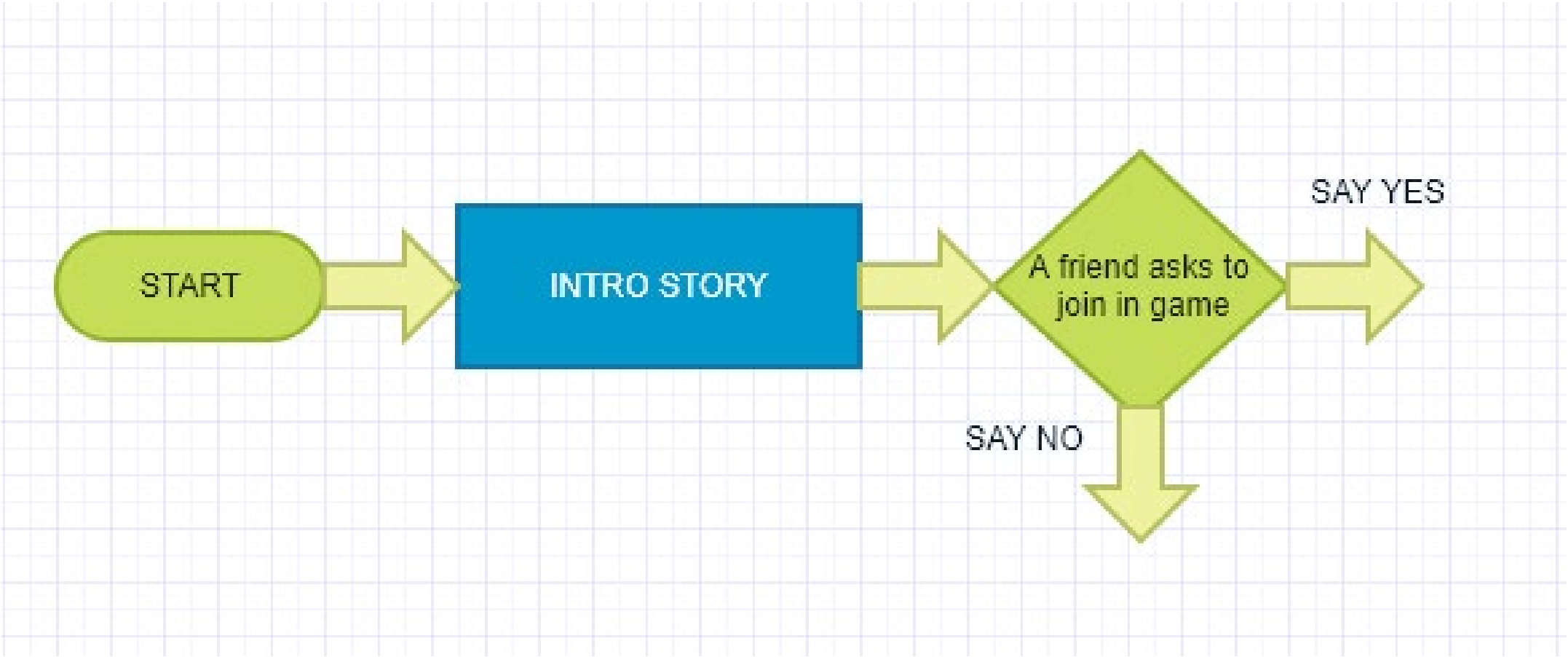


# Probot: Integrate with Mathematics

## Angles and 2D shapes

| Aspect of Computational Thinking | 2D shapes |
|---|---|
| Decomposition | Define the parts of the shape, including angles |
| Abstraction | Focus on the shape's properties |
| Algorithm design | Describe the commands needed to draw the shape (using probot buttons) |
| Pattern recognition | Repeat code<br>Simplify steps |
| Evaluation | Did the commands result in the desired 2D shape? |

# Branching: (decision making)
## Choose your own adventure story



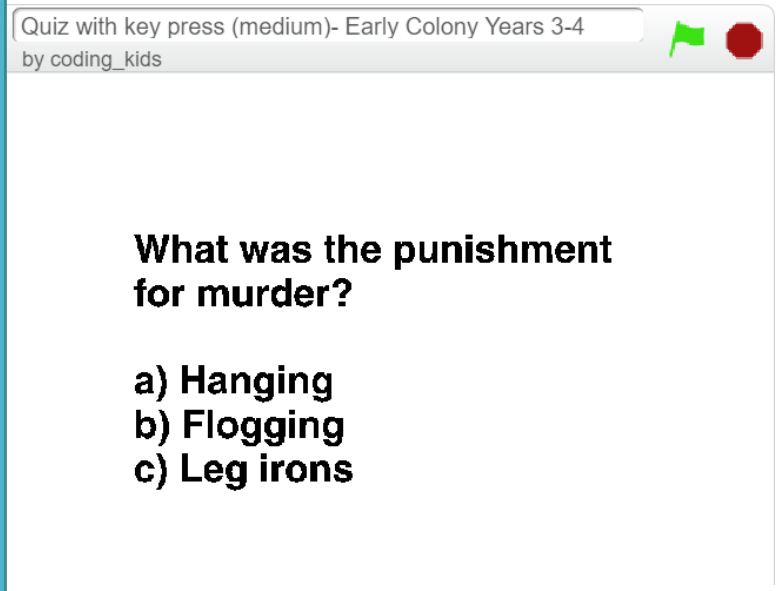
# Design and production process

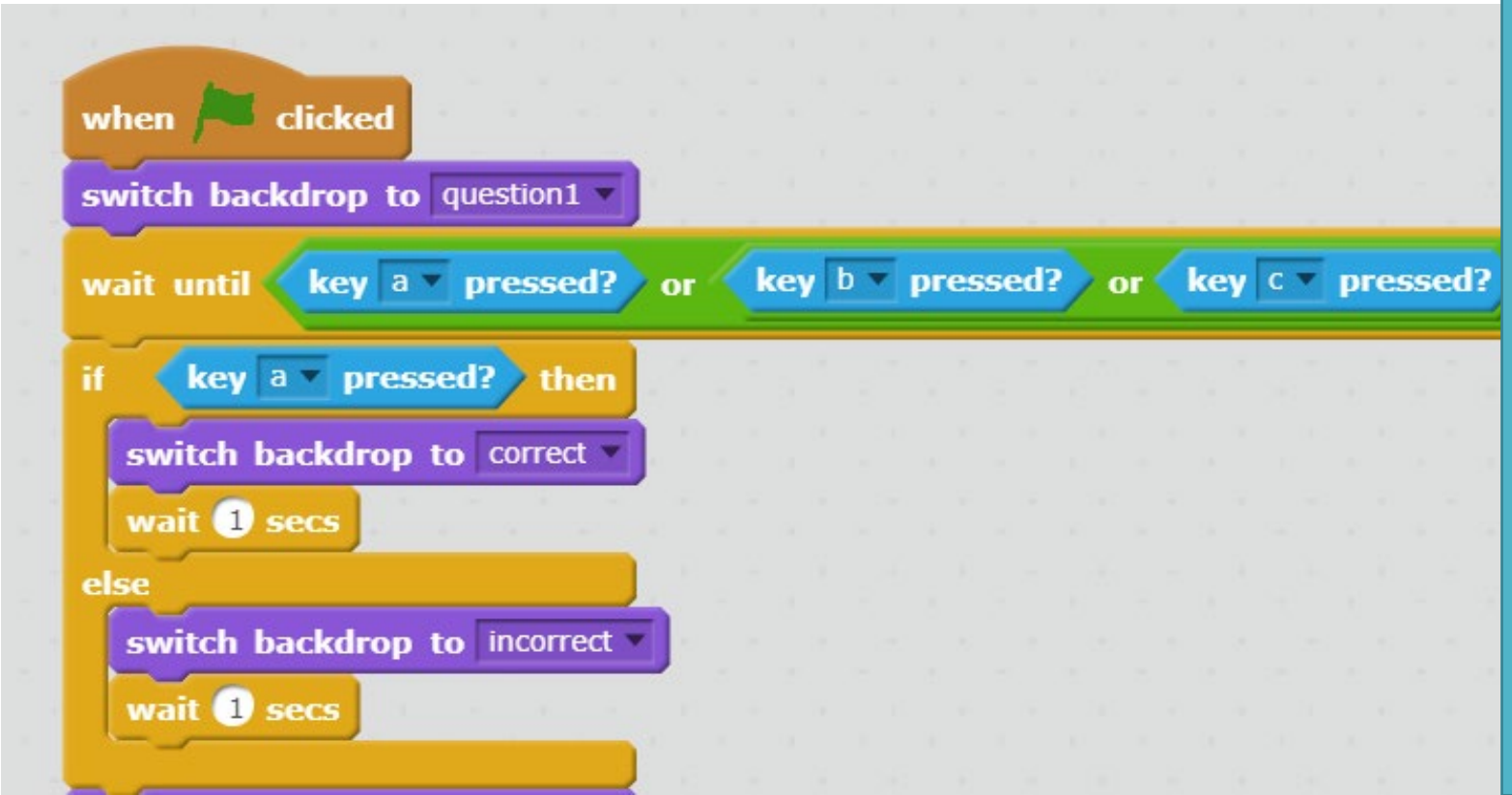
| DIGITAL TECHNOLOGIES Stage 2 | Example: choose your own adventure |
|---|---|
| Investigating and defining | Define the problem (students not being nice to others)<br>How will it function? - Create a flow chart to describe the flow of events (sequence of steps)<br>What data is required? |
| Producing and implementing | Implement the solution using a visual programming language |
| Testing and evaluating | Develop criteria<br>Test and debug as you build<br>Gain feedback in the evaluation phase |

# Quiz: remixing

**User input** Keystrokes a, b or c

**Branching**: If – then – else



# Computational thinking (ComT)

| Aspect of CT | Choose your own adventure | Design a quiz (Colonial Australia) |
|---|---|---|
| Decomposition | Key stages of the story | How many questions?<br>Question style |
| Abstraction | Which behaviours can be addressed? | Questions of interest |
| Algorithm design | User interaction<br>What is the flow, what are the decisions and where do they take the user? | What is user input?<br>What feedback does the user get?<br>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? |

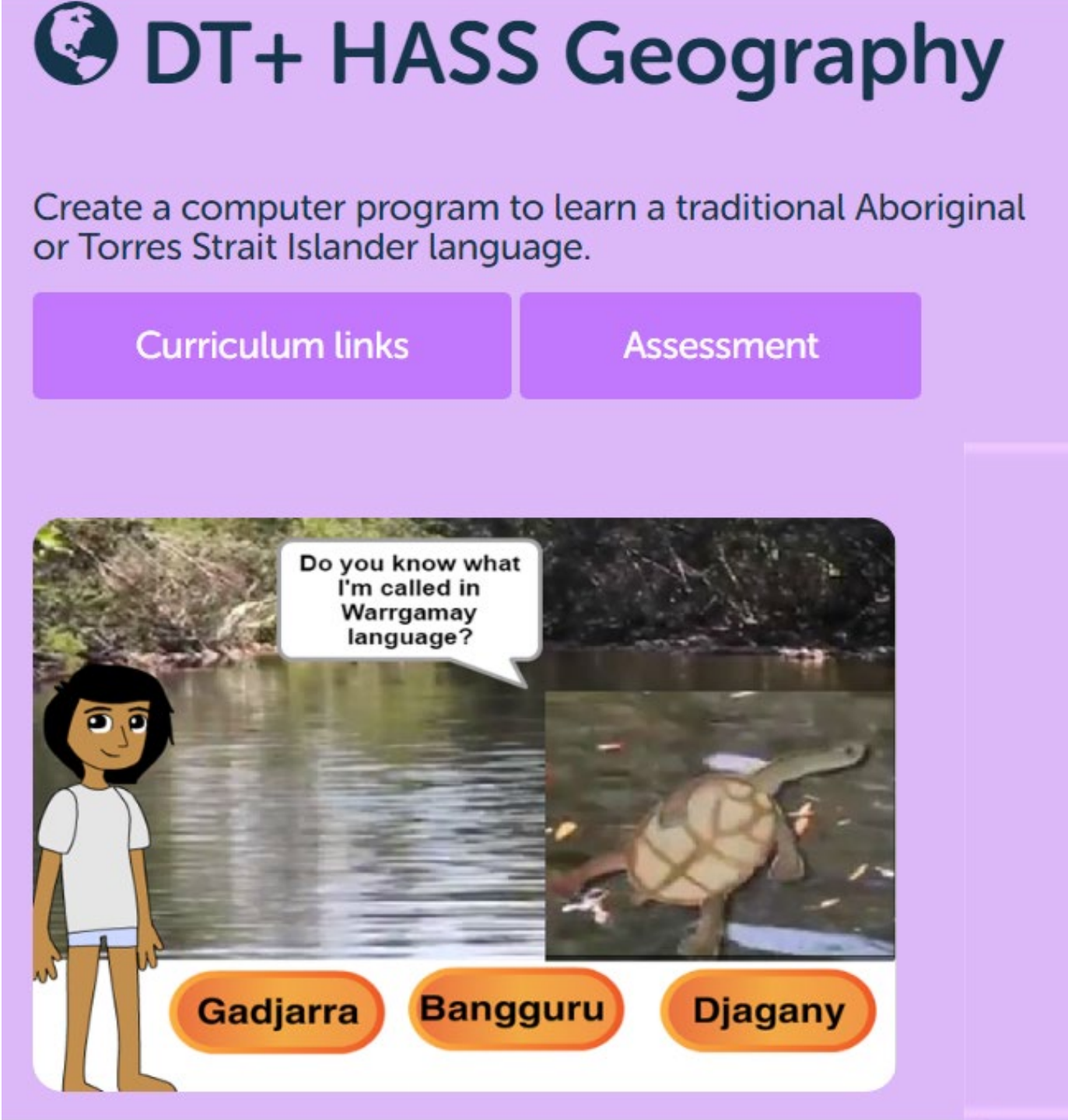
# Visual programming

Quiz

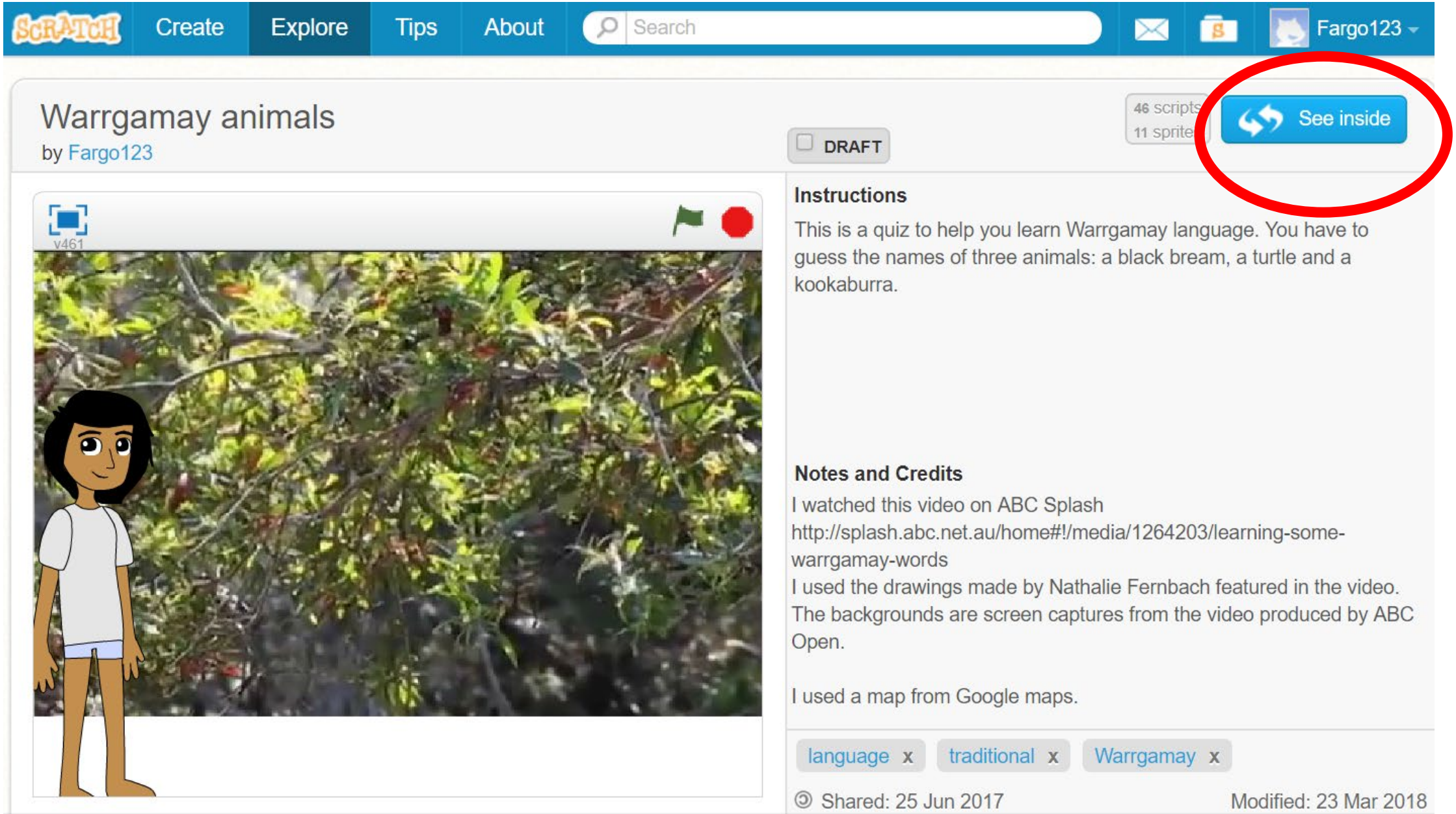
Consider

1. user input
2. user feedback
3. discuss with a ‘think aloud’ and algorithm

[Create a language-learning program]

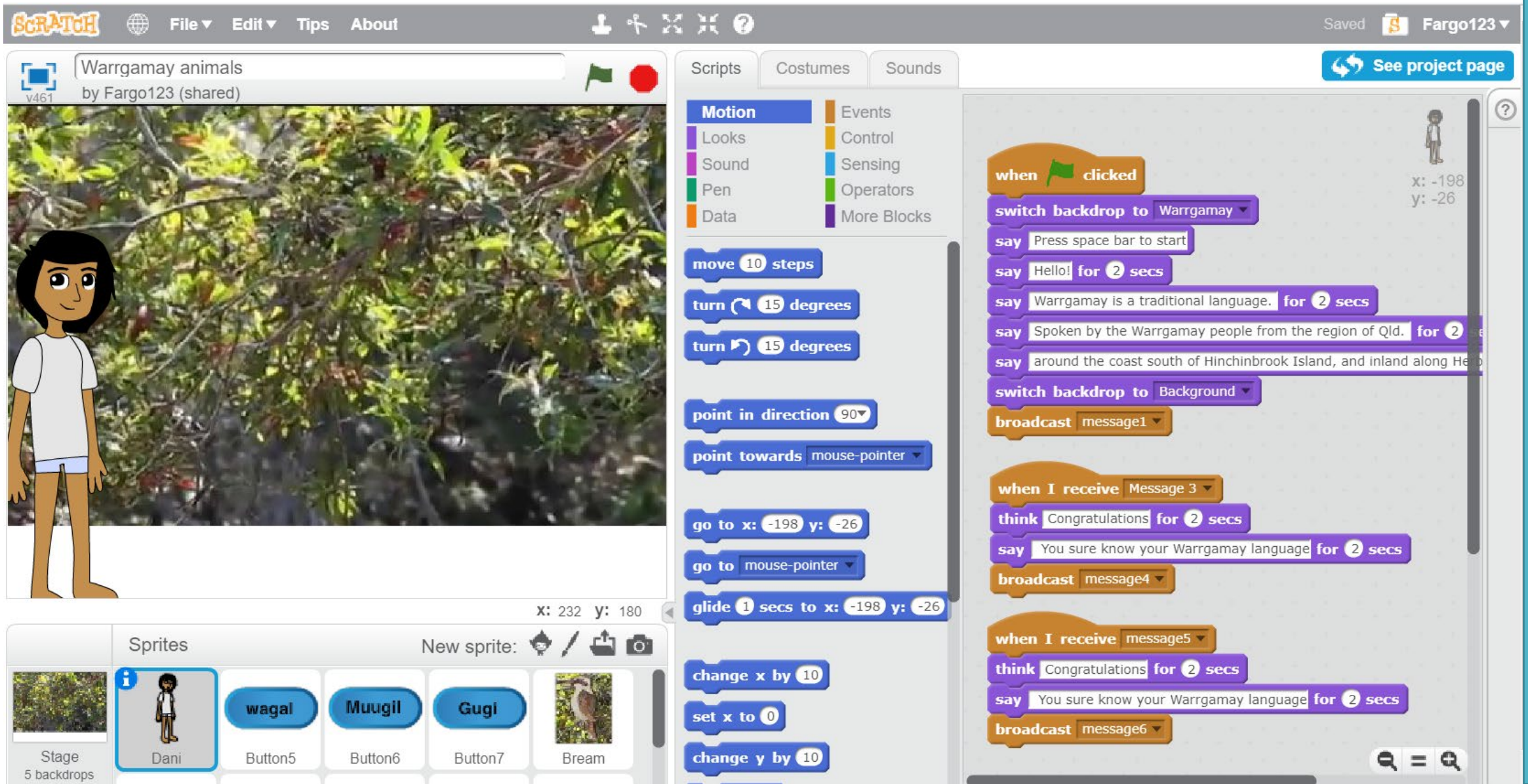


# Visual programming



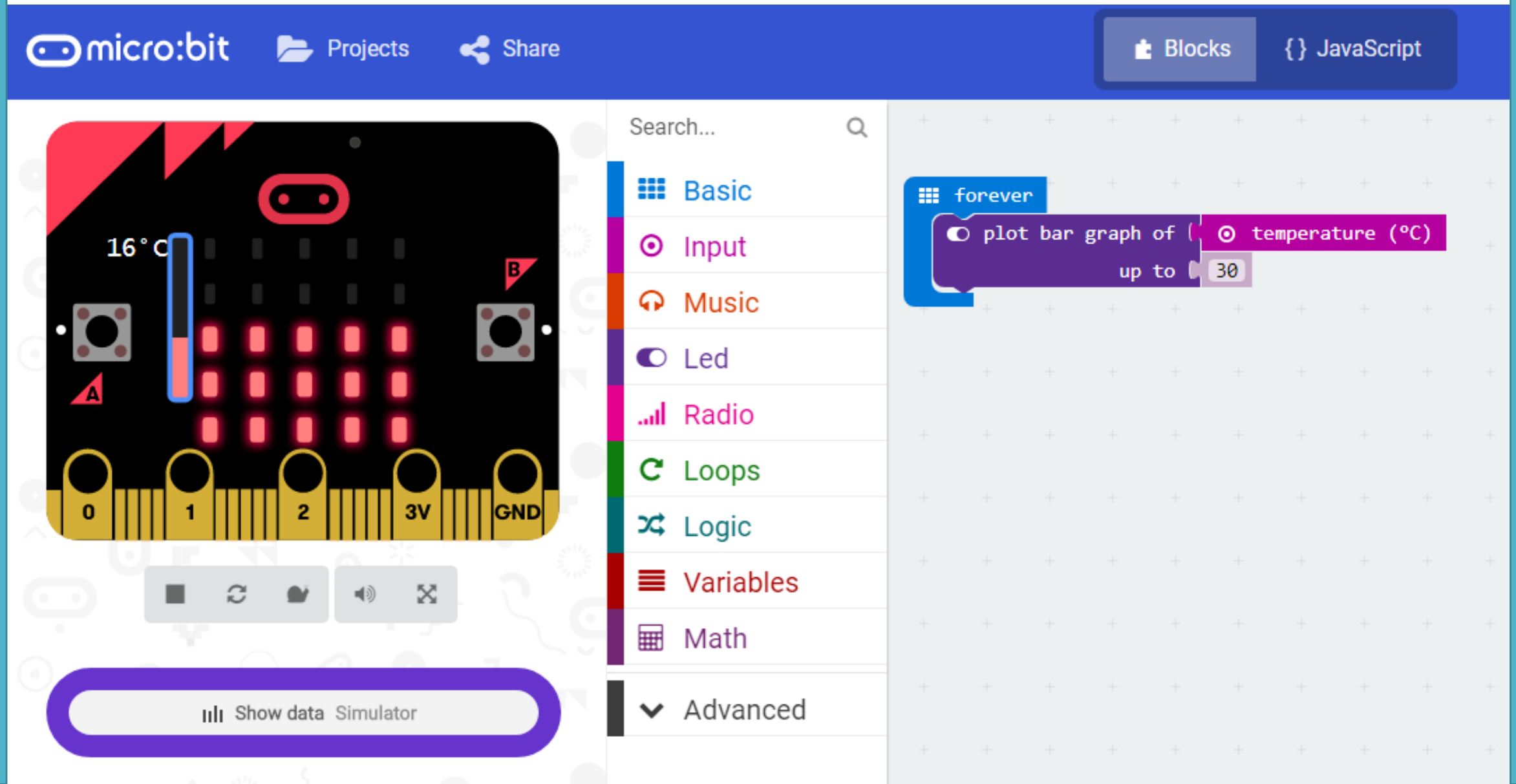
# Visual programming: Think aloud

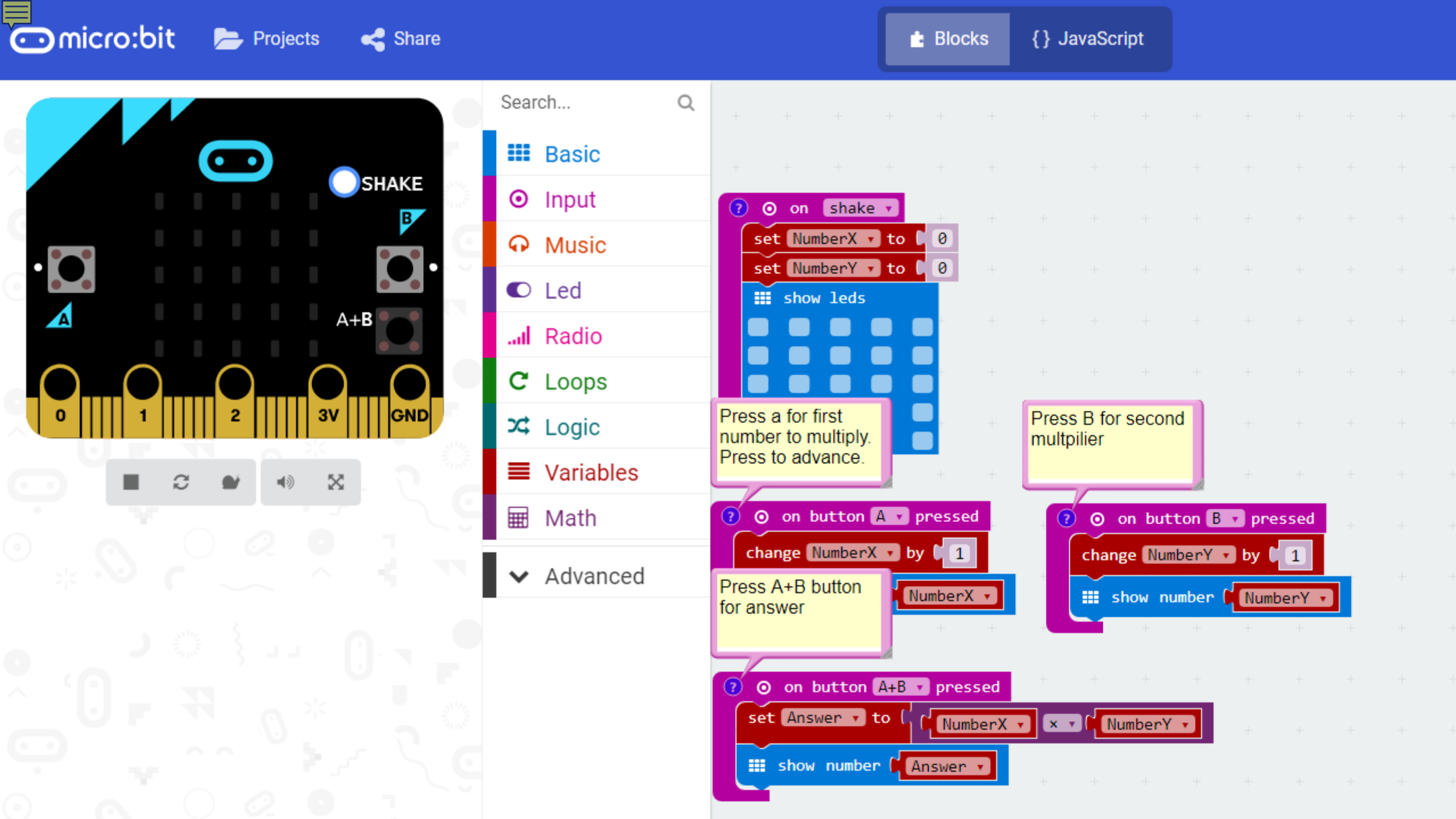
## Can you explain how this works? Can you show me as you tell me?



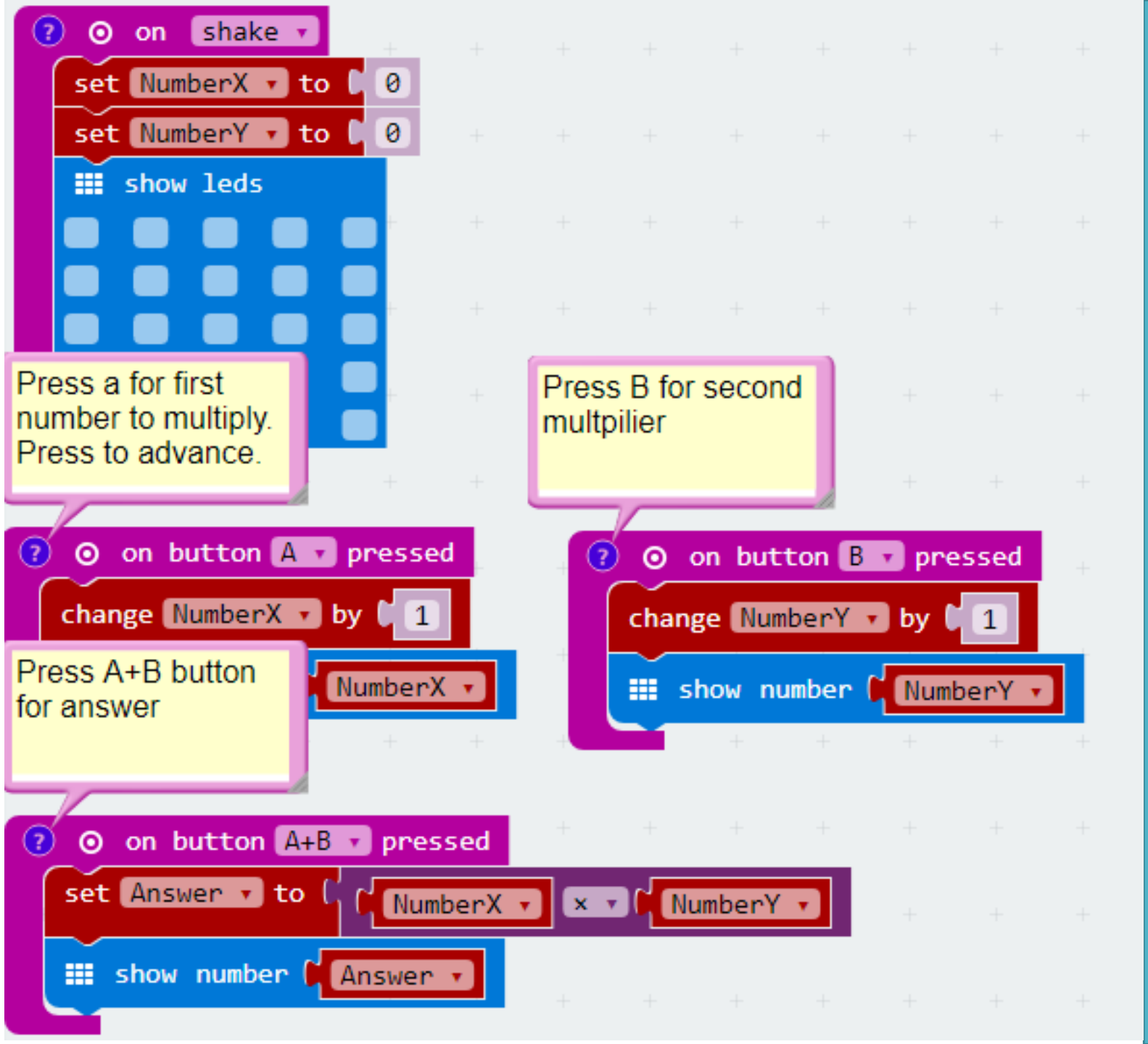
# BBC micro:bit

## Make your own temperature sensor to gather data

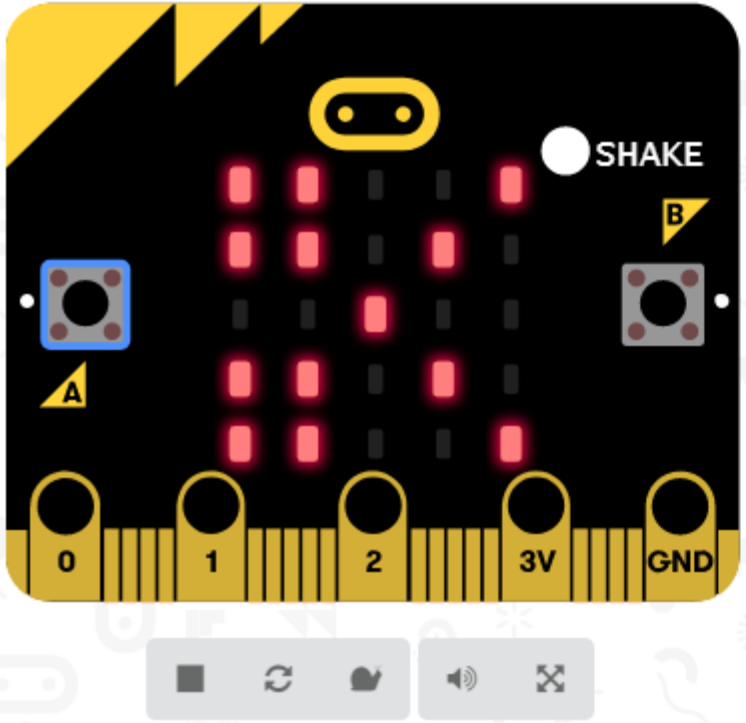


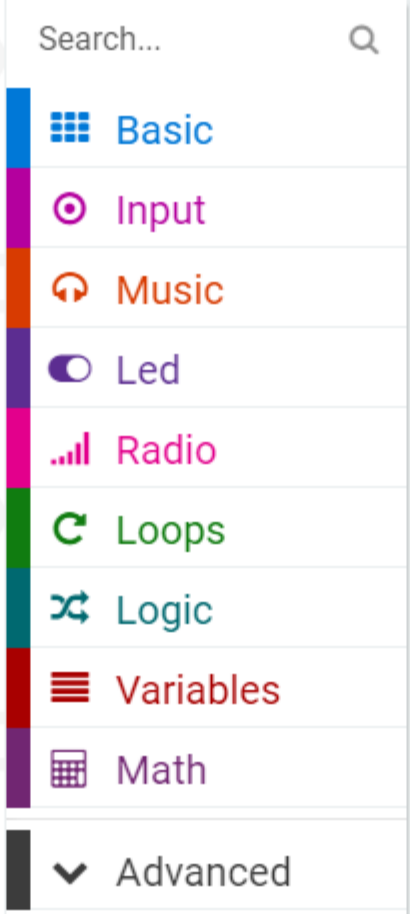

micro:bit
Projects
Share
Blocks
{} JavaScript
SHAKE
B
A
A+B
0
1
2
3V
GND
Search...
Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
Advanced
on shake
set NumberX to 0
set NumberY to 0
show leds
Press a for first number to multiply. Press to advance.
Press B for second multpilier
on button A pressed
change NumberX by 1
NumberX
Press A+B button for answer
on button B pressed
change NumberY by 1
show number NumberY
on button A+B pressed
set Answer to NumberX × NumberY
show number Answer

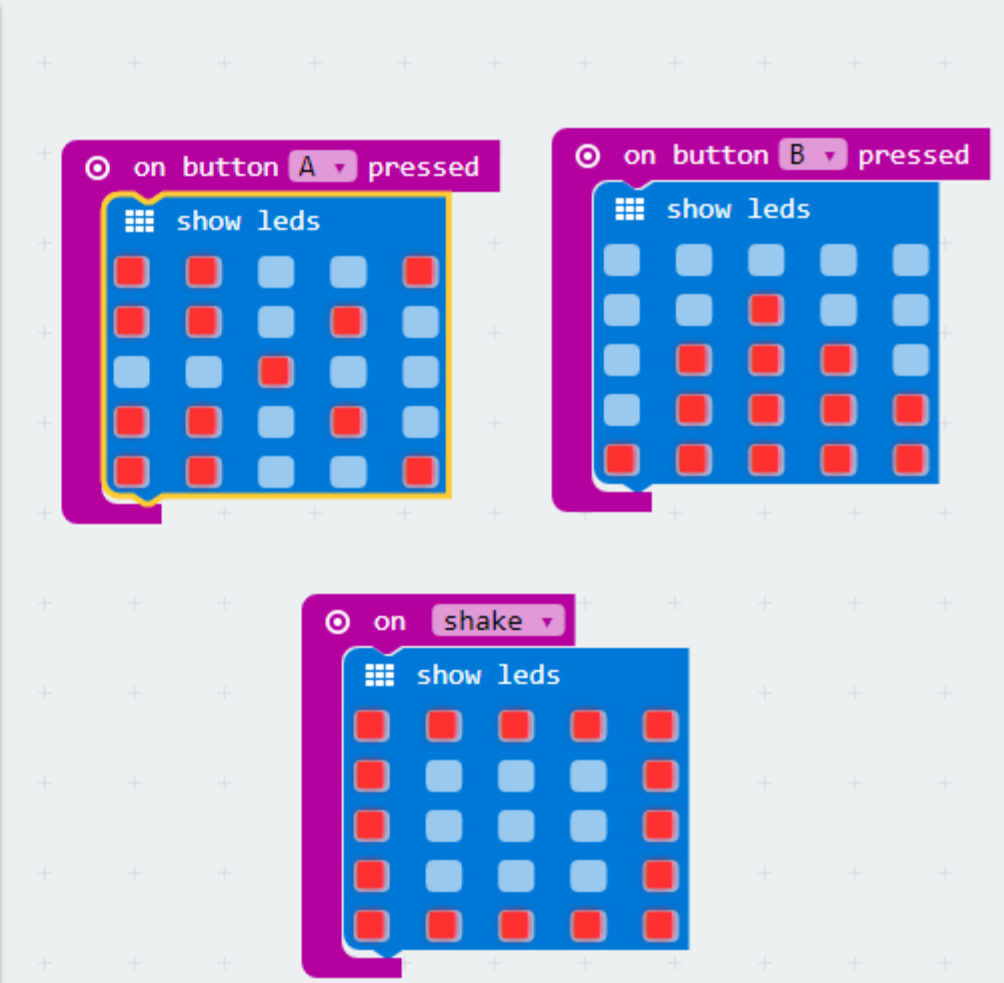
# BBC micro:bit Calculator



# BBC micro:bit: Game paper, scissors, rock



# BBC micro:bit: Game paper, scissors, rock

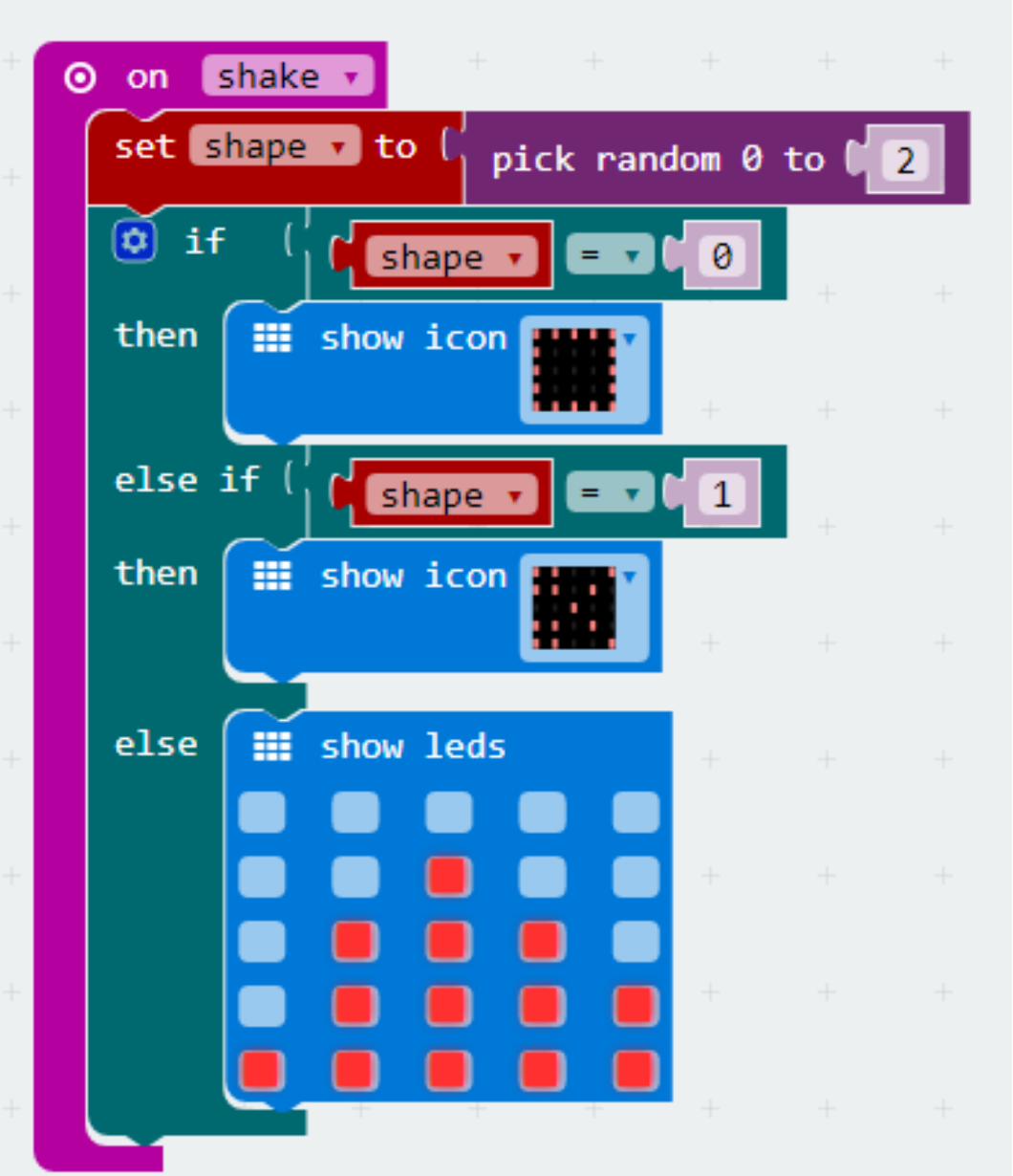
Make your own game

**Random**, 0, 1 or 2

**Variable: ‘shape’**

**Branching**: if/then

**User input**: on shake




https://makecode.microbit.org/_afPTb3UKmhUY

# BBC micro:bit:

Project log

Screen captures/video

Copies of programs saved in files

Think aloud

Project log

name _____

My project

| Date: __ / __ / __ | Date: __ / __ / __ | Date: __ / __ / __ |
|---|---|---|
| What I did: | What I did: | What I did: |
| I had trouble with: | I had trouble with: | I had trouble with: |
| I feel ☺ 😐 ☹ | I feel ☺ 😐 ☹ | I feel ☺ 😐 ☹ |

I can

Show LEDs

Use random value

Use if/then

Use different inputs

Show a string

Use maths blocks

Create a variable

Create a sound

Use a sensor

Use loop

**Secret messages and code** 5 hours Year 3
Explore ways to represent data using the context of secret messages and codes.

**Use data to solve problems** 7 hours Year 4
Use a meaningful context to collect and organise data to answer a question.

**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 INTRO TO PROGRAMMING

YEAR LEVEL: **3** TOPIC: **DIGITAL SOLUTIONS** TIME: **8 HOURS**

Programming is one process of the larger problem-solving methodology of creating digital solutions. Using a programming language can create a solution to a problem. The starting point for the problem-solving methodology is finding out about (investigating) and working out (defining) the problem. Once the problem has been defined the next step is to represent the solution as a series of steps (an algorithm). The algorithm can highlight any decisions (branching) that need to be made and what pathways might result, as well as how a user might engage and provide input. Algorithms at this level might be described verbally, written as a series of steps, represented on card, drawn or created digitally. The algorithm may then be implemented using a programming solution where students use a visual programming language that involves dragging and dropping programming blocks into a sequence. The final process is to evaluate how well their solution solved the problem.

Topic Overview

Topic Unit Map

Unit Overview

Unit SOLO Taxonomy

## FLOW OF ACTIVITIES

1

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

Learn More

2

**Coming up with a solution**
Create a storyboard or flow chart to record relationships between the content and processes.

Learn More

3

**Implementing a solution**
Use a visual programming language as part of the digital solution.

Learn More

4

**Evaluation**
Evaluate how well the solution met the desired outcome.

Learn More

# 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