# Year 1 Programming 1 Move my Bee-Bot (Core)

| | |
|---|---|
| **Timing**<br>3 sessions of approximately 40 minutes | **Children will**<br>• Meet Bee-Bot and think about what they may already know<br>• Give instructions to each other in pairs<br>• Use counters to investigate how far Bee-Bot moves with one forward press and how much Bee-Bot turns with the right turn or left turn buttons<br>• Draw their own 'town' on a large sheet of paper and plan routes for Bee-Bot<br>• Self-assess their confidence to implement their algorithm as a program<br>• Use the pause button when traffic lights are added to their town<br>• Use the word algorithm (it is not expected they fully understand the word at this stage)<br>It is assumed that children will have had previous open ended play opportunities with the Bee-Bots. They will create their own problems through constructing 'play areas' for the Bee-Bots and programming the Bee-Bots to move around them. |
| **e-safety links**<br>• I am careful with technology devices (R)<br>• I can talk about why it's important to be kind and polite | **Objectives**<br>**Programming**<br>• I can give instructions to my friend and follow their instructions to move around.<br>• I can describe what happens when I press buttons on a robot.<br>• I can describe what actions I need to do to make something happen.<br>• I can begin to predict what will happen for a short sequence of instructions.<br>• I can use the word debug when I correct mistakes when I program. |
| **Links to other learning**<br>**Mathematics:** support problem solving, direction and turns, shape, counting moves | |
| **Resources:**<br>BeeBots / BlueBots / other floor robot<br>Large sheets of | **Preparation**<br>• Watch Phil Bagge's video showing development of understanding with Bee-Bots [https://www.youtube.com/watch?v=cTxZbIELt1U](https://www.youtube.com/watch?v=cTxZbIELt1U). This video is acknowledged as contributing to sessions two and three.<br>• The first session uses the idea of Direct Drive (UCL ScratchMaths) to check children's understanding of what each button does. Using buttons individually is a key step towards purposefully combining buttons in a sequence. |

Thanks to Phil Bagge Code-It for ideas for sessions 2 and 3

| | |
|---|---|
| paper with grids<br>Counters/cubes<br>KS1 Self-assessment poster | • Draw out a grid with 15cm x 15cm squares on large sheets of paper – one for each group<br>• Have a copy of the algorithm chant ready for display. Could use Algorithm rap PowerPoint.<br>*Algorithm, algorithm*<br>*What do we need to do?*<br>*Algorithm, algorithm*<br>*How will we do it?*<br>• Print out 'I can debug' stickers from Education Technology website<br>• Download, print, laminate and cut out Bee-Bot instruction cards from TTS resources<br>• Display KS1 Self-assessment poster<br>You could use I Am a Computer Programmer document to support you in developing the language of programming. |

Phil Bagge advises:

- Set a rule that the children cannot pick up Bee-Bot once they have pressed go. They must wait for Bee-Bot to get to the end of the program. This encourages children to think more carefully and plan the buttons they will push.

- Encourage children to take it in turns to press buttons.

| | Expectations | Activity | Success Criteria |
|---|---|---|---|
| 1 | **Programming**<br>I can describe what happens when I press buttons on a robot<br>I can give instructions to my friend and follow their instructions to move around. | **Bee-Bot direct drive**<br>• Where possible have children working in pairs with a Bee-Bot or Blue-Bot. Allow them initial 'time for play' and observe the level of confidence. Bring children together as required to refresh:<br>○ Can they turn it on?<br>○ Do they use clear button to start again?<br>• Ask children, '*What can Bee-Bot do? How do we make Bee-Bot move? What happens when we press the different buttons?'* Answers will be a baseline to see increase in confidence to answer at the end of the session. Check children can use the clear button.<br>• Give the children two counters/cubes to use to see the measurement of one press of the forward arrow.<br>• **Predict:** Place one counter/cube in starting position. Clear Bee-Bot. Where do you think Bee-Bot will stop when you press forward button once? Place second counter/cube where you think Bee-Bot will stop.<br>• **Run:** Put Bee-Bot next to first cube. Let children press forward button and Go. How close does Bee-Bot get to the stop position counter/cube?<br>• Encourage children to look to see how far Bee-Bot goes with one press of the button.<br>• Let them have another go to see if they want to move the stop position counter/cube.<br>**Direct drive through a tunnel**<br>• **Make**: In pair / three / four, 'Can you make Bee-Bot go through the tunnel?' One child is in charge of Bee-Bot, other child(ren) make tunnel by standing with legs apart. Which button will make the Bee-Bot go forward through the tunnel? Swap | Gold: Can I remember to use the language of GO, CLEAR and PAUSE in my instructions?<br>Silver: Can I follow and give a short sequence of instructions?<br>Bronze: Can I follow and give instructions to make my partner move forward and backward, and turn left and right? |

Thanks to Phil Bagge Code-It for ideas for sessions 2 and 3

around for everyone to have a turn controlling Bee-Bot.

- Do they use forward button? This can be repeating one press of the forward button and then go; or could be a number of presses of the forward button and then go.
- Can children make Bee-Bot go back through the tunnel? Some may turn the floor robot around and press go. Initially, let them do this.
- Can they make it go back through the tunnel without picking it up and turning it around?
- Can they make it go forwards and backwards through the tunnel?
- Can they make it go forwards, pause and then go backwards?

**Direct drive turning**

- Can children use the two counters to see how much Bee-Bot turns when a turn arrow is pressed? (See Phil Bagge video). Place one counter/cube in starting position. Clear Bee-Bot. Press turn button once. Place second counter/cube where Bee-Bot stops. Encourage children to look to see how far Bee-Bot turns with one press of the button. What do we call this? (quarter turn / right-angle)
- Repeat for turning the other way. Do children recognise it is still turning a right-angle.

Ask questions again, *What can Bee-Bot do? How do we make Bee-Bot move? What happens when we press the different buttons?* Ensure children understand that the arrows represent a quarter turn, that you need to press CLEAR before entering a new instruction, and that a forward/ backward move is a set distance (a constant measure).

**Becoming Bee-Bots**

- Take the children to the hall/outside – say they are Bee-Bots and you are going to give them instructions to follow (stress it doesn’t matter what direction they start facing – they just need to follow the instructions from their starting point). Give

Thanks to Phil Bagge Code-It for ideas for sessions 2 and 3

| | | | |
|---|---|---|---|
| | | them a range of instructions to follow – build from single instructions to a short sequence. **Stress their forward/backward move needs to always be the same size! Use the language of GO and CLEAR.**<br>• Allow a child to have a go at issuing instructions/commands to whole class, then explore in pairs. Set up some 'stations' (e.g. with toys/games equipment) – in pairs, children give instructions to move to a station, PAUSE for a short period of time, then CLEAR and move onto another. | |
| 2 | **Programming**<br>I can describe what actions I need to do to make something happen.<br>I can use the word debug when I correct mistakes when I program. | **Bee-Bot home**<br>This session based on Phil Bagge video.<br>• Each group to have a large sheet of paper with a grid of 15cm x 15cm squares. Ask them to decide where Bee-Bots home is and to draw it on the paper.<br>• Which way should Bee-Bot face?<br>○ Be ready to prompt children to decide that Bee-Bot shouldn't face diagonally on a square as the distance across the diagonal of a square is longer so Bee-Bot won't get across the square.<br>○ Does Bee-Bot go from square to square if facing diagonally? Give children time to try this out.<br>○ Talk about Bee-Bot needing to face horizontally or vertically.<br>• Tell the children to add other locations on the paper. Set the rule that the locations must not be adjacent to each other. Make sure children understand this rule. Locations could connect at the corners but not along the side of a square. Get the children to draw the locations lightly with a pencil so that you can check they have stuck to the rule before they colour in the pictures. Emphasise that it is okay to make mistakes. Use the word debug when the children sort out any mistakes.<br>• Add roads to the town. These are lines that link squares together and are drawn through the centre of squares. Make sure children have remembered the roads | Gold: Can I explain why Bee-Bot needs to face horizontally or vertically and not diagonally? Can I watch a program carefully and debug any mistakes?<br>Silver: Can I make sure Bee-Bot faces horizontally or vertically on my mat? Can I put to the appropriate instruction card as Bee-Bot moves through the program?<br>Bronze: Can I make Bee-Bot face horizontally or vertically when I start each program? |

cannot be diagonal lines. Let children have an opportunity to program Bee-Bot to move around their town. Can Bee-Bot go the shops? Can Bee-Bot go from the park to the school? Use stickers to encourage the children to use the word debug when they need to sort out a program that doesn't do what they want.

- Ask the children why it can be difficult to debug a program. Expect answers such as, 'It is difficult to remember which buttons we pressed.'
- Show the children a set of program cards. Model how you can 'write' a program before inputting it with the buttons on Bee-Bot.
- Set the children another problem. Ask them to talk through the algorithm. Take children's ideas for the algorithm. Record in words or short (moving forwards or backwards a short distance) / long arrows (moving forwards or backwards a long distance) and turning. Do they know how to make each part happen? Could they write a program with the cards? Model using KS1 Self-assessment poster to check you are ready to program. 'I don't know what to do yet.'
    - Talk with my friends, can any of the children help you?
    - Act it out, show the children how you can pretend to be Bee-Bot to check out the moves you will need to do.
    - Try something; see what happens; decide that you will have a go.

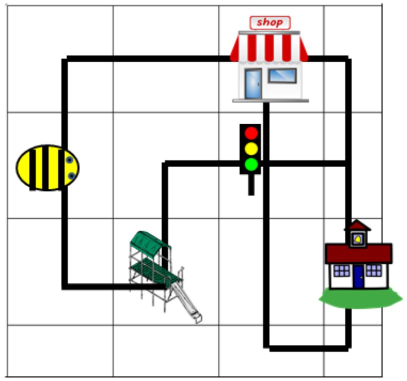
- Model implementing the algorithm as a program by sequencing the appropriate instruction cards. Get one of the children to point to each card as you input the program into Bee-Bot.
- Run the program by pressing Go. Model pointing at each card as Bee-Bot make that movement.
- Set more problems for the children. Working them towards a fluent process of
    - Talk through the algorithm.
    - Do they know how to make each part happen? – use self-assessment poster

| | | | |
|---|---|---|---|
| | | ○ Use the instruction cards to write a program.<br>○ Input the program on Bee-Bot.<br>○ Execute the program by pressing Go.<br>○ Encourage the children to watch carefully as the program runs with one of the children pointing to each instruction.<br>• Ask the children whether they have been able to more easily spot when Bee-Bot makes a mistake.<br>• Encourage the children to debug their program when there is a mistake by handing out Debug stickers. Keep emphasising how important it is to make mistakes which help you to solve the problem.<br>• Ask children to name their towns. Tell them we will use them again in the next session. | |
| 3 | **Programming**<br>I can describe what actions I need to do to make something happen and begin to use the word algorithm. | **Bee-Bot town with traffic lights** (This session based on Phil Bagge video.)<br>• What could you add to your town that might slow Bee-Bot down? eg traffic lights, forest. Get children to use PAUSE button when Bee-Bot arrives at the traffic lights to other feature they have added. Let children have time to include pauses as they program Bee-Bot.<br>• Ask children how Bee-Bot would need to move to get from home to school via the shops? Get them to talk about this in pairs. Expect children to say things like, 'We'll start Bee-Bot facing vertically. Bee-Bot will need to go forward and then turn right. Bee-Bot will go forward again and pause at the traffic lights. Bee-Bot will need to turn left and go forward until he/she gets to the school.<br>• Tell children they have been talking through the algorithm to solve the problem.<br>• Introduce the children to the algorithm chant: (The claps could be done as if the children are the jaws of an alligator – we'll be alligators saying the algorithm rap.)<br>*Algorithm, algorithm (fast with fast hand claps – a clap for each syllable)*<br>*What do we need to do? (slower with slow hand claps – clap on word what,* | Gold: Can I talk about the sequence of things that will need to happen to solve a Bee-Bot problem and call this an algorithm?<br>Silver: Can I talk about the sequence of things that will need to happen to solve a Bee-Bot problem?<br>Bronze: Can I make Bee-Bot move from one location to another? |

Thanks to Phil Bagge Code-It for ideas for sessions 2 and 3

*do, need and do)*
*Algorithm, algorithm (fast with fast hand claps – a clap for each syllable)*
*How will we do it? (slower with slow hand claps – clap on word how and do)*

- Set each group a problem to solve. Remind them of self-assessment poster to consider when they have a new problem.
- Give each group a set of program cards to write the program they will input into Bee-Bot. Make sure they always begin by talking through the way Bee-Bot will need to move; using self-assessment poster to consider whether they know how to make that happen, before they 'write' and input a program.
- Use PowerPoint slide 2 to set problems for children to talk through with their partner. Children could use a wipeboard or jottings on paper to show their algorithm.



**Plenary review**

- What are the new words you have learned? Make a list and then get children to check each other's knowledge in pairs:
  - Algorithm (can the children think of anything else they do in school where they use an algorithm – could be maths, phonics, problem on the playground etc)
  - Program
  - Debug
- Rehearse the self-assessment poster with the children. Talk about when they could use it in computing and in other areas of learning.

Thanks to Phil Bagge Code-It for ideas for sessions 2 and 3