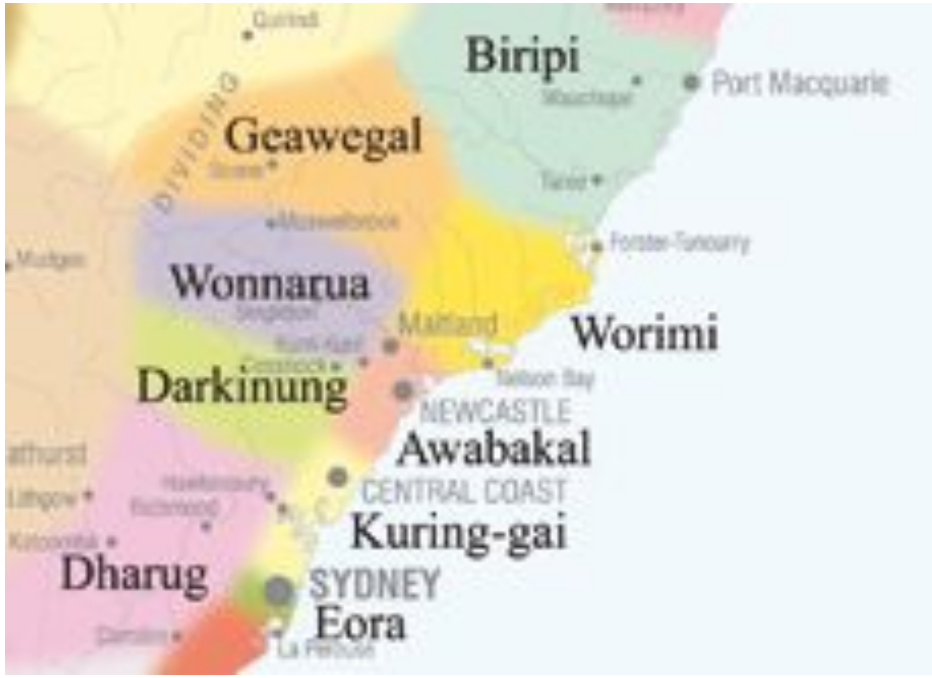
# SCRATCH Maths

Professional development

## Session 1

Introductions

THE UNIVERSITY OF NEWCASTLE AUSTRALIA



# This land

We acknowledge and respect the Pambalong clan of the Awabakal people, traditional custodians of this land.

# Introductions

- I'm Elena
  - I am a mathematician turned computer scientist
  - Since 2012 I have worked at UoN, training teachers-to-be
  - I have been running PD for teachers with Google since 2013

# Introductions

- I’m Jess
  - I’m a Stage 3 Teacher
  - I’m also the IT Coordinator at my school
  - I have completed Scratch PL with Elena and Dan
  - I’m passionate about technology
  - I have run Stage 3 Code Clubs with students
  - I had the pleasure of meeting the Scratch Maths creators, Professor Celia Hoyles and Richard Noss at the beginning of 2020

# Introductions

- I’m Bec
  - I’m also a Stage 3 teacher
  - I’ve only been teaching for a few years but in that time have worked hard to grow my understanding of and application of tech tools in teaching
  - I completed Scratch and Scratch Maths PL with Elena and Dan
  - Richard and Celia, Scratch Maths founders came to visit my class while Jess and I taught the kids Scratch!

# Introductions

- And you?

# Housekeeping

- Toilets
- Fire alarm
- Login

# Workshop Aims

- Introduce ***UCL ScratchMaths***, a program designed to integrate coding and computational thinking into Stage 3 mathematics lessons (or vice-versa!)

- Provide hands-on activities that can be used in the classroom for this integration

- Provide resources for you to explore further

# Workshop Schedule – Day 1

| Time | **Session** | **Location** |
|---|---|---|
| **9:00am – 9:15am** | Registration | CT220 |
| **9:15am – 9:30am** | Workshop overview | CT220 |
| **9:30am – 10:30am** | Combining Mathematics and Coding | CT220 |
| **10:30am – 10:45am** | Morning Tea | CT218 |
| **10:45am – 12pm** | ScratchMaths Module 1 – Investigation 1 | CT220 |
| **12pm – 12:30pm** | Lunch | CT218 |
| **12:30pm – 1:30pm** | ScratchMaths Module 1 – Investigation 2 | CT220 |
| **1:30pm – 2:30pm** | ScratchMaths Module 1 – Investigation 3 | CT220 |
| **2.30pm – 3pm** | Lesson planning | CT220 |

# Workshop Schedule – Day 2

| Time | Session | Location |
|---|---|---|
| **9:00am – 10:30am** | ScratchMaths Module 1 – Investigation 4 | CT220 |
| **10:30am – 10:45am** | Morning Tea | CT218 |
| **10:45am – 12pm** | ScratchMaths Module 3 – Investigation 1 | CT220 |
| **12pm – 12:30pm** | Lunch | CT218 |
| **12:30pm – 1:30pm** | ScratchMaths Module 3 – Investigation 2 | CT220 |
| **1:30pm – 2:30pm** | ScratchMaths Module 3 – Investigation 3 | CT220 |
| **2:30 – 3pm** | Unit programming | CT220 |
| **3pm** | Workshop wrap-up and future contact | CT220 |

# Terms we use in this workshop

- **Computer Science** is a large and diverse field of study, its focus is on problem solving (usually with solutions involving the use of computers)
- **Coding (or Programming)** is the act of writing instructions for a computer in a programming language
- **Computational Thinking** is a way of approaching problems – *"thinking like a Computer Scientist"*[1]

# Computational Thinking in K – 12?

- Should every student become a Computer Scientist or Software Engineer?
- By 2020 half of all STEM jobs will be in computing[8]
- Automation and "innovation" are creating and changing current careers
- Are there any benefits other than preparing students for their careers?

# Coding

- Encouraging students to learn how to code has become a global movement
  - Hour of Code[2]
  - Code Club[3]
- The Digital Technologies subject in the National Curriculum includes programming and algorithms[4]

# Computational Thinking

- “Computer science is **no more about computers than astronomy is about telescopes**, biology is about microscopes or chemistry is about beakers and test tubes. Science is not about tools, it is about how we use them and what we find out when we do”[5]

# Computational Thinking

- A way of approaching a problem in a way that a computer can be used to solve it
- Involves breaking a problem into a step-by-step solution (an algorithm)

# Computational Thinking

- According Brennan and Resnick[7] – the creators of Scratch – there are 3 key dimensions of computational thinking :
  - computational concepts (such as loops, parallelism, etc.),

# Computational Thinking

- According Brennan and Resnick[7] – the creators of Scratch – there are  3 key dimensions of computational thinking :
  - computational concepts (such as loops, parallelism, etc.),
  - computational practices (such as debugging projects or remixing others' work),
  - and computational perspectives

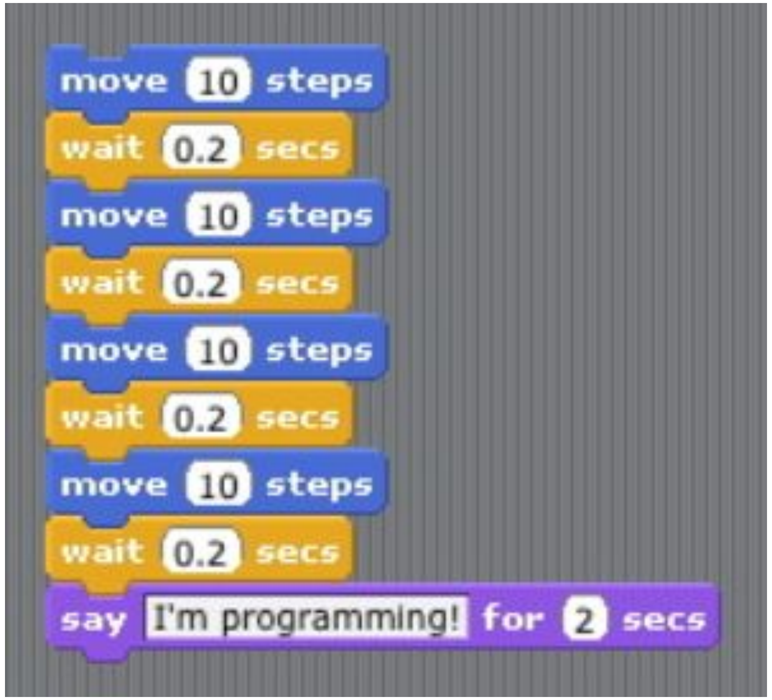
# Computational Concepts

- **Sequences**

“a particular activity or task is expressed as a series of individual steps or instructions that can be executed by the computer. Like a recipe, a sequence of programming instructions specifies the behavior or action that should be produced”

# Computational Concepts

- **Sequences**



```
move 10 steps
wait 0.2 secs
move 10 steps
wait 0.2 secs
move 10 steps
wait 0.2 secs
move 10 steps
wait 0.2 secs
say I'm programming! for 2 secs
```

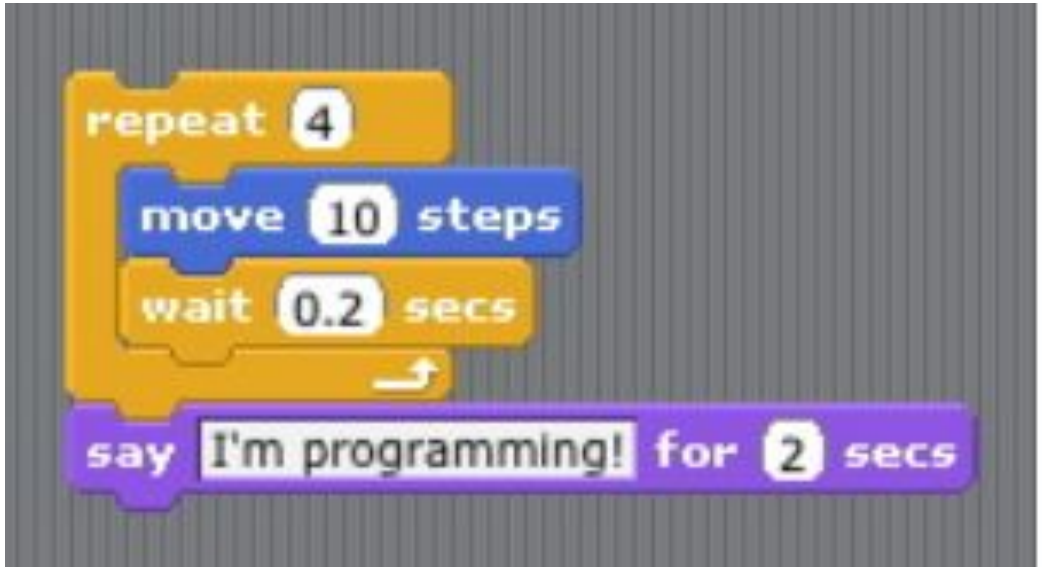
# Computational Concepts

- **Loops**

“But what if we wanted the cat to move and wait 50 or 100 or 1000 more times? Loops are a mechanism for running the same sequence multiple times”

# Computational Concepts

- **Loops**



```
repeat 4
    move 10 steps
    wait 0.2 secs
say I'm programming! for 2 secs
```

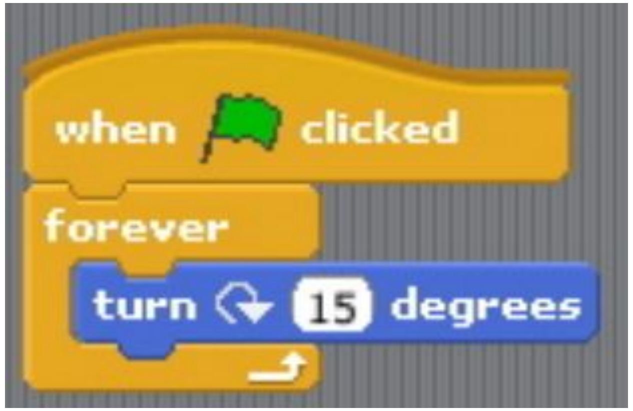
# Computational Concepts

- **Events**

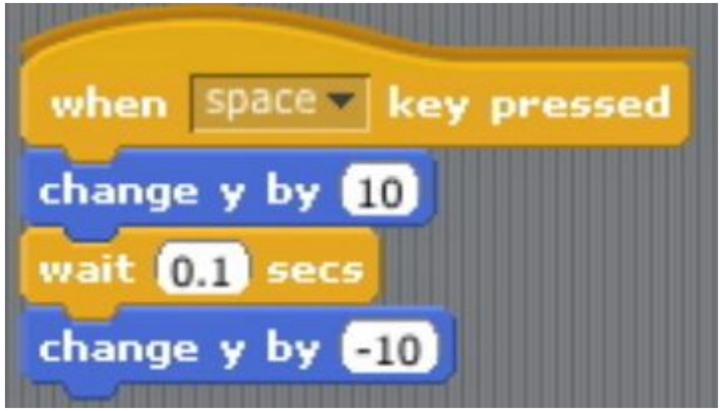
> "Events – one thing causing another thing to happen – are an essential component of interactive media"

# Computational Concepts

- **Events**



```
when ⚑ clicked
forever
    turn ↻ 15 degrees
```



```
when space ▾ key pressed
change y by 10
wait 0.1 secs
change y by -10
```

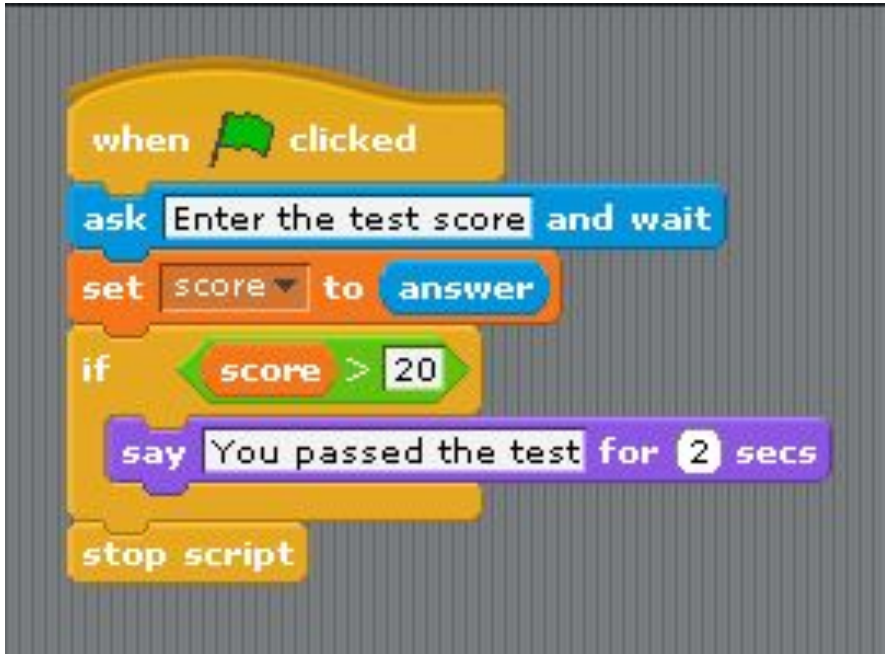
# Computational Concepts

- **Conditionals**

"the ability to make decisions based on certain conditions, which supports the expression of multiple outcomes."

# Computational Concepts

- **Conditionals**



```
when ⚑ clicked
ask Enter the test score and wait
set score to answer
if score > 20
    say You passed the test for 2 secs
stop script
```

# How much do you know about it?

- Could you do a brief anonymous survey?
- That was we can see at the end how much you've learned!

https://www.surveymonkey.com/r/ScratchMaths2020-pre

# Resources

- Web Sites
  - Computer Science 4 Fun: http://www.cs4fn.org/
  - Computer Science Field Guide: http://csfieldguide.org.nz/

- Online Courses
  - CSER (Uni of Adelaide) Digital Technologies MOOCS: https://csdigitaltech.appspot.com/course
  - Google's Exploring Computational Thinking Course: https://www.google.com/edu/resources/programs/exploring-computational-thinking/

# References

1. **“Computational Thinking Benefits Society” - Jeannette M. Wing**

http://socialissues.cs.toronto.edu/index.html%3Fp=279.html

2. **Hour of Code -** https://code.org/learn

3. **Code Club Australia -** http://www.codeclubau.org/

4. **“Digital Technologies Curriculum” – ACARA**

http://www.australiancurriculum.edu.au/technologies/digital-technologies/curriculum/f-10?layout=1

5. **Michael R. Fellows, Ian Parberry (1993) "SIGACT trying to get children excited about CS". in: Computing Research News. January 1993.**

6. **Computational Thinking – Barefoot CAS**

http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/

7. **New frameworks for studying and assessing the development of computational thinking**

http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

8. **Rebooting the Pathway to Success – ACM**

http://pathways.acm.org/