Early Childhood Education Journal (2022) 50:969–981
https://doi.org/10.1007/s10643-021-01236-8

Can Pre-school Children Learn Programming and Coding Through Guided Play Activities? A Case Study in Computational Thinking

Valerie Critten[1] · Hannah Hagon[1] · David Messer[1]

Accepted: 23 June 2021 / Published online: 3 July 2021

Abstract

Guided play activities were developed so that coding clubs could promote computational thinking skills in preschool children. The clubs involved fifteen children aged between 2 and 4 years, including a group of children with communication difficulties. The children took part in an action-research scoping study over three coding clubs involving six 45–60-min sessions. The activities were developed to teach computational skills and, ultimately, concepts of programming and coding. The findings suggested that the children began to develop many of the skills necessary for programming and coding as well as computational thinking skills such as collaboration, logical thinking and debugging algorithms. However, they found programming specific algorithms into Bee-Bots complicated and they needed support from adults to direct the robots along routes on simple maps. Overall, the guided play activities could be used in nurseries and preschool establishments to teach early computational thinking skills.

Keywords Programming · Coding · Computational thinking · Preschool · Communication

Introduction

Skills developed through programming and coding are considered to have links with STEM subjects: science, technology, engineering, and particularly with mathematics (Feurzeig et al., 1969; Messer et al., 2018). The UK government's promotion of digital literacy in schools has focused on the education of computational skills and digital creativity to ensure that all children are digitally literate and are able to use computational skills in the workplace (Department for Education [DfE], 2013). One of their initiatives consists of a grant to provide IT training for teachers so that all children have access to a computer and digital technology education. Furthermore, government statutory guidelines for mainstream schools promote the teaching of the basics of programming and coding to young children in Key Stage 1 schools (ages 5–7 years): '(to) *understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions.*' (DfE, 2013).

One of the pioneers of computational science, particularly regarding programming and coding, was Papert who strongly believed that the teaching of computer languages such as Logo had general effects on thinking and learning (Papert, 1988). In support of this premise, researchers report that children benefited from programming as it helps the development of computational skills such as collaboration, logical thinking and the organization and evaluation of ideas (Fessakis et al., 2013; Gomes et al., 2018). However, there is some dispute as to how much cognitive development is attributable to computer programming, and other findings suggest that the children's development is constrained by the level of their abstract thinking (Clements & Gullo, 1984).

Assessing computational thinking skills is considered quite problematic as it is such a wide and newly developing field of research. Some teachers and researchers assess children on the type or level of tasks they achieve. Wing's original concept of computational thinking was procedural and related to the principles involved in giving instructions to a computer or a tangible manipulative, e.g., a robot (Wing,

✉ Valerie Critten
valerie.critten@open.ac.uk

Hannah Hagon
Hannah.Hagon@hotmail.co.uk

David Messer
david.messer@open.ac.uk

[1] Centre for Research in Education and Educational Technology, The Open University, Milton Keynes MK7 6AA, UK

2006). Others regard computational thinking as a cognitive process of formulating problems, evaluating and debugging, and improving a solution. Many of these processes can be thought of as abstract concepts ‘thinking about thinking’ (Allsop, 2019). Thus, tests such as the Coding Development Test for children from 3 to 6 (CODE Test 3–6) (Marinus et al., 2018), and Allsop’s multi-evaluation approach (Allsop, 2019) were developed to assess the cognitive processes of children and their progress in learning coding and computational thinking using digital technology.

The teaching of programming and coding has mainly focused on children above 3 to 4 years of age, for example, Bers et al. (2014). This could be because there is doubt as to whether appropriate activities can be organised for children younger than 3 years who have limited language and other relevant capacities. These limits are also suggested by Piaget’s theory as he supposes children between 2 and 6 years have difficulty mentally manipulating information, using logic, and taking the perspectives of other people or things (Piaget & Inhelder, 1973). Different perspectives also suggest that young children may have limited cognitive capacities, so that they have difficulty memorizing several items of information or processing several items of information simultaneously (Case, 1985). Alternatively, constructionism, as championed by Papert and Harel (1991), suggests that given enough relevant experience and support even young children can accomplish surprisingly difficult tasks. Thus, there are different perspectives and theories about whether or not preschool children are capable of learning about programming and coding.

The limited research findings that we have suggest that 3- to 6-year-olds are able to work with programming software in the form of robots when taught on a one-to-one basis but can find it difficult to understand the principles behind programming and coding unless teaching methods are adapted to their needs (Gomes et al., 2018). Furthermore, the teaching materials need to be suitable for the children’s abilities, as young children usually do not have the reading skills required to access software programs such Scratch or Alice (Gomes et al., 2018). Thus, there is a question whether related research into the successful teaching of coding to children aged 5–7 years, through the medium of simple programming software (the Bee-Bot iPad app), can be extended to younger children (Messer et al., 2018).

Recently, there has been publicity about the establishment of coding clubs to encourage children to develop computational and digital skills. Prior to the Covid19 pandemic, there were nearly 7000 clubs throughout the UK, and these were provided for free through the organization of local libraries and other community venues, with educators and volunteers teaching children to develop their programming skills (Code Club, 2021). The clubs mostly taught children aged 9–13 years to follow projects involving the use of language software such as Scratch and Python and/ or using mini computers such as a Raspberry Pi (Raspberry Pi, 2018). However, some libraries offered one-day workshops for much younger children using hardware such as the Cubetto, which is a small robot that can be easily programmed without the need for the children to know how to read. What is lacking are proposals about the ways to help younger children and the success of such attempts.

Aim of the Study

The first two authors designed a set of activities suitable for coding clubs attended by preschool children. The activities concerned understanding algorithms, making plans, coding, and programming. Our aim was to investigate whether young children, with support, could carry out these activities.

Rationale for the Study

The children, because of their ages, were unable to read or write or use a computer and therefore unable to use software language programs such as Logo or Scratch to develop their programming and coding skills. Consequently, it was decided to use a small table-top robot (see Fig. 3) called a Bee-Bot (TTS Group Ltd, 2021), which can be programmed directly using the seven buttons on its top surface (forward, back, left turn, right turn, pause, clear and go). The teaching of programming had to take account of the need for the children to understand the concepts behind the functioning of the robot such as directional language; landmarks to aid in planning or mapping routes; routes; distances; using simple maps; planning, and recording. However, we also wanted to teach the children computational skills which might involve aspects of communication and collaboration; logical thinking such as planning and sequencing; and the ability to analyse (e.g., correcting errors or debugging algorithms); and we decided to use guided play activities as a medium for teaching these skills.

Programming language and software such as Scratch require children to learn geometric measurements such as distance and angles to make the sprites (cartoon figures) move and change direction on a computer screen. We decided to teach routes around a garden using landmarks to help the children develop concepts of measuring distance and changing direction, and to use the landmarks as a means of planning and recording specific routes. Although the children were familiar with body movements such as in dancing, they had to be made aware of using instructions in a certain order to achieve a desired outcome (sequencing or programming), and we thought of using objects in the children’s own environment (e.g., dolls and toy vehicles). We used simple songs and rhymes to help the children develop symbolic thinking such as directional language (Palmer, 2017) and to help them develop concepts of a specific order of events such as in the words and movements songs like Hokey Cokey (Hokey Pokey). As the children were unable to read or write, photographs, symbols and images were used in activities such as route planning and coding (to provide a sequence of codes that formed algorithms), this allowed pictures to be used as codes instead of a written block code such as seen in Scratch software.

Research Question

Can the experiences and support provided by a coding club help children of different ages and abilities develop computational abilities comprising:

Communication, collaboration and engagement
Routes and landmarks
Programming and coding a table-top robot

Methodology

Participants

Fifteen children between the ages of 2:9 and 4:10 took part in three coding clubs (see Table 1). The Summer Club was mostly children under 34 months, and the two preschool groups were mostly children over 42 months. The children in the Summer Club were recruited through contacts of the first two authors. The children in the pre-school groups were chosen by the preschool manager and were made up of children of mixed gender and ethnicity. Included within the preschool groups were children with special educational needs (SEN) particularly regarding their communication and collaboration abilities.

The findings are presented in relation to the characteristics of the children, those who were less than 34 months (three children from the Summer Club), those with SEN (five children with chronological ages 33–48 months) who were mostly from the pre-school Group 2, and seven children with typical development aged between 42 and 58 months who mostly were from the pre-school Group 1.

Materials

Table 2 displays the equipment that we used for each activity, including distractors, as we wanted the children to choose the correct equipment for each activity from a selection of items. The aim was to encourage the children to envisage the activity beforehand and to plan a procedure.

The Organization of the Coding Clubs

All three coding clubs took place over six sessions held once a week: each session lasting for 45–60 min. The first two authors ran the clubs. The Summer Club took place at the home of one of the authors with the support of some parents or carers, the pre-school clubs took place in a separate classroom to the other children in the setting with the first two authors and one member of staff to support the children.

The timetable of the sessions were as follows (see Table 3) and this was used in all three clubs.

An action research approach was utilised in which:

1. Each session was planned with appropriate activities for the age of the children in order to teach the children concepts and skills leading to the programming of the Bee-Bots.
2. After each session, the children's activities were evaluated by all the adults to establish whether the tasks were too difficult or too easy; and whether the activities were appropriate to fulfil their objectives.
3. Afterwards, plans were made for the next session in which some activities were repeated, and new equipment and activities were introduced to develop the children's knowledge and understanding (see Table 2). This means there were some variations between the three groups, but these were relatively minor.

Each session started with an introduction in which the children, with parents or preschool staff, sat around a table with toys to make up a series of movements (an algorithm) to make their toys move, e.g., forward, back, turn around, clap hands. This was followed by nursery rhymes in which there were directional movements, e.g., Hokey Cokey (Hokey Pokey) or Simon Says.

Table 1 The participants of the three coding clubs

	Summer club	Preschool group 1	Preschool group 2	Total
Number of children	4	5	6	15
Ages in years: months (mean ages of group)	2:9–4:10 (3:04)	3:6–3:10 (3:09)	2:9–4.0 (3:07)	2.9–4:10 (3:07)
Gender	4 f: 0 m	3 f: 2 m	2 f: 4 m	9 f: 6 m
Communication diffs (SEN)	0	1	4	5

Table 2 Materials for the activities

Communication, collaboration and engagement	
Introduction activities	Dolls and soft toys: children brought their own or chose from a selection
Bathe the baby	Small bowl or bath, doll, flannel (face cloth), soap, towel, jug of warm water Distractors: teaspoon, book, saucepan
Dress the baby (see Fig. 1)	Doll, nappy (diaper), vest (undervest), babygro (onesie), socks, bib, hat Distractors: pen, glue
Getting dressed for a party (see Fig. 2)	Picture code grid (6 squares), glue, pictures of clothes: underclothes, nappy (diaper), dress, shoes, socks, trousers (pants), shirt
Landmarks and routes	
Identifying landmarks	Photos of landmarks, corresponding clip-art images of landmarks
Following a route	Large cardboard arrows
Treasure hunt (see Fig. 3)	Maps of the garden, treasure (e.g., gold chocolate coins)
Planning a route	Picture code grid (four squares), clip-art pictures of landmarks, glue
Programming and coding a robot	
Exploring functions of a robot	Bee-Bot tabletop robots
Planning routes (see Figs. 4 and 5)	Bee-Bots, Bee-Bot Farm map, Bee-Bot treasure map, template, green starter shapes, X cardboard shapes
Planning three-move algorithm	Bee-Bots, Farm map
Planning five-move algorithm (see Fig. 6)	Bee-bots, Farm map, Picture code grid (five squares), Pictures of arrows (forward, left and right turns), glue

Table 3 The main aims of the sessions

Session no.	Main aims of sessions
1	To introduce the concept of an algorithm To plan a sequence of events and evaluate the plan (e.g. a recipe; getting dressed for a party; bathing a doll)
2	To identify landmarks on a map To make a map and plan a route using a picture code sequence (Treasure Hunt)
3	To each design a picture code sequence of a route To follow the route around the garden (e.g., Hide the doll)
4	To introduce basic programming using a Bee-Bot To explore the functions of the Bee-Bot and move it around a table top
5	To plan routes for the Bee-Bots on the Bee-Bot maps To program the Bee-Bot and correct any errors (algorithms containing three moves)
6	To plan a route on the farm map, program the Bee-Bot and correct any errors in the algorithms (algorithms containing five moves)

Coding the Support Needed for the Children

Previous research studies have suggested that children needed sufficient adult support to help them with learning digital or computational skills, otherwise they lose interest and give up (Kucirkova & Radesky, 2018). We wanted to ensure that the children were fully engaged with the tasks so a high level of encouragement and scaffolding was provided, for example, if the children had any difficulty in accessing or carrying out an activity, we worked out an amendment that would support the children. The level of adult support needed by the children to engage in the tasks was based on the Coding Development (CODE) Test 3–6 by Marinus et al. (2018) in a study of 3–6-year-olds. The children were individually scored using field notes recorded at the time or by the adults during evaluations after each session. If the children could complete an activity independently, this was given a score of 3; if they needed some reminders (either verbal or non-verbal), a score of 2 was given; or if they needed one-to-one help due to the fact that they were otherwise unable to complete a task, a score of 1 was awarded.

Coding Club Activities and the Findings About the Support Required

Three sets of activities and findings are considered in this section to illustrate and describe computational thinking tasks: (i). communication, collaboration, and engagement, (ii). using landmarks, maps, and picture sequences, (iii). programming robots and coding routes. In each of these sections there is a description of one or more of the major activities and then a commentary and analysis about the achievements of the three groups when engaged in this activity (three children < 34 months, five children with SEN (33–48 months), seven children > 42 months), this keeps relevant information about the activities and behaviours together. Where appropriate, tables provide data about the need for support from adults or the ability to carry out a task unaided.

Communication, Collaboration, and Engagement

We developed activities to promote communication and collaboration between the children in the groups and engagement in the activities. We wanted the children to help each other to think logically and debug any errors (work out what caused a problem and how to solve the problem). One such activity involved 'Bathing the baby', in which the children were shown a range of materials which could be used, plus some pieces of equipment which were not appropriate for the activity (distractors). These consisted of a doll bath, a towel, a flannel (face cloth), a bar of soap, a teaspoon, a book, a baby doll, a jug of water, and a saucepan. The children were first asked to identify the appropriate materials to bathe a baby doll, and the children were asked to point out any errors. The children were able to identify the inappropriate materials, and order the actions needed, e.g., undress the doll, put it in the bath, etc. However, they forgot that they would need water in the bath when they started the activity.

A follow-on activity involved dressing the baby, in which the children sat around a table working out which clothes the baby should wear and the order in which the baby should be dressed. The children all collaborated which clothes were needed and took turns to put an article of clothing on the doll. Figure 1 shows where the children made an error as they put on the bib after the baby's under-vest and before the babygro (onesie) outfit.

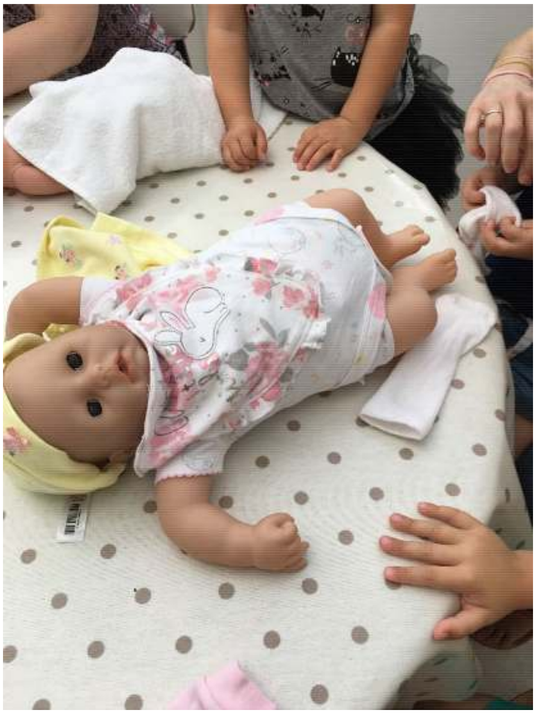

Fig. 1 Dressing the baby: the bib was put on before the babygro (onesie)

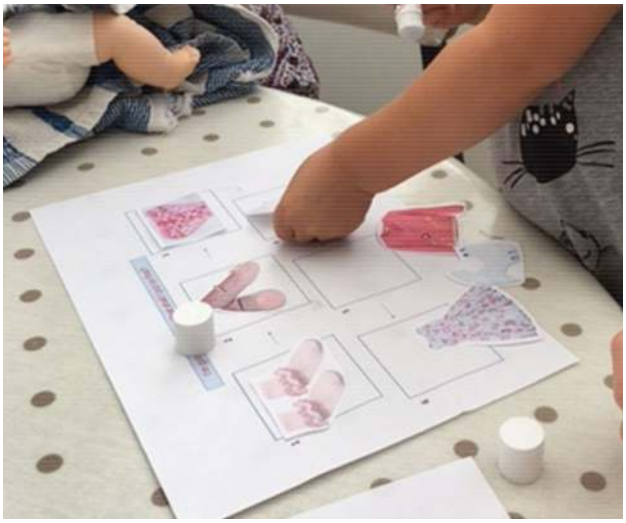

Fig. 2 Dressing for a party: this activity was to enable the children to think logically and to debug errors ordering photos on a Picture Grid

Another example of this was 'Dressing for a party' in which children were given pictures rather than materials. In each group children were given photos of clothes appropriate for a party and asked to situate them on a numbered grid showing the order in which they got dressed. The children chose from a selection of pictures and placed them on the grid. In Fig. 2, the girl in the picture (Sophie) has chosen underwear, a dress, a cardigan, some shoes, and socks. However, we can see that they have been placed in the 'wrong' order, with the socks and shoes being stuck on first. Sophie was asked to read out her numbered picture code: *'What is your algorithm Sophie?'* She read out her code: socks-shoes-pants-vest (which she is sticking down in Fig. 2)-empty box-dress, and the pictures of a nappy and a cardigan were not used. The children were asked: *'Is the* *algorithm correct? Do you put your socks and shoes on first when you get dressed?'* The children were encouraged to call out their answers and work out together the 'correct' order of the algorithm.

We found that all the children had difficulties initially in thinking about the order in which the activities took place as they had concentrated on the items they preferred, e.g., the sparkly shoes in the 'Dressing for a party' activity. However, these activities encouraged the children to talk to the adults and other children about their choices of materials and the order in which the dolls got bathed or dressed. In this way they often corrected their initial decisions which encourages the development of procedural or computational thinking (Manches & Plowman, 2017). In collaboration, the children as a group were encouraged to evaluate errors and correct the sequence of their algorithms (debugging).

We wanted to record the children's levels of communication with each other and the group as a whole as this is considered an important area of computational thinking. The individual categories concerning communication and collaboration were based on those found in the Children Communication Checklist (CCC-2) (Bishop, 2003). We also wanted to record the levels of engagement within the activities as we found that while some of the children were actively engaged throughout the activities we found that other children became distracted, e.g., they left the activity or did not want to take part for short periods throughout the sessions. Sometimes the children just needed a reminder to encourage them to take part, while at other times a child needed 1–1 help to stay on task (see Table 4).

Looking at Table 4, we found that the youngest children in the summer group (< 34 m) were the most spontaneous in their comments and collaborated well in the sessions, only one child needing some prompts from her parent, and this may have been because these children had the least concept of 'making a mistake' (Gomes et al., 2018). The SEN group needed the most help in all of the categories of communication and collaboration apart from non-verbal engagement and, of those, three children (60%) needed 1–1 help throughout to fully communicate and collaborate with the rest of the children and adults. Those three children had the shortest attention spans of all the children and found it difficult to engage with working out sequences and debugging errors, thus their interactions with the other children in their groups were limited and the adults needed to give them extra support so that the rest of the children did not lose interest (Fessaskis et al., 2013). However, two of the children in this group only needed support because of a communication difficulty, e.g., because of EFL (English as a foreign language) and those children were mostly or fully engaged in all the activities. The typical group aged above 42 months were very quiet and did not speak with each other at the start of the sessions but joined in with the Introduction sessions and worked well together as a group. One boy needed to have a verbal cue to join in at times. Otherwise, this group needed the least input from adults and were fully engaged in all the activities.

Observations and field notes suggest that even the youngest children had the communication and collaboration abilities to engage in computational activities. Those with the lowest language abilities, the youngest children and two of the children in the SEN group were able to fully collaborate and engage with the other children and adults. However, three of the children in the SEN group, who were the most distracted by their environment and went off task, often had difficulties in engaging with the others.

Landmarks, Routes and Maps

We asked the children to pretend to be robots acting as their own manipulatives (Sullivan & Heffernan, 2016) and introduced directions such as forward, turn left, turn

Table 4 Percentage of children in the three groups who were able to communicate and collaborate within the groups: only with 1–1 help from an adult (1–1), only with a reminder, e.g. a verbal or non-verbal cue (Remd), or independently (Indep)

	Groups (numbers of children in brackets)								
	< 34 m (3)			SEN (5)			> 42 m (7)**		
	1–1	Remd (%)	Indep (%)	1–1 (%)	Remd (%)	Indep (%)	1–1	Remd (%)	Indep (%)
Communication									
Non-verbal engagement			100			100			100
Verbal engagement			100	20		80		30	75
Collaboration within the group									
Answering direct questions		33	66		60	40		15	90
Suggestions/comments to an adult			100	20	60	20		15	90
Suggestions/comments to the group		33	66	20	40	40		15	90
Total engagement throughout the activities		100		60	20	20			100

**Percentages rounded up for clarity

right. However, we found the younger children were unable to understand left and right moves. So we just used the term 'turn' and gestured with our hands or arms which way to turn such as used in the study by Palmer (2017).

The map-related activities were designed to help the children understand the way to follow routes and use simple maps. The activities mostly took place out in the gardens so that the activities could incorporate landmarks, such as archways, gates, and stepping stones; and routes between them such as paths. The children in each group were made familiar with the landmarks in their group's garden, and the children were asked to find routes between the landmarks, e.g., '*Start at the gate, how do you get to the apple tree?*'.

Once the children became knowledgeable about the landmarks and the gardens, they were given large cardboard arrows to mark a route so that others could follow it (see Table 5, line 1). Most of the youngest and the SEN groups needed reminders or 1–1 help to find an arrow or follow the correct arrow. However, most of the older group could carry out the task independently. This suggests that the children could understand the use of the arrows as a code, but sometimes the younger ones may have had difficulty in remembering the significance of the arrows. The route-marking using arrows were successful in encouraging the children to work together and helped with problem-solving (Allsop, 2019).

Next, the children were individually given simple maps of the gardens with the landmarks and routes marked with arrows (see Fig. 3). They were asked to finger-walk, pretending their fingers were legs, the routes between two or three landmarks. To help the children the start point was marked with a green rectangle ('for go') and the end of the route was marked with an X ('X marks the spot'). All of the children were able to find and relate to the landmarks on the maps and were able to finger-walk the routes, but some needed help with following the arrows in the correct direction (see Table 5, line 2).

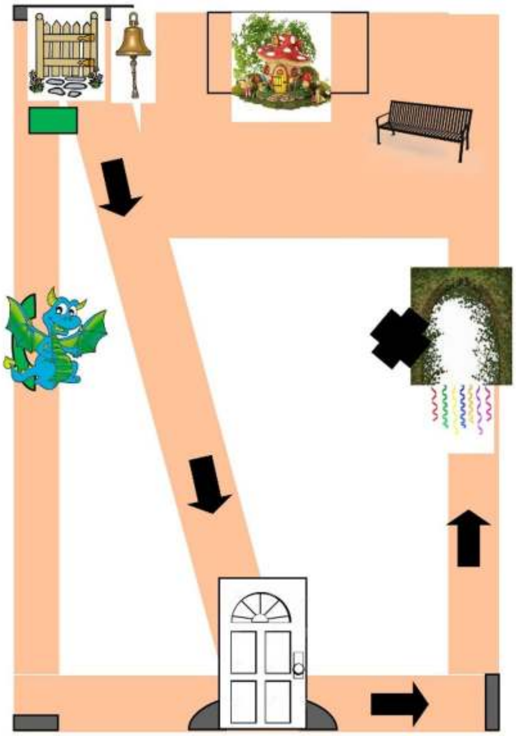

Fig. 3 Treasure hunt map of the preschool garden. The paths are marked in pink and the arrows indicate the route to be followed between the landmarks. The starting point is the bell-gate and the treasure is hidden in the ribbon tunnel

The children also were given the same finger walking activity in relation to a treasure map. The route to the treasure was marked out on the preschool garden map as shown in Fig. 3; the starting point was marked with a green rectangle, and the treasure (gold chocolate coins) was hidden at the position marked with an X on the map. All the children were asked to finger-walk the routes to see if they were able

Table 5 Percentage of children in the three groups who were able to carry out an activity, related to maps and picture sequences only with 1–1 help (1–1), only with a reminder (Remd), or independently (Indep)

	Groups of children (numbers of children in brackets)								
	< 34 m (3)			SEN (5)			> 42 m (7)**		
	1–1 (%)	Remd (%)	Indep (%)	1–1 (%)	Remd (%)	Indep (%)	1–1 (%)	Remd (%)	Indep (%)
Following marked route in the garden (arrows)	66	33	0	40	40	20	15	15	75
Following marked route on map (finger walking)	66	33	0	80	20	0	30	60	15
Making and explaining picture code from the map	66	33	0	60	40	0	15	30	60
Following picture code in the garden	100	0	0	80	20	0	30	60	15

**Percentages rounded up for clarity

to find the treasure on the map and all the children were able to follow the routes with a little help (see Table 5, row 2).

Across both of these finger-walking activities, most of the youngest group and the children with SEN needed 1–1 help (see Table 5, row 2). Most of the oldest group could carry out the tasks with a simple reminder, and one child was able to carry out the activities independently. This indicates that the tasks were difficult for children to accomplish alone, and common problems that were observed were needing help to signal the beginning and end to the route, or forgetting the end point when they were in the middle of a route. It may be that with more practice the children could have become fully independent, or it might be that the children had to remember too many pieces of information on each occasion (Clements & Gullo, 1984).

Next, we wanted to see if the children could plan a route and then follow it in the garden. The children were asked to plan a route to a hiding place for some treasure. We gave the children the treasure map they had previously used with pictures of the landmarks on it. We also gave them cut out pictures of the landmarks that could be placed on a grid of four squares in a horizontal line; together these could be made into a picture code of the route (see Table 5, row 3). Most of the younger children and SEN group needed 1–1 help with this activity, although a sizeable proportion in these groups were able to carry out the activities with simple reminders. In contrast, most of the children in the oldest group were able to carry out the activity independently. This suggests that all the children had a reasonable idea about using a picture code to describe their plan of a route.

The children were then asked to follow the picture sequence code in the garden, and they found this more difficult. Most of the children in the youngest and SEN groups needed 1–1 help and most of the oldest children needed prompts from the adults (see Table 5, row 4). This suggests that the children found it much easier to make a plan, rather than to follow it once they were out in the garden.

To summarise, Table 5 shows that almost all of the youngest group and the SEN group needed 1–1 help or reminders for all of these route and map activities (only one child from the SEN group was able to carry out an activity independently). This means that the children in these two groups needed support to be able to follow a route marked by arrows in their garden, by finger walking a route on a map, following a picture code sequence in their garden and using picture codes to describe a route. They appeared to have some understanding of these activities, but usually needed help to carry out the activity. They gained a concept of the garden from a 2D representation on paper but found it difficult to transpose this to a 3D representation in the garden (Meilinger et al., 2013) and from an aerial perspective to a walking perspective.

In contrast, most of the oldest typical group were able to work independently on activities where they followed a route marked with arrows in the garden or created a picture code sequence from a map. However, these children like the others, found it difficult, without some support, to finger walk a marked route on a map or follow a picture code in their garden.

Although using a picture code took away the contextual aspect of using the map to record routes (Buchner and Jansen Osmann, 2008), it meant that the children learned to code routes in a basic but similar way to that used by programming software such as Scratch where the codes are procedural. Despite it seeming to be one of the easier tasks, following the picture code of a route out in the garden was the most difficult for all the children and only one child in the typical group was able to follow the sequence of the code completely independently.

Programming and Coding a Robot

We wanted to apply the new-found experiences to making codes to make a robot follow a route. We introduced the children to robots (Bee-Bots) to see if they were able to program an algorithm for the Bee-Bots to follow. Once they had explored the features of the Bee-Bots (by pressing buttons on the Bee-Bots—up, down, right, and left arrows; 'go' and 'clear'), the children were encouraged to program them with instructions of their choice. We then asked them to program the Bee-Bots to send them to each other across a table so that they could gauge distance, and then program them to move around an object so that they could practice distance and turns.

We also introduced the children to two Bee-Bot maps that were supplied with the robots: a treasure map and a farm map each of which was marked with squares (one click of the forward arrow moves the Bee-Bot forward into the next square). Initially we gave the children toy vehicles and asked them to find routes on the maps, and to verbalize the routes they used as an algorithm, e.g., two squares forward, turn right, two squares forward. Once the children were able to program the Bee-Bots with basic algorithms and had identified the landmarks on the maps, we asked them to describe a route on the maps for the Bee-Bots, e.g., on the treasure map, start at the pirate ship and describe how to get to the beach on the island. In other words, the children had to identify the instructions needed for a route and then program the algorithm into the Bee-Bot (e.g., forward 1 click¸ right click, forward one click; see Fig. 4).

All of the children initially found it difficult to complete an algorithm containing three movements, e.g., forward, turn right, forward. As a result, all groups were given an activity involving a simple route. We made a template to help them understand planning a route (see Fig. 5) which consisted

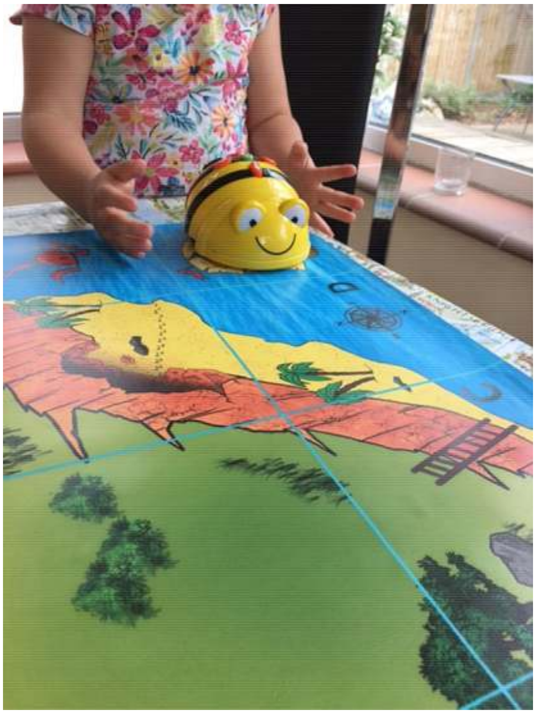

Fig. 4 Programming an algorithm into the Bee-Bot to make a route from the pirate ship to the beach on the island

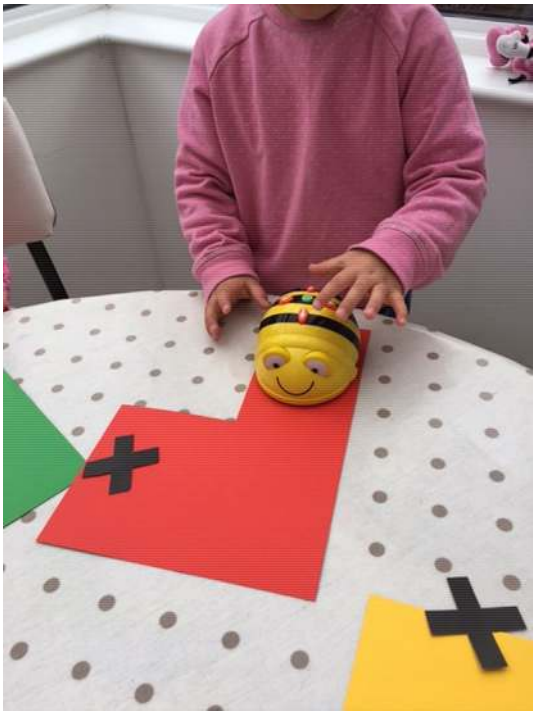

Fig. 5 A cardboard L-shaped template was made so that the children could practice making an algorithm involving three movements, e.g., forward, turn, forward

of an L-shaped piece of card, and used a green rectangle to mark the start, and an X to mark the finish. This is a similar idea to that of Palmer's study (2017) in which the participants were asked to draw arrows of a route on a small portion of a map.

We found that the template helped the children to understand how to carry out the individual moves required. However, all of the youngest group and most of the SEN group needed either 1–1 help or reminders in identifying the correct algorithm of this route (Table 6 line 1) or programming a Bee-Bot with these moves (Table 6, line 2). In contrast, most of the oldest group could carry out these activities independently and this suggests that they had acquired some of the basic elements of programming and coding.

Because so many of the children found it difficult to work out and enter the directions on the Bee-Bots, we asked them to create a picture code of a sequence of five moves. For example, the children were asked "Where will you start your Bee-Bot? Where do you think it will finish?" and then they were asked to give the instructions needed for the Bee-Bot. As with the previous picture codes, the children placed pictures of the instructions in a sequence on a grid with five spaces (see Fig. 6). The pictures had forward arrows and turns to code their choice of a route on the Bee-Bot farm map. The longer sequence of five moves appeared to create difficulties for the youngest and SEN groups as nearly all these children needed 1–1 help with identifying an algorithm (see Table 6, row 3) and programming a Bee-Bot (Table 6, row 4). Palmer (2017) found that many of the participants in her study were able to decompose and produce subsets of instructions when working out routes, while the children in this study attempted to create their algorithms in one attempt. The level of support needed might have been necessary due to the amount of information on the map, the additional task of programming the Bee-Bot was too difficult, or they were unable to process several pieces of information simultaneously (Clements & Gullo, 1984), although two of the oldest children were able to carry out these activities independently.

The children mostly stuck the arrows on the grid in a random manner without thinking about the order of the arrows or where the Bee-Bots would start or finish, however, two of the children from the typical older group worked out a route on the farm map and ordered the arrows to complete a specific route and it was observed that these children decomposed the route into sections in the same way as those in Palmer's study (2017). Those two children also programmed the Bee-Bot correctly so that the Bee-Bot followed the correct route see Table 6, row 4). The other children all needed 1–1 support to correctly program the Bee-Bots.

All of the children in this study attended the same preschool although, at the time of the study, the Summer Club children were in different classes. Table 7 shows the

Table 6 Percentage of children in the three groups who were able to carry out an activity with the Bee-Bots, only with 1–1 help, only with a reminder, or independently

	Groups (number of children in brackets)								
	< 34 m (3)			SEN (5)			> 42 m (7) **		
	1–1 (%)	Rem (%)	Ind (%)	1–1 (%)	Rem (%)	Ind (%)	1–1 (%)	Rem (%)	Ind (%)
Identifying the correct algorithm (three moves)	0	100	0	40	40	20	15	30	60
Programming a Bee-Bot (three moves)	66	33	0	60	20	20	15	30	60
Identifying the correct algorithm on a grid (five moves)	100	0	0	80	0	20	45	30	30
Programming a Bee-Bot from a grid (five moves)	100	0	0	100	0	0	70	0	30

** Percentages rounded up for clarity

breakdown of the levels of independence shown by all the children, suggesting that they needed less support for the initial sessions in which the activities focused on dolls and toys, or the activities based in the gardens. However, in the sessions involving the Bee-Bots the children needed much more support, in particular the activities involving creating the five-movement algorithms and programming the Bee-Bots.

The number of children who were classified as being at the different levels of support was evaluated with a one-tailed Binomial test. This showed that there were significantly more children who were classified as independent than as needing help or a reminder for verbal ($p < 0.001$) and for non-verbal communication ($p < 0.018$). For all the collaborative activities there were significantly more children classified as independent and needing a reminder than as needing one-to-one help for all these activities ($p < 0.001$). In addition, there were significantly more children classified as independent or needing a reminder than needing one-to-one help for creating a correct 3-move algorithm with the beebots ($p < 0.018$). These analyses provide evidence that by the end of the sessions most of the children were able to engage in these activities independently or with a reminder, however, it needs to be acknowledged that this does not establish that the coding sessions were the cause of this achievement, as no baseline assessments were obtained.

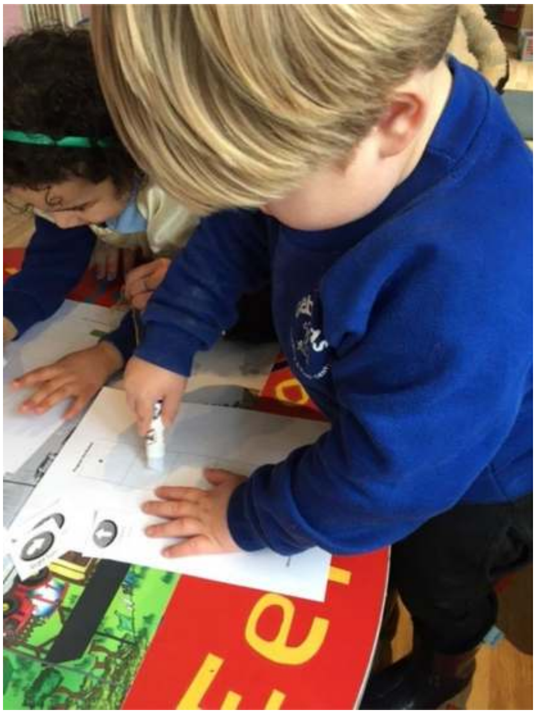

Fig. 6 The children stick forward and turn moves onto a five-square grid to create an algorithm

Discussion

This discussion will focus on whether the guided play activities enabled the children to develop computational thinking concepts leading to programming and coding abilities.

No articles appear to have been published regarding programming and coding with children under 36 months, as researchers may consider them too immature to develop computational skills (Clements & Gullo, 1984). However, we found that using toys and equipment familiar to the children in a guided play approach, rather than using computers, was successful in promoting aspects of computational thinking such as communication and collaboration (Bers et al., 2014). All the children were happily engaged in the introduction to the sessions, even those with communication difficulties. This may have been due to the choice of materials and activities, as well as to the organization of small groups of children around a table interacting with each other while being guided by the adults involved (Dockrell et al., 2015). We found that an informal and friendly approach worked well to promote group cohesion and encouraged the children to work together when planning activities such as choosing equipment for bathing a doll or debugging a picture code when there was an error. (Fessakis et al., 2013; Gomes et al., 2018).

Table 7 Overall participation in the activities (N = 15)

	1–1	Reminder	Independent
Communication			
Non-verbal communication	0	0	15
Verbal communication	1	2	12
Collaboration			
Answers direct questions	0	5	10
Suggestions/comments to adult	1	4	10
Suggestions/comments to group	1	4	10
Total engagement in the activities	3	4	8
Landmarks and routes			
Follow route marked with arrows	5	4	6
Finger-walk a route on a map	8	6	1
Making/explaining picture code of map	5	6	4
Follow picture code in garden	9	5	1
Beebots			
Creating correct 3-move algorithm	3	7	5
Correctly program 3-move algorithm into Beebot	6	4	5
Create correct 5-move algorithm	10	2	3
Correctly program 5-move algorithm into Beebot	13	0	2

Many of the children needed some support in the mapping tasks, although they were able to complete most of the initial tasks independently or with a reminder such as identifying landmarks, creating routes, and following routes marked with arrows. The children were able to identify landmarks and routes on the maps during table-top activities indicating that they were gaining configurational knowledge of the relationship between landmarks and routes (Farran et al., 2012). However, they needed assistance and prompting to follow the routes once they were out in the garden, and this was probably due to the difference in perspective and the need to mentally manipulate information from a 2D image to a 3D environment (Piaget & Inhelder, 1973). We also found that the children were able to make picture codes of the routes and were able to explain them but were unable to follow them independently when they were out in the garden. This may have been because they were excited and distracted in the garden: going off task for a short time; turning their picture codes upside down; or not reading them from left to right. Providing symbols to display the start and finish of routes on the maps and on the picture codes helped the children, especially the older ones, to work more independently.

The aim of this study was to teach the children, via play activities, to learn how to program (plan algorithms) and code (using pictures or symbols) so that ultimately, they could control Bee-Bots. All the children were able to learn and use the basic functions of the Bee-Bots so that they could make them move across a surface. However, there was a difference between the random routes programmed by all the children when they were exploring the functions, to being able to program specific routes on the Bee-Bot farm map. Similar orientation problems using ladybugs on routes were found by Fessaskis et al. (2013) in their study of 5–6-year-old children. Researchers such as Palmer (2017) encouraged their participants to draw routes on paper maps blown up from the Farm map. We used templates to help the children learn how to program basic three-move algorithms involving a turn (see Fig. 5). Only three of the children in our research group were able to program the three-move algorithm independently. The remaining twelve required prompting or 1–1 support. This need for support was probably because of the difficulty posed by multiple frames of reference when following a route on a map (Meilinger et al., 2013) and similar to the difficulties the children had in using maps once they were out in the garden e.g., do you turn left or right on the route if the map is upside down to your viewpoint?

Assessing the children throughout the study using observations by the researchers and the parents or preschool staff, as well as field notes taken at the time, helped us to assess the children's strengths and to adapt the activities accordingly. There were no assessment scales available for such young children, so we modified aspects of those used by Marinus et al., (2018) and Allsop (2019), but focused on the support necessary for the children to complete the activities. This is quite important as generally very young children need a high degree of support when using digital technology and Coding Clubs and preschool establishments may not be able to offer either the equipment or the level of staffing that might be required

(Manches & Plowman, 2017). Thus, although the youngest children and those in the SEN group needed more support in some of the activities, they succeeded in many of the activities particularly those involving toys and equipment which can be found in any preschool environment. Some of the activities such as those in the introduction sessions can be carried out with all the children in a class, while others might be suitable for small groups such as bathing or dressing a doll within a nursery class.

Implications and Limitations

Many preschool teachers and support staff are unaware of or are untrained in adapting pedagogical approaches that can support computational thinking (Manches & Plowman, 2017). Furthermore, government guidelines for preschool digital technology are very vague or non-existent and it would be helpful for staff if more specific guidelines, such as those used in Swedish preschools, are promoted (Otterborn et al., 2020). The findings from this study suggests play-based, pedagogic practices can be used with children as young as 2 years to learn many of the basic concepts involved in computational thinking skills. It may be the case that because the younger children had not developed 1–1 correspondence, they found it difficult to use the map grids to create algorithms to program the Bee-Bots. This suggests that Bee-Bots are utilised by teachers for older children rather than those aged below 4 years.

There is a lack of studies involving children as young as 2-year-olds and further research is needed as to whether such young children benefit from learning computational thinking skills. Our findings suggest that the children were happy to engage in the guided play activities even though some of them were challenging for them to master without support. Papert (1988) calls this '*hard fun*' when the children are not discouraged because they are enjoying the activities.

Conclusion

This study was designed to investigate whether coding clubs for preschool children could provide, through guided play, a programme of activities that would teach the children basic computational thinking abilities which would lead to the development of programming and coding skills. We found that many of the children were able to gain skills such as communication and collaboration, planning, logical thinking, and problem solving through the play activities, but needed support when programming routes for the Bee-Bots. Ultimately, the children were able to learn some of the key computational thinking concepts around logic, sequencing, algorithms and debugging but to varying degrees and with varying degrees of adult support. With further lessons in computational thinking concepts in small groups, it is likely that the children would be able to complete some of the tasks independently.

Declaration

Ethical Approval Ethical approval was obtained from the Open University Ethics Committee. Written consent was given by the parents and by the preschool, and verbal consent was given by the children.

References

Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International Journal of Child-Computer Interaction*, *19*, 30–55. https://doi.org/10.1016/j.ijcci.2018.10.004

Bers, M., Flannery, L., Kazakoff, E., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education, 72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Bishop, D. (2003). *The children's communication checklist*. Harcourt Assessment.

Buchner, A., & Jansen-Osmann, P. (2008). Is route learning more than serial learning? *Spatial Cognition and Computation, 8*(4), 289–305. https://doi.org/10.1080/13875860802047201

Case, R. (1985). *Intellectual Development from Birth to Adulthood*. Academic Press.

Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology, 76*(6), 1051. https://doi.org/10.1037/0022-0663.76.6.1051

Code Club (2021). https://www.codeclub.org.uk/

Department for Education. (2013). *National curriculum in England: computing programmes of study*. https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study

Dockrell, J. E., Bakopoulou, I., Law, J., Spencer, S., & Lindsay, G. (2015). Capturing communication supporting classrooms: The development of a tool and feasibility study. *Child Language Teaching and Therapy, 31*(3), 271–286. https://doi.org/10.1177/0265659015572165

Farran, E. K., Courbois, Y., Van Herwegen, J., & Blades, M. (2012). How useful are landmarks when learning a route in a virtual environment? Evidence from typical development and Williams syndrome. *Journal of Experimental Child Psychology, 111*(4), 571–586. https://doi.org/10.1016/j.jecp.2011.10.009

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers and Education, 63*, 87–97. https://doi.org/10.1016/j.compedu.2012.11.016

Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. (1969). Programming languages as a conceptual framework for teaching mathematics. Final report of the first fifteen month of the LOGO Project. *Mass Inst Technol Interface, 4*(2), 13–17.

Gomes, T. C. S., Falcão, T. P., & Tedesco, P. C. D. A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. *International Journal of Child-Computer Interaction, 16*, 77–84. https://doi.org/10.1016/j.ijcci.2017.12.005

Kucirkova, N., & Radesky, J. (2018). Digital media and young children's learning. In K. Sheehy & A. Holliman (Eds.), *Education and new technologies: Perils and promises for learners* (1st ed., pp. 15–32). Routledge.

Manches, A., & Plowman, L. (2017). Computing education in children's early years: A call for debate. *British Journal of Educational Technology, 48*(1), 191–201. https://doi.org/10.1111/bjet.12355

Marinus, E., Powell, Z., Thornton, R., McArthur, G., & Crain, S. (2018, August). Unravelling the cognition of coding in 3-to-6-year olds: The development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (pp. 133–141).

Meilinger, T., Frankenstein, J., & Bülthoff, H. H. (2013). Learning to navigate: Experience versus maps. *Cognition, 129*(1), 24–30. https://doi.org/10.1016/j.cognition.2013.05.013

Messer, D., Thomas, L., Holliman, A., & Kucirkova, N. (2018). Evaluating the effectiveness of an educational programming intervention on children's mathematics skills, spatial awareness and working memory. *Education and Information Technologies*. https://doi.org/10.1007/s10639-018-9747-x

Otterborn, A., Schonborn, K., & Hulten, M. (2020). Investigating preschool educators' implementation of computer programming in their teaching practice. *Early Childhood Education Journal, 48*, 253–262. https://doi.org/10.1007/s10643-019-00976-y

Palmér, H. (2017). Programming in preschool-with a focus on learning mathematics. *International Research in Early Childhood Education, 8*(1), 75–87.

Papert, S. (1988). Does easy do it? Children, games, and learning. *Game Developer Magazine, 5*(6), 88–89.

Papert, S. & Harel, I. (1991). *Situating constructionism*. http://www.papert.org/articles/SituatingConstructionism.html

Piaget, J., & Inhelder, B. (1973). *Memory and intelligence*. Routledge and Kegan Paul.

Raspberry Pi. (2018). Raspberry Pi Foundation. https://www.raspberrypi.org/

Sullivan, F. R., & Heffernan, J. (2016). Robotic construction kits as computational manipulatives for learning in the STEM disciplines. *Journal of Research on Technology in Education, 48*(2), 105–128. https://doi.org/10.1080/15391523.2016.1146563

TTS Group (2021) TTS. https://www.tts-group.co.uk/primary/computing/bee-bot-blue-bot-pro-bot-ino-bot/

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.