Scratch Professional Development Day 1

Designed and led by

Phil Bagge

Educator & Computing Inspector HIAS

Why Teach Programming?

Session One

Phil Bagge @baggiepr

Digital Devices

Work with a neighbour

You have 90 seconds to list all the digital devices you have in your home

A digital device is a piece of physical equipment that uses digital data *(can be programmed)*

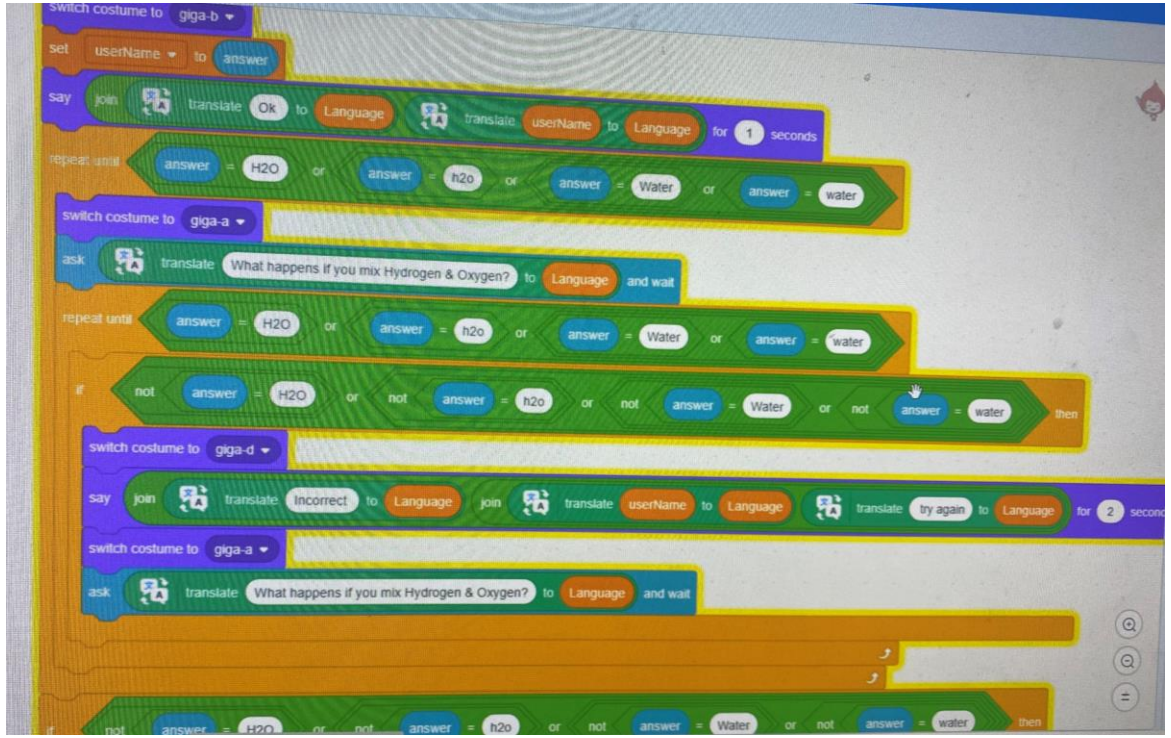
Time is up

Gatekeepers of knowledge

If students don't encounter programming knowledge early on many will discount it and exclude it from their worldview.

"It is too complex"

"Just for geeks"



Science quiz created by Bobby, age 10

code-it.co.uk

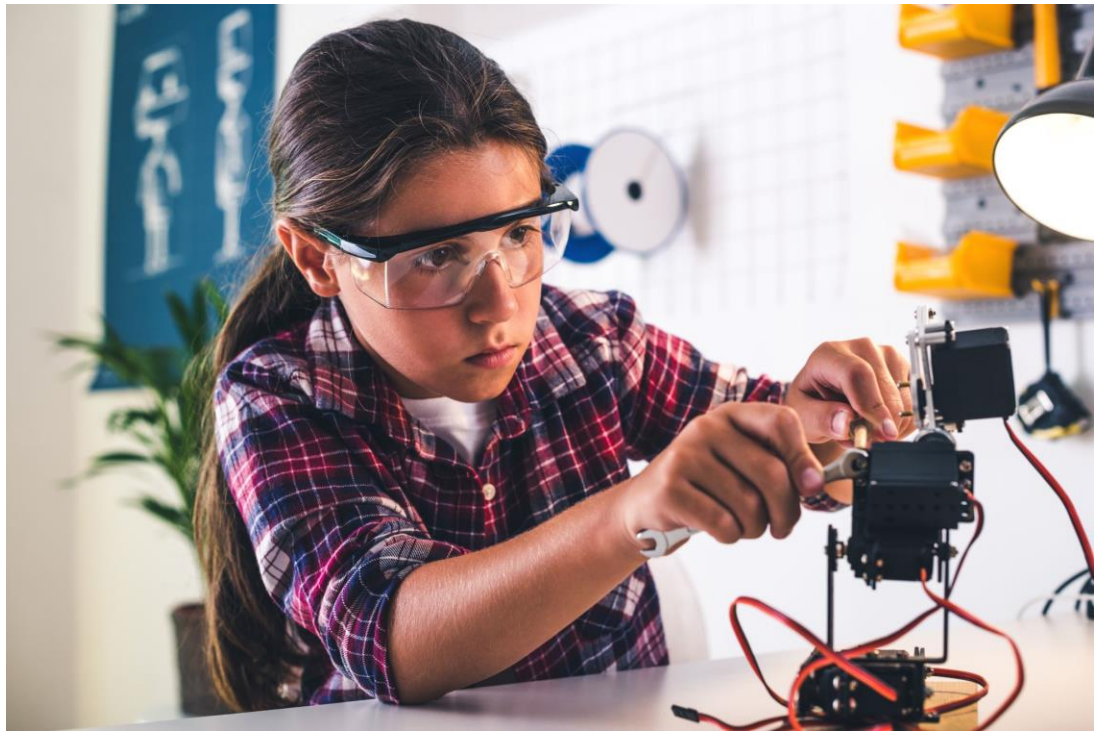
Addressing the gender imbalance

In elementary education there is the possibility of positive female tech role models (80% female teachers)

Once they get to the tech industry it can be as low as 26% female co-workers/role models

Is it our job to create programmers? No

- It takes 10 years to create a competent programmer
- We are demystifying how our digital world works
- We are opening the door to the possibility of more knowledge if that student wants it
- Programming is a great tool to explore shape, space & maths



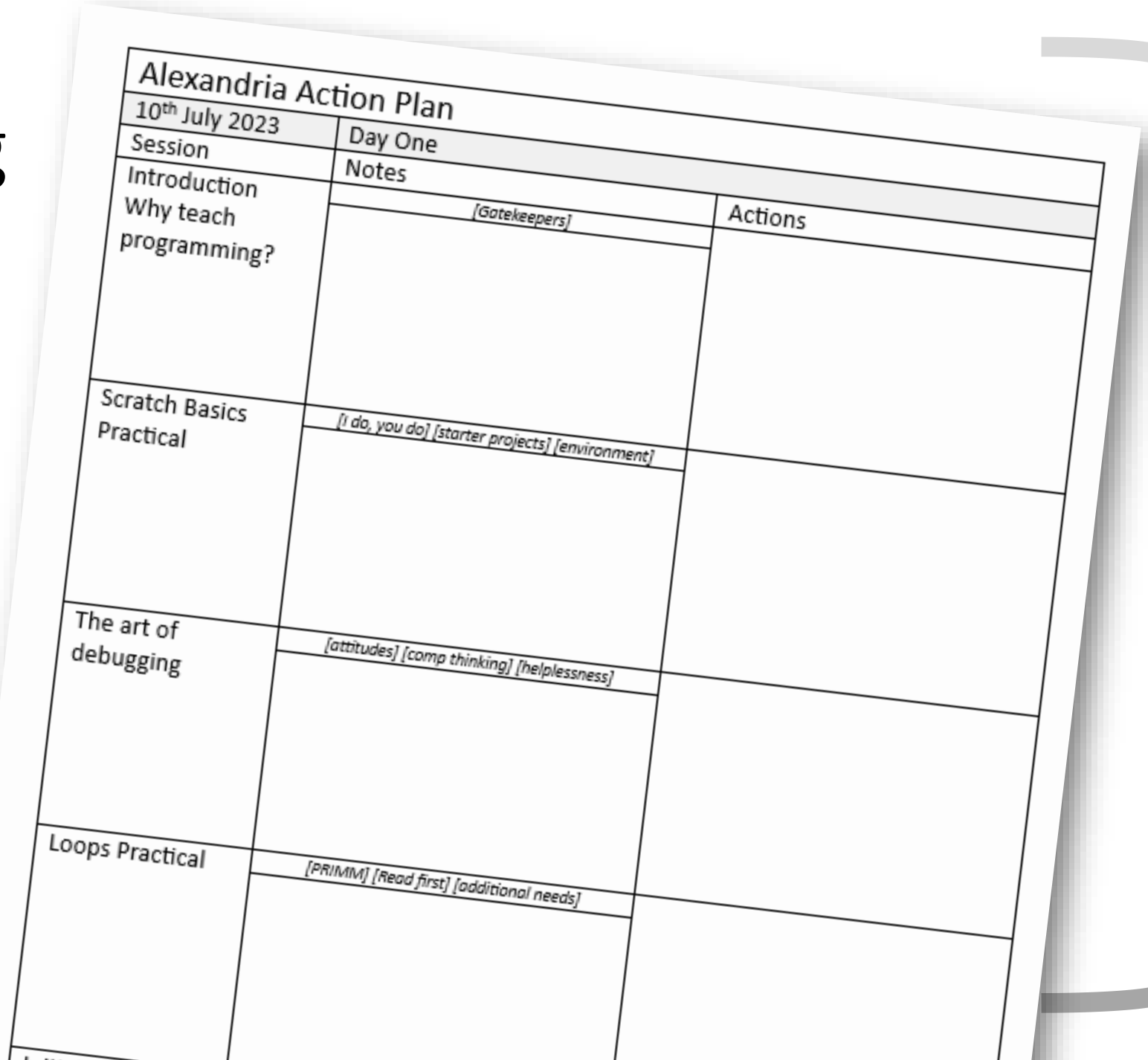
Programming is about thinking (computational thinking)

- Algorithmic thinking
- Decomposition
- Generalisation
- Abstraction

Action Planning

A place to make notes and think about follow up actions

You can find it on page 2



Alexandria Action Plan

10th July 2023	Day One	
Session	Notes	Actions
Introduction Why teach programming?	[Gatekeepers]	
Scratch Basics Practical	[I do, you do] [starter projects] [environment]	
The art of debugging	[attitudes] [comp thinking] [helplessness]	
Loops Practical	[PRIMM] [Read first] [additional needs]	

Online Resources

http://code-it.co.uk/alex

Scratch Basics Practical

Session Two

Phil Bagge @baggiepr

code-it.co.uk

Scratch Basics Practical Aims

Red	Little knowledge
Amber	Some knowledge
Green	Good knowledge

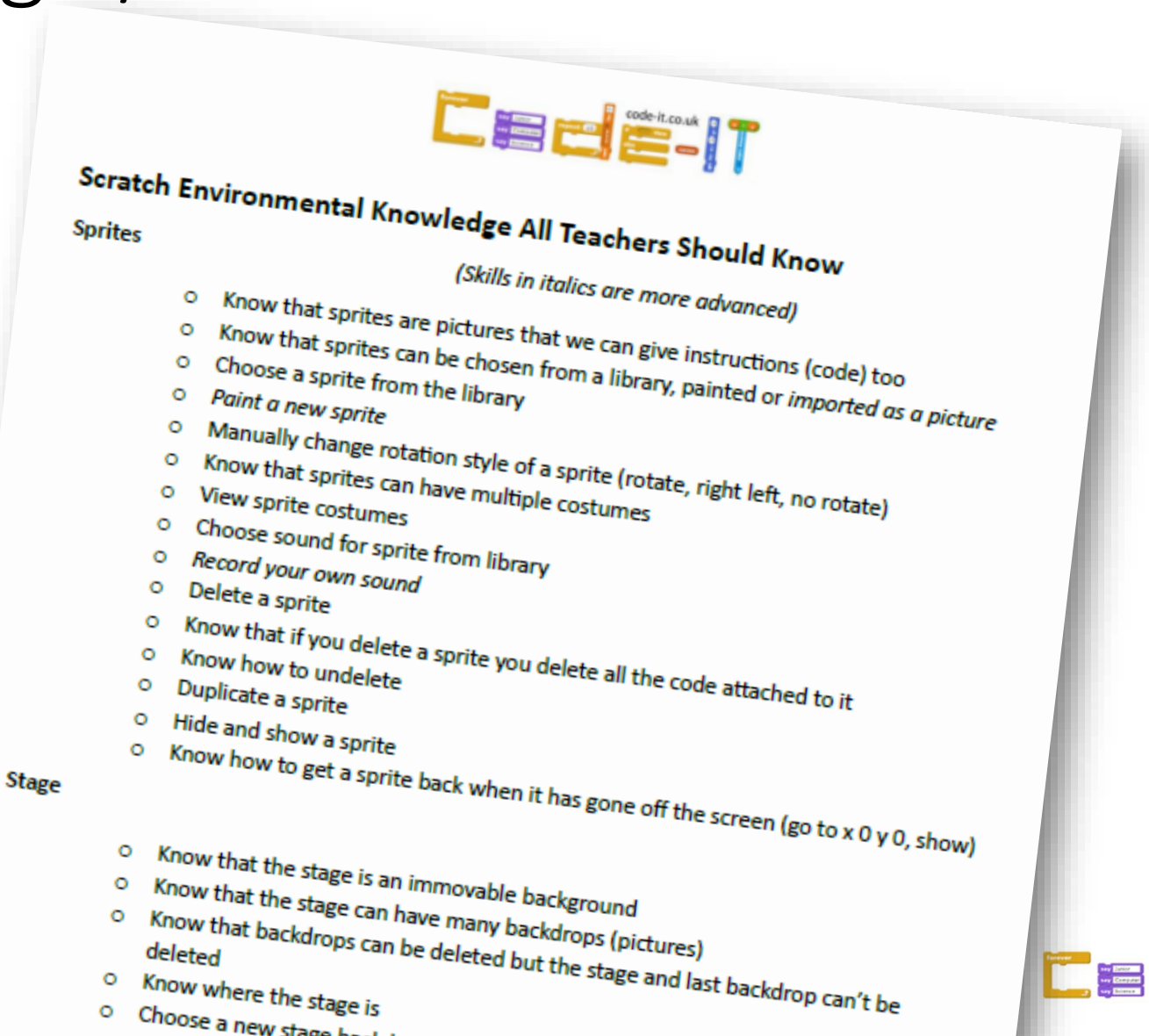
Aim	At start	At end
I can find my way around Scratch, costumes, backgrounds, sound editor, sprites, simple code creation and deletion		
I know how to provide a clear Scratch demonstration to my pupils		
I can evaluate I do you do / copy code as a teaching technique		

code-it.co.uk

I do you do / copy code / demonstration

Key knowledge / skills

A teacher's checklist for us and to use with colleagues



code-it.co.uk

Scratch Environmental Knowledge All Teachers Should Know

Sprites

(Skills in italics are more advanced)

- Know that sprites are pictures that we can give instructions (code) too
- Know that sprites can be chosen from a library, painted or *imported as a picture*
- Choose a sprite from the library
- *Paint a new sprite*
- Manually change rotation style of a sprite (rotate, right left, no rotate)
- Know that sprites can have multiple costumes
- View sprite costumes
- Choose sound for sprite from library
- *Record your own sound*
- Delete a sprite
- Know that if you delete a sprite you delete all the code attached to it
- Know how to undelete
- Duplicate a sprite
- Hide and show a sprite
- Know how to get a sprite back when it has gone off the screen (go to x 0 y 0, show)

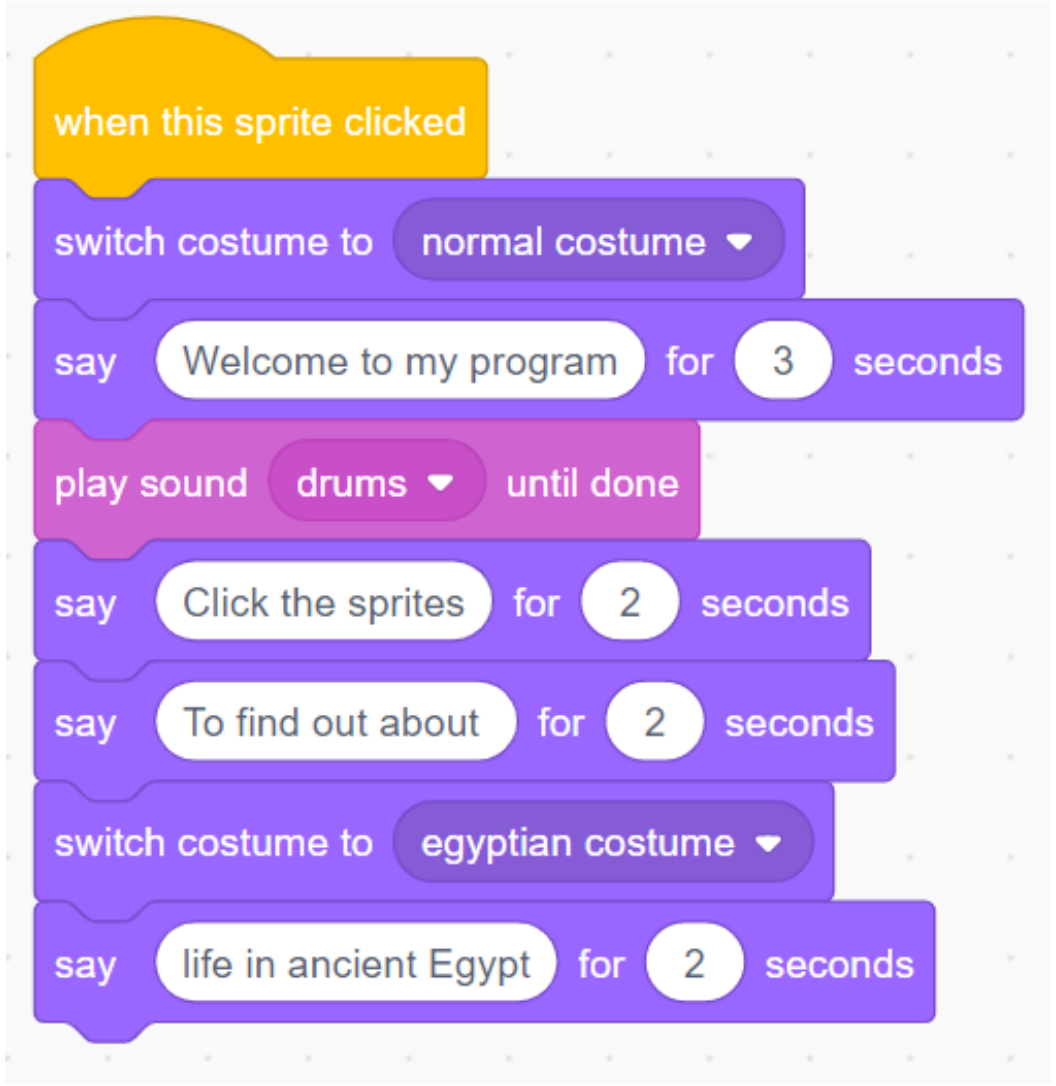
Stage

- Know that the stage is an immovable background
- Know that the stage can have many backdrops (pictures)
- Know that backdrops can be deleted but the stage and last backdrop can't be deleted
- Know where the stage is
- Choose a new stage

Phil Bagge @baggiepr

code-it.co.uk

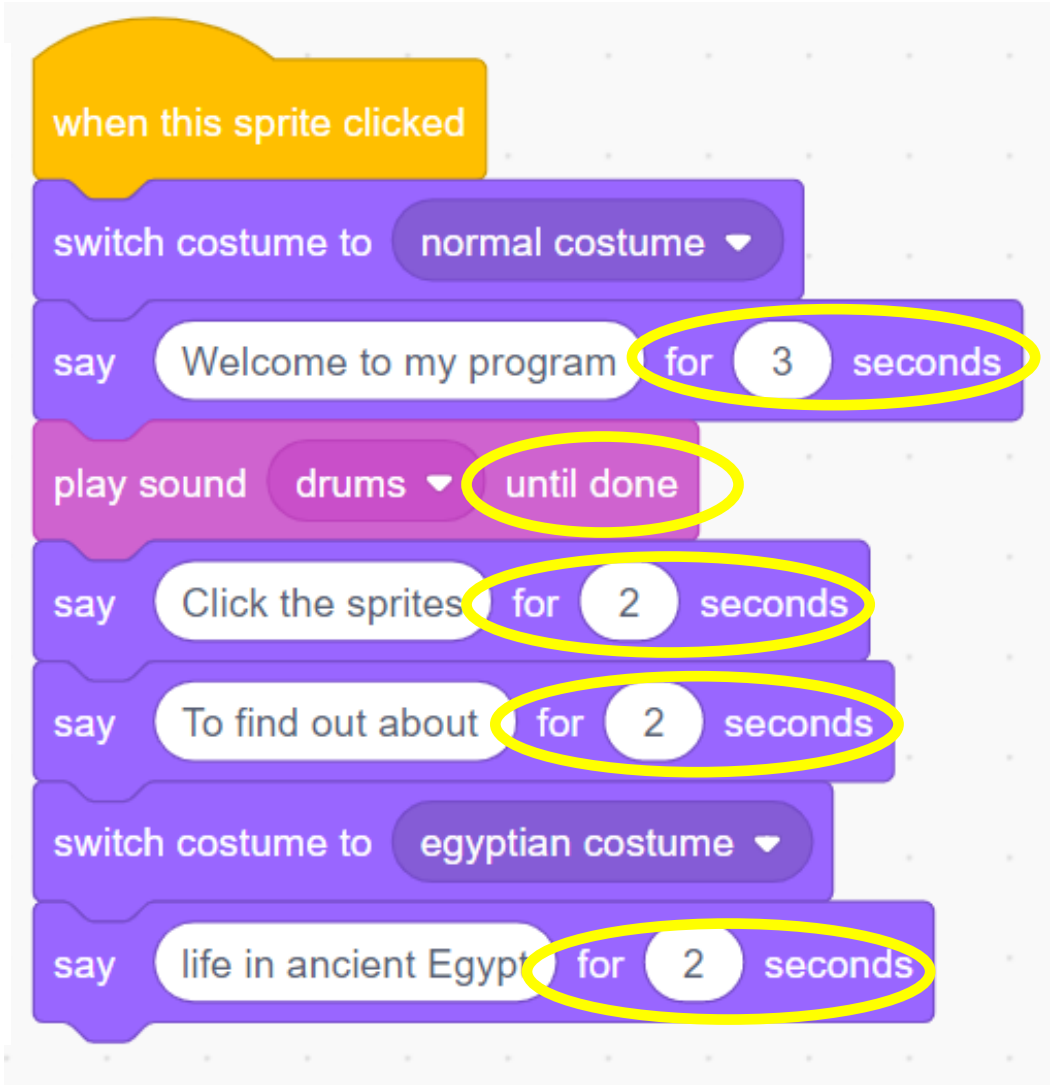
Sequence Example in Scratch Programming



Which blocks will run fast, and which will run slowly?

code-it.co.uk

Sequence Example in Scratch Programming



1,2 & 7 will run very fast

The rest will run slowly

code-it.co.uk

I do you do / copy code / demonstrations

Positive factors

- Useful to help Scratch novices make sense of the programming environment
- Hands on coding works well with younger users as there is less reading
- Some form of short demonstration is always useful alongside other strategies

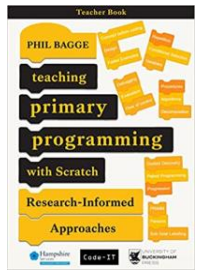
Negative factors

- Doesn't make you think deeply about code
- Not enough reading code before writing code (code comprehension)

Useful reading

I build you build p127

More clues p197



Scratch Basics Practical Aims Progress by end of session

Red	Little knowledge
Amber	Some knowledge
Green	Good knowledge

Aim	At start	At end
I can find my way around Scratch, costumes, backgrounds, sound editor, sprites, simple code creation and deletion		
I know how to provide a clear Scratch demonstration to my pupils		
I can evaluate I do you do / copy code as a teaching technique		

code-it.co.uk

The Art of Debugging

Session Three

Phil Bagge @baggiepr

code-it.co.uk

Bugs are

Errors in code

Unfinished code

Bugs are universal

- **Every** programmer makes **mistakes**
- Mistakes and debugging is a **normal** part of the programming cycle
- **Not** a teacher's job to debug pupil code
- We need to encourage **independence**
- We need to liberate pupils into messy problem solving

Learnt Helplessness

Define Helpless

Helpless ≠ stuck

Helpless = stuck + no attempt to find solution

Two types

1. Sweet helpless
2. Aggressive helpless

Where does it come from?

For Example

What aspect are you stuck on?

Everything

Can you describe the problem?

No answer

Which parts do work?

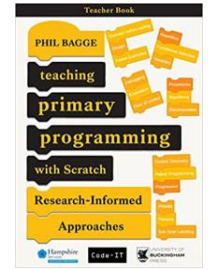
No answer

NO attempt to engage with process

Teachers, pupils and parents who solve too much for children

Tips and Tricks

Useful reading
Debugging p149
More than one method p159



FAQ Board

Debugging wall of fame

Loops Practical

Session Four

Phil Bagge @baggiepr

code-it.co.uk

Loops Practical Aims

Red	Little knowledge
Amber	Some knowledge
Green	Good knowledge

Aim	At start	At end
I know how PRIMM (predict, run, investigate, modify, make) can help pupils learn to programme		
I know the difference and similarities between programming and algorithm design		
I know how the flow of control works within a count-controlled loop		

Programming & Algorithms

Limited Commands

Written for a digital device to read and act on

Sequence
Repetition
Conditional selection
variables

Unlimited language & symbols

Written for a human to understand

Can be used to plan a programme or any instructions

code-it.co.uk

Everyday loops

Which parts of the dance are repeated?

Piaget linking new learning to old learning

Brain Breaks - Action Songs for Children - Happy Dance - Kids Songs by The Learning Station

Introduce Count controlled loop

do 4 times

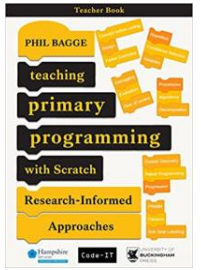
wave

Number controls how many times actions repeats

Useful reading

Concept before code

P65



Shuchi Grover

Concept

before code

code-it.co.uk

Count controlled loop

do 4 times

Actions **inside** a loop are indented

wave

code-it.co.uk

Count controlled loop

do 4 times
wave

Can you act out the algorithm?

code-it.co.uk

Count controlled loop

Linking sequence to our new learning

do 4 times

wave

Wave

Wave

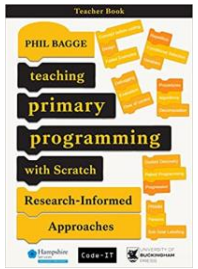
Wave

wave

Did you carry out these actions?

Useful reading

Simple sequence p13



code-it.co.uk

Count controlled loop

Stand

do 3 times

wave

Sit

Actions **inside** a loop are indented

Algorithms can be written in any way but indentation works well and is mirrored in Scratch code

code-it.co.uk

Count controlled loop

```
Stand
do 3 times
    wave
Sit
```

Actions **outside** a loop are not indented

Actions **outside** a loop are not indented

code-it.co.uk

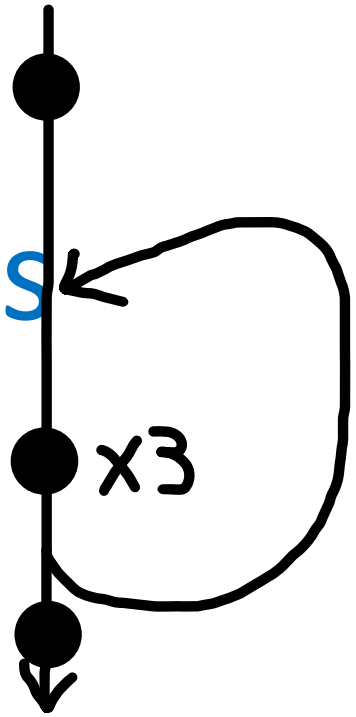
Flow of control

Stand

do 3 times

wave

Sit



Count controlled loop

```
stand
loop 4 times
    wave
    bow
sit
```

Can you act out the algorithm?

code-it.co.uk

Count controlled loop

```
stand
loop 4 times
    wave
    bow
sit
```

stand
Wave
Bow
Wave
Bow
Wave
Bow
Wave
Bow
sit

Did you carry out these actions?

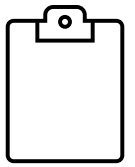
code-it.co.uk

Count controlled loop

```
stand
loop 4 times
    wave
    bow
sit
```

stand
Wave
Bow
Wave
Bow
Wave
Bow
Wave
Bow
sit

What actions are **inside** the loop?



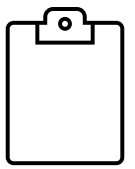
code-it.co.uk

Count controlled loop

```
stand
loop 4 times
    wave
    bow
sit
```

stand
Wave
Bow
Wave
Bow
Wave
Bow
Wave
Bow
sit

What actions are inside the loop?



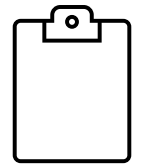
code-it.co.uk

Count controlled loop

```
stand
loop 4 times
    wave
    bow
sit
```

stand
Wave
Bow
Wave
Bow
Wave
Bow
Wave
Bow
sit

What actions are **outside** the loop?



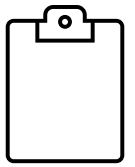
code-it.co.uk

Count controlled loop

```
stand
loop 4 times
    wave
    bow
sit
```

stand
Wave
Bow
Wave
Bow
Wave
Bow
Wave
Bow
sit

What actions are outside the loop?



code-it.co.uk

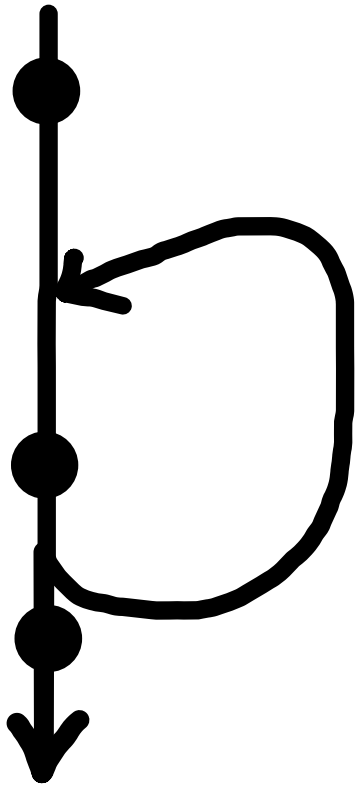
Count controlled loop

Stand

do 3 times

wave x3

Sit



Flow of control reminder

code-it.co.uk

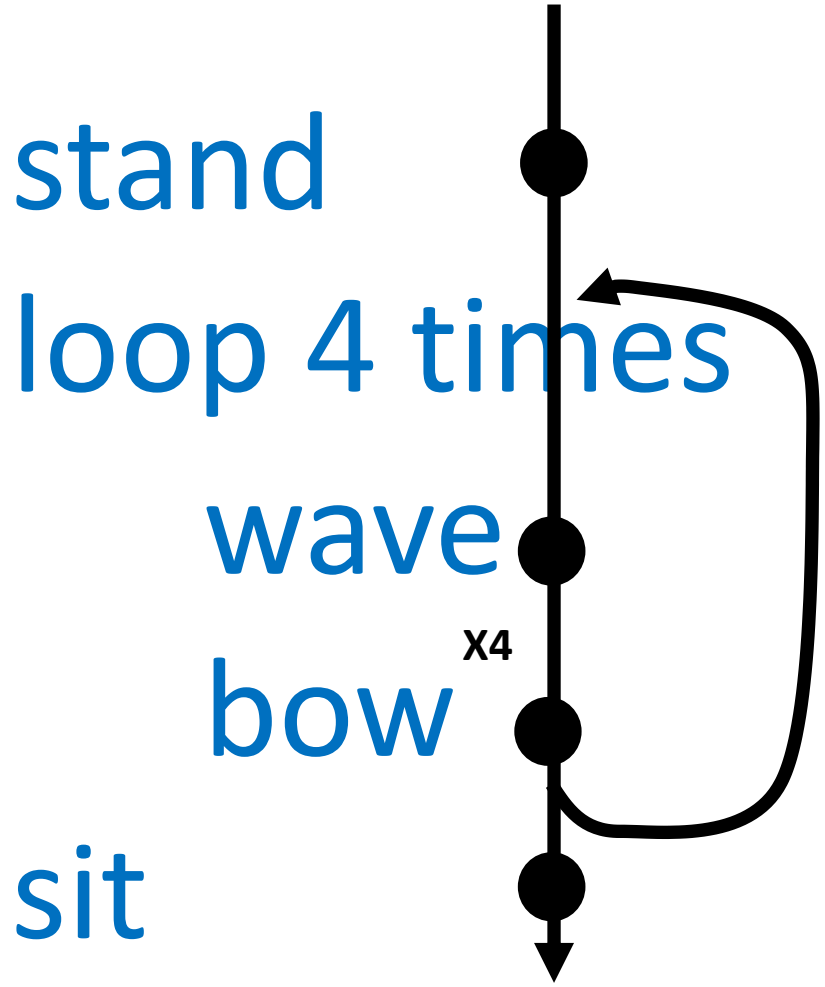
Count controlled loop

```
stand
loop 4 times
    wave
    bow
sit
```

Draw the flow of control

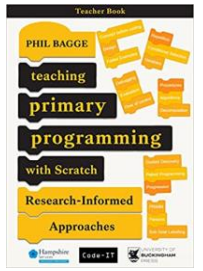
code-it.co.uk

Count controlled loop



Useful reading

Repetition p19

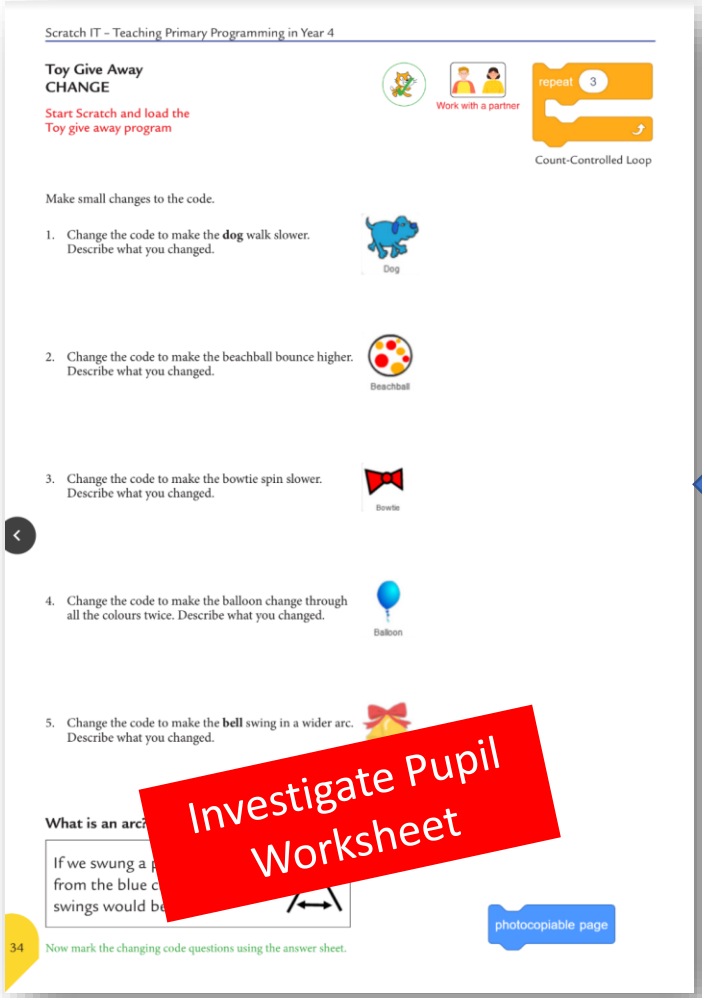


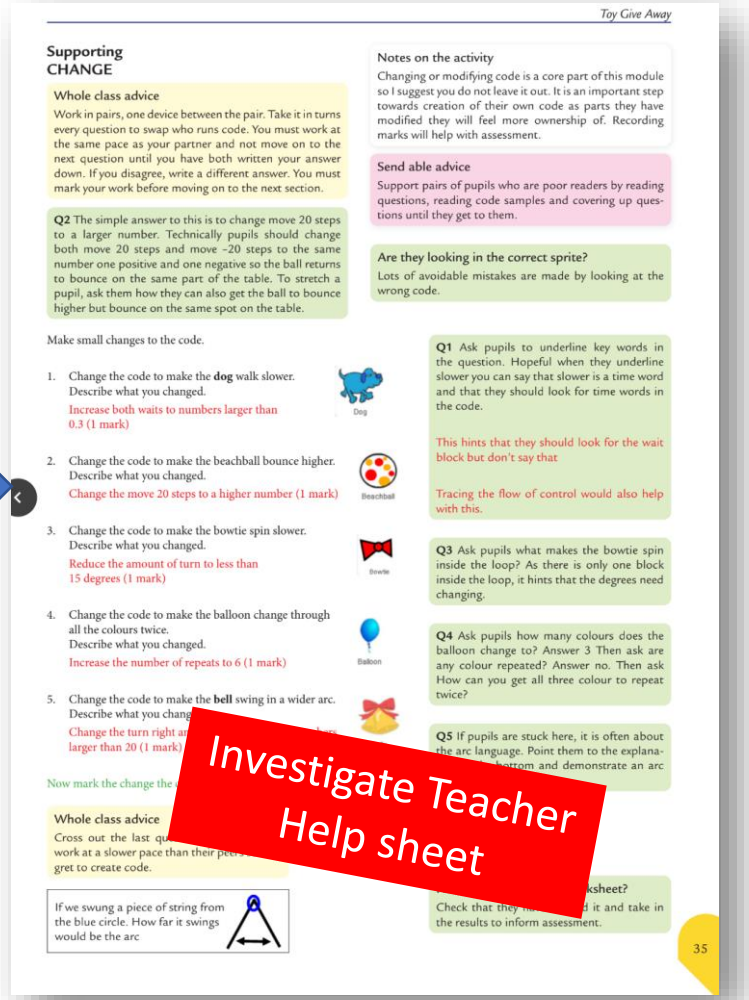
PRIMM Challenge

1. Predict Sheet first without your PC
2. Load Toy Give Away on one computer between two teachers
3. Work together on Run, Investigate & Modify before marking these challenges
4. Finally on a computer each, Make

What problems would some students have whilst using PRIMM?

Stuck Pupils





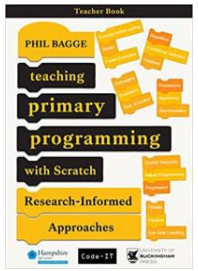
Most pupils will not need this help but it will support teachers for those that do

code-it.co.uk

Recommended Paired Programming

- Concept (Solo)
- Predict (Paired)
- Run (Paired)
- Investigate (Paired)
- Modify (Paired)
- Make (Solo partner supported)

Useful reading
Paired programming
p137



Assessment

- Concept (Formative observations)
- Predict (Pupil Marked)
- Run
- Investigate (Pupil Marked
- Modify (Pupil Marked)
- Make (Summative Teacher Marked)

Assessment

Teacher & Pupil Assessment

Circle one column on each row to show what you think you have achieved

	Not used a count control loop	Copied a count control loop from the toy give away project	Copied and changed a count control loop idea	Used a count control loop in a way not shown
Count controlled Loops	0 Marks	1 mark	2 marks	3 marks

	Not used previous programming concepts for real purpose	Used previous programming concepts for real purpose
Used previous programming concept such as sequence and inputs	0 Marks	1 mark

(Only to be used if make second has been planned and created)	No theme in planning or code	Has a theme in planning or code
Has a project theme in planning or code	0 Marks	1 mark

code-it.co.uk

Key Knowledge

Key Repetition Knowledge

- A loop is a sequence of instructions that are repeated
- **Indefinite** means that the number of loops completed will not be known
- **Definite** means the number of loops are known.
- A **count-controlled-loop** ends after the count is finished and is a definite loop
- An **infinite** loop never ends and is an indefinite loop

Declarative knowledge

Static facts or knowledge stored in your memory (what)

Procedural knowledge

How to perform a specific skill or task (how)

Conditional Knowledge

When to use declarative and procedural knowledge (when & why)

code-it.co.uk

Key Knowledge

Declarative knowledge

Static facts or knowledge stored in your memory (what)
Procedural knowledge

How to perform a specific skill or task (how)
Conditional Knowledge

When to use declarative and procedural knowledge (when & why)

Repetition Algorithmic Knowledge

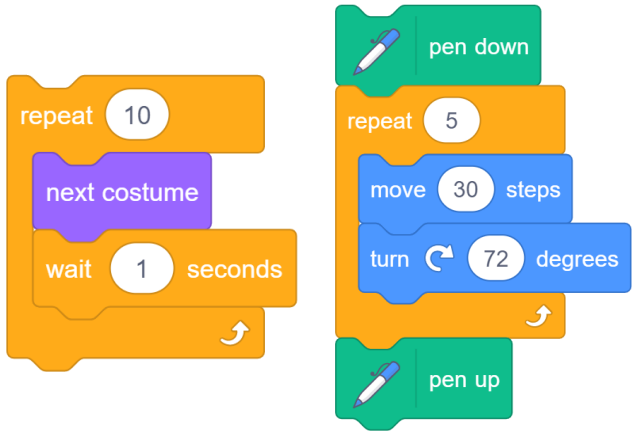
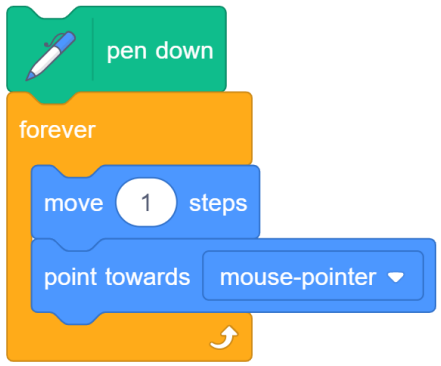
- I can write a loop as an algorithm using appropriate loop language such as do so many times, loop always, loop until or repeat x times.
- I can indent actions inside a loop to make it clear they are inside the loop

Repetition Programming Knowledge

- I can use a count-controlled-loop where I want a sequence of actions to take place for a specific amount of time or over a specific distance
- I can use an indefinite infinite loop where I don't want the loop to end once it has started

code-it.co.uk

Loops compared

	Count controlled loop	Indefinite forever loop
Algorithm	**Stand** **Loop 4 times** **bow** **wave** **Sit**	**Stand** **Loop always** **Forward I step** **Right turn 90 degrees**
Code		

Progression within loops is supported by repetition trajectories produced by university of Chicago everyday computing team

code-it.co.uk

Understanding Count Controlled Loops

A loop is a set of instructions that are repeated

Abstract & complex meanings

There are more than one type of loop

Some loops are controlled by a number we call them count controlled loops

Dance moves or song lyrics often contain count controlled loops

We can write everyday algorithms that use count controlled loops

Concrete and simpler meanings

Pick up drum stick
Loop 10x
hit drum skin
Put down drum stick

Pupils write own everyday algorithm that uses a count controlled loop

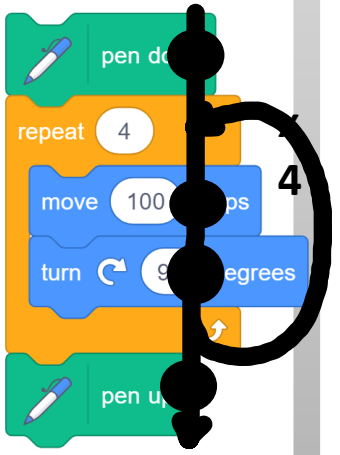
Pupils introduced to count controlled loop flow of control (notional machine)

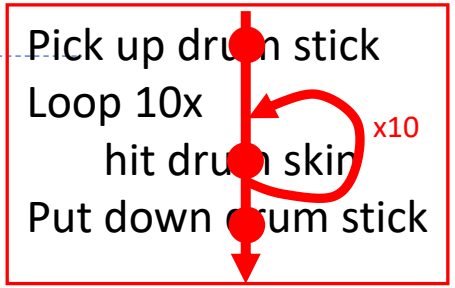
What a count controlled loop looks like in code

How a count controlled loop ends

How a sequence can be written as a count controlled loop

Count controlled loops are called definite loops because we know when they will end





Semantic Wave K Maton

Phil Bagge @baggiepr

code-it.co.uk

Understanding Infinite Loops

Similarities and differences between infinite and count controlled loop

A loop is a set of instructions that are repeated

Abstract & complex meanings

There are more than one type of loop

Some loops never end and we can call them always, forever or infinite loops

The days of the week and planets circling the sun are good examples of (almost) infinite loops

We can write everyday algorithms that use infinite loops

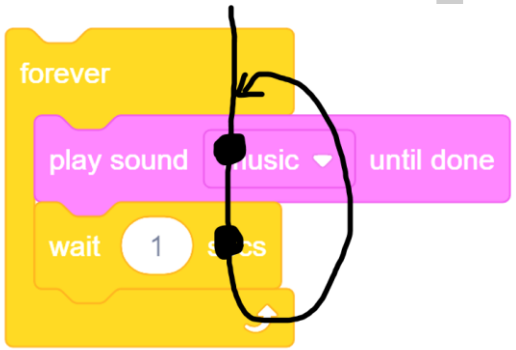
Concrete and simpler meanings

lean forward in the car
Loop always
say are we there yet
Wait 2 minutes

Pupils write own everyday algorithm that uses an infinite loop

Infinite loops are called indefinite loops because we don't know how many repetitions they will do or when they will end (if ever)

What an infinite loop looks like in code



How an infinite loop never ends

Pupils introduced to infinite loop flow of control (notional machine)

lean forward in the car
Loop always
say are we there yet
Wait 2 minutes

Semantic Wave K Maton

Loops Practical Aims
Progress by end of the session

Red	Little knowledge
Amber	Some knowledge
Green	Good knowledge

Aim	At start	At end
I know how PRIMM (predict, run, investigate, modify, make) can help pupils learn to programme		
I know the difference and similarities between programming and algorithm design		
I know how the flow of control works within a count-controlled loop		

Initialization Matters

Session Five

Phil Bagge @baggiepr

code-it.co.uk

Initialization Matters

Red	Little knowledge
Amber	Some knowledge
Green	Good knowledge

Aim	At start	At end
I know what initialisation is and can identify it in code		
I know initialisation analogies and can think of ways to support students understanding		

Initialisation

Later, in our student's journey towards programming knowledge initialisation will come to mean assigning a value to a variable before it is used.

At this point it is better described as **creating instructions so that the program always runs the same way every time it is run**

code-it.co.uk

Can you give me two instructions a move followed by a turn?

Initialisation Analogies

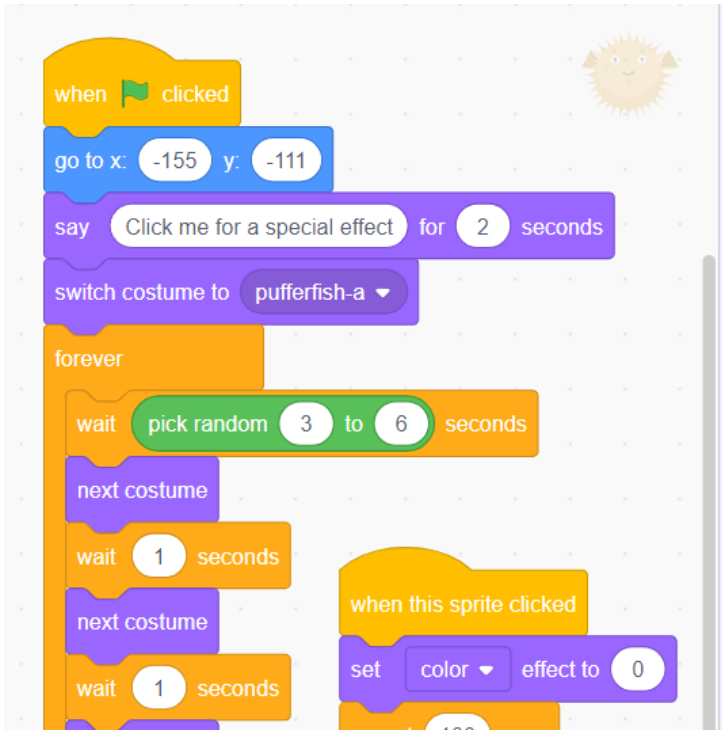
code-it.co.uk

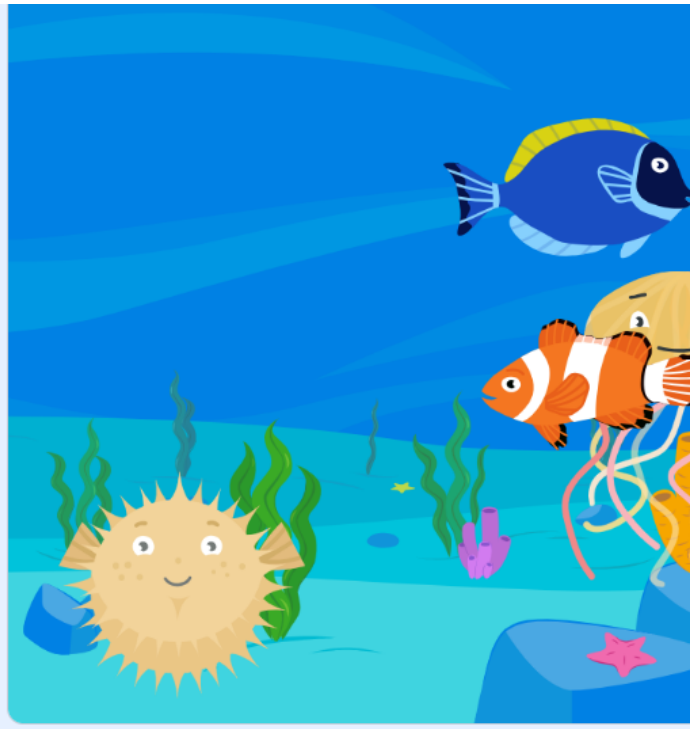
Identify initialisation code

Work with a partner to label examples of initialisation in this Fish Tank programme written to introduce pupils to indefinite forever loops

https://scratch.mit.edu/projects/509454779/editor/

Are there any initialisation blocks you don't understand?





code-it.co.uk

Compare a game with initialisation to one without

Helicopter with initialisation

https://scratch.mit.edu/projects/316961043/editor/

Helicopter without initialisation

https://scratch.mit.edu/projects/869441256/editor/

What goes wrong without initialisation?

Are there any blocks that might classify as initialisation that have not been removed?

Initialization Matters
Progress by end of the session

Red	Little knowledge
Amber	Some knowledge
Green	Good knowledge

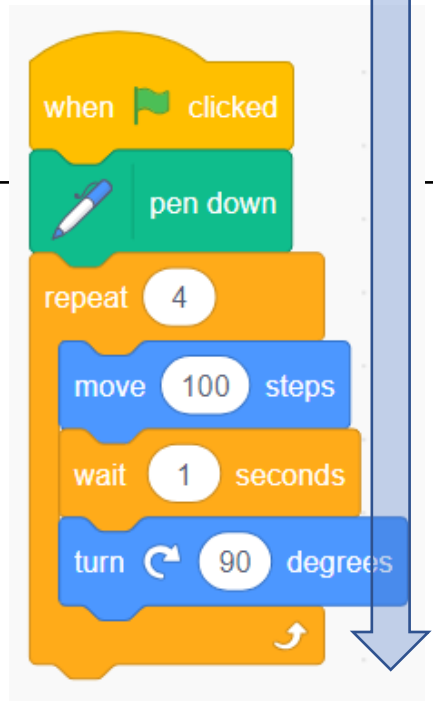
Aim	At start	At end
I know what initialisation is and can identify it in code		
I know initialisation analogies and can think of ways to support students understanding		

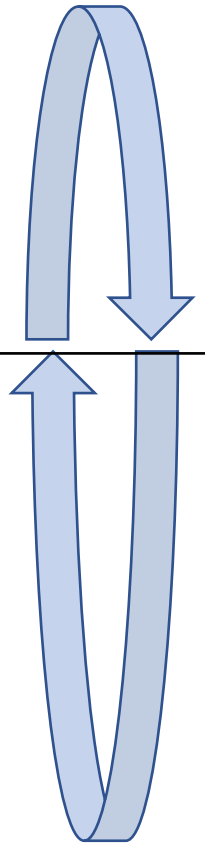
Parsons Problems

Phil Bagge @baggiepr

code-it.co.uk

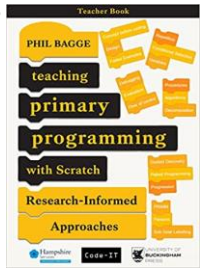
Task Level	Draw a square shape using a loop. *Pictorial representation very useful for many projects.*
Design Level (algorithm, objects & initialisation) *Language a human will understand, code agnostic*	One character to draw the shapes (objects). Use clear to remove previous drawing (initialisation) Start drawing Loop 4 times walk 100 steps quarter turn to the right Stop drawing (algorithm)
Code Level	
Execution Level	I needed to put a wait block in so I could see it as it drew on screen. *What did we learn, change, adapt when we ran the code?*





Levels of abstraction

Useful reading
Design
p 107



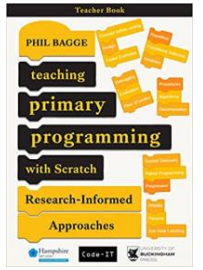
Supported by the research of Armoni 2013, Waite et al 2017

code-it.co.uk

A Parson's Problem is a task that involves giving learners all of the blocks of code required to solve a problem, but with the blocks jumbled so that they are not in the correct order.

Parsons Problem

Useful reading
Parsons Problems
p 133



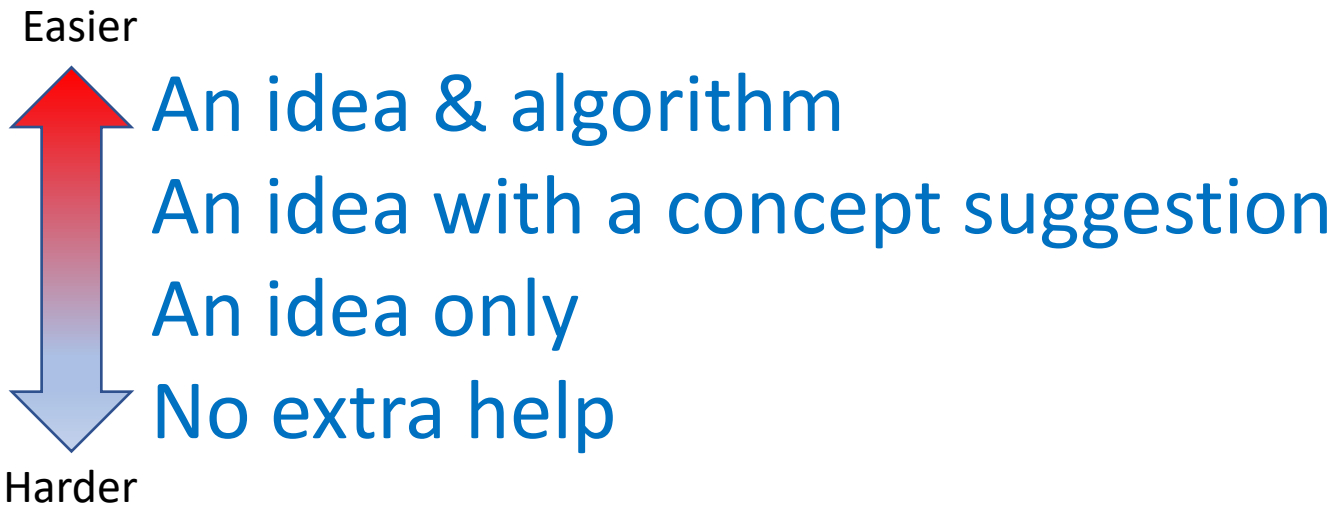
Phil Bagge @baggiepr

code-it.co.uk

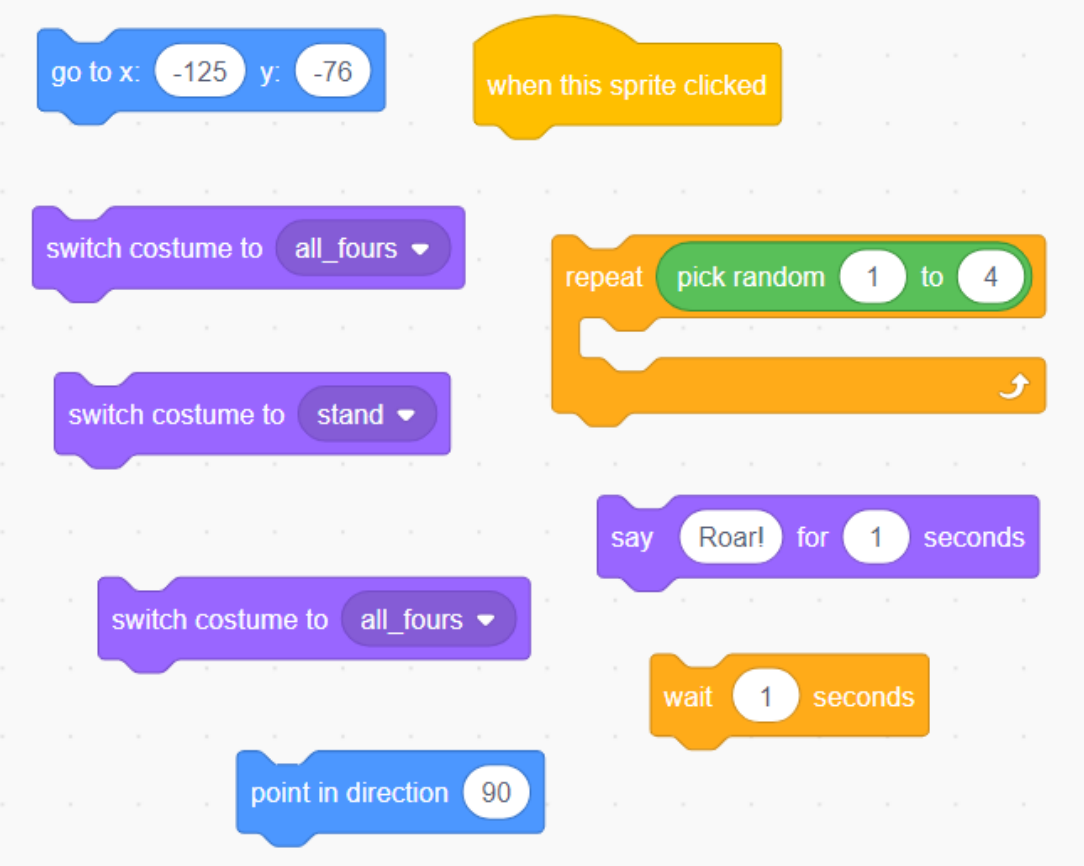
Parsons problems

Pupils are given a Scratch file with all the code disconnected for them to build a programme

they could also be given



A plan & algorithm



PLAN

Program to make a bear rear up and roar a random number of times

ALGORITHM

Click sprite

Go to place on screen x and y

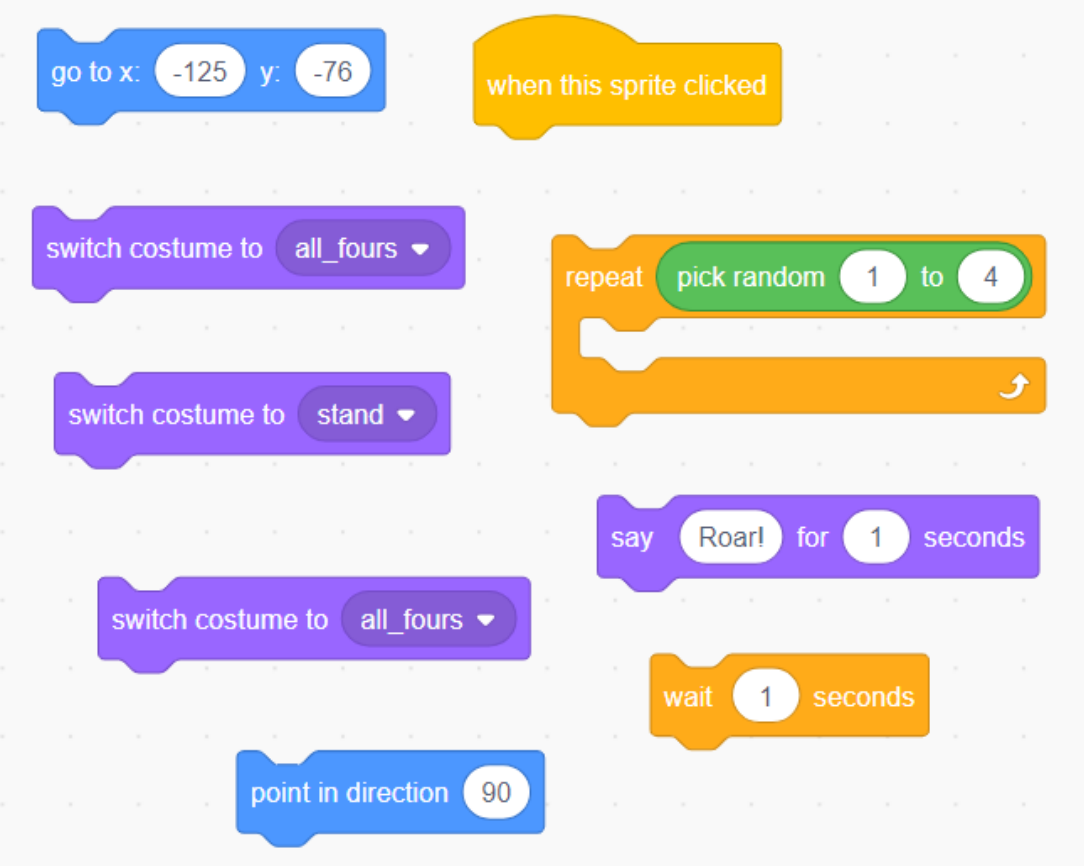
Point direction 90 (right)

Costume all fours

Loop 1, 2, 3 or 4 times randomly

- costume stand
- Say roar
- costume all fours
- pause 1 second

A plan & concept suggestion



PLAN

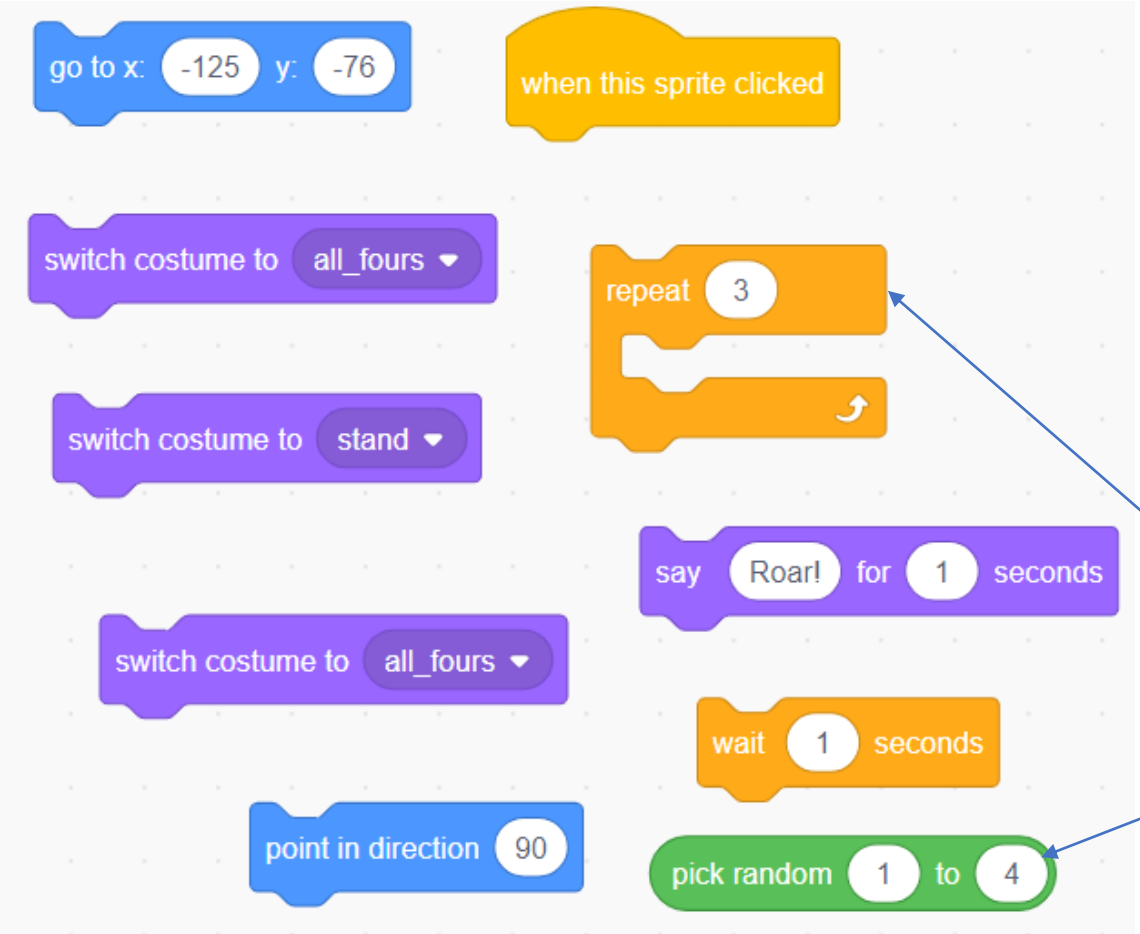
Program to make a bear rear up and roar a random number of times

Concepts to use

Repeat loop

Pick random block

An idea only



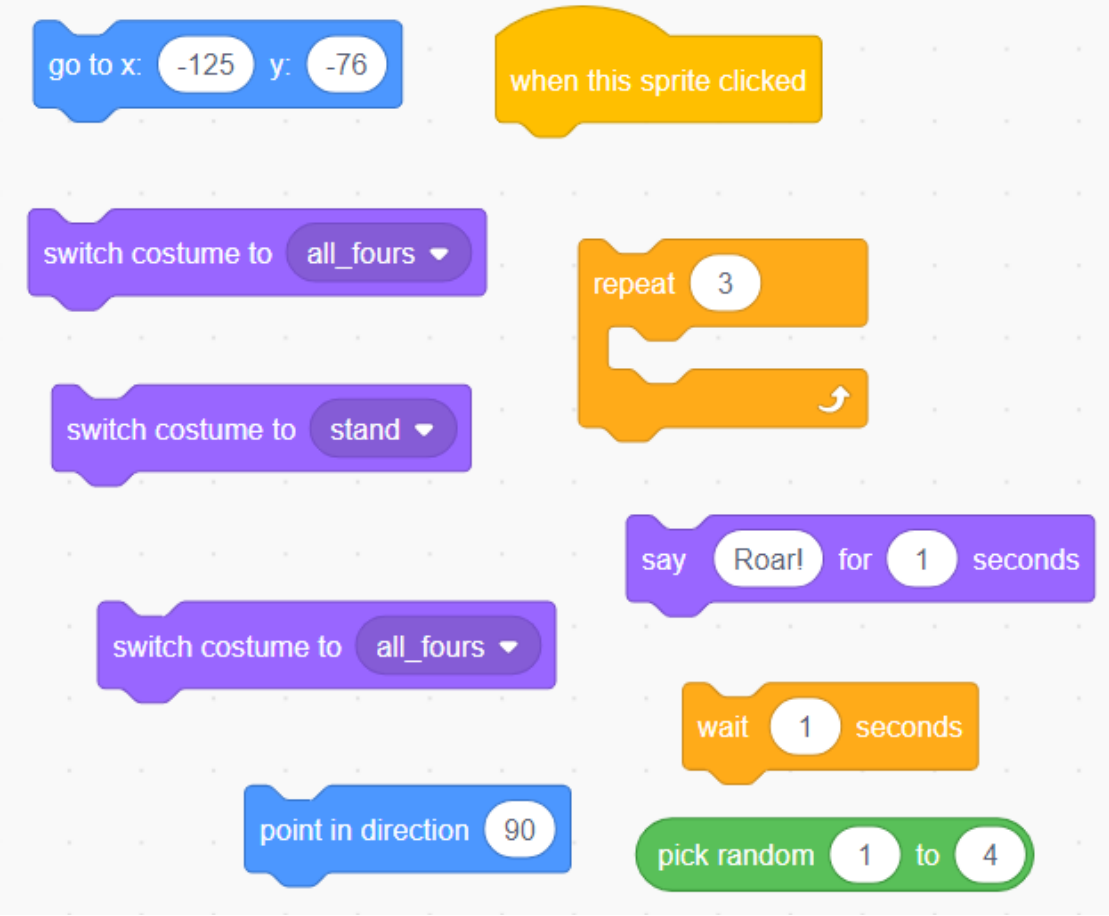
PLAN

Program to make a bear rear up and roar a random number of times

NOTE

Random has been separated from repeat loop to increase complexity

Code only with no planning suggestion



Some would add distractors

Blocks that are not needed

For many pupils that can be a step too far.

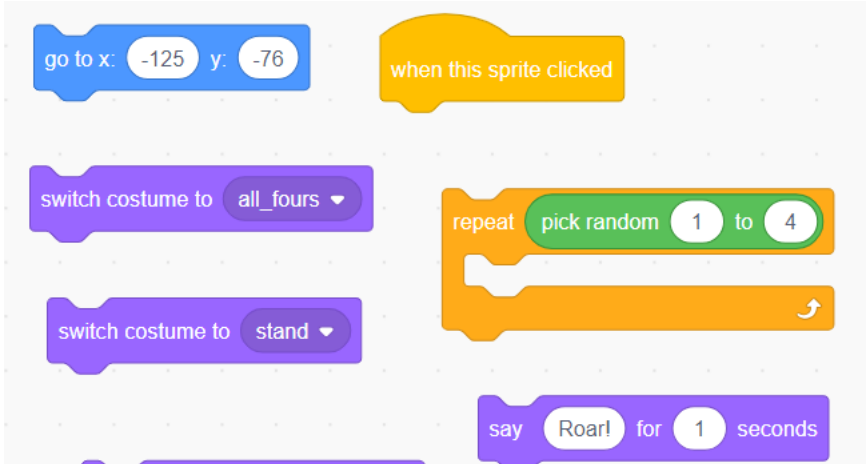
Parsons code positives and negatives

Positives

- Hands on
- Pupils get to see parts of planning phase
- Easy to adapt for pupils needs
- Good for pairs

Negatives

- Encourages the idea that there is only one way to solve a problem when there might be many
- The longer the algorithm the harder it is to turn into code



Progression in Programming

Session Eight

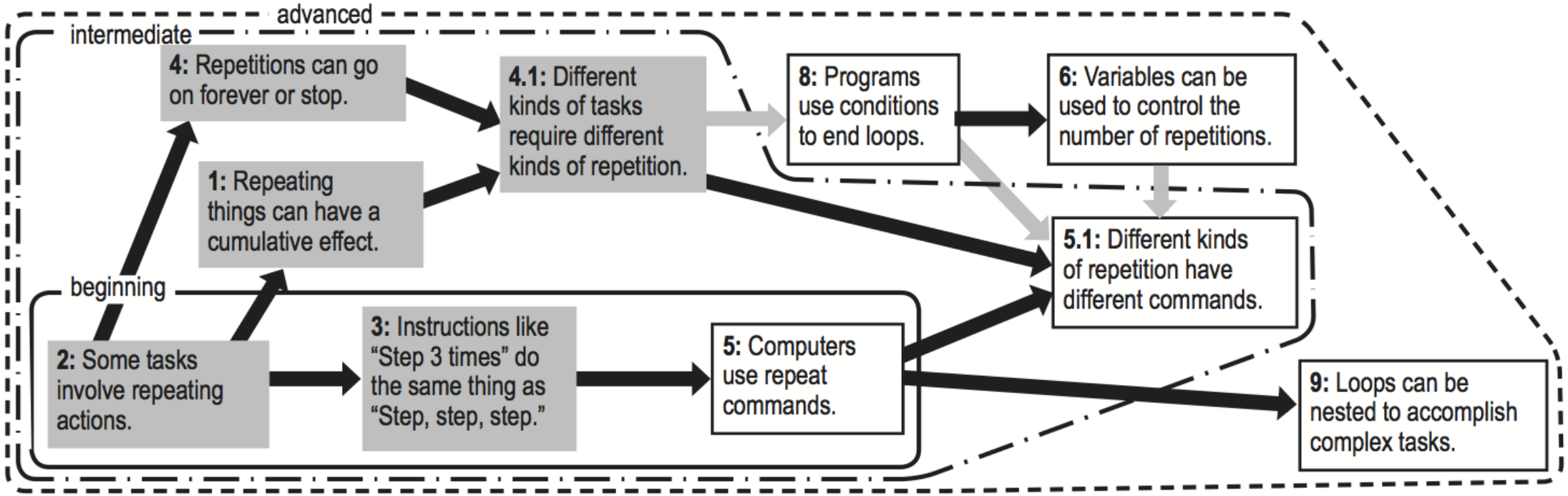
Phil Bagge @baggiepr

code-it.co.uk

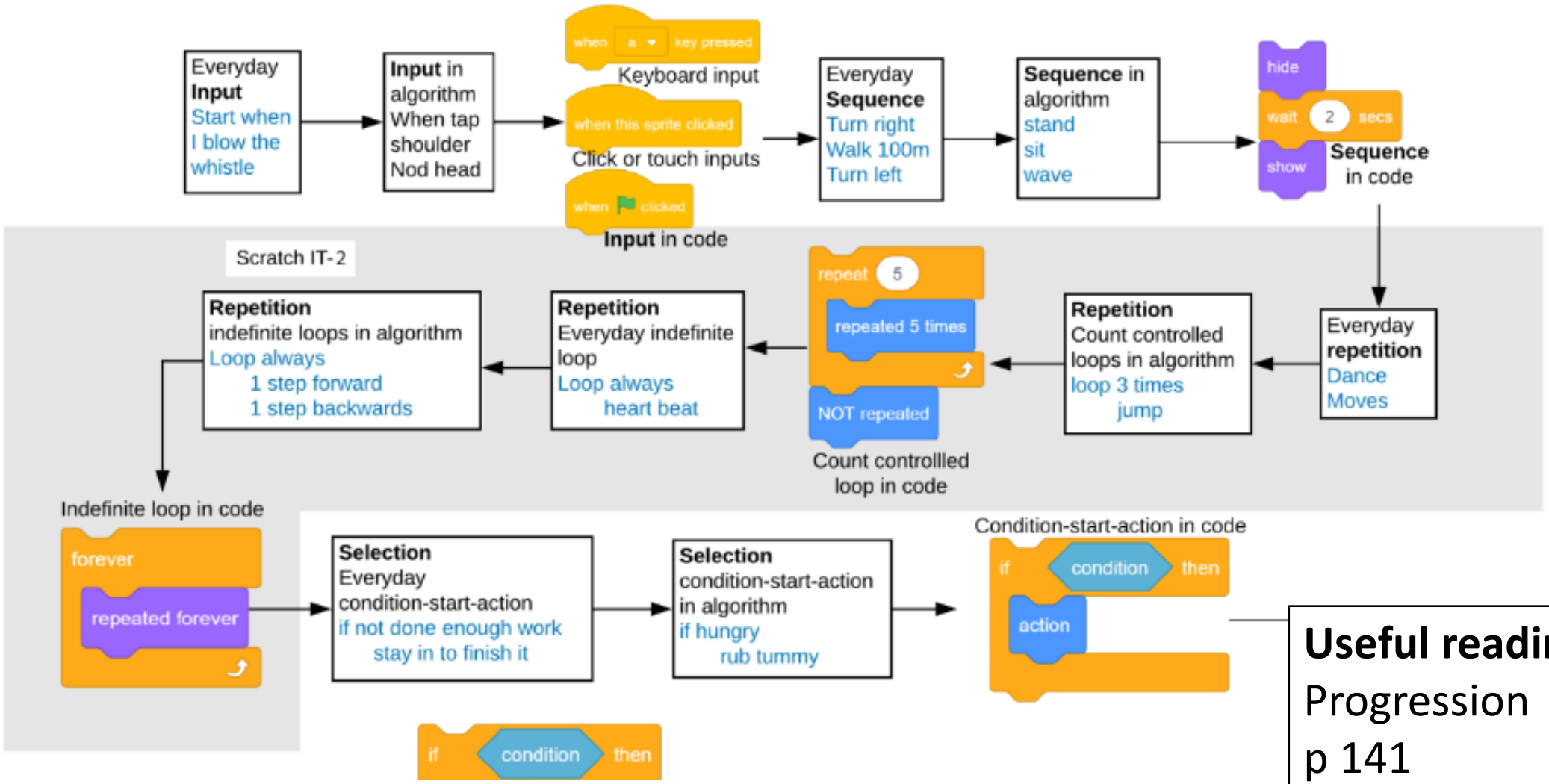
Progression research

Everyday computing

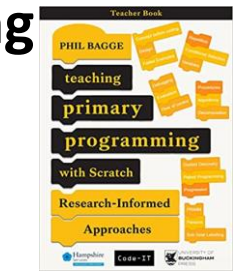
- http://everydaycomputing.org/public/visualization/



Progression



Useful reading
Progression
p 141



Shapes in the wider curriculum

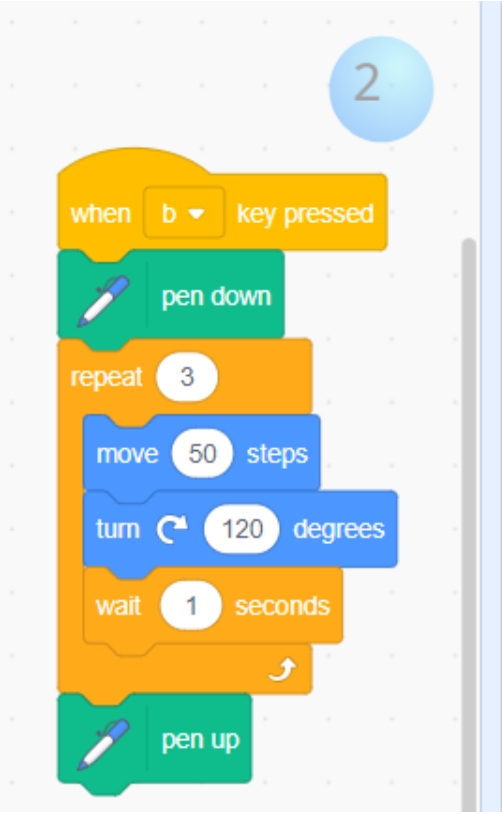
Session Seven

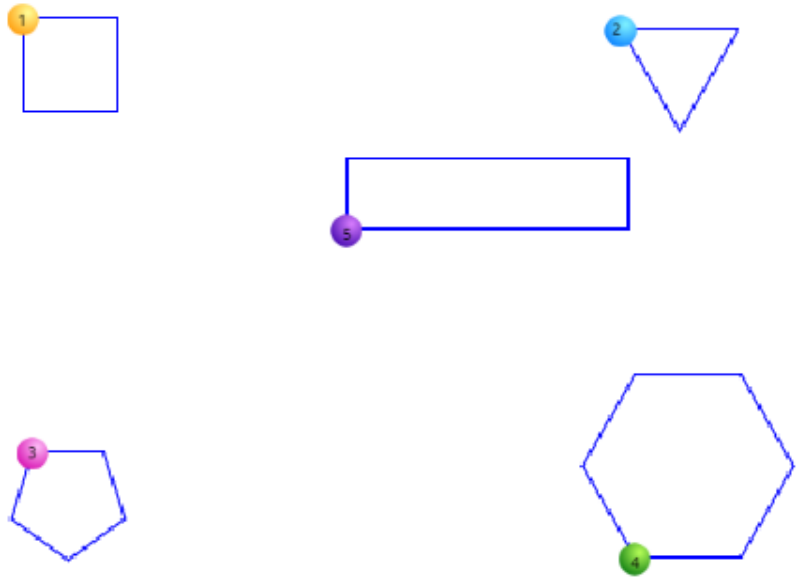
Phil Bagge @baggiepr

code-it.co.uk

Integrating with maths

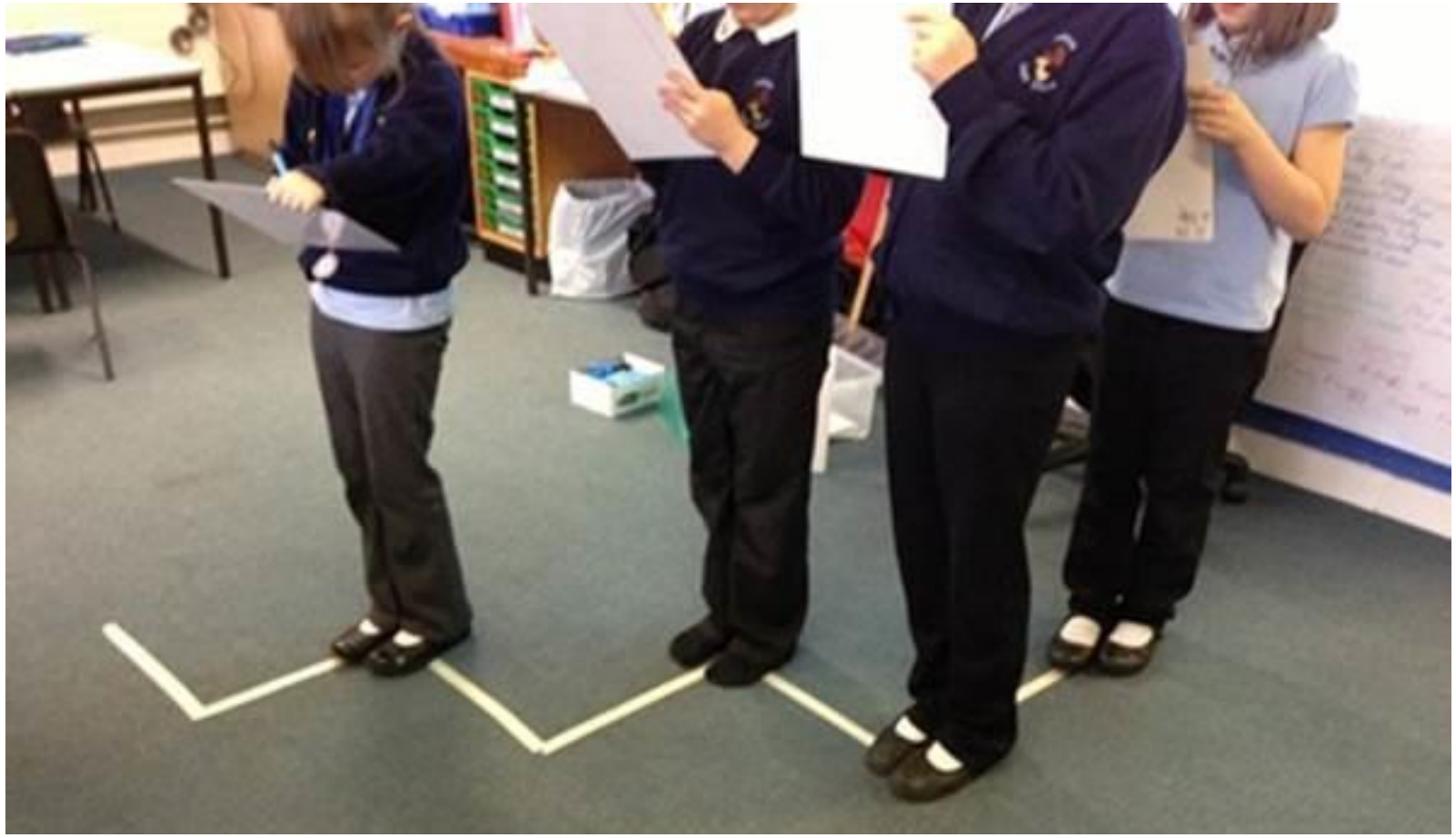
- 2D shape knowledge (properties of shapes)





http://code-it.co.uk/goldshape/5/

Body syntonic shape algorithms



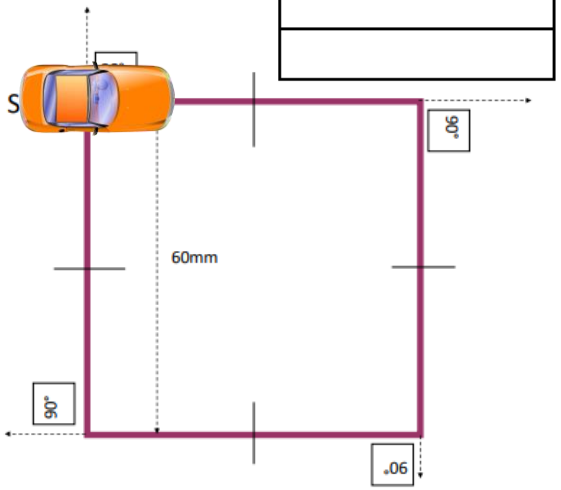
code-it.co.uk

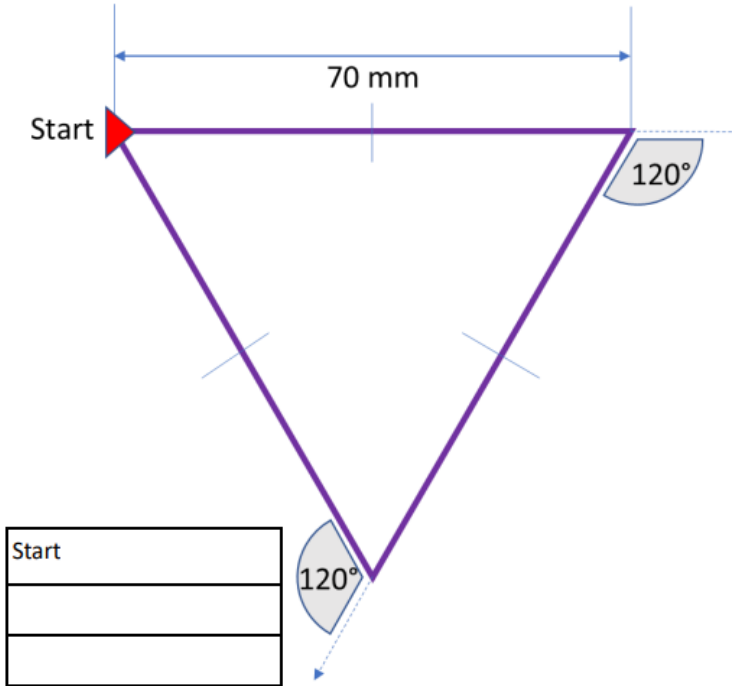
Drawing shape algorithms

Pupil Shapes Sequence Algorithm

Recording Sheet

Start





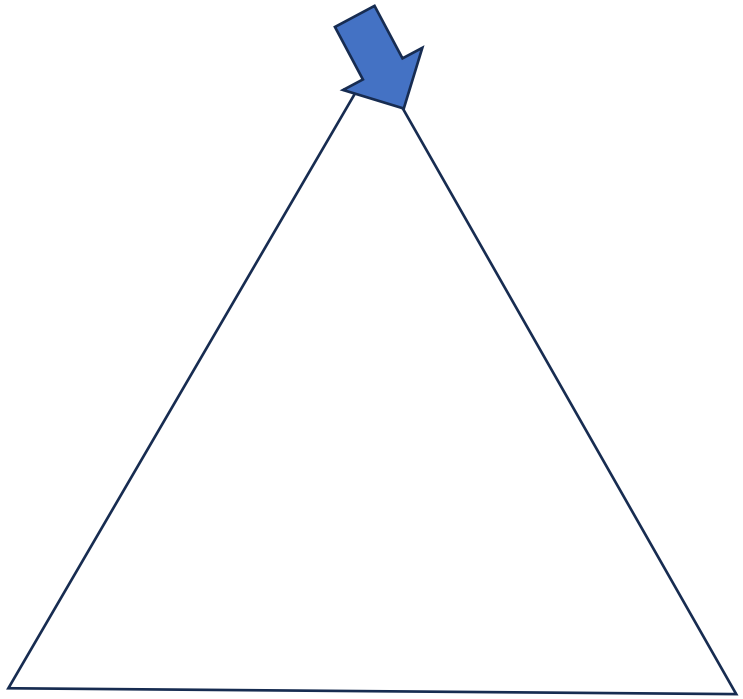
Start

Name
Class

CC BY NC ND

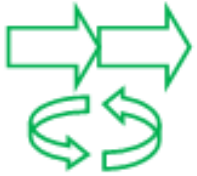
http://code-it.co.uk/wp-content/uploads/2018/04/PupilsShapesSequenceSheet.pdf

Misconceptions Inner and outer angles



- Using move and turn commands
- We use outer angle
- In early maths we are often using inner angle
- Need to make this clear

A resourced shape progression using PRIMM

Name	Sequence	Exploring Loops	Sequence to Repetition	Basic Procedures	Nested loop with procedures	Nested Procedures	Shape & Variables	Adaptable Procedures
Programming Concepts								
Difficulty								
Has offline unplugged introduction to								

http://code-it.co.uk/goldshape/

Adaptable procedures work sample

Context

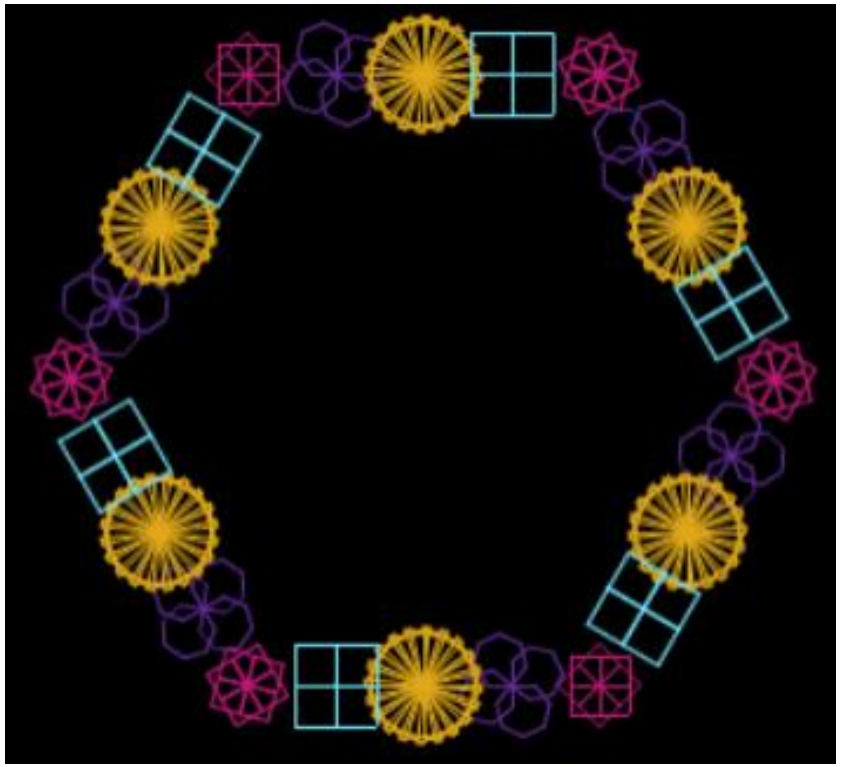
Age 9-10

Sequence of learning

- basic procedures,
- nested loops with procedures,
- nested procedures and
- adaptable procedures.
- Worked on variables to solve maths problems using Scratch
- Worked on loops using control board

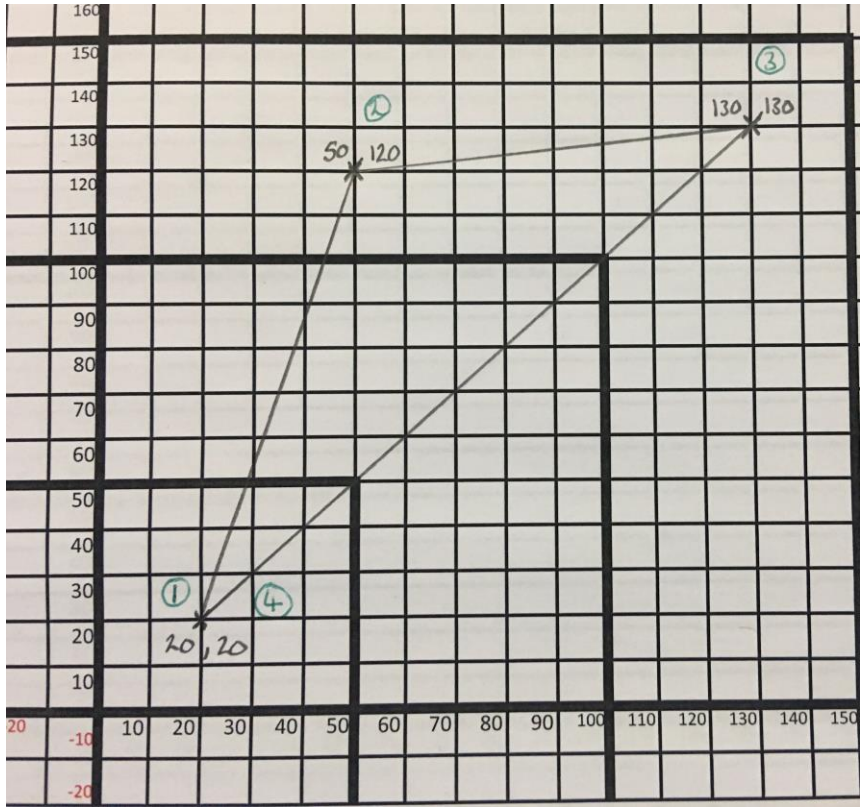
2019

Work samples

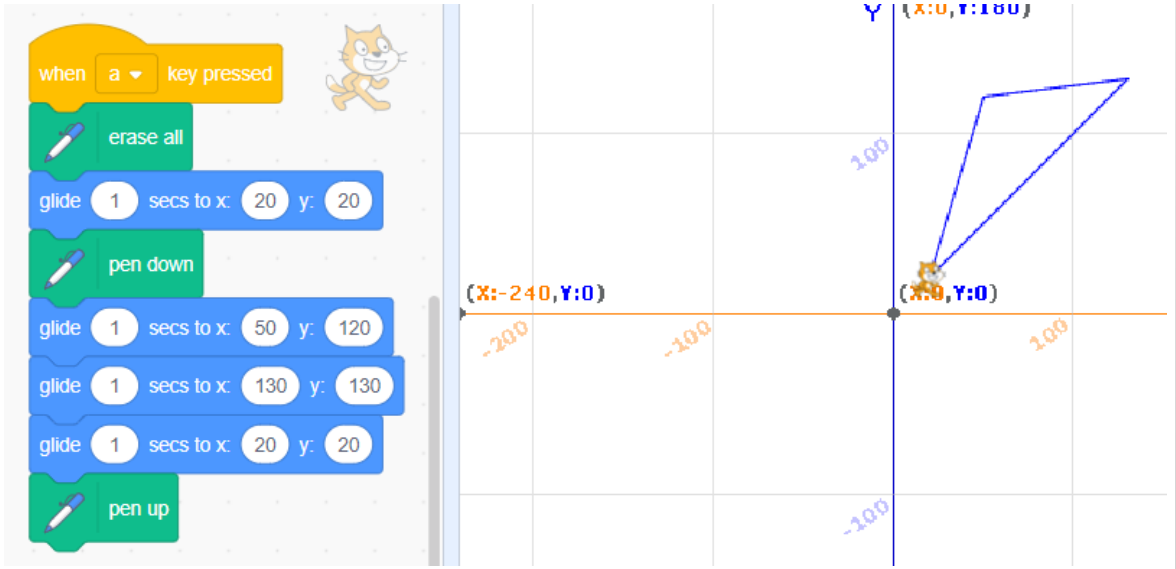


Linking to coordinates

Algorithm



Code



code-it.co.uk

Linking to coordinates progression & resources

- Simple shapes 1st quadrant
- Complex shapes 1st quadrant
- Simple shapes all four quadrants
- Complex shapes all four quadrants
- Translation of shape using variables
- Enlargement of shapes using variable

- 1st Quad planner
- 4 Quad planner

Culturally relevant context

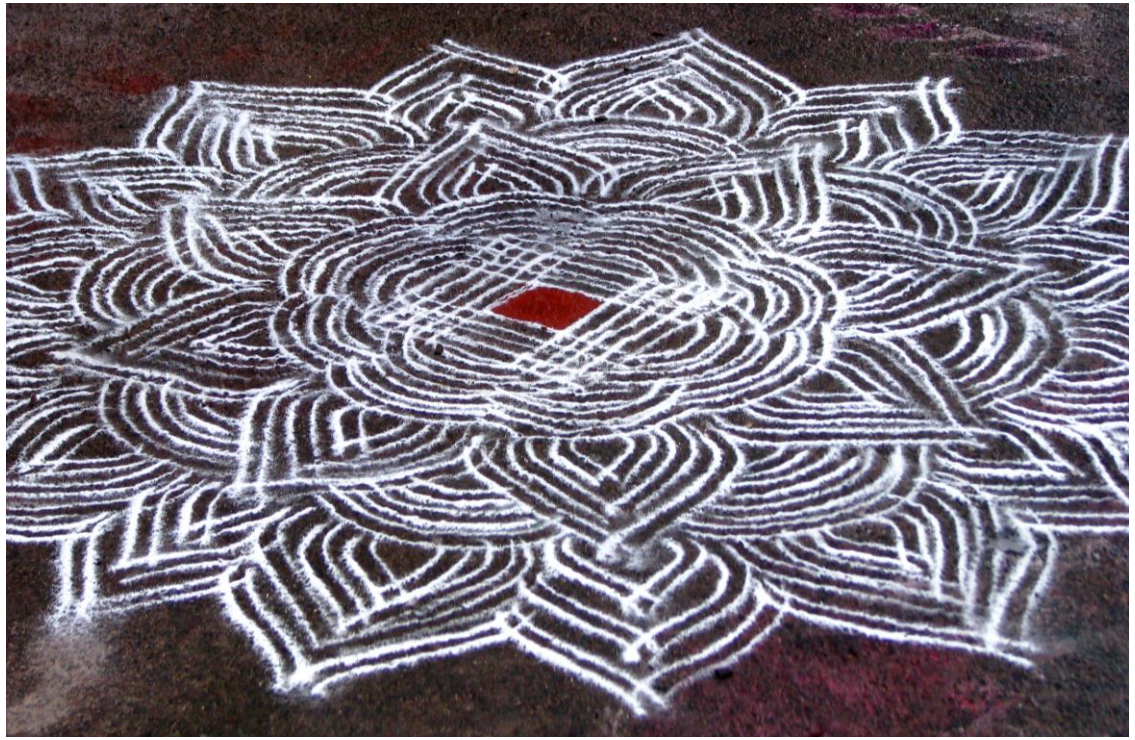
- Islamic Art

https://greenlane.bradford.sch.uk/wp-content/uploads/2020/06/22_06-Art-KS2.pdf

- Rangoli Patterns (work in progress)

- Celtic Patterns

- Indian Kolam rice pattern



code-it.co.uk

BREAK