COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE

Barefoot Computing

KS1 Bee-Bots 1,2,3 Activity:

An introduction to programming with Bee-Bots

Recommended Year Group: Year 1 or 2 (although can be adapted for other years)
Activity Duration: 30 mins per group + 15 mins for overall introduction & plenary

Concepts and approaches

Algorithms Programming Sequencing Tinkering Debugging Collaborating

Curriculum links

Computing

- understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions;
- create simple programs

Maths

Geometry – position and direction:

- Year 1: Pupils should be taught to: describe position, direction and movement, including whole, half, quarter and three-quarter turns
- Year 2: Pupils should be taught to: use mathematical vocabulary to describe position, direction and movement, including movement in a straight line and distinguishing between rotation as a turn and in terms of right angles for quarter, half and three quarter turns (clockwise and anti-clockwise)

Introduction

Pupils create sequences of instructions (an algorithm) to draw the shape of numeral e.g. 3. An algorithm is a sequence of instructions, or a set of rules, for performing a specific task.

Programming in this activity involves taking the algorithm and using it to program a Bee-Bot to navigate a route tracing out the shape of the numeral.

You and your pupils develop an understanding of programming. Your pupils will:

- work out the route to trace the shape of a numeral (e.g. 3): to do this they create an algorithm
- use the algorithm to help them program the Bee-Bot
- work as a team (collaborating)

In this activity you and your pupils will learn about programming, debugging, algorithms and persevering. You will also be introduced to decomposition, logic and collaborating.

Allow time for children to understand how to work effectively with Bee-Bots. Once they understand how to work out (design) an algorithm, it can be useful to record it (e.g.on a whiteboard or in an exercise book). Pupils use their algorithm to create their program. The basic concepts and approaches they learn now will be used as they move to program in other programming languages.

Prior knowledge

Children should ideally have had been introduced to algorithms e.g. Crazy Characters, Spelling Rules, Sharing Sweets.
It is expected that pupils have already tinkered with Bee-Bots and have some ideas about how they work: for example allow children to play with Bee-Bots in spare time e.g Bee-Bots Tinkering

Organisation

For the group sessions, you will probably want to work with only four or five pairs of children at a time. Other pupils will need independent work during this time, or some could work on this activity with a TA. Some suggestions have been made but please adapt to fit in with your class. Each pair in the group session will need a Bee-Bot.

Pupil objectives

- I can record an algorithm.
- I can program a Bee-Bot.

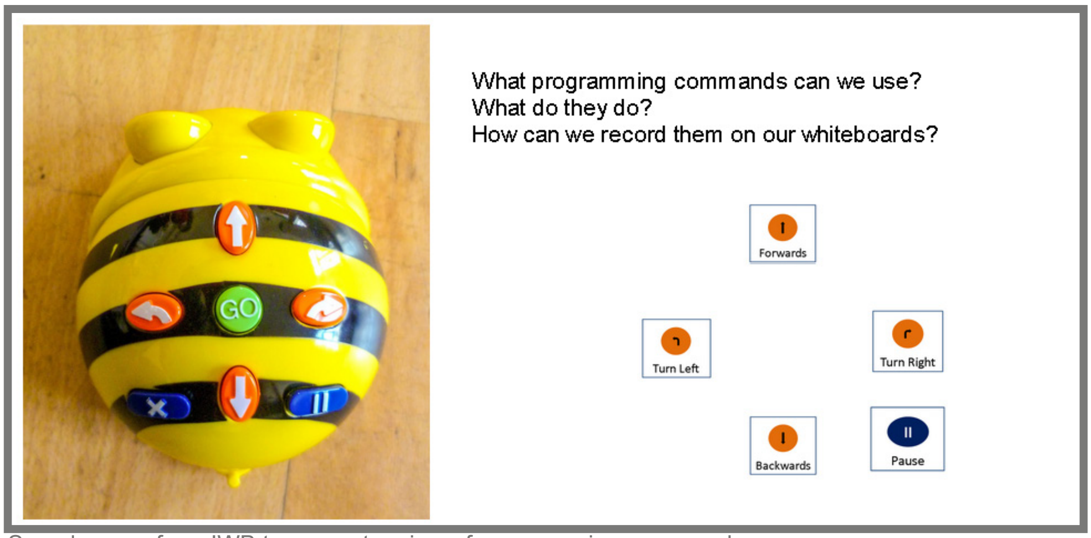
Resources

- pupils' small whiteboards / exercise books and pens
- Bee-Bots (5 or 6 dependent on your classroom organisation – one per pair of pupils)
- ideas for activities for children not working with Bee-Bots (see Before you start in the teaching notes)
- Bee-Bot A4 numeral cards see (see Before you start in the teaching notes)
- teacher's IWB or paper flipchart (adapt to your needs)
- perhaps an 'algorithm designer' hat or band and 'coder' hat or band (optional) (see Before you start in the teaching notes)
- (optional command cards and fakebot cards (see Before you start in the teaching notes)

Introduction Whole Class (5 mins)

- Bring children to carpet with a whiteboard or exercise book and pen each.
- Be 'bossy' and instruct a child to do something e.g. stand up, go to door, open it, come back to carpet place and sit down.
- Say that sequences of instructions are important as they help us to know what to do and how to make things happen.
- Explain the lesson is going to be about programming and how algorithms help us write code. Remind children that algorithms are steps to make something happen and are for people to understand but that programs are for computers . (Children will find out more about programming languages as they progress.) If you have a class chant about algorithms quickly say it together e.g. "Alligator Algorithm, step, step, step, she's very, very bossy but she gets things done, how to make a cup of tea, how to open the door, Alligator algorithm think of some more. "
- Show children a Bee-Bot. Ask children how we program it.

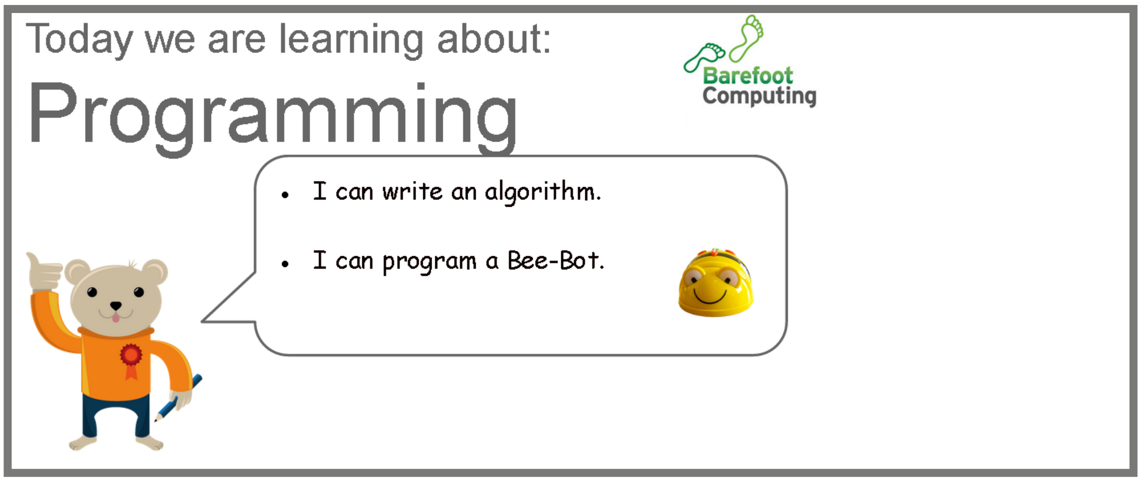
- Ask pupils on their whiteboards or in their books to show symbols of commands they can remember from when they tinkered with the Bee-Bots.
- Ask pupils to show you what they have jotted down and discuss commands they have and have not remembered.
- If necessary, show the image from the presentation of the Bee-Bots buttons and revise the different commands. Otherwise do this when working with smaller groups.



What programming commands can we use?
What do they do?
How can we record them on our whiteboards?

Sample page from IWB to support review of programming commands.

- Share Learning Intentions if that is your normal practice.



Today we are learning about:

Programming

- I can write an algorithm.
- I can program a Bee-Bot.

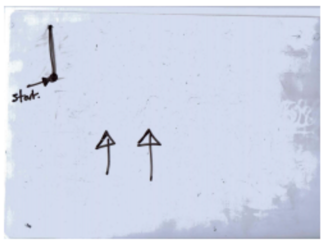
Learning Intention IWB page.

- Depending on how you are organising the class, with the children who will not doing the Bee-Bot challenge this session, share their activity and send off to their independent activities (see Before you start for ideas).
-

Bee-Bot Group explanation (5 mins)

- Do not give pupils the Bee-Bots at this point, keep them in their box or some-where away from children.
- Sit your group in a circle and ask them how we could get the Bee-Bot to write the numeral 1. Write the numeral 1 on a whiteboard to show them. They might suggest forward, forward. Some might start at the top of the number, others at the bottom. Both options are valid.

- The former set of instructions does it in the same way we write the number, so this might be better for reinforcing how to write numbers. Both sequence of steps takes the same number of moves and make the same shape.
- Ask pupils how they could record their plan. Ask them what they call this plan. Hopefully they remember from earlier work on algorithms that this is an algorithm. At this point let all the children have a go at drawing their algorithms on their whiteboards.
- Pop the algorithm hat on if you're using one. Explain you are now going to do the same as them and write an algorithm to draw a 1 shape.

Show them the A4 numeral card	Show how you can jot your algorithm down on your whiteboard. You could include a jotting of the numeral and where you are going to start (top or bottom).	Some children might find using command cards helps (optional)	Try the algorithm by using a fakebot to walk the algorithm through.
		Forwards Forwards	

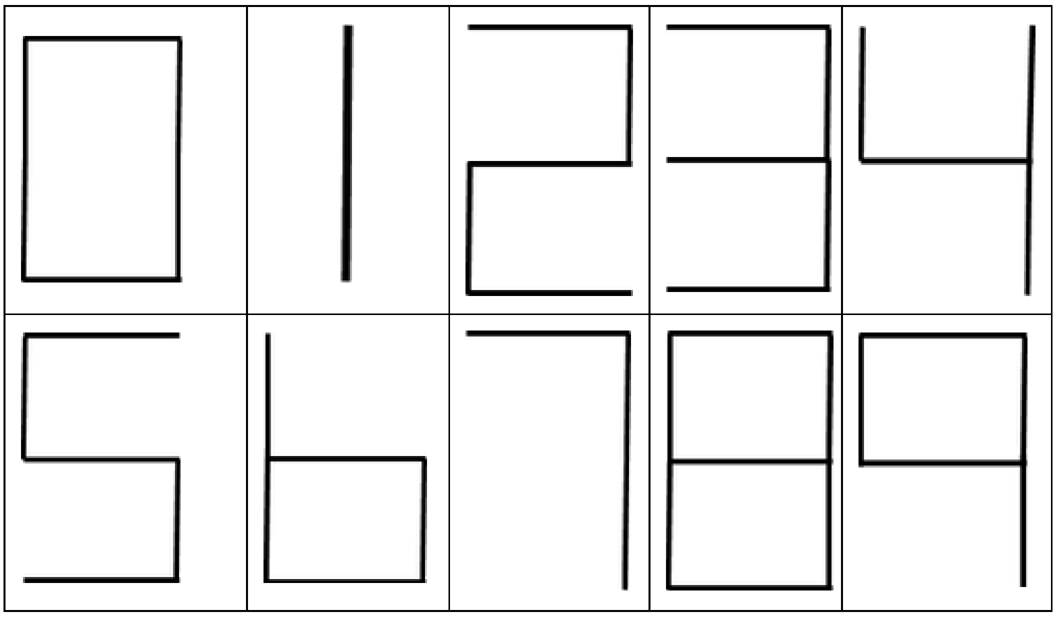
Steps to create and then debug the algorithm.

- Remind children algorithms are for people to understand the steps to solve a problem.
- Explain it is often a good idea to walk through an algorithm before we program it. Stand up and say, 'I start here, facing this way, I take a step forward, and another step forward – yes that makes a number 1'. If you have a fakebot (printed image of a Bee-Bot) show how you can put the fakebot on the numeral card and move it forward, move it forward to make the one. Say 'Now we can program it'. Note you can walk through what should happen as you are writing the algorithm.
- Choose a child to be your partner – your 'coder'. Ask them to pop the coder hat/band on if you are using one.
- Explain to pupils that the only person who should touch the Bee-Bot is the coder, but that they each get a turn when they swap.
- Give the coder your algorithm and the numeral card ask them to use it to type in the commands on the Bee-Bot.
- Ask the children what went well? (e.g. they remembered to press clear, the Bee-Bot went forward) and what could be improved? (This is evaluation although this is not the focus of this lesson, e.g. adding a little symbol to show which way the Bee-Bot should face at the start.)

Group activities (20 mins)

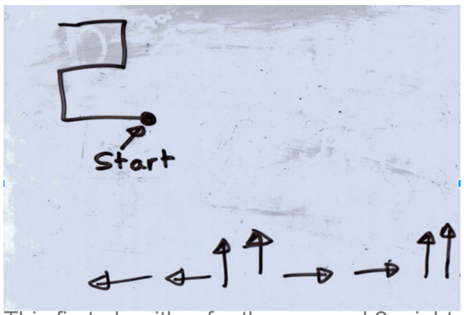
- Choose the pairs and groups for your children for the main Bee-Bot challenge. In each pair, pupils will take it in turns (and swap over roles) to:
    - write the algorithm to solve the challenge (wearing algorithm designer hat) on a pupils whiteboard

- Together they will review their success and try and fix any problems.
- Now say you want the children to work with their partner to do just the same – but for a different numeral. Show children the numeral cards to help them understand how the shapes of numbers need to change for a Bee-Bot to trace them. Ask them where they might have seen numbers written this way before.

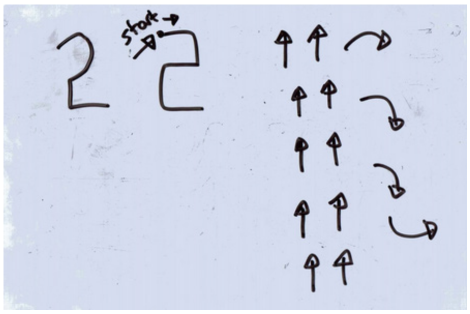

Numeral Card

- This is an opportunity to talk about direction (half turns, quarter turns), and ordering (first, then and next) to link to maths objectives.
- Put children in pairs, based on your normal approach for grouping. Assign who is going to be the algorithm designer and who the coder.The hats can help. Remind the coder they can help the designer when they are working out the algorithm and particularly when they are walking through the steps. Take the whiteboard / exercise book away from coder – as they only need one between them. Remind them they can only have the Bee-Bot once they have an algorithm and have walked through it.
- Give them a numeral card based on the confidence of the pair. 7 is easiest as it needs only one turn, 0 seems to be the next easiest. Some numerals need either a 360 turn or a backwards move (such as 4 and 3). The fun is finding out!
- Some children will be enjoy working out where to start, others may need a suggestion e.g start at the top, and mark on your card. You or your pupils can draw arrows on the numeral cards to help children get started. Don't forget that perseverance and trial and improvement are important aspects of computational thinking.

- Examples of algorithms you might see:



This first algorithm for the numeral 2 might not do at all what the child thinks



This might be closer for the numeral 2, but do we need to start at the top of the commands or the bottom when coding this algorithm?

Normally we read from from the top to the bottom and left to right, but children often build their algorithm starting with their physical location and may add steps from bottom to top. How they record their algorithm is less important than whether they can explain it and use it to program the Bee-Bot.

- Work with pairs to encourage them to think what the steps are. Show them how to jot them down, and walk through them. Give them the command cards if they are finding it tricky to work out the sequence of steps.
- Encourage the pupils to try just 2 or 3 commands before they test it. This can be a useful approach to developing code: to write a part of the program, debug it, write the next part debug it, working on small bite-sized chunks.
- Algorithms that work could be photographed, photocopied, or copied out and put in a class algorithms book or blog.
- Once they program the Bee-Bot to follow the path of the shape of a number successfully, ask the pair to swap roles and have a go at a different numeral. Choose their next number based on how they seem to be getting on under-standing turns, or allow pupils to choose for themselves.

Plenary (5 mins)

[After all groups have had a go at programming]

- Ask each pair to share with another pair one of the programs they created, explaining what was easy and what was hard about creating the algorithm and coding that numeral.
- As a whole class talk about what were the successes, challenges and surprises from the programming activity.
- If time ask children what an algorithm is. Can they remember your chant if you are using one? Eg ‘Alligator Algorithm, step, step, step, she’s very, very bossy but she gets things done, how to make a cup of tea, how to open the door, Al-ligator algorithm think of some more?’ Can you add actions e.g. snapping arms, big step legs, hands on hips, drinking tea etc. Or work out a chant with actions for your class.

Differentiation

Support:
Pupils who find thinking about breaking a problem down into steps (decomposition) to create the algorithm challenging may need to be paired with a pupil who is comfortable with this idea, or adult support might be needed. Provide pupils who find drawing algorithm pictures difficult Bee-Bot direction cards, which they can use to sequence their steps.

Stretch & Challenge:
When breaking the problem down into steps encourage more able pupils to be very precise. Encourage pupils to solve a multi-step problem and to tackle it in distinct parts (decomposing). Encourage pupils to debug as they go along, checking their algorithm first.
Pupils could be asked to return the turtle back to its starting position, facing the original way. This encourages pupils to have broken the problem down into at least two parts: heading out and coming back. Each of these could be thought of as a procedure.
Another option is if pupils are asked to program a number of numerals. Say 3,2,1. To show the way they have 'decomposed' their problem they might adding pauses between each numeral. Each numeral could be considered a procedure. Encourage pupils to think about each part separately. If using a Roamer Too or ProBot, these could be programmed as procedures.

Assessment Opportunities

- Algorithms – Pupils' answers to specific questions and jottings on whiteboards related to set of steps.
- Programming – Pupils successfully complete problem through having a go and changing their programs as they debug it.

Assessment	Questions or evidence
I can write an algorithm.	What is an algorithm? Can you give me an example? What does your algorithm do? e.g. can say an algorithm is a set of precise/accurate/careful/detailed steps to draw a number 3. Can you tell me how it works? How did you check it? *Pupils will have checked their algorithm by walking through it with either pencil and paper or by standing up and doing it.*
I can create an algorithm	How did you use your algorithm?How did you program your Bee-Bot? What did you do? Pupils should have used their algorithm as a basis for their programming. How did you test it worked? *Pupils will have tested their algorithm at the end and shown further progression if they debugged/tested as they went along. For example if drawing an 8, they might have decomposed it into a backwards s first and debugged that before moving on to complete the 8.*

Teaching Notes

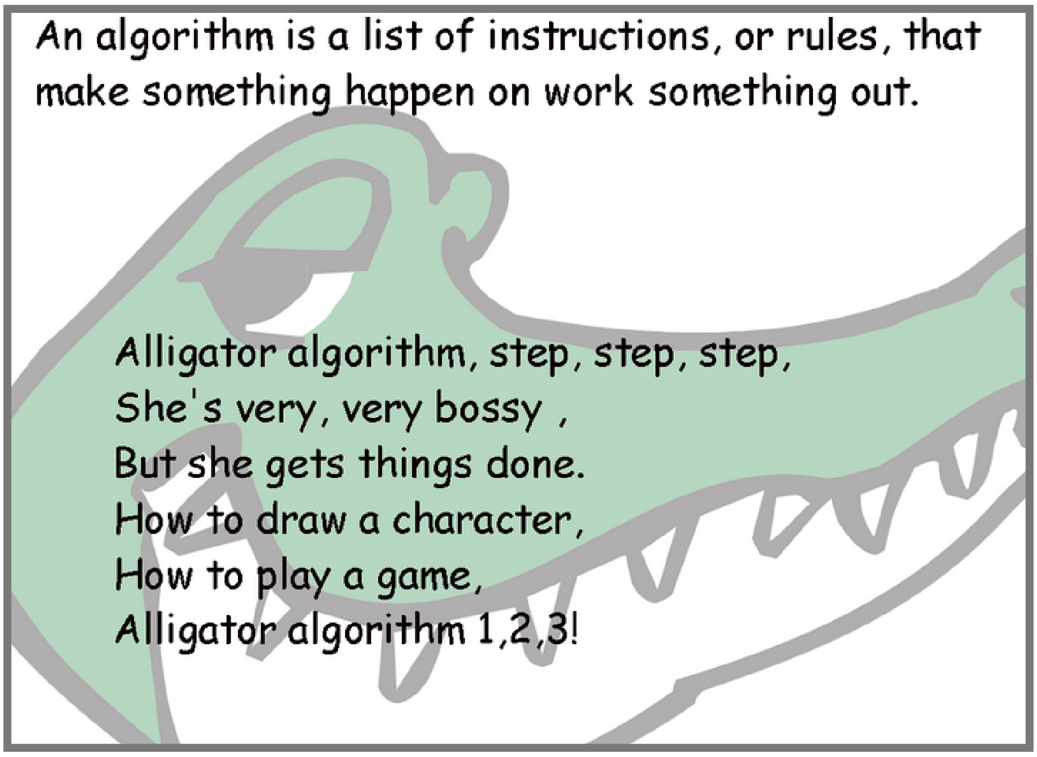
Before you start

- Adapt the IWB_file to your schools format (if using one is your normal practice).
- You might like to think of an algorithm chant and character for your class, or you could use one of these:

  "Alligator Algorithm, step, step, step, she's very, very bossy but she gets things done, how to make a cup of tea, how to open the door, Alligator algorithm think of some more?"

  "Alien algorithm, 1,2,3 tell me how to do it, be bossy. We can make a hat, we can line up. Alien algorithm, 1,2,3 what are the steps, tell me"

  "Albee algorithm, 1,2,3 tell me how to do it, be bossy. We can solve problems, we can work it out. Albee algorithm come on help me!"



An algorithm is a list of instructions, or rules, that make something happen on work something out.

Alligator algorithm, step, step, step,
She's very, very bossy ,
But she gets things done.
How to draw a character,
How to play a game,
Alligator algorithm 1,2,3!

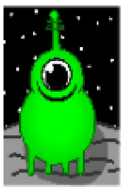
An algorithm chant.
Image from www.pixabay.com Creative Commons Deed CC0

- Choose the pairs and groups for your children for the main Bee-Bot challenge.
- Photocopy onto A4 paper the numeral cards – you may wish to give most pairs a 7 to start with, and then one of the other numerals based on confidence. If so, photocopy around fifteen 7s (assuming you have 30 in your class), a 1 for your demonstration, and then two or three copies of each other number.

- Choose other activities for children who are not doing the Bee-Bot activity e.g.
  - tinker with floor turtle simulators e.g. Daisy the Dinosaur, the Bee-Bot apps;
  - work on maths activities related to ordering and direction;
  - work out algorithms for numerals 0,1,2,3,4,5,6,7,8,9;
  - work out algorithms for letters of the alphabet;
  - work out the algorithm to write their name;
  - sequencing activities such as ordering lifecycles, making instructions (e.g. how to make a lego cat), following instructions (how to make a cube);
  - IT activities such as recording algorithms using a word processor or presentation package;
  - online sequencing and programming activities
    - http://www.iboard.co.uk/teacher/jlisaw8/1
    - http://www.iboard.co.uk/teacher/jlisaw8/2
  - design and create Bee-Bot floor mats and challenges for their peers (e.g. design a maths floor mat, story setting floor mat, or a topic floor mat) with associated locations/objects to navigate between and route challenges. For example pause on all the places Handa's basket was filled, find two numbers that add up to 10, pause on each multiple of 5, pause on all the healthy foods, travel over events from a history topic in the order they happened. Remember if making the mats – Bee-Bot mats are made up of 15cm by 15cm squares. A Bee-Bot transparent grid is useful to layover A3 designs, or you can simply draw the grid on a large A3 card and pupils then add their designs or use transparent pocket mats.

Pupils can make their own floor mats and then set each other challenges

- (Optional – but sometimes useful. Make hats, bands, or cards to signal the roles of each pupil in the group. These can be useful for your roleplay corner or when pupils are tinkering.)

Role	algorithm designer - tailor to your classes chant character for algorithm.	coder- a picture of a Bee-Bot and Scratch cat or other programming tools you use
Image ideas		

A note about duration

To accommodate your working with small groups this activity could be split over a number of sessions – this will depend on the size of your class. Each group session will take approx 30 minutes. Group sizes will vary according to the independence and experience of your pupils, space available and Bee-Bot resources.
As well as say three, 30 minute group sessions you will also need an overall introduction of 10 minutes and 5 minute plenary after all your groups have worked with the Bee-Bots.

A note about fakebots

A fakebot is a printed image of a Bee-Bot or other programmable toys. Pupils can use these to test out their algorithms before they actually code their solution. The design of the fakebots used in this activity are based on a resource created by British Telecoms' education group and are used here with their kind permission. A set of laminated fakebots can be kept with your programmable toys or used in role play areas. Fakebot cards can be downloaded from the bottom of this webpage.

Concepts and approaches

In this activity you and your pupils will learn about programming, debugging, algorithms and persevering. You will also be introduced to decomposition, logic and collaborating.

Decomposition

Decomposition is where pupils break down the task into parts and then into steps. Teaching pupils to think about breaking their problem down into manageable parts is an important concept. For example, if drawing an 8, they might have decomposed it into a backwards s first, and debugged that before moving on to complete the 8. Pupils who just try to program it in one go, with no algorithm or intermediate steps are losing a valuable opportunity to develop decomposition.

Algorithms

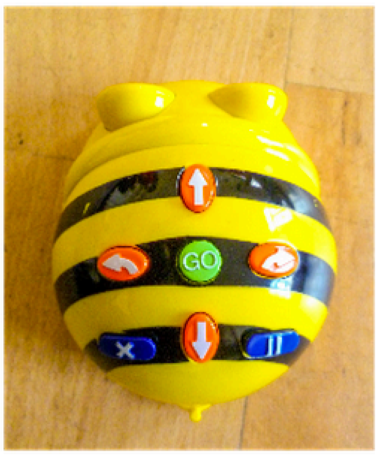
Algorithms are the sets of instructions which children create for Bee-Bots. Algorithm recording here is through informal jottings, words, diagrams that pupils use to work out the route. An important part of algorithm design (and evaluation) is when children start to think about which is the best route. For example if they were making a number 8 they could do it in many ways. All are valid if they create a shape that looks like an 8. But some may use 13 steps, others may use 20. You could then discuss which is the best route? In doing this pupils are starting to evaluate their design.

Programming

Bee-Bots are digital devices. They are programmable toys. They are simple floor robots. As pupils press the keys on a Bee-Bot they are writing their program.The Bee-Bot stores the set of commands and executes them when the GO button is pressed. The Bee-Bot stores only the current program being coded. This program can be executed many times, it is repeatable. Pupils cannot change commands they have already entered but they can add further commands. To edit instructions they have already entered they must cancel the whole program by pressing the clear memory button (x) and re-enter their commands. It is useful for pupils to the steps that they plan to use (their algorithm) before they type it in. They can at least then check back to this, when things go wrong (when they are debugging).

Programming languages

Programming languages enable us to communicate instructions to machines, particularly computers. Programming languages are made up of commands and programs are constructed by joining commands together. The Bee-Bot programming language consists of only a few commands.

The Bee-Bot programming language consists of only five movement commands • forward 150mm, • backward 150mm, • right turn 90 degrees, • left turn 90 degrees, • pause (II) for 1 second and make a tick sound. Plus two device control commands. • clear (X), • GO - executes commands. These are based on a small subset of the Logo programming language.	 When a program has finished the Bee-Bot makes a sound and flashes its lights.

Summary of Bee-Bot programming language

The activity has been written with a KS1 class is mind. The programming language of Bee-Bots isn't sophisticated enough to meet the requirements of the KS2 computing curriculum, although other programmable toys are available which might be appropriate for use in KS2.

Predicting

Predicting is when pupil think what will happen when the algorithm is programmed on the Bee-Bot. They use previous knowledge about what happened already for steps like this. They make assumptions that the same kinds of thing will happen again and so predict. As they use more algorithms to program the Bee-Bots they gain more knowledge and experience and their predictions should become more accurate. This is one aspect of logic and logical reasoning.

Collaborating

Collaborating means working with others to ensure the best result. Paired work in computing is vital to supporting pupils' progress not only in them gaining the underlying knowledge, skills and understanding by discussing and working things out with their partner but also helping them to work with others. Pupils have to work closely in these tasks to set the problem, work out a route to solve it and then program the route.

Persevering

Persevering is when pupils use a trial and improvement approach to create algorithms and programs, particularly for more challenging numerals. For example an 8, where they have to loop back round, or a 4 where they need a reverse or a 360 turn. If pupils are asked to think about algorithm design and work out an algorithm with the least number of steps they will have to try many combinations. You could ask them to work out which numeral has the most number of steps, so that they have to test many algorithms.

Digital devices

Digital devices are any electronic item that process input and produce output according to a stored program. They include:

- mobile devices (e.g. iPads, ipods, tablets)
- recording devices (e.g. talking tins, flipcameras, cameras)
- programmable toys
- control based software and materials (e.g. Lego WeDo, lego mindstorms)
- computers, including laptops, desktops and servers, although strictly speaking all the above examples are computers too!

Alternative to Bee-Bots include BigTrack, Roamer, ProBots to name a few. Some of these alternative devices give a wider range of programming commands e.g. repeats, procedures and even conditional (if…else) constructs that are useful for progression. Another approach would be to program a sprite on screen, perhaps using a Bee-Bot simulator in Scratch or ScratchJr on the iPad.

Debugging

Debugging is the process of detecting and correcting the errors in a program. Bugs happen in programs all the time and therefore debugging is an important skill to have. In this activity as pupils construct their program they are encouraged to constantly test their program after they add new commands and debug it if an error has occured. It is particularly important skill for children to learn that they decompose their problem into parts, and program a part, debug it, and then add the next set of steps. Giving pupils challenges with intermediate stages where they need to pause the Bee-Bot is a great way to encourage decomposition and debugging as you go skills.

Further reading

There are a wealth of resources to use with Bee-Bots online
http://www.tes.co.uk/teaching-resources/primary-40069/ks1-ict-41488/controlling-and-modelling-41489/floor-turtle-41491/
If you are using pro bots:
http://www.simonhaughton.co.uk/pro-bot-lessons/
BBC KS1 How do you program a robot: http://www.bbc.co.uk/guides/zqnc4wx

Related activities

KS1 Bee-Bots Tinkering activity