

| | |
|--|---|
| <p>Timing</p> <p>4 sessions of approximately 40 minutes</p> | <p>Children will</p> <ul style="list-style-type: none"> • Give Bee-Bot instructions to move around a space to reach a toy (can be physical objects or 15cm x 15cm picture representations used with a Bee-Bot mat or laminated images for durability). Children begin by inputting one instruction at a time, then progress to building up a sequence – use Bee-Bot cards instruction cards if available to talk through and represent instructions needed. • Explore debugging when Bee-Bot doesn't reach intended toy. • Guess my toy - Talk aloud inputting a sequence of instructions into Bee-Bot to reach a particular toy (or represent using cards) – children predict where Bee-Bot will land. • Use Kodable activity to describe route to pick up stars, write and run a program and then design own route. • Navigate Bee-Bot around a racing track/town mat (download free example here if needed) by programming series of instructions. • Use 2Go in 2Simple Infant Video Toolkit or Purple Mash and select car track background to complete activity on-screen OR JIT5 Town map for children to play a route to walk around the town. As with Bee-Bot, '1' turn = a quarter turn. |
| <p>e-safety links</p> <ul style="list-style-type: none"> • I am careful with technology devices (R) • I can talk about why it's important to be kind and polite | <p>Objectives</p> <p>Programming</p> <ul style="list-style-type: none"> • I can give instructions to my friend and follow their instructions to move around. • I can describe what happens when I press buttons on a robot. • I can describe what actions I need to do to make something happen. • I can begin to predict what will happen for a short sequence of instructions. • I can begin to use software/apps to create movement and patterns on a screen. • I can use the word debug when I correct mistakes when I program. |
| <p>Links to other learning</p> <p>Mathematics: support problem solving, directions and turns, shape, counting moves</p> | |

| Resources | Preparation |
|---|---|
| <p>BeeBots / BlueBots / other floor robot</p> <p>Kodable activity</p> <p>Infant Video toolkit 2Go / PurpleMash OR JIT www.j2e.com/j2code (no cost unless you want to save work)</p> <p>Algorithm rap Debug stickers BeeBot cards Self-assessment poster</p> | <ul style="list-style-type: none"> • Have a copy of the algorithm chant ready for display. See Algorithm Rap PowerPoint slide 1 <i>Algorithm, algorithm</i> <i>What do we need to do?</i> <i>Algorithm, algorithm</i> <i>How will we do it?</i> • Create a selection of images of (up to 20) toys (each image sized to fit within a 15cm x 15cm square). • Create a 4x5 'mat' from your images (use blank squares if you don't want an image in every square) – <i>if available, use with the TTS Bee-Bot transparent grid mat for durability (or laminate squares if desired) – otherwise, just stick the sheets together.</i> • Add duplicate set of toy images to cards for 'Guess my toy' game • Check KS1 self-assessment poster is displayed • Print out 'I can debug' stickers from Education Technology website • Download, print, laminate and cut out Bee-Bot instruction cards from TTS resources • We are suggesting use of Kodable Maze Maker online. This can be used on any device without a log in. The Kodable app requires children to log in or use a class code. Ask technician to create a desktop shortcut for Kodable activity on laptops or 'Add to home screen' for tablets. NOTE: We have found that a QR code doesn't always take you directly to the correct part of the Kodable website. The hyperlink above takes you directly to the screen where children choose Kindergarten or First Grade. Alternatively go via https://game.kodable.com/hour-of-code#maze-maker Play without saving/Playing at School/Kindergarten or First Grade. |

| | Expectations | Activity | Success Criteria |
|---|--|---|---|
| 1 | <p>Programming</p> <p>I can describe what happens when I press buttons on a robot.</p> <p>I can describe what actions I will need to do to make something happen and begin to use the word algorithm.</p> | <p>Bee-Bot Guess My Toy</p> <ul style="list-style-type: none"> • Set up Toy 4x5 ‘mat’ (see details in preparation section above) • Play Guess My Toy – talk aloud inputting a sequence of instructions (program) into Bee-Bot to reach a particular toy (if possible represent with Bee-Bot instruction cards). <i>Where do you think Bee-Bot will land?</i> Paired talk to predict – take suggestions. Make clear that the mat helps us as 1 forward move = 1 square on the mat. <i>What button do I need to press to execute my program?</i> (Go). Press and see where Bee-Bot end up (correlate moves with command cards if using). Repeat – clearly model pressing CLEAR so Bee-Bot ‘forgets’ the instructions in his memory. • Ask a child to select a toy card (from a duplicate set of the toy cards used to make the mat). <i>How can I make Bee-Bot move from where I am now to this toy?</i> Talk about the sequence (algorithm) needed to solve the problem. Use the algorithm rap to remind children of this word. • Do I know how to make this happen? Remind children of the KS1 self-assessment poster. ‘Write the program/code’ using instruction cards. • Press buttons to input the program (pointing to each instruction card as they go) and execute/Run the program/code (either step-by-step or with several instructions/the whole program at once, depending on children’s confidence). • In groups, children repeat this activity; selecting a toy card, discussing the algorithm needed, ‘writing the program/code with the cards’, entering and running the program to move Bee-Bot to the correct toy. • Model thinking through algorithm to move Bee-Bot to a toy, making a deliberate mistake. Do the children interrupt to correct you? Let the children guide you to the correct algorithm. • Model writing a program/code with cards (stick them to board in a sequence), including a mistake. Read the program/code together. Ask the children to agree | <p>Gold: Can I begin to identify an algorithm to achieve a more complex route?</p> <p>Silver: Can I begin to identify an algorithm to make Bee-Bot move to a certain square?</p> <p>Bronze: Can I press a sequence of buttons on Bee-Bot to make it move to a certain square?</p> |

| | | |
|---|---|---|
| | <p>(predict) in pairs what will happen. Input the program/code and run the program. Do they spot the mistake? Can they help you debug your program/code? Input the debugged program/code and check Bee-Bot now moves to the toy.</p> <ul style="list-style-type: none"> • <i>For added challenge, ask children to select additional cards and ask them to control Bee-Bot to reach target toy by travelling via the additional toys on the way.</i> • iPad extension – Encourage children to complete challenges using Bee-Bot App or Blue-Bot App | |
| <p>Programming</p> <p>I can begin to predict what will happen for a short sequence of instructions.</p> <p>2 I can describe what actions I will need to do to make something happen and begin to use the word algorithm.</p> | <p>Kodable activity</p> <ul style="list-style-type: none"> • Introduce the children to Kodable activity Kindergarten or Grade 1 level. • What do you think you have to do? How do you think you control Fuzz? How is Kodable the same/different from Bee-Bot? • Refer to self-assessment poster. How confident are the children to have a go? • Model saying what Fuzz will need to do to pick up stars for level 1. Move right / up / down. This is my algorithm. Say it again with the children. • Model inputting this as a program/code. Read the program/code with the children. Run the program/code. Did it do what we wanted it to do? • Children work in pairs. Can they say the algorithm/tell their partner how Fuzz will have to move to pick up stars? Then program/code Fuzz to pick up the stars. • Pause the activity and model making a mistake with level 3. <i>‘My program/code isn’t working. Fuzz can’t get to the end. ‘Can you help me debug my program?’</i> Children will need to Read the code that has been written and Predict what will happen. Can they spot where it goes wrong? <i>‘What do I need to change?’</i>  | <p>Gold: Can I create my own maze for a partner to solve?</p> <p>Silver: Can I say the sequence of moves needed for Fuzz to pick up the stars?</p> <p>Bronze: Can I say the moves for Fuzz, one instruction at a time?</p> |

| | | | |
|---|---|--|---|
| | | <ul style="list-style-type: none"> • Children to continue through five levels – say the algorithm, write the program/code, run the program and debug if any mistakes made • Challenge: Use the design maze feature to create their own route. They need to have a start (click on a square in the first column on left) and finish (click on a square in the last column on right). Now they can create the pathway, add the three stars and Preview. Children may need to go back to edit/debug their maze if there are too many turns. They only have eight spaces for their program/code. • Swap and complete each other’s mazes: think of the algorithm, write the program/code and run the program/code. • Show children a maze you have made with the program/code completed incorrectly. Read the program/code together. Ask the children in pairs to read the program/code again and predict what will happen. Will Fuzz pick up all the stars and get to the finish? Run the program. Does it do what they predicted? <i>‘Can you help me debug my program?’</i> Assess the children’s confidence to read, predict, spot mistake and debug. • What have we learnt? How did we learn it? Encourage children to think about how they tried things out and learnt from mistakes. • What is an algorithm? We think through what we need to make happen. Say algorithm rap together. • What is a program/code? We write our program/code so that the computer knows what to do. We write it in a language the computer understands. | |
| 3 | <p>Programming</p> <p>I can describe what actions I will need to do to make something happen and</p> | <p>Bee-Bot race track</p> <ul style="list-style-type: none"> • Prior to lesson, download and assemble Early Learning HQ race track background to create a Bee-Bot race track mat (see here for what the assembled mat should look like). If available, use with the TTS Bee-Bot transparent grid mat for durability – otherwise, just stick the sheets together. You can also add a racing car ‘jacket’ to the Bee-Bot. | <p>Gold: Can I accurately use the word debug when suggesting changes to make to a program?</p> <p>Silver: Can I predict the correct direction of turns for Bee-Bot to make in a sequence?</p> |

| | | |
|---|---|---|
| <p>begin to use the word algorithm.</p> <p>I can use the word debug when I correct mistakes when I program.</p> | <ul style="list-style-type: none"> • <i>What is the route for Bee-Bot to move around track? How many turns will Bee-Bot need to make on its way round the track? What direction?</i> Ask the children to talk through the algorithm to make the Bee-Bot go round the track. Do they know how to make this happen? Get children to think about self-assessment poster in pairs. Are they confident to have a go? Identify those that may need further support. • Use Bee-Bot instruction cards to write the program needed to make the Bee-Bot move from start to finish (otherwise note using arrow notation on whiteboard). Read the program/code together and predict what it will make happen. Program the Bee-Bot to execute the sequence and observe what happens – if Bee-Bot doesn't reach the finish, look back at sequence and debug – <i>where did we make a mistake? What change do we need to make to our algorithm?</i> Clear and execute again. • Using symbol cards, spread the 6 cards face down. Select child to roll a dice to determine how many cards to pick. Select children to pick required number of cards and position on race track. Discuss what the cards represent, e.g. flat tyre – 2 x pause – <i>what will our sequence look like now?</i> Amend original sequence to incorporate pauses – execute and check that pauses happen on correct squares and for correct duration. Debug as needed. Repeat execution to time how long it takes Bee-Bot to complete track. As time allows, repeat activity, choosing new cards. <i>Do you think this sequence will be faster or slower? Why?</i> • What have you learned? How did you learn it? Encourage children to think about how much they learned from each other and from the mistakes they made during the activity. | <p>Bronze: Can I begin to predict a sequence of moves to help Bee-Bot achieve a goal?</p> |
| <p>Programming</p> <p>4 I can begin to use software/apps to create movement and patterns on a</p> | <p>2Go Racing Track or JIT5 Town (This is written for 2Go but JIT5 Turtle Town background can be used for children to plan a route around Town.)</p> <ul style="list-style-type: none"> • Load 2Go (from the Infant Video Toolkit or PurpleMash). Click on the 'new page' icon and select the racing track background. <i>How is this the same/different from using Bee-Bot? What do you think we need to do? What do we need to find out to</i> | <p>Gold: Can I begin to accurately predict the distance required to make the car move forwards in one command?</p> <p>Silver: Can I accurately use the word</p> |

| | | |
|----------------|---|---|
| <p>screen.</p> | <p><i>help us achieve this outcome? (What distance is covered by '1'?).</i></p> <ul style="list-style-type: none"> • Move the car so it is on the chequered line (drag with mouse, turn with keypad) – agree what our first move should be. • Ask the children to talk through the algorithm to make the car go round the track. Refer to self-assessment poster. Do they know how to make this happen? • Ask a child to input on IWB and see what happens (debug as needed if children didn't make their first move a turn to make car face the right way! • Draw attention to the car icon in the centre of the arrows on control pane – <i>this reminds us what direction the car is facing.</i> Work through instructions to move car around track as efficiently as possible (not just repeated moves of forward1). • When they have turned, debug as necessary if the 'forward' instruction results in a move backward. <i>Why has this happened? What direction are we moving in? What way is the car facing now?</i> • Ask children to explore moving the car around the track on their own/in pairs. • After what you judge to be sufficient exploration time, ask the children to move their car back to the start. Show them how to select a pen to mark the route they travel. Ask them to make the car move from start-finish with the route traced to show the accuracy of their instructions. • Model making a mistake. Ask children to read and predict what you are making happen. Can they spot where the mistake is? Can they help you debug? • What have you learned? How did you learn it? Encourage children to think about how much they learned from each other and from the mistakes they made during the activity. | <p>debug when I need to make corrections to my car's movements?</p> <p>Bronze: Can I explore making a car move around a track on the computer?</p> |
|----------------|---|---|