technoteach

technocamps

Llywodraeth Cymru
Welsh Government

Prifysgol Abertawe
Swansea University

Cardiff University
Prifysgol Caerdydd

Prifysgol Bangor University

Cardiff Metropolitan University | Prifysgol Metropolitan Caerdydd

University of South Wales
Prifysgol De Cymru

Cyngor Cyllido Addysg Uwch Cymru
Higher Education Funding Council for Wales
hefcw

Prifysgol Cymru Y Drindod Dewi Sant
University of Wales Trinity Saint David

Prifysgol Aberystwyth University

Prifysgol Glyndŵr Wrecsam | Wrexham Glyndŵr University

institute of CODING in wales
technocamps

Welcome!

CPD for Secondary Education

Introductions

Your lecturers for the course will be:

Daniel North
daniel.north@technocamps.com

Alex Southern
alex.southern@technocamps.com

Course Overview

The full course will be taught over 3 units:

Unit 1 – 10th Sep to 15th Oct

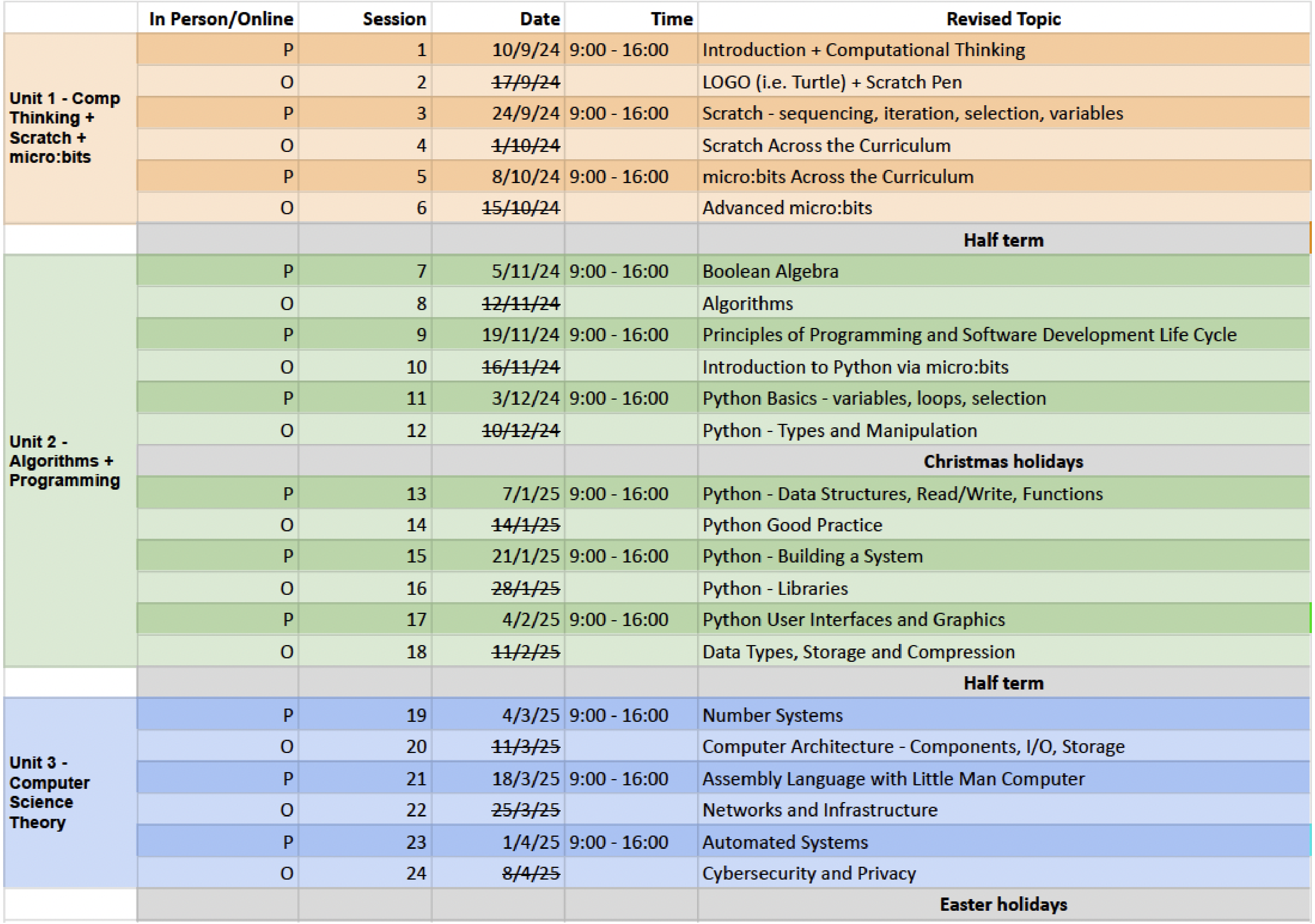
- Computational Thinking
- Block-Based Coding

Unit 2 – 5th Nov to 11th Feb

- Boolean Algebra
- Python Programming

Unit 3 – 4th Mar to 8th Apr

- Number Systems
- Assembly Language & Little Man Computer
- Networks & Infrastructure
- Cybersecurity



	In Person/Online	Session	Date	Time	Revised Topic
Unit 1 - Comp Thinking + Scratch + micro:bits	P	1	10/9/24	9:00 - 16:00	Introduction + Computational Thinking
	O	2	~~17/9/24~~		LOGO (i.e. Turtle) + Scratch Pen
	P	3	24/9/24	9:00 - 16:00	Scratch - sequencing, iteration, selection, variables
	O	4	~~1/10/24~~		Scratch Across the Curriculum
	P	5	8/10/24	9:00 - 16:00	micro:bits Across the Curriculum
	O	6	~~15/10/24~~		Advanced micro:bits
					Half term
Unit 2 - Algorithms + Programming	P	7	5/11/24	9:00 - 16:00	Boolean Algebra
	O	8	~~12/11/24~~		Algorithms
	P	9	19/11/24	9:00 - 16:00	Principles of Programming and Software Development Life Cycle
	O	10	~~16/11/24~~		Introduction to Python via micro:bits
	P	11	3/12/24	9:00 - 16:00	Python Basics - variables, loops, selection
	O	12	~~10/12/24~~		Python - Types and Manipulation
					Christmas holidays
	P	13	7/1/25	9:00 - 16:00	Python - Data Structures, Read/Write, Functions
	O	14	~~14/1/25~~		Python Good Practice
	P	15	21/1/25	9:00 - 16:00	Python - Building a System
	O	16	~~28/1/25~~		Python - Libraries
	P	17	4/2/25	9:00 - 16:00	Python User Interfaces and Graphics
	O	18	~~11/2/25~~		Data Types, Storage and Compression
					Half term
Unit 3 - Computer Science Theory	P	19	4/3/25	9:00 - 16:00	Number Systems
	O	20	~~11/3/25~~		Computer Architecture - Components, I/O, Storage
	P	21	18/3/25	9:00 - 16:00	Assembly Language with Little Man Computer
	O	22	~~25/3/25~~		Networks and Infrastructure
	P	23	1/4/25	9:00 - 16:00	Automated Systems
	O	24	~~8/4/25~~		Cybersecurity and Privacy
					Easter holidays

CPD Course Calendar (**NOT FINAL**)

Assessments

Each unit will be assessed via:

- 50% Exam
- 50% Assignment

Each unit will have exactly one exam and one assignment to be completed.

The exam and assignment for each unit will be set at the same time, towards the end of that unit. Each will have a 3-week deadline.

You require 40% in each unit to pass the course

Computational Thinking

You will need...

- A pen/pencil
- Paper

Computational Thinking

Computational thinking allows us to take a complex problem, understand what the problem is and develop possible solutions. We can then present these solutions in a way that a computer, a human, or both, can understand. – Bitesize

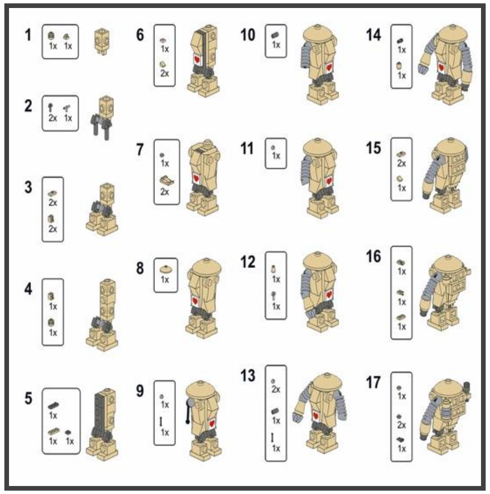
So basically it is working out how to solve a problem and then writing the solution in a way so the computer can do the hard work for you.

The Strands of Computational Thinking

Abstraction

Decomposition

Pattern Recognition



Algorithms

Abstraction

Activity: Get Arty!

Get your paper and pencils ready! You have 1 minute to draw the picture that will come up in the next slide.

Activity: Get Arty!

Artist Reflection

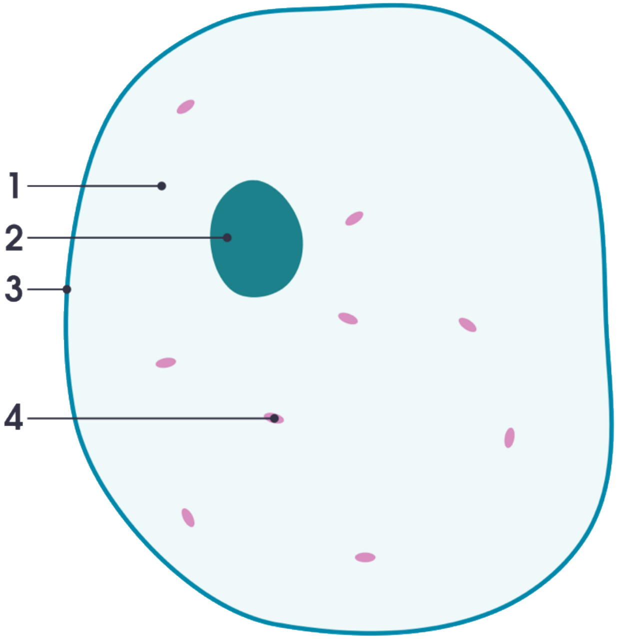
Abstraction

Abstraction is the process of removing unnecessary detail and simplifying. Abstraction is used to remove unnecessary detail from a real-world situation and to model the simplified result in an algorithm or program.

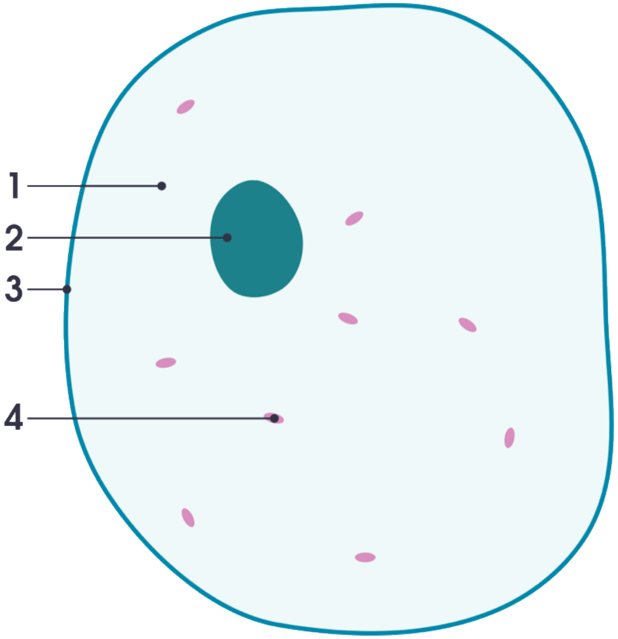
Real world examples of abstraction in action:

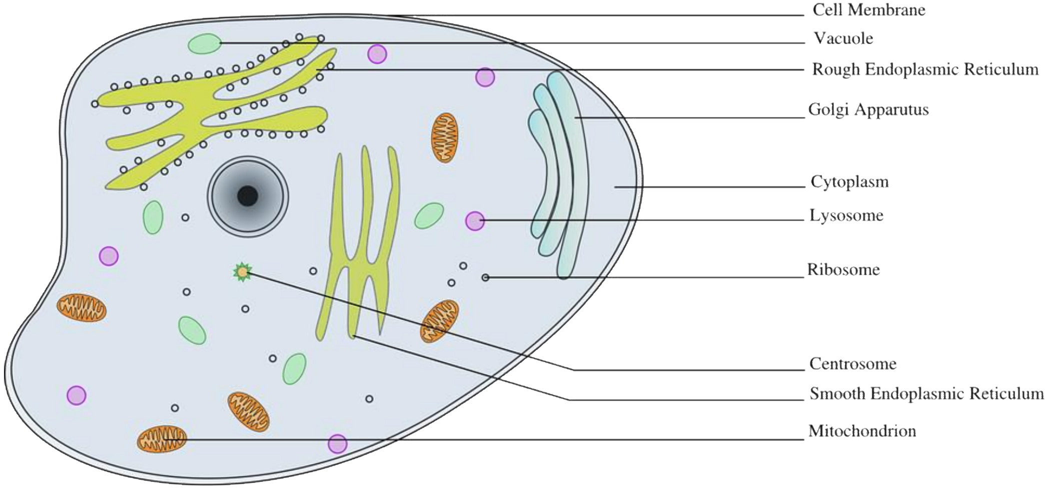
- In driving…
- In programming…
- In teaching…

Teachers using Abstraction in Biology



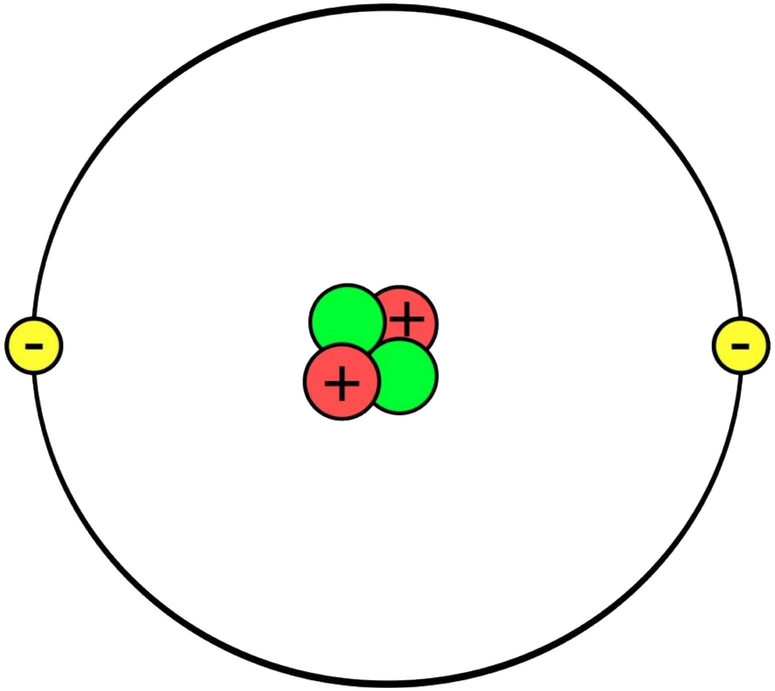
Teachers using Abstraction in Biology



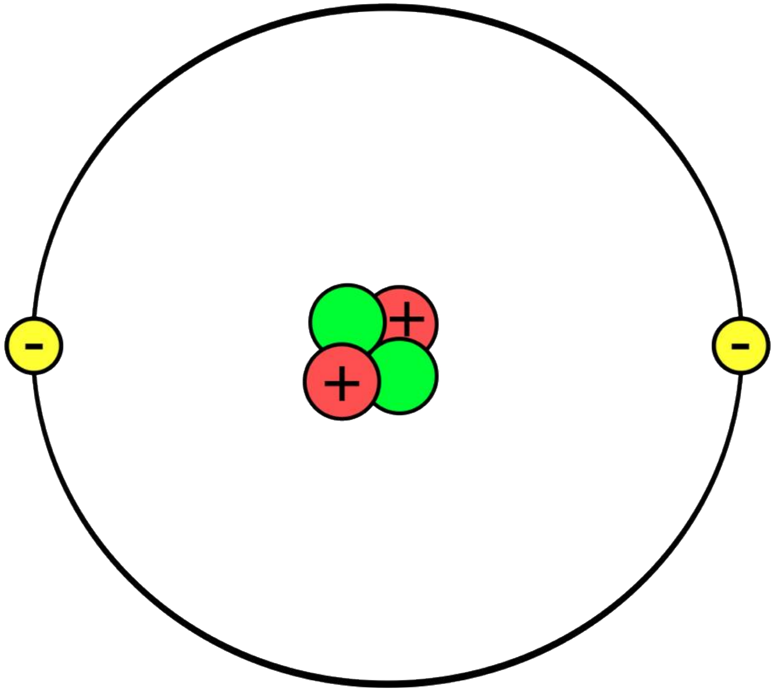


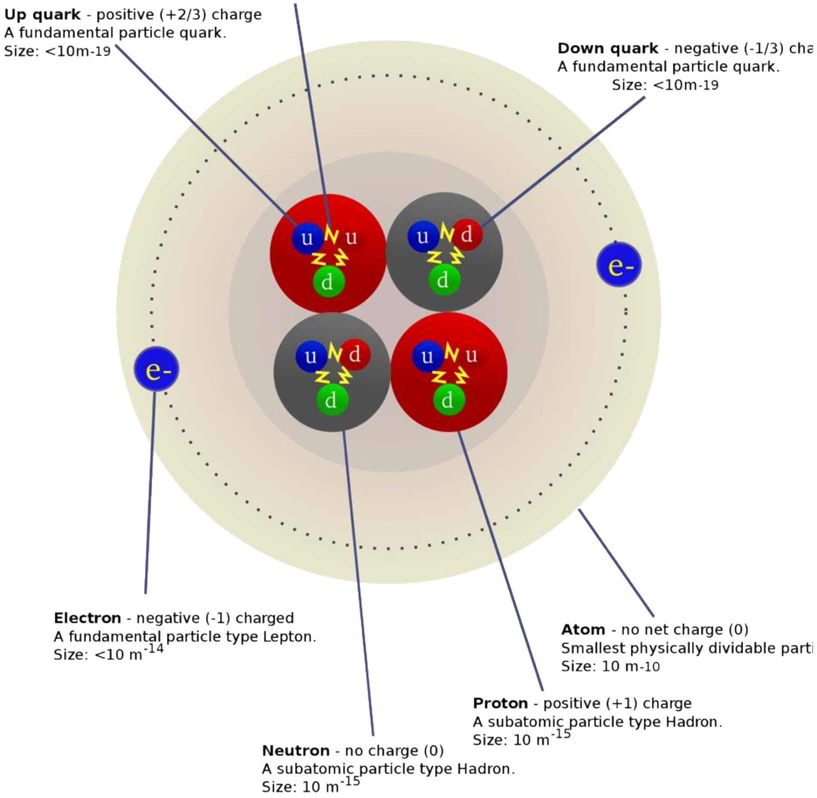
Cross Section of an Animal Cell

Teachers using Abstraction in Physics

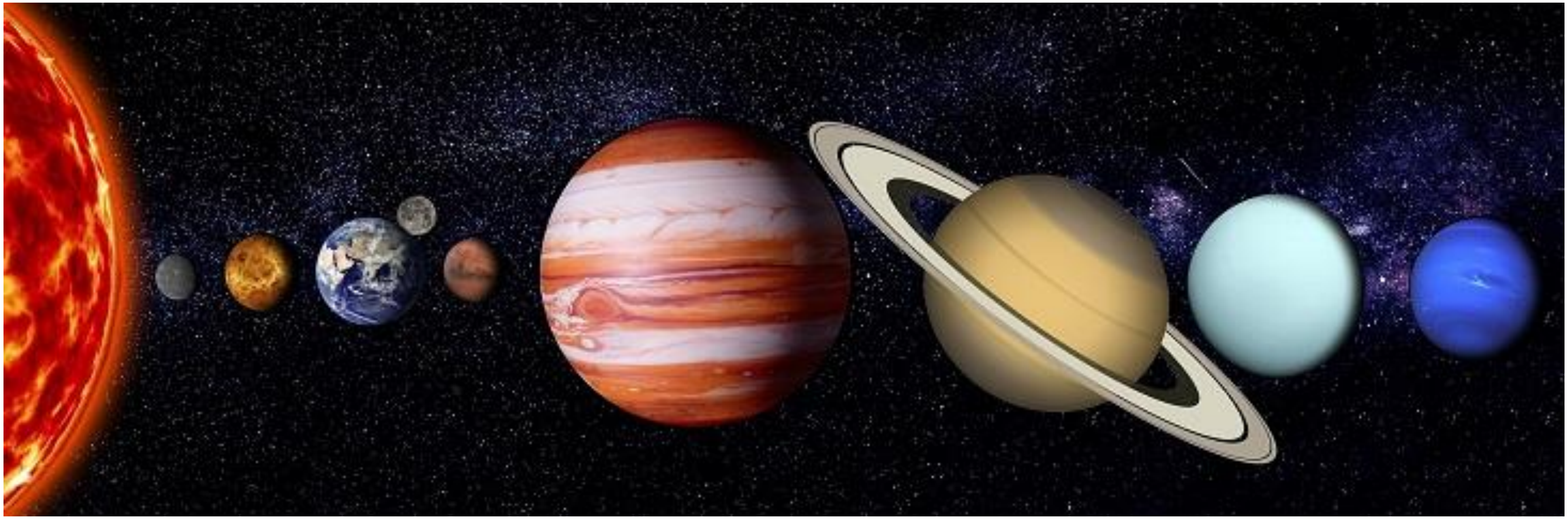


Teachers using Abstraction in Physics

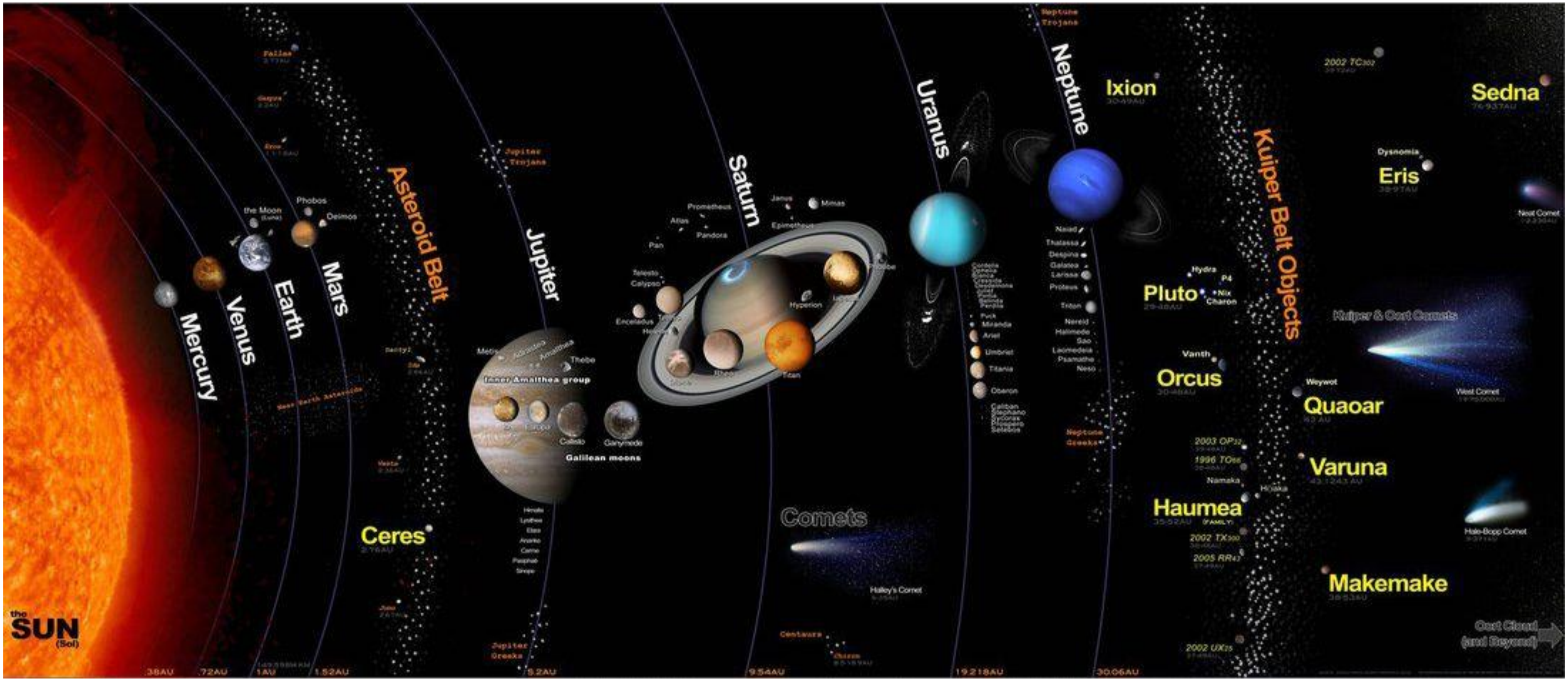



Up quark - positive (+2/3) charge
A fundamental particle quark.
Size: <10m-19
Down quark - negative (-1/3) cha
A fundamental particle quark.
Size: <10m-19
u
u
d
u
d
d
u
d
d
u
u
d
e-
e-
Electron - negative (-1) charged
A fundamental particle type Lepton.
Size: <10 m^{-14}
Atom - no net charge (0)
Smallest physically dividable parti
Size: 10 m-10
Proton - positive (+1) charge
A subatomic particle type Hadron.
Size: 10 m^{-15}
Neutron - no charge (0)
A subatomic particle type Hadron.
Size: 10 m^{-15}

Abstraction



Abstraction



Abstraction in Robotics

When simulating or modelling things, we don't always need to include all of the details:

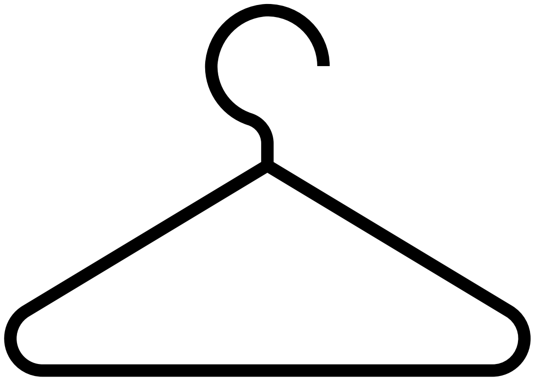
technoteach technocamps

Decomposition

Get Arty

Draw for me a triangle, attached to a quarter circle

Decomposition

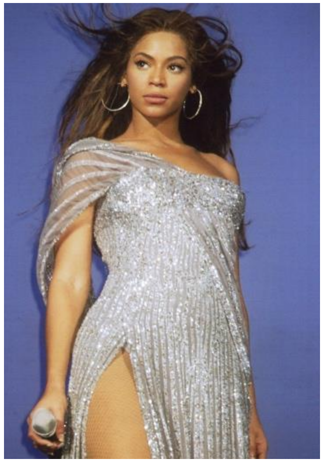


Decomposition

Activity: Guess Who

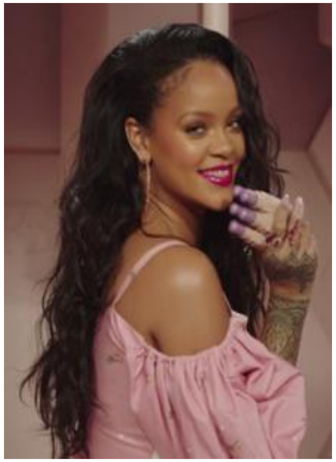
Cristiano Ronaldo

Beyoncé



Lionel Messi

Rihanna



Serena Williams

Drake

Kylie Jenner

Robert Downey Jr.

Guess Who Reflection

How many questions were needed?

Which questions were useful or not useful?

Did your partner's answers influence your next question? In what way?

Decomposition

Decomposition is the process of breaking a complex problem into smaller component parts.

Real world examples of using decomposition:

- Complex Maths Problems.
- Cooking.
- Cleaning your room!
- Creating a Game.

Example: Decomposition of a Game

When creating a game, what would we need to think about?

Think of your favourite game. How could you break it down into the important features? For example:

- What is the objective of the game?
- Who are the characters?
- What is the world like?
- Is it single or multi player?
- How do the characters interact?

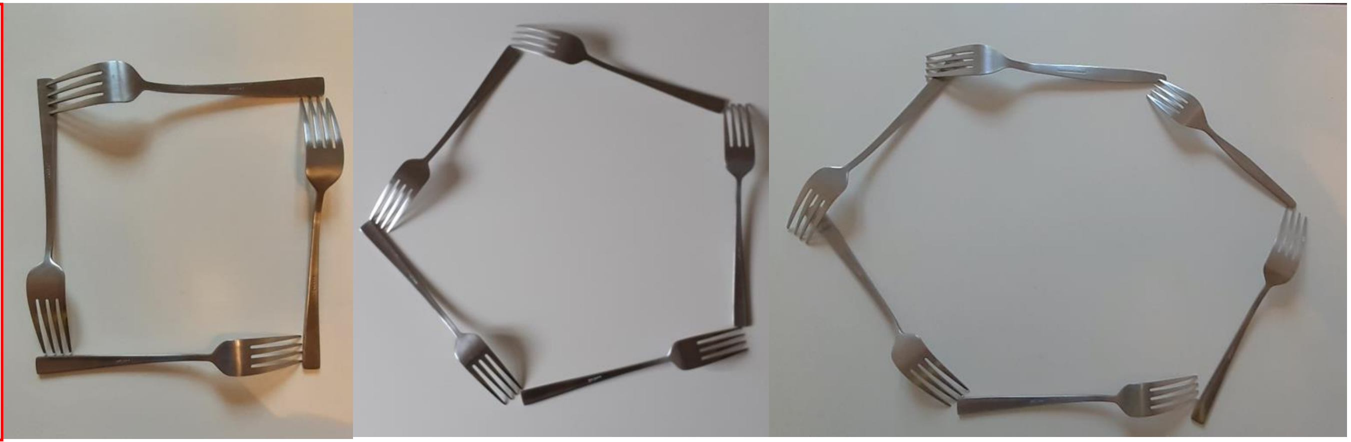
Activity: Guess the Game

Team 1 Points	Team 2 Points

Pattern Recognition

Activity: Complete the Pattern (1)

?



Activity: Complete the Pattern (2)

Complete the Pattern:

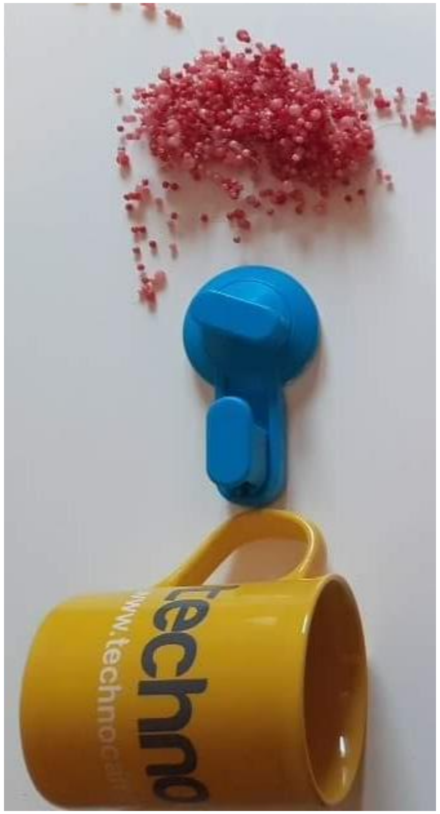
	🟥					

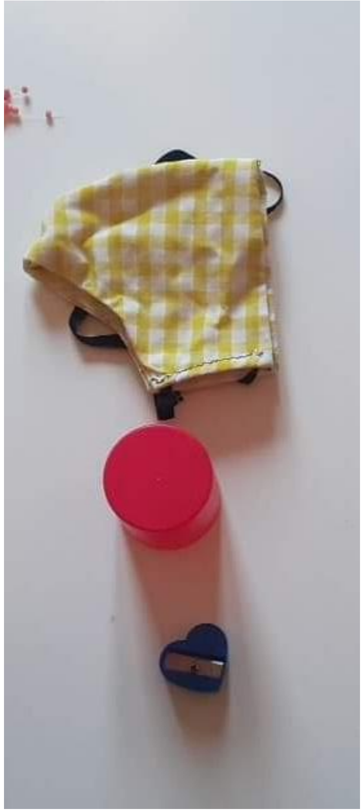
	🟥	🟥				
	🟥	🟥				

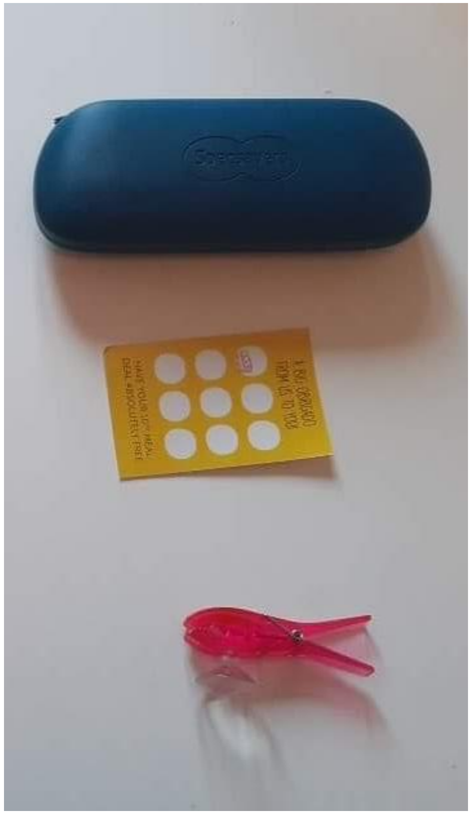
	🟥	🟥	🟥			
	🟥	🟥	🟥			
	🟥	🟥	🟥			

What would come next?

Activity: Complete the Pattern (3)







Activity: Number Sequences

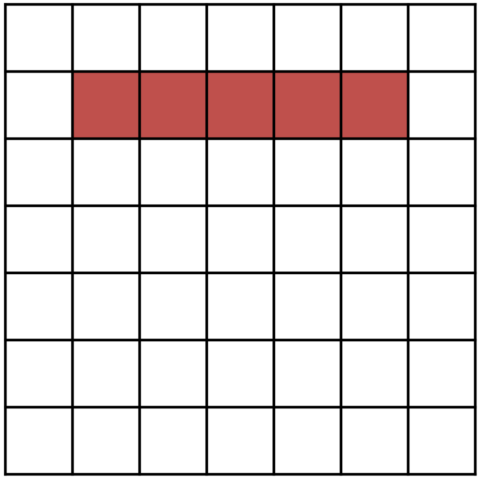
Can you spot the pattern in these sequences? Write the number that will come next:

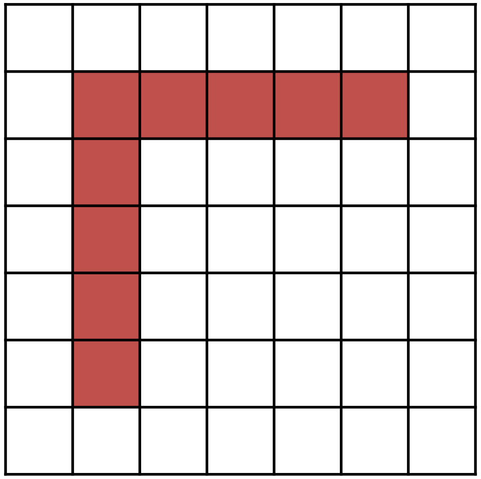
1, 2, 4, 7, 11, 16, 22, …

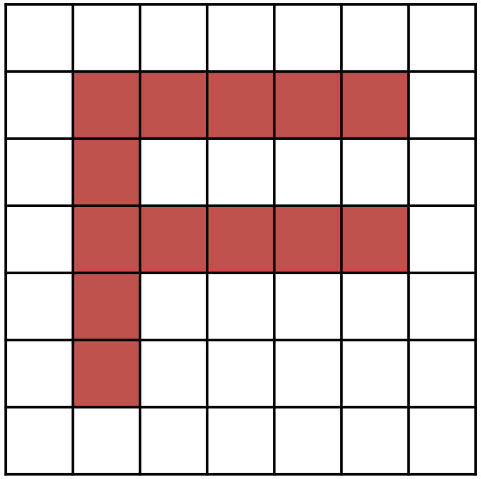
1, 1, 2, 3, 5, 8, 13, 21, 34, …

Activity: Complete the Pattern

Complete the Pattern:

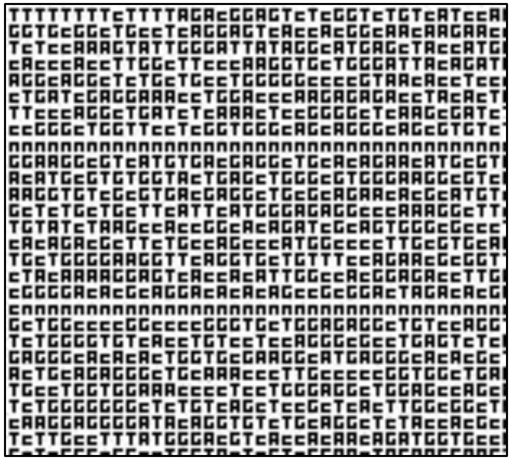






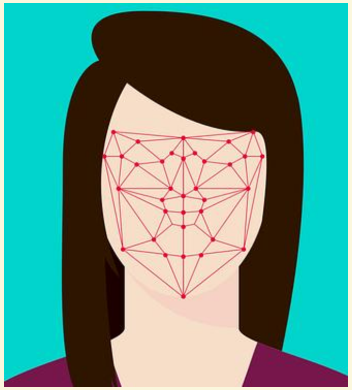
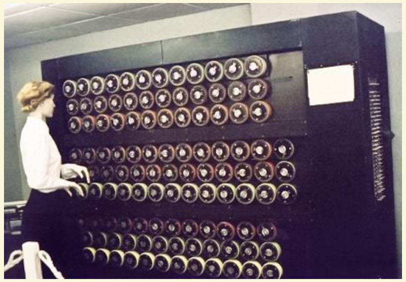
What would come next?

Everyday Patterns



What Use is Pattern Recognition?

Name	Use	Image
Computer Aided Diagnosis	Helps doctors diagnose a patient.	sin 13.3 14.2 13.3 14.1 13.7 13.6 12.8 13.5 12.9 13.3 BA (GP): 13.49 y (13y6m) (M) BA SD: 0.45 y BA SDS: -2.57 (CauEu) BA (TW3): 13.21 y Age: 16.27 y BHI: 4.77 BHI SDS: -0.49 (CauEu) BoneXpert: 2.4.5.1
Speech Recognition	Recognises a person's voice	

Name	Use	Image
Image Recognition	Recognising faces, Handwriting, Registration Plates	
Predicting Weather Patterns	Predict Extreme, life-threatening weather	Hurricane Rita September 22, 2005 10 PM CDT Thursday NWS TPC/National Hurricane Center Advisory 22 Current Center Location 26.2 N 90.3 W Max Sustained Wind 140 mph Current Movement WNW at 10 mph Current Center Location Forecast Center Positions H Sustained wind > 73 mph S Sustained wind 39-73 mph Potential Day 1-3 Track Area Hurricane Warning Tropical Storm Warning Tropical Storm Watch 7 PM Sun 7 PM Sat 7 PM Fri 10 PM Thu TX LA AR OK KS MO CO NM Mexico 35N 30N 25N 105W 100W 95W 90W 85W True at 30.00N SM 125 250 375 500 Approx. Distance Scale (Statute Miles)
Codebreaking	Decrypting encrypted messages	

Algorithms

Activity: Follow My Instructions

Follow My Instructions

1. Fold your sheet of paper in half

Follow My Instructions

1. Fold your sheet of paper in half

2. Unfold your paper

Follow My Instructions

1. Fold your sheet of paper in half
2. Unfold your paper
3. Fold 2 corners to the crease

Follow My Instructions

1. Fold your sheet of paper in half
2. Unfold your paper
3. Fold 2 corners to the crease
4. Fold your sheet of paper in half

Follow My Instructions

1. Fold your sheet of paper in half
2. Unfold your paper
3. Fold 2 corners to the crease
4. Fold your sheet of paper in half
5. Fold one half back on itself

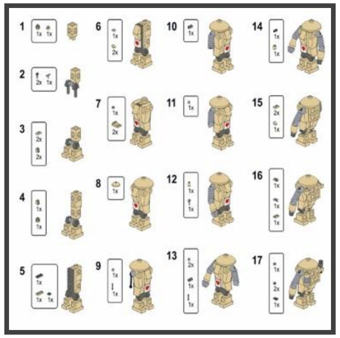
Follow My Instructions

1. Fold your sheet of paper in half
2. Unfold your paper
3. Fold 2 corners to the crease
4. Fold your sheet of paper in half
5. Fold one half back on itself
6. Fold the other half back on itself

Follow My Instructions

7. Throw!

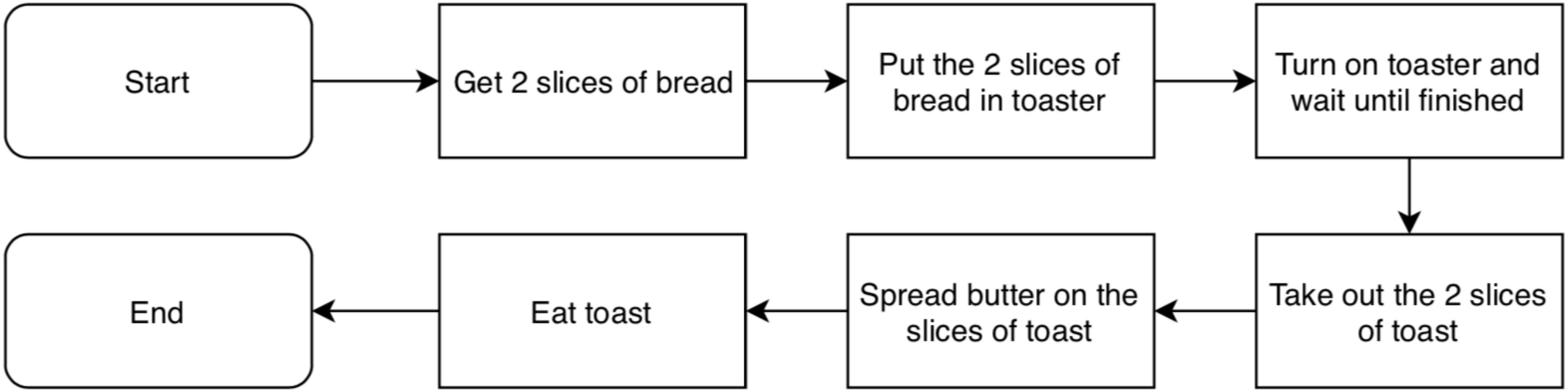
Making a Paper Plane Reflection



Algorithms

An **Algorithm** is a set of simple instructions that are done in a certain order to solve a problem.

Here's an example: Making and Eating Toast



Start

Get 2 slices of bread

Put the 2 slices of bread in toaster

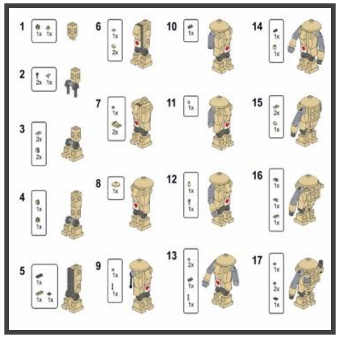
Turn on toaster and wait until finished

Take out the 2 slices of toast

Spread butter on the slices of toast

Eat toast

End

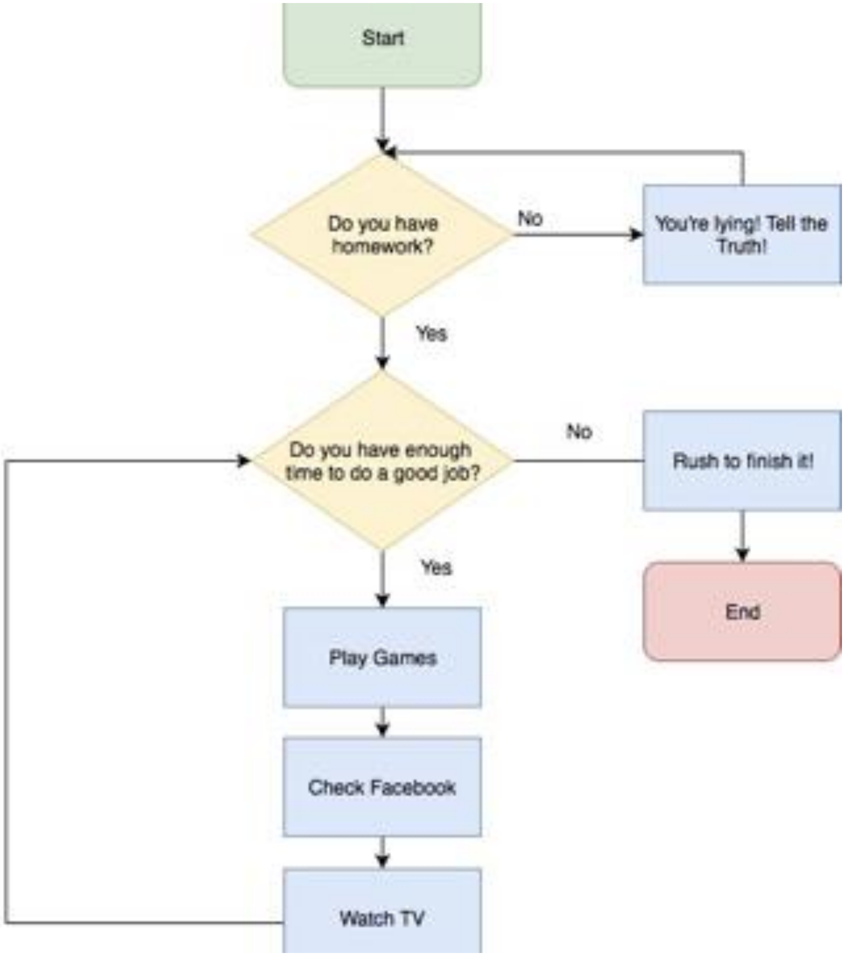


Algorithms

It is important to remember when writing an algorithm to keep instructions:

- simple
- in the correct order
- unambiguous
- relevant to solving the problem

Where do we use algorithms in everyday life?



Start

Do you have homework?

No

You're lying! Tell the Truth!

Yes

Do you have enough time to do a good job?

No

Rush to finish it!

End

Yes

Play Games

Check Facebook

Watch TV

Computational Thinking Puzzles

The River Crossing Conundrum

A Man needs to cross a river. There is no other way to cross safely other than the use of a rowing boat nearby. The Man cannot swim across. The Man has three things with him:

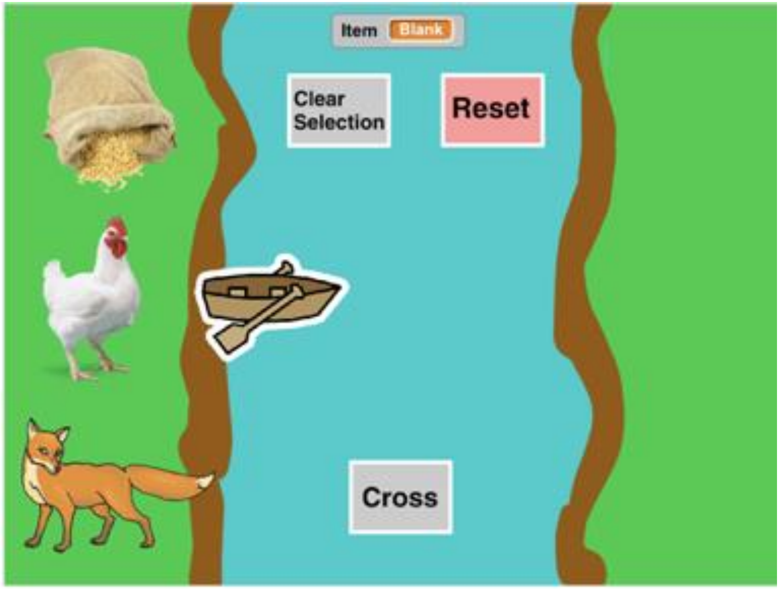
- a sack of corn
- a chicken
- a fox

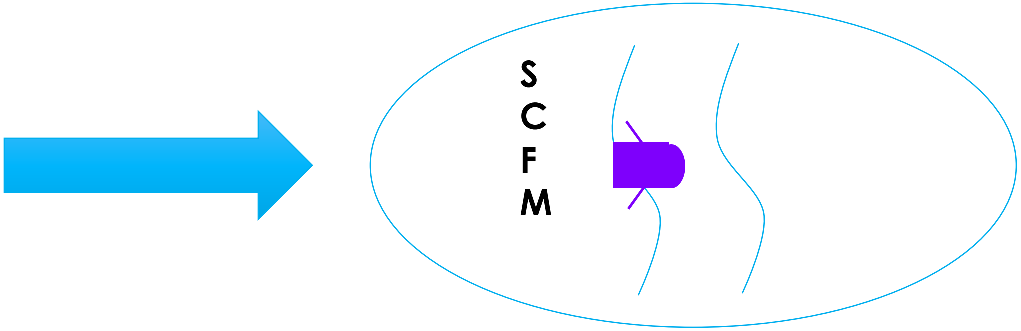
He must safely bring these across, however, the **boat can only carry the Man and one other object at a time e.g. The Man and the sack of corn.**

Another problem is that the **chicken cannot be left alone with the sack of corn**, and the **chicken can not be left alone with the fox.** How can the Man cross the river?

Visualising the Problem

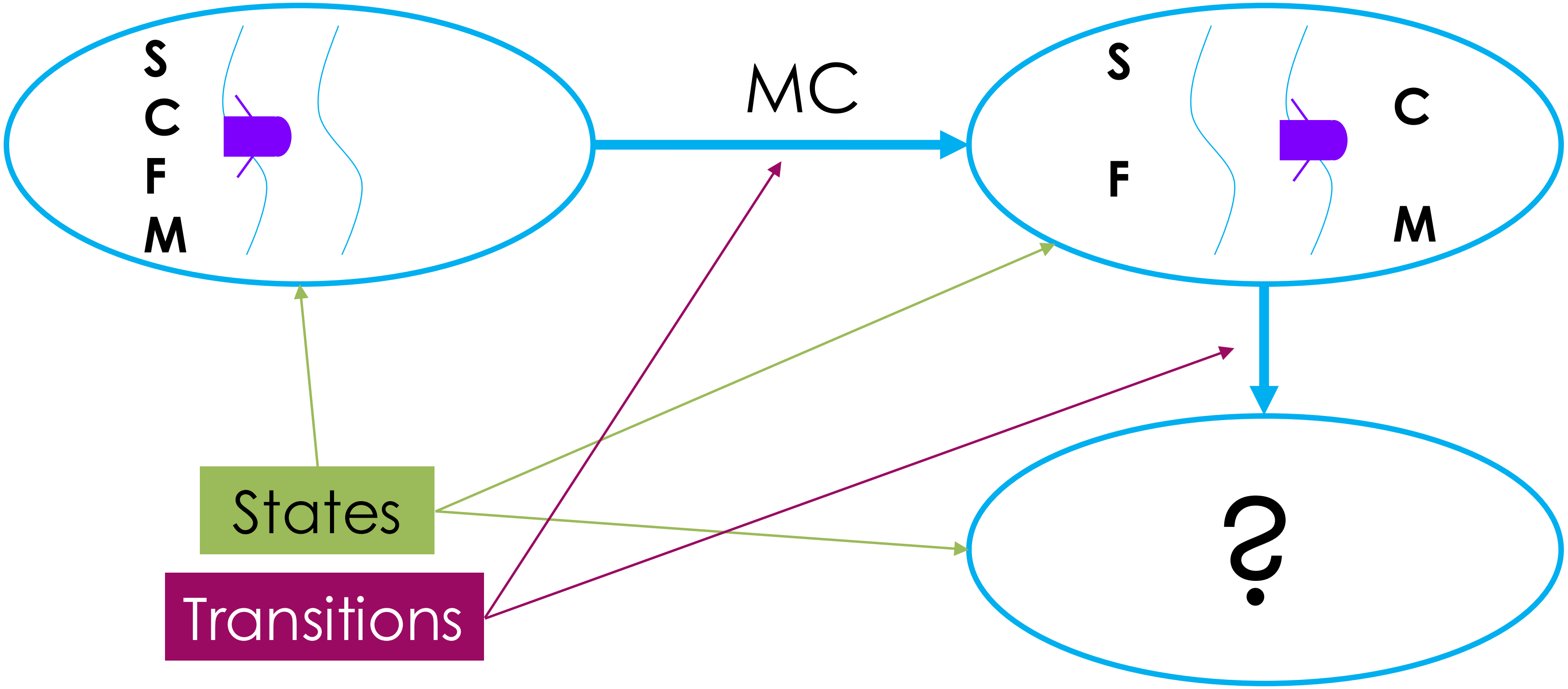
Using abstraction we can simplify the problem to only the necessary details.





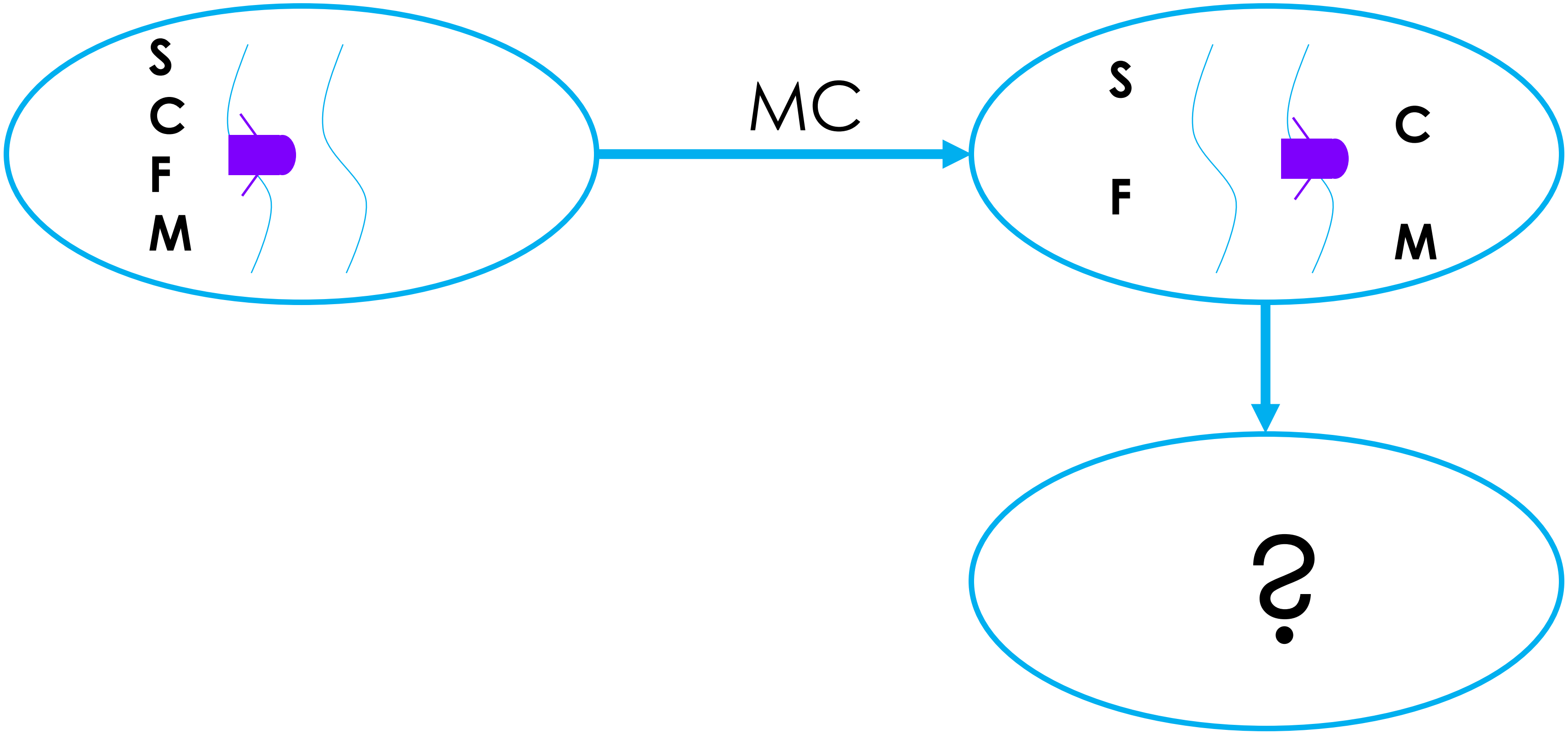
Labelled Transition System

Mapping out the steps of the puzzle using the different states is called a Labelled Transition System. It is made up of states and transitions.

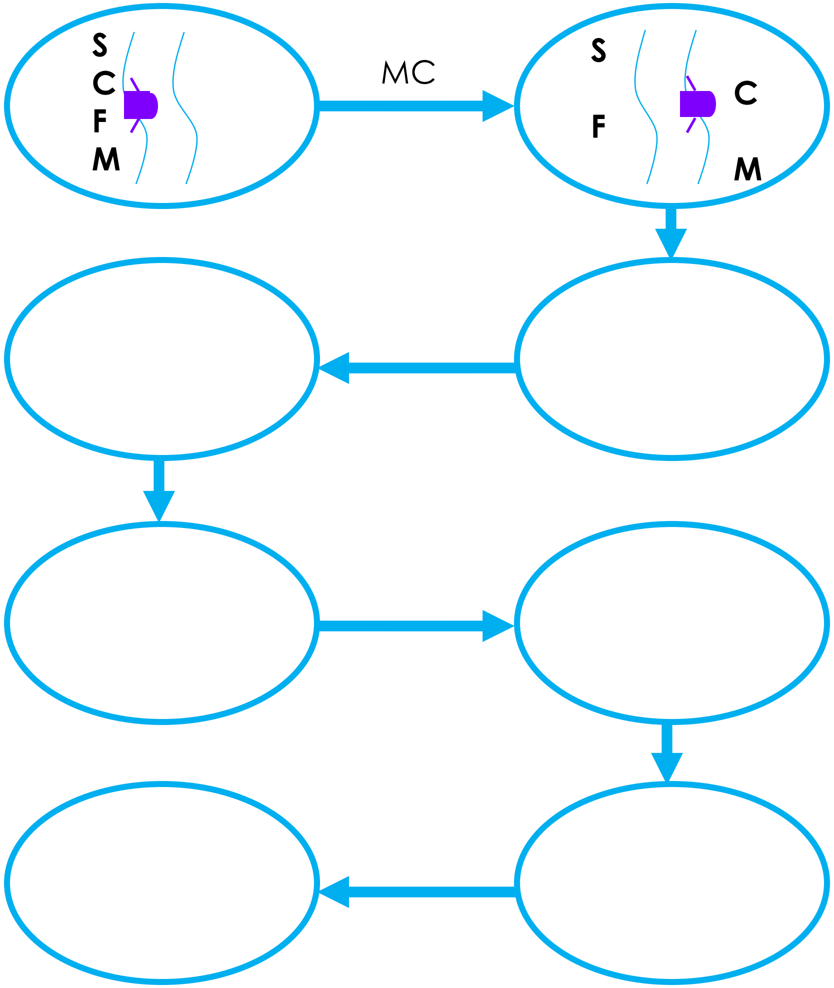


Activity: River Crossing LTS

In your workbooks complete the LTS diagram to solve the River Crossing Conundrum.

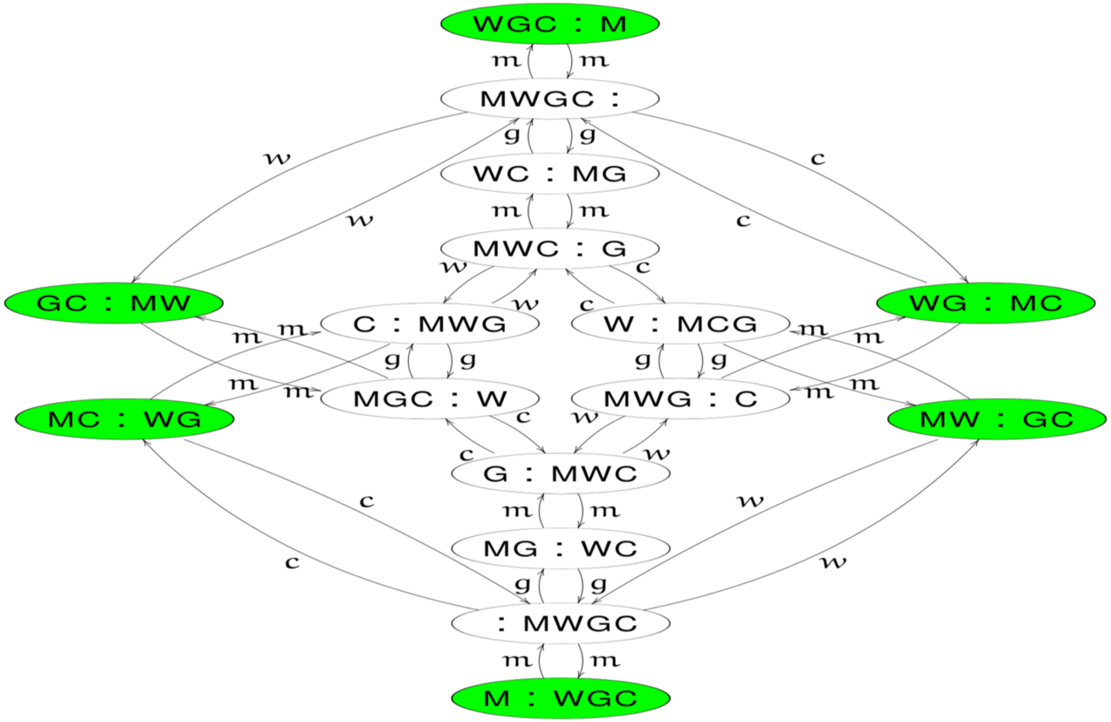


Activity: River Crossing LTS



Labelled Transition Systems

Labelled Transition Systems in reality show every possible state, even the ones that lead to losing the puzzle.



Finding the Scratch Game

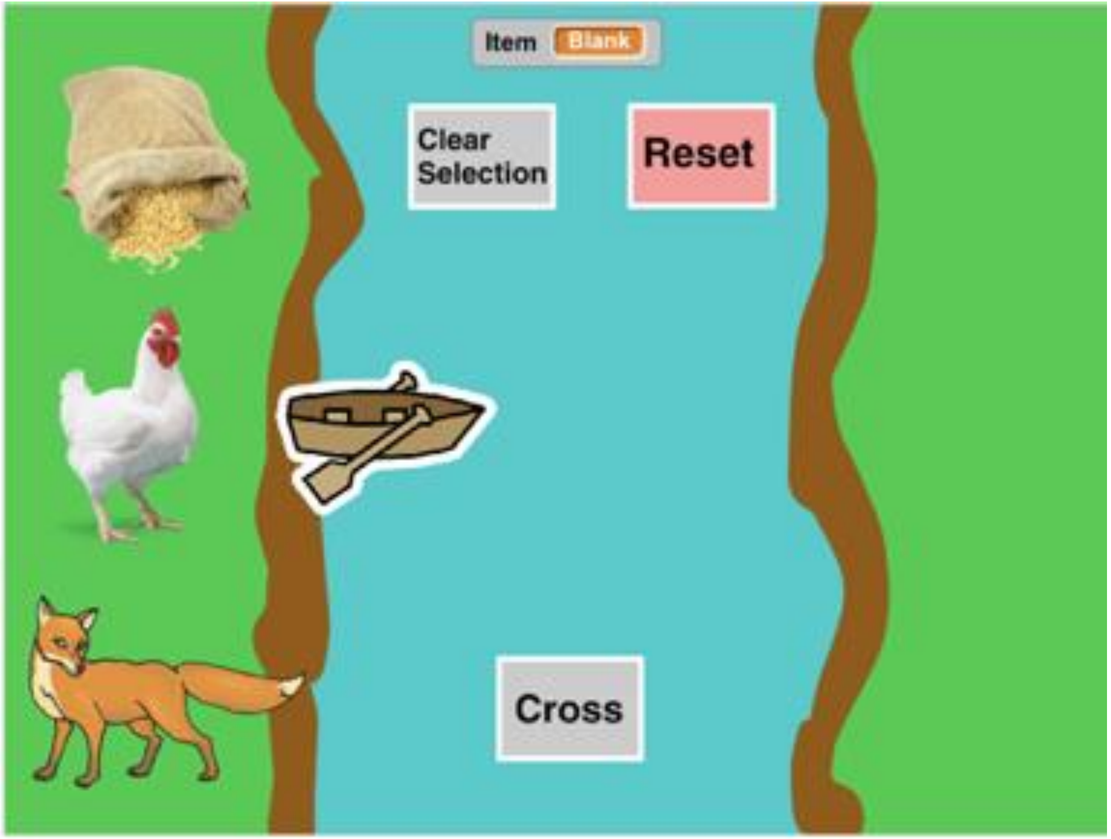
Use the following link to find the game:

tc1.me/TechnoThink

Choose your Language by clicking the English or Welsh Flag.

The River Crossing Conundrum

Complete the River Crossing Conundrum game.



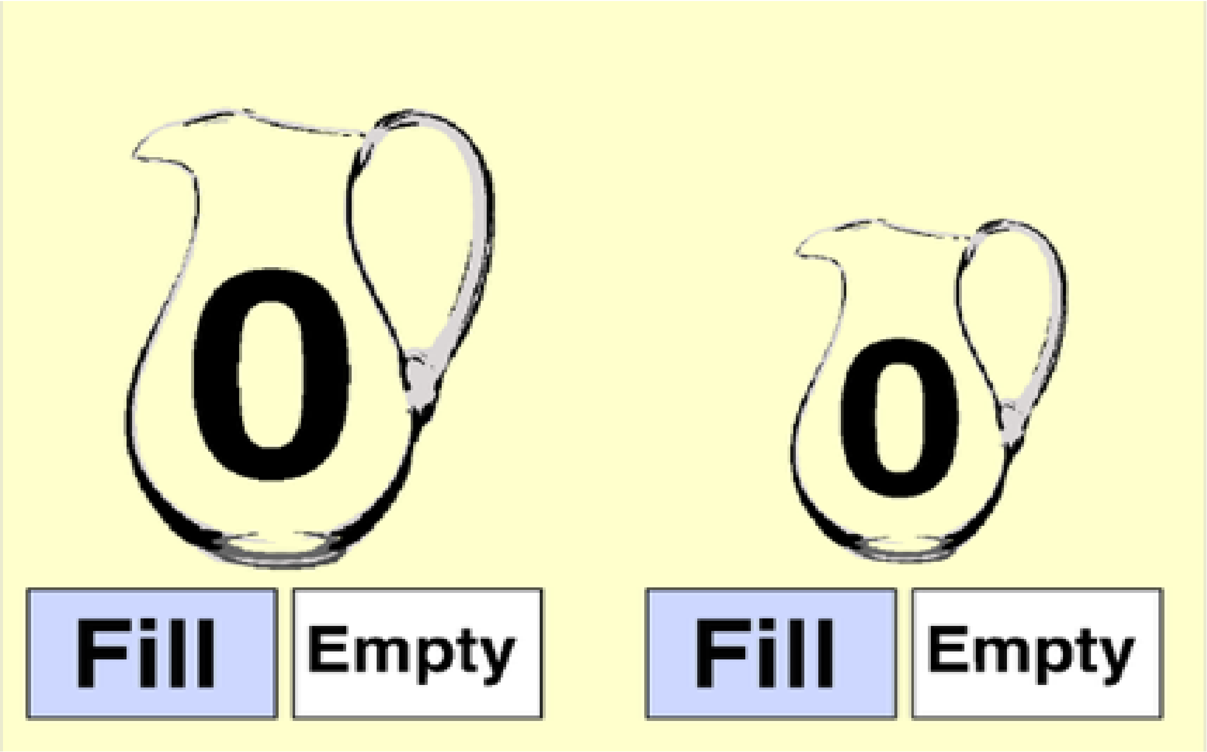
The Water Jug Challenge

There are two empty jugs: a 5 litre jug and a 3 litre jug. The jugs do not have any markings on them to help with measurements.

The only way to solve the puzzle is by filling the empty 5 litre jug with **precisely 4 litres of water**.

Activity: The Water Jug Challenge

Complete the Water Jug Challenge game. Write down the steps in your books.



The Bridge Crossing Conundrum

Alice, **Bob**, **Carol** and **Dave** have to cross a bridge in the dark of night. The bridge is rigged to explode in **17 minutes**.

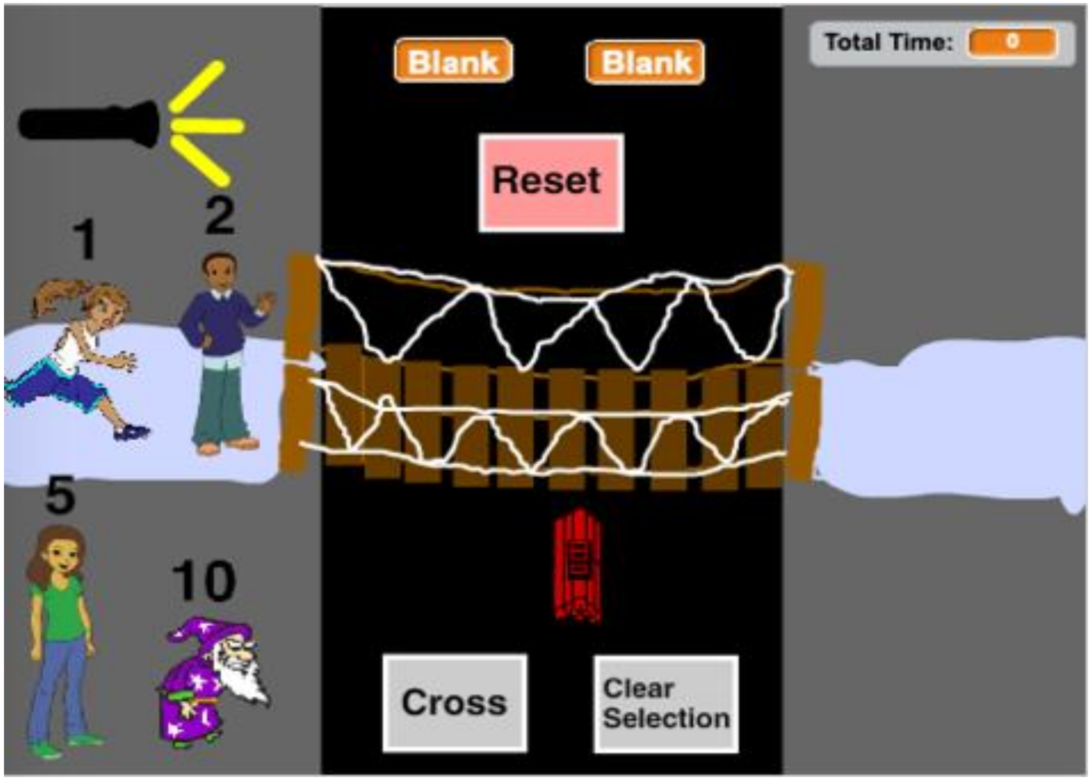
Their walking speeds allow them to cross in **1**, **2**, **5** and **10** minutes, respectively.

They have one flashlight which must be used to cross the bridge, but the bridge can only hold two people at once.

Try to get them all across the bridge before the bridge explodes.

Activity: The Bridge Crossing Conundrum

Complete the Bridge Crossing Conundrum game. Write down the steps in your workbooks.



Activity: Defining an Algorithm

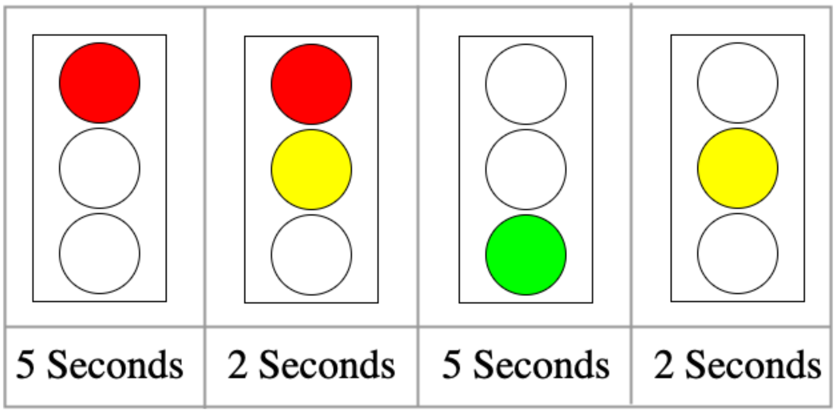
Iteration

An **iteration** is a single pass through a set of instructions. Most programs contain a set of instructions that are executed over and over again. The computer is said to be iterating through the loop.

Some processes include steps or a series of steps that are iterated.

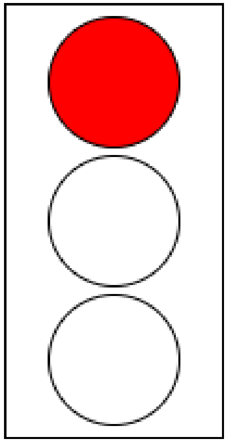
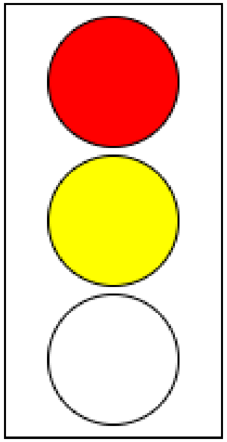
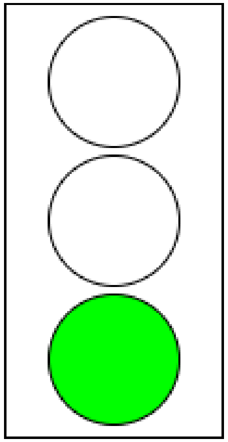
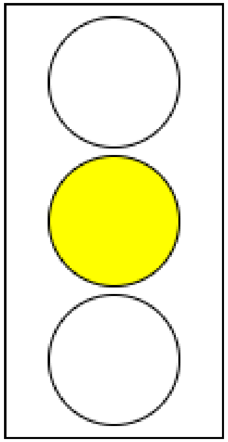
Example: Simple Traffic Lights

- What instructions would you use for this process?
- What needs to be iterated?
- Does this process ever end?



Activity: Traffic Lights Algorithm

Create a Scratch program which simulates the following traffic lights.

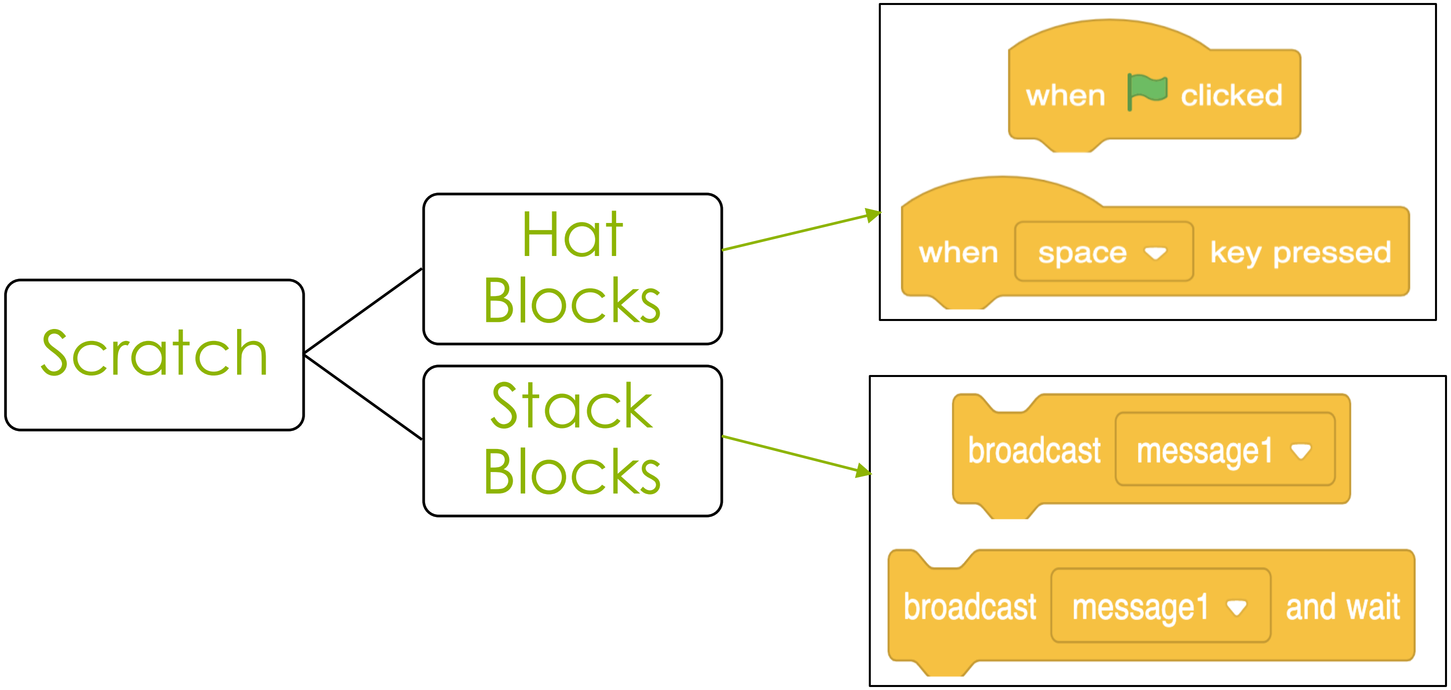
			
5 Seconds	2 Seconds	5 Seconds	2 Seconds

technoteach
technocamps

Programming in Scratch

Scratch

- ➢ Drag and drop blocks to give instructions
- ➢ Easy to create and other useful programs
- ➢ User friendly

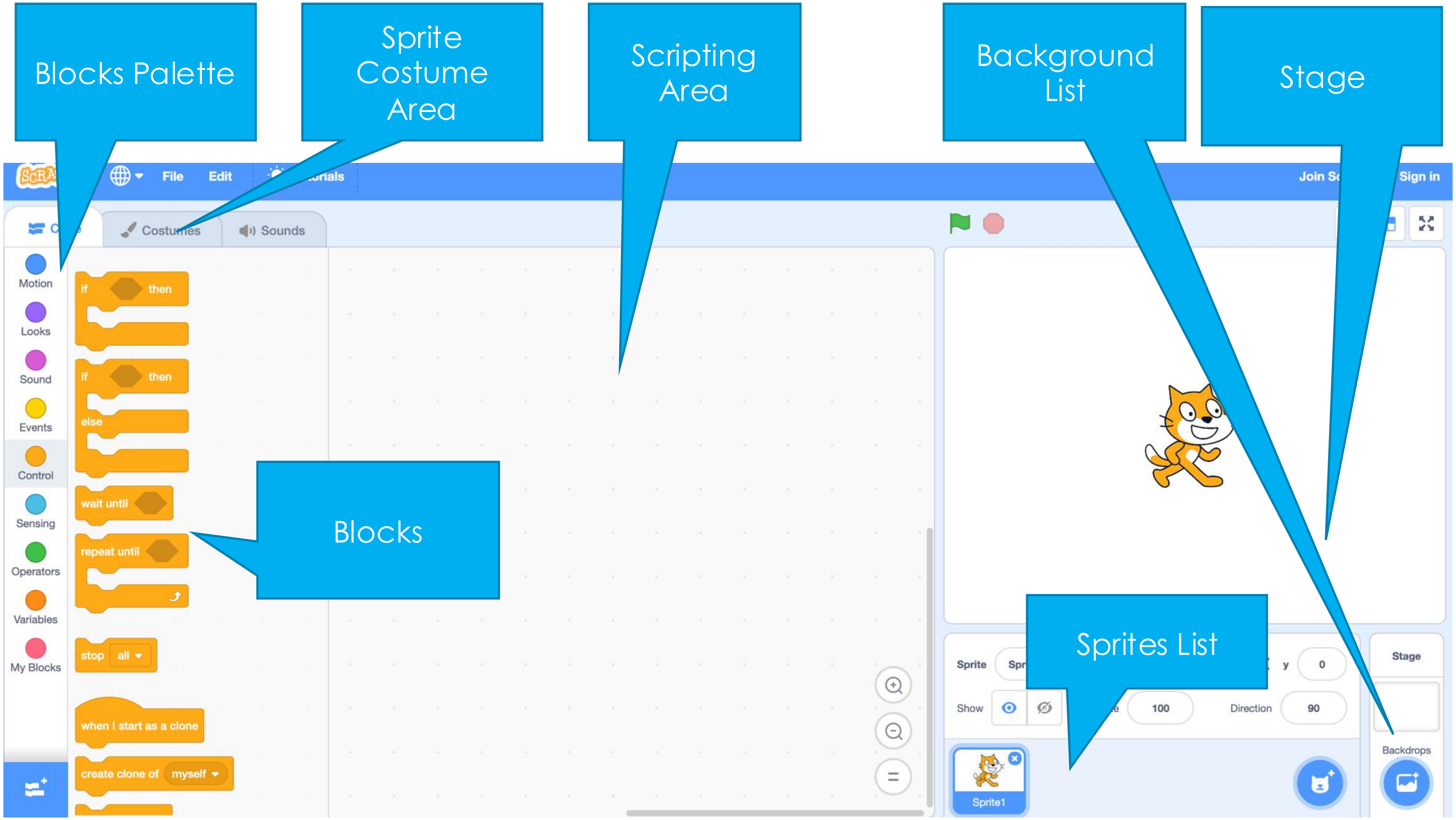


Scratch

Hat Blocks

Stack Blocks

Scratch Basics



Blocks Palette

Sprite Costume Area

Scripting Area

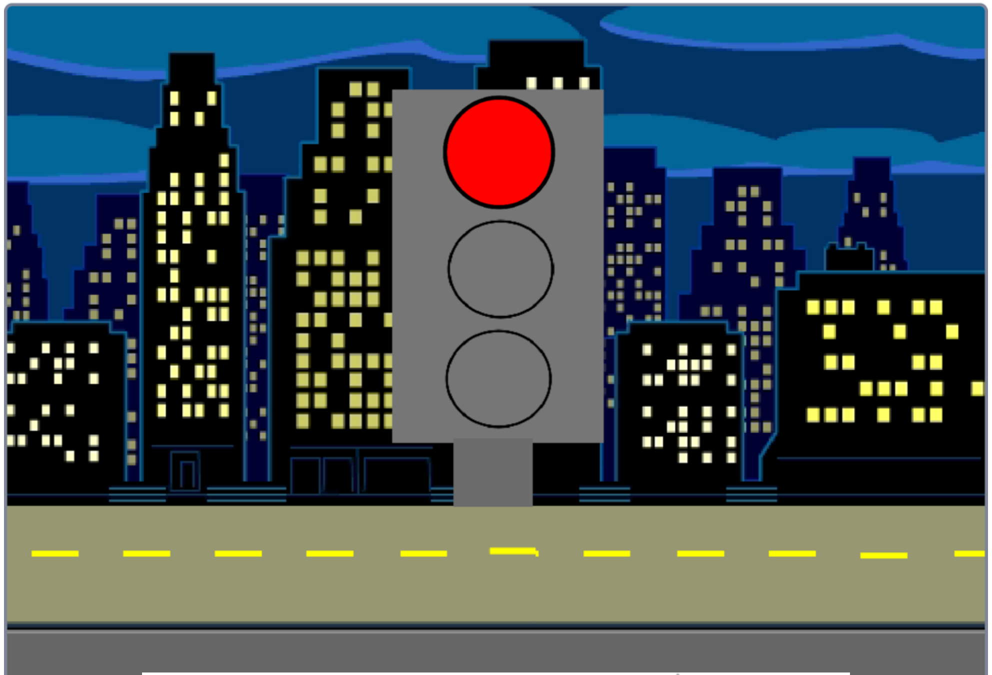
Background List

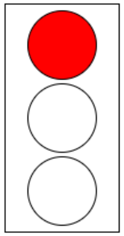
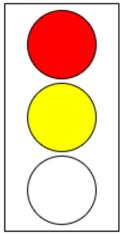
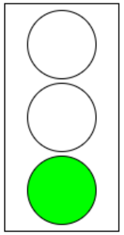
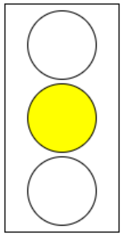
Stage

Blocks

Sprites List

Traffic Lights Algorithm - How Could We Design It?



			
5 Seconds	2 Seconds	5 Seconds	2 Seconds

Activity: Modifying Your Lights

Here are a few ways to extend your program:

1. **Add a Button** for pedestrians and modify the sequence so that it **stays green until the button is pressed.**
2. Add a crossing on the street and a **light to tell pedestrians when to cross.**
3. Add a car which stops at the crossing if the light is red.
4. Add pedestrians who stop at the crossing until it is safe to cross.

Scratch Tours