Learning to code using ro-Bots.

Presentation Overview

- STEM - PBL - Code
- Why learn to code?
- Basic coding terminology
- Bots and Apps.
- Challenges.
- Your turn to code.

PBL

‘**Project-based learning (PBL)** is an approach to teaching and learning that engages students in rich and authentic…hands-on, interactive learning experiences…

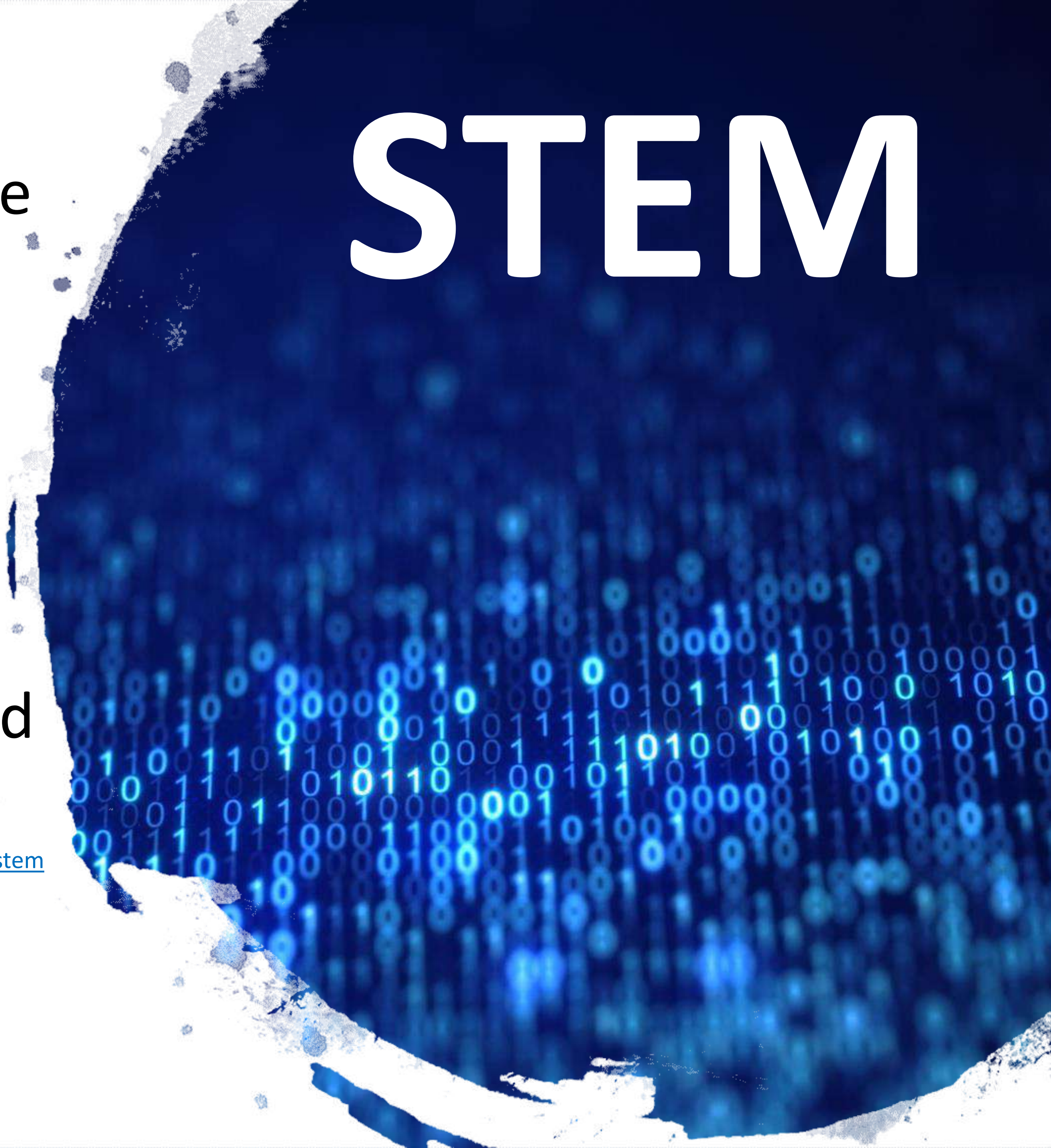
…students gain knowledge and skills by **investigating and responding to an engaging question, problem or challenge**.’

https://education.nsw.gov.au/teaching-and-learning/curriculum/learning-for-the-future/future-focused-learning-and-teaching/project-based-learning-resource-guide/introducing-project-based-learning

STEM

‘STEM is a curriculum based on the idea of educating students in four specific disciplines — **science, technology, engineering** and **mathematics** — in an **interdisciplinary** and applied approach.’

https://education.nsw.gov.au/teaching-and-learning/curriculum/learning-for-the-future/stem



Coding

‘…is a list of **step-by-step instructions** that get computers to do what you want them to do.’

https://www.learningpotential.gov.au/what-is-coding

‘… let’s define coding as the basic act of writing – in a programming language – a **script** that a computer can understand.’

https://www.bitdegree.org/tutorials/what-is-coding/

It’s‘…the new literacy for the 21st century.’

‘Coding…’

‘Computer Programming…

https://ase.tufts.edu/DevTech/publications/computersandeducation.pdf

Why learn to code?

It is the future!

We need more computer scientists!

Why learn to code?

nurtures and expands creative expression

teaches causality (cause and effect)

teaches problem solving (DEBUG)

gives students a challenge and helps them develop resilience and persistence

students learn by thinking about doing

http://educationstandards.nsw.edu.au/wps/portal/nesa/k-10/learning-areas/technologies/coding-across-the-curriculum

Why learn to code?

Learning to code teaches children how to **think**. Computer programming isn't just about teaching how to type lines of code. It is more about teaching children how to **think differently**. Being able to code effectively, a programmer needs to use **logical thinking**. They need to be able to **see a large problem** and **break it down** into smaller pieces in order to **solve** it in an effective manner.

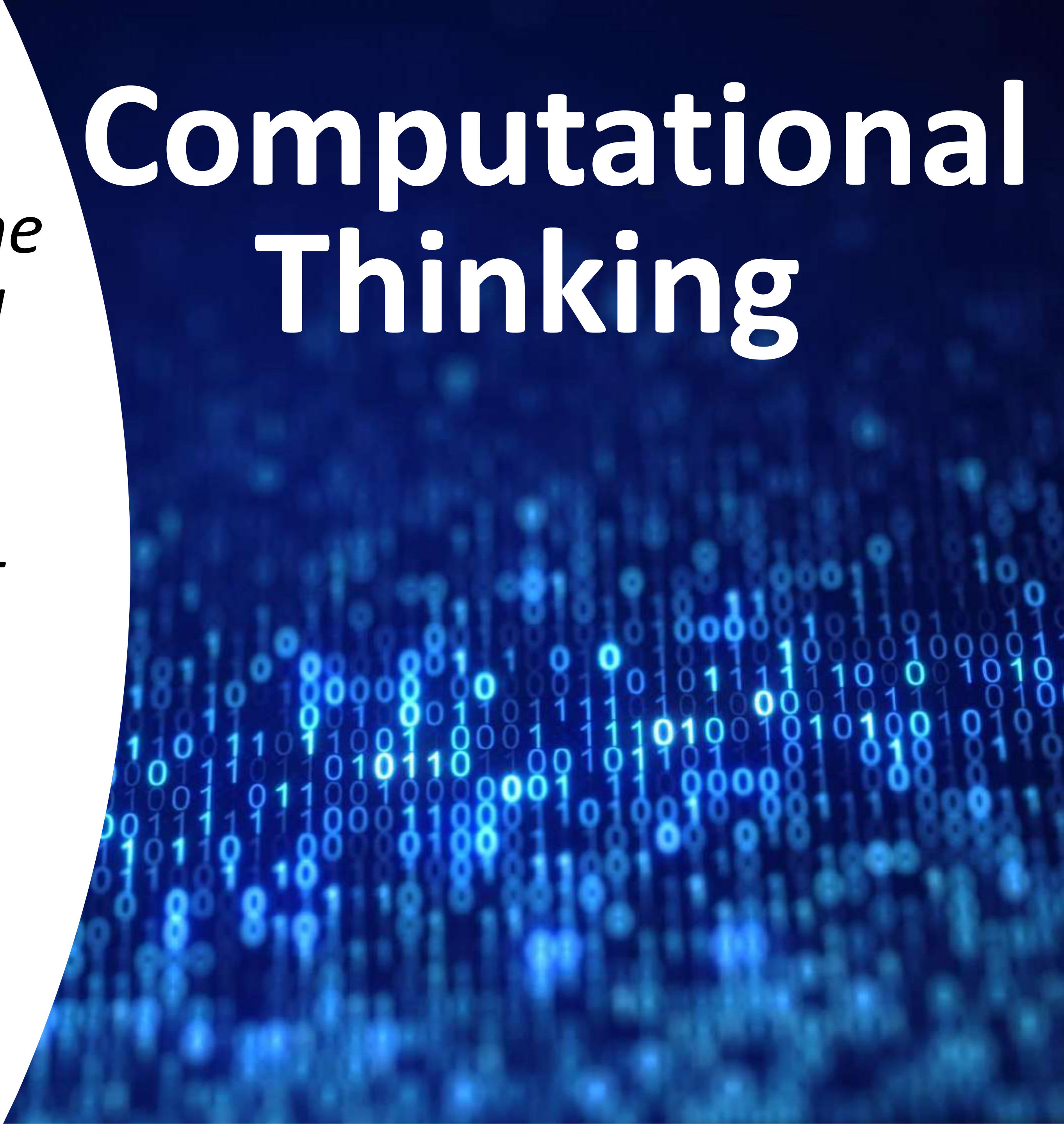
This is called **decomposition** and is one of the key features of **computational thinking.**

http://educationstandards.nsw.edu.au/wps/portal/nesa/k-10/learning-areas/technologies/coding-across-the-curriculum

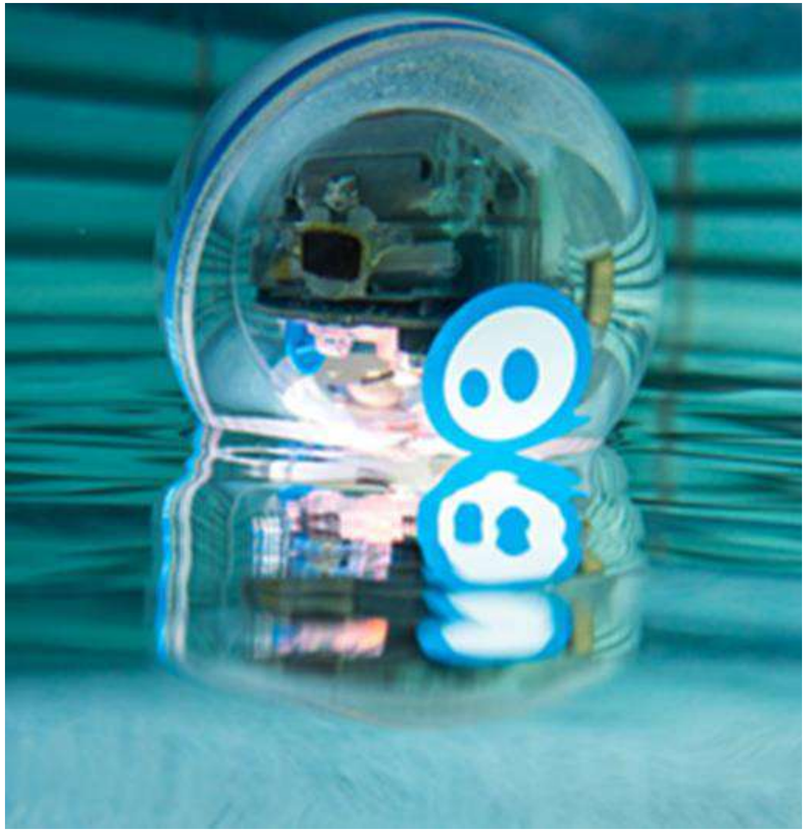
Computational Thinking

‘Computational thinking is the ***thought processes*** *involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out.’*

http://educationstandards.nsw.edu.au/wps/portal/nesa/k-10/learning-areas/technologies/coding-across-the-curriculum



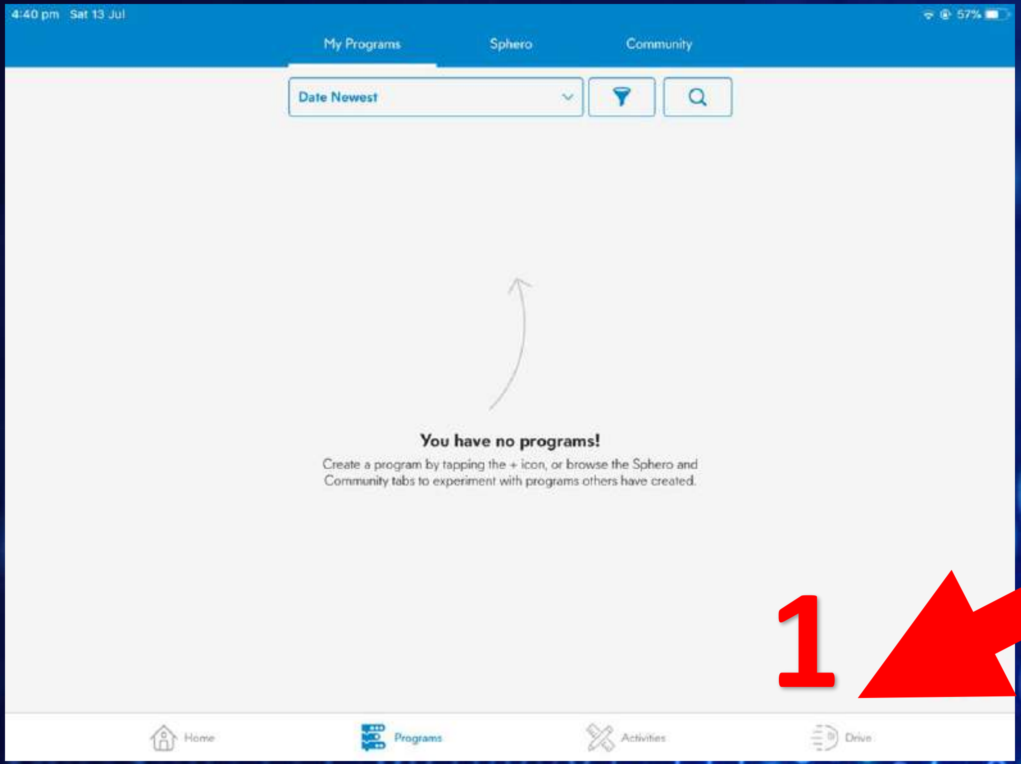
How it all started?



What is Sphero?

VIDEO

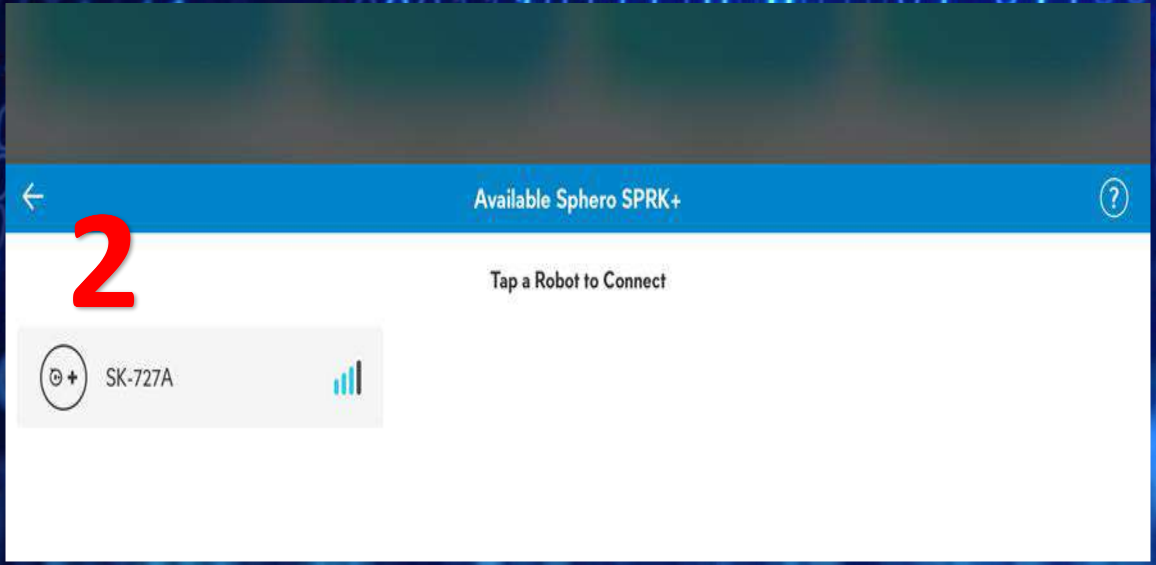
Sphero Basics



You have no programs!

Create a program by tapping the + icon, or browse the Sphero and Community tabs to experiment with programs others have created.

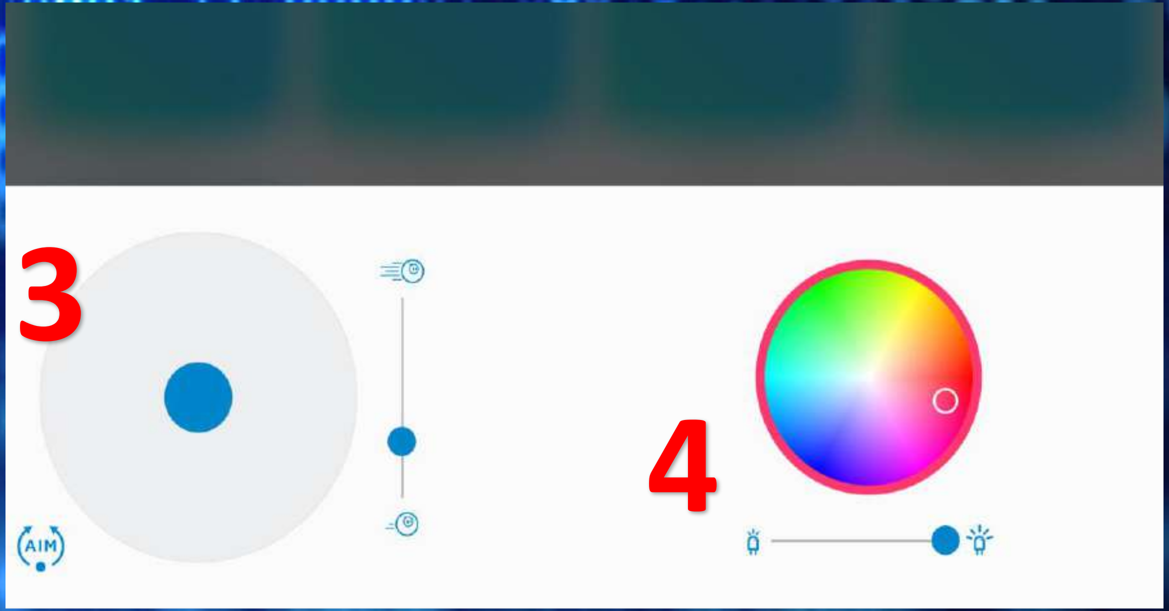
1



Tap a Robot to Connect

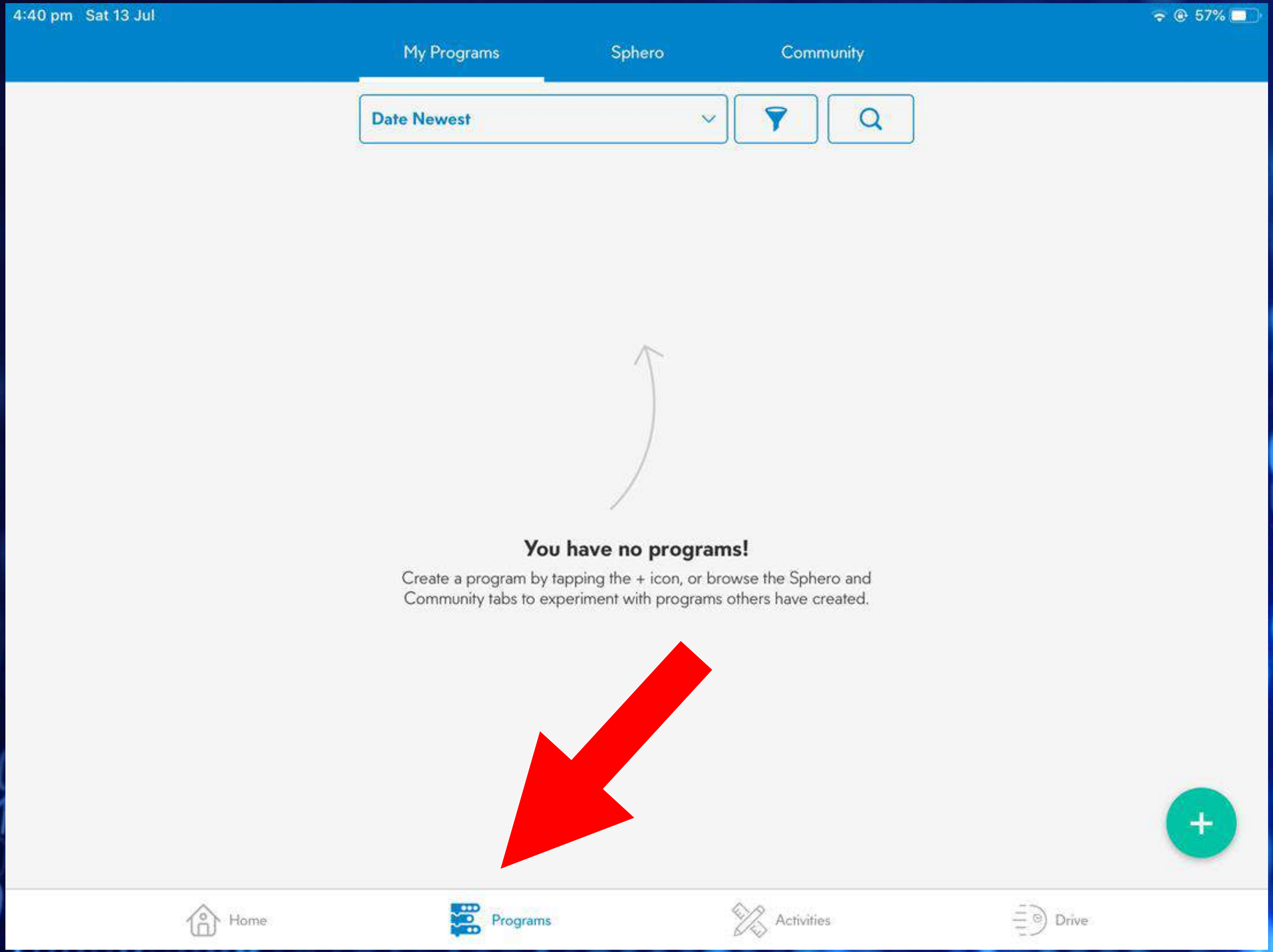
SK-727A

2



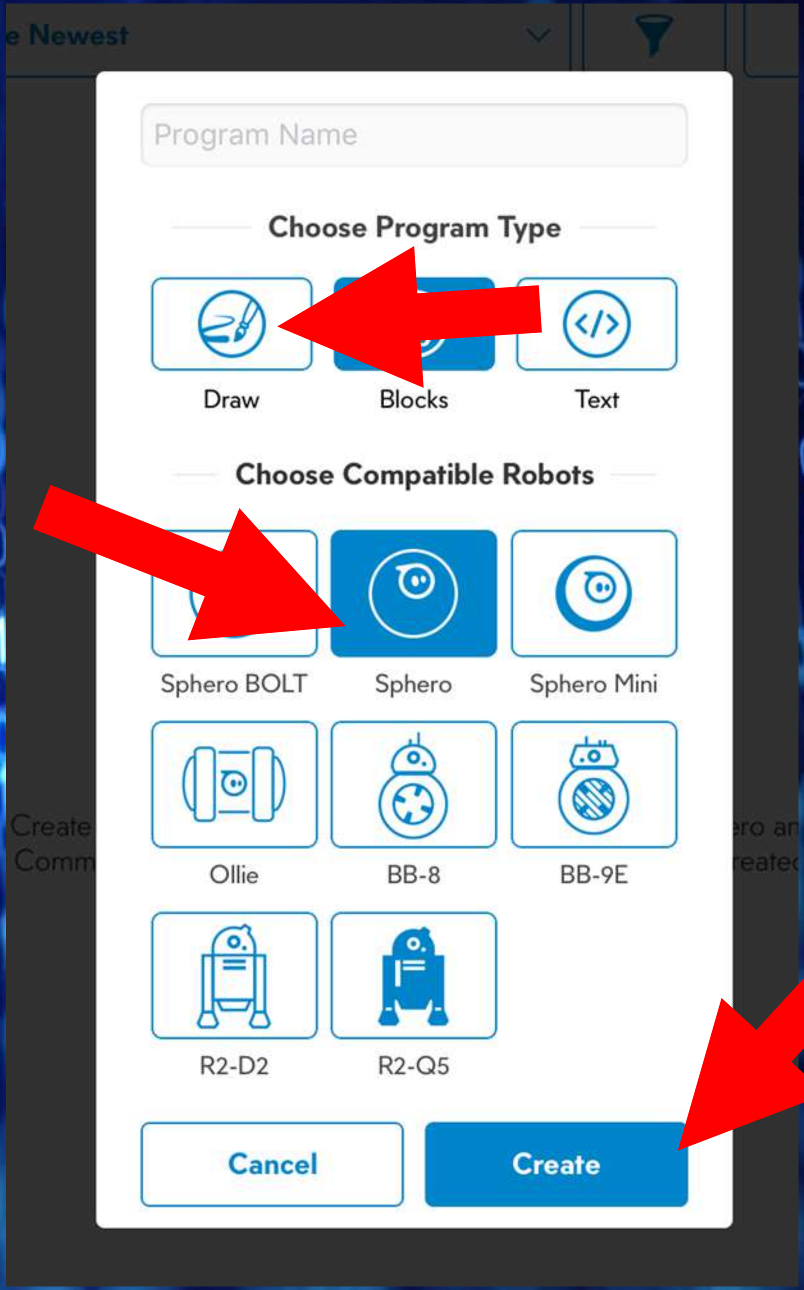
3

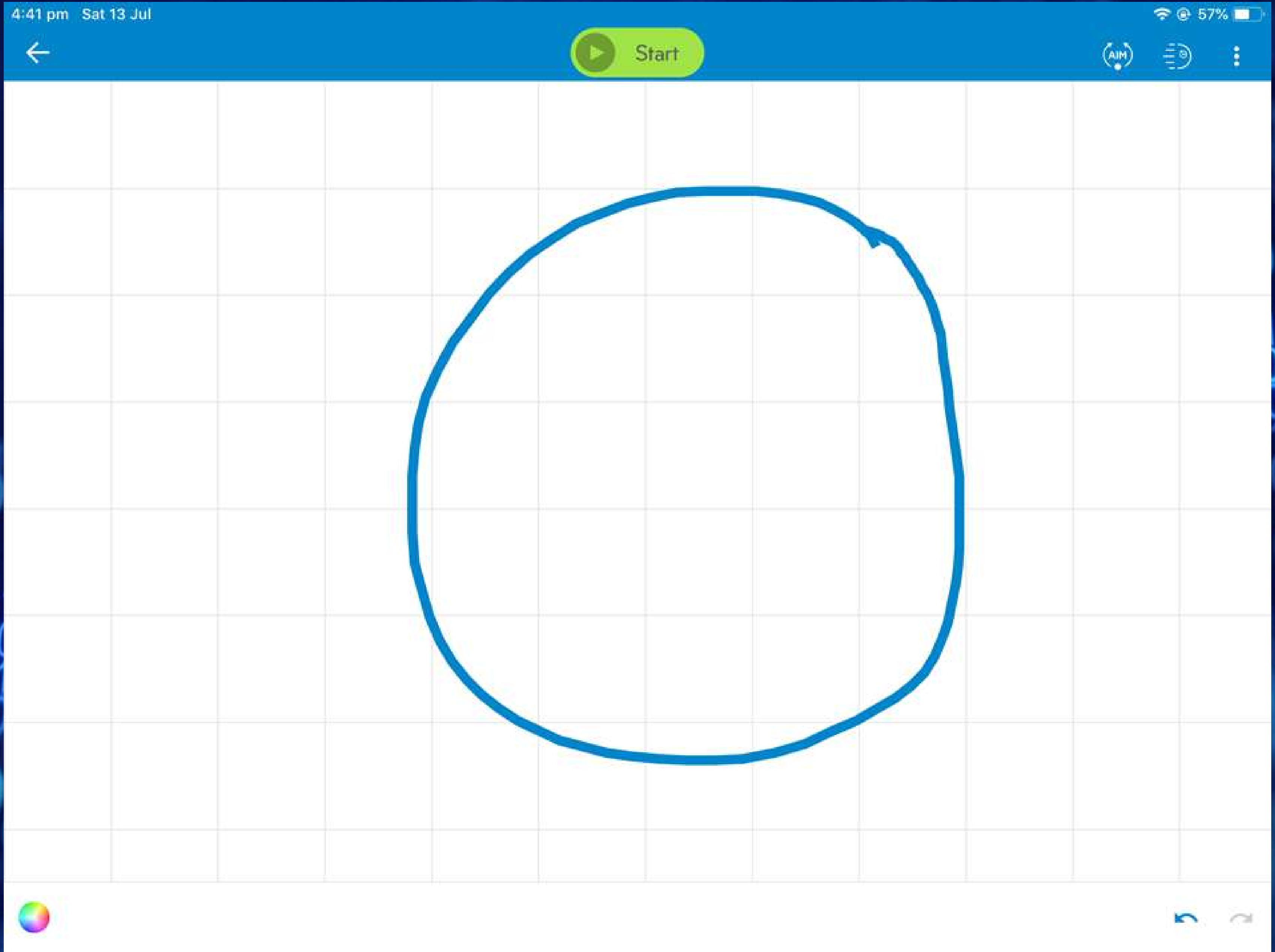
4

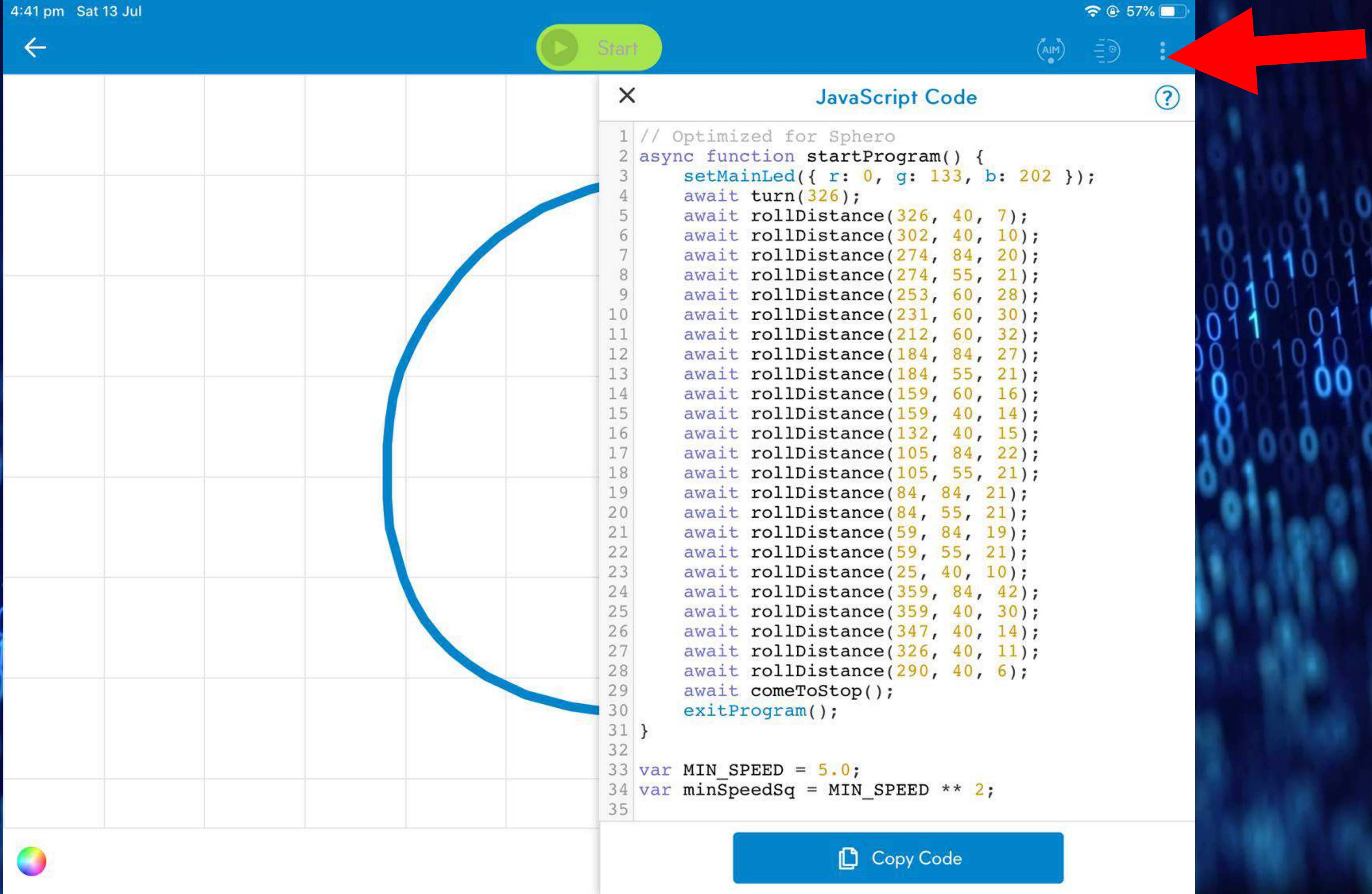


You have no programs!

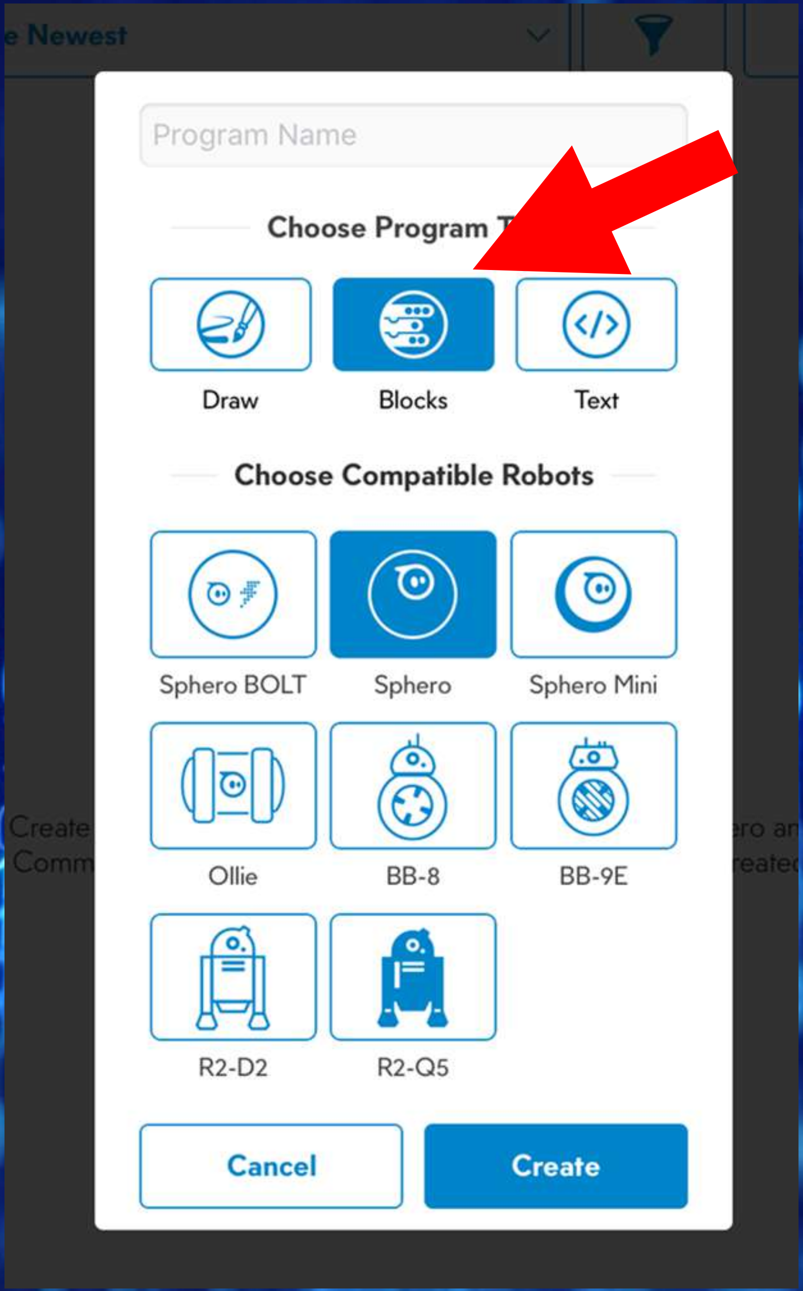
Create a program by tapping the + icon, or browse the Sphero and Community tabs to experiment with programs others have created.

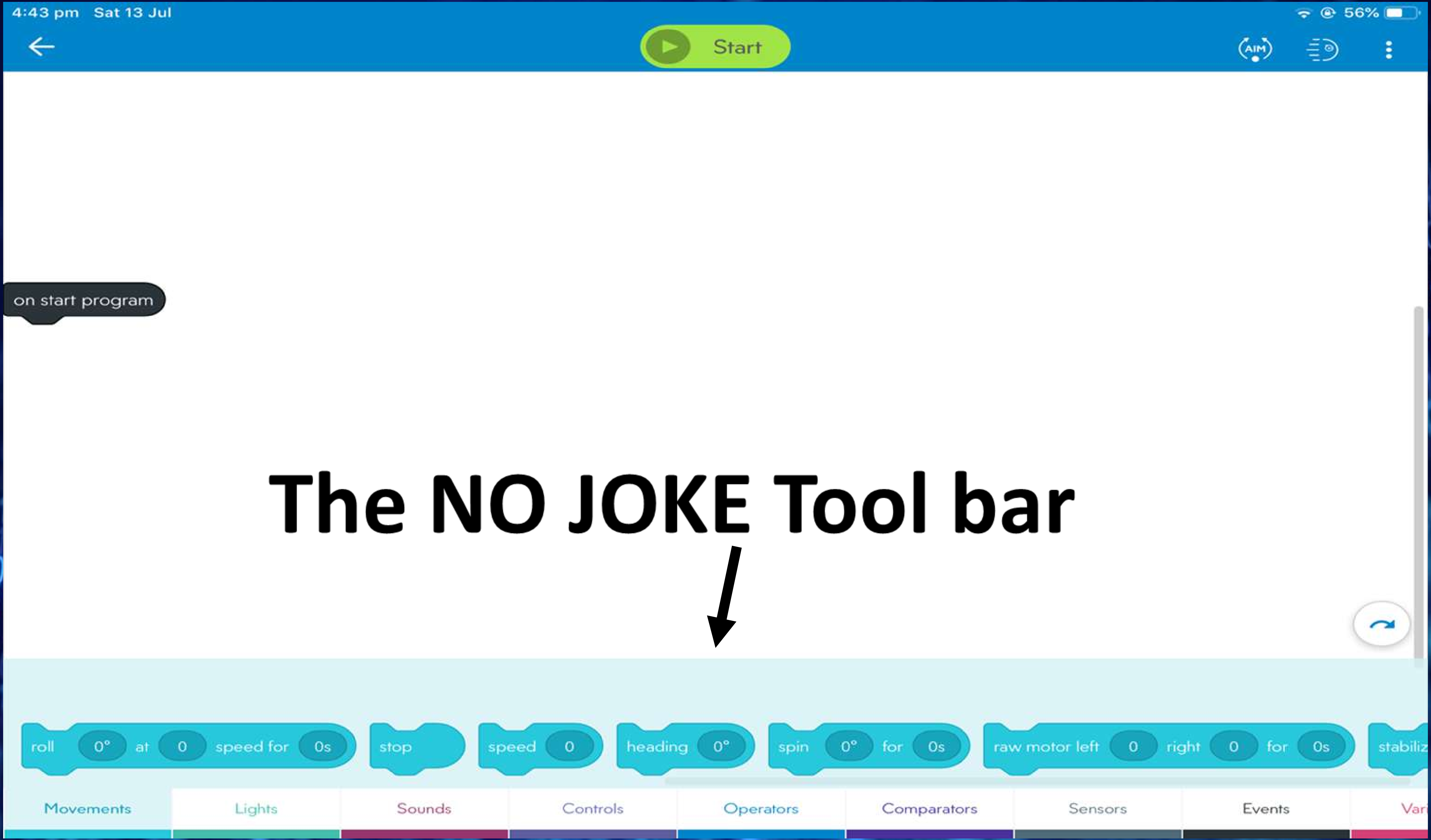



4:41 pm Sat 13 Jul
57%
Start
AIM

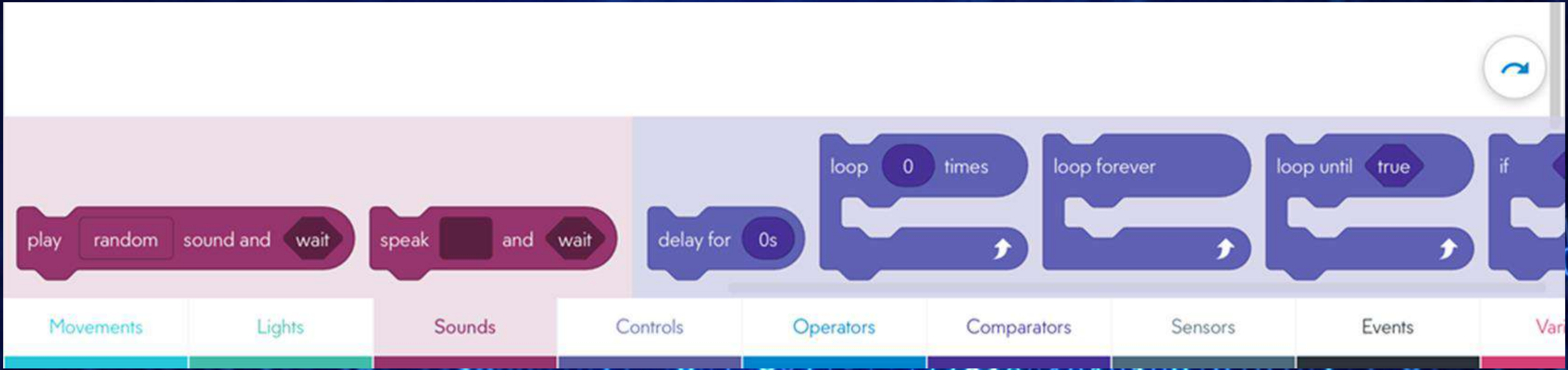


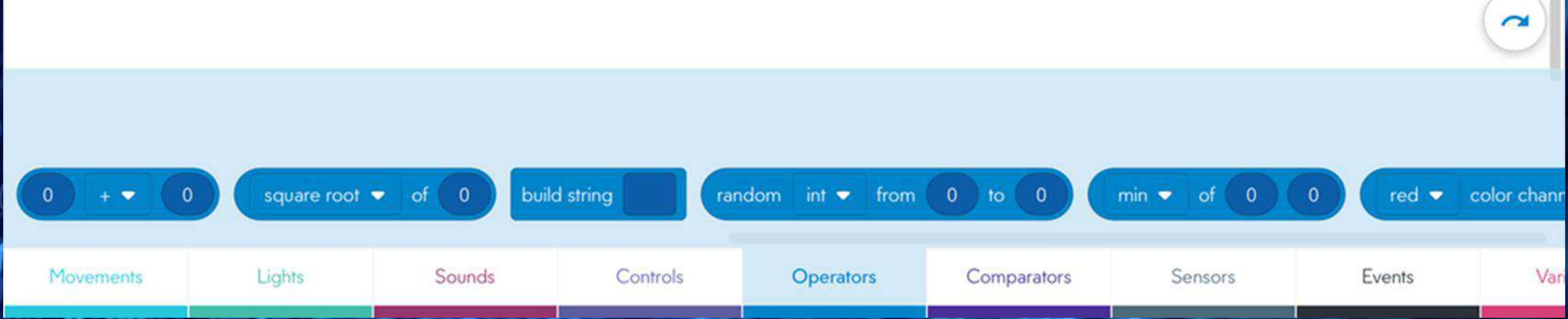
```
// Optimized for Sphero
async function startProgram() {
    setMainLed({ r: 0, g: 133, b: 202 });
    await turn(326);
    await rollDistance(326, 40, 7);
    await rollDistance(302, 40, 10);
    await rollDistance(274, 84, 20);
    await rollDistance(274, 55, 21);
    await rollDistance(253, 60, 28);
    await rollDistance(231, 60, 30);
    await rollDistance(212, 60, 32);
    await rollDistance(184, 84, 27);
    await rollDistance(184, 55, 21);
    await rollDistance(159, 60, 16);
    await rollDistance(159, 40, 14);
    await rollDistance(132, 40, 15);
    await rollDistance(105, 84, 22);
    await rollDistance(105, 55, 21);
    await rollDistance(84, 84, 21);
    await rollDistance(84, 55, 21);
    await rollDistance(59, 84, 19);
    await rollDistance(59, 55, 21);
    await rollDistance(25, 40, 10);
    await rollDistance(359, 84, 42);
    await rollDistance(359, 40, 30);
    await rollDistance(347, 40, 14);
    await rollDistance(326, 40, 11);
    await rollDistance(290, 40, 6);
    await comeToStop();
    exitProgram();
}

var MIN_SPEED = 5.0;
var minSpeedSq = MIN_SPEED ** 2;

```


e Newest
Program Name
Choose Program T
Draw
Blocks
Text
Choose Compatible Robots
Sphero BOLT
Sphero
Sphero Mini
Ollie
BB-8
BB-9E
R2-D2
R2-Q5
Cancel
Create
Create
Comm
ero an
reated

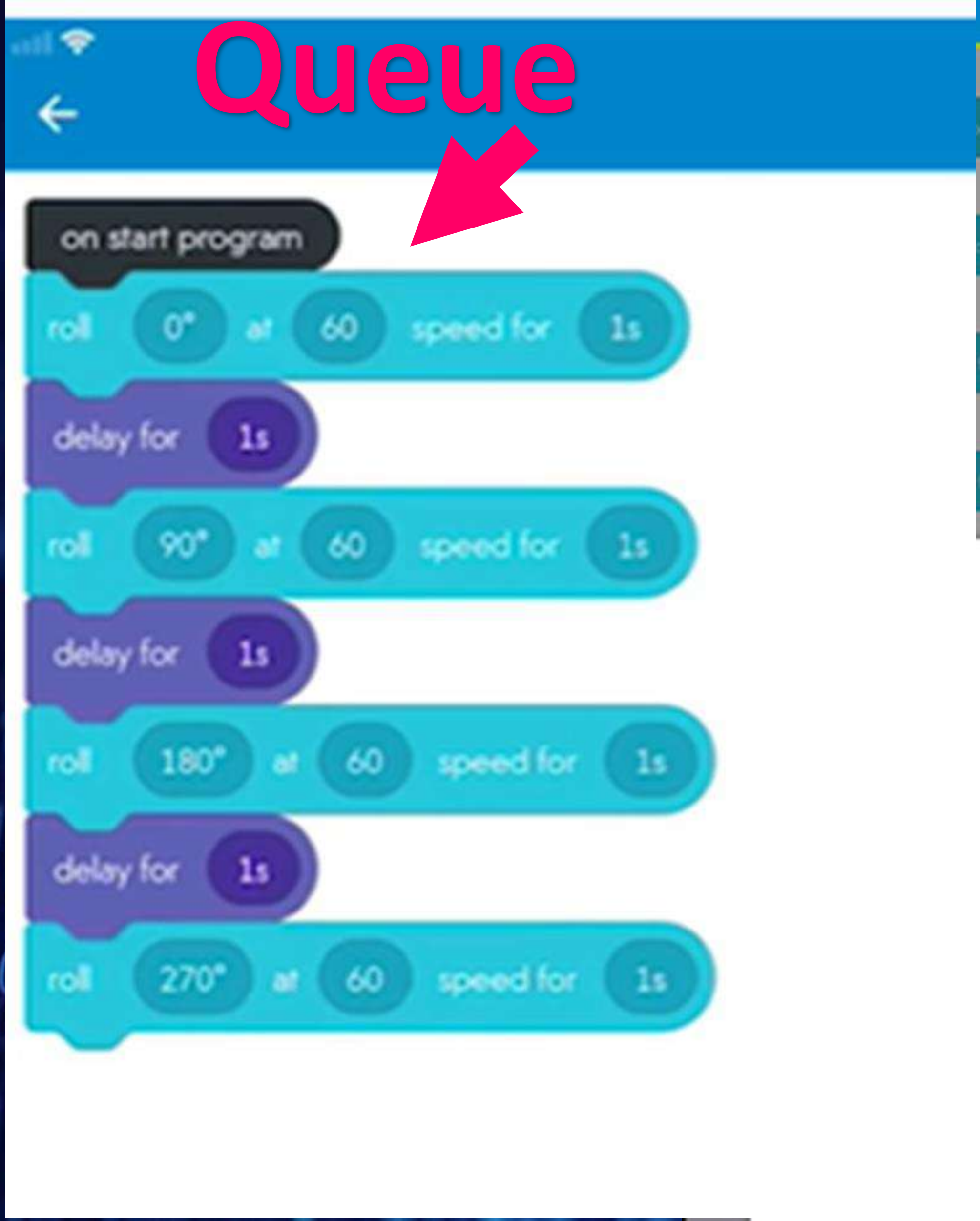


The NO JOKE Tool bar

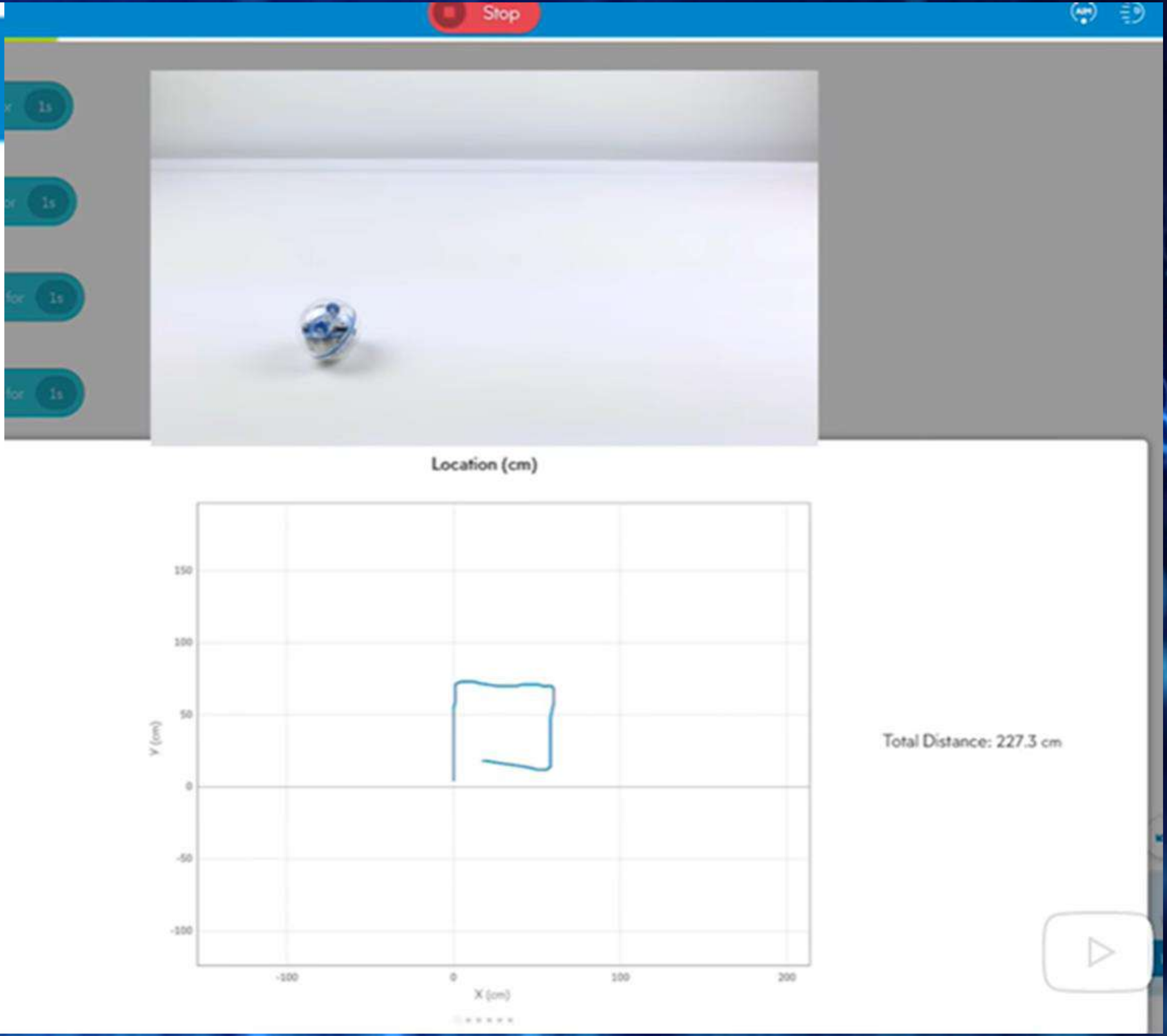

play
random
sound and
wait
speak
and
wait
delay for
0s
loop
0
times
loop forever
loop until
true
if
Movements
Lights
Sounds
Controls
Operators
Comparators
Sensors
Events
Vari


0
+
0
square root
of
0
build string
random
int
from
0
to
0
min
of
0
0
red
color chann
Movements
Lights
Sounds
Controls
Operators
Comparators
Sensors
Events
Vari

Queue

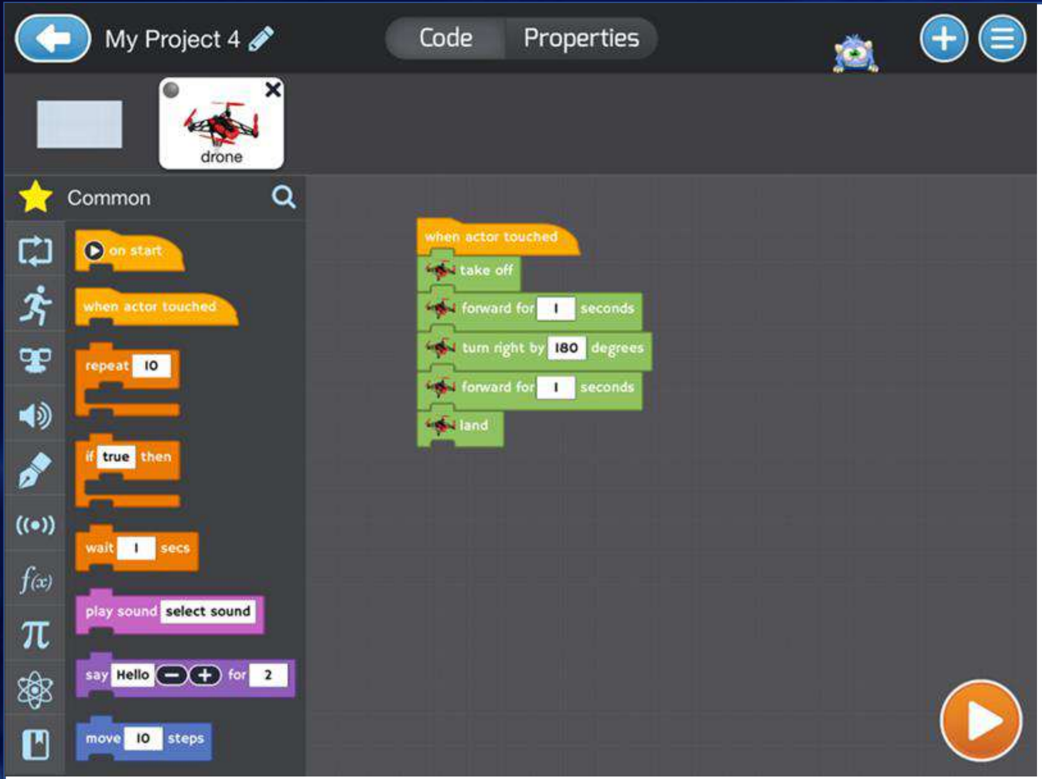


```
on start program
roll 0° at 60 speed for 1s
delay for 1s
roll 90° at 60 speed for 1s
delay for 1s
roll 180° at 60 speed for 1s
delay for 1s
roll 270° at 60 speed for 1s
```



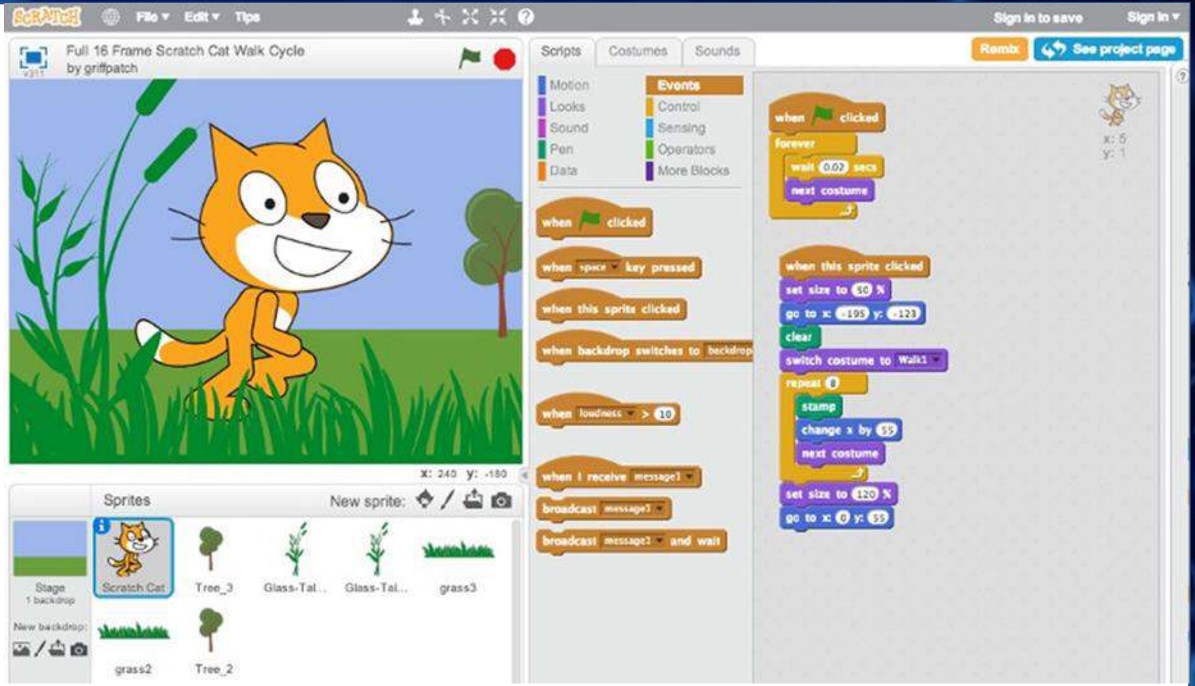
Total Distance: 227.3 cm

https://edu.sphero.com/cwists/preview/1671x



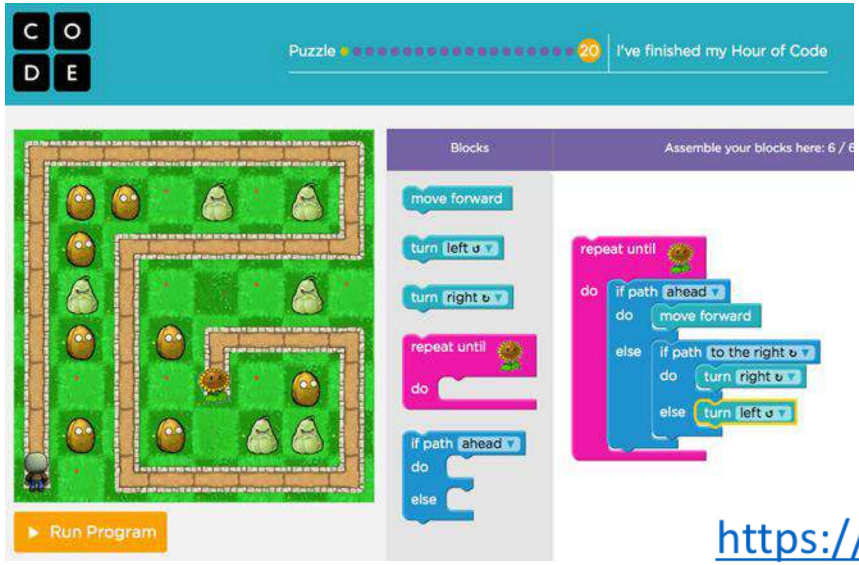
```
when actor touched
  take off
  forward for 1 seconds
  turn right by 180 degrees
  forward for 1 seconds
  land
```

https://www.tynker.com/



```
when clicked
forever
  wait 0.02 secs
  next costume

when this sprite clicked
set size to 50 %
go to x: -195 y: -123
clear
switch costume to Walk1
repeat 8
  stamp
  change x by 55
  next costume
set size to 120 %
go to x: 0 y: 55
```

https://scratch.mit.edu/



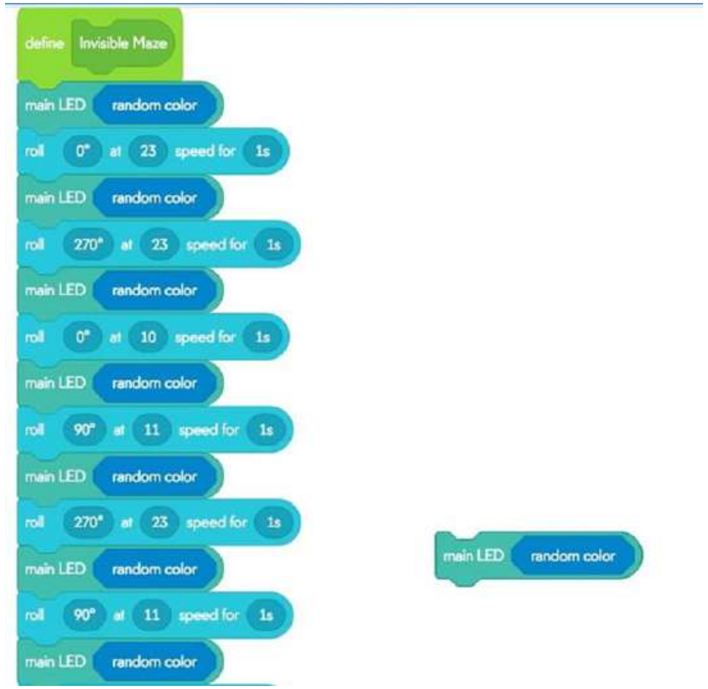
```
repeat until
do  if path ahead
    do  move forward
    else  if path to the right ↻
          do  turn right ↻
          else  turn left ↺
```

https://code.org/

Coding Terminology

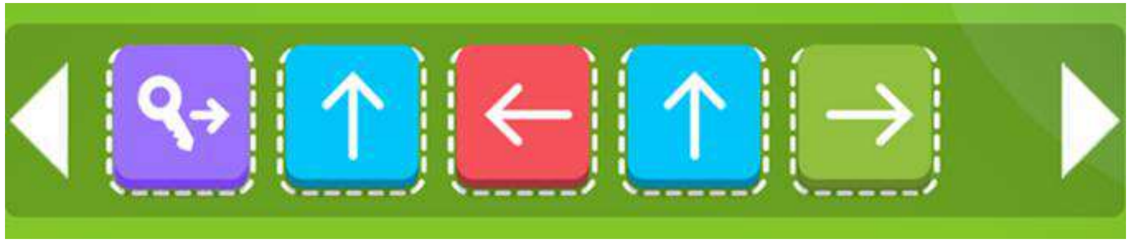
Coding Terminology

Queue - how the algorithm is presented (a visual schedule)



```
define Invisible Maze
main LED random color
roll 0° at 23 speed for 1s
main LED random color
roll 270° at 23 speed for 1s
main LED random color
roll 0° at 10 speed for 1s
main LED random color
roll 90° at 11 speed for 1s
main LED random color
roll 270° at 23 speed for 1s
main LED random color
roll 90° at 11 speed for 1s
main LED random color
```

```
main LED random color
```



Coding Terminology

Algorithm – a precise set of instructions (each instruction is a coding block)

Coding Terminology

Algorithm – a precise set of instructions (each instruction is a coding block)

stomp
right leg

stomp
left leg

Coding Terminology

stomp
right leg

stomp
left leg

Coding Terminology

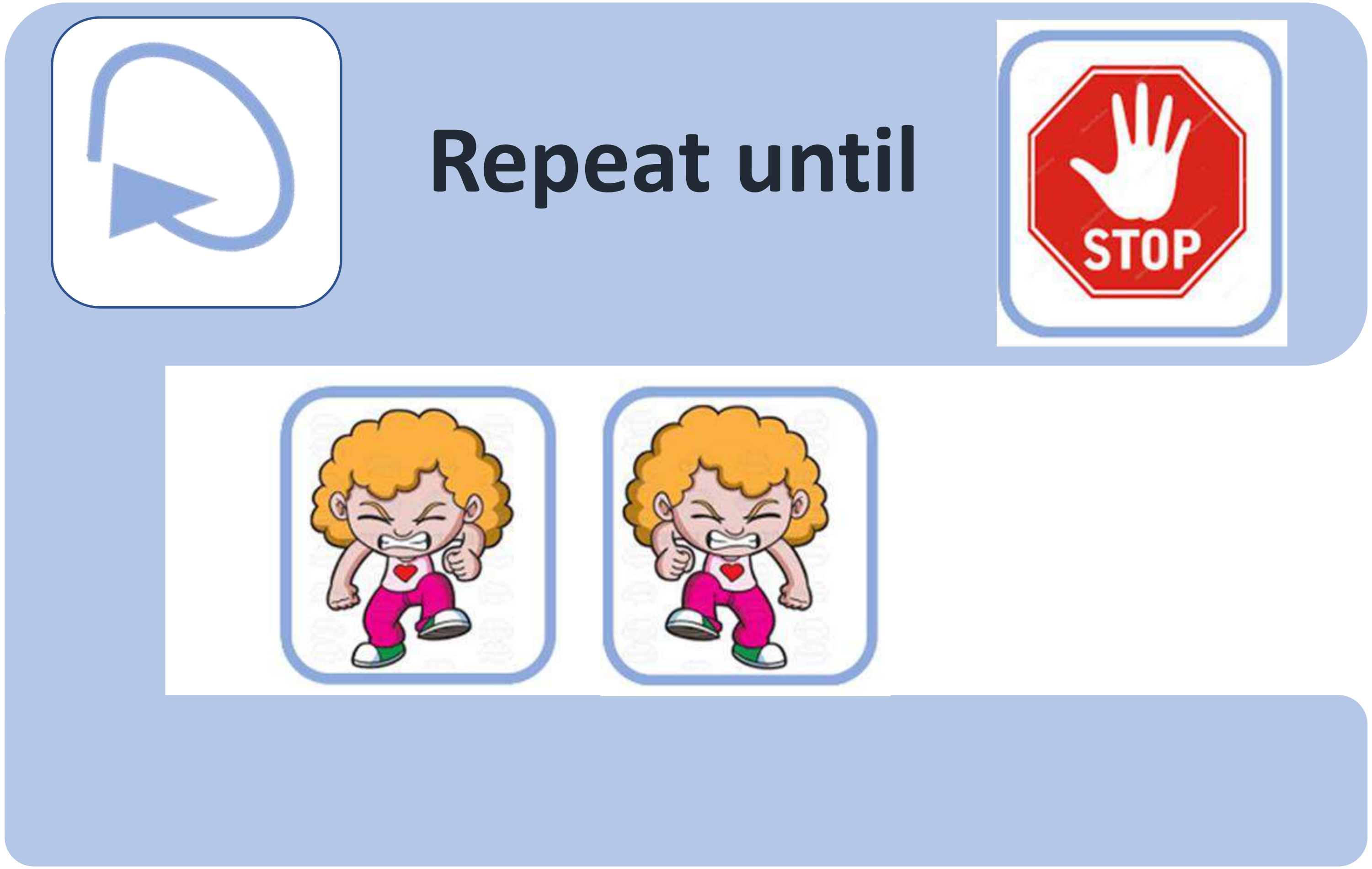
stomp
right leg

stomp
left leg

Coding Terminology

Loop – an instruction which is repeated until an indicated point

Coding Terminology



Repeat until

Coding Terminology

right leg stomp

Coding Terminology

left leg stomp

Coding Terminology

right leg stomp

Coding Terminology

left leg stomp

Coding Terminology

right leg stomp

Coding Terminology

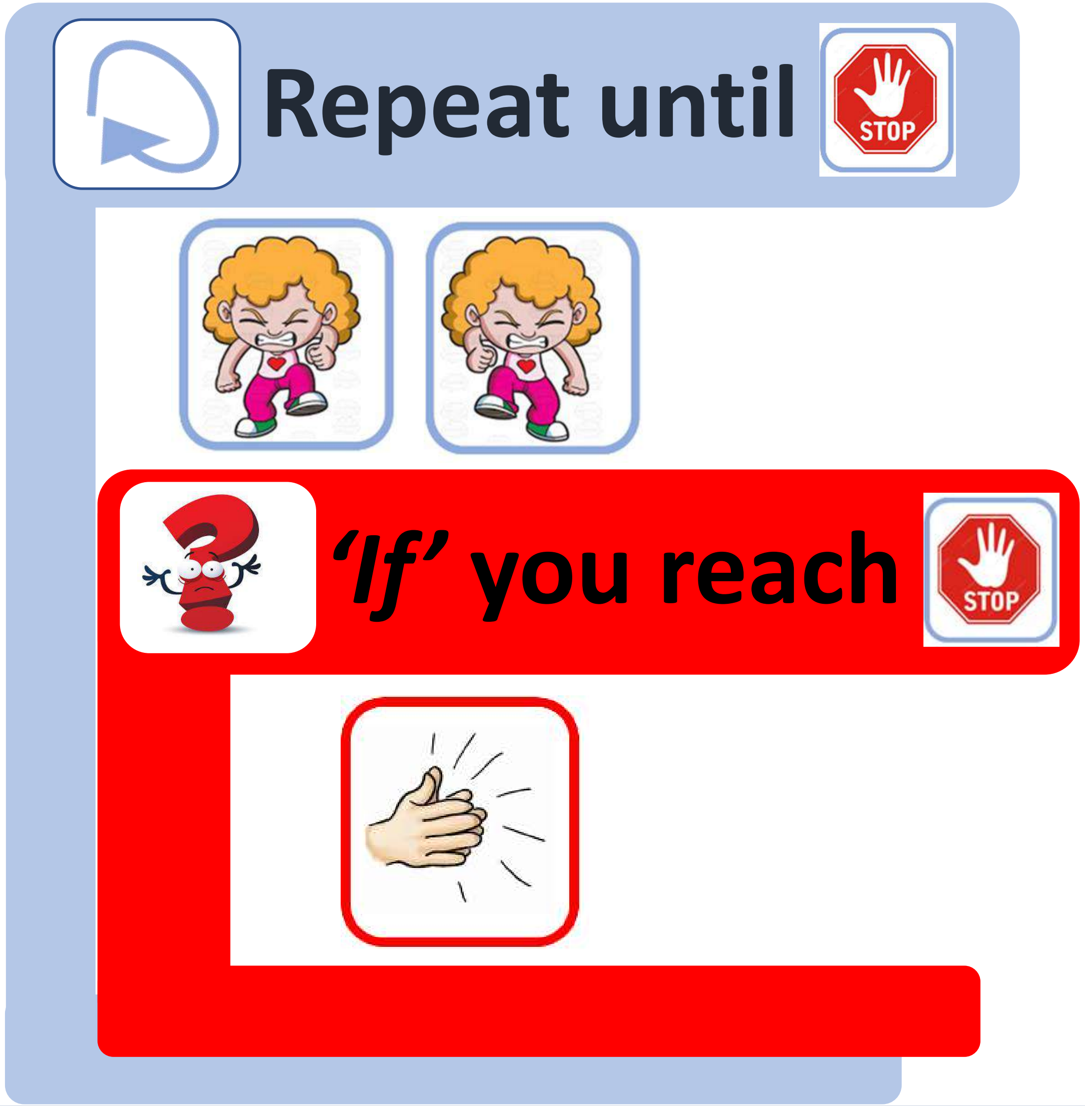
left leg stomp

Coding Terminology

Coding Terminology

Conditionals - execute different instructions using an ‘if’ and ‘result’

Coding Terminology



Repeat until

‘If’ you reach

Coding Terminology

right leg stomp

Coding Terminology

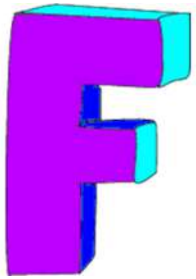
left leg stomp

Coding Terminology

Coding Terminology

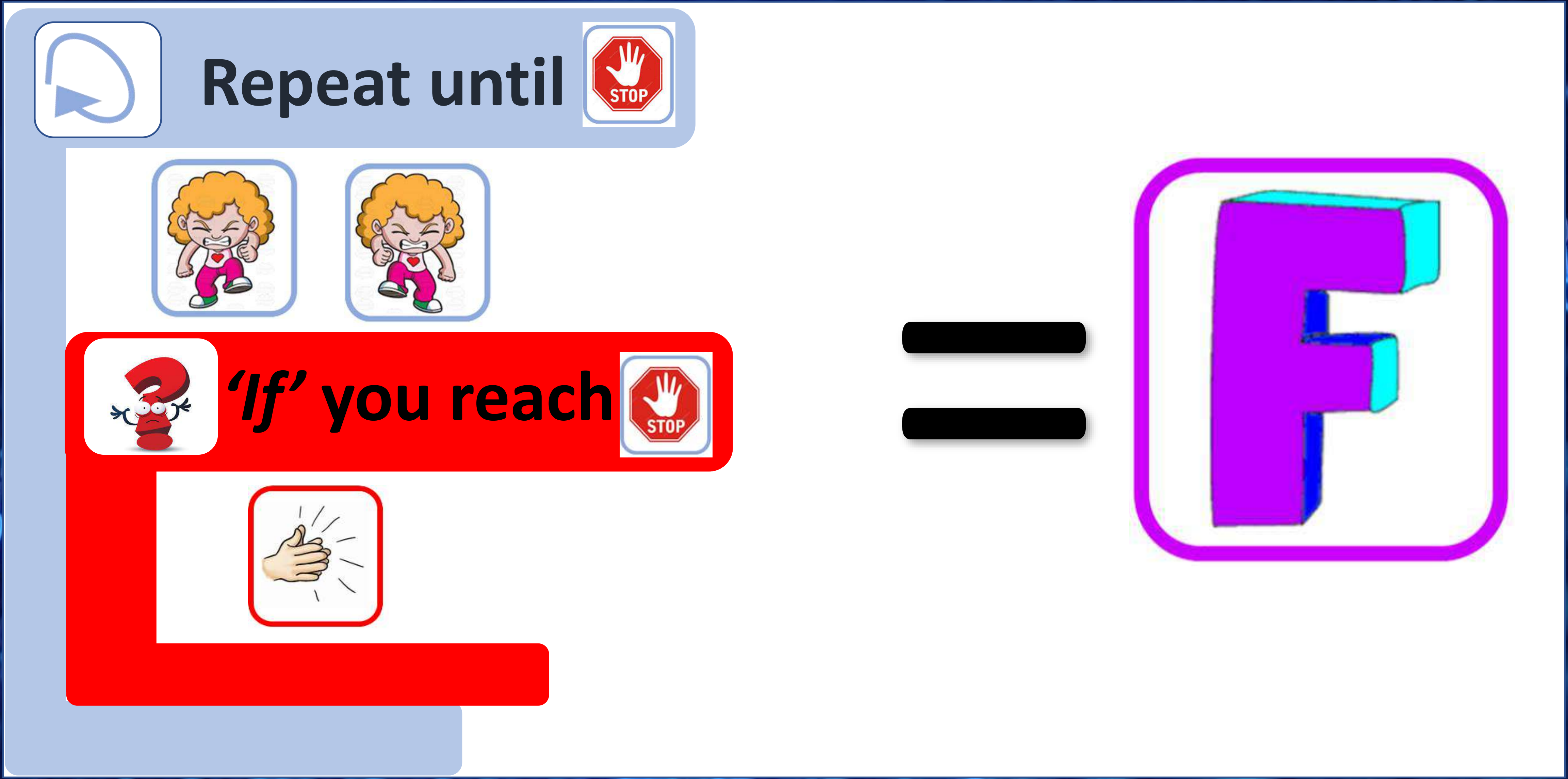
Clap

Coding Terminology



Function – a section in a program which performs a specific task (one coding block can represent multiple instructions)

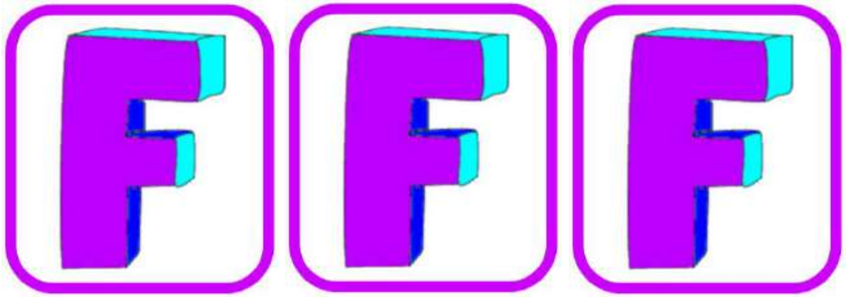
Coding Terminology



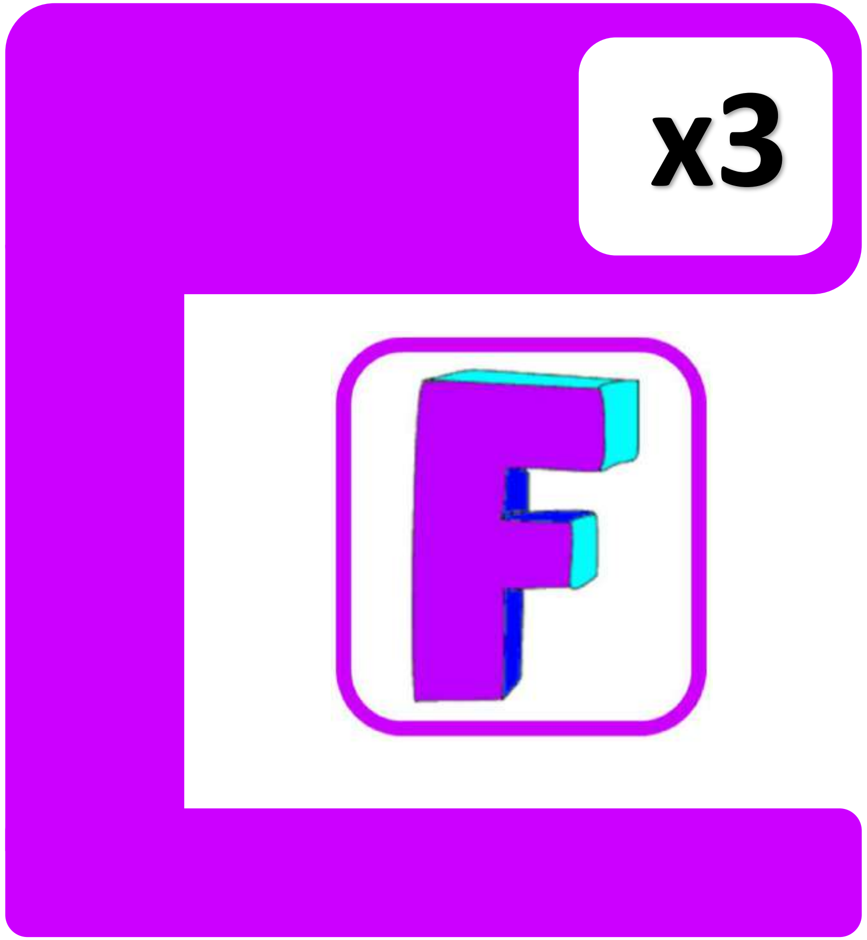
Repeat until

‘If’ you reach

Coding Terminology



or

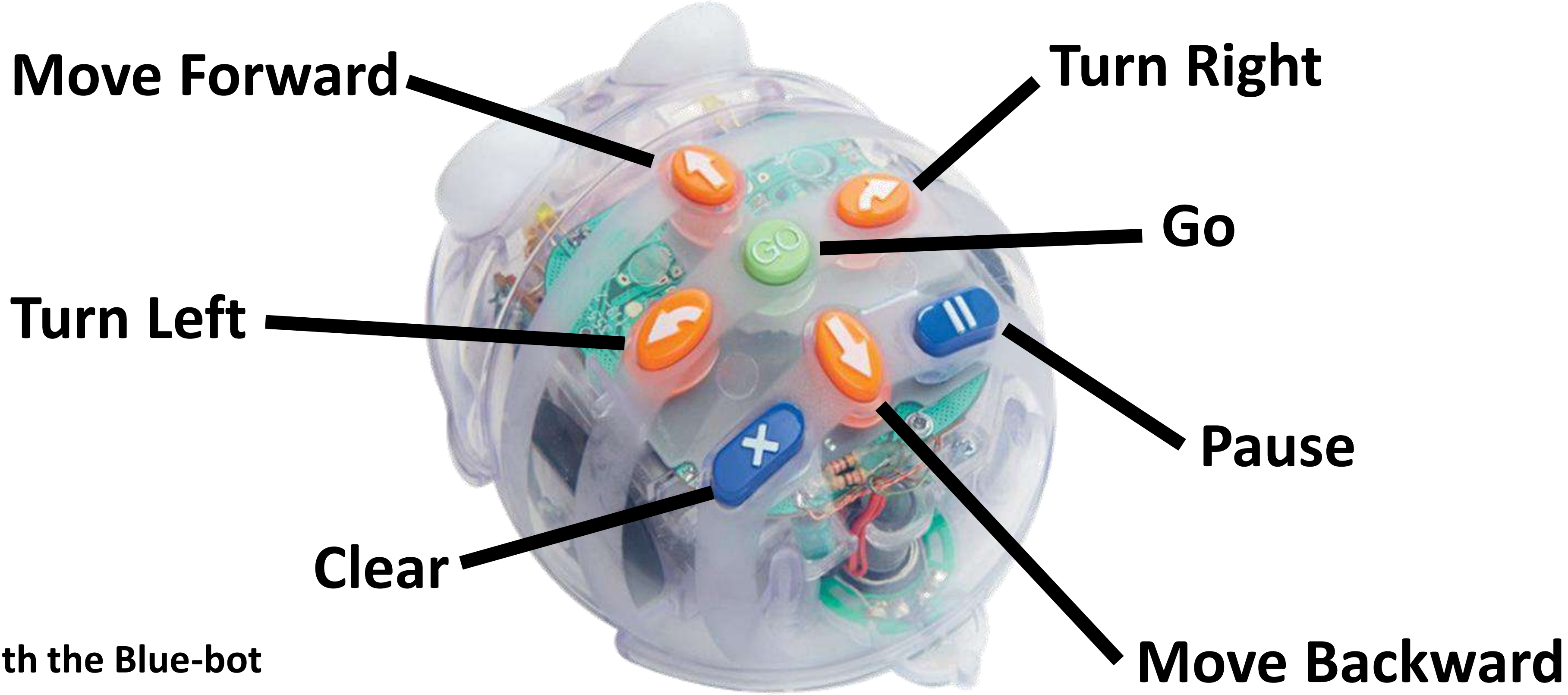


Where to next?

Blue Bot

- 240 commands
- rechargeable
- network capability (Bluetooth)
- two apps compatible with iOS and Android *(Blue Bot and Blue Bot Remote)*
- Blue Bot app allows for 45 degree turns and algorithm edits
- controlled with push buttons, tactile reader or apps

Blue Bot Push Butttons



Move Forward

Turn Right

Go

Turn Left

Pause

Clear

Move Backward

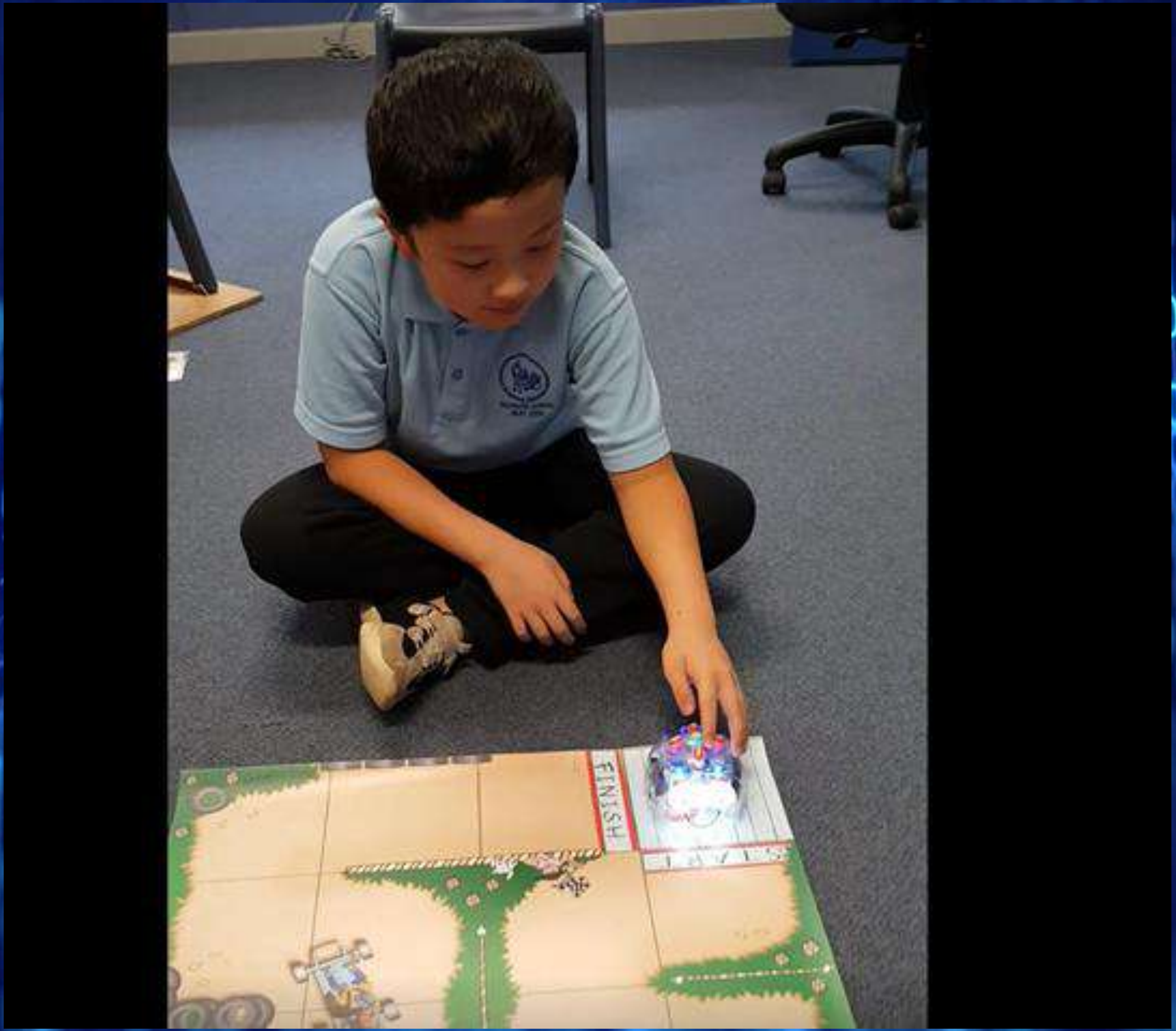
Underneath the Blue-bot

Power – On/Off

Sound – On/Off

https://www.teaching.com.au/product/TTSB485#

VIDEO



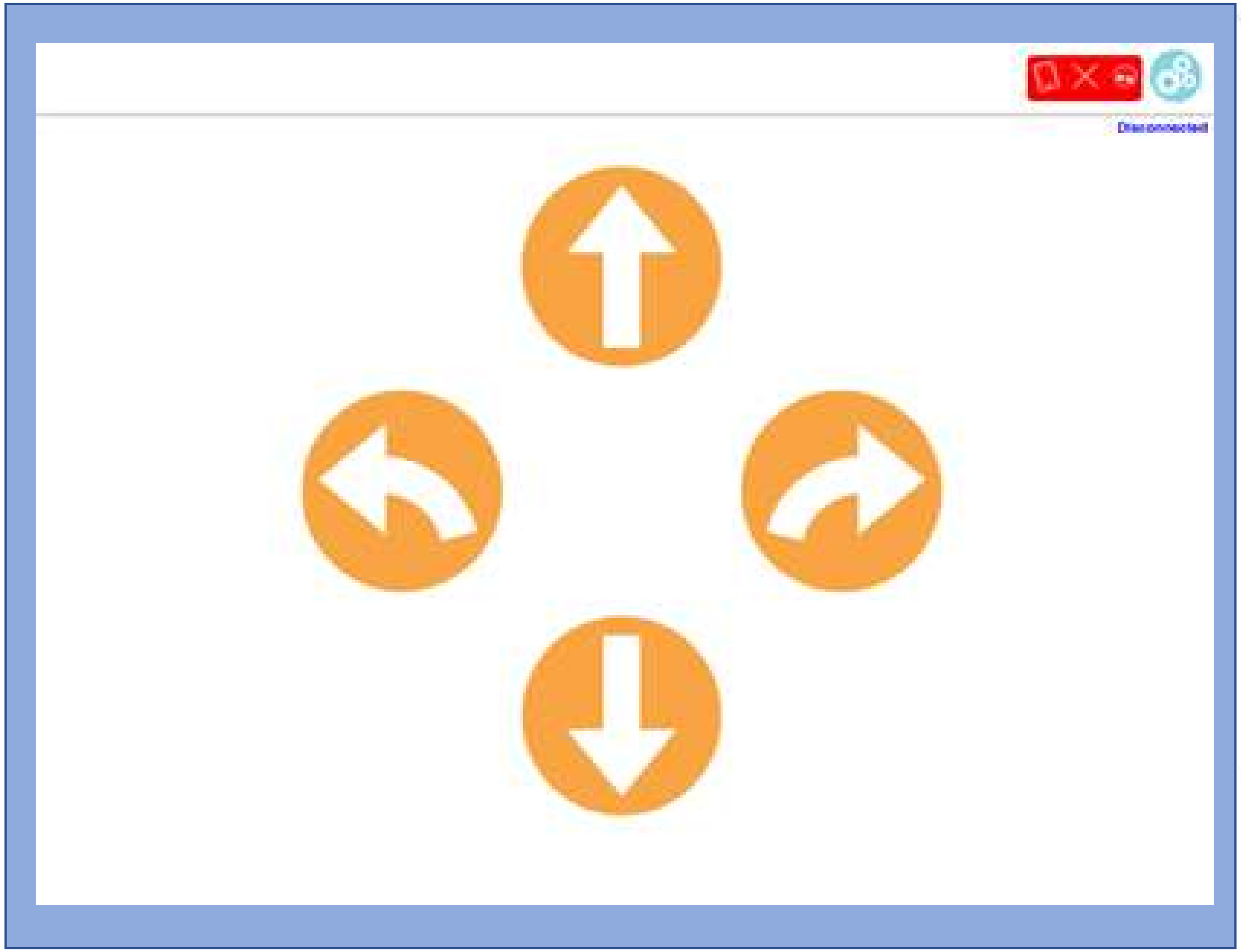
Blue Bot Tactile Reader

https://www.teaching.com.au/product/TTSB485#

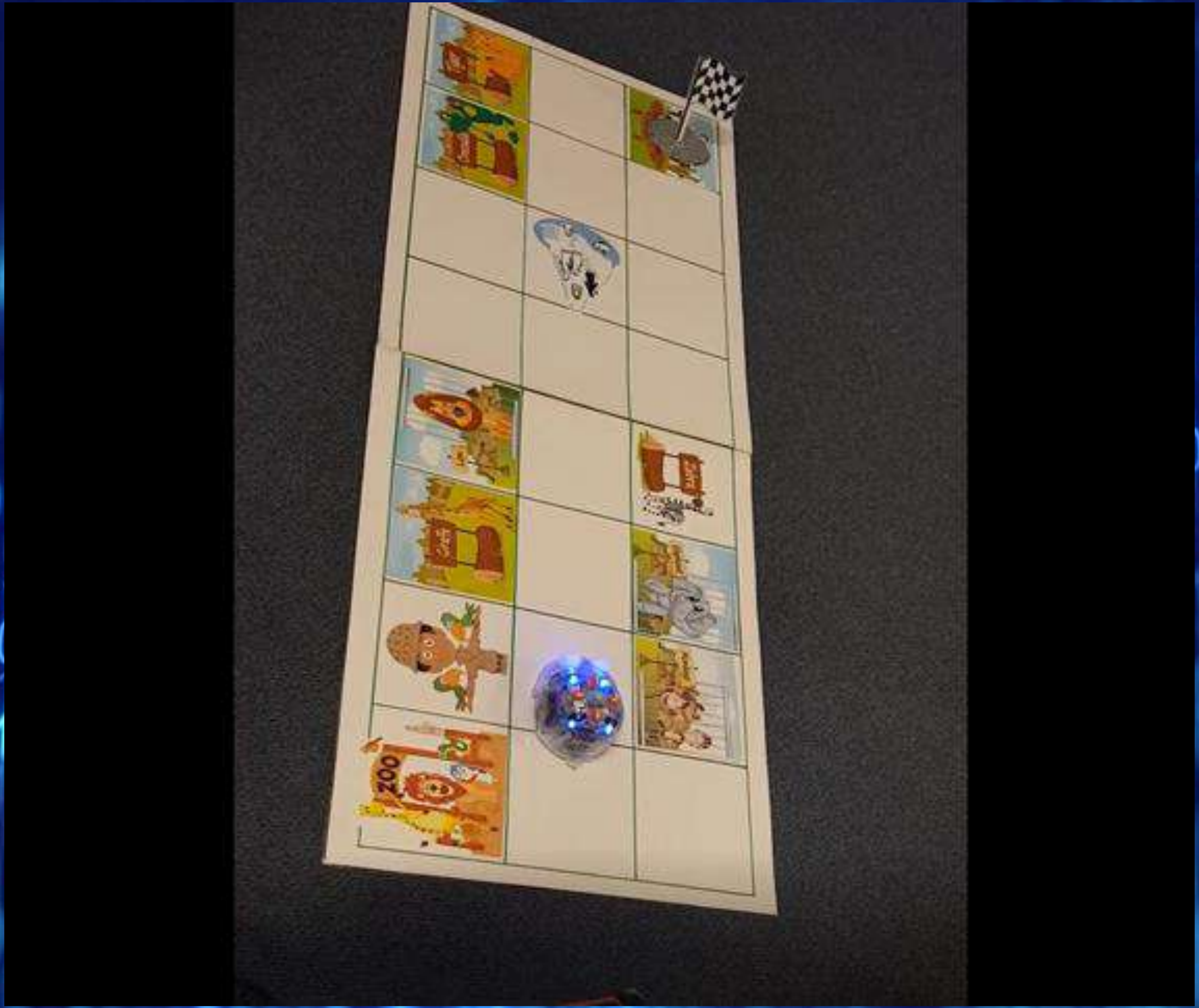
Blue Bot Apps

Remote

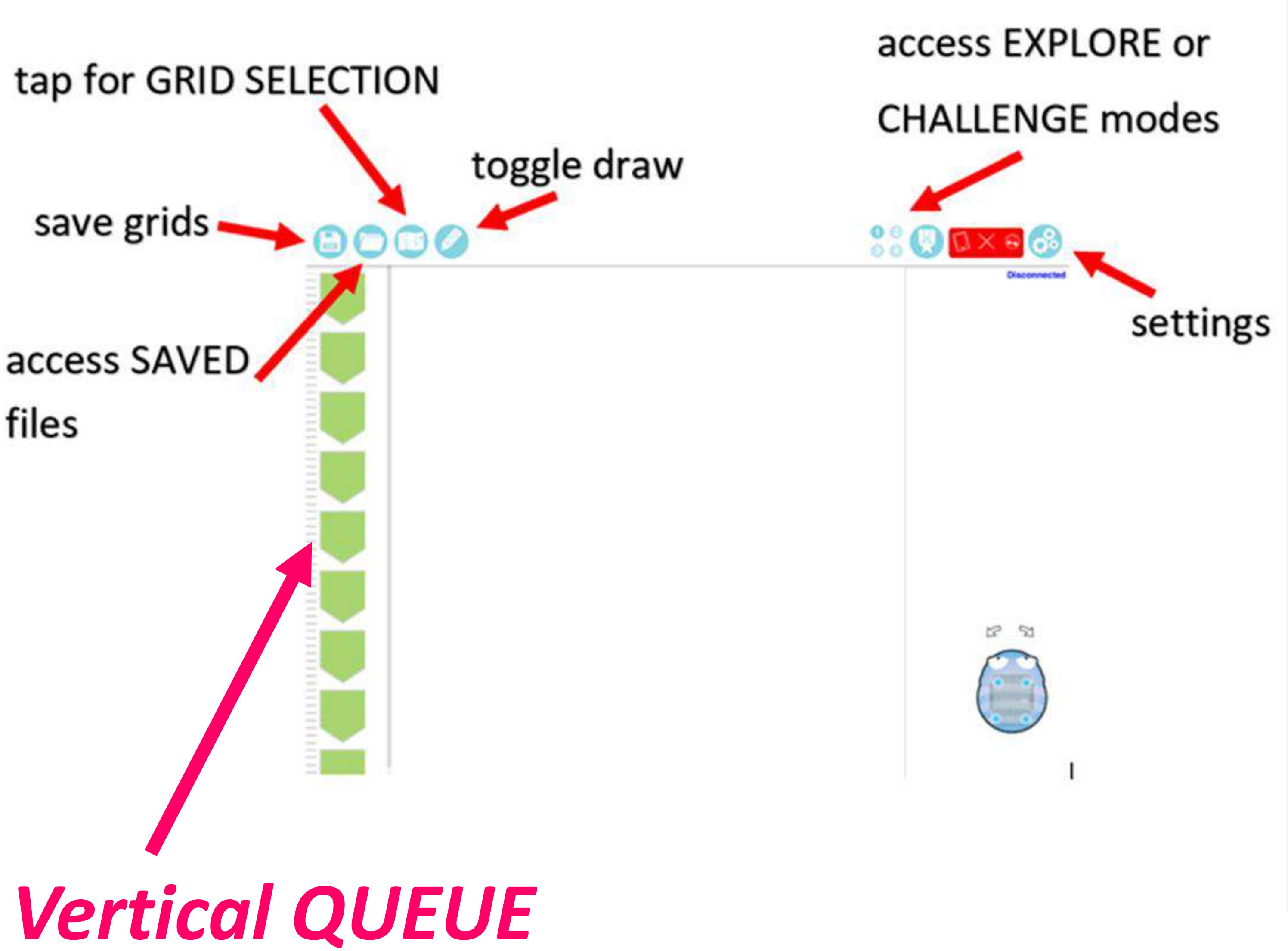
Blue Bot Remote



VIDEO



Blue Bot App



tap for GRID SELECTION

access EXPLORE or CHALLENGE modes

toggle draw

save grids

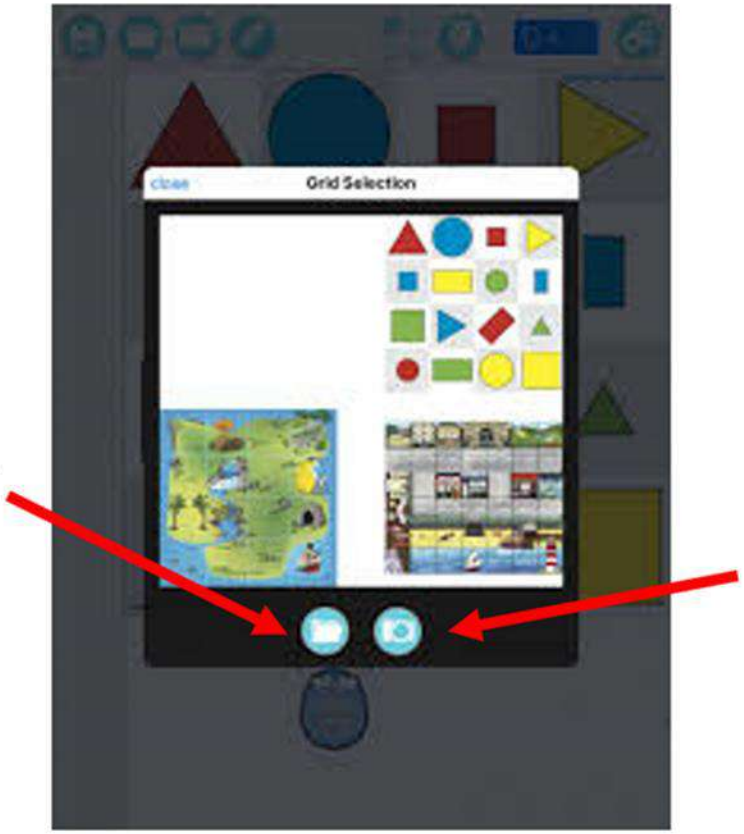
settings

access SAVED files

Vertical QUEUE

Blue Bot App

Choose from PRE EXISTING GRIDS / MAPS.

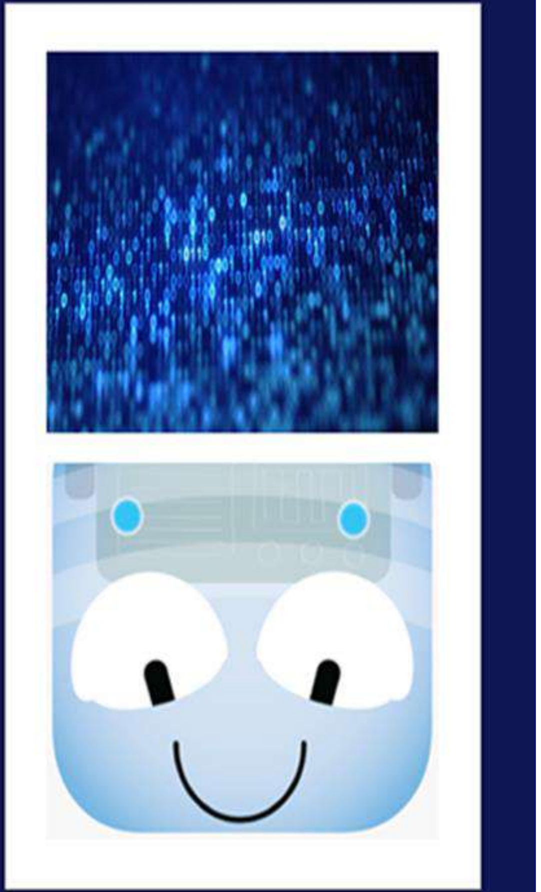


tap for

SAVED FILE

tap for CAMERA

Blue Bot App

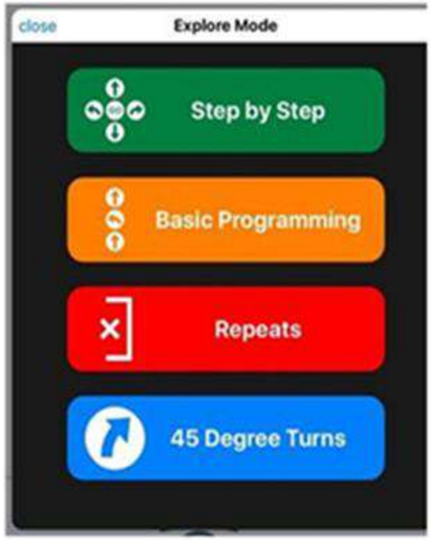


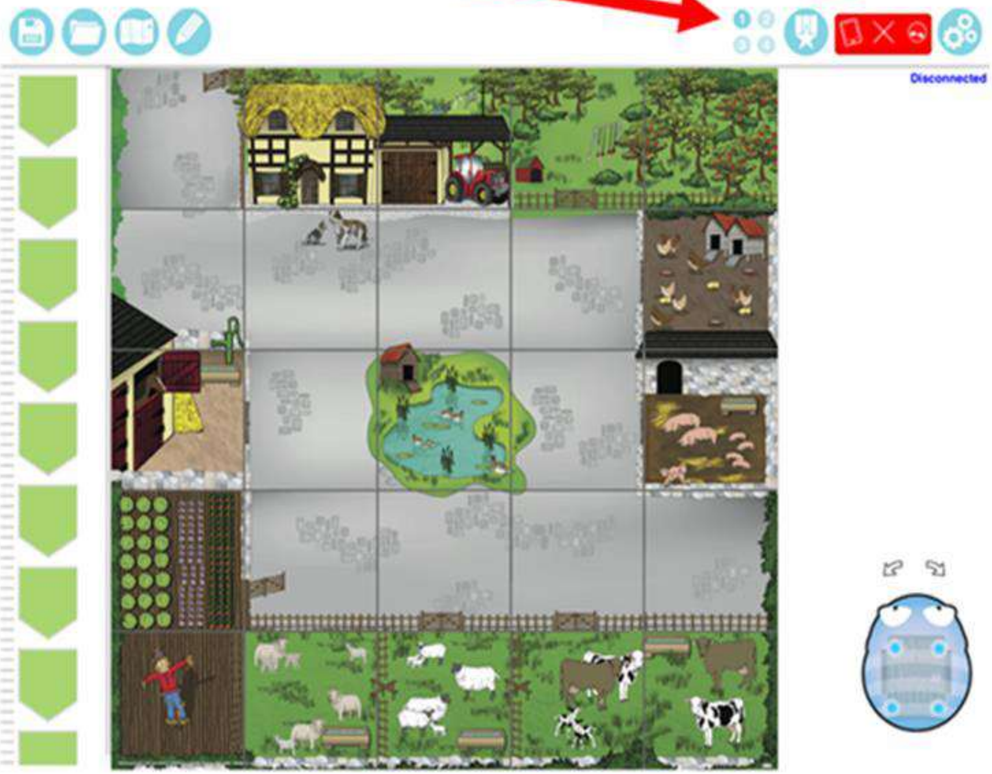
for use **with** Blue Bot -

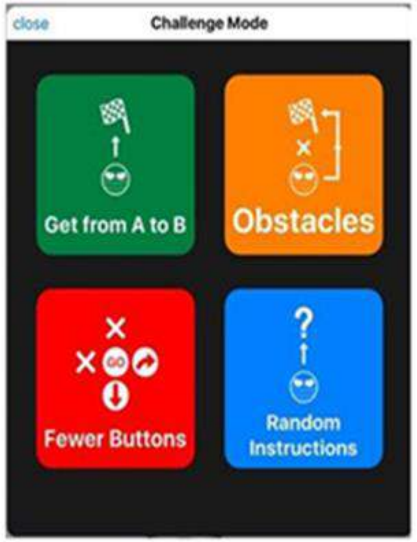
EXPLORE MODE

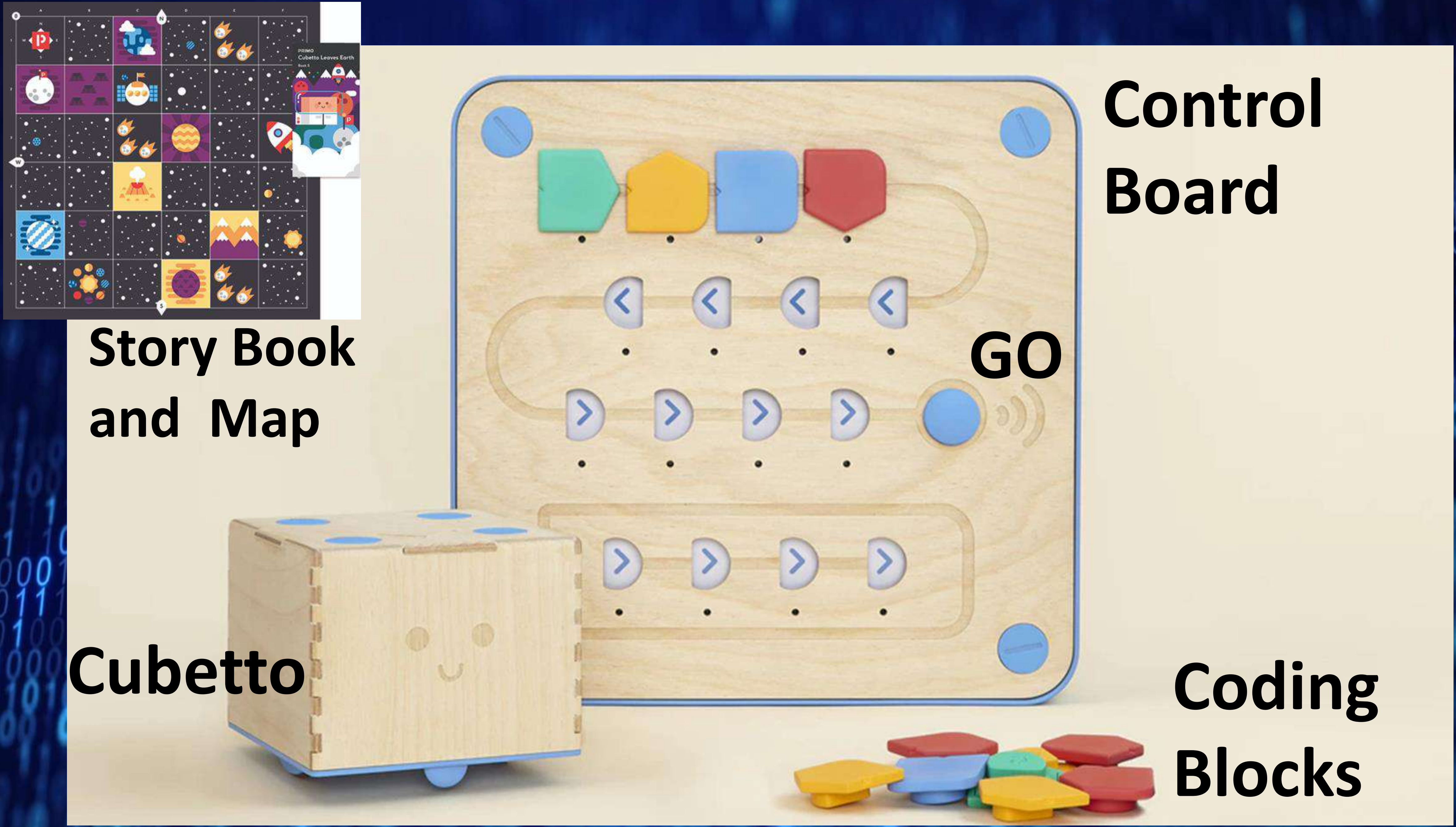
for use **without** Blue Bot -

CHALLENGE MODE









Story Book and Map

Control Board

GO

Cubetto

Coding Blocks

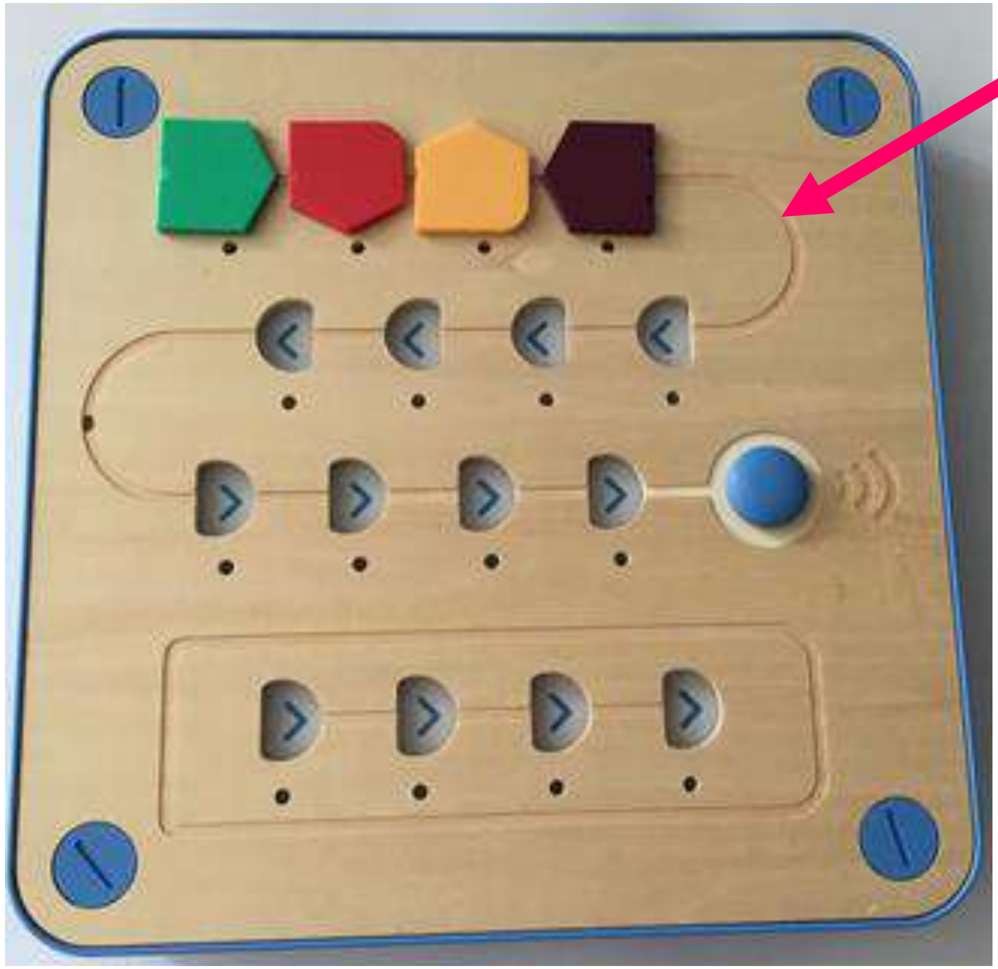
Coding Blocks

GREEN – forward

RED – right

YELLOW – left

PURPLE - backward



Snake-like QUEUE

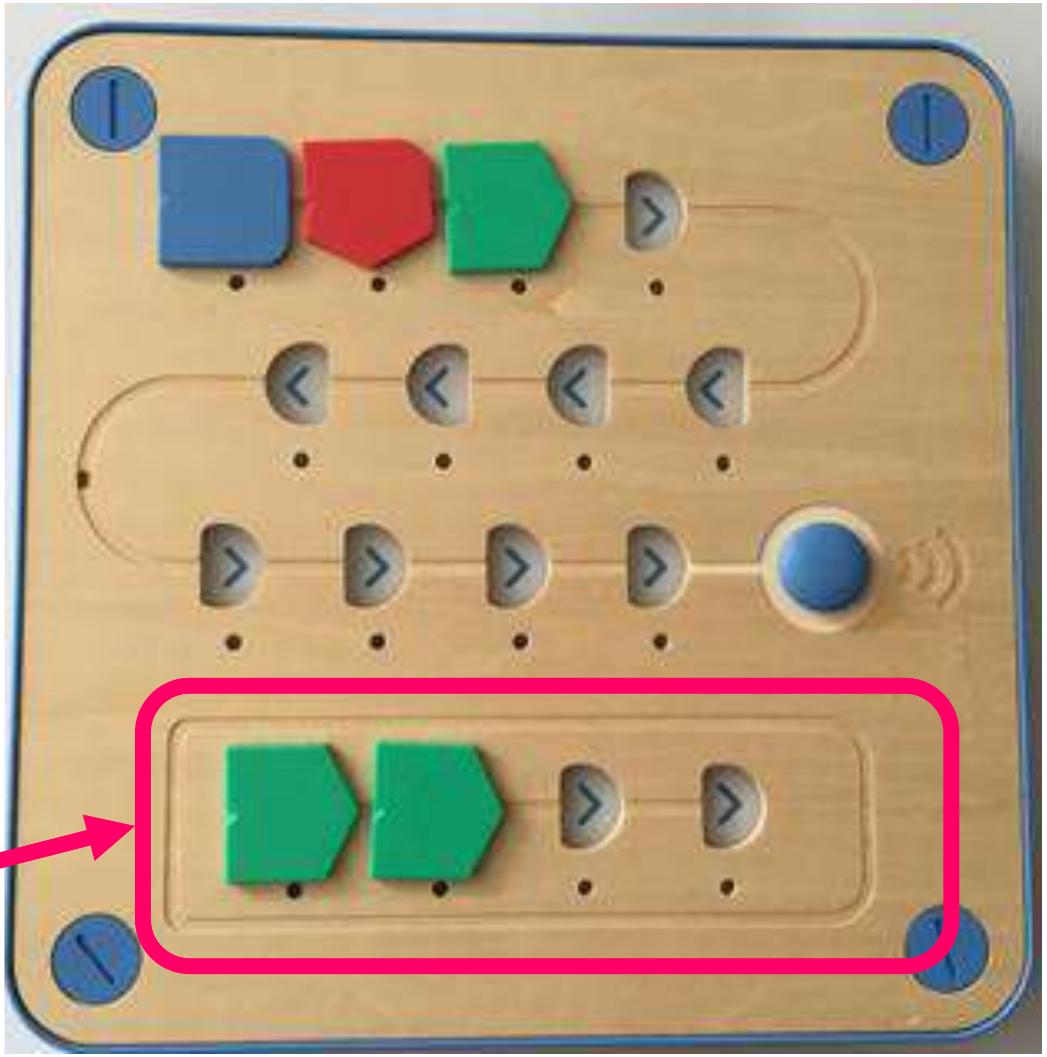
VIDEO

Function Block

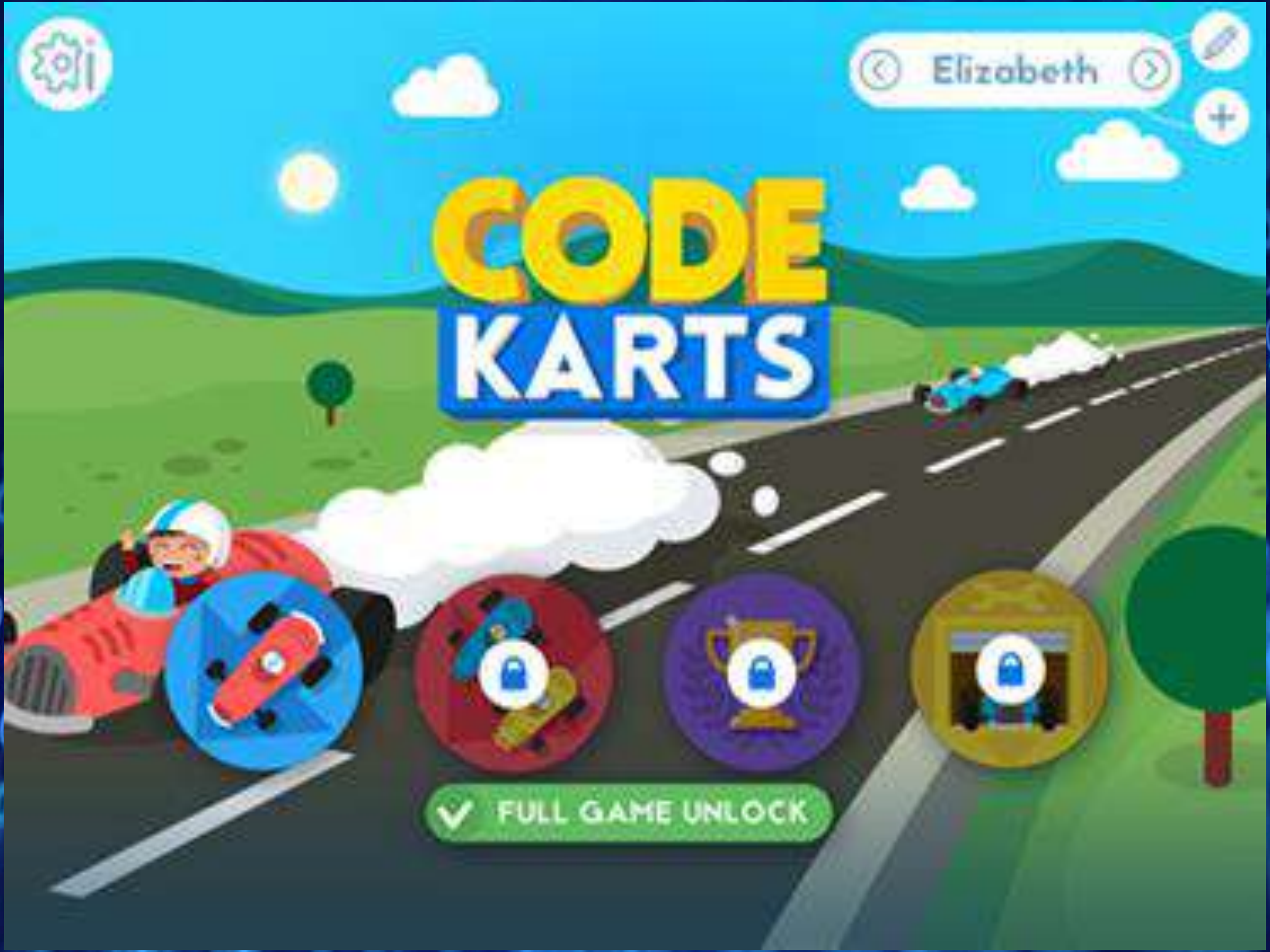
BLUE - function

The **FUNCTION** block allows for **multiple** steps to be executed.

The steps to be executed are decided by the blocks placed on the FUNCTION LINE.



VIDEO


Elizabeth
CODE KARTS
FULL GAME UNLOCK

DRAG and DROP to *Horizontal QUEUE*

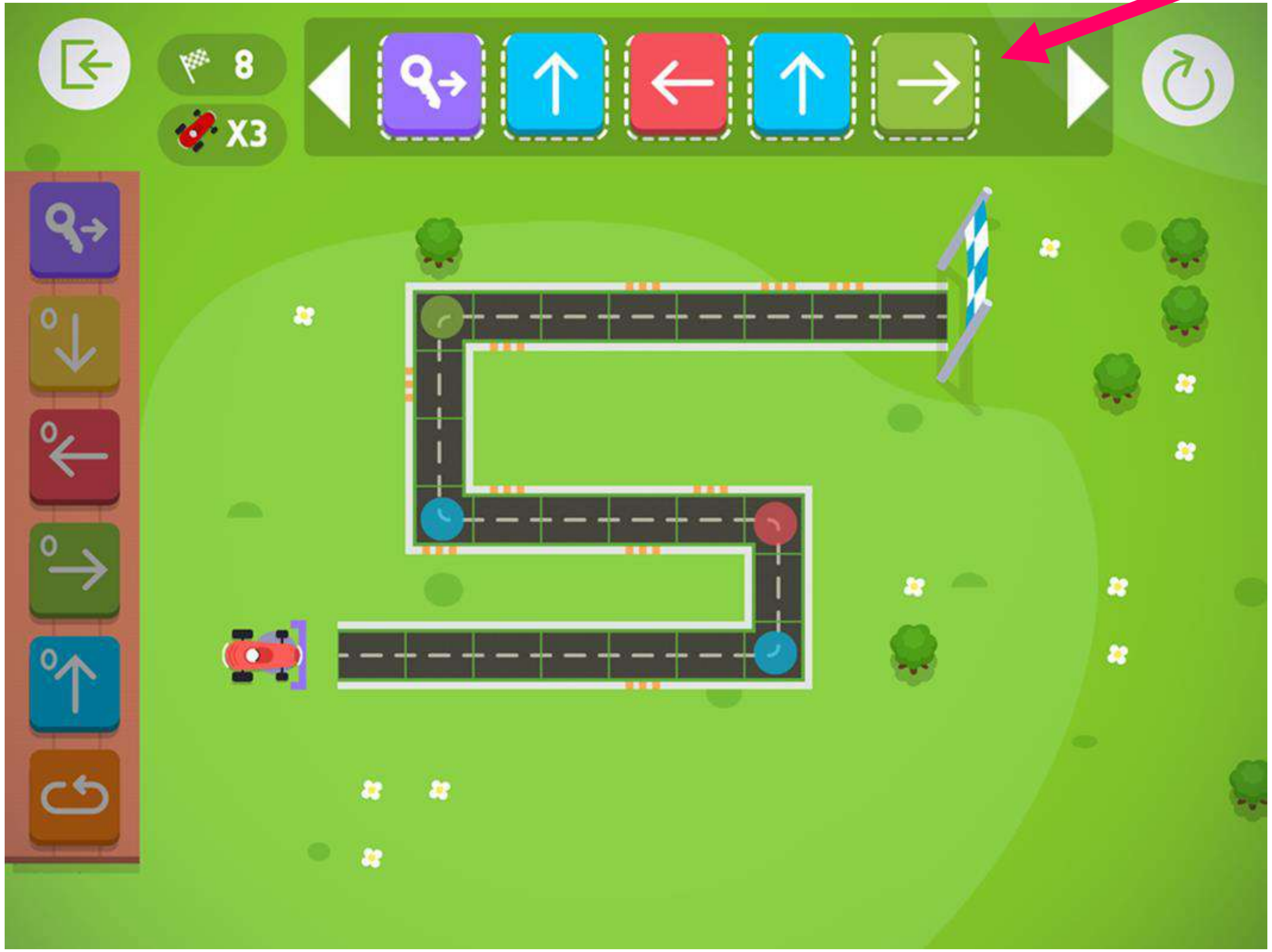
FORWARD

DOWN

LEFT

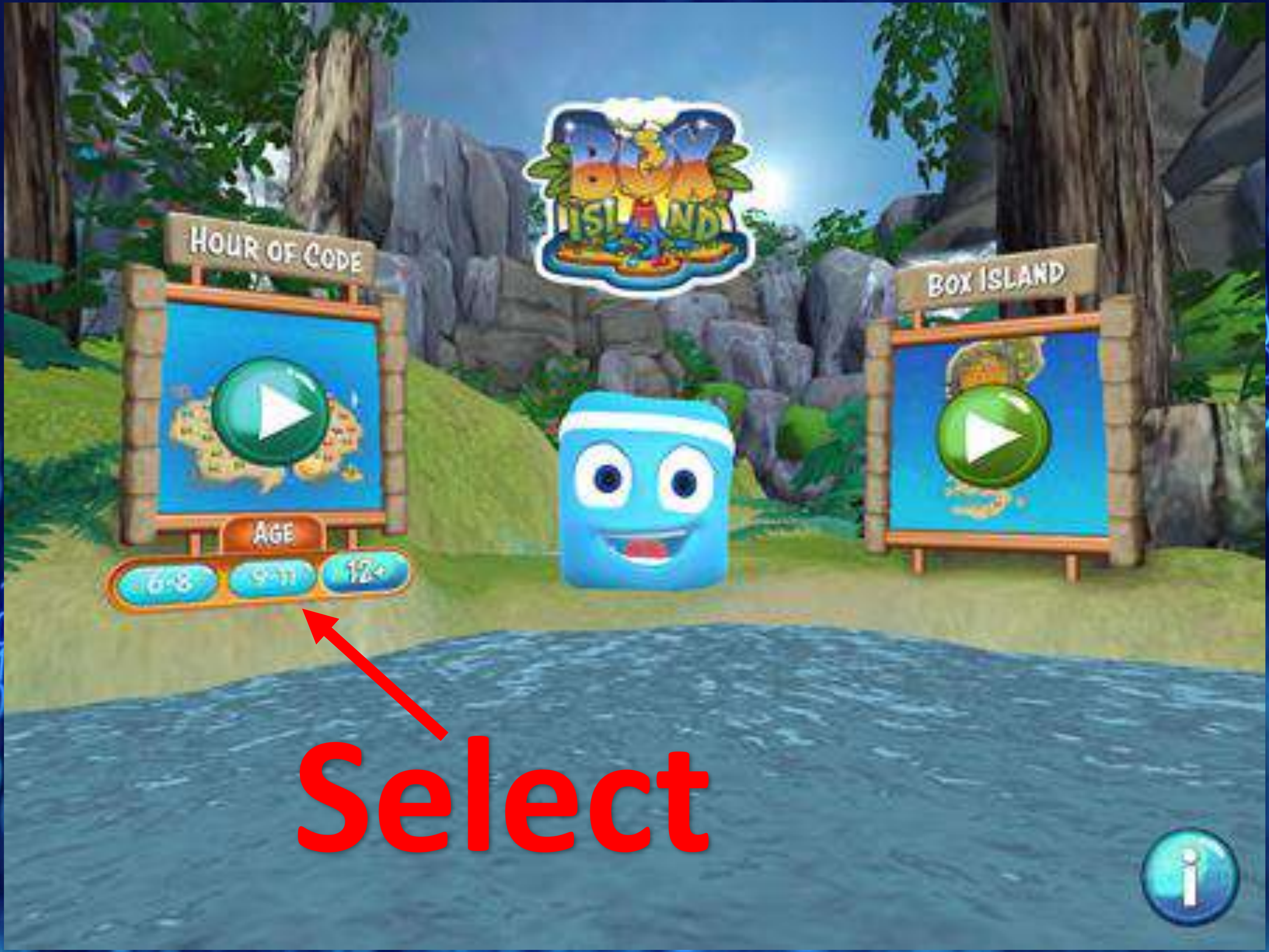
RIGHT

UP



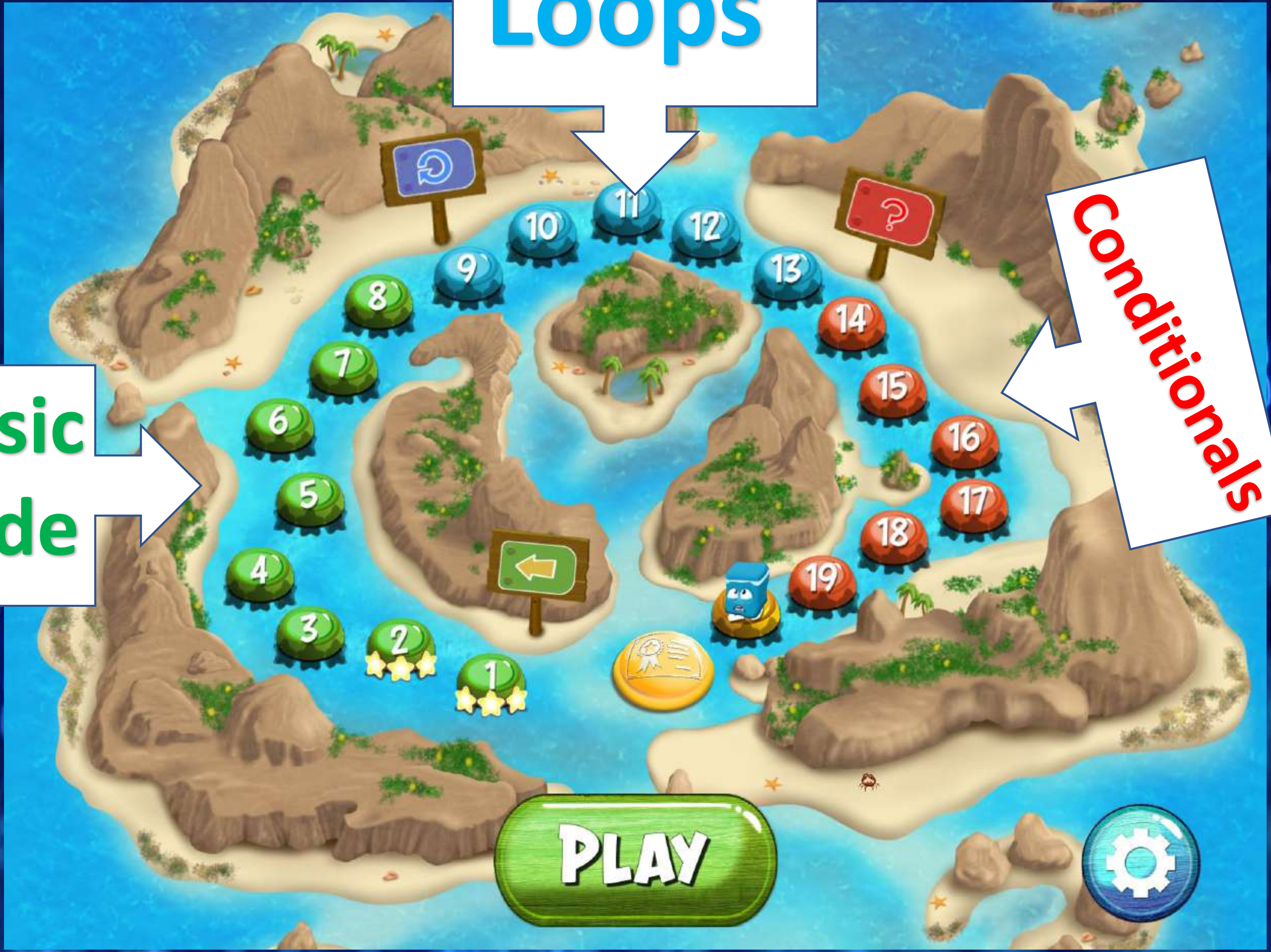
VIDEO

Box Island



Select

Levels



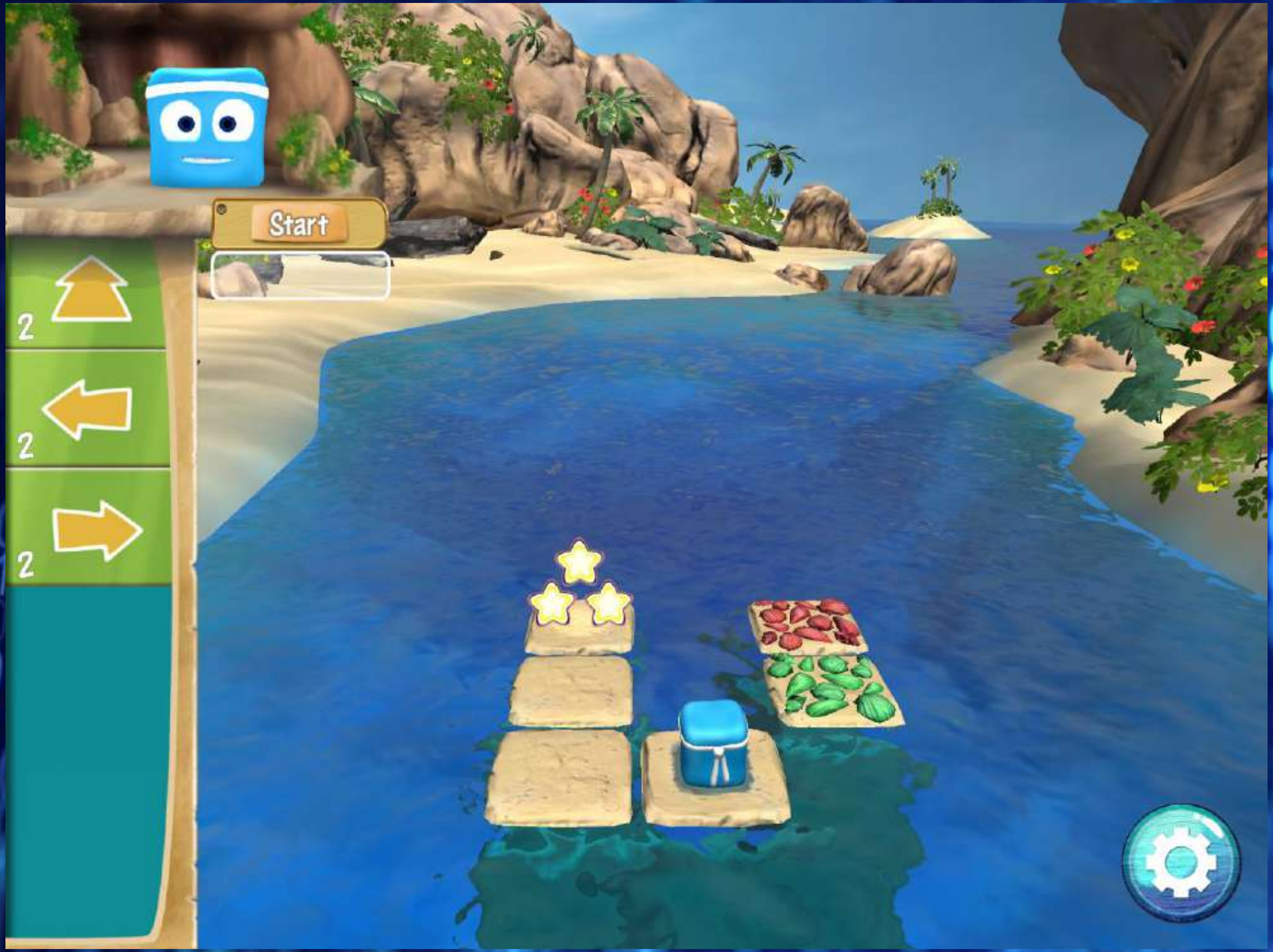
Loops

Basic code

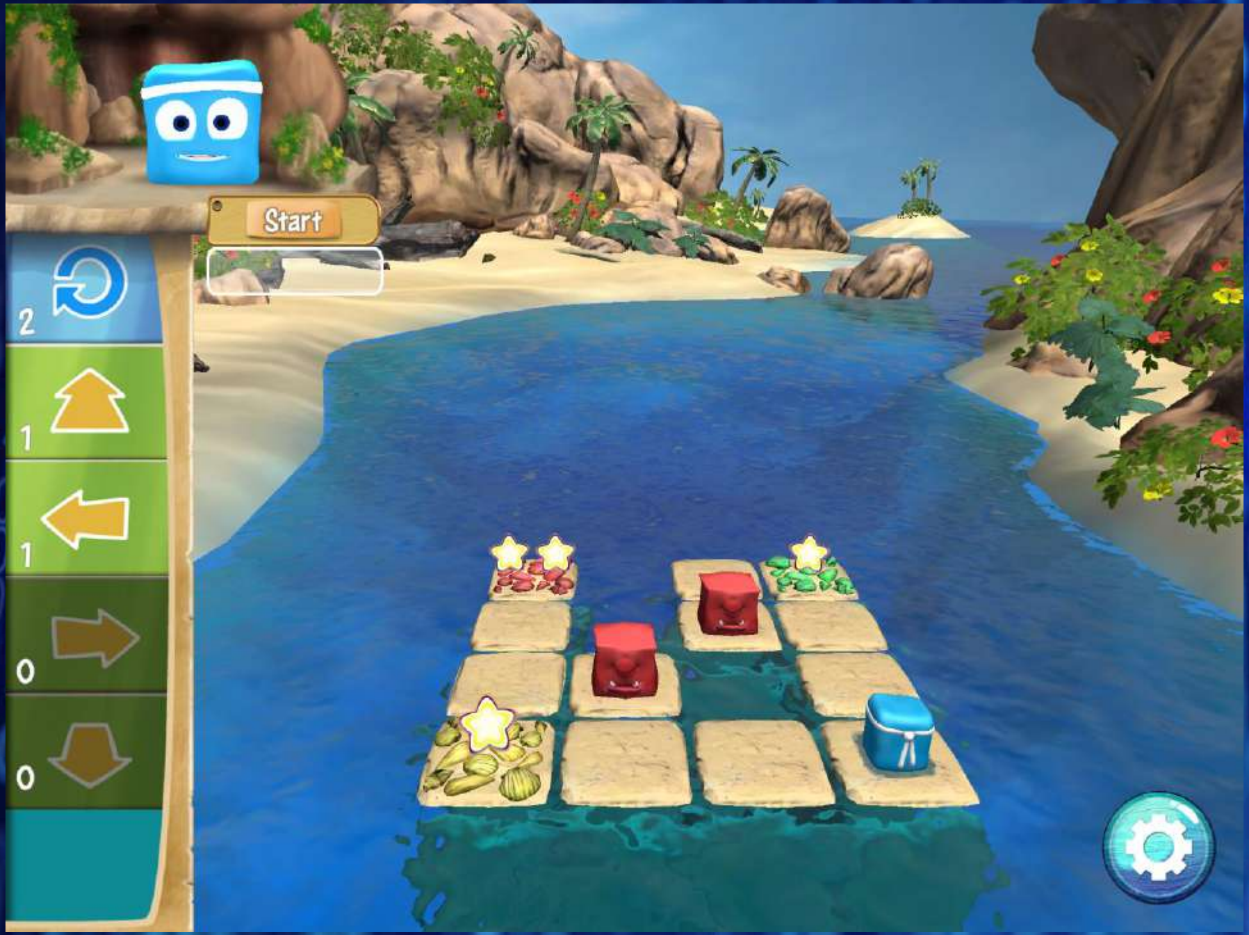
Conditionals

VIDEO

Basic code



Loops



VIDEO

The Australian Curriculum: Digital Technologies (F-10) comprises two related strands:

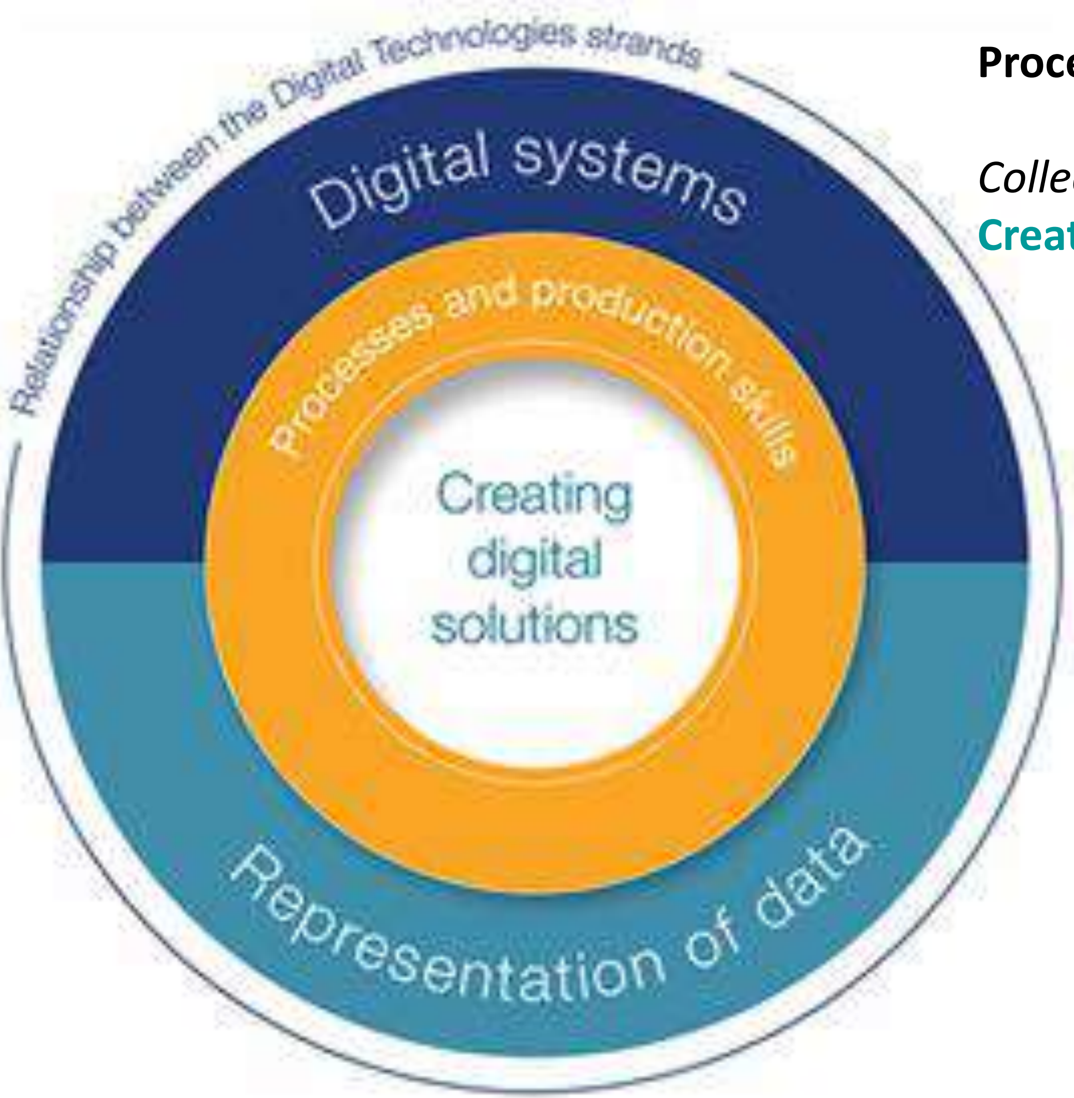
Knowledge and understanding

Digital systems

- the components of digital systems: hardware, software and networks and their use

Representation of data

- how data are represented and structured symbolically


Relationship between the Digital Technologies strands
Digital systems
Processes and production skills
Creating digital solutions
Representation of data

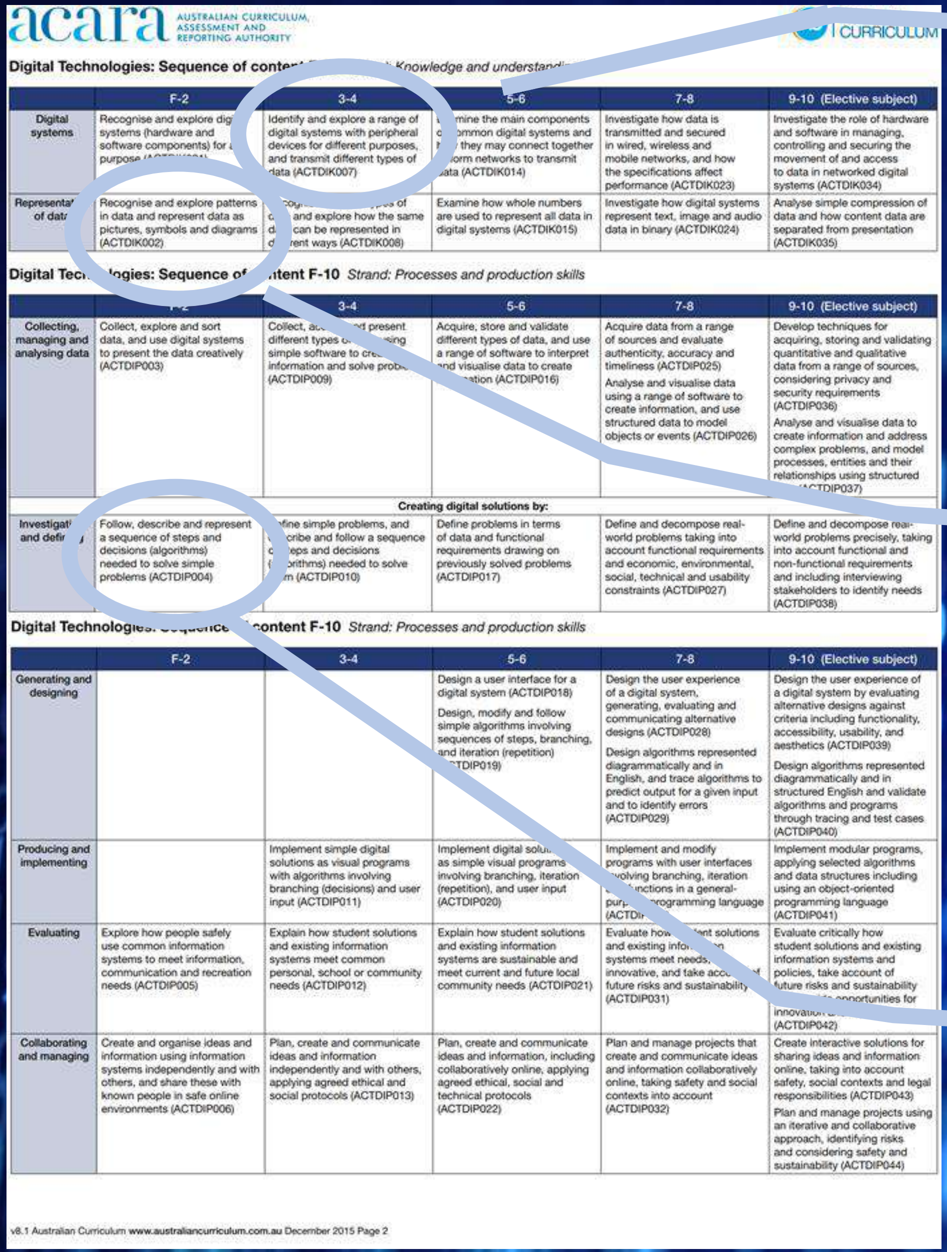
Processes and production skills

Collecting, managing and analysing data

Creating digital solutions by:

- investigating and defining
- generating and designing
- producing and implementing
- evaluating
- collaborating and managing

https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/structure/



acara AUSTRALIAN CURRICULUM, ASSESSMENT AND REPORTING AUTHORITY

CURRICULUM

Digital Technologies: Sequence of content F-10 *Strand: Knowledge and understanding*

	F-2	3-4	5-6	7-8	9-10 (Elective subject)
Digital systems	Recognise and explore digital systems (hardware and software components) for a purpose (ACTDIK001)	Identify and explore a range of digital systems with peripheral devices for different purposes, and transmit different types of data (ACTDIK007)	Examine the main components of common digital systems and how they may connect together to form networks to transmit data (ACTDIK014)	Investigate how data is transmitted and secured in wired, wireless and mobile networks, and how the specifications affect performance (ACTDIK023)	Investigate the role of hardware and software in managing, controlling and securing the movement of and access to data in networked digital systems (ACTDIK034)
Representation of data	Recognise and explore patterns in data and represent data as pictures, symbols and diagrams (ACTDIK002)	Recognise different types of data and explore how the same data can be represented in different ways (ACTDIK008)	Examine how whole numbers are used to represent all data in digital systems (ACTDIK015)	Investigate how digital systems represent text, image and audio data in binary (ACTDIK024)	Analyse simple compression of data and how content data are separated from presentation (ACTDIK035)

Digital Technologies: Sequence of content F-10 *Strand: Processes and production skills*

	F-2	3-4	5-6	7-8	9-10 (Elective subject)
Collecting, managing and analysing data	Collect, explore and sort data, and use digital systems to present the data creatively (ACTDIP003)	Collect, access and present different types of data using simple software to create information and solve problems (ACTDIP009)	Acquire, store and validate different types of data, and use a range of software to interpret and visualise data to create information (ACTDIP016)	Acquire data from a range of sources and evaluate authenticity, accuracy and timeliness (ACTDIP025) Analyse and visualise data using a range of software to create information, and use structured data to model objects or events (ACTDIP026)	Develop techniques for acquiring, storing and validating quantitative and qualitative data from a range of sources, considering privacy and security requirements (ACTDIP036) Analyse and visualise data to create information and address complex problems, and model processes, entities and their relationships using structured data (ACTDIP037)
			Creating digital solutions by:		
Investigating and defining	Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems (ACTDIP004)	Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them (ACTDIP010)	Define problems in terms of data and functional requirements drawing on previously solved problems (ACTDIP017)	Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints (ACTDIP027)	Define and decompose real-world problems precisely, taking into account functional and non-functional requirements and including interviewing stakeholders to identify needs (ACTDIP038)

Digital Technologies: Sequence of content F-10 *Strand: Processes and production skills*

	F-2	3-4	5-6	7-8	9-10 (Elective subject)
Generating and designing			Design a user interface for a digital system (ACTDIP018) Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019)	Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028) Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029)	Design the user experience of a digital system by evaluating alternative designs against criteria including functionality, accessibility, usability, and aesthetics (ACTDIP039) Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases (ACTDIP040)
Producing and implementing		Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input (ACTDIP011)	Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input (ACTDIP020)	Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP030)	Implement modular programs, applying selected algorithms and data structures including using an object-oriented programming language (ACTDIP041)
Evaluating	Explore how people safely use common information systems to meet information, communication and recreation needs (ACTDIP005)	Explain how student solutions and existing information systems meet common personal, school or community needs (ACTDIP012)	Explain how student solutions and existing information systems are sustainable and meet current and future local community needs (ACTDIP021)	Evaluate how student solutions and existing information systems meet needs, are innovative, and take account of future risks and sustainability (ACTDIP031)	Evaluate critically how student solutions and existing information systems and policies, take account of future risks and sustainability and provide opportunities for innovation and enterprise (ACTDIP042)
Collaborating and managing	Create and organise ideas and information using information systems independently and with others, and share these with known people in safe online environments (ACTDIP006)	Plan, create and communicate ideas and information independently and with others, applying agreed ethical and social protocols (ACTDIP013)	Plan, create and communicate ideas and information, including collaboratively online, applying agreed ethical, social and technical protocols (ACTDIP022)	Plan and manage projects that create and communicate ideas and information collaboratively online, taking safety and social contexts into account (ACTDIP032)	Create interactive solutions for sharing ideas and information online, taking into account safety, social contexts and legal responsibilities (ACTDIP043) Plan and manage projects using an iterative and collaborative approach, identifying risks and considering safety and sustainability (ACTDIP044)

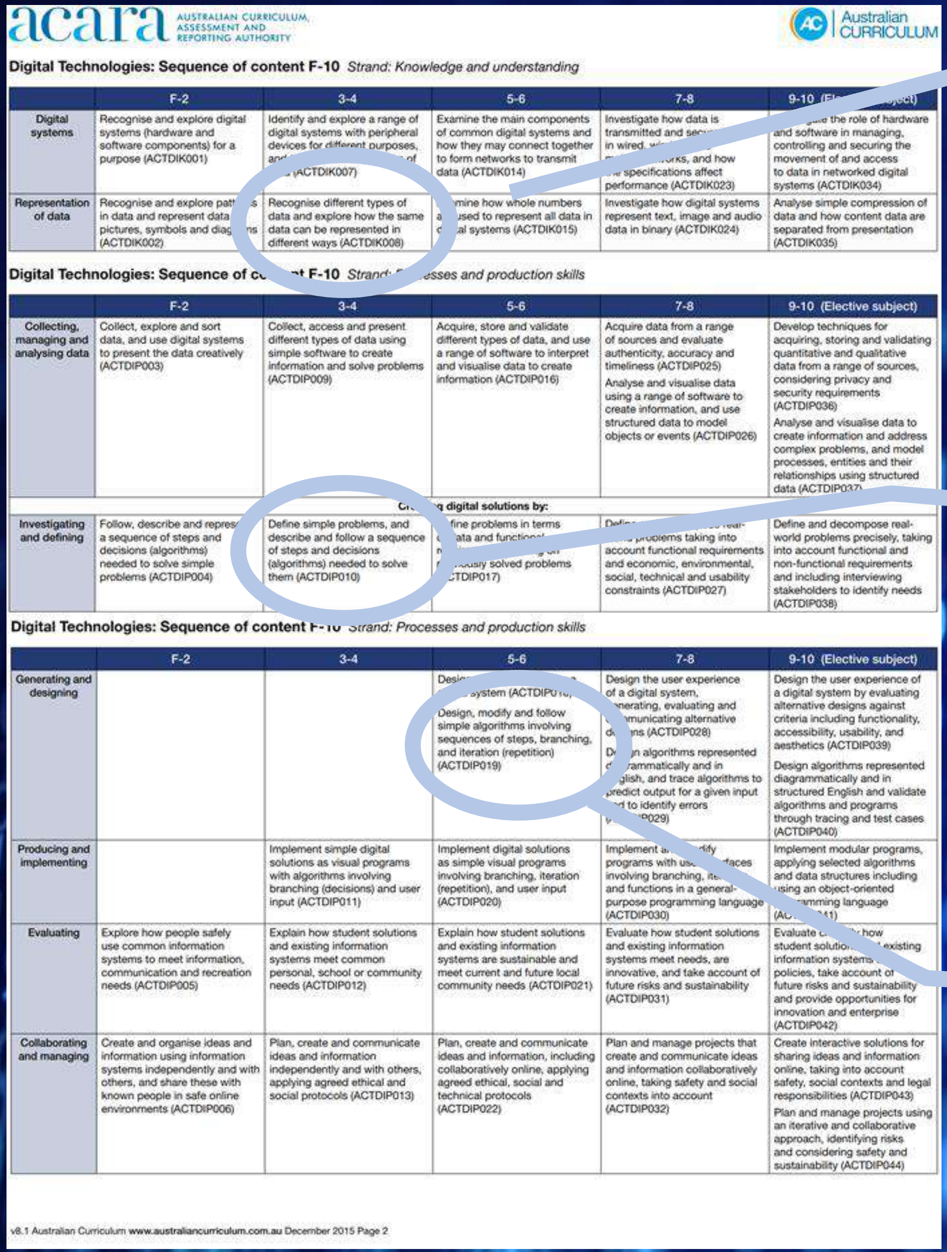
v8.1 Australian Curriculum www.australiancurriculum.com.au December 2015 Page 2

Identify and explore a range of digital systems with peripheral devices for different purposes **(ACTDIK007)**

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

Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems **(ACTDIP004)**

https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/structure/



acara AUSTRALIAN CURRICULUM, ASSESSMENT AND REPORTING AUTHORITY

Australian CURRICULUM

Digital Technologies: Sequence of content F-10 *Strand: Knowledge and understanding*

	F-2	3-4	5-6	7-8	9-10 (El... ...ject)
Digital systems	Recognise and explore digital systems (hardware and software components) for a purpose (ACTDIK001)	Identify and explore a range of digital systems with peripheral devices for different purposes, and ... of ... (ACTDIK007)	Examine the main components of common digital systems and how they may connect together to form networks to transmit data (ACTDIK014)	Investigate how data is transmitted and sec... in wired, wi... ...orks, and how ...e specifications affect performance (ACTDIK023)	...gate the role of hardware and software in managing, controlling and securing the movement of and access to data in networked digital systems (ACTDIK034)
Representation of data	Recognise and explore patt...s in data and represent data pictures, symbols and diag...ns (ACTDIK002)	Recognise different types of data and explore how the same data can be represented in different ways (ACTDIK008)	...mine how whole numbers a... used to represent all data in d...al systems (ACTDIK015)	Investigate how digital systems represent text, image and audio data in binary (ACTDIK024)	Analyse simple compression of data and how content data are separated from presentation (ACTDIK035)

Digital Technologies: Sequence of co... ...nt F-10 *Strand: ...esses and production skills*

	F-2	3-4	5-6	7-8	9-10 (Elective subject)
Collecting, managing and analysing data	Collect, explore and sort data, and use digital systems to present the data creatively (ACTDIP003)	Collect, access and present different types of data using simple software to create information and solve problems (ACTDIP009)	Acquire, store and validate different types of data, and use a range of software to interpret and visualise data to create information (ACTDIP016)	Acquire data from a range of sources and evaluate authenticity, accuracy and timeliness (ACTDIP025) Analyse and visualise data using a range of software to create information, and use structured data to model objects or events (ACTDIP026)	Develop techniques for acquiring, storing and validating quantitative and qualitative data from a range of sources, considering privacy and security requirements (ACTDIP036) Analyse and visualise data to create information and address complex problems, and model processes, entities and their relationships using structured data (ACTDIP037)
			Cr... ...g digital solutions by:		
Investigating and defining	Follow, describe and repres... a sequence of steps and decisions (algorithms) needed to solve simple problems (ACTDIP004)	Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them (ACTDIP010)	...fine problems in terms ... ...ata and functional ... ...usly solved problems ...CTDIP017)	Define ... real- ... problems taking into account functional requirements and economic, environmental, social, technical and usability constraints (ACTDIP027)	Define and decompose real-world problems precisely, taking into account functional and non-functional requirements and including interviewing stakeholders to identify needs (ACTDIP038)

Digital Technologies: Sequence of content F-10 *Strand: Processes and production skills*

	F-2	3-4	5-6	7-8	9-10 (Elective subject)
Generating and designing			Desi... ...system (ACTDIP01...) Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019)	Design the user experience of a digital system, ...nerating, evaluating and ...mmunicating alternative d...ns (ACTDIP028) D...n algorithms represented d...rammatically and in ...glish, and trace algorithms to predict output for a given input ...d to identify errors (...P029)	Design the user experience of a digital system by evaluating alternative designs against criteria including functionality, accessibility, usability, and aesthetics (ACTDIP039) Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases (ACTDIP040)
Producing and implementing		Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input (ACTDIP011)	Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input (ACTDIP020)	Implement a... ...ify programs with us... ...faces involving branching, ite... and functions in a general-purpose programming language (ACTDIP030)	Implement modular programs, applying selected algorithms and data structures including using an object-oriented ...amming language (AC...41)
Evaluating	Explore how people safely use common information systems to meet information, communication and recreation needs (ACTDIP005)	Explain how student solutions and existing information systems meet common personal, school or community needs (ACTDIP012)	Explain how student solutions and existing information systems are sustainable and meet current and future local community needs (ACTDIP021)	Evaluate how student solutions and existing information systems meet needs, are innovative, and take account of future risks and sustainability (ACTDIP031)	Evaluate c... ...y how student solutio... ...existing information systems ... policies, take account of future risks and sustainability and provide opportunities for innovation and enterprise (ACTDIP042)
Collaborating and managing	Create and organise ideas and information using information systems independently and with others, and share these with known people in safe online environments (ACTDIP006)	Plan, create and communicate ideas and information independently and with others, applying agreed ethical and social protocols (ACTDIP013)	Plan, create and communicate ideas and information, including collaboratively online, applying agreed ethical, social and technical protocols (ACTDIP022)	Plan and manage projects that create and communicate ideas and information collaboratively online, taking safety and social contexts into account (ACTDIP032)	Create interactive solutions for sharing ideas and information online, taking into account safety, social contexts and legal responsibilities (ACTDIP043) Plan and manage projects using an iterative and collaborative approach, identifying risks and considering safety and sustainability (ACTDIP044)

v8.1 Australian Curriculum **www.australiancurriculum.com.au** December 2015 Page 2

Recognise different types of data and explore how the same data can be represented in different ways **(ACTDIK008)**

Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them **(ACTDIP010)**

Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) **(ACTDIP019)**

https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/structure/

Challenges

- Keep tasks simple
- Use consistent language
- Adjust timeframe
- Start and finish points
- Position of Bot
- Coding Blocks
- The Queue
- Maps/Grids
- Keeping up with everything
- Accessible to all

VIDEO

Your turn to CODE!

Reference Links.

- https://ase.tufts.edu/DevTech/publications/computersandeducation.pdf
- https://code.org/
- https://education.nsw.gov.au/teaching-and-learning/curriculum/learning-for-the-future/future-focused-learning-and-teaching/project-based-learning-resource-guide/introducing-project-based-learning
- https://education.nsw.gov.au/teaching-and-learning/curriculum/learning-for-the-future/stem
- https://educationstandards.nsw.edu.au/wps/portal/nesa/k-10/understanding-the-curriculum/programming/stem-support
- http://educationstandards.nsw.edu.au/wps/portal/nesa/k-10/learning-areas/technologies/coding-across-the-curriculum
- https://scratch.mit.edu/
- https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/structure/
- https://www.bitdegree.org/tutorials/what-is-coding/
- https://www.learningpotential.gov.au/what-is-coding
- https://www.primotoys.com/
- https://www.primotoys.com/wp-content/uploads/2016/04/Cubetto_teachers_guide.pdf
- https://www.terrapinlogo.com/downloads/file/Getting%20Started%20with%20Blue-Bot%20App.pdf
- https://www.tynker.com/
- https://www.youtube.com/watch?v=S95KiPws54M