Programming: passing trend or vital curriculum component

Phil Bagge

Computing Inspector/Advisor Hants

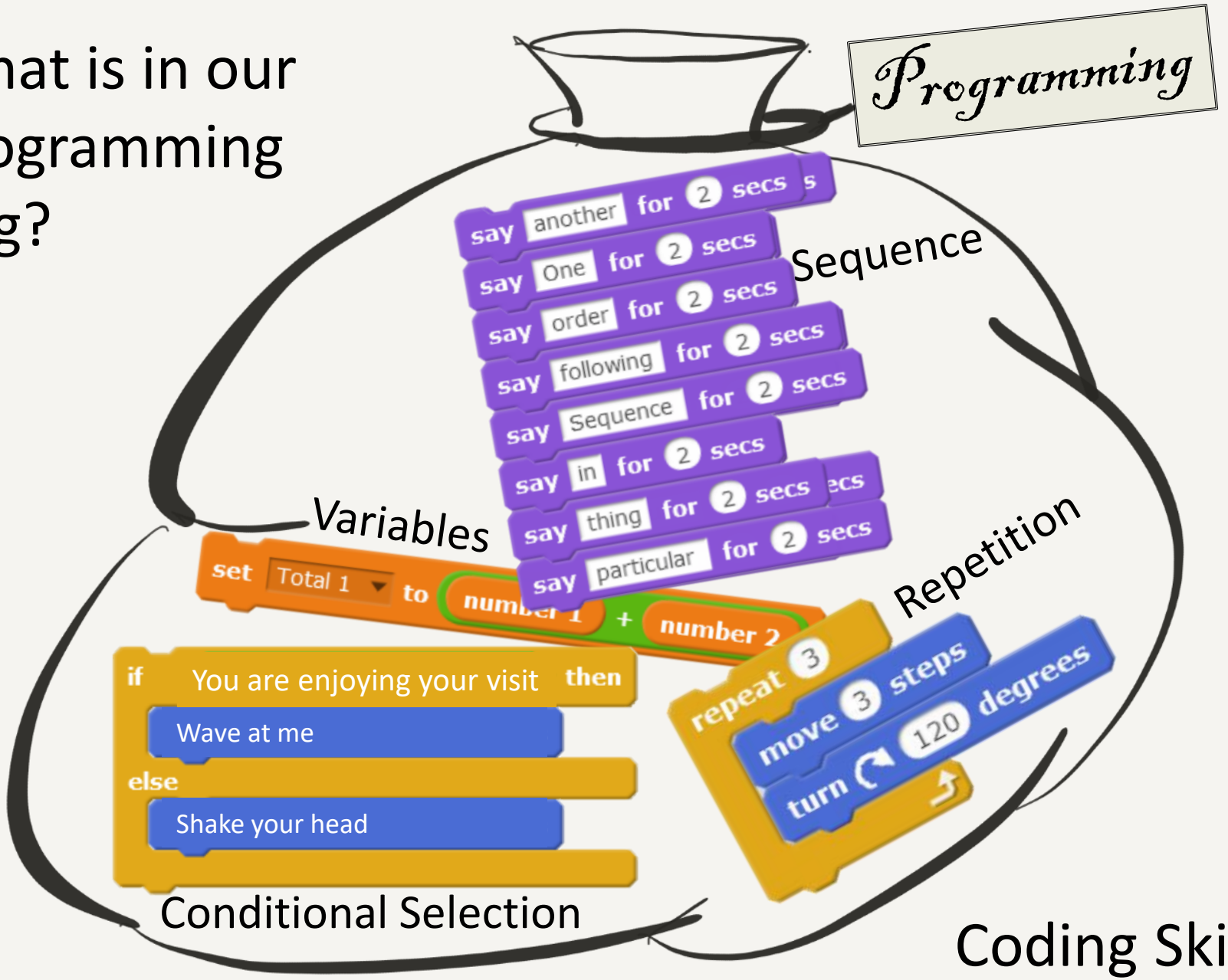
CAS Master Teacher

Teacher Ringwood Junior & Otterbourne Primary Schools

Author code-it.co.uk computing resources

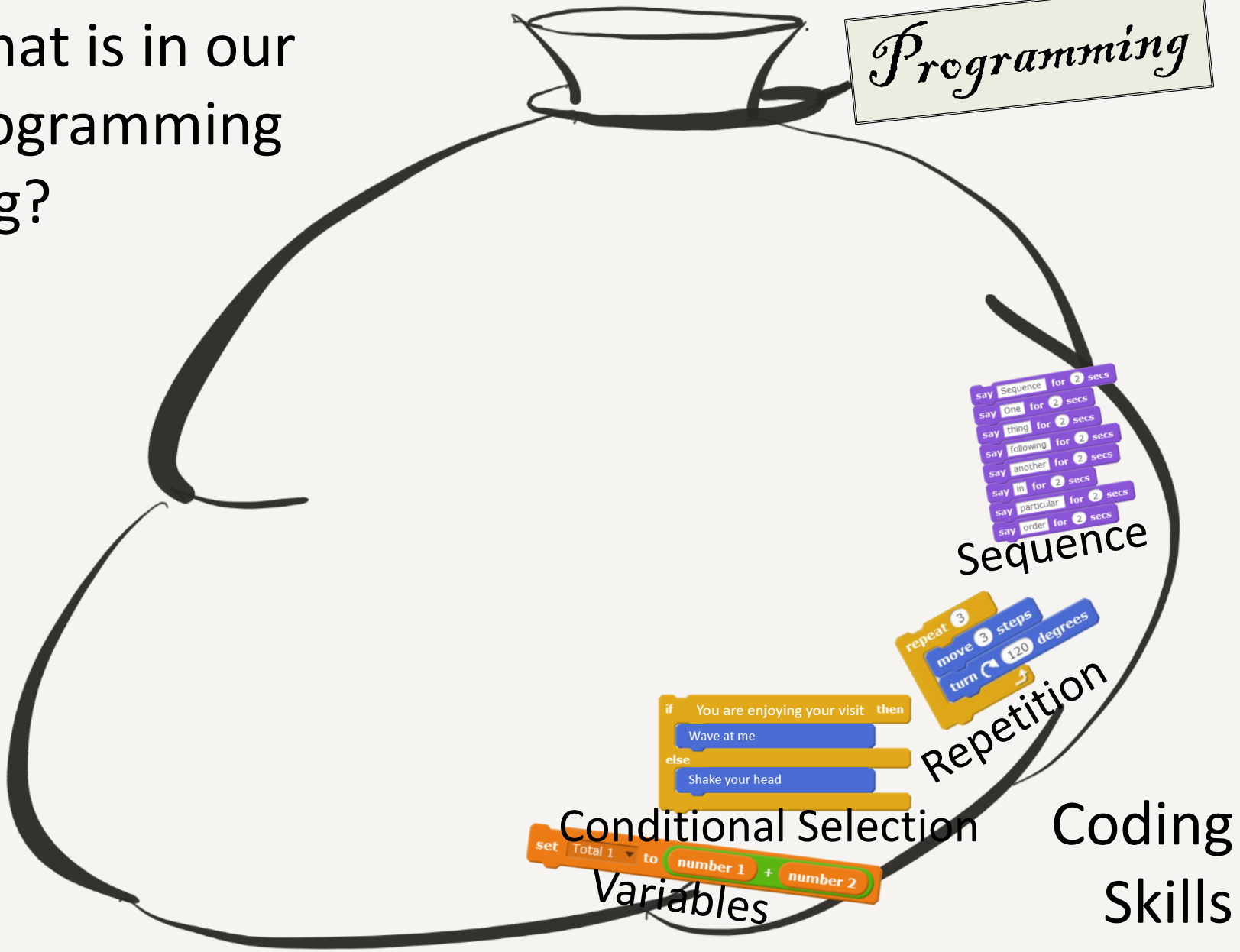
Author How to teach primary programming using Scratch

What is in our programming bag?



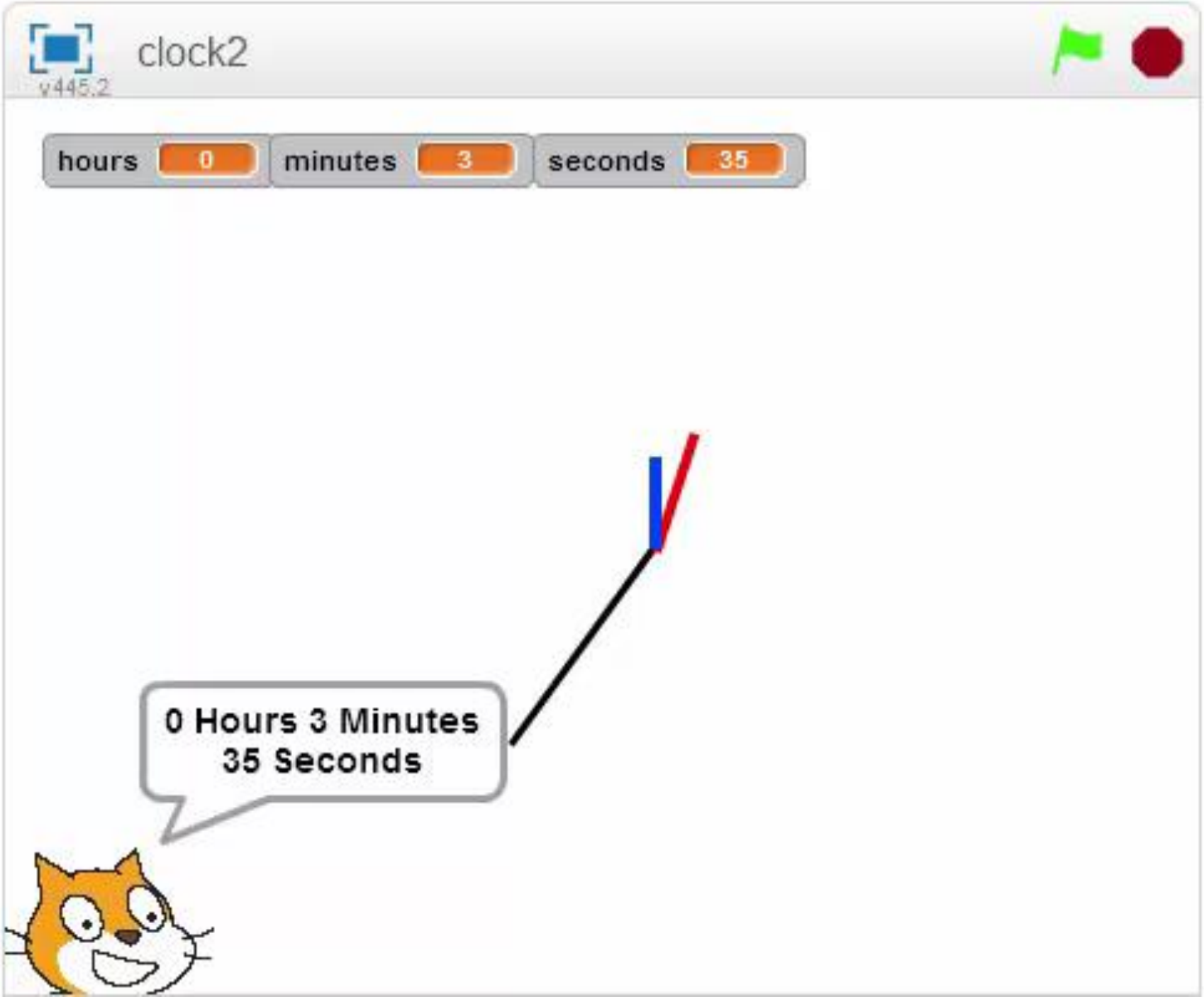
Coding Skills

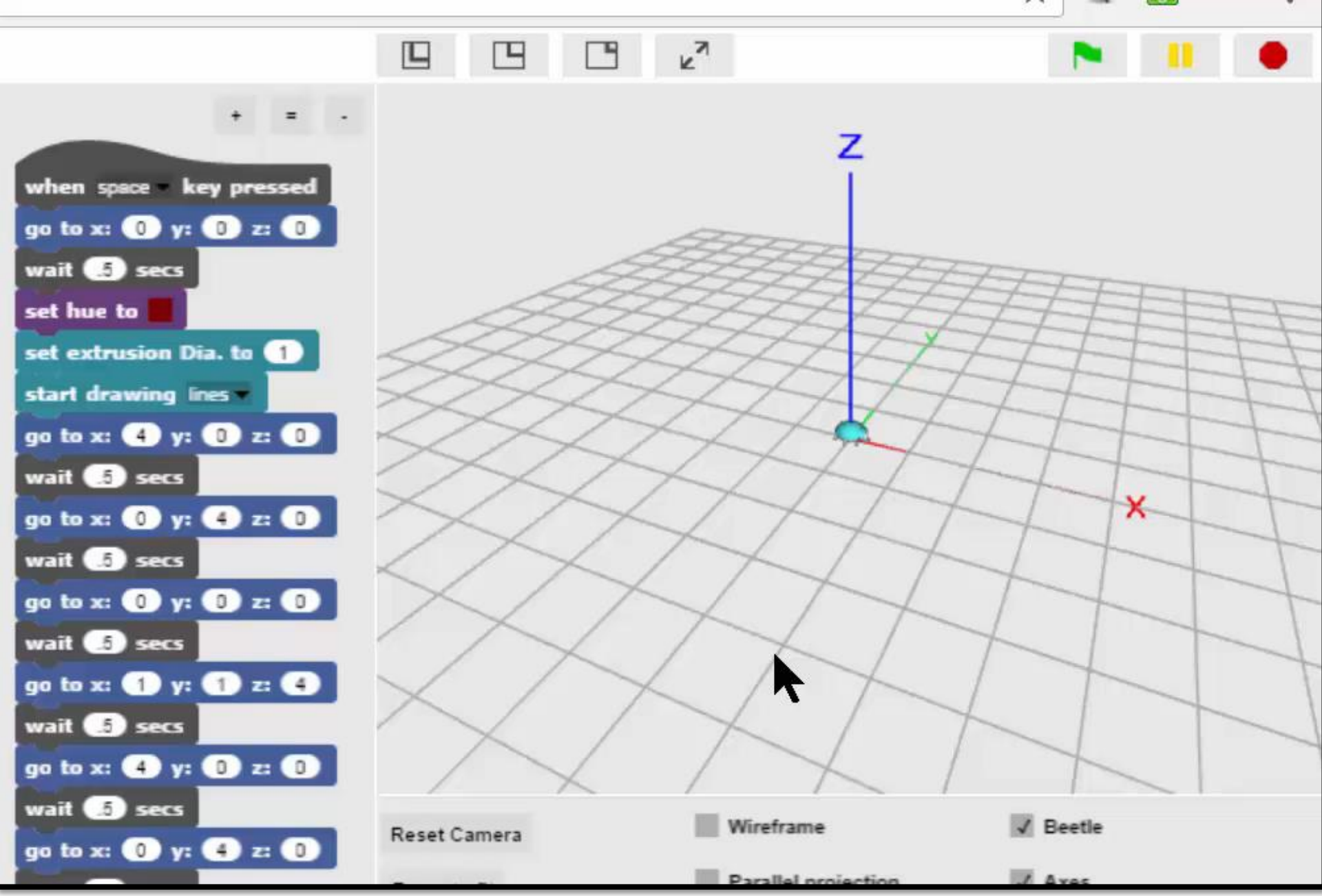
What is in our programming bag?

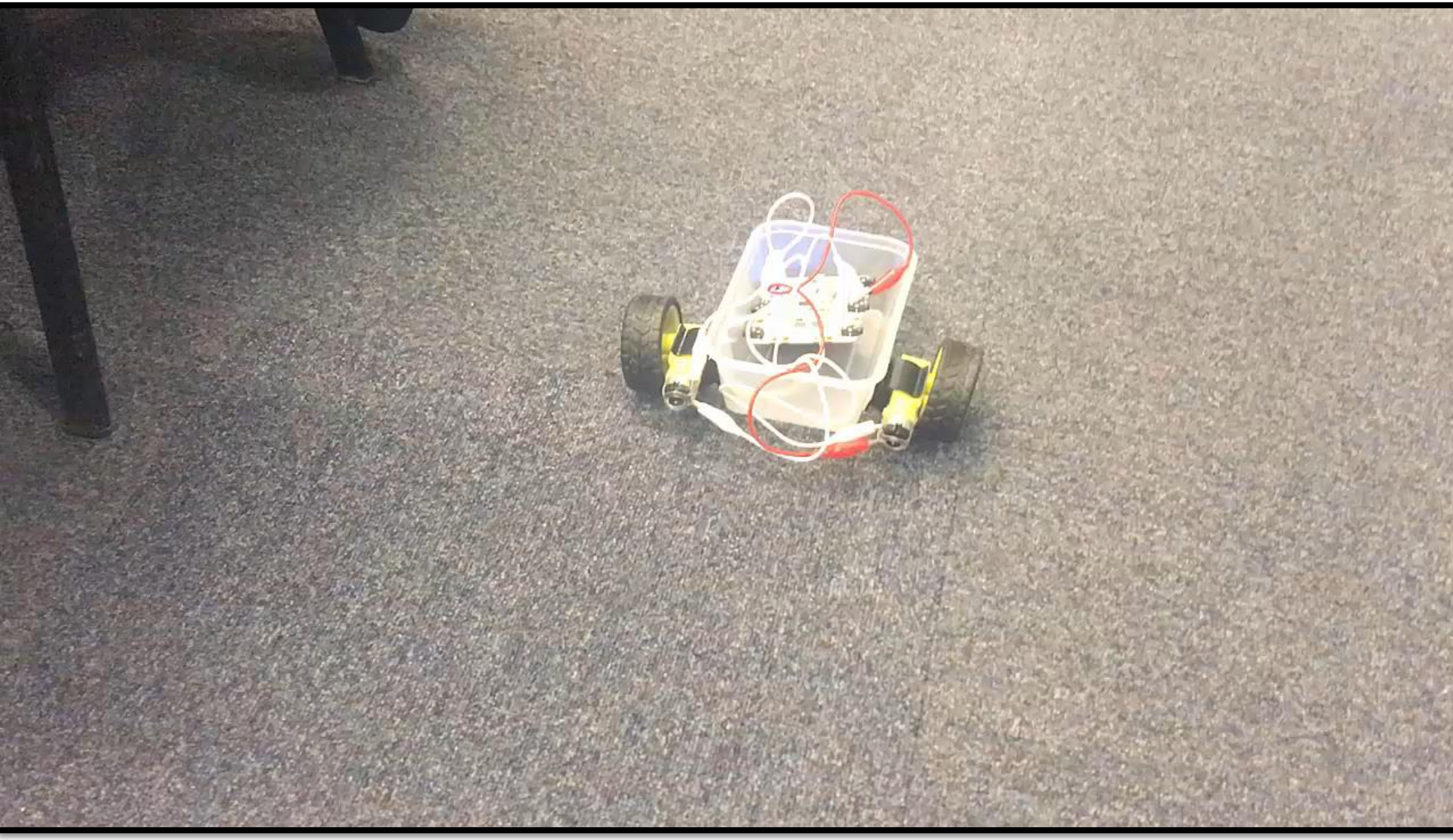


Purpose is intrinsic to programming

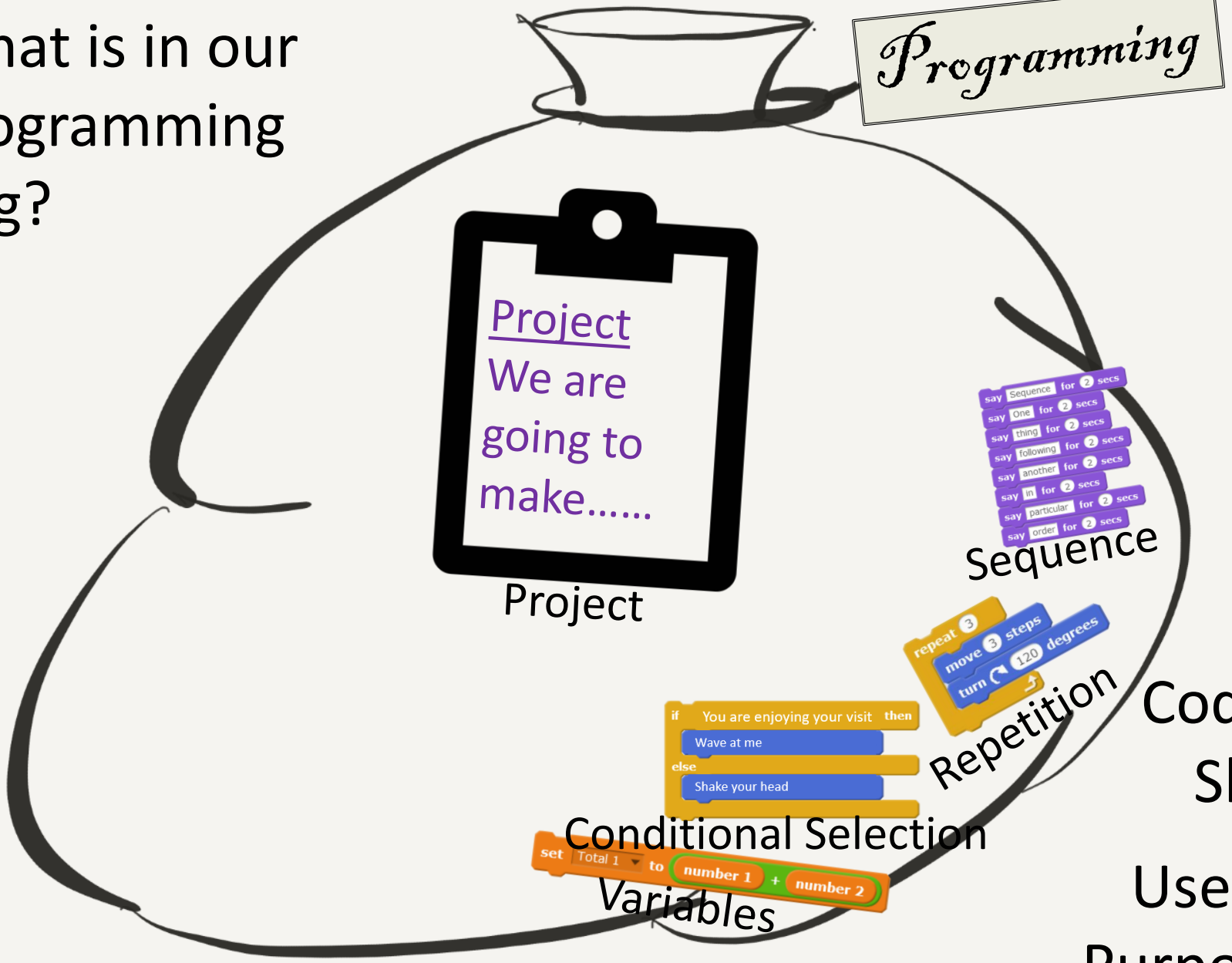
We program to make things that meet a need


clock2
v445.2
hours 0
minutes 3
seconds 35
0 Hours 3 Minutes
35 Seconds

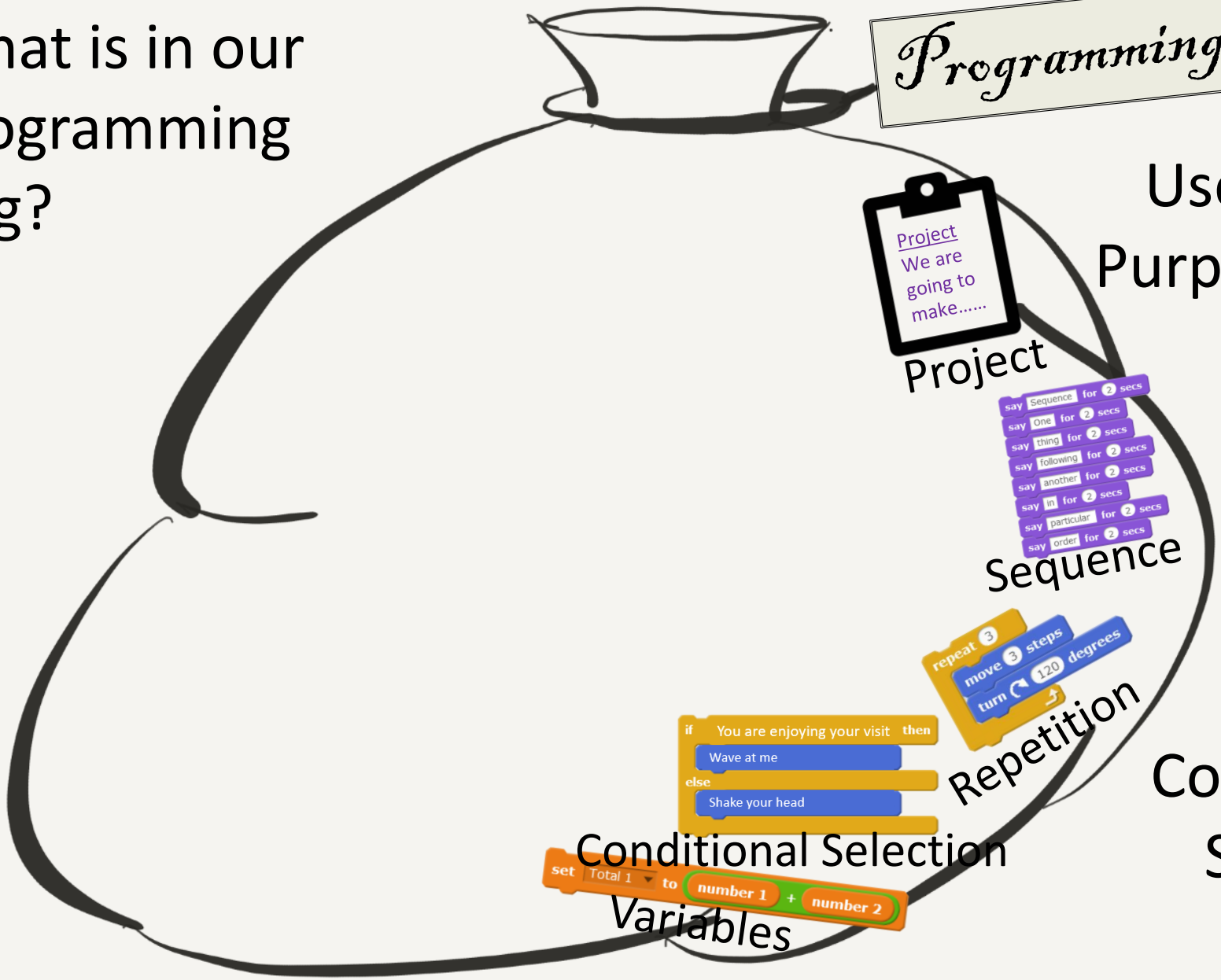

+
=
-
when space key pressed
go to x: 0 y: 0 z: 0
wait .5 secs
set hue to
set extrusion Dia. to 1
start drawing lines
go to x: 4 y: 0 z: 0
wait .5 secs
go to x: 0 y: 4 z: 0
wait .5 secs
go to x: 0 y: 0 z: 0
wait .5 secs
go to x: 1 y: 1 z: 4
wait .5 secs
go to x: 4 y: 0 z: 0
wait .5 secs
go to x: 0 y: 4 z: 0
Z
Y
X
Reset Camera
Wireframe
Beetle
Parallel projection
Axes



What is in our programming bag?



What is in our programming bag?



Programming = Algorithm + Coding

Berry

Programming = Computational Thinking + Coding

Woollard

Phil Bagge
@Baggiepr

adding two variables would be coding, adding them to solve a maths problem would be programming

Twitter

Develop Pupils into Computational Thinkers

- Algorithmic thinkers
- Algorithmic evaluators
- Decomposers
- Abstractors
- Generalisers

Think through steps or rules to achieve something

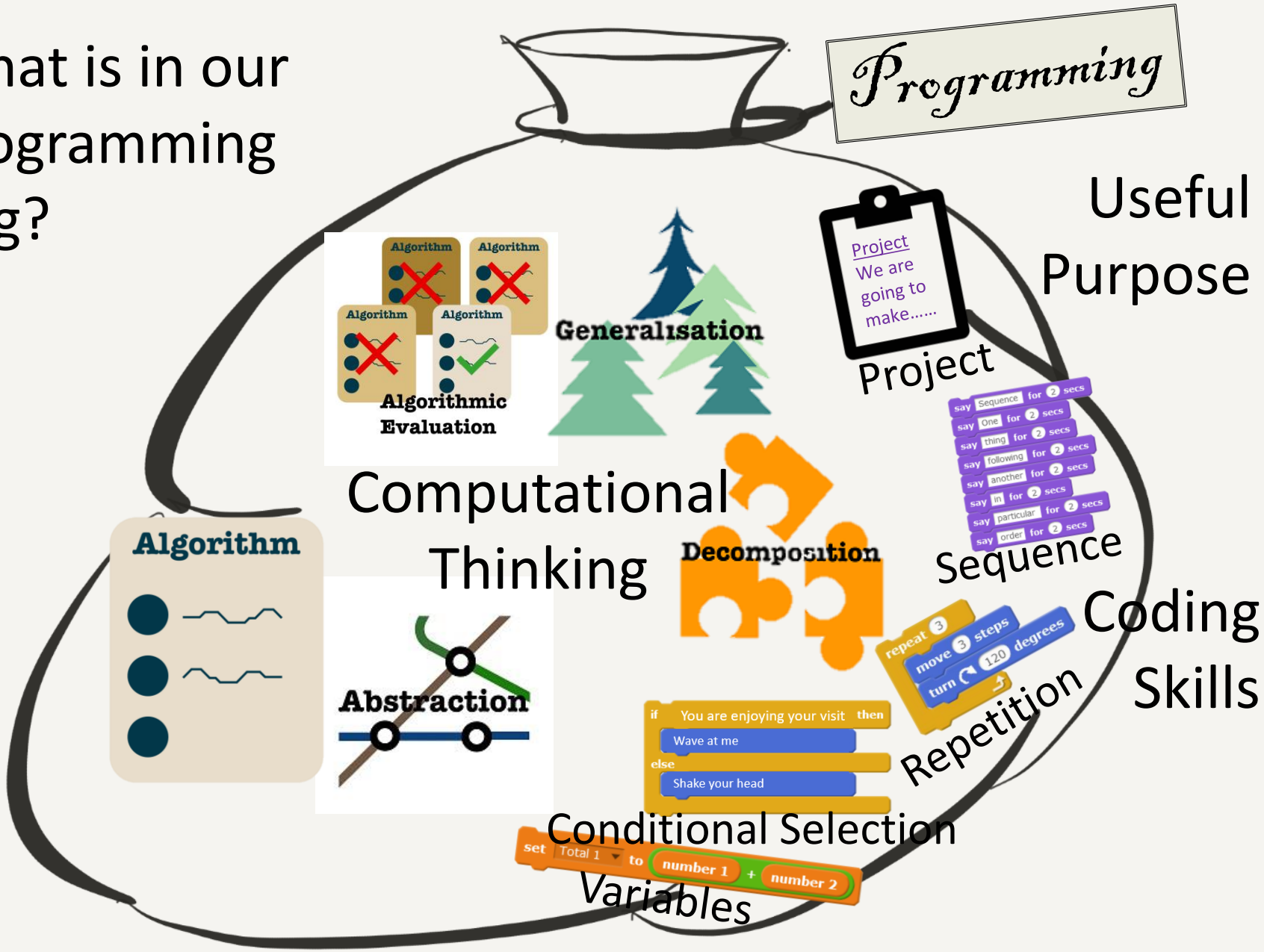
Which algorithm is best, most efficient?

Break problem up into parts and solve parts separately

Find the most important part of problem

Adapt solution from one problem to solve something else

What is in our programming bag?



Computational Thinking

- Algorithmic thinkers
- Algorithmic evaluators
- Decomposers
- Abstractors
- Generalisers

Really useful but approach can be too narrow without a focus on behaviours / attitudes as well

Chris Stephenson and Joyce Malyn-Smith from Google suggest a focus on **dispositions**

I recognise there is more than one way to solve/describe a problem

I don't just accept the first solution

I look for a range of solution to the same problem

I can evaluate my solutions against a set criteria

Handles Ambiguity

Open Ended Problem Solver

I look for how a project can be extended

I can design criteria to evaluate my creations

I can break complex problems into parts

I can contribute useful ideas to a partner or group

Evaluates

Computing Problem Solver

Copes with Complexity

I can discover / concentrate on the most important part of a problem

I can encourage others to share their ideas

Communicates

Adapts

I can identify patterns in problems & solutions

I lead using all the people talent in my group

I can adapt existing ideas to solve new problems

I learn from setbacks and don't let them put me off

Investigates

I can develop, test and debug until a product is refined

I can persevere even if the solution is not obvious

Perseveres

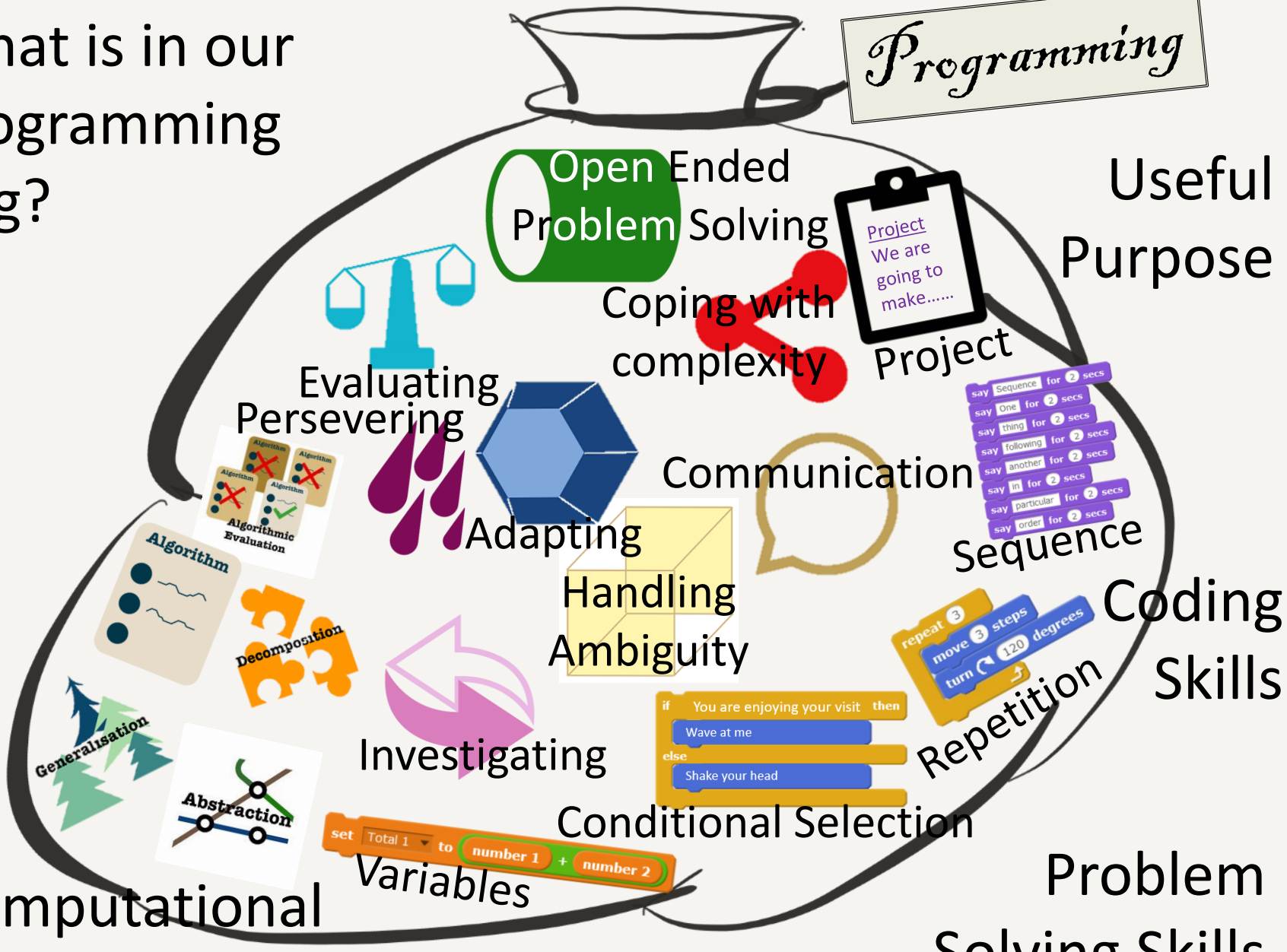
I repeatedly experiment through making, testing & debugging

Inspired by Behaviour Rubric created with @MarkDorling and linked at http://code-it.co.uk/attitudes/

Complexity	I can break complex problems into parts	I can discover / concentrate on the most important part of a problem	I can explain how I used decomposition & abstraction
Ambiguity	I recognise there is more than one way to solve a problem	I recognise there is more than one way to describe a problem	I can explain how I managed ambiguity
Open Ended	I look for a range of solution to the same problem	I don't just accept the first solution	I can describe how a project can be extended
Adapt	I can adapt existing ideas to solve new problems	I can identify patterns in problems & solutions	I can explain how I adapted a solution to solve a new problem
Evaluate	I can evaluate my solutions against a set criteria	I can design criteria to evaluate my creations	I can explain how evaluation helped me improve a project
Experiment & Debug	I can develop, test and debug until a product is refined	I repeatedly experiment through making, testing & debugging	I can explain how using the iterative cycle improves my work
Persistence	I can persevere even if the solution is not obvious	I learn from setbacks and don't let them put me off	I can describe how I overcame problems
Communicate	I can contribute useful ideas to a partner or group	I can encourage others to share their ideas	I can lead using all the people talent in my group

Decomposition Abstraction Generalisation Algorithmic Evaluation Algorithm

What is in our programming bag?



Anatomy of a programming project: Building a Times Tables Game

- **Aim**

Build your own version of a times tables game

- **Age**

10+

- **Computational Thinking Skills**

Algorithmic rules, Decomposition, Generalisation, Algorithmic Efficiency, Abstraction

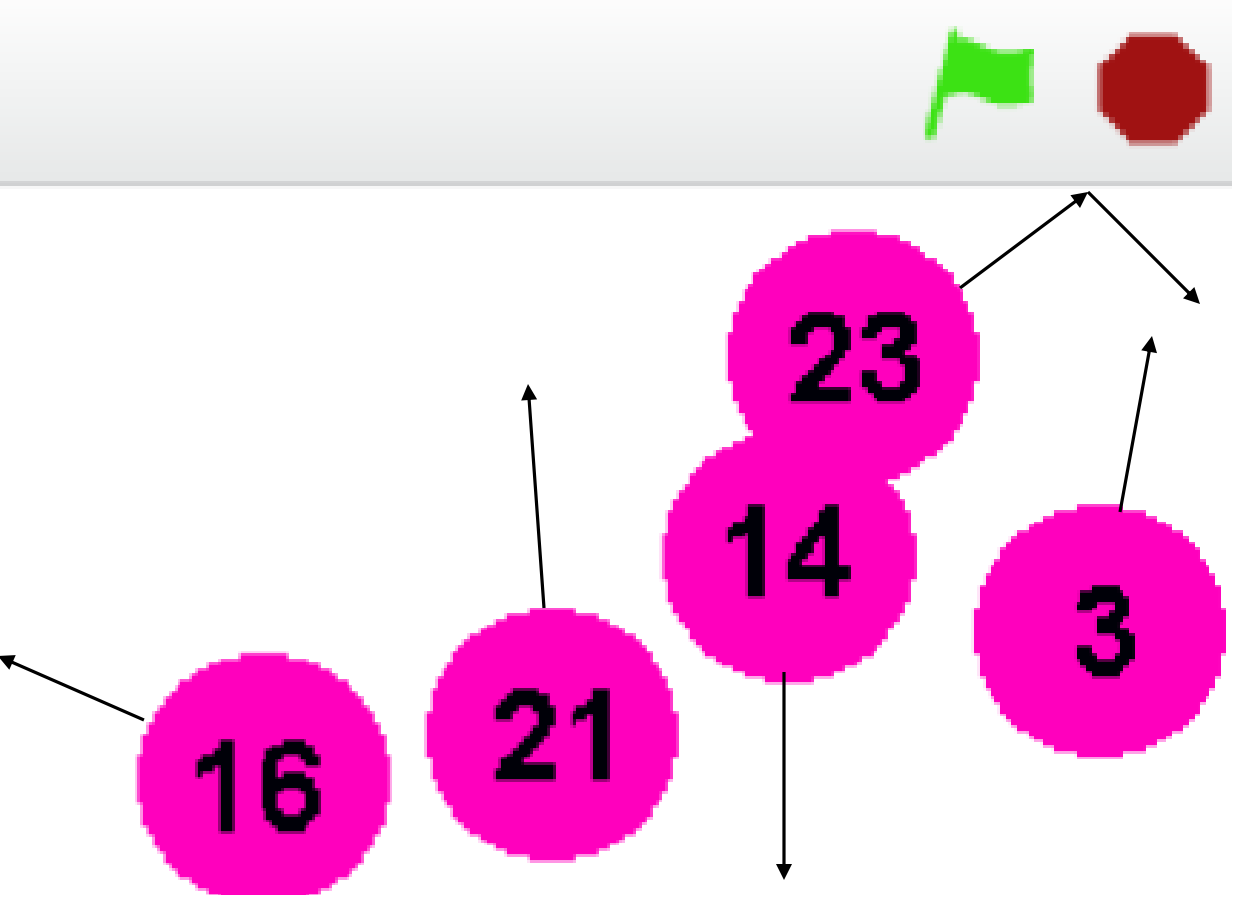
- **Programming Tools**

Sequence, iteration, selection, variables

Anatomy of a programming project: Building a Times Tables Game

Copes with Complexity

I can break complex problems into parts (decomposition)



What do all numbers do?

-Move
-Bounce off edge
-Start in random direction

I can break complex problems into parts (decomposition)

Copes with Complexity

u need to make?

r Sprits

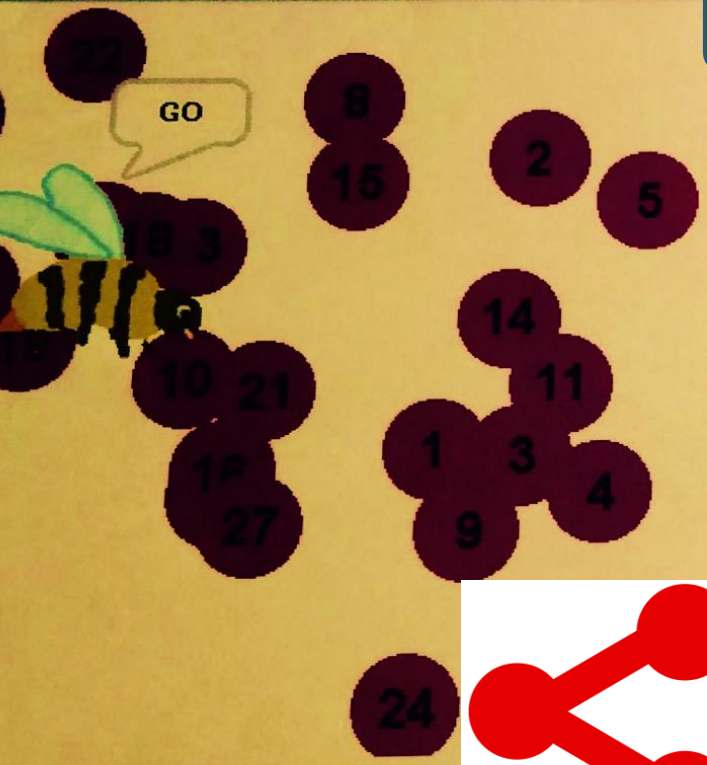
loch

eirt

more blok

Score block

Timer block



es Game Algorit.....

ophia

What do you need to make **all** the number sprites do?

You need to make sprits disappear when kicked. Make them move. Only number whi[illegible] are right to a point if [illegible] point. Say then disapper [illegible] back and tell score

Start Here

[illegible]eed to make **only the correct numbered** sprites do? Add a point when they kick on a number in the 2x tables. subtrac a point each time they kick on a wrong.

What do you need to make **only the wrong numbered sprite** do? when the kick on a wrong answer subtract 1 point

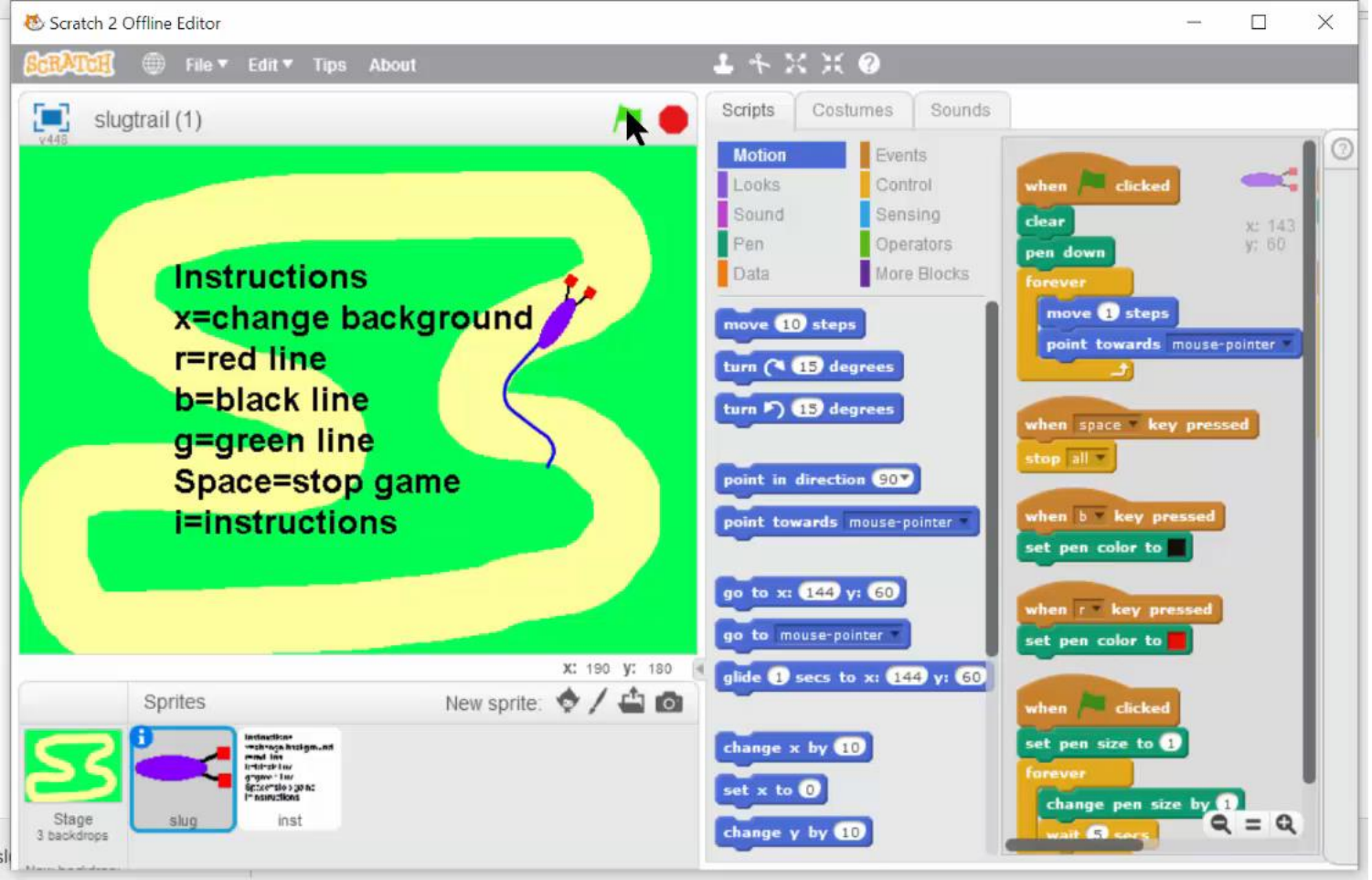
What do you need [illegible]

When you [illegible] because of [illegible]

[illegible] says wh[illegible] reappe[illegible] points

Copes with Complexity

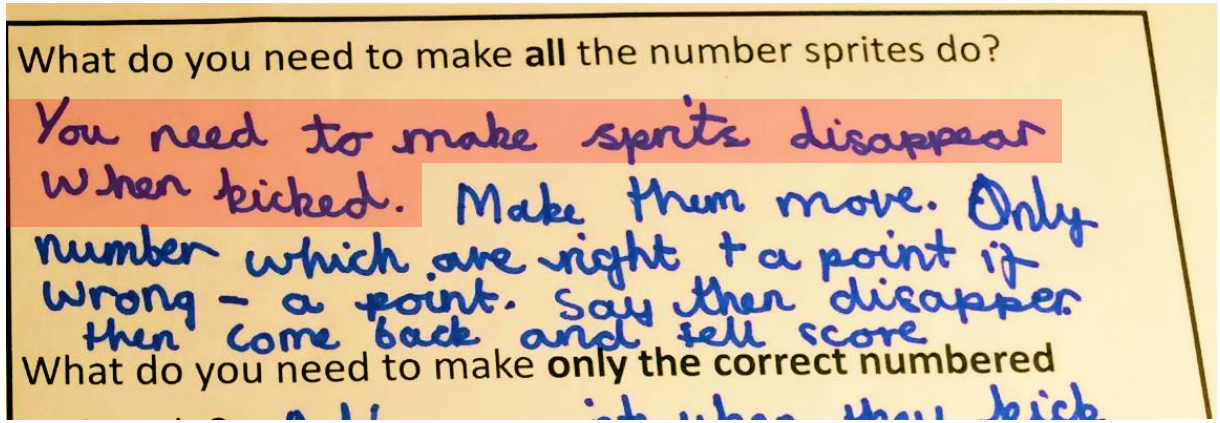
I can concentrate on the most important part of a problem



Adapts

Adapting an idea from previous programming

I can adapt existing ideas to solve new problems



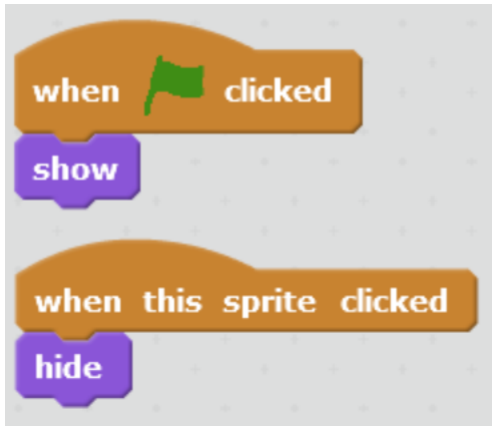
Debugging

16

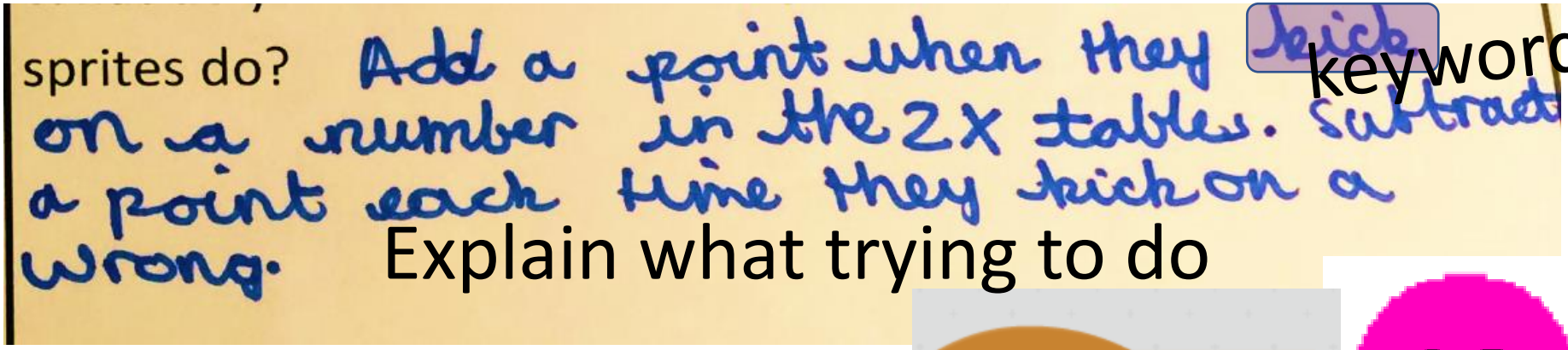
Making

Testing

Investigates

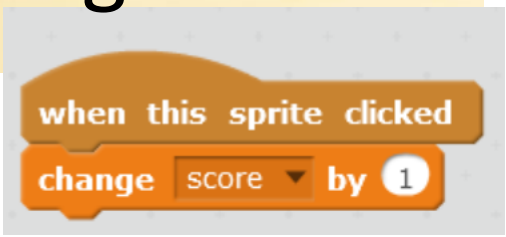


I repeatedly experiment through making, testing & debugging



keyword

Explain what trying to do



16

21

Hint

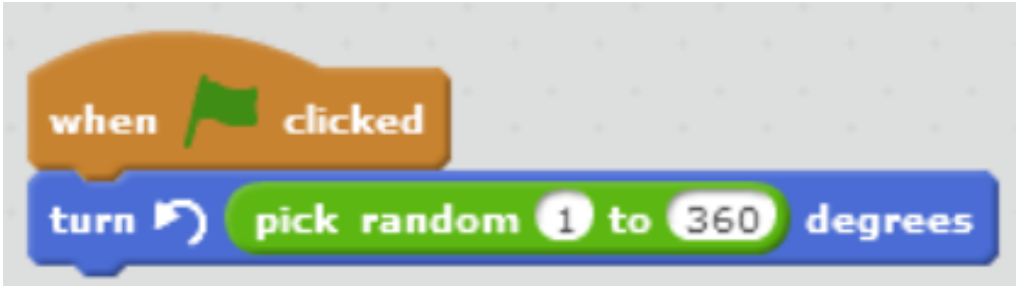
Perseveres

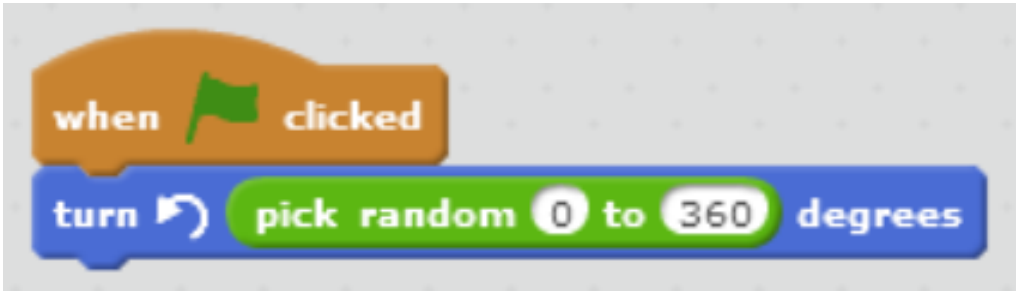
I can persevere when solving difficult problems even if the solution is not obvious

I learn from setbacks and don't let them put me off from solving difficult problems

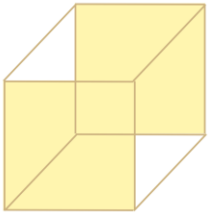
File
Edit
Tips
About
Scripts
Costumes
Sounds
Motion
Looks
Sound
Pen
Data
Events
Control
Sensing
Operators
More Blocks
when clicked
show
hide
switch costume to penguin1
next costume
switch backdrop to background1
change color effect by 25
set color effect to 0
clear graphic effects
change size by 10
set size to 100 %
go to front
when up arrow key pressed
set points to 0
say Hello! for 2 secs
say are you ready for a times tables quiz for 4 secs
say you have 20 seconds for 2 secs
say ready for 2 secs
say set for 2 secs
say go for 1 secs
say nice try, now try and do it quicker! for 2 secs
New sprite
Sprites
Sprite27
Sprite28

Starting in a Random Direction

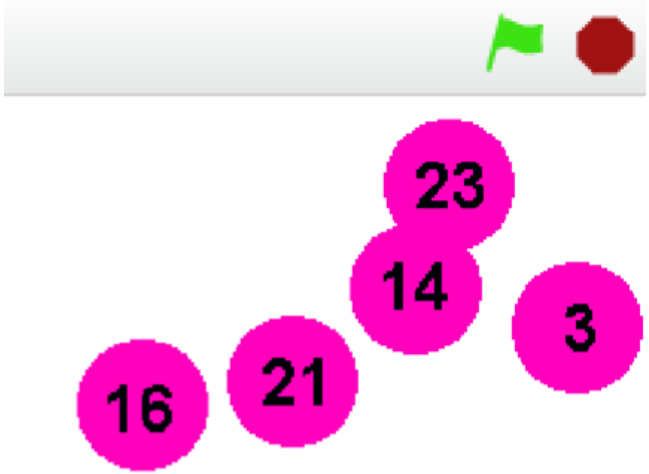




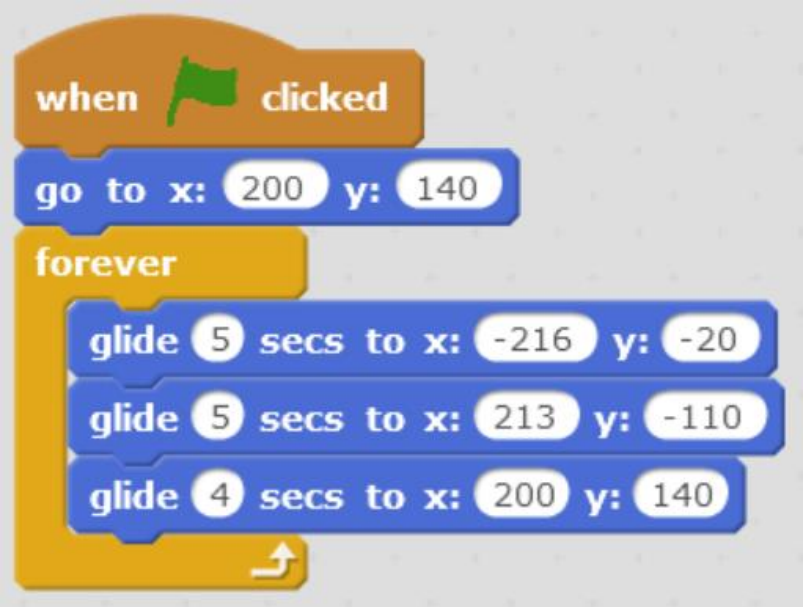
Recognising that the there are a range of different ways to solve same problem

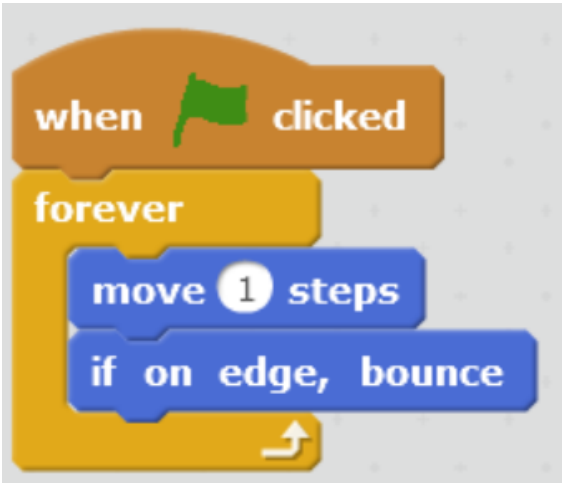


Handles Ambiguity



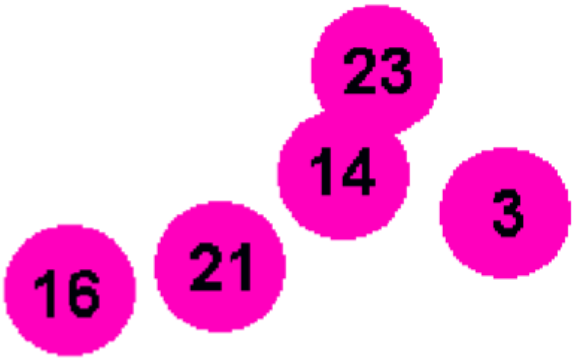
I recognise that there is often more than one way to solve a problem

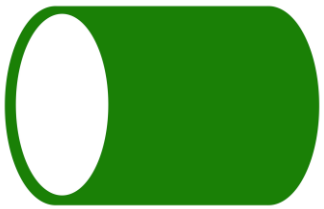




Actively looking for more than one way to solve the problem

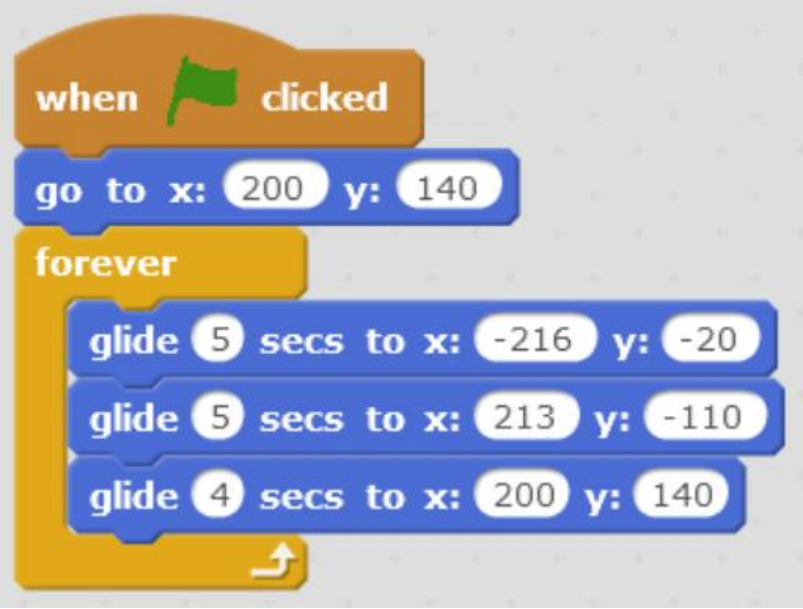
Moving and bouncing off the edge

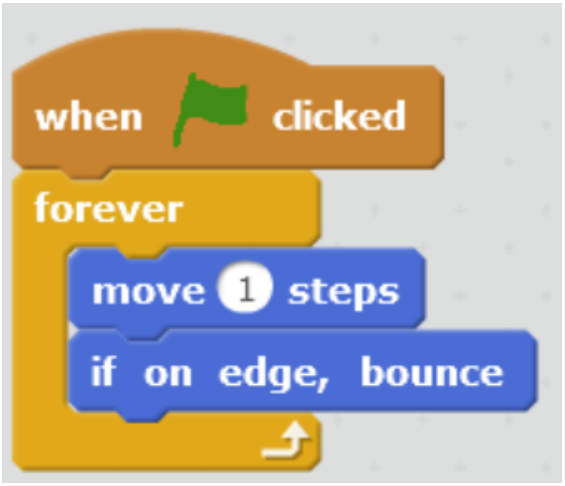




Open Ended Problem Solver

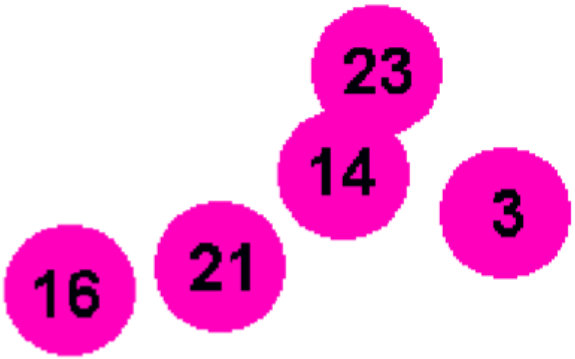
I look for a range of solutions to the same problem





Which method of solving the problem is best?

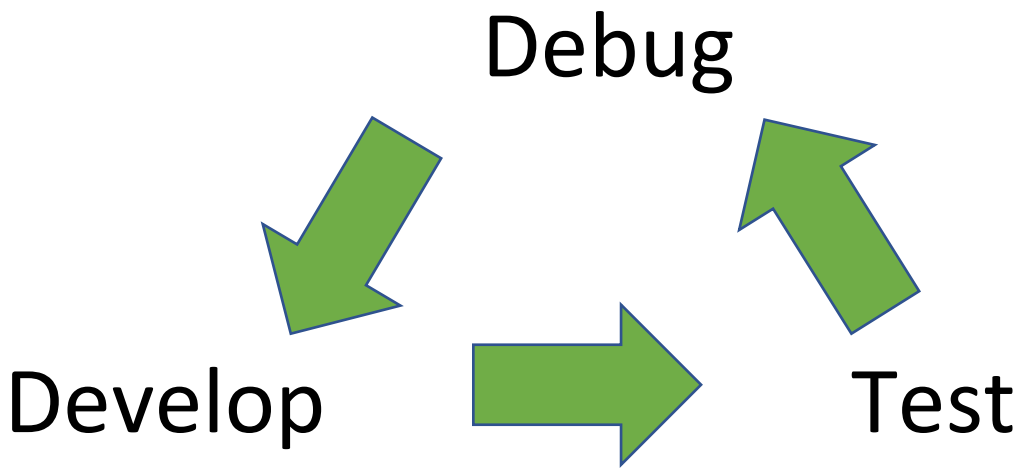
Moving and bouncing off the edge



Evaluates

I can design criteria to evaluate my creations

Refining for a great user experience



16

Investigates

I can develop, debug and test more than once until a product is refined

What makes for a good user experience?

- Re-play value
- Increased Challenge
- Few bugs
- Fun factor

Evaluates

Working with class to define the criteria

I can evaluate my solutions against set criteria

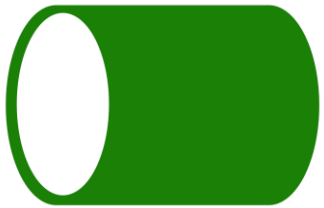
Where Next?

How can I make this better?

What can I develop out of this idea?

Common Next Step Ideas

Division tables
Multiple levels
Shooting the coins
Time limited coins
Coins that shrink over time



Open Ended Problem Solver

I can describe how a project could be extended

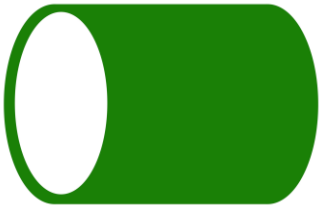
In one programming project that typically takes 4 hours

Computational Thinking

Copes with Complexity

Investigates

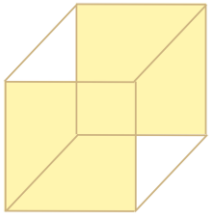
Abstraction
Generalisation
Decomposition
Algorithm Creator
Algorithm Evaluator



Open Ended Problem Solver

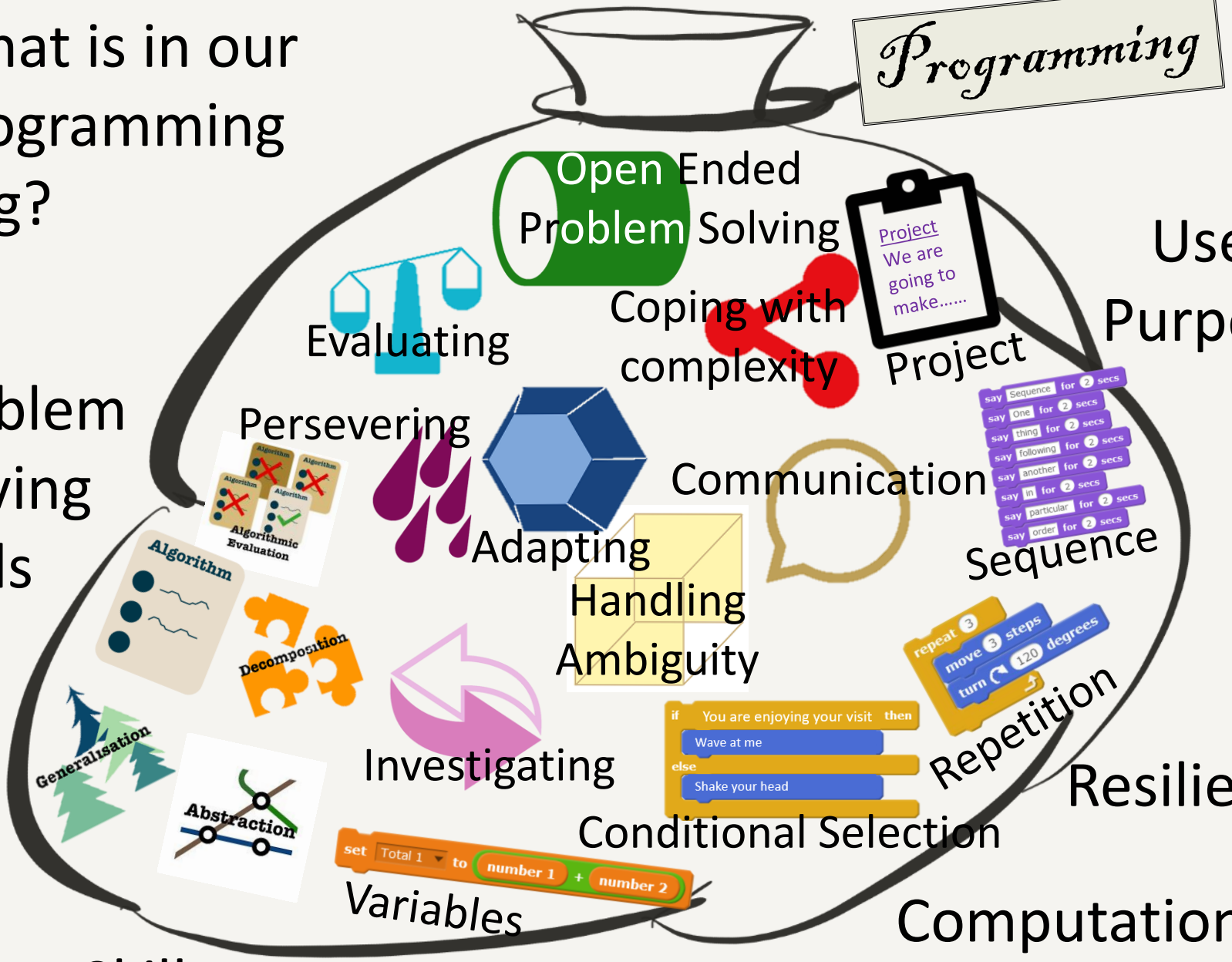
Evaluates

Handles Ambiguity



Perseveres

What is in our programming bag?



Programming is a vital curriculum component if we make the most of wider problem solving attitudes, computational thinking, resilience & project learning