National Centre for Computing Education

Supporting autistic pupils in primary computing

Session 3 - Computing-specific pedagogy - how we teach computing

Learning outcomes

- Explore different approaches to teaching programming that can address strengths and barriers to learning of autistic students

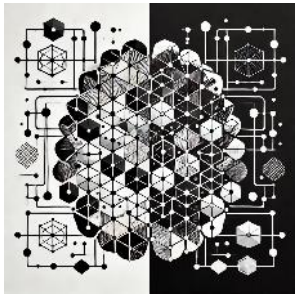
- Recognise how problem-solving approaches can be developed through computational thinking

Autism and Strengths

Attention to detail

Creative

Visual



Logical reasoning

12 Pedagogy Principles

Effective teaching approaches (pedagogy) are at the heart of good teaching practice.

These principles continue to develop and evolve and are documented by the NCCE with this poster.

Pedagogy-principles.pdf

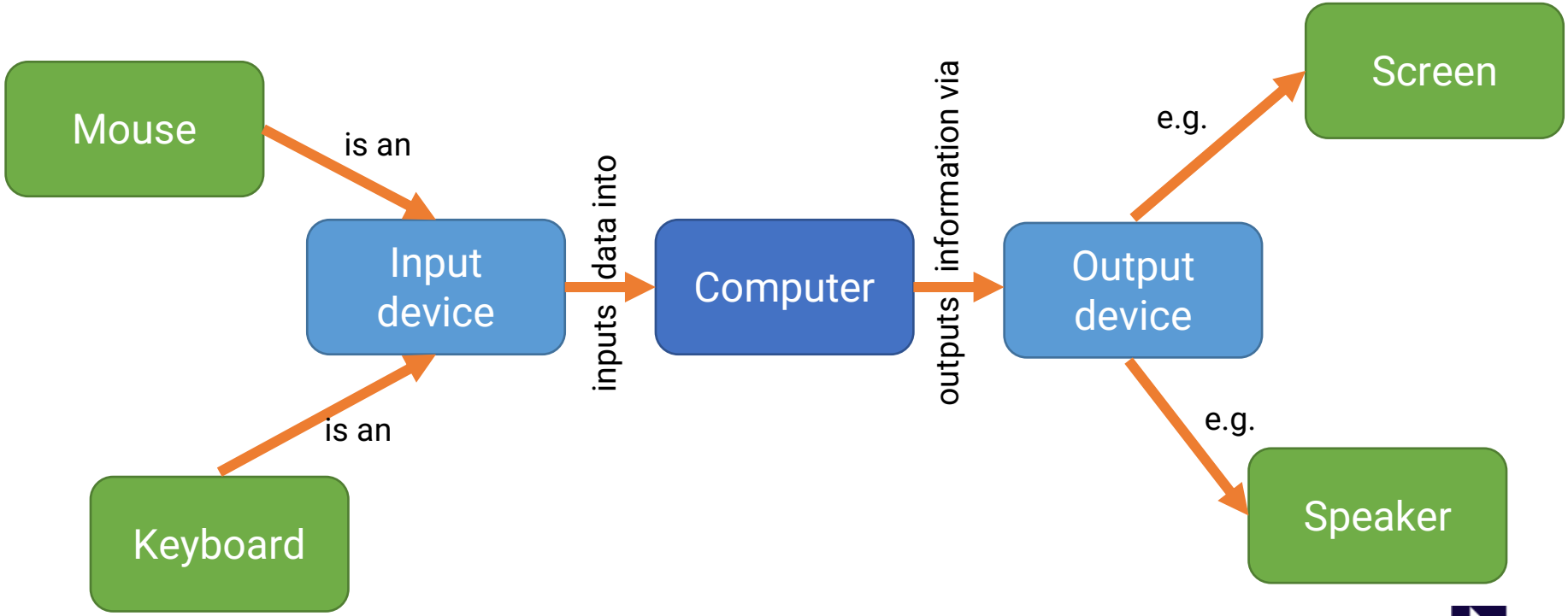
Lead with concepts

Support pupils in the acquisition of knowledge, through the use of key concepts, terms, and vocabulary, providing opportunities to build a shared and consistent understanding. Glossaries, concept maps, and displays, along with regular recall and revision, can support this approach.

Unplug, unpack, repack

Teach new concepts by first unpacking complex terms and ideas, exploring these ideas in unplugged and familiar contexts, then repacking this new understanding into the original concept. This approach (semantic waves) can help pupils develop a secure understanding of complex concepts.

Concept Maps



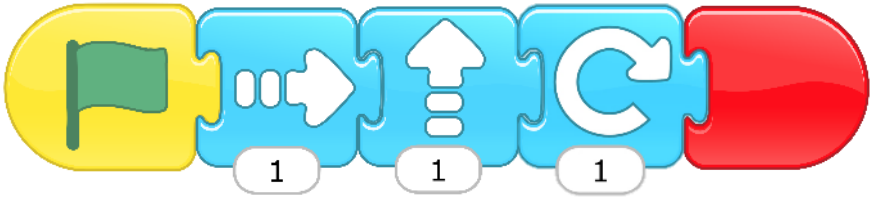
Unplugged activities

Lego Algorithm activity from Barefoot

Unplugged activities

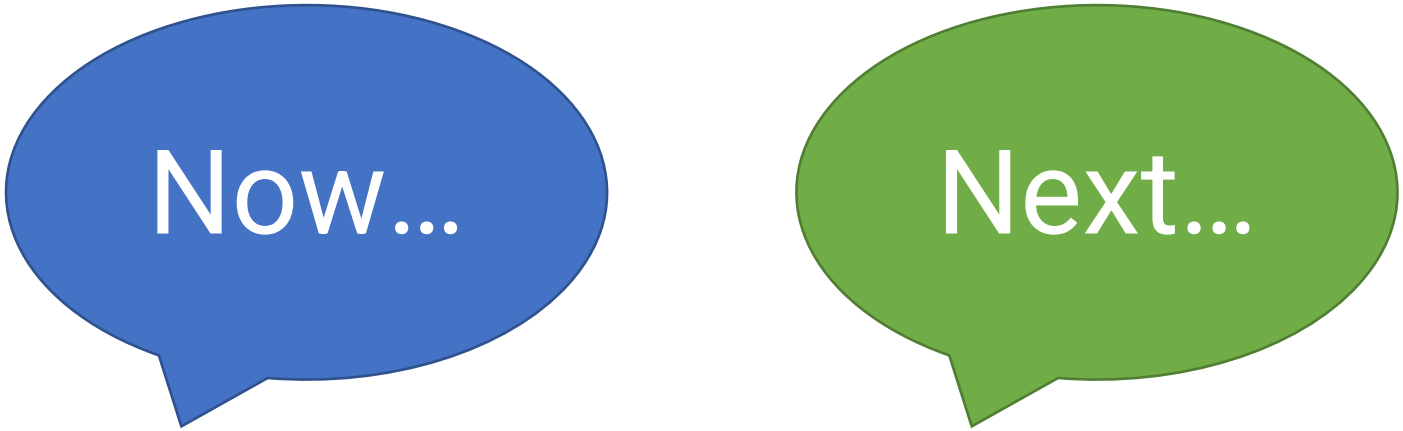
Make explicit links with a computational model

When green flag clicked:
Move forward
Move up
Turn right
Stop



ScratchJr is a collaboration between the DevTech Research Group at Boston College and the SCracth Foundation - ScratchJr - Terms and Conditions

Unplugged activities



Challenge misconceptions

Use formative questioning to uncover misconceptions and adapt teaching to address them as they occur. Awareness of common misconceptions alongside discussion, concept mapping, peer instruction, or simple quizzes can help identify areas of confusion.

Peer Instruction

In peer instruction, instructors pose a challenging question to students, students answer the question individually, students work with a partner in the class to discuss their answers, and finally students answer the question again.

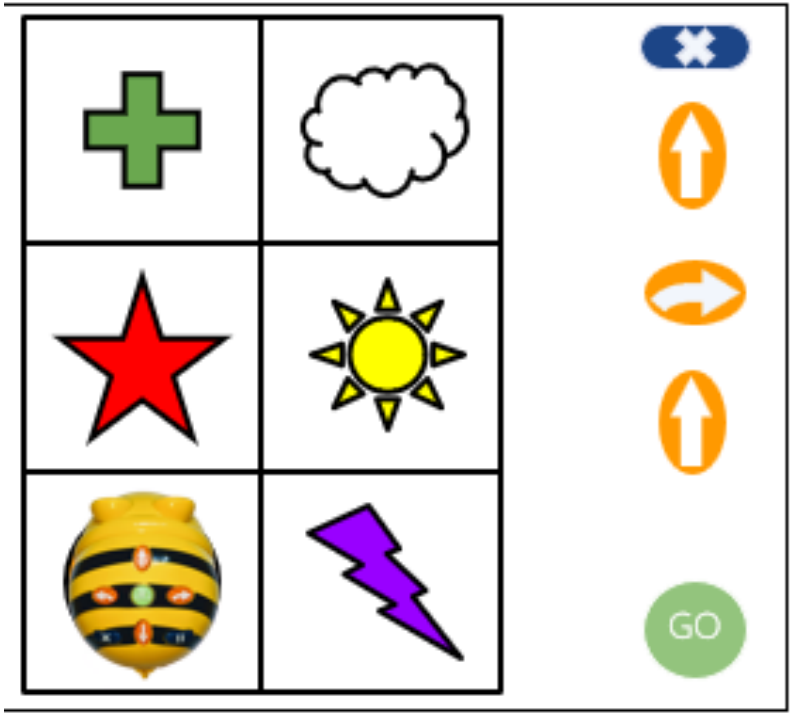
Where does the Bee-Bot end up?

A. 

B. 

C. 

D. 



Images from Teach Computing Curriculum

Structure lessons

Use supportive frameworks when planning lessons, such as PRIMM (Predict, Run, Investigate, Modify, Make) and Use-Modify-Create. These frameworks are based on research and ensure that differentiation can be built in at various stages of the lesson.

Read and explore code first

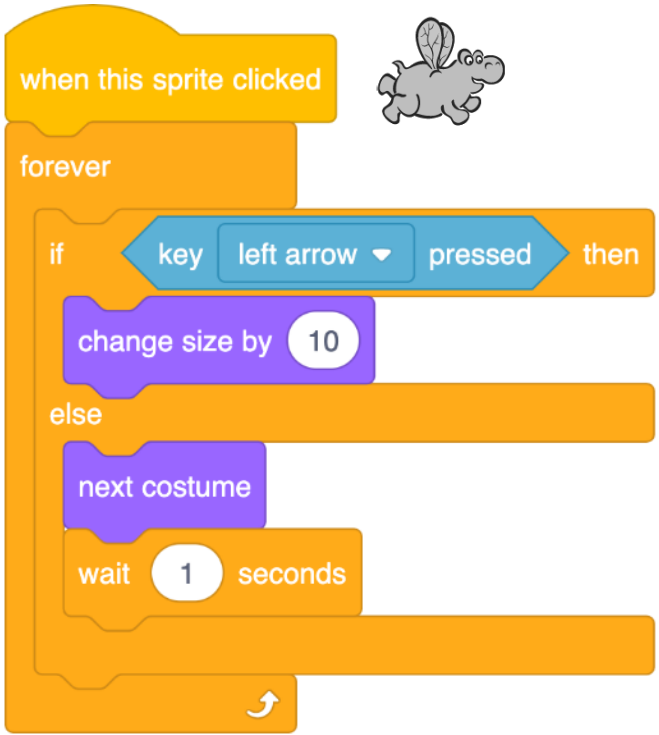
When teaching programming, focus first on code 'reading' activities, before code writing. With both block-based and text-based programming, encourage pupils to review and interpret blocks of code. Research has shown that being able to read, trace, and explain code augments pupils' ability to write code.

The PRIMM Approach

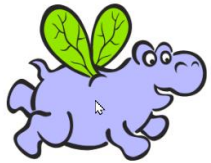
Predict: Students discuss a program and predict what it might do; they can draw or write out what they think will be the output. At this level, the focus is on the function of the code.
Run: Students run the program so that they can test their prediction and discuss in class.
Investigate: The teacher provides a range of activities to explore the structure of the code, such as tracing, explaining, annotating, debugging, etc.
Modify: Students edit the program to change its functionality via a sequence of increasingly more challenging exercises; the transfer of ownership moves from the code being ‘not mine’ to ‘partly mine’ as students gain confidence by extending the function of the code.
Make: Students design a new program that uses the same structures, but solves a new problem (ie has a new function).

Quick Read: ncce.io/qr11

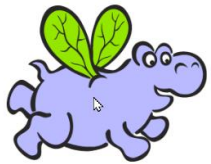
Predict: What happens if the condition is false?



A



B

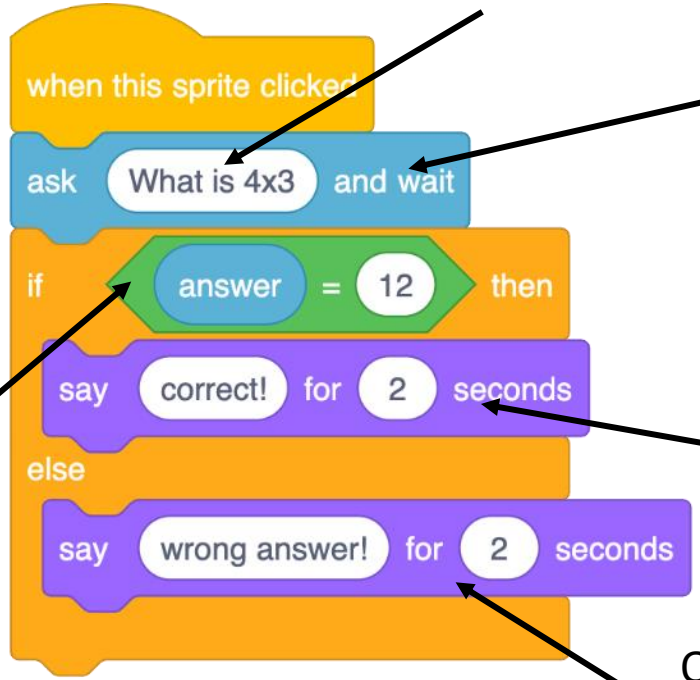


Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See https://scratch.mit.edu/

Investigate

Where is the:

- Condition
- Outcome if the condition is true
- Outcome if the condition is false



Question to be asked

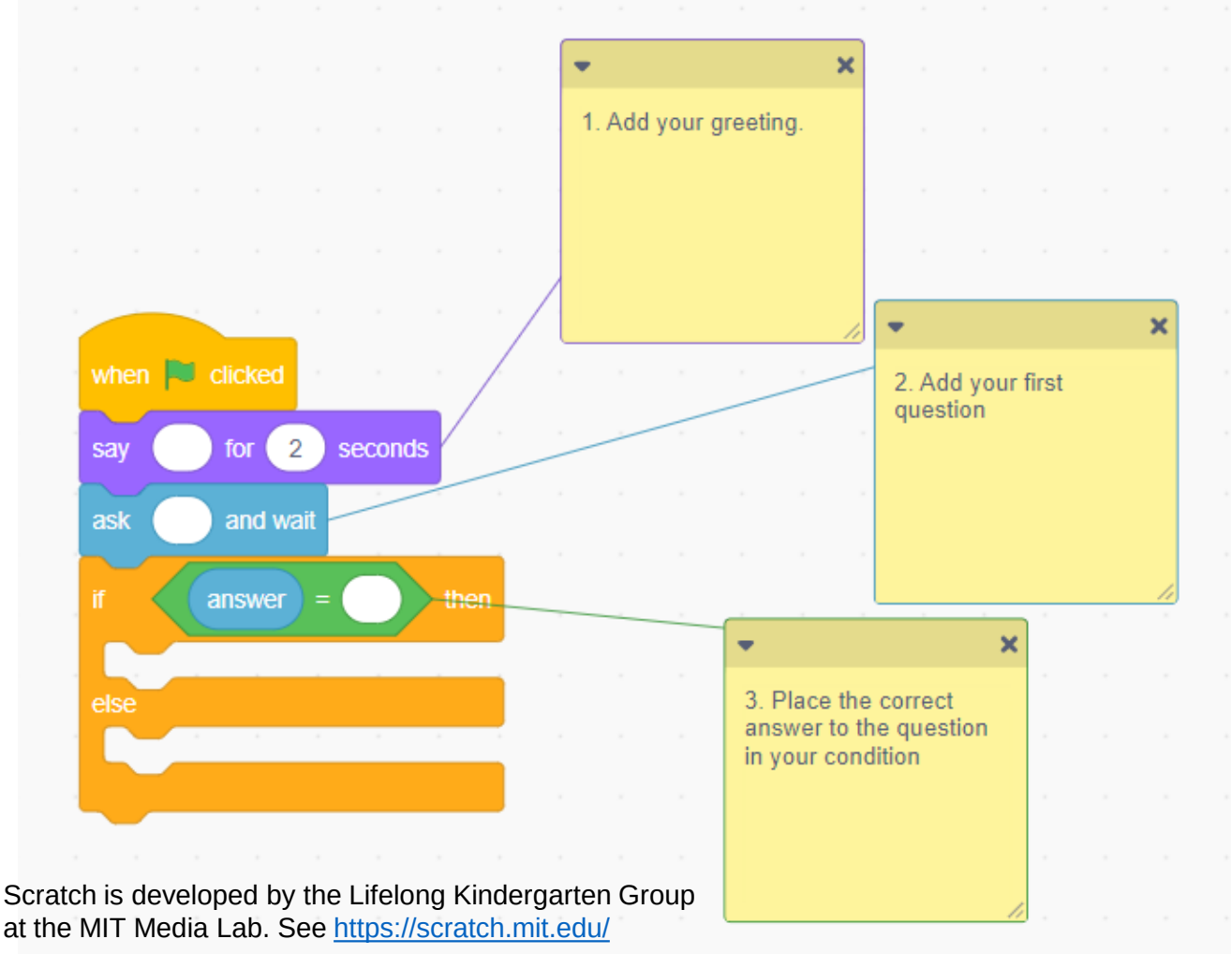
Wait means an infinite loop is not needed

Condition — the answer is the same as '12'

Outcome if condition is true (**then**)

Outcome if condition is false (**else**)

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See https://scratch.mit.edu/



Modify:

https://scratch.mit.edu/projects/1027342510/editor/

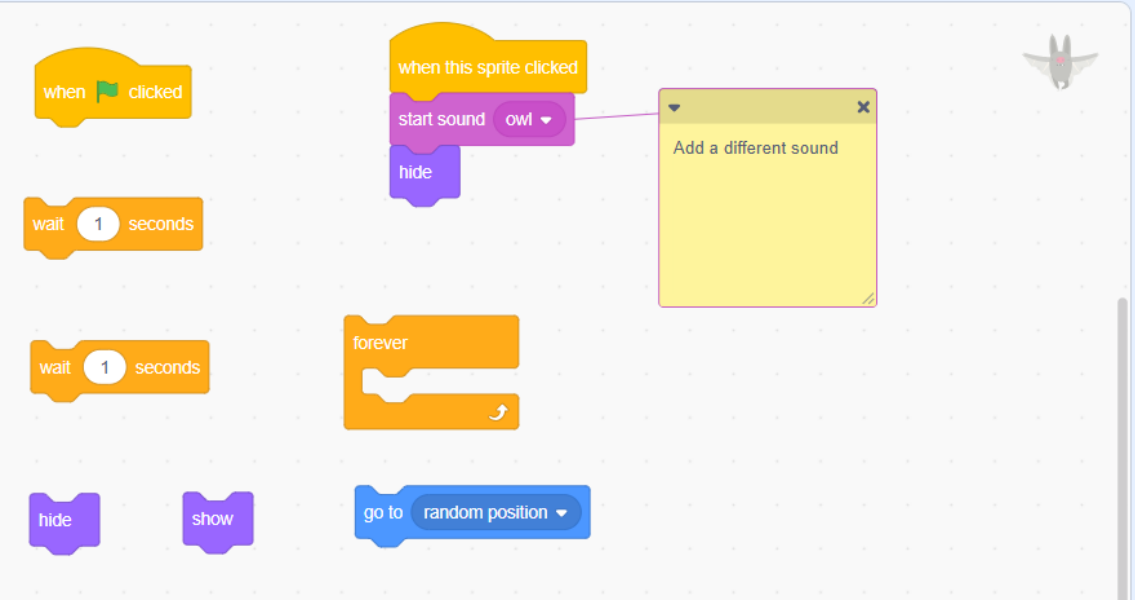
Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See https://scratch.mit.edu/

Foster program comprehension

Use a variety of activities to consolidate knowledge and understanding of the function and structure of programs, including debugging, tracing, and Parson's Problems. Regular comprehension activities will help secure understanding and build connections with new knowledge.

Parsons Problem

Y4 L5 – Bat catching order



Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See https://scratch.mit.edu/

Debugging

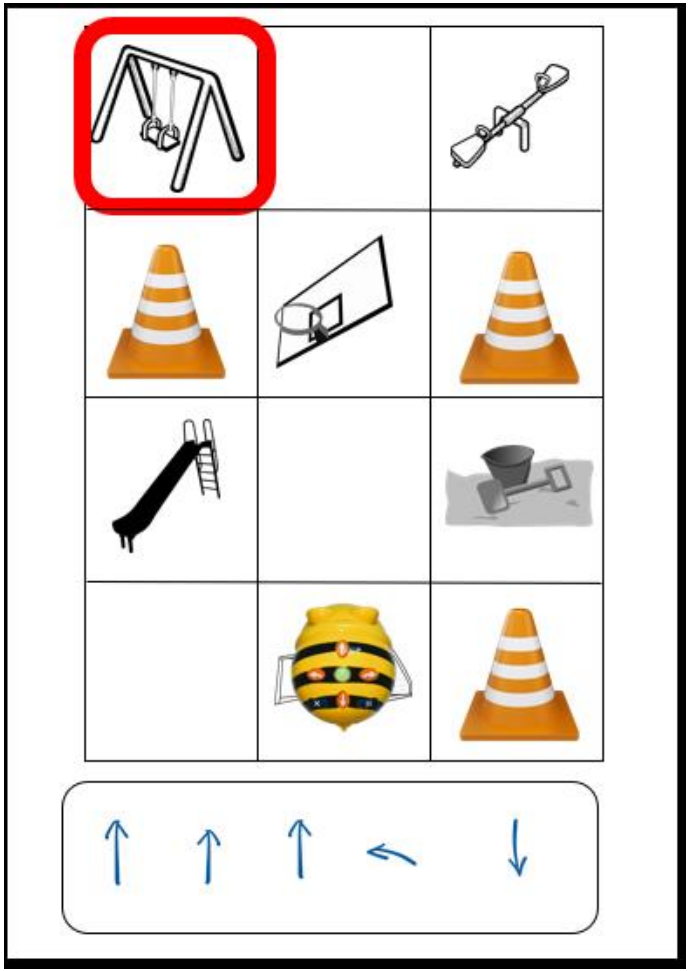


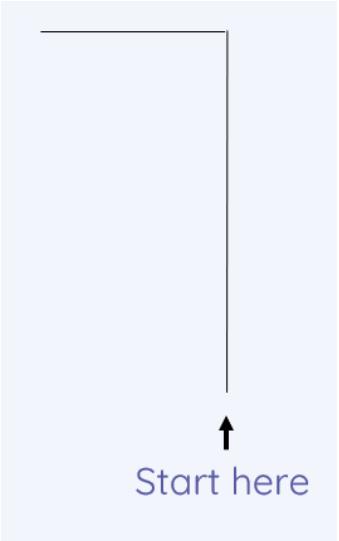
Image from Teach Computing Curriculum

DEBUGGING STRATEGIES:

1. Explain to your neighbour what you want the program to do.
2. Describe what isn't working. What does it do instead?
3. Look at the code/algorithm and run the program. Point to each part of the code as it runs (this is called tracing).
4. Can you see which part of the code is going wrong?
5. Read through your program again and explain what each bit of code does. Now can you spot the error?

Debugging

Go to
https://www.transum.org/Software/Logo/



Get hands on

Get hands-on

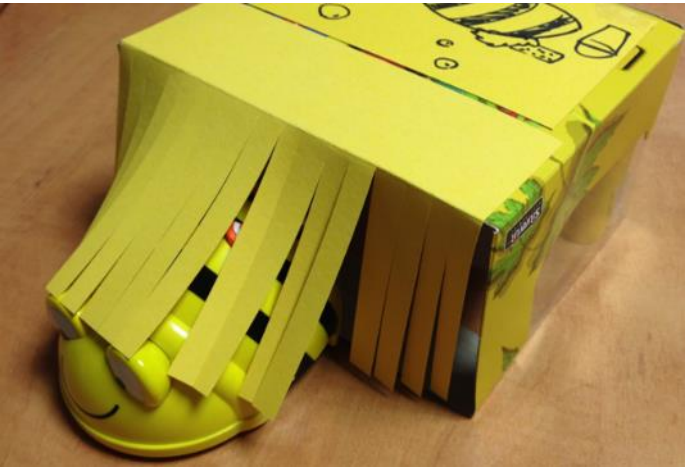
Use physical computing and making activities that offer tactile and sensory experiences to enhance learning. Combining electronics and programming with arts and crafts (especially through exploratory projects) provides pupils with a creative, engaging context to explore and apply computing concepts.

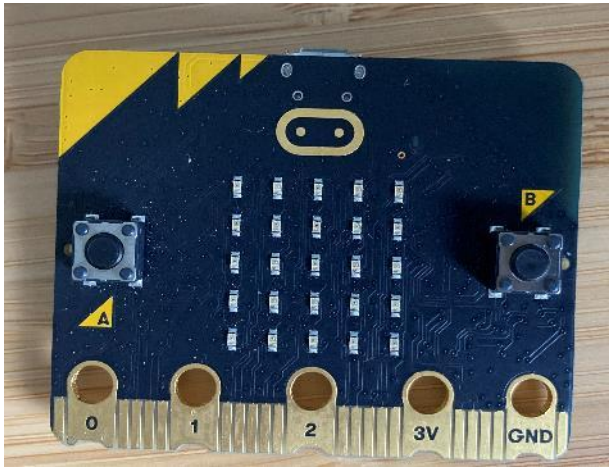
“Overall, physical computing activities result in higher motivational values than the average of any other activities in the computer science classroom”

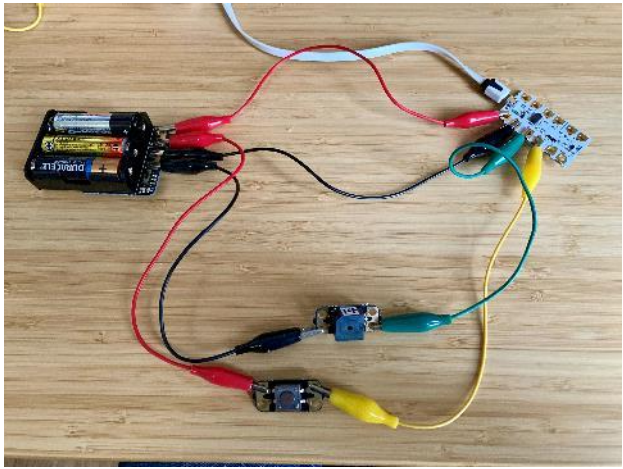
Przybylla, M. and Romeike, R (2018): Impact of Physical Computing on Learner Motivation

EYFS/KS1:
Glow & Go Bot
Bee-Bot
Blue-Bot
Code-a-pillar

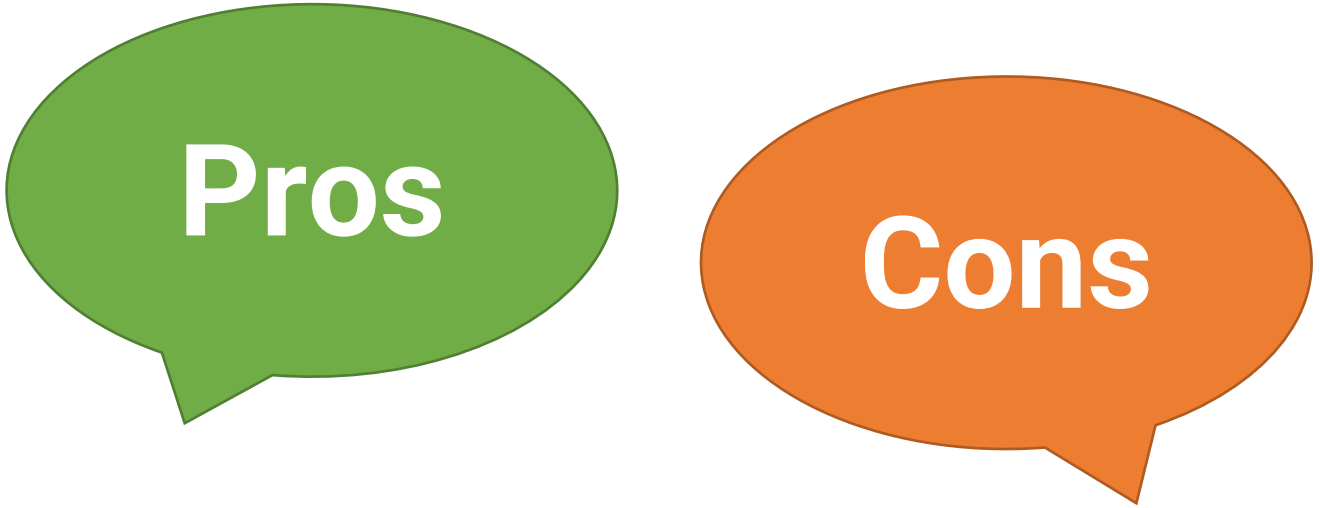
KS2:
Microbit
Crumble
Makey Makey







Physical Computing



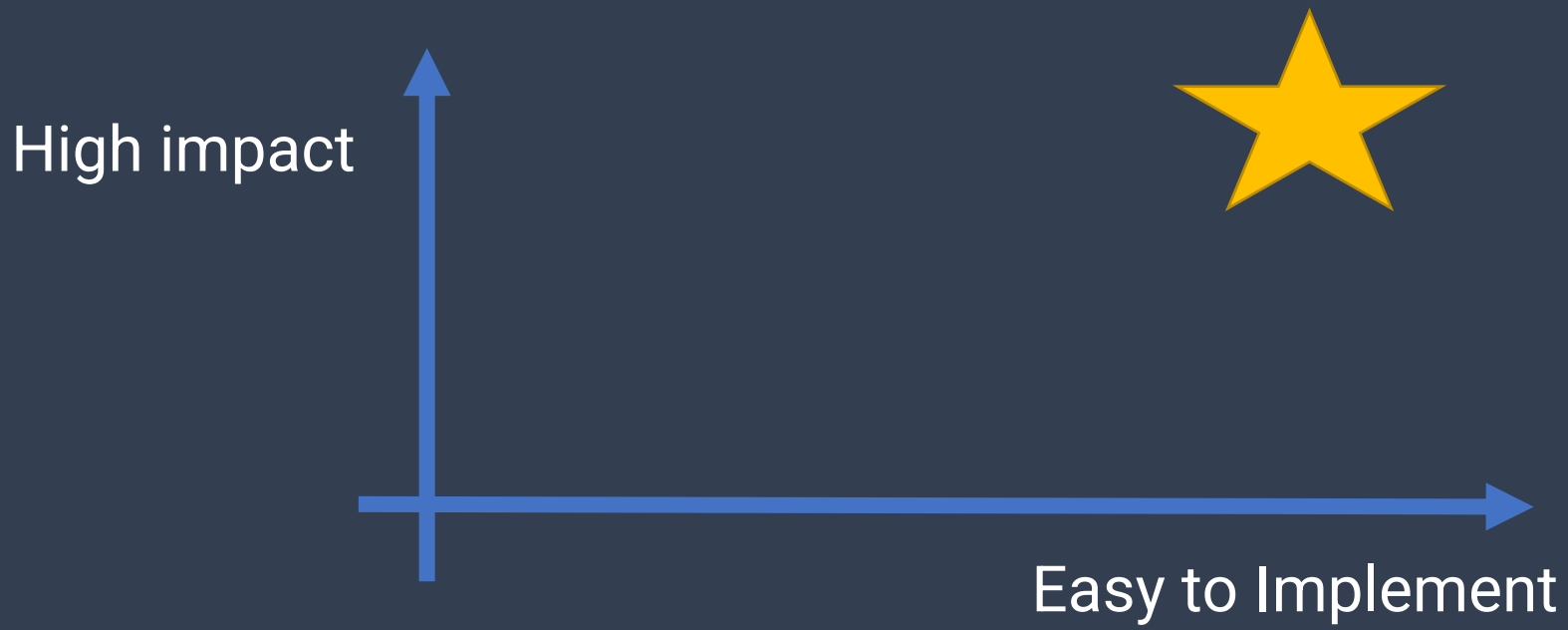
Square Pegs and Round Holes: Pedagogy for Autistic Students in Computing Education (2024)

Top 3 effective approaches according to teachers:

- Physical Computing
- Unplugged activities
- PRIMM

https://www.researchgate.net/publication/377245770_Square_Pegs_and_Round_Holes_Pedagogy_for_Autistic_Students_in_Computing_Education

Miro Board – Frame 5



Learning Outcomes

- Recognise how different approaches to teaching programming can address strengths and challenges of autistic students

- Consolidate knowledge of key concepts through a range of pedagogical approaches

Thanks