Reception : Introduction to programming through guided Bee-Bot play

Program Aim: Pupils need lots of opportunities to play and explore using the Bee-Bots in reception before they move onto KS1 algorithmic activities. Playing helps pupils really understand the logical rules the device is constrained by. Too little play and they will struggle later on. You can tell children that the Bee-bot turns without moving forward or backwards but they won't fully understand that until they have explored it and played with it and worked it out by themselves.

This module is followed by a 6 week module in Year 1 and a 5-6 week module in Year 2.

Step 1: Unstructured play

Bee-bot programming devices need to be available for pupils to explore. They need to be accompanied by the NO PUSH cards.
http://code-it.co.uk/wp-content/uploads/2015/05/prog_not_push.pdf

Step 2: Guided play

Guided play activities ensure that, pupils with less inclination to explore playing with the Bee-bot programming device, get something meaningful from their interactions. These should always only be a starting point and if pupils come up with extension activities, or new challenges these should be encouraged. Teachers need to record the following activities onto a talking postcard or talking tin and leave these with the Bee-bot. They are not meant as an exclusive list but as a starting point that reception teachers can use to design their own activities. Teachers may wish to write the instructions as well so pupils can choose to read, listen or both. All of these activities are designed for individuals but can easily be used by pairs or pupils working collaboratively. Recording and assessing can be done through discussion with adults or photographs to capture the learning.

Activity: Bee-Bot home	Activity: To the wall
Resources: - Bee-bot, bricks, talking tin, optional activity card Can you build a home for Bee-bot? Can you program Bee-bot to enter its home and leave without knocking the bricks over?	Resources: - Classroom wall, Bee-bot, talking tin, optional activity card Can you program Bee-bot to move close to the wall without hitting it? Can you program it to return?

Activity: Figures

Resources: Three figures or animals that stand up, Bee-bot, 30cm ruler, talking tin, optional activity card Stand up three figures around Bee-bot one ruler length away. Can you program Bee-bot to visit each figure from its starting position?

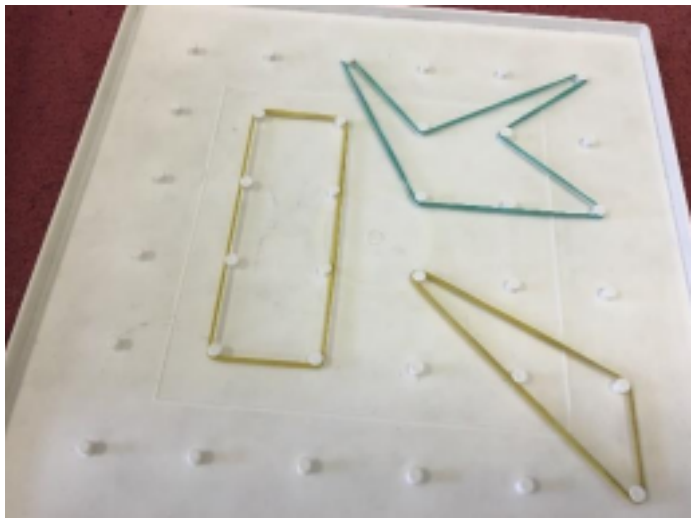
Activity: Train track

Resources: - 6 lengths of train track, 2 straight and 4 curves, Bee-bot, talking tin, optional activity card Build a train line using all the tracks given. Can you program Bee-bot to follow the train line next to it?

Notes: Bee-bot turns on the spot in 90 degree arcs. Train tracks make more gradual turns. Can the pupil tell you or show you why Bee-bot has difficulties following the train track?

Activity: Peg board

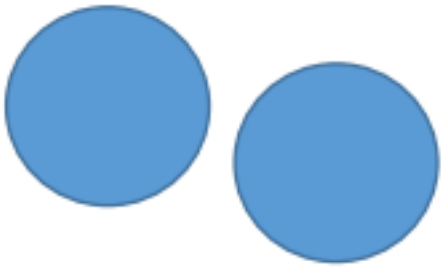
Peg Board Resources: Peg Board, elastic bands, Bee-bot, talking tin, optional activity card Can you create a shape on the peg board that Bee-bot can make on the floor? Can you create a shape on the peg board that Bee-bot can't make on the floor?



Note: If you have Blue-bots these can be programmed to turn 45% if using the Blue-bot app on a tablet computer but pupils are unlikely to have encountered this yet.

Activity: Distance

Resources: Counters, Bee-bot, talking tin, optional activity card Does Bee-bot always go the same distance each time it is programmed?

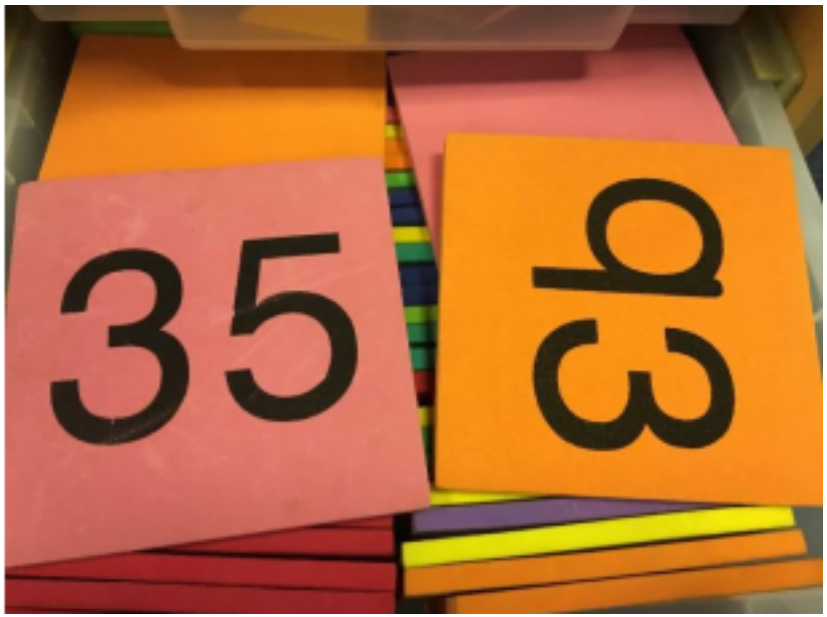


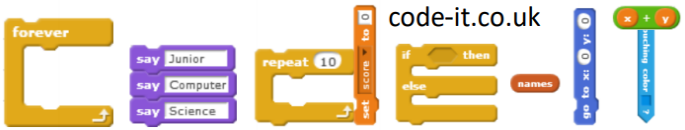
Activity: Turn

Resources: Counters, Bee-bot, talking tin, optional activity card Does Bee-bot always turn the same amount every time it is programmed to turn right or left?

Activity: Highest number

Resources: Number Cards, 30 cm ruler, Bee-bots, talking tins, optional activity card Choose two or three cards. Place them one ruler length away from Bee-bot. Can you program Bee-bot to move to the highest number?





Where next?

Explore other programming planning at code-it.co.uk/csplanning
Design your own programming planning using computational thinking & problem solving skills.

Author: Philip Bagge
Find more resources at code-it.co.uk

Follow me on twitter @baggiepr

Designer: Amanda de Fraine
adefraine@hotmail.com

Follow me on LinkedIn

Complexity	I can break complex problems into parts	I can discover / concentrate on the most important part of a problem	I can explain how I used decomposition & abstraction
Ambiguity	I recognise there is more than one way to solve a problem	I recognise there is more than one way to describe a problem	I can explain how I managed ambiguity
Open Ended	I look for a range of solution to the same problem	I don't just accept the first solution	I can describe how a project can be extended
Adapt	I can adapt existing ideas to solve new problems	I can identify patterns in problems & solutions	I can explain how I adapted a solution to solve a new problem
Evaluate	I can evaluate my solutions against a set criteria	I can design criteria to evaluate my creations	I can explain how evaluation helped me improve a project
Experiment & Debug	I can develop, test and debug until a product is refined	I repeatedly experiment through making, testing & debugging	I can explain how using the iterative cycle improves my work
Persist	I can persevere even if the solution is not obvious	I learn from setbacks and don't let them put me off	I can describe how I overcame problems
Communicate	I can contribute useful ideas to a partner or group	I can encourage others to share their ideas	I can lead using all the people talent in my group

Decomposition Abstraction Generalisation Algorithmic Evaluation Algorithm

Adapted from Behaviour Rubric created with @MarkDorling and linked at http://code-it.co.uk/attitudes/ Adapted with permission

This work is licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/3.0/ or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA