Primary Developments

Final report on the *Coding in Primary Schools Initiative*

NCCA
An Chomhairle Náisiúnta Curaclaim agus Measúnachta
National Council for Curriculum and Assessment

June 2019

Table of Contents

Chapter 1: Introduction

In July 2016, the NCCA was asked by the then Minister for Education and Skills, Mr Richard Bruton, TD., to consider approaches to integrating coding and computational thinking into the primary curriculum. In responding to the request, Council highlighted the importance of working with schools to identify current practice in relation to coding and computational thinking, and to explore with schools how, where and to what extent the two could be integrated into a redeveloped primary curriculum. From this, the *Coding in Primary Schools Initiative* was set out in the NCCA Plans of Work for 2017, 2018 and 2019 and in the DES Action Plans for Education for those same years.

Preparatory work for the initiative included several components:

- an initial desktop audit of twenty-two jurisdictions with a focus on identifying the presence of coding in the curriculum[1].
- a follow-on in-depth curriculum investigation which explored how six international jurisdictions integrated coding and computational thinking into their curricula[2].
- a review of literature on computational thinking and its importance for children's learning as well as its place in a primary curriculum. This was done to clarify questions regarding computational thinking and its use as a foundation for teaching the fundamentals of coding in a primary school setting[3].

Through this research, the NCCA identified several common approaches used internationally to integrate coding into a primary school curriculum. These included locating coding within a broader curriculum area such as computing or computer science; applying some of the

[1] NCCA, 2016. *Desktop audit of coding in the primary curriculum of 22 jurisdictions.* See https://www.ncca.ie/en/resources/primary-coding_desktop-audit-of-coding-in-the-primary-curriculum-of-22-jurisdictions to read full report.
[2] NCCA, 2017. *Investigation of curriculum policy on coding in six jurisdictions*. See https://www.ncca.ie/media/3545/primary-coding_investigation-of-curriculum-policy-on-coding-in-six-jurisdictions.pdf to read full report.
[3] Milwood *et al*, 2018. *Review of Literature on Computational Thinking.* See https://www.ncca.ie/media/3557/primary-coding_review-of-literature-on-computational-thinking.pdf to read full report.

fundamental underpinnings of coding within subjects such as mathematics, science and the arts; and developing it across the curriculum as an overarching competence.

This report provides a brief overview of the research carried out to inform the developmental work on coding and computational thinking, and then describes, in greater detail, the developmental work conducted with schools. Finally, it provides a summary of findings and recommendations, both short and long-term, regarding the integration of coding, computational thinking and technology, more broadly, into the primary curriculum.

International curriculum investigations

In July 2016, the NCCA initiated a desktop audit of curriculum policy related to coding in 22 jurisdictions. From here on, this desktop study is referred to as the NCCA Curriculum Audit.

A broad range of jurisdictions were included in the audit—neighbouring UK systems due to their obvious cultural similarities, Scandinavian and other European jurisdictions which appear to be leading developments in this area, the United States owing to their strong emphasis on research-driven approaches, and finally Pacific and Asian countries which have emerged as leading trends in technology and ICT within curriculum policy.

Key findings from the curriculum audit

- Computer science and coding are increasingly integrated within curriculum policy at primary education level in most jurisdictions.
- In 2014, the UK[4] was the first country to introduce coding as a mandatory component of the (computing) curriculum at primary level. Since then, Slovakia, Poland, Finland and France have also introduced coding as a mandatory component in the primary curriculum.

[4] Computing is mandatory in state-maintained schools but not in academies, free schools and independent schools, although many will teach it.

- The location of coding within the curriculum varies across jurisdictions. In Finland and France, coding is taught in maths and as a cross-curricular activity. In England, Slovakia and Poland, coding is part of a broader computer science subject[5].

- In the countries investigated, creating with technology, understanding technology, and using technology are all named very differently, but are all linked when children learn how to code.

- In some jurisdictions, e.g., England and Finland, coding is introduced to children from an early age (5+ years). Other jurisdictions, e.g., France and Spain (Navarra) do not introduce coding until later in the primary years (10+ years).

Towards the end of 2017, the NCCA selected six jurisdictions for a more in-depth examination—England, Finland, New Zealand, Scotland, Northern Ireland, and the USA. In the case of the sixth jurisdiction, the USA, as there was no mandatory state-wide coding curriculum, it was necessary to look at recommendations made by the Computer Science Teachers Association (CSTA) and how they were being used and implemented in Washington State's primary curriculum.

The purpose of the curriculum investigation was to inform the NCCA's development of a set of support materials for schools to work with as they piloted different approaches to integrating coding into the Irish primary school curriculum. In this report the in-depth study of six jurisdictions is referred to as the NCCA Curriculum Investigation.

Key findings from the curriculum investigation

- There is commonality in terms of what is taught at each age level and across the six primary curricula.

- Computational thinking, common coding standards, and strands or themes can be clearly identified and are being implemented from the child's first year in primary school in all six selected countries.

[5] Computing (England), Infomatics (Poland & Slovakia)

- The countries that have computing or technology identified as a separate subject in the curriculum also recommend that coding (or computational skills) needs to be integrated throughout several other primary curriculum subjects.
- Continuing professional development for teachers is a priority in all six countries and is ongoing. Because the world of technology is constantly changing the necessity for ongoing support for teachers in this area is widely recognised.
- As with the findings in the initial desktop audit, creating with technology, understanding technology, and using technology were highlighted as key digital competencies—although named differently in each jurisdiction—and all were linked when children learn how to code.

Review of literature on computational thinking

Recent research has shown that computational thinking lays some of the foundations for coding through its emphasis on problem-solving and creative, flexible thinking skills. It has also been proposed that computational thinking will better prepare every child for living in an increasingly digitalised world, and that computational thinkers will be superior problem-solvers in all fields (Denning, 2017).

To understand fully the essence of computational thinking, its importance for children's learning and its place in a redeveloped Irish primary school curriculum, the NCCA commissioned a review of literature in this area. The aim of the review was to define what computation thinking is in relation to a primary school curriculum and to clarify questions regarding its use as a foundation for teaching the fundamentals of coding in a primary school setting.

Key findings from the review

- Computational thinking is the right focus in primary education and can and should be supported and developed through activities in every subject.

- Playful and meaningful approaches should be used to maintain interest and zest in pupils when teaching coding and computational thinking skills.
- Professional development approaches must be creative and collaborative, to enable teachers to develop their personal competence and to understand the related pedagogical and content knowledge.

Learning outcomes

To help schools in Phase 2 to investigate a physical computing and play-based pedagogical approach to coding and computational thinking a set of learning outcomes was developed, which could be used from junior infants to sixth class. For the initial development of the learning outcomes it was essential to reflect on the findings from the international NCCA Curriculum Investigation and the work with schools participating in Phase 1.

Development process

The first draft of the learning outcomes was compiled from the outcomes identified through the NCCA curriculum investigation which showed a high degree of similarity across the jurisdictions. Schools from Phase 1 were then asked to analyse and rank the draft learning outcomes in order importance in the context of their current teaching of coding and computational thinking.

The learning outcomes worked on by the teachers showed progression from junior infants to sixth class, reflecting and supporting their classroom activities and use of physical or tangible computing devices. It was also necessary, for continuity and in light of developments at both Junior and Senior Cycle, to look closely at the potential progression of the outcomes from the initiative to link to the intended learning outcomes of the Junior Cycle short course on coding and the newly-introduced Computer Science Leaving Certificate subject.

Finally, the learning outcomes for use in Phase 2 were then reviewed and further developed in cooperation with the NCCA staff working on the Computer Science Leaving Certificate specification.

Overview of the school-based initiative

Aims and rationale

The school-based initiative itself involved two phases. The broad aim of the work with schools selected for Phase 1 was to document the current practice of teachers and schools in relation to coding, and to share their stories, examples of classroom activities and whole school practices as a support to other schools interested in this area.

These classroom stories would help to answer questions such as:

- what types of coding experiences are teachers providing?
- where in the primary curriculum is this work happening?
- which classes are involved?
- why did the teacher/school start work on coding? What are the benefits and challenges?

The aim of Phase 2 was to work with schools which had little or no experience of coding and computational thinking in their classrooms and to capture the experience of the teachers' implementation of the area in their classrooms. It also provided an opportunity for principals, teachers, parents and children to express their views on its implementation and inform developments in this area.

Phase 2 was designed to address the following questions:

- to what extent, for what purpose, and where could coding and computational thinking be integrated in a redeveloped primary curriculum?
- what are teachers', parents' and children's experiences and perspectives on coding/computational thinking?
- what types of resources and continued professional development would teachers and schools require?

Phase 1

Phase 1 of the initiative began in September 2017 with an open call for schools to participate. The schools in this phase were early adopters or had prior knowledge of how to teach coding and computational thinking in an Irish primary school setting.

A total of 47 primary schools applied to participate and, from these, 15 schools were selected. The schools in Phase 1 of the intitiative represented both a geographical and contextual spread of school type including: urban DEIS, rural DEIS, scoil sa Ghaeltacht, Gaelscoil, school with special classes, small rural, and large urban. It was also important to include teachers working across the eight classes from junior infants to sixth class as well as teachers working in multi-grade classrooms and in SEN settings.

Informed by the current experiences of schools in Phase 1 and by international practice, the NCCA developed broad guidelines and a set of physical computing and play-based pedagogical approaches to support the teaching of coding and computational thinking. The findings from Phase 1 highlighted the importance of using:

- constructivist methodology
- project-based learning
- links to the current curriculum.

The draft learning outcomes were developed in the light of classroom experiences of teachers in Phase 1 of the initiative and in consideration of the findings from the NCCA Curriculum Investigation of coding curricula in six jurisdictions.

Phase 2

The second phase of the initiative took place between May 2018 and February 2019. The aim of Phase 2 was to collaborate closely with teachers who were not already familiar with or had taught coding and/or computational thinking in their classrooms before.

This time, 153 primary schools applied to participate and, from these, 25 schools were selected. These schools, as in Phase 1 of the intitiative, represented both a geographical and contextual spread of school type, including: urban DEIS, rural DEIS, scoil sa Ghaeltacht, Gaelscoil, school with special classes, small rural, and large urban schools. As in Phase 1, teachers working across the eight classes from junior infants to sixth class as well as teachers working in multi-grade classrooms and in SEN settings were included.

Phase 2 of the school-based initiative aimed to capture the experience of teachers *beginning* to work with coding and computational thinking in their classrooms. Informed by the outcomes of Phase 1, Phase 2 aimed to understand the potential benefits of teaching coding and computational thinking and physical computing (tangible coding) through a playful and project-based pedagogy. Phase 2 also provided an opportunity for principals, teachers, parents and children to express their views and inform developments in this area.

Teachers used a variety of resources and recommendations drawn from the findings of Phase 1, the NCCA Curriculum Investigation and the review of literature on computational thinking. Further, they were provided with support guidelines which drew on examples of good practice and which offered suggestions for how teachers and schools might provide children with experiences in coding and computational thinking in the context of the current primary curriculum. As they got involved in hands-on, project-based approaches to teaching coding and computational thinking in the classroom the Phase 2 schools had the opportunity during face-to-face workshops to work with and learn from teachers in the other participating schools.

In addition, the schools were offered on-going professional development and support, in part through collaboration with the PDST Technology in Education team. The CPD included an online course, *An Introduction to Scratch Programming,* which several schools completed and an adaptation of the PDST summer course on Computational Thinking, completed by all schools at the inaugural face-to-face meeting in June 2018.

An online community of practice using the Microsoft Teams environment was also established. The Teams online collaboration space was intended for the schools in Phase 1 to share resources and collaborate with teachers selected for Phase 2.

Table 1 provides an overview of the timeline of supports and face-to-face meetings offered to the schools participating in Phase 2.

Table 1: Face-to-face workshops

Action	Description	Date
2-day face-to-face workshop	▪ Introduction to Phase 2 initiative ▪ Computational Thinking course based on PDST summer course specification	June 2018
1-day face-to-face workshop	▪ Introduction to teaching resources and Teams facility from NCCA Office 365 platform ▪ Review of classroom resources	September 2018
1-day face-to-face workshop	▪ Learning Outcomes review ▪ Pedagogy frameworks	October 2018
1-day face-to-face workshop	▪ Introduction to project-based tangible computing resources and activities	November 2018
1-day face-to-face workshop	▪ Research meeting	February 2019
Ongoing	▪ Online teacher support throughout the initiative for all participants through Microsoft Teams facility from NCCA Office 365 platform	Ongoing

Chapter 2: Methodology

Phase 2 of the *Coding in Primary Schools Initiative* began in May 2018 and concluded in February 2019. This chapter outlines the overall design of the initiative as well as the methods of data collection and analysis employed. As indicated above, the initiative aimed to gather data from three main cohorts – principals and teachers, parents, and children.

A case study research methodology was employed for the work with schools. Case study research allows the exploration and understanding of complex issues (Zainal, 2017). It can be considered a robust research method particularly when a holistic, in-depth investigation is required. Recognised as a tool in many social science studies, the significance of a case study methodology in research becomes more prominent when researching educational issues (Gulsecen and Kubat, 2006), sociology (Grassel and Schirmer, 2006) and community-based problems (Johnson, 2006). Yin (1984, p.7) defines case study research as *an empirical inquiry that investigates a contemporary phenomenon within its real-life context and in which multiple sources of evidence are used*. A multiple-case design methodology was used to capture the schools' use of the various technological resources for coding and computational thinking in multiple class levels and in a variety of classroom settings.

The research investigation involved collecting both quantitative and qualitative data through a mixed methods approach where the central premise is that the use of quantitative and qualitative approaches, in combination, provides a better understanding of research problems than either approach alone (Creswell and Plano Clark, 2007). Mixed methods research works particularly well for case study research as it allows the researcher to take the rich empirical data yielded from case studies and apply either quantitative or qualitative analysis to the data (Mills, 2017). Quantitative data was gathered through online surveys of principals, teachers, parents and children, while qualitative data was collected through focus group discussions, field notes, video interviews and, in the case of the children, drawings.

Data collection

Teachers: Class teachers participating in the initiative were asked to complete three online surveys using the Microsoft Forms platform. The first survey was completed anonymously by 25 teachers in May 2018 and the following two surveys, also completed anonymously, by 23 teachers during the afternoon of the final face-to-face meeting on February 6th 2019. A further three teachers, who were absent on that day, completed the surveys later. A teacher who started the initiative, and subsequently left on maternity leave, also completed the online survey — a total of 26 respondents in all. The surveys included both closed- and open-ended questions, enabling NCCA to gather qualitative and quantitative data.

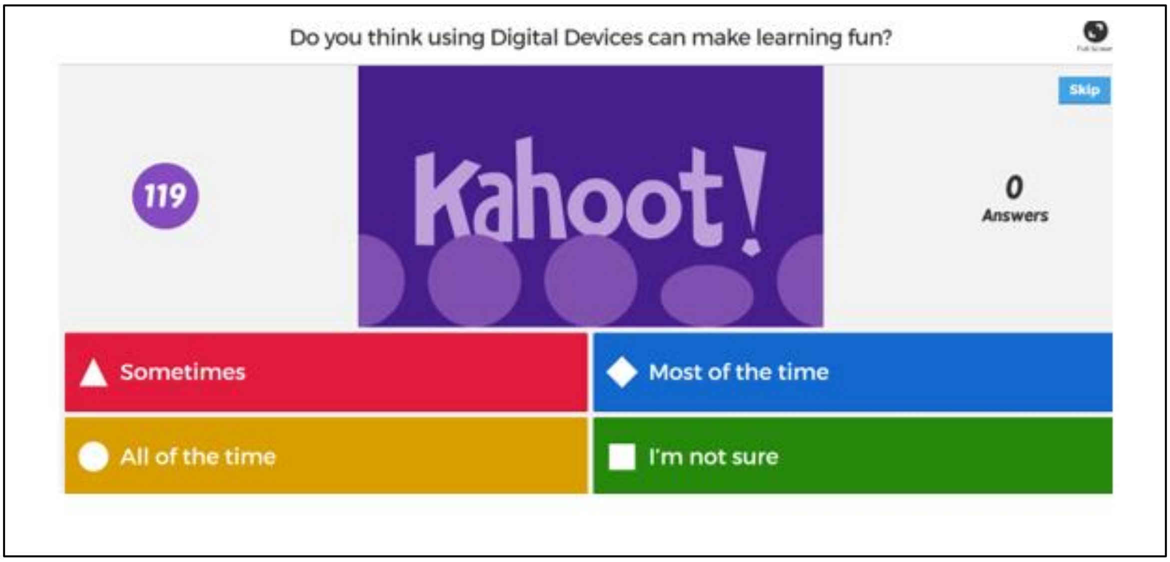
Principals: The school principals were asked to complete an online survey anonymously, again using the Microsoft Forms platform. Seventeen principals in total completed this. School visits were conducted during which the five principals were video interviewed.

Parents: Parents were invited to participate in one online survey, again hosted on the Microsoft Forms platform. Class teachers were asked to inform parents of the survey and were given a link to send home for parents to participate. The survey was open for 14 days and 146 parents completed it. Again, as with the surveys conducted with teachers and principals, all online surveys were completed anonymously.

Children: Teachers participating in the initiative were provided with some suggestions as to how they could gather children's voices. Using Kahoot, a game-based online learning platform, the NCCA provided teachers with a readymade survey they could conduct with their pupils. This platform presented questions to children in a fun and attractive way (see Figure 1), while allowing the teacher to monitor the progress of the class as they work through the survey. The application is fully in line with data protection requirements and does not collect or store any personal information from the person (in this case the child) who plays on Kahoot.

The total number of individual responses received on this survey was 284. For the younger classes, it was suggested that teachers could collect the *voices* of their children by using images or pictures. They were provided with broad guidelines for this option and, therefore, the pictures collected are not in a standardised format. In some cases, teachers scribed the voice of the child and in other instances, children themselves expressed some of their thoughts. The total number of physical artefacts returned by schools for analysis was thirty-three.

Figure 1: The *Kahoot* platform on which the children's survey was conducted



Data analysis

Online surveys: For the online surveys completed by principals, teachers, parents and children, analysis of the quantitative data involved running descriptive statistics (frequencies) and tabulating the data before presenting it in graphic form for this report.

The final question on the principal, teacher and parent survey provided respondents with an open-ended question. These qualitative responses were analysed by coding each response according to the main topic, or topics addressed. Responses were then quantified under each topic, and example responses were identified for reporting purposes. Where a response covered more than one topic, the category that the comment primarily addressed was assigned.

Focus group discussions, open survey questions, video interviews and children's artefacts
For the main qualitative pieces of data gathered, the primary method of data analysis was thematic analysis, which is a method for identifying, analysing, and reporting themes within data (Braun and Clark, 2006, p.6). Braun and Clarke (2006) provide a six-phase guide which is a framework for conducting this kind of analysis. The process itself is interactive and

reflective, develops over time and involves a constant moving back and forward between phases (Nowell, Norris, White & Moules, 2017). Table 2 summarises the process of analysis applied, in keeping with the Braun and Clark (2006) framework.

Table 2: Six-step approach to thematic analysis

Step 1: Become familiar with the data	Step 2: Generate initial codes	Step 3: Search for themes
Step 4: Review themes	Step 5: Define themes	Step 6: Write-up

Step 1: Becoming familiar with the data

Initially, repeated reading of the data was conducted in an active way searching for meanings and patterns. Before coding began, the entire data was read through, with ideas and possible patterns being identified as familiarity with the data grew. At this stage, ideas for coding were noted.

Step 2: Generating initial codes

The second step involved systematically working through the entire data set, giving full and equal attention to each data item. Interesting aspects in the data items that may form the basis of themes were noted. For this step, the data was organised in a meaningful and systematic way using a spreadsheet. Peer debriefing and reflexive writing throughout the coding process was used, which helped examine how ideas evolved.

Step 3: Searching for themes

The third step began when all data had been initially coded and collated, and a list of the different codes identified across the data set had been developed. This phase involved sorting and collating all the potentially relevant coded data extracts into themes.

Step 4: Reviewing themes

Once a set of themes was devised, the coded data extracts were reviewed for each theme to consider whether they appeared to form a coherent pattern. The validity of individual themes was considered to determine whether the themes accurately reflected the meanings evident in the data set as a whole. Some themes collapsed into each other.

Step 5: Defining and naming themes

The fifth step determined the aspect of the data each theme captured and identified what was of interest about them and why. For each individual theme, a detailed analysis was written, sharing the story that each theme told. Theme names were chosen to give the reader an immediate sense of what the theme was about.

Step 6: Producing the report

Once the themes were fully established, the final step consisted of writing up the analysis. The write-up of a thematic analysis provides the reader with concise, coherent, logical and interesting account of the data within and across themes.

Research ethics

Mukherji and Albon (2015) state that research ethics should be of central importance in any research project and this initiative applied the highest ethical principles to data collection and analysis. All participants were treated with respect and courtesy throughout the initiative, their views reported accurately, and full confidentiality and anonymity were respected in the data storage, analysis and reporting. Informed consent was invited from each participant.

There is some debate in the literature about the vulnerability of children in research contexts (Dockett, Einarsdóttir and Perry, 2011) but the agentic nature of children’s involvement in relevant research is now recognised as important (Harcourt and Conroy, 2011) provided that informed consent of parents accompanied by assent from children is gained and that appropriate data gathering methods are used. Both parental consent and child assent were requested for the participation of the children in the initiative.

Chapter 3: Phase 2 findings

Findings from each cohort—principals and teachers, parents and children—are presented in stand-alone sections for clarity. These findings result from the data collection and analysis processes described in Chapter 2 and they form the basis for the discussion and recommendations that will be presented in the final chapters.

Principals and teachers

In this section, four main themes are used to present the findings:

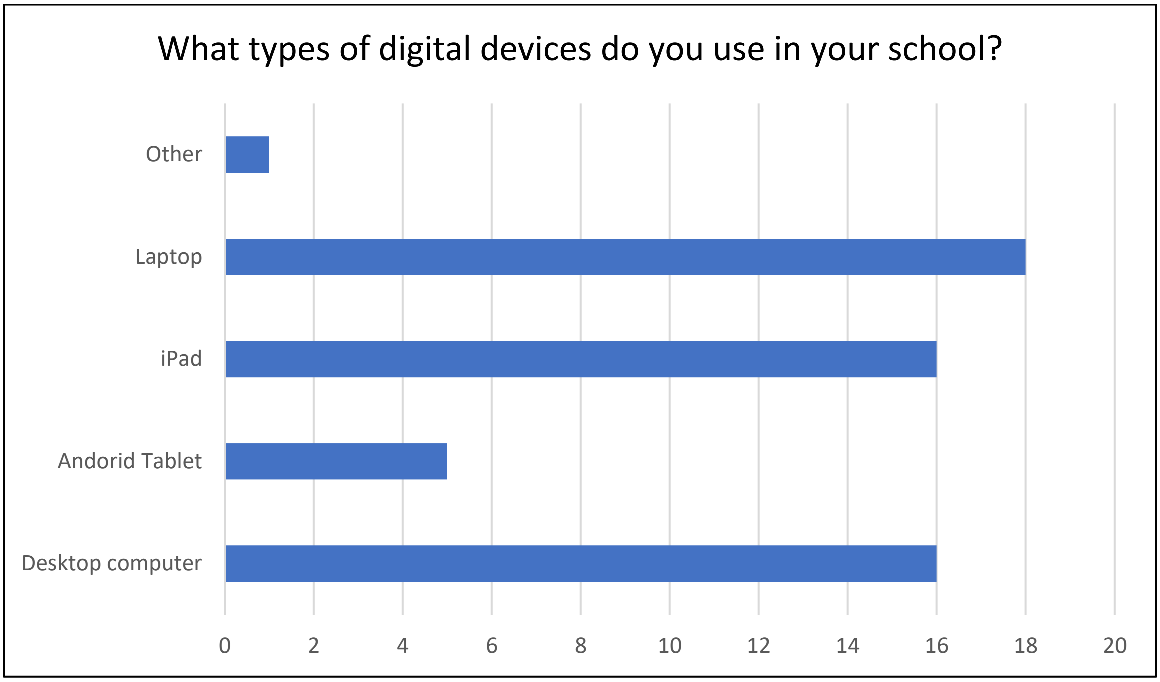
- Classroom experiences and practice
- Curriculum alignment
- Continued professional development
- Resources.

Within each of these, further sub-headings are used to group and present findings.

Contextual information

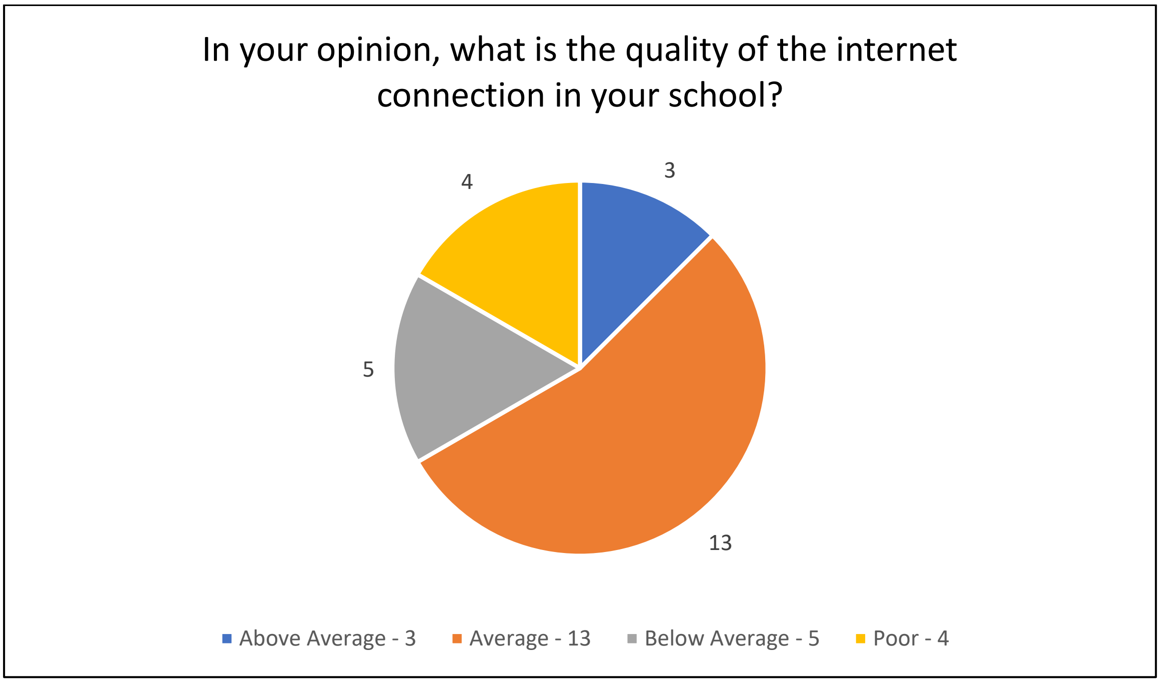
As the schools selected for this phase of work had little or no experience in coding it was important to establish the types of technology, if any, used in each school. They reported having multiple devices either within classrooms or shared across the school. These devices included desktop computers, Android tablets, iPads, and laptops. Figure 2 shows an overview of the types of devices used within the schools. Row 1 titled 'Other' refers to mobile phones.

Figure 2: Types and quantities of digital devices currently used in participating schools



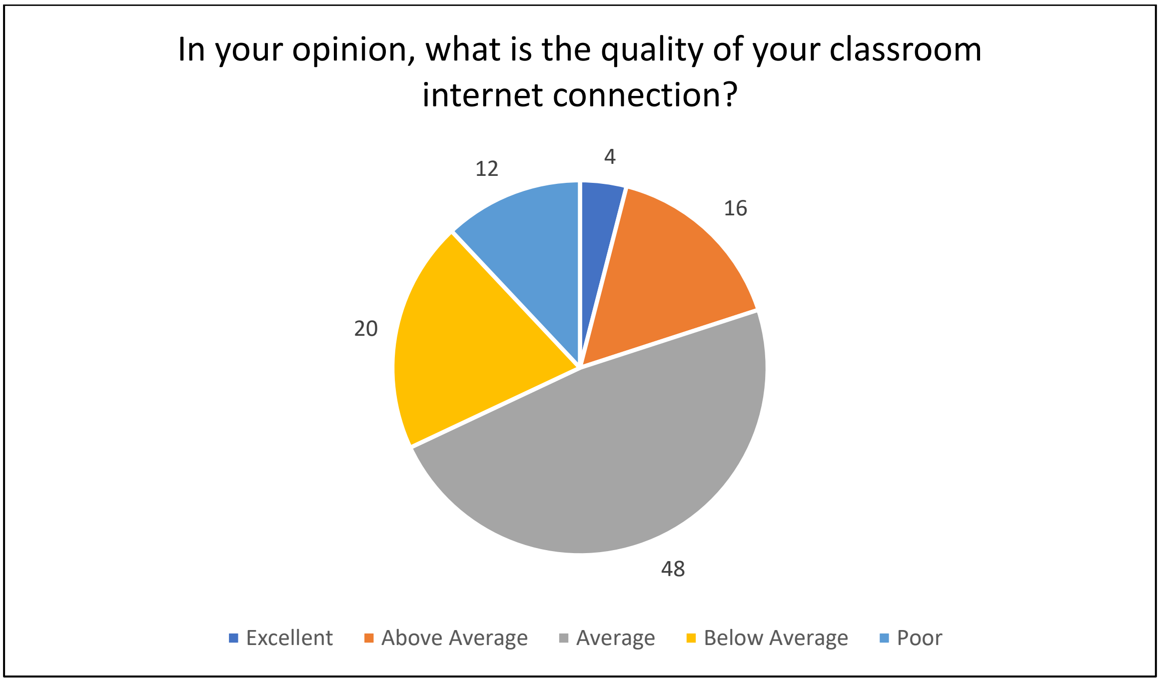
The schools were asked about access to broadband and the current performance of the internet in their school and, more importantly, in their classrooms. In the case of whole school access to the internet, three schools reported having above average internet access, 13 schools average, five below average and four poor (see Figure 3).

Figure 3: School access to the internet



The teachers were also asked about internet access in their classrooms. In this case, (see Figure 4) one teacher regarded the access as excellent, four above average, 12 average, five below average, and three reported it being of poor quality.

Figure 4: Classroom access to the internet



Prior to the start of this second phase of the initiative, the participating teachers were asked for their thoughts regarding coding in the curriculum and from what class level they thought coding should be introduced. Seven responded saying they were unsure, two thought 3rd and 4th was the most appropriate class level and the remaining 16 considered junior infants to be the ideal starting point. When asked whether they thought coding should be taught as a separate subject the group of 25 were split with 12 thinking it should be and 10 saying it should not, while three were unsure. There was consensus, however, in their opinions on the potential challenges to integrating coding into the primary school curriculum. All mentioned a combination or variations of the following: teacher skills and confidence, IT infrastructure, and time for teacher learning given the demands of the current primary curriculum. They also noted the possible need for additional funding to purchase equipment or to upgrade broadband and Wi-Fi capabilities within their classrooms.

While a number of challenges were identified, 24 of the 25 participating teachers agreed that it is necessary to introduce coding into the curriculum. They identified several reasons for this. Open-ended responses included:

> *We need to give pupils skills necessary for the future*
>
> *We live in an increasingly developed technological world and we need to equip children for that world*
>
> *I think it is something that will engage pupils and enhance creativity and problem-solving skills*

It is important to note that the one respondent who thought it was not necessary to introduce coding identified the shortage of time in an already overcrowded curriculum as the reason, rather than coding itself being unimportant for children's educational experiences:

> *Mainly because I struggle to see where it would fit into the school day. There isn't enough time at the moment, but I would love to do it*

The importance of integrating a coding- and or technology-based competency into the primary curriculum was evidenced in the initial applications by schools hoping to participate in this initiative. As part of the selection process, schools were asked to explain why they would like to be involved in Phase 2 of the initiative. The range of open-ended statements from the schools selected included:

> *We are living in a world that is dominated by technology, as teachers it is our responsibility to try and prepare our students for their future*
>
> *This generation of students will experience a world even more online and digital than the one we are living in today*
>
> *It's vital for children to function and blossom in a world where digital technologies are the core of every functional device and service they will encounter in learning and work*

Similar statements to these were noted during work with the teachers from Phase 1 of the initiative. They also clearly align with some of the broader statements made by countries in the NCCA Curriculum Investigation (NCCA, 2018) outlining the rationale for including coding or digital technology in their primary school curricula.

Summary

Although the teachers had limited or no experience in coding or computational thinking, each of the schools was already using a range of devices such as iPads, tablets, laptops and PCs in their teaching practice. When asked about internet connection within the school and their classrooms there was a wide range of experience from poor to excellent, with most teachers saying they considered their access to be average.

Most teachers agreed that teaching coding and computational thinking should be introduced to the curriculum and from an early age. However, when asked whether it should be taught as a separate subject their responses were divided, with no clear majority for either option of teaching it as a separate subject or integrated across subjects.

Curriculum overload, IT infrastructure, and teacher confidence were all viewed as potential challenges to the inclusion of coding and computational thinking in a redeveloped curriculum.

Learning outcomes

Process

To support the investigation by the Phase 2 schools of a physical computing and play-based pedagogical approach to coding and computational thinking a set of learning outcomes was developed for use from junior infants to sixth class. The development of the learning outcomes reflects the findings from the NCCA Curriculum Investigation and from Phase 1 of the school-based initiative.

Table 3 sets out the learning outcomes which the Phase 2 schools used during the initiative—one set for junior infants to second class, and one set for third to sixth class.

Table 3: Draft Learning Outcomes

Learning Outcomes for junior infants – 2nd class
1. Give step-by-step instructions and act according to instructions.
2. Explain how algorithms can be used to express ideas or to solve problems.
3. Demonstrate that an algorithm is a precisely defined set of instructions for completing a task or goal.
4. Design solutions to solve simple problems using a sequence of steps and decisions.
5. Decompose (break down) the steps needed to solve a problem into a sequence of instructions.
6. Demonstrate that programmed devices follow specific instructions to complete or carry out a task.
7. Create a simple programme containing sequences and simple loops.
8. Create a simple programme using visual instructions or tools that do not require a textual programming language.
9. Plan using logical reasoning and problem-solving strategies when creating programmes using a visual programming language.
Learning Outcomes for 3rd – 6th class
1. Test and debug programmes that include sequences, events, loops and conditions.
2. Create algorithms to solve problems that accomplish specific goals.
3. Use problem-solving strategies to generate solutions recognising that different solutions may exist for the same problem.
4. Plan and execute programmes in a visual programming language.
5. Follow the problem-solving process, design and implement digital solutions using algorithms that involve decision-making and user input.
6. Write programmes, individually and collaboratively, that include sequences, variables, events, loops and conditionals.
7. Write programmes that receive and respond to real world inputs and displays using a range of computing outputs.
8. Work collaboratively to design programmes that accomplish specific goals including control or simulation of physical systems.

Teacher feedback

All teachers who participated in Phase 1 and Phase 2 of the initiative were asked to reflect on their classroom experiences and to comment on the newly proposed learning outcomes. They were asked to reflect on each outcome and to choose from three statements: 'I don't understand it', 'I kind of understand it' or 'Okay'. They were also asked to provide additional opinions or thoughts on each outcome if they wished.

Figures 5 and 6 below give a snapshot of the types of responses to the learning outcomes for the range of class levels. Figure 5 shows an example of the feedback in relation to learning outcome 1: 'Give step-by-step instructions and act according to instructions', for infants to second class. Figure 6 relates to learning outcome 8: 'Work collaboratively to design programmes that accomplish specific goals including control or simulation of physical systems', for third to sixth class.

Figure 5: Reflections on Learning Outcomes for junior infants to second class

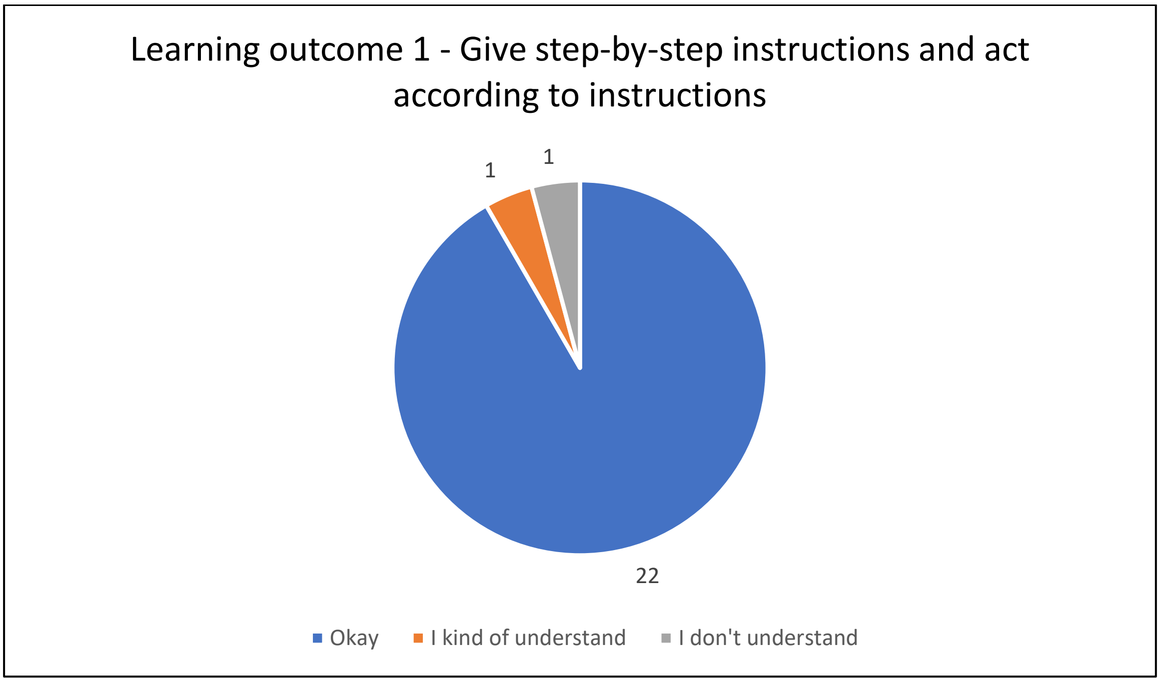


Figure 6: Reflections on Learning Outcomes for third to sixth class

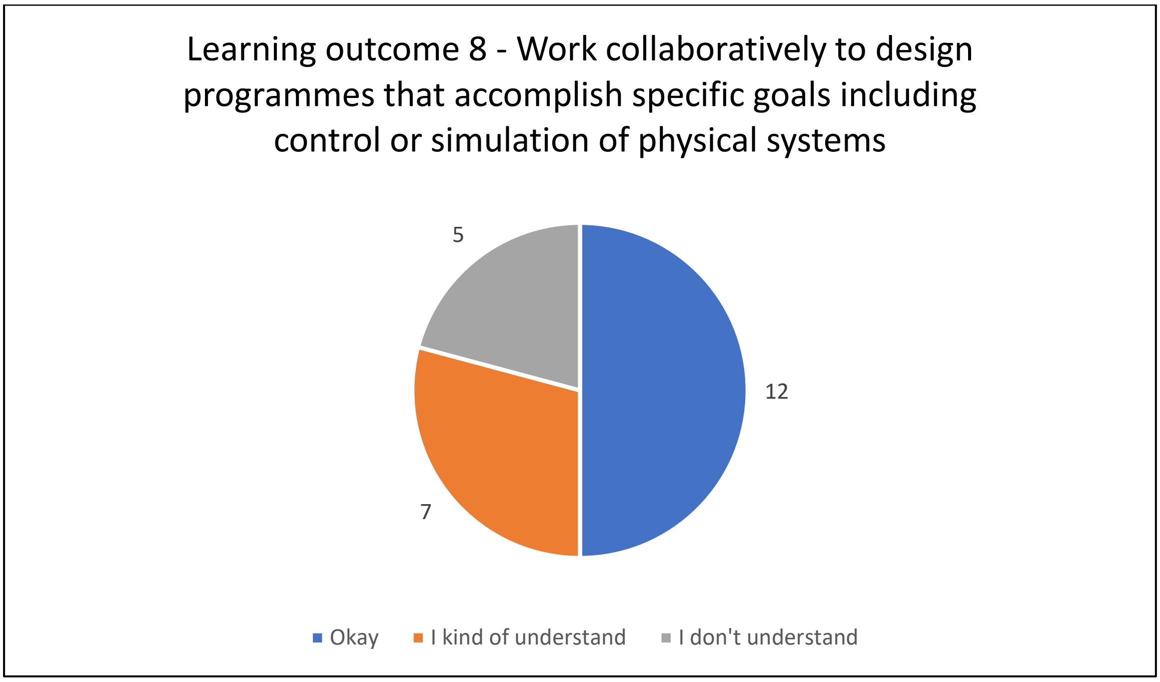


Table 4 gives a sample of the types of open-ended responses given by teachers.

Table 4: Sample of open-ended responses to Learning Outcomes

Learning Outcome	Response
Work collaboratively to design programmes that accomplish specific goals including control or simulation of physical systems	▪ *I don't understand what 'control or simulation of physical systems' means* ▪ *I had to read this a couple of times before it made sense to me* ▪ *Maybe some examples of the physical systems would be beneficial* ▪ *Add in *definition of physical systems and examples of specific goals*
Give step-by-step instructions and act according to instructions	▪ *This is a clear and logical way of introducing the concept of an algorithm without having to use language that might be overly complex for younger children*

	▪ *Outcomes are generally from the viewpoint that the "child should be able to..."; bearing this in mind I think the outcomes are worded a bit unclearly*
Explain how algorithms can be used to express ideas or solve problems	▪ *This is clear to me. I see it as an algorithm being a systematic step by step way of working through a problem. The expressing idea part is less clear to me though* ▪ *Not just sure how algorithms can express ideas* ▪ *Okay, but the term algorithm may be too difficult for infants might be better to use set of instructions or step*

Summary

Comments from the Phase 2 teachers on the learning outcomes were generally positive. However, there was a consensus that the technical terminology in some of the outcomes, particularly for senior classes, would need to be amended further.

Most of the teachers from Phase 2 felt that the language used in the outcomes was too technical and that there needed to be clearer explanation of some of the terms. Some also stated that they did not see a clear enough progression from the early years outcomes through to the senior classes.

Many teachers felt that it would be useful to have exemplars for each outcome as a reference, to enable teachers interested in coding and computational thinking to see what each learning outcome might look like in a practical classroom context.

Classroom experiences and practice

Informed by the Phase 1 schools' experience and the NCCA Curriculum Investigation, the initiative introduced the teachers to the concept of a coding continuum of unplugged, plugged and physical/tangible computing devices. The teachers were asked to reflect on these in the context of their classrooms and were given complete autonomy as to which resources or approaches they would use for the duration of their work in the initiative.

They were also asked to investigate and possibly integrate some of the play-based, thematic or project-based pedagogical approaches they had been introduced to during the face-to-face workshops. Again, they were given complete freedom to implement those approaches in the classroom, or not. They were then asked to reflect on their classroom activities, the children's participation, and their overall thoughts on the work, both negative and positive.

The following questions were asked during informal group discussions recorded by an NCCA Education Officer and as part of an online survey completed individually by the teachers.

- Do you see playful, collaborative and project-based learning as a useful pedagogy when teaching code?
- How engaging was this project for the children you work with? Were the children engaged throughout? Were they excited/interested and did it last?
- How did you implement the activities? Whole class, small groups or individually?
- Was using unplugged, plugged and tangible computing strategies an effective way of introducing CT/coding skills to your class?
- Did the children become less reliant on you, as the teacher, to complete activities?
- In terms of assessment, what did you use to monitor the progress of children's learning?

Playful, project-based pedagogical approach / Children's engagement

> *I feel using classroom methodologies, like Aistear, left a far more meaningful and relatable influence and experience for my class*
>
> *Opened me up to a whole new world; went back to college because of this project and upskilled myself*

Most teachers commented that taking a more playful, or child-led, approach was very rewarding. They also stated that the children were actively engaged and enthusiastic throughout the initiative. Some went so far as to say that it was the most engaged, they had ever seen their students.

> *It's helped some of the boys socially; it's helped those low academic achievers realise that they do have strengths just in a different way to 'normal' classroom activities*

For some, it was a very different approach to how they usually worked in the classroom and they were pleasantly surprised at how independent the children had become. Others commented that, because of the open-ended nature of the activities, there was not the same pressure to have the same solution or answers as others.

> *I must admit that I was surprised at how much they took to the lessons and coding*

Others had commented that play, collaborative and project-based learning was the most effective way to teach coding/computational thinking. They found their children were working collaboratively in groups, learning from each other, and that the open-ended tasks facilitated this much more than more closed tasks and activities.

> *Aistear, the construction table with coding was the most anticipated station each week!*

> *When I introduced them to the Microbits they were so excited. One girl nearly jumped out of her skin with excitement!!*

Some teachers commented that regardless of the lesson content, when coding and play were used as a methodology in other subject areas, children were always keen to participate.

> *They were informing parents about what we were doing, which was then leading to further involvement and cooperation from them*

Several teachers also observed that using a play-based approached enabled them to join in with the class activities in a different way. Not being an 'expert' in coding or computational thinking wasn't so much of an issue as everyone was experimenting, failing and learning at their own, or the group's pace. Some children became natural leaders within their groups and most teachers were happy to play a guiding role.

> *A whole different way of teaching; you have to let go*

> *I learned that I don't need to be the expert; that there will be kids who will teach me*

I learnt that you have to give control to the children at certain times and take a step back from teacher-led lessons

Implementation

Teachers described how they organised their children to participate in coding/computational thinking activities. There was consensus that the following practices were the most effective:

- station teaching
- individual activities
- paired activities
- small group activities
- whole class
- coding corner.

Teachers commented that, in some instances, how they approached this depended on the lesson or devices they were using. Some were restricted by the number of devices they had in their classrooms. For example, a school that had a mixture of iPads, laptops, Lego Wedo or Microbits would arrange their station teaching accordingly.

The children were introduced to and discussed the topic on a Monday and then worked in mixed ability pairs throughout the week in the 'coding corner'

Generally, a topic or activity was introduced to the whole class and then the follow-up lessons were broken down into unplugged, plugged or physical/tangible device activities for small groups of children or in pairs.

Stand-alone lessons to teach specific skills, allow time for free play and tinkering and then integrated lessons with focus on specific tasks

Typically, when children worked/played in pairs or groups to solve problems or start project work, there would be initial teacher guidance. They then would be given deadlines or targets, depending on the activity, and left to work independently to complete tasks, with teacher supervision.

There were always opportunities for stronger kids in particular, to go further than the planned activities and adapt/improve projects, using their own ideas

Most teachers observed that as lessons progressed there was a notable development in the independence levels of the children. Some children still needed help but working in groups in a playful way helped less confident children to learn from their peers and to experience success in their work.

I was a shadow in the room and I also found students tended to remain on task longer

They also became very good teachers themselves, helping others if people were stuck

Unplugged, plugged and tangible devices

The NCCA welcomed the opportunity to collaborate with the PDST Technology in Education team. The purpose of this collaboration was to facilitate an introduction to coding/computational thinking for the teachers participating in Phase 2 of the initiative. A two-day workshop in May 2018 introduced the teachers to the concept of computational thinking using 'Uplugged' and 'Plugged' activities and to the wide range of exemplars that had already been developed by the PDST and published on their website. An 'Unplugged' activity being defined as an activity that can be conducted without the use of computers or electronic technology. The workshop also focused on the use of physical or tangible computing devices to support progression in the children's coding/computational thinking skills through junior infants to sixth class. The message that 'unplugged' approaches are useful but that they must be clearly linked with progression to 'plugged' activities when learning how to code was reiterated throughout the two-day workshop.

Teachers were asked to reflect on the effectiveness of the transition from unplugged to plugged activities and then on to tangible or physical digital devices. The overall response was very positive. Most, if not all, teachers commented that starting a lesson or a project with an unplugged activity was very effective. Some said that it enabled the children to learn the basics of coding without using a device and others thought their children were able to apply unplugged strategies to plugged tasks very effectively. Others found that their children became less reliant on the teacher and were able to problem-solve and think more independently, particularly when completing either unplugged or physical/tangible computing tasks.

> *Doing this project made me realise how valuable unplugged lessons were; it makes the coding experience more effective*

> *I very much took a back seat once the children had the basics particularly when we were doing the unplugged activities*

Teachers also thought there was a natural progression between the different strategies and having a variety of unplugged coding/computational thinking activities allowed them to continue with lessons whether the children had devices or not. They also noted that the skills the children developed were transferred easily to their work with tangible devices.

Generally, most teachers taught unplugged activities with the whole class, plugged activities in groups of no more than three, and worked with physical/tangible devices in slightly larger groups. One or two teachers found that after completing some unplugged activities they moved straight to a physical/tangible device such as a Bee-Bot to reinforce what was learned before moving on to a plugged activity.

Assessment

Several teachers commented that they were at the very early stages of deciding what assessment tools or strategies would be most appropriate for capturing the learning from the wide range of activities their children participated in. The following list gives a snapshot of the various tools and strategies most teachers used:

- teacher observation
- peer assessment
- class presentation of children's projects
- class, group and individual conferencing
- samples of children's artefacts
- assessment sheets

- photo, video and voice recordings
- coding journals
- children's learning logs
- e-portfolios/digital journals.

Several teachers used a variety of online tools such as Seesaw, a digital portfolio app, which they found very useful. The application is fully in line with data protection requirements and complies with all European Union GDPR guidelines.

> *I found the app very easy to use; it was time-efficient, gave the children ownership of recording their own work and provided us with a platform to share their work with parents*

Others used class devices such as iPads and tablets to capture videos, photos and sound recordings of the children's projects or presentations. A small number of teachers found that some of the resources they were using such as Lego Wedo and Code.org had integrated assessment sheets and online tools for documenting progress.

Summary

When asked about a playful, project-based approach to teaching code and computational thinking, most of the teachers commented that the approach was very rewarding. They found their children were actively engaged and enthusiastic, particularly because of the open-ended nature of the activities. They also found there wasn't the same pressure on children to have the same solution to problems or tasks associated with other classroom activities. In addition, the process enabled their children to learn from each other and gave some children the opportunity to become natural leaders. Failure was not an issue but gave further opportunities to learn. Most teachers commented that the playful aspect of the activities gave children the chance to become the experts and teachers were happy to play a guiding role.

Teachers in general found that the transition from unplugged to plugged activities and then on to tangible or physical digital devices to be very effective when teaching code or computational thinking concepts. A number found that starting a lesson or a project with an unplugged activity was very effective and that children become less reliant on the teacher during plugged and tangible tasks after they had been exposed to a related unplugged activity.

Teachers commented that they used various tools and strategies when assessing what their children had done in class. Some teachers used recommended online tools, while others accessed integrated assessment sheets and online tools for documenting progress from the resources they had used.

Curriculum alignment

To understand more fully the essence of computational thinking, and the implications of introducing coding and computational thinking into a redeveloped curriculum, the NCCA commissioned a review of literature in the area. Conclusions from the review showed that computational thinking is arguably the correct focus in the implementation of coding in the Irish primary school classroom and that it can, and should, be supported and developed through activities in every subject.

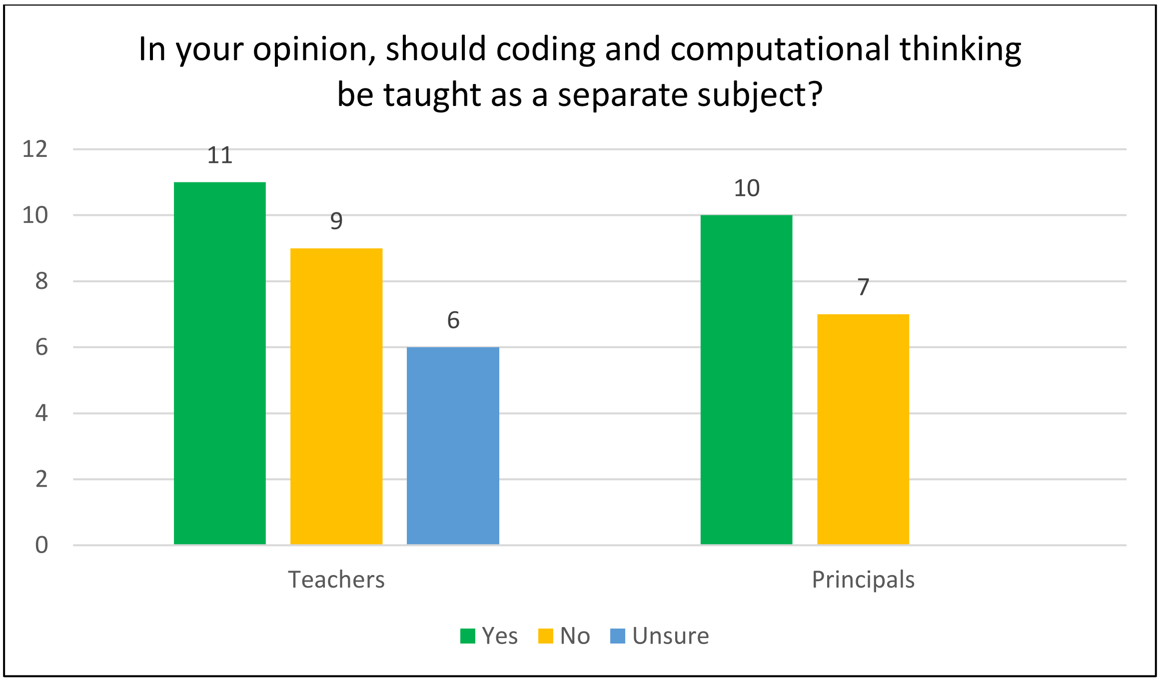
Considering the findings from the review of literature on computational thinking (2018), participating teachers were asked to reflect on their classroom practice and to consider the implications for a redeveloped primary school curriculum. Principals, too, were asked to reflect and give their opinions. Responses were sought to the following questions:

- Did you integrate computational thinking and coding into other curriculum subjects or see it as a stand-alone subject?
- Where best do you see coding and computational thinking sitting within the primary curriculum?

Stand-alone or integrated?

Participating teachers and principals were asked whether they thought coding and computational thinking should be taught as a separate subject. Some 35% of the teachers and just over half of principals agreed with this approach. However, a significant number of teachers were still unsure (see Figure 7).

Figure 7: Principal and classroom teacher perspectives about coding and computational thinking as a separate subject



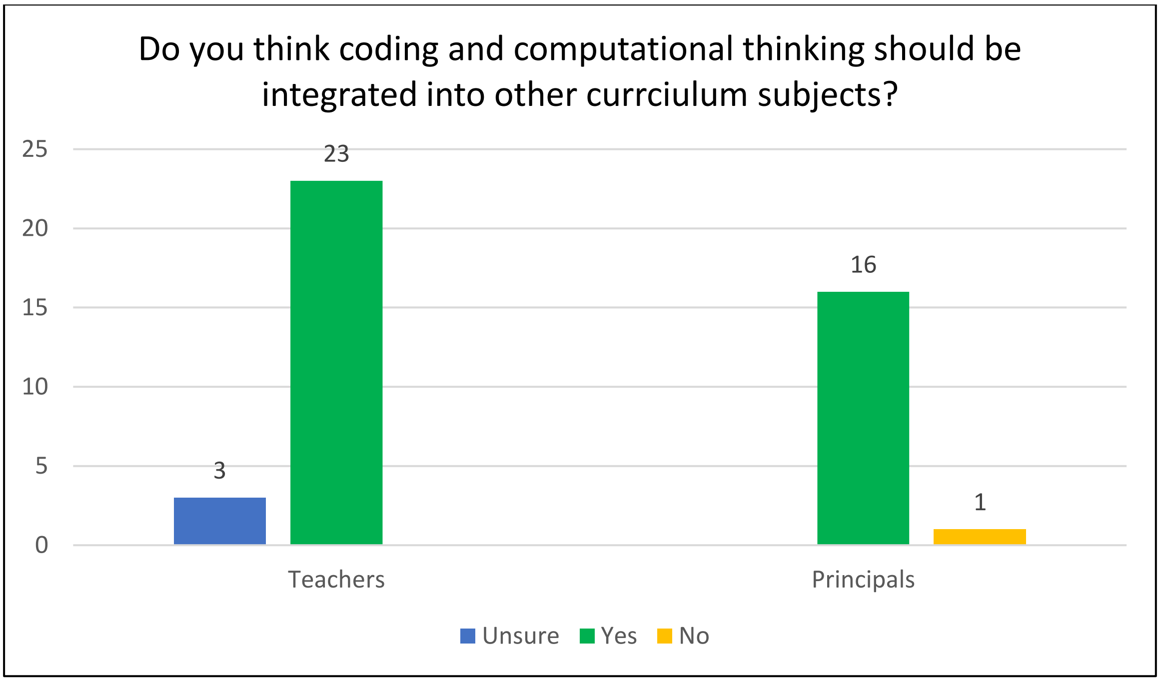
I think there needs to be stand-alone lessons to teach the children the fundamentals before expecting them to be able to complete any tasks

Some teachers felt that children would need to become comfortable with the basics of coding before using it as a means of creating or exploring in other subjects. Others thought that there was a danger that if it wasn't a stand-alone subject, it might receive limited attention given the busyness of the school day.

Somewhere in between -I feel coding can integrate well - once the children have learnt the skills

When asked whether they thought coding/computational thinking should be integrated into other curriculum subjects most teachers and 94% of principals agreed. (see Figure 8).

Figure 8: Principal and classroom teacher perspectives about coding and computational thinking and subject integration



Stand-alone lessons to teach specific skills, allow time for free play and tinkering and then integrated lessons with focus on specific task

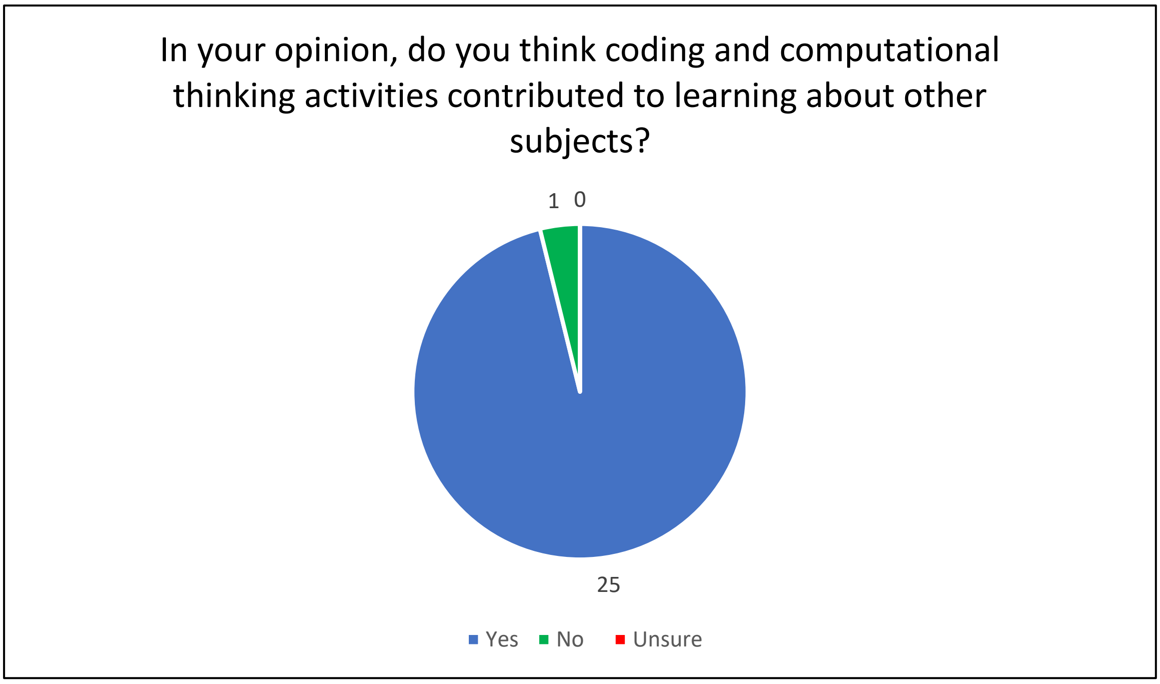
Somewhere in between. I think it would work well timetabled as a stand-alone subject and then also integrated into other subjects when relevant

However, teachers commented that children would need to complete introductory coding and computational thinking lessons first, to gain a basic understanding of the fundamentals before it could be integrated.

I think the best thing would be to have it integrated with other subjects but leaving a bit of leeway to be able to do some stand-alone lessons to get the children started

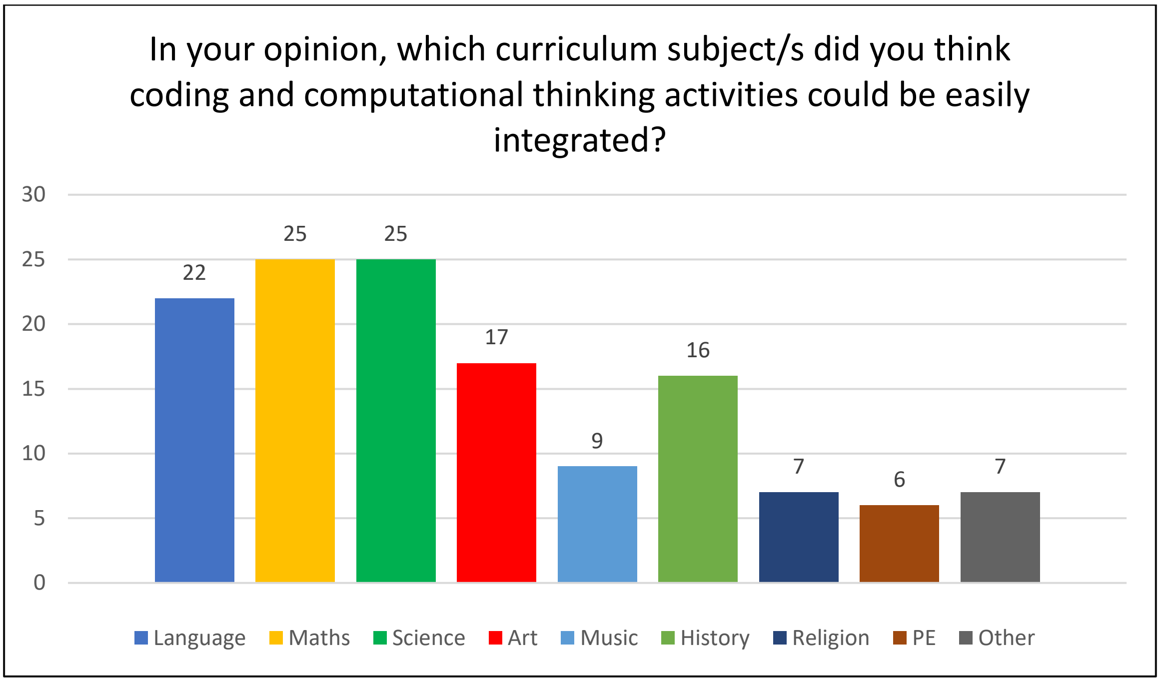
Teachers were asked whether, from their experience, they felt that coding and computational thinking contributed to cross-curricular learning. A significant number agreed (see Figure 9).

Figure 9: Classroom teacher perspectives coding and computational thinking and interdisciplinary knowledge



Teachers were asked to reflect on whether they had integrated coding and computational thinking into other curriculum subjects or treated it as a stand-alone subject during the initiative. Figure 10 shows their responses, subject by subject.

Figure 10: Classroom teacher perspectives on integrating coding and computational thinking in other curriculum subjects



Mathematics and science were identified as being most appropriate for curriculum integration, but teachers saw potential for integration across other subjects too.

> *Coding naturally lends itself to Maths and Science but would also work very well in Literacy, Art, Geography and Music*

Teachers shared examples of how they taught coding and computational thinking as a separate subject or in an integrated way during the initiative.

> *We integrated with Irish through creation of 'towns' the children had to navigate around using Irish directions and vocals with the Bee-Bots*

> *I used the rhymes and games and adapted them where necessary, e.g. "get the Bee-Bot to the words that rhyme with wall (from Humpty Dumpty)"; "how many syllables in alligator?" Clap and count...you can move your Bee-Bot that many steps to get to his goal*

> *I integrated Microbits and Science, the children coded Microbits to measure temperature*

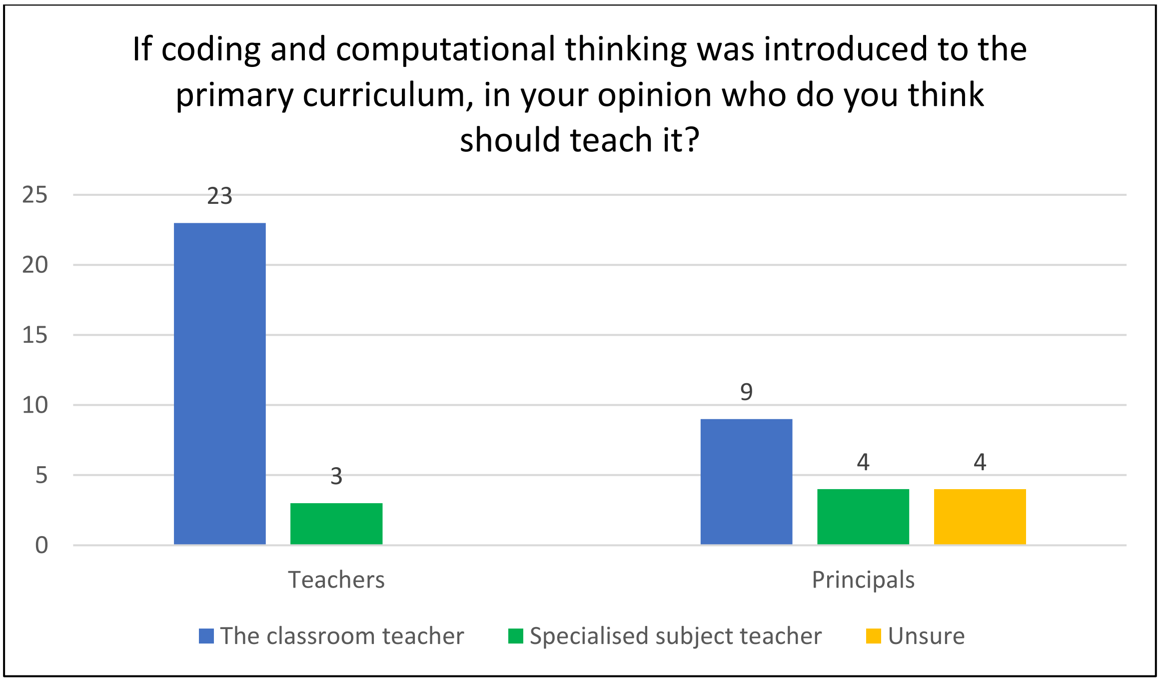
However, when starting to work with coding in their classrooms, most teachers introduced coding and computational thinking as a stand-alone subject. In their opinion, it was important to learn the skills first before being able to integrate.

> *I felt at the beginning it was important to develop the children's competence at coding*
>
> *However, over time I have begun to use coding programmes as part of other lessons*

Implementation

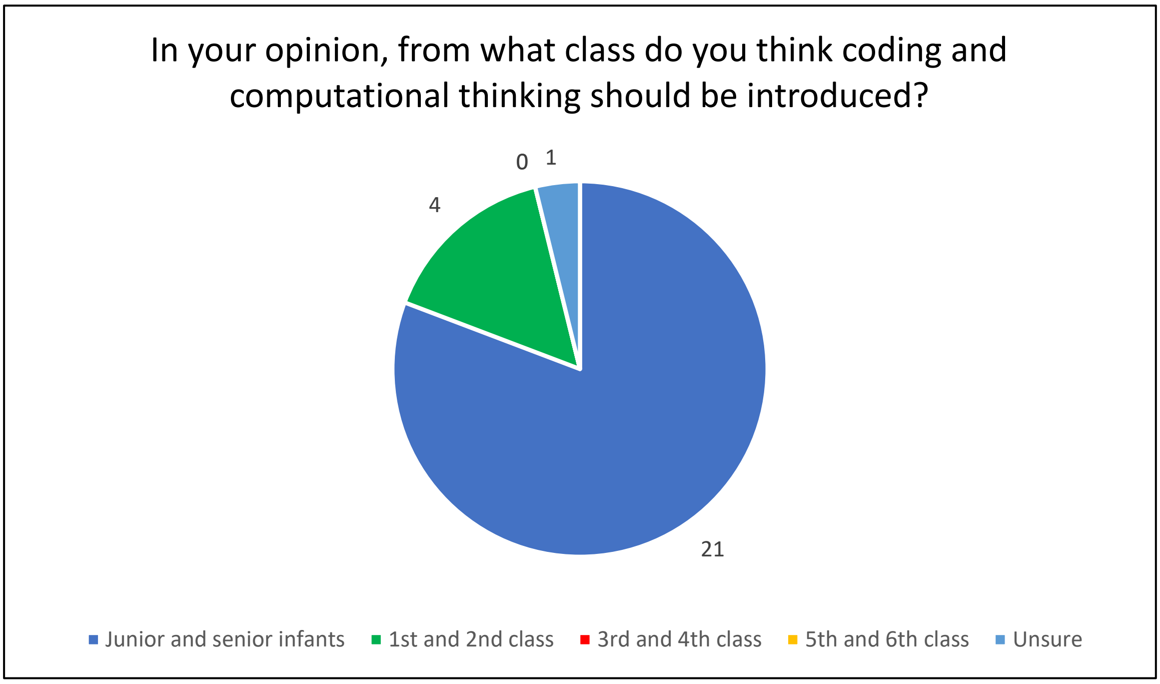
When classroom teachers were asked about who they thought should have the responsibility for teaching coding and computational thinking in the classroom, most thought that classroom teachers should be given this. However, principals seemed to be less confident (see Figure 11) with just over half surveyed thinking it would be a classroom teacher's responsibility whilst the remainder were either unsure or thought it should be the job of a specialist teacher.

Figure 11: Principal and classroom teacher perspectives on who should teach coding and computational thinking



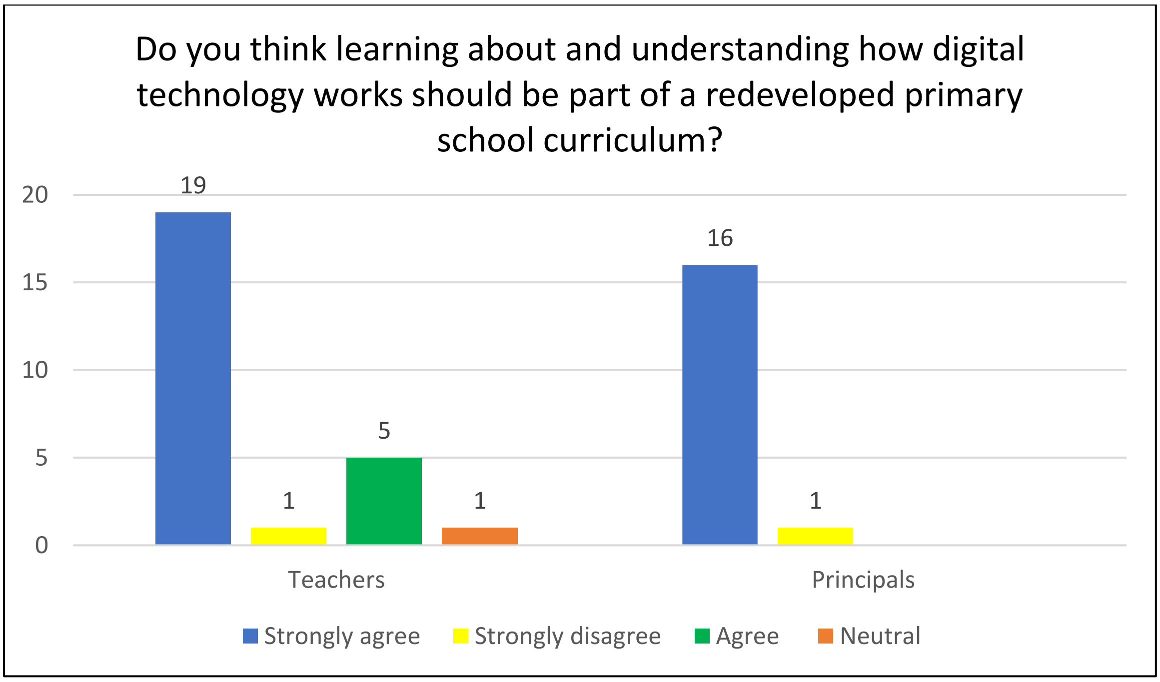
Teachers were asked at what class level they thought coding and computational thinking should be introduced. Most teachers surveyed, 81%, thought that coding and computational thinking activities should be introduced from junior infants onwards (see Figure 12) with four thinking first and second class, and one teacher expressing uncertainty.

Figure 12: Teachers' perspectives on introducing coding and computational thinking



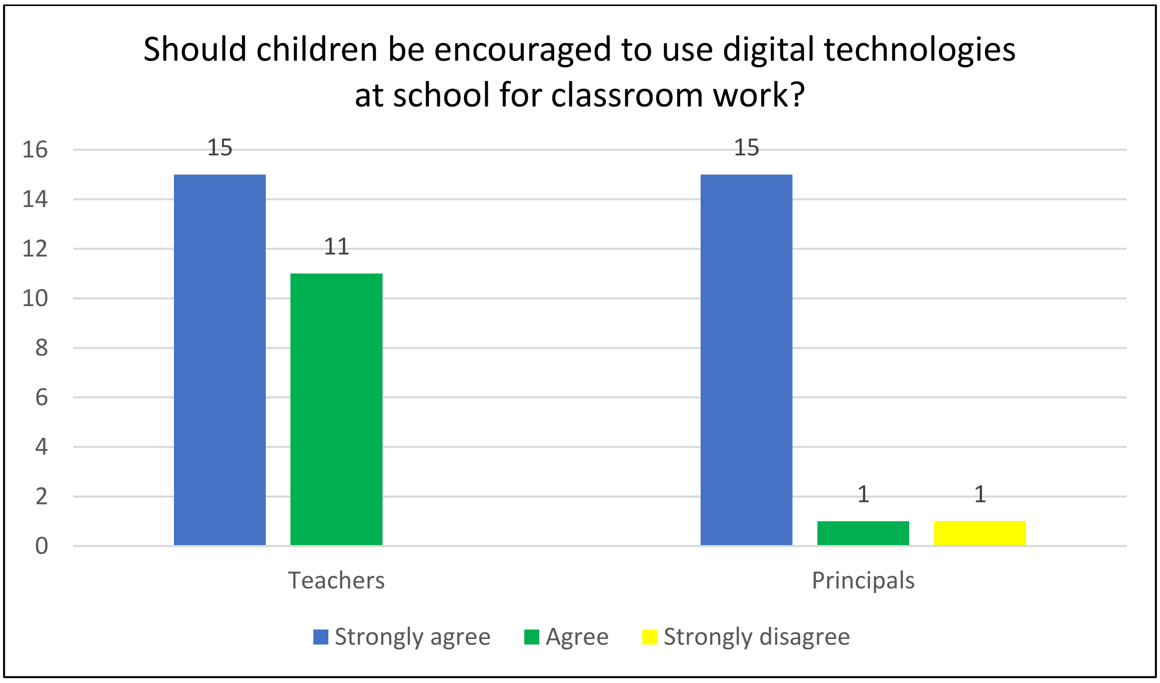
When teachers and principals were asked whether they thought learning about digital technology in school was necessary 92% of the teachers and 94% of school principals (see Figure 13) either agreed or strongly agreed.

Figure 13: Principal and classroom teacher perspectives on the importance of learning about digital technology in school



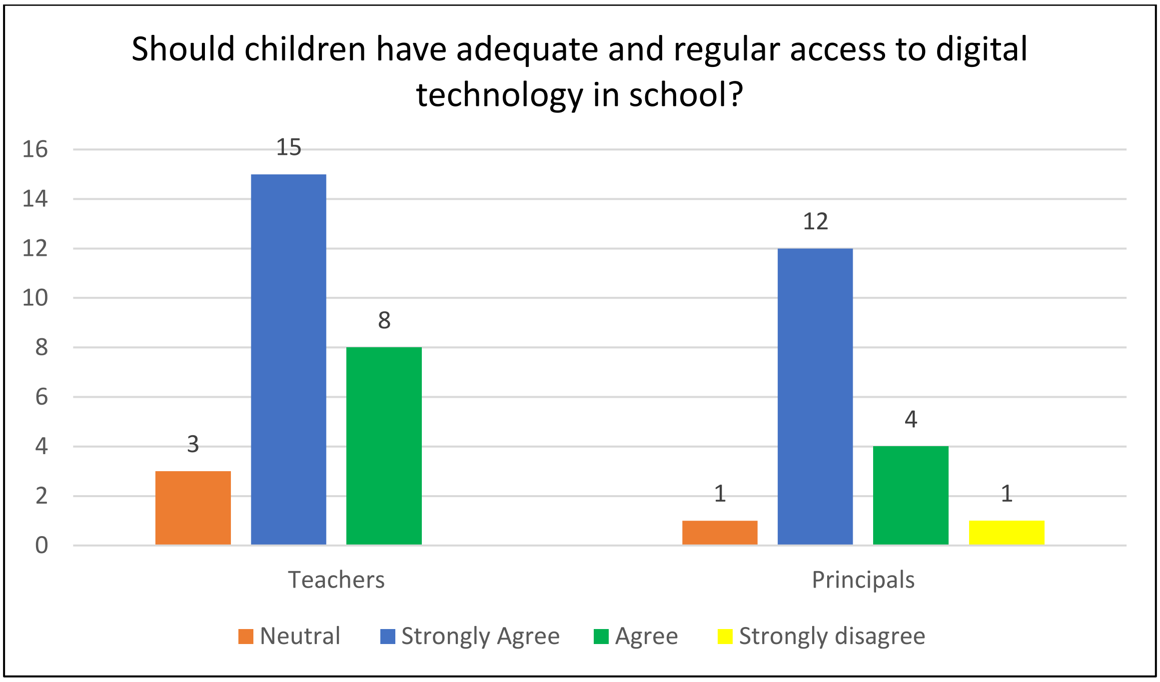
Teachers and principals were also asked if they thought their children should be encouraged to use digital technology at school for work/projects in the classroom. In total, 94% of the teachers and most school principals surveyed either agreed or strongly agreed that children should be encouraged to use digital technology to complete classroom work (see Figure 14).

Figure 14: Principal and classroom teacher perspectives on using digital technology in school



Finally, when asked whether they thought children should have access to digital technology in their classrooms 88% of the teachers and 94% of principals either agreed or strongly agreed (see Figure 15).

Figure 15: Principal and classroom teacher perspectives on the importance of access to digital technology in school



Challenges

Several teachers were of the view that while they saw potential for integrating coding and computational thinking into multiple curriculum subjects, there might not be widespread agreement with that view. They suggested that unless it was identified as a separate subject, teachers might not want to teach it.

> *I think CT should be integrated into other subjects with assessments to be carried out to ensure that it is being completed*

> *I think if it's not a stand-alone subject it may get left as the teaching day is so busy*

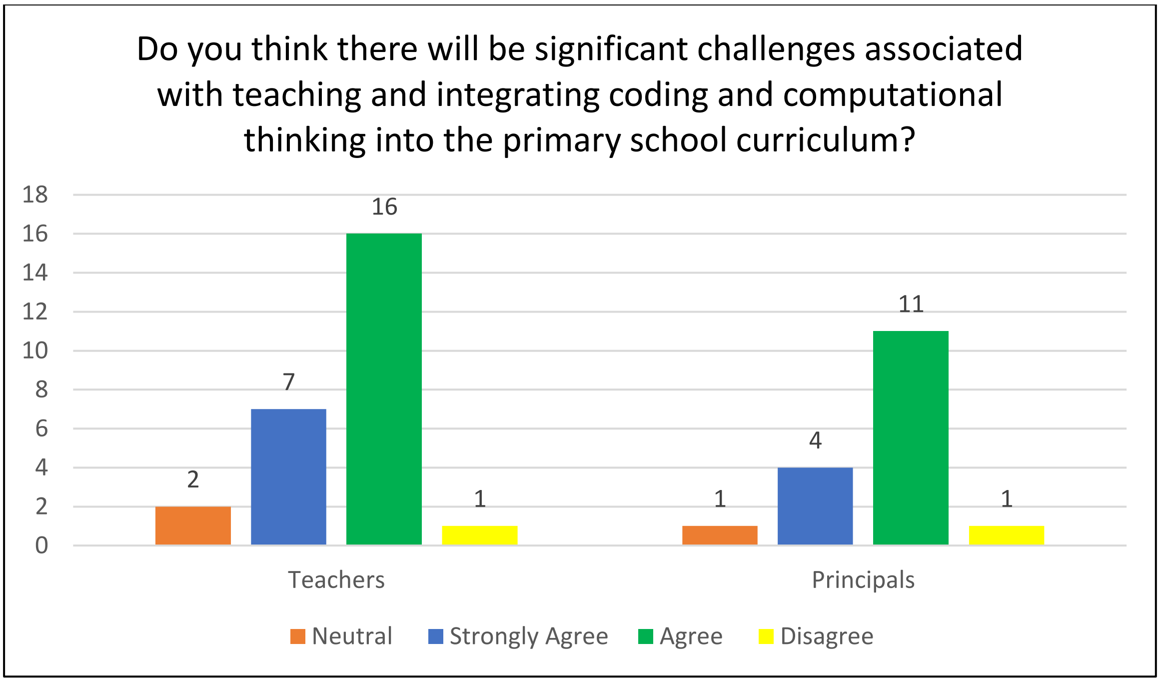
Teachers identified current curriculum overload as a major hurdle to the introduction of another subject such as coding and computational thinking.

> *I think there would definitely be resistance to another subject being added to the curriculum so it would be best to integrate with other subjects*

> *What another curricular subject! Where can I find the time?*

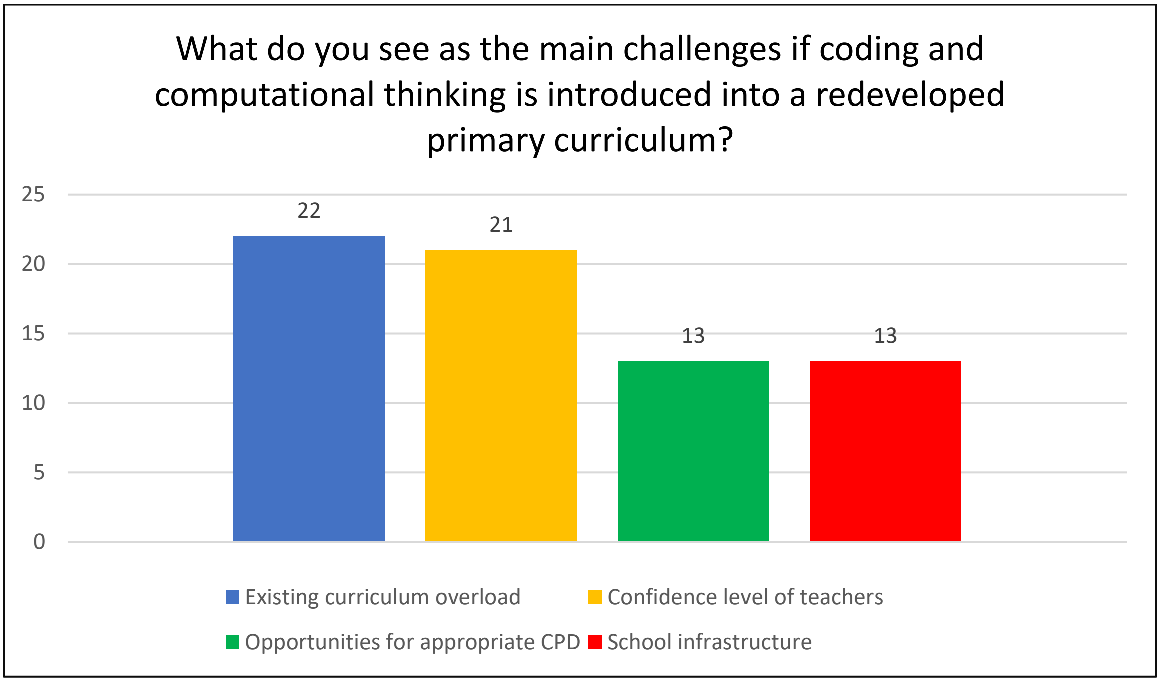
Most teachers and principals agreed (see Figures 16) that integrating coding and computational thinking and technology into a redeveloped primary school curriculum would be challenging.

Figure 16: Principal and classroom teacher perspectives on integrating coding and computational thinking into a primary curriculum



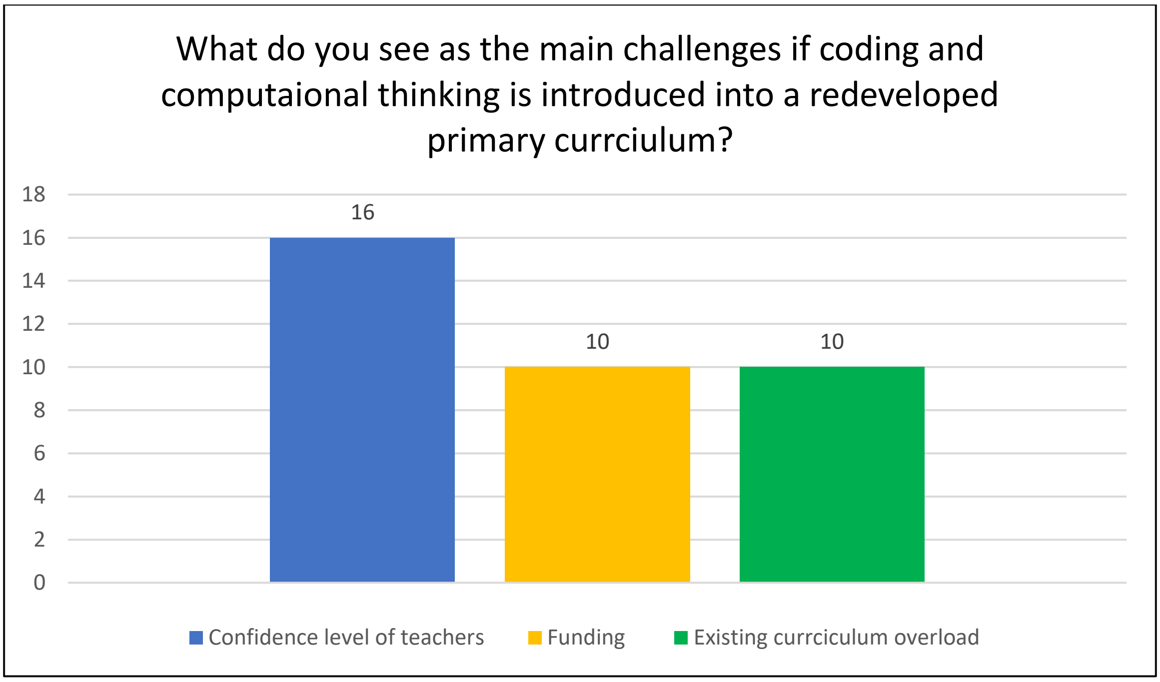
When teachers were asked what the main challenges might be, they identified curriculum overload, teacher confidence, professional development, and school infrastructure as being the most pressing issues (see Figure 17).

Figure 17: Classroom teacher reflections on the challenges of introducing coding and computational thinking



Principals agreed with classroom teachers that curriculum overload was a significant issue, but they also highlighted funding and teacher confidence as other significant challenges (see Figure 18).

Figure 18: Principals reflections on the challenges of introducing coding and computational thinking



Summary

Teachers felt that to become comfortable with the basics of coding, children should complete introductory coding and computational thinking lessons as part of a stand-alone subject. This would enable children to gain a basic understanding of the fundamentals prior to integration into other subjects. The overwhelming majority of teachers also agreed that once the fundamentals of coding and computational thinking were taught that development of those ideas should be taught through curriculum integration.

Mathematics and science were identified as being the most appropriate subjects for integration initially, but some teachers said that they integrated the concepts into other subjects such as visual arts, music, history, religion and physical education.

Most teachers thought that classroom teachers should be given the responsibility for teaching coding and computational thinking in the classroom, but some principals were less confident and were either unsure or thought it should be the job of a specialist teacher. A significant number of teachers thought that coding and computational thinking activities should be introduced from junior infants onwards. Most teachers and principals felt that learning about digital technology in school was necessary and the majority strongly agreed that using and having access to digital technology was important for children.

Most teachers and principals identified current curriculum overload, teacher confidence, professional development, funding, and school infrastructure as being major barriers to the introduction of coding and computational thinking to classrooms.

Continuing Professional Development (CPD)

The schools participated in four face-to-face workshops between May 2018 and November 2018. Apart from the workshops conducted in collaboration with the PDST Technology in Education team, schools were also offered additional CPD in the plugged and tangible devices they would use in their classrooms. These included Apple Swift playgrounds, CS first, Code Club, Micro Bits, Cubetto, KIBO, Bee-Bots and Crumble, Lego Wedo and VEX robotics.

The teachers were asked to reflect on the standard and type of professional development they had participated in during the initiative. They were also asked to respond to the following open-ended questions:

- How useful was the CPD experience you had during this project?
- What CPD do you think teachers would need if this was to be introduced into classrooms?
- Do you think teachers have the capacity to include this in their teaching?
- What do you see as the major challenges in teacher CPD in this area?

Initial professional development – two-day workshop

The consensus within the group following the initial face-to-face two-day Computational Thinking course, facilitated in collaboration with the PDST, was very positive. It must be noted that during informal discussions in the morning, before the course started, most of the teachers were very nervous and felt they would struggle.

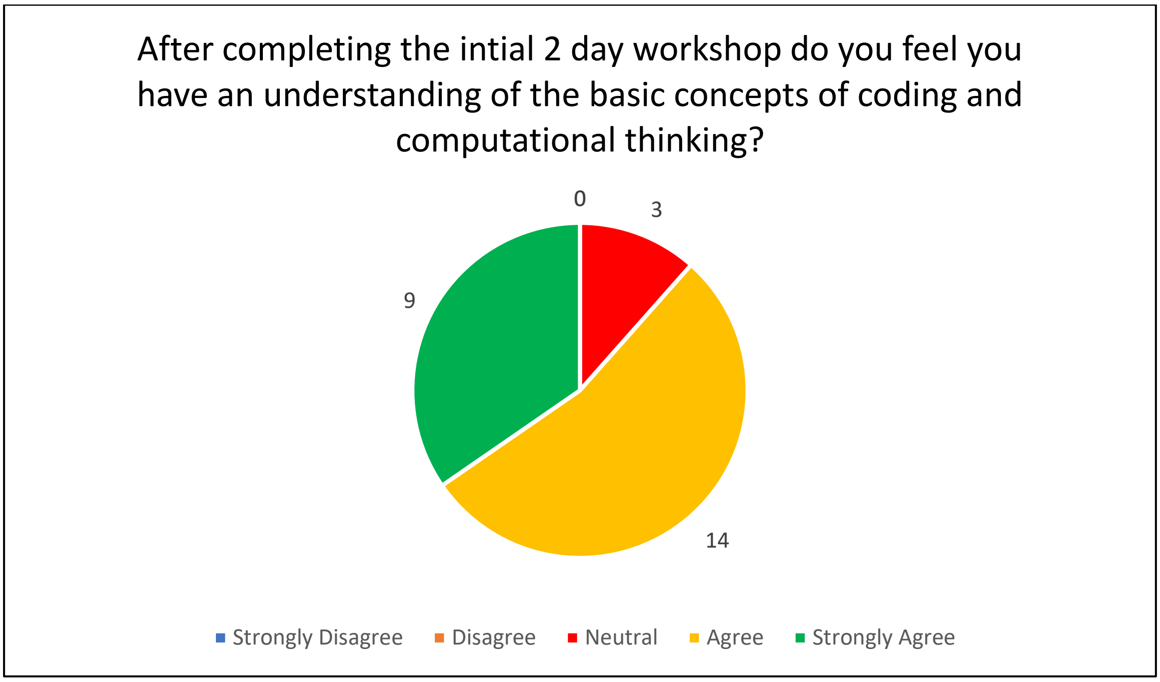
> *I felt like I had a very poor understanding of coding prior to that day and imagined that I would find the whole experience very daunting*

However, as the two-day course continued it was evident that the hands-on collaborative nature of the CPD and the examples of how the recommended resources could be used in the classroom were hugely beneficial. Participants commented on how useful it was to have the types of lessons they could do with their classes modelled by the PDST facilitators. Also, having the time to 'play' with the resources in an informal, collaborative manner and to share with colleagues how they could see the resources being used was key to building confidence within the group.

> *The hands-on experience that we received on the first two days was fantastic and gave me a great introduction to coding and resulted in me feeling much more confident*

When the teachers were asked if they felt by the end of the two-day workshop if they understood basic concepts in computing, coding and computational thinking, 23 either agreed or strongly agreed that they did (see Figure 19).

Figure 19: Teachers' perceptions of the usefulness of the face-to-face workshops



They noted that they were given a wide variety of resources to choose from, and that instructions were clear about activities they were expected to do with the children and about the timeframe for carrying out each activity.

> *I felt the PDST training we received was excellent. The sessions gave me great ideas of ways in which I could incorporate unplugged and plugged lessons into my teaching*

Ongoing professional development

To facilitate ongoing professional development within the group, an online space for collaboration using the Microsoft Teams platform was created, and further face-to-face meetings were scheduled during the initiative.

The online space worked well initially with members of the group sharing resources, website links and details about national competitions and other initiatives related to coding and computational thinking. As time went on this early enthusiasm seemed to wane; most teachers commented that they did 'keep an eye' on the space but rarely contributed. In retrospect, having a dedicated facilitator for the online space who would be responsible for updates and initiating conversations might have improved participation among the group.

The teachers were also encouraged to take part in an online course, *An Introduction to Scratch Programming,* facilitated by PDST. Initial interest in the course was positive with a total of 18 enrolling, but completion rates were lower with less than half completing it. The teachers who did complete the course found it to be excellent, commenting that it offered several good examples of how Scratch could be integrated into a variety of curriculum areas particularly numeracy and literacy.

> *The scratch training online should be compulsory for teachers who have never done any coding*

They also liked the facilitated discussion forum, finding it useful to share classroom experiences and resources. The majority who completed the online CPD thought it would be hugely beneficial for all teachers.

The teachers reported that the ongoing face-to-face meetings throughout the initiative were extremely beneficial. It gave them time to look again at resources and classroom practices used by other teachers and to share what worked and what didn’t.

> *The follow up days were great for getting feedback from other teachers about what they were doing and how they were getting on*

It was also clear that the hands-on approach of the remaining workshops was key to building confidence and enabling more effective management of classroom activities. Teachers working with very young children found it beneficial to speak with teachers who were teaching in other schools and in classroom levels different from their own. Modelling by workshop facilitators continued to be hugely beneficial as all workshops were pitched at levels that participants could relate to.

> *The microbit training was excellent. The instructor modelling how the lessons could be implemented enabled me to plan and deliver lessons with my class*

Some commented that it would have been a good idea to include visits to the classrooms of the teachers from Phase 1 to gain further insight into good classroom practice. A small number of teachers remarked that the workshops were very good but perhaps not long enough.

Recommendations

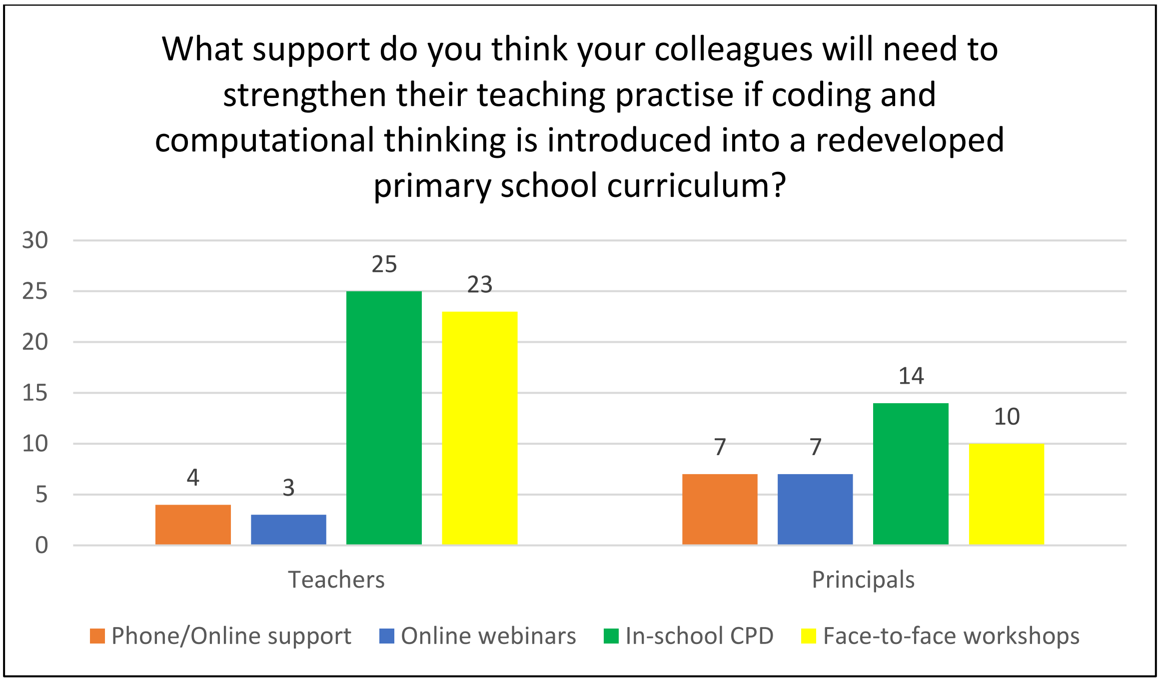
When asked what types of continuing professional development they felt would be most appropriate for primary teachers wishing to become familiar with coding and computational thinking, respondents prioritised:

- practical, hands-on professional development
- time to engage informally with fellow practitioners
- specific guidance and support regarding age-appropriate resources
- ongoing upskilling and time to reflect on best practice.

> *I believe that hands-on training where they can experience coding in a variety of forms is key to allowing teachers to develop their own confidence in this area*

Most teachers, and over half of principals surveyed, identified in-school professional development and face-to-face workshops the most important supports for teaching practice (see Figure 20) were coding and computational thinking to be introduced into a redeveloped primary curriculum.

Figure 20: Principal and classroom teacher perspective on the types of supports needed for schools



Most participants agreed that teachers have the capability to implement coding and computational thinking in their classrooms. However, this was dependent on several factors:

- teacher willingness and confidence
- sustained professional development
- time to implement.

> *I think there should be an area of continued support put in place that teachers have the ability to liaise with other professionals or experts that can alleviate their fears or concerns*

It was suggested that teachers would need to be given time and a variety of professional development strategies to introduce coding and computational thinking into their classrooms. It was also felt that teachers would need to be given an opportunity to experiment in their classrooms informally and reflect with peers on what did and did not work.

Teachers also indicated that continuing professional development would be vital as to ask teachers to teach something completely new would be unfair without appropriate CPD and support. It was suggested that school support for planning and implementation should follow a two-year phasing-in of coding and computational thinking. This would enable teachers and schools to identify the types of CPD and resources they might need at local level.

Some felt that the available range of IT courses could be a cause of some difficulty as there were so many on offer. They recommended that there should be one course which was seen to be the foundational course for all teachers.

Challenges

Teachers and principals from the participating schools highlighted the following as being key challenges for teachers:

- teacher mindset
- lack of knowledge
- ongoing professional support.

In the early stages of the initiative, teachers reported that other colleagues in their schools were apprehensive towards coding and computational thinking. To some degree the following statement summed up the general feeling.

> *In discussing the Coding Initiative with my colleagues at school, I did find there was a certain resistance among many teachers to become involved in teaching code. They seemed to have little interest in the topic or were intimidated at their perceived lack of expertise in the area. I feel that a lot of hands-on training will be needed*

Practical questions arose as to how the professional development programmes could be implemented. Would they be part of Croke Park hours at a whole school level and would full participation by all staff be necessary? Would teachers be expected to attend CPD in their own time? Would there be enough funding to support this CPD?

There was broad agreement that a phased approach, for example, starting with CPD for the junior class teachers only, was not a good idea and a whole school approach would be more beneficial.

Summary

The teachers felt the hands-on, collaborative nature of all the face-to-face workshops was excellent and enabled them to see how the resources and methodologies could be used. They also commented that being given the time to 'play' with the resources in an informal, collaborative manner was hugely beneficial. Time to meet, discuss and revisit strategies with colleagues was also key to the success of the professional development process.

The online space worked well initially, teachers used the links to resources and to other national competitions, and they shared resources that worked well in their classrooms. Nonetheless, over time the early enthusiasm waned. This was also evident in the participation rates of the Scratch online training programme facilitated by the PDST. Initial numbers of participants were high, but completion rates were comparatively low.

Teachers and principals identified several challenges to the practical implementation of a CPD programme for teachers in coding and computational thinking. They highlighted the need for whole staff participation and queried how the CPD could be facilitated at a national level. They also highlighted the need for schools to be given time to implement curriculum change in relation to coding and computational thinking.

Resources

Participating teachers were introduced to a range of unplugged, plugged, and tangible device resources they could use with the children. Several recommendations came from the classroom teachers who participated in Phase 1 and from members of the PDST Technology Team. Further recommendations came from findings from the NCCA's Audit and Curriculum Investigation.

It was important to recommend resources that teachers and schools would have ready access to and so cognisance was taken of

- schools' access to the internet
- types of digital devices currently used in each classroom

- the ages of the children
- funds available to purchase new equipment
- access to recommended resources through local Education Centres.

Teachers were not limited to the types of resources recommended; they were actively encouraged to investigate other resources that would be useful in their individual classrooms.

Each school participating in Phase 2 was given a grant of €1,000 from the NCCA to offset costs associated with involvement in the initiative. Purchasing resources was one such cost. The teachers were asked to identify the specific needs of their own school and to use the grant accordingly.

Later, the teachers were asked to reflect on the types of resources they used during their work on coding and computational thinking. They responded to the following open-ended questions:

- What resources did you use and why?
- Were the recommended resources, in your opinion, age appropriate?
- Where did you source resources and was funding an issue?
- Did you encounter any difficulties with the resources you used?
- Are you planning to acquire any further resources?

Resources used

Teachers in Phase 2 were given opportunities to see how resources could be used in their schools. To this end, hands-on workshops were conducted by members of the PDST Technology team, representatives of the Raspberry Pi Foundation, and Apple.

During the face-to-face sessions, teachers had opportunities to 'play' with a selection of tangible devices and investigate resources linked to unplugged and plugged activities. Teachers who had participated in Phase 1 of the initiative were also invited to attend and to offer guidance or advice.

The Phase 2 teachers then decided on the most appropriate unplugged and plugged resources for their classes. Because of the wide range of class levels (junior infants to 6th class) and settings (for example, SEN), and the lack of familiarity with the tangible devices, a semi-structured approach was taken regarding which devices would be used and at what classroom level including:

- junior, senior infants and 1st class – Bee-Bots
- 2nd and 3rd class - Lego WeDo 2.0
- 3rd and 4th class – Crumble device
- 5th and 6th class - Microbits.

For teachers working in multi-grade classrooms a combination of devices were used. Three schools participated in a national competition using the VEX Robotics kit. Junior class teachers were also offered the age-appropriate KIBO and CUBETTO devices. However, through a hands-on workshop investigating the potential of both, it was decided, given their current level of confidence and expertise, that the Bee-Bot would be the most appropriate device to use.

Many of the teachers said that they tried out or used most of the recommended resources. A small number of teachers had looked at, and used, other coding/computational thinking resources such as Tynker, Sphero, Lightbot Hour, Daisy the Dinosaur, Hopscotch and Makey-Makey. A small number had also used a range of other non-coding/computational thinking resources, such as iZak9, Book Creator, iMovie, and Seesaw.

A few teachers decided to use Code.org, a non-profit organisation which has a website focused on making computer programming accessible. They found the resources, both unplugged and plugged, to be user-friendly and the site easy to navigate. Others commented that the lessons were interesting, easy to follow, sequential and age-appropriate. The Bebras Challenge and the Barefoot Computing resource were also highly recommended.

The tangible devices recommended and used during the study received positive feedback, with all teachers commenting that the devices were very engaging and age-appropriate. The Bee-Bots and Lego Wedo were the devices most often purchased or borrowed from Education Centres, while the Microbit and the Crumble also received very positive feedback.

Sourcing resources

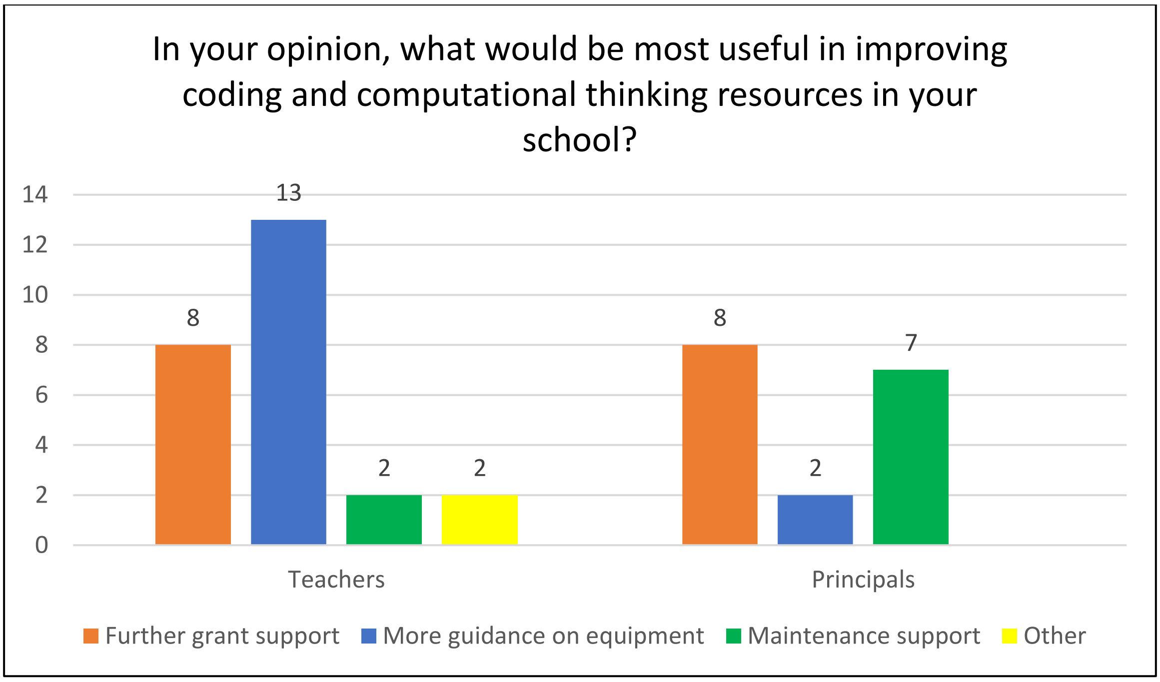
Several of the schools accessed the tangible devices such as Bee-Bots and Lego Wedo through their local Education Centre. Others used allocated funds to purchase a small number of devices to test before deciding to order larger quantities. At the time of reporting, a small number of the schools had not made significant purchases and were waiting to meet colleagues from the initiative to get advice and recommendations. Others commented that they would still like more advice in general and were thinking about practical solutions for wi-fi and hardware within their schools.

Challenges

Although several schools were able to access resources such as Bee-Bots and Lego Wedo from their local Education Centres, some schools did encounter difficulties. The resources seemed to be very popular and were unavailable, and when they did arrive, pieces were missing. Others noted that their local Education Centres were quite a distance from the school and getting time to travel and pick up resources was a problem.

Teachers also thought that more guidance on types of equipment/resources to purchase and further grant support for the purchase of new equipment would be needed. Likewise, principals highlighted the need for further grant and maintenance support for their schools (see Figure 21).

Figure 21: Principal and classroom teacher perspective on the types of supports needed in relation to resources



Issues at a local level seemed to vary. Some schools had access to funding while others relied on parents' associations and fundraising activities to supplement the purchasing of equipment.

We will need a lot of extra funding from the government for this to work

Access to the internet/quality of wi-fi needed to use some of the resources was also an issue for several schools.

Summary

The resources used during Phase 2 were recommended by the classroom teachers who participated in Phase 1 and members of the PDST Technology Team. Further recommendations came from findings from the NCCA's Curriculum Investigation.

Generally, comments on the suggested resources for unplugged and plugged activities were very positive. The teachers remarked on how user-friendly, interesting, sequential and age-appropriate most of the resources were.

Selection of the tangible/physical computing resources and devices was quite structured. Schools received age-specific device recommendations depending on the class level or levels they were teaching. The teacher reflections on these resources were very positive; they found all the recommended tangible resources to be very engaging and age-appropriate.

Schools were given a grant by the NCCA and either purchased their own resources or borrowed from Education Centres. Teachers said that they would like further advice, recommendations and guidance on types of equipment/resources to purchase and further grant support would be needed. Some schools had access to funding while others relied on parents' associations and fundraising activities to supplement further purchases.

Parents

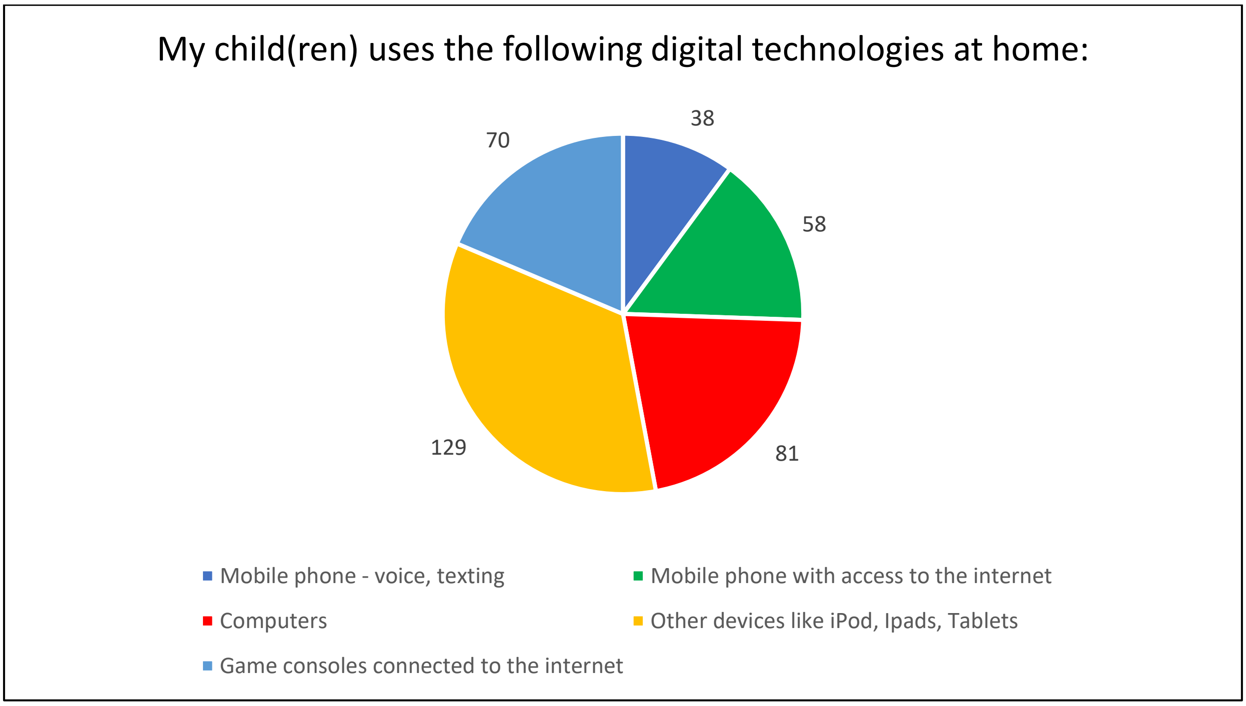
Two methods were used to gather the views of the parents of the children attending the schools in Phase 2 of the initiative—an online survey and focus group discussions. There were 146 responses to the online survey, and the focus group sessions were held with parents in two of the schools – a small rural school and a large urban DEIS school. The findings below combine both data sources whereby the survey results are complemented by feedback received from the focus group discussions. The findings are presented in three sub-sections.

Digital technology in the home

The first section explores parents' views on the availability, and children's use, of technology in the home environment.

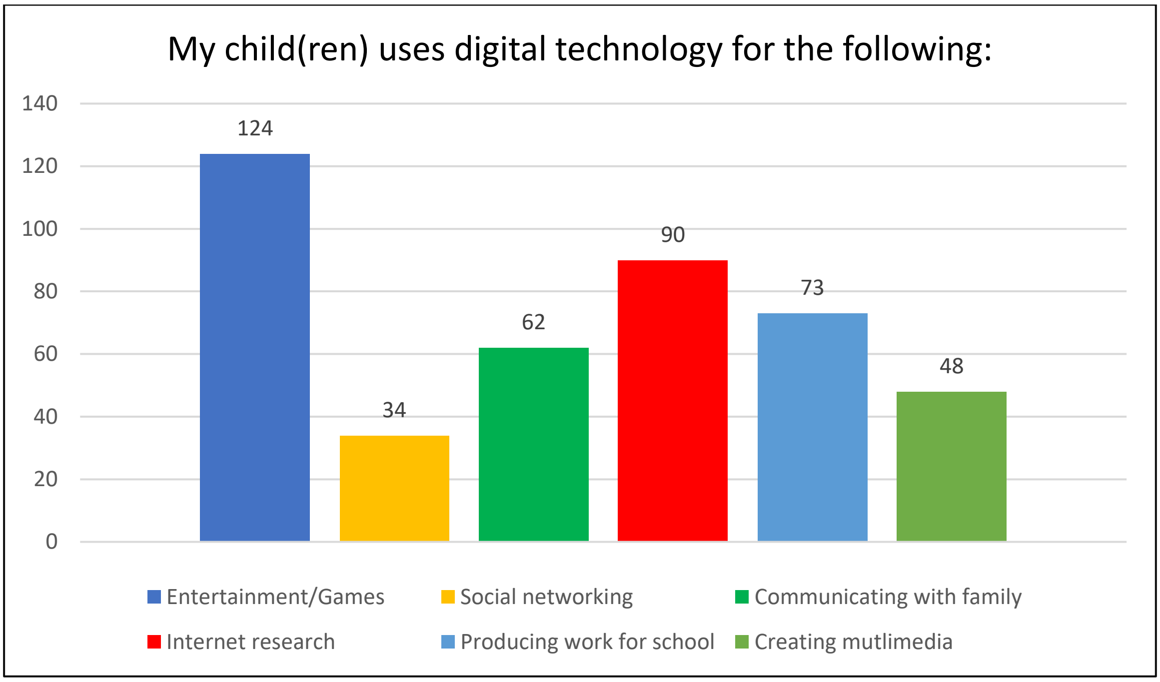
Ninety-nine per cent of respondents said their children had access to devices in the home. Alongside this, 100% signified that they had internet access available in their home.

Figure 22: Home access to devices and the internet



When asked to specify what types of digital technology their children used at home (see Figure 22), 88% of parents identified 'Other devices like iPod, iPad, Tablets or Kindles'. This was followed by computers (55%), games consoles connected to the internet (48%), mobile phones with access to the internet (40%) and finally mobile phone with voice and texting functions (26%).

Figure 23: Children's use of technology in the home



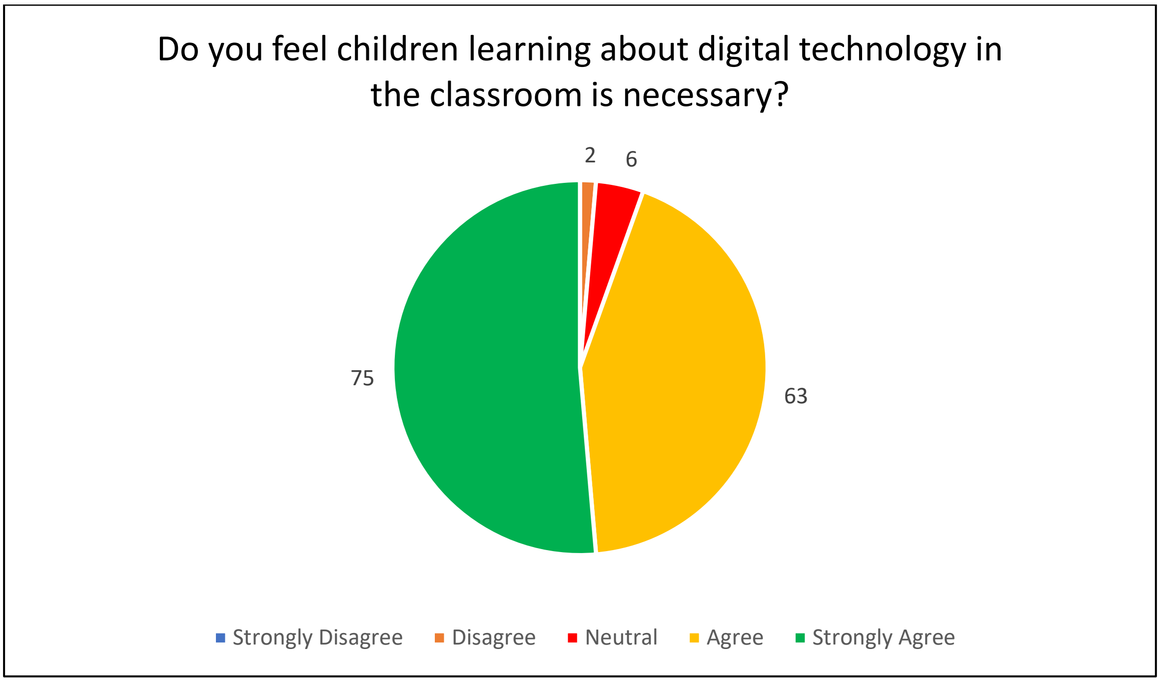
Parents were asked what their children used technology for. Again, they could select more than one answer where appropriate (see Figure 23). The most common option selected (85%) was 'Entertainment/games'. This was followed by 'Internet research' (62%), 'Producing work for school' (50%), 'Communicating with family and friends' (42%), 'Producing multimedia for personal use' (33%) and 'Social networking (Facebook, Snapchat etc.)' (23%).

During the focus group discussions parents aired concerns about the amount of screen time their children had access to at home. Some parents said they were worried about the apparently addictive nature of digital games their children were playing. Consequently, they called for a balance in any future proposals on the use of screen time in schools.

Digital technology in school

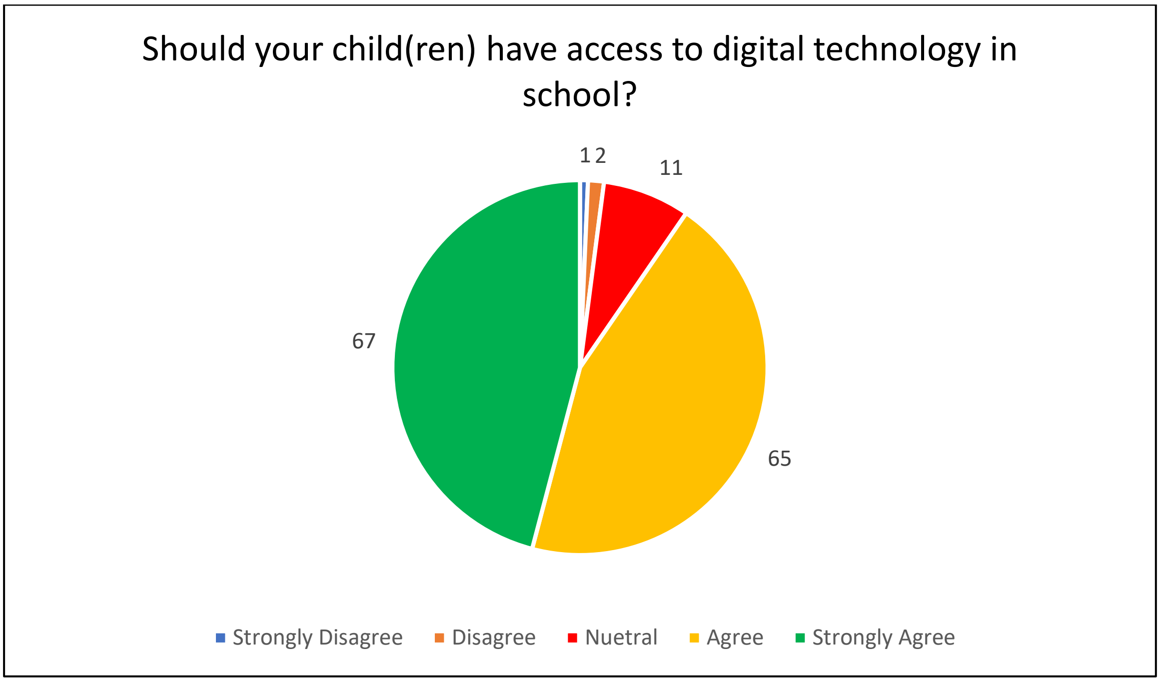
The second section looks at parents' perspectives on digital technology in schools.

Figure 24: Digital technology in the classroom



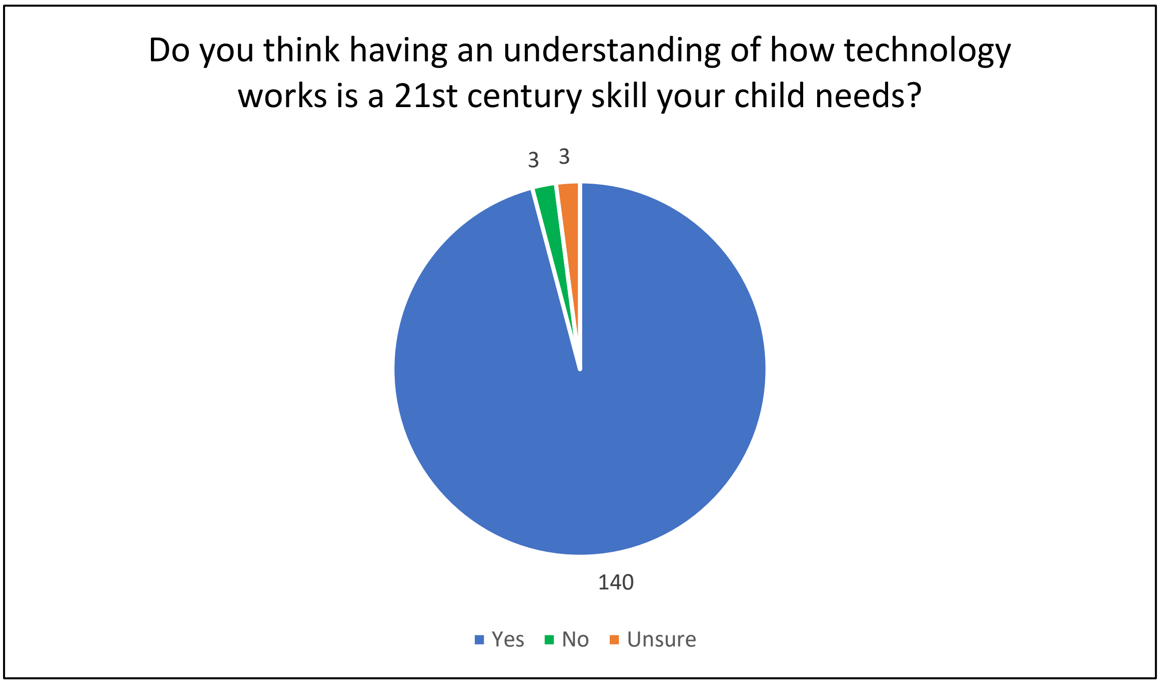
Responding to the online survey, 95% of parents either strongly agreed or agreed with the need for 'learning about digital technology' in the classroom, with 1% disagreeing and 4% undecided (see Figure 24). This was also reflected in the focus group discussions where there was strong consensus that children should be learning about technology in primary schools. One parent shared: *We are surrounded by technology in our lives today, in our homes, supermarkets, libraries. Children need to have the ability to understand it, use it and be ready for a future where technology will become even more prevalent*. Parents also recognised the already busy curriculum with some parents calling for technology to be integrated into what already is taking place in schools. The view of one parent was: *It should be naturally integrated, as in the future workplace, technology will be one part of the puzzle. People will still have to utilise other skills and information outside of what technology can do.*

Figure 25: Access to technology in school



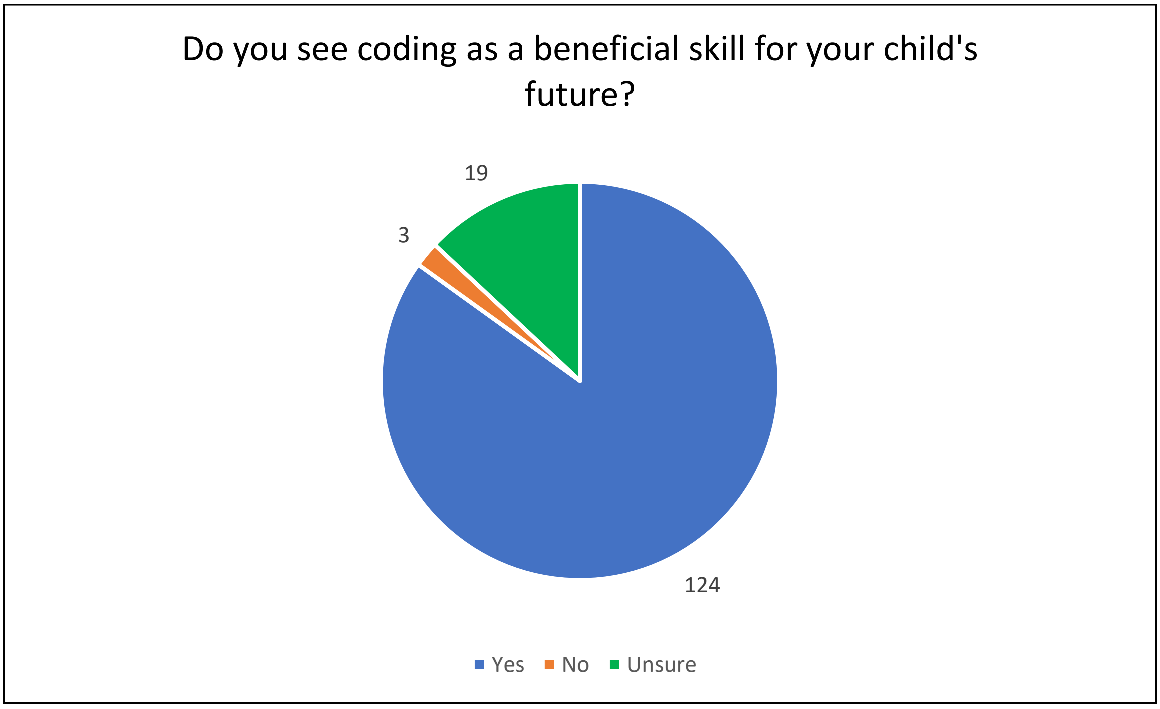
Collectively, 90% of parents surveyed agreed that their children should have access to technology in school (see Figure 25). Some 8% of respondents were neutral on the topic while 2% disagreed. During the focus group discussions, the view was expressed that technology should not just be viewed as phones, games or television which are all consumer-based. Instead the focus in schools should be on productive-based technologies, such as Microbits, iPads and Lego Wedo, which engage children’s critical thinking and encourage the use of creative skills.

Figure 26: Digital technology as a 21st century skill



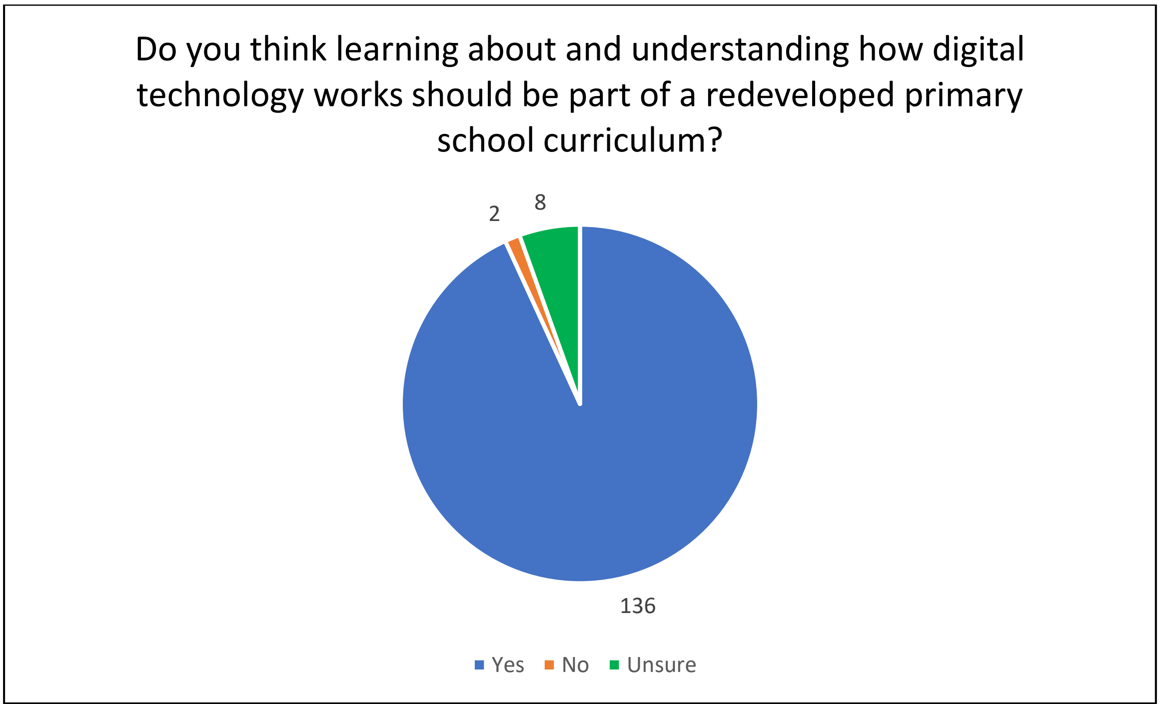
When asked if understanding how digital technology works is an essential 21st century skill for their children, 96% of parents agreed with 2% disagreeing and 2% unsure (see Figure 26).

Figure 27: Coding as a beneficial skill



In response to the question on whether or not they saw coding as a beneficial skill for their child's future, 85% agreed that it was while 2% disagreed and 13% were unsure (see Figure 27). During the focus group discussions, there was consensus that learning about digital technology was important to meet the demand of future employment trends. *It gives them a head-start in life, and it is important for future jobs.*

Figure 28: Digital technology as part of a redeveloped curriculum



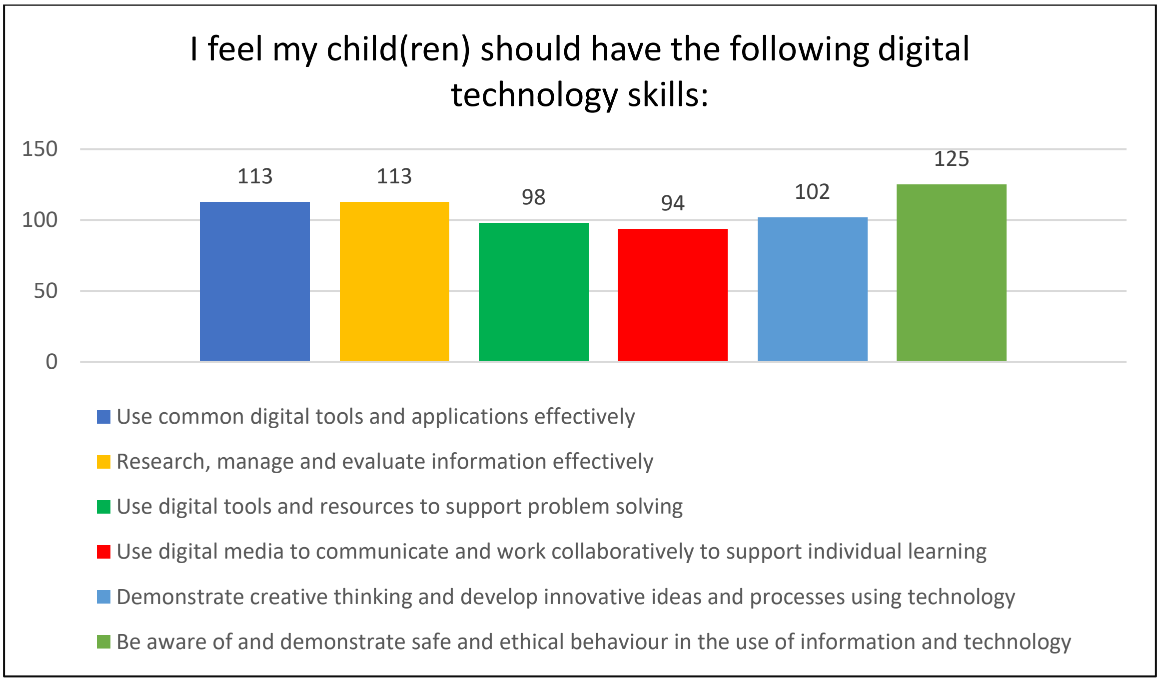
When asked whether or not digital technology should be part of a redeveloped primary school curriculum, 93% agreed that it should be, 1% disagreed and 6% were unsure (see Figure 28). During the face-to-face discussion, parents noted the significance of changes at post-primary level with the introduction of Computer Science at Leaving Certificate and the coding short courses in junior cycle: *Children are going to be exposed to these subjects, so they need to have some basis in working with technology in primary school.*

The focus group discussions also reflected concerns some parents had about the challenges facing schools regarding the funding and maintenance of technology equipment. Parents were of the view that *ad hoc* fundraising is a limited approach and that significant central funding would be required to allow schools to purchase and maintain digital technology.

What should children learn?

The final section presents feedback from parents on what children should learn about digital technology in school.

Figure 29: Children's digital competence



Asked which technology skills they would like to see their children develop (see Figure 29), 86% of parents chose 'Be aware of and demonstrate safe and ethical behaviour in the use of information and technology'. 'Use common digital tools and applications effectively' and 'Research, manage and evaluated information effectively' were chosen equally by 77%. Some 70% of those surveyed identified 'Demonstrate creative thinking and develop innovative ideas' as important, while 67% and 64% identified 'Use digital tools and resources to support problem solving' and 'Use digitally media to communicate and work collaboratively to support learning'. These choices mirror discussions from focus groups where safety was among the most frequently raised issues. Parents were adamant that children needed to be taught about the possible dangers that come with technology, *Technology and the internet has a lot of dangers associated with it and children need to be aware of that.* Parents praised the effectiveness of the *Stay Safe* and healthy eating programmes in schools and called for a similar focus on the use of technology. One parent called for a 'healthy screen time' programme which would contain guidelines as to what was and was not appropriate in regard to using technology, *Children need to be taught how much time on the iPad or in front of the TV is too much.*

Critical-thinking, creativity and problem solving also featured in their priorities. One parent's view was that, *Technology provides so many opportunities for children to think critically about* *things,* while another felt that *it gives children a chance to see there is more than one possible outcome; it allows them be creative*. Parents also highlighted a preference for children learning about the functional operations of digital technology, with several suggesting that children should be explicitly taught how to communicate through e-mail, to touch-type, and to create presentations and spreadsheets.

Finally, during the face-to-face discussions, parents offered some insights into their children's experience of participation in this initiative. They said that their children were enjoying what they were doing and were excited to relay the learning to them. One parent commented that her children *were more eager than ever to tell me what they had done*. One group of parents had been invited into the classroom to witness the work first-hand and one said that they *could sense the creativity that was happening in the classroom*. Another noted that *it was different to traditional learning*. *The children were learning without them even realising it.*

Summary

Most parents said their children had access to technology at home and all had access to the internet. The children predominantly used technology and the internet at home for entertainment purposes, searching the internet for research for school work, communicating with family and friends, producing their own multimedia, and social networking.

Parents were concerned about the amount of time their children were spending on screens at home and called for balance in any future proposal on the use of screens in school. They also worried about the addictive nature of digital games and the effect this might be having on their children. Online safety and ethical behaviour were among the biggest issues raised during focus group discussions.

A number of parents cited the changes at post-primary level, particularly the developments in the junior cycle short course in coding and the introduction of computer science at Leaving Certificate, the demands for future employment, and the invasive nature of technology in everyday life as being valid reasons for the introduction of coding and computational thinking into the primary curriculum.

Parents also commented on the need for their children to have access to technology in school and for technology to be integrated across the curriculum. They hoped that a greater focus would be placed on productive-based technology, thereby enabling their children to think more critically and creatively.

Children

The initiative afforded children in the participating schools the chance to voice their experiences of learning about coding and to share their views on the use of digital devices in school. This piece of the research aimed to collect, analyse and represent children's voices on these learning experiences. Taking account of the perspectives of children on matters which affect them enables them to participate as active citizens in research projects and ensures their voice is heard. James (2007) notes that 'voice' represents children as participants rather than objects in the research process.

The findings outlined here are presented in two sections—children's survey and children's artefacts—and have been compiled following a systematic analysis of data collected.

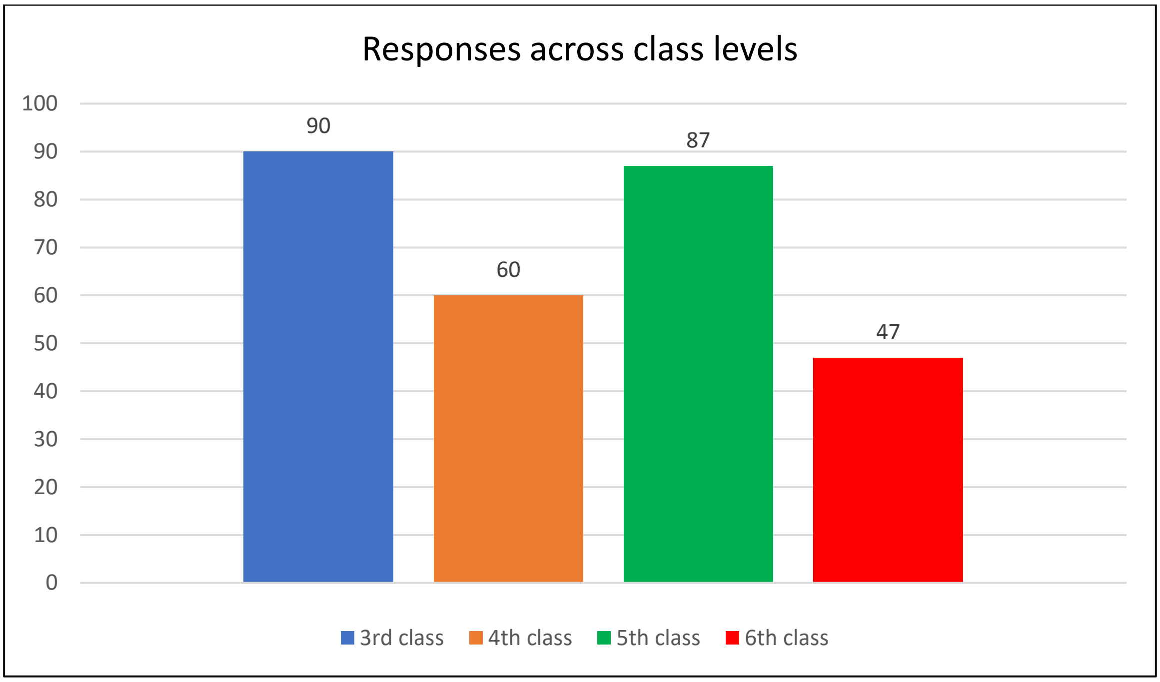
Children's survey

The children's survey was carried out through the *Kahoot* platform and contained short, closed questions. The survey, completed by 284 children, was recommended for use with children from 3rd – 6th class but it was up to the discretion of the teacher to decide if it was appropriate for their class(es).

Two initial identifier questions revealed that 133 females and 151 males completed the survey. In the interests of preserving children's anonymity, no other identifier data was gathered.

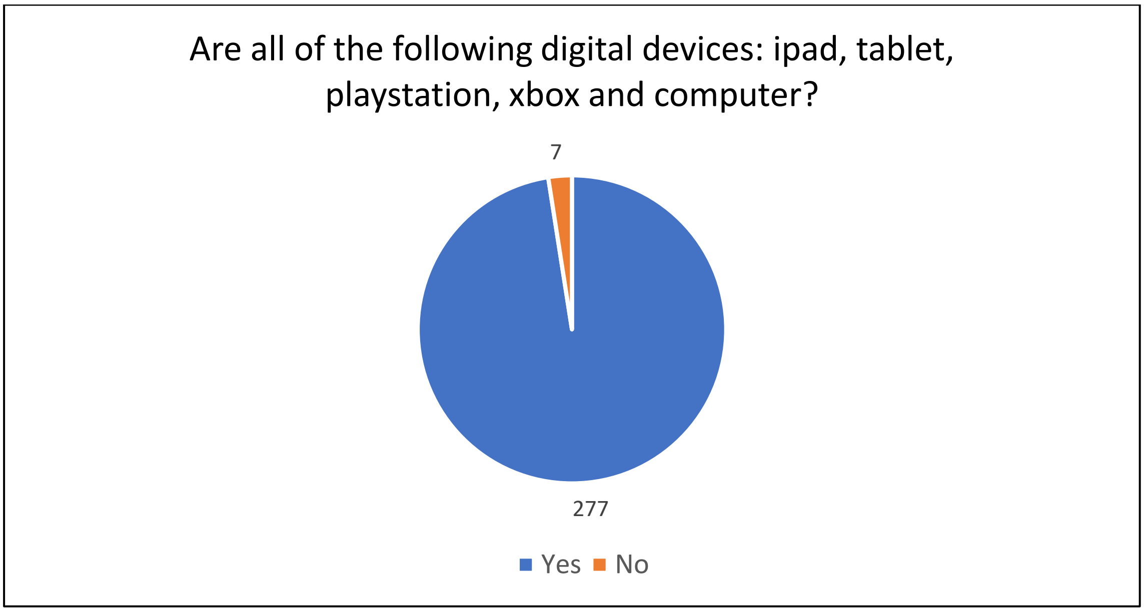
There were ninety respondents from 3rd class, sixty from 4th class, eighty-seven from 5th class and forty-seven from 6th class (see Figure 30).

Figure 30: Respondents according to class level



Introductory questions asked children about digital devices. These questions functioned as a safe introduction to the survey to allow them to get used to the answering platform as well as providing clarity as to what a digital device is. Figure 31 shows the results obtained from one example of the introductory questions.

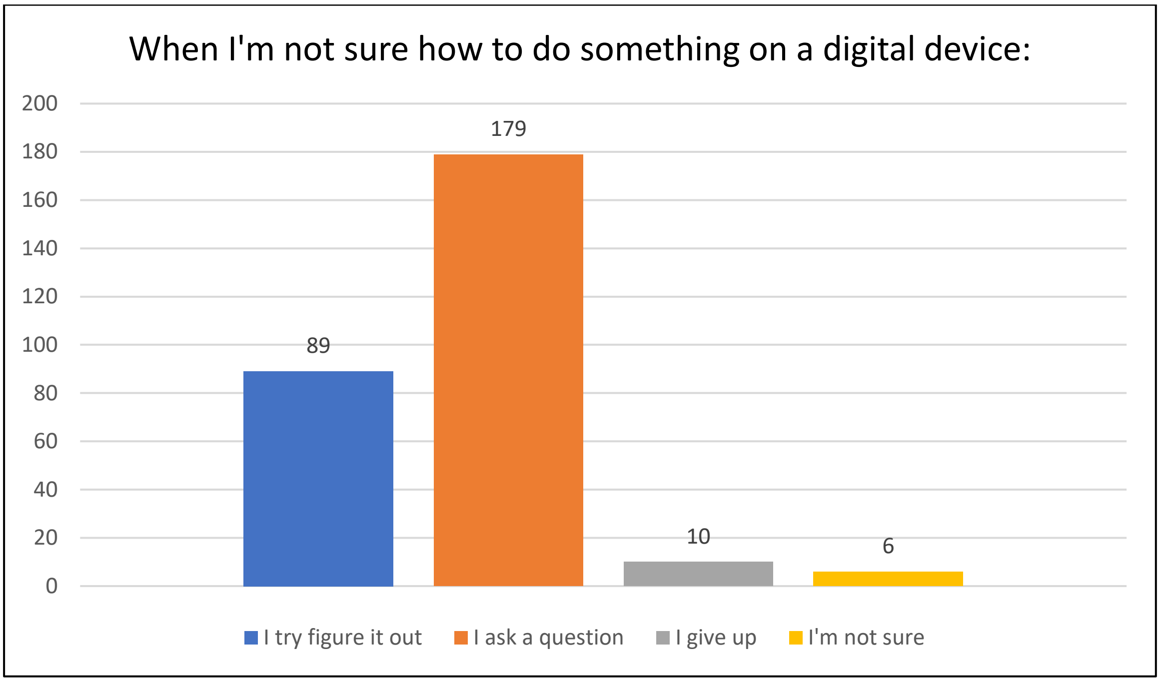
Figure 31: Sample introductory question about digital devices



As can be seen from this example, the children showed good awareness of what is meant by a digital device, with 98% of children achieving a correct answer.

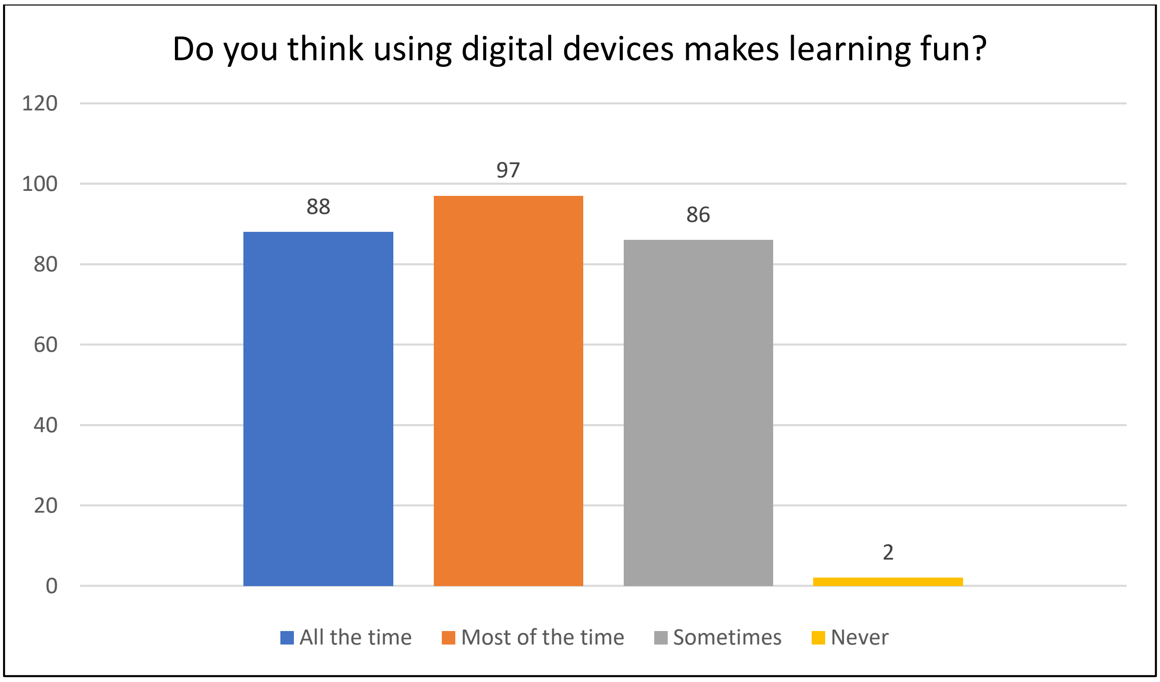
When asked what they would do when they did not understand how to do something on a digital device, the children showed a strong desire to continue to pursue the task (see Figure 32). Encouragingly, 31% and 63% of respondents answered that they would either try to figure it out or ask a question. Under 6% of respondents reported that they would give up or were not sure.

Figure 32: Children's strategies when unsure about devices



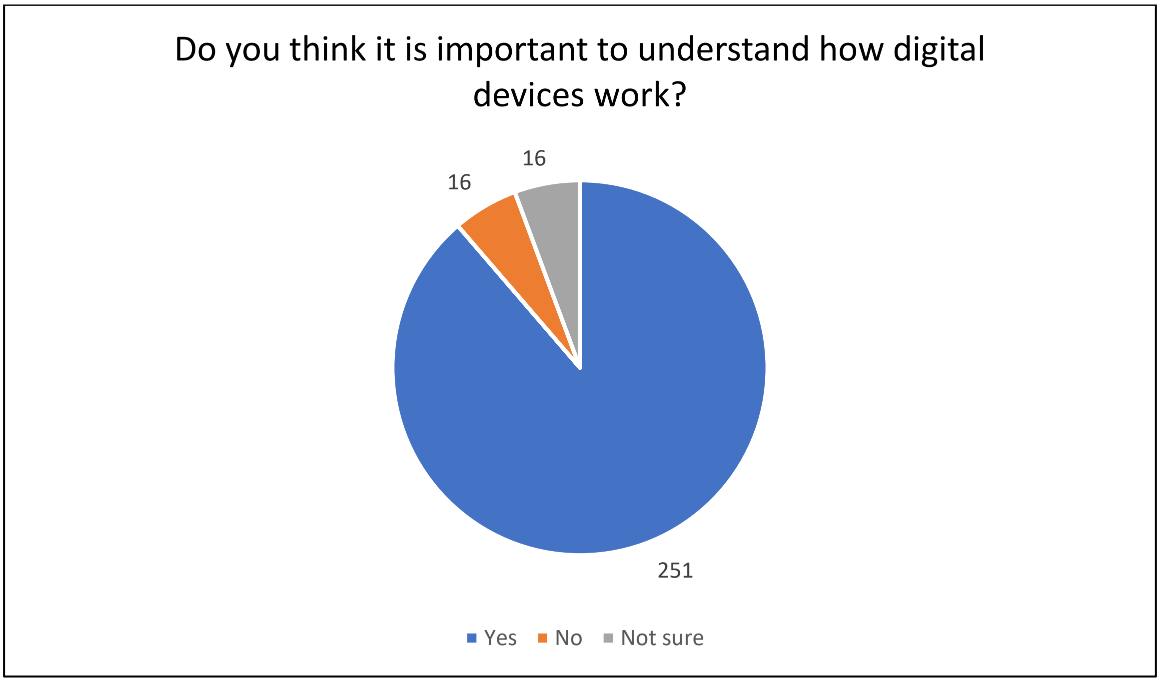
When asked if they thought digital devices make learning fun, 65% of children responded with 'All the time' or 'Most of the time'. 34% of respondents claimed that they 'sometimes' made learning fun, while 1% responded with 'Never' (see Figure 33).

Figure 33: Making learning fun through digital devices



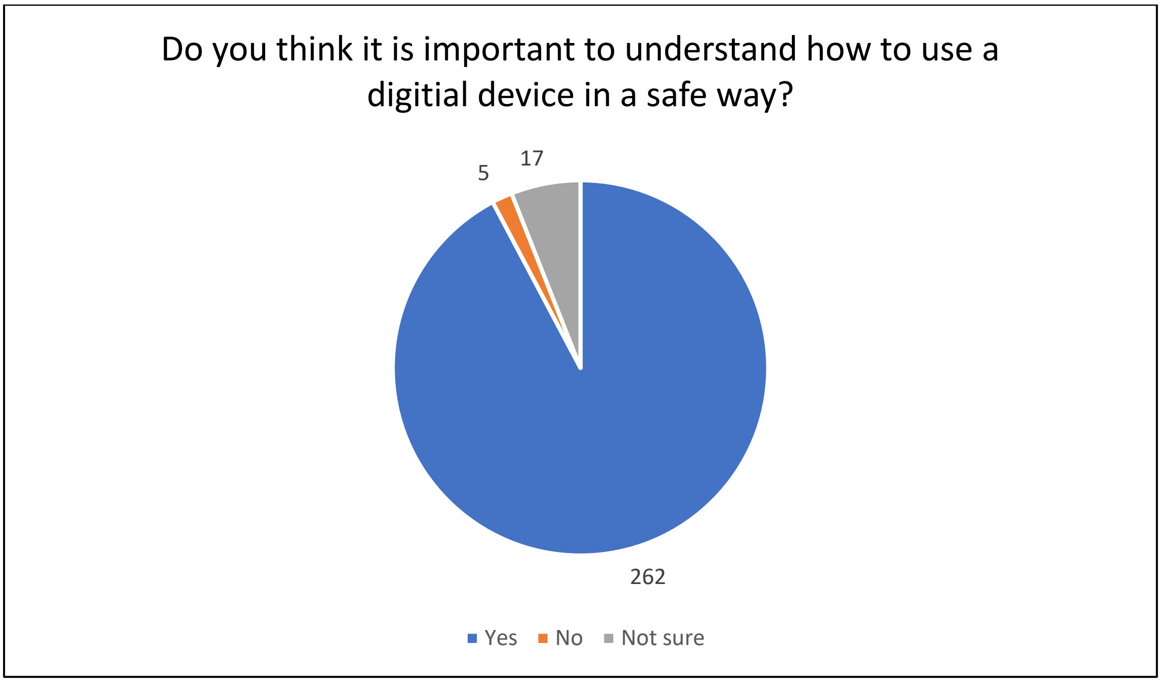
Responding to a question relating to understanding how digital devices work, 88% of children agreed that it was important to understand, while 6% felt it was not, and a further 6% were unsure (see Figure 34).

Figure 34: Importance of children understanding how digital devices work



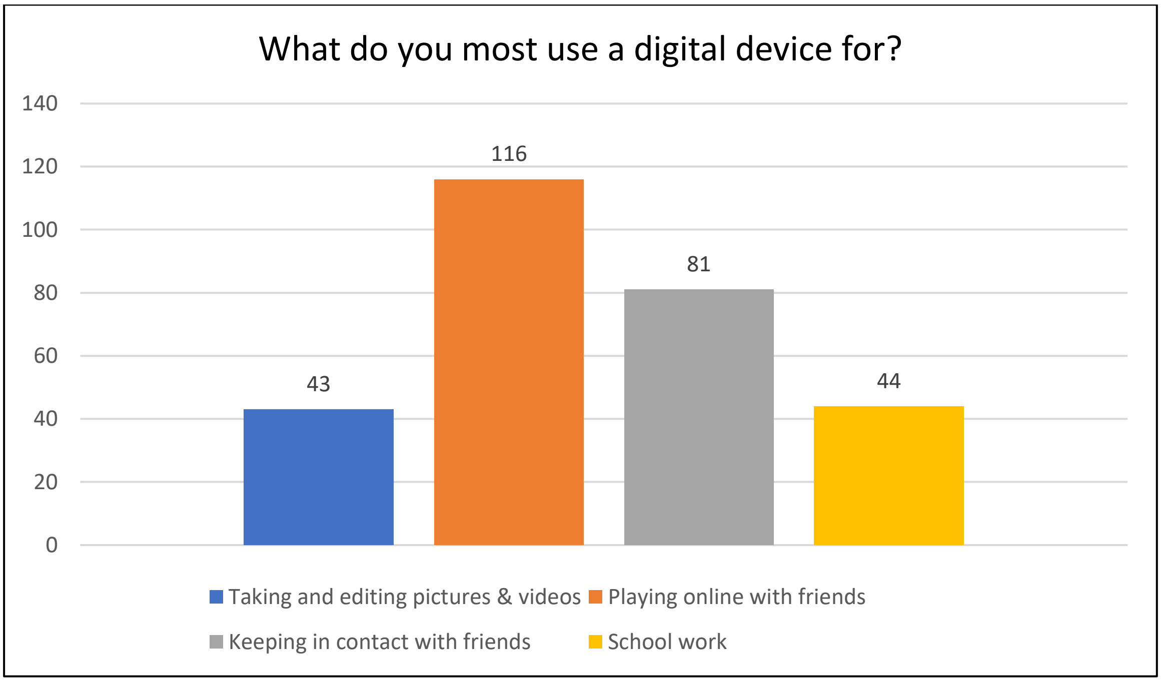
In a similar vein, when asked whether it was important to know how to use a digital device in a safe way, the responses were overwhelmingly positive. 92% of children agreed that it was important, with 6% stating it was not and 2% not sure (see Figure 35).

Figure 35: Importance of children understanding how to use digital devices safely



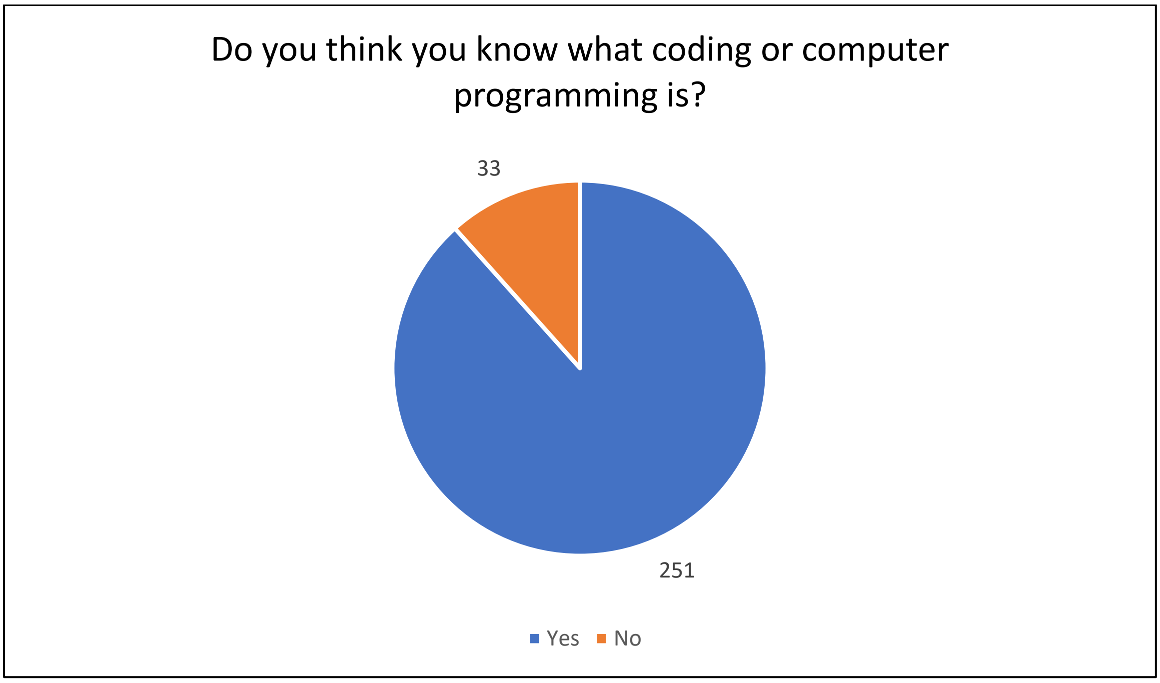
When asked how they interacted with digital devices, 41% of children chose ‘Playing online with friends’ as their most frequent reason for using digital devices. This was followed by 29% of respondents selecting ‘Keeping in contact with friends.’ Finally, 15% said they used digital devices mostly to take and edit photographs and videos, while an equal number said they used them to complete schoolwork (see Figure 36).

Figure 36: Purposes for which children use digital devices



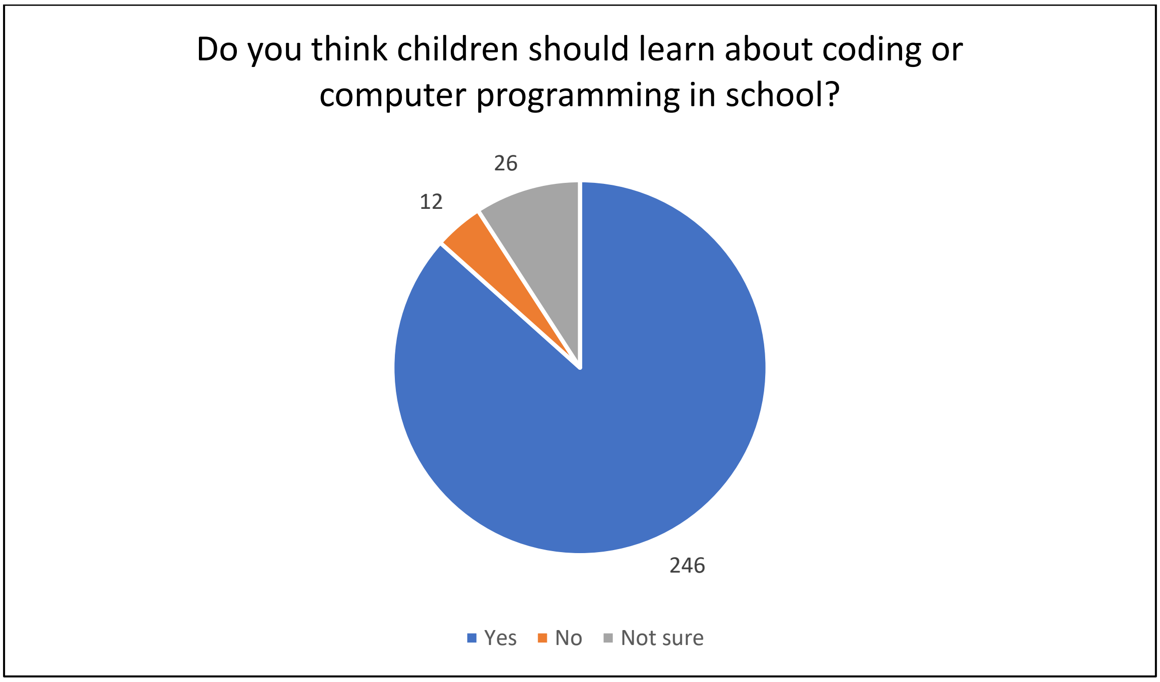
In the last section, children were asked two questions relating to coding. The first question asked children if they understood what coding or programming was. The vast majority of the children (88%) answered in the affirmative, with 12% stating they did not think they understood what was meant by the terms (see Figure 37).

Figure 37: Children's self-reporting of their understanding of coding or computer programming



Finally, the children were asked if they felt they should learn about coding or computer programming in school. 87% of children agreed with the statement, 4% disagreed while 9% were not sure (see Figure 38).

Figure 38: Children's perspectives on the importance of learning about coding or computer programming in school



Children's artefacts

A thematic data analysis of children's pictures and images has brought to light understandings of and reflections on the learning experiences the children engaged in during the initiative. Almost all the thirty-three artefacts feature children in the younger classes (junior infants – 2nd class), with a small number coming from the senior classes. Some key themes emerged (see Figure 39).

Figure 39: Themes identified from children's artefacts

Theme One – Collaboration

Collaboration was the first theme that emerged from the artefacts. Children described their learning experiences of working with their peers. Seán (senior infants), as recorded by his teacher, stated *I am the inputter and I am going to put in the orders to make an algorithm, so the Bee-Bot can tell the story of Little Red Riding Hood. One girl is writing the program and the other girl is going to test it and tell me what to do* (see Figure 40).

Figure 40: Image described by Seán (senior infants)

Other examples of drawings by children show them working alongside a peer or peers. Sarah (senior infants) described how she and a fellow classmate worked together to get the bee-bot to the letter 'i' (see Figure 41).

Figure 41: Sarah's drawing

Theme Two – Fun and Enjoyment

Children shared great enthusiasm for working with the various devices. The words 'fun', 'play' and 'enjoy' were frequent across the feedback gathered. Susan (junior infant) said *I love playing with the Bee-Bot* (see Figure 42).

Figure 42: Drawing by Susan

When asked what he didn't like about the lesson, Alex (senior infants) said that he 'liked it all'. Tom (senior infants) enjoyed working independently, *I liked how we got to build the maze ourselves,* and also enjoyed *playing with the Bee-Bot* (see Figure 43).

Figure 43: Tom's drawing

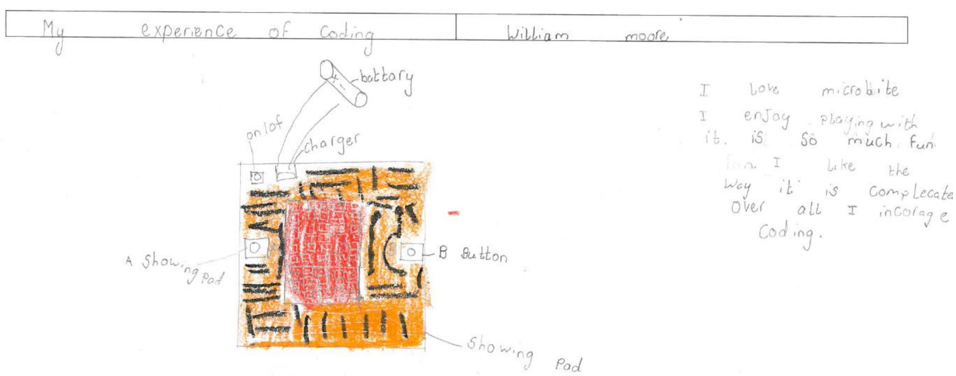
Theme Three – Challenging

Feedback from children revealed the challenging dimension of the work or tasks they were completing. Tara (1st class) said that she didn't like it when the Bee-Bot went to the wrong place and they had to start all over again. Ben (senior infants) said: *I didn't like when it kept crashing into the walls because we had the wrong moves* (see Figure 44).

Figure 44: Image described by Ben

Seán (senior infants) described the challenge very well: *I don't think there's anything I don't like but I got angry when the bot wasn't doing the orders right because I had to start again to fix the program*. The challenging aspect of the work was welcomed by some children such as Kevin (5th class) who enjoyed the fact that it was complicated (see Figure 45).

Figure 45: Kevin's drawing



One senior infant (Cara) was able to recite all the directional moves her Bee-Bot had just made (see Figure 46). *My Bee-Bot is going down the lollipop road. He has to forward 5 times, then turn left and forward 2 more times. Then he gets to the shop*.

Figure 46: Cara's drawing

Summary

Most children indicated that they have access to technology at home and that they mainly used technology to play games and interact with friends online. Some used it for completing school work or creating multimedia artefacts.

The children showed great awareness of what a digital device could do, and most were confident of solving independently any problems they might encounter when using technology. The children felt that understanding how digital technology works and how to use it effectively was important to them.

They felt they had a good understanding of what coding was and the majority thought it should be taught in schools. They almost all agreed that learning how to use a digital device in a safe way was very important.

A feeling of collaboration with peers emerged strongly from their comments, as did the sense that working with devices and completing activities or projects was challenging. Their perspective was that collaborating with their friends was fun, although sometimes hard, and that they enjoyed the challenge of the projects or tasks they were asked to do.

Chapter 4: Discussion

As outlined previously, the aim of the school-based initiative was primarily to address the following questions:

- to what extent, and for what purpose could coding and computational thinking be integrated in a redeveloped primary curriculum?
- what are the experiences and perspectives of teachers, parents, and children on coding/computational thinking?
- what types of resources and continuing professional development would teachers and schools require?

The evidence collected addresses each of these key questions and points to recommendations as to how such integration might be achieved.

The report reflects the agreement of principals, teachers, parents and children that we are living in a world immersed in technology. Teachers and principals participating in the initiative clearly stated that they feel an obligation to enable their students to become more digitally competent. While parents expressed a desire that their children be able to use and understand how technology works, they were fearful of the negative impact technology might have on children with screen-time, online behaviour, and the child’s personal safety being their major concerns.

Internationally, there is a growing recognition that learning about computational thinking and learning how to code enhances critical thinking and problem-solving skills, and, just as important, builds our understanding of a world suffused with digital technology. In the context of the digital agenda, coding is explicitly regarded as a key 21st century skill: *Coding is the literacy of today and it helps practice 21st century skills such as problem-solving, team work and analytical thinking* (EU Digital Single Market, 2016). Along the same lines, the European e-Skills Manifesto (McCormack, 2014, p.57) declares that *...the world is going digital and so is the labour market... Skills like coding are the new literacy. Whether you want to be* *an engineer or a designer, a teacher, nurse or web entrepreneur, you'll need digital skills*[6] (European Commission).

However, as coding and computational thinking is such a recent trend in curriculum development, there is still much to learn about the purpose and 'management' of this area of learning especially in a child's primary education. In the context of the Irish education system, understanding and applying the fundamentals of computational thinking and coding from an early age have added significance when we consider recent curriculum developments at post-primary level, such as the introduction of Computer Science as a Leaving Certificate subject, coding as an optional short course at junior cycle level, and broader developments in engineering and applied technologies.

Yet, aside from curriculum change at second level, the motivation for teaching computational thinking and coding in primary school emerges from the value we attach to developing children's awareness of the technology which pervades the world in which they live. It should also serve the purpose of empowerment, so that they are not just users of digital devices but that they can understand and be creative with digital technology to contribute to, and to change the world in which they live, for the better. In responding to the key questions posed at the outset, this chapter discusses the main findings from the initiative using the broad headings of *Approach to curriculum integration*, *Provision for digital technology* and *Conditions* which the principals and teachers deemed important to ensure digital technology becomes embedded in schools. In doing this, the chapter links the findings from the school-based initiative with national and international research.

Approach to curriculum integration

Careful consideration must be given to how best to design and integrate coding and computational thinking into the primary curriculum to empower children to understand and change their world. This report suggests that there is clear alignment of the findings from

[6] European Commission, *Developing Computational Thinking in Compulsory Education*, 2016, para 3, p.10.

Phase 2 of the initiative, the review of literature on computational thinking[7] and the investigation of curriculum policy on coding in six jurisdictions.[8]

In the NCCA Curriculum Investigation, creating with technology, understanding technology and using technology were highlighted as the key components of a technology and/or coding curriculum, albeit named differently in each jurisdiction. Principals, teachers and parents highlighted the need for all three of these to be integrated in any proposed redevelopment of the primary curriculum. They also stated that learning how to use digital devices and technology in a safe way was very important and that online safety and ethical behaviour needed to be addressed.

Teachers participating in the school-based initiative were divided as to whether coding and computational thinking should be taught as a separate subject or fully integrated in existing subjects. That said, the consensus was that the essential skills associated with learning how to code should be taught explicitly. Once these had been mastered, the consolidation of the concepts of coding and computational thinking would be best achieved in a cross-curricular fashion. Cross-curricular integration was also referenced and recommended in the commissioned research paper on computational thinking.

Introducing coding and computational thinking concepts to children at an early age is commonly held to be the ideal scenario, a belief reflected across some of the countries investigated in the Curriculum Audit and reflected in the opinions of the teacher participants in the initiative. As Bell *et al*.[9] states, *children learn languages better at a young age, and this may transfer to learning the language of computing*.

It can be argued that for children to thrive in an ever-changing digital world the focus should not only be on technical skills such as coding or computational thinking but also on developing soft skills[10] such as play, creativity, communication, collaboration, and critical thinking.

[7]Milwood et al., 2018. Review of Literature on Computational Thinking. www.ncca.ie/media/3937/ncca-coding-in-primary-schools-initiative-research-paper-on-computational-thinking-final.pdf
[8]NCCA, 2017. Investigation of curriculum policy on coding in six jurisdictions. www.ncca.ie/media/3545/primary-coding_investigation-of-curriculum-policy-on-coding-in-six-jurisdictions.pdf
[9] Bell et al., 2015, Proposed Digital Technologies curricula up to NZC level 5, pp.2-3.
[10]Hackman, J., 2012, Hard evidence on soft skills www.sciencedirect.com/science/article/abs/pii/S0927537112000577?via%3Dihub

Some of these skills are fundamental to the curriculum specifications for Leaving Certificate Computer Science[11], the junior cycle coding short course[12] and the specifications for engineering[13] and applied technology.

> *Skills that encourage the student to solve problems through creation, innovation, communication, collaboration and exploration, all of which are developed in an active learning environment where students can advance their ideas from conception to realisation*[14].

These skills can also be identified in *Aistear* which supports an emergent and inquiry-based approach to curriculum development in early childhood whereby the curriculum is based on children's interests, questions and experiences.

To enable teachers to explore these soft skills, the school-based initiative focused on a playful, hands-on, project-based approach to learning about coding and computational thinking. Children worked with peers in a playful way on projects they felt passionate about, to develop, as Resnick[15] describes them, the four Ps of creative learning—projects, peers, passion and play. The use of unplugged activities and tangible or physical computing devices also enabled the children to experience coding in a tactile, physical and creative way.

Children participating in the initiative clearly embraced the use of technology in their classrooms, using a project-based, child-centred, playful methodology with no fear of failure. Several teachers commented that they had become the 'guide on the side' or the facilitator of classroom activities. They also commented on how children themselves had become the 'experts' and mentors not only to fellow classmates but to the classroom teachers themselves.

---

[11] NCCA, Computer Science Leaving Certificate specification, (2018) nccacurriculum.azurewebsites.net/Senior-cycle/Senior-Cycle-Subjects/Computer-Science/Introduction

[12] NCCA, Coding short course, Junior cycle specification, (2018) nccacurriculum.azurewebsites.net/Junior-cycle/Short-Courses/Coding/Statements-of-Learning

[13] NCCA, Engineering, Junior Cycle specification, (2018) nccacurriculum.azurewebsites.net/Junior-cycle/Junior-Cycle-Subjects/Engineering/Rationale

[14] NCCA, Applied Technology Specification nccacurriculum.azurewebsites.net/Junior-cycle/Junior-Cycle-Subjects/Applied-Technology/Rationale

[15] Mitchel Resnick: The Four P's of Creative Learning scratch.by/en/about/news/mitchel_resnick_the_four_p_s_of_creative_learning/

These approaches can be clearly aligned with and identified in the *UNESCO ICT Competency Framework for Teachers*[16] and the European Framework for the *Digital Competence of Educators: DigComEdu*[17] which were localised for the formation of Ireland's *Digital Learning Framework for Primary Schools*[18] which states that:

> *It is not enough for technologies to be merely available in the classroom – they should be deeply embedded in all classroom activities by supporting a constructivist approach to teaching and learning*.

Any approach adopted to support the future development of coding and computational thinking should include a whole school, student-centred, holistic, inclusive, age- and developmentally-appropriate curriculum underpinned by the core principles mentioned above, and the constructivist methodological approach indicated in the Digital Learning Framework and adopted in the *Coding in Primary Schools* initiative.

Provision for digital technology

It is clear from evidence gathered in Phase 1 of the school-based initiative, that an ever-increasing number of primary schools are moving beyond the current curriculum and are already teaching about coding and computational thinking in their classrooms. It was also evident that children's classroom experiences of coding and computational thinking varied greatly depending on the expertise of their teacher, the pedagogical approach taken, and the types of resources used. To ensure greater consistency and to support progression in the teaching and learning of coding and computational thinking in primary school it will be necessary to clarify learning outcomes and pedagogical approaches and to exemplify good practice. Teachers highlighted that a playful, child-centred, project-based approach to the teaching of coding and computational thinking was very effective. Given the extensive literature in this area it follows that we need to define and describe play within the area of

[16]Unesco Competency Framework for Teachers unesdoc.unesco.org/ark:/48223/pf0000213475
[17] Digital Competence of Educators: DigComEdu ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/european-framework-digital-competence-educators-digcompedu
[18]Digital Learning Framework for Primary schools www.pdsttechnologyineducation.ie/en/Planning/Digital-Learning-Framework/Digital-Learning-Framework-Primary.pdf

digital technology and explore how it relates to computational thinking skills including experimenting, tinkering, collaborating, failing and succeeding. Furthermore, we need to describe how digital technology connects with the *Aistear* framework, as well as examining connections in learning in this area at post-primary level. This will ensure that the learning pathway for children is clearly defined and developed.

Further investigation in Phase 2 revealed a broader conception of digital technology and what children should learn. Using, understanding, being creative and interacting ethically and responsibly with digital technology were all prominent within the feedback received. Parents, in particular, pointed to the need to give consideration to the issue of digital wellbeing. This broader conception requires that we think more carefully about the integration of these aspects of digital competence in a redeveloped curriculum, while at the same time focusing on teaching the core concepts of coding and computational thinking.

A further issue to be explored in the context of a redeveloped primary curriculum is to examine how digital technology can become a named space within the curriculum in such a way as to allow some explicit teaching and learning to take place. In this, we need to consider the extent to which digital technology can be integrated across all curriculum areas and subjects and how a proper understanding of digital competence can be embedded. Work on the redevelopment of the primary curriculum includes the articulation of a set of key competencies to be supported and developed across the curriculum's areas and subjects. Such an approach is evident in the curricula of Finland and Northern Ireland[19] where digital technology is represented in a cross-curricular manner. Finland sets out 'ICT competence' as a cross-curricular competency, while Northern Ireland names 'Using Information and Communications Technology' as a cross-curricular skill. Within the Irish context, there is scope for naming digital technology as a cross-curricular competency, to ensure it is integrated across the range of learning areas and subjects in the redeveloped primary curriculum.

Furthermore, as highlighted by teachers in the initiative, an explicit subject space within the curriculum will be required to support learning outcomes in the area of coding and computational thinking. Building on the experience of other countries, such as France and

[19] Investigation of curriculum policy on coding in six jurisdictions www.ncca.ie/media/3545/primary-coding_investigation-of-curriculum-policy-on-coding-in-six-jurisdictions.pdf.

Canada (Ontario)[20], such explicit learning is commonly placed alongside science. Articulating digital technology as a cross-curricular key competency, as well as providing an explicit learning space, for example, through a redeveloped science and technology curriculum, would ensure that coding and computational thinking are developed from an early age by focusing on an initial mastery of foundational skills and concepts and then building on these across the curriculum. This approach also takes account of the current overloaded curriculum and the need to ensure that, as much as possible, the integration of relatively new curriculum content doesn't contribute further to overload.

Enabling children to develop the necessary 21st century skills and digital competencies to thrive in a world increasingly dominated by technology is crucial. Providing for these digital competencies in a redeveloped primary curriculum would align well with the current implementation of the DES *Digital Strategy for Schools, 2015-2020*[21] which states that it is *focused on realising the potential of digital technologies to transform the learning experiences of students by helping them become engaged thinkers, active learners, knowledge constructors and global citizens who participate fully in society and the economy.* Other initiatives—such as the implementation of the *Digital Learning Framework for Primary Schools*[22] which reflects the embedding of technology in teaching, learning and assessment using a constructivist methodology and *The Digital Learning Clusters Initiative*[23] whose purpose is to demonstrate the innovative use of digital technology in schools—are also relevant in this regard. Integrating computational thinking and coding in a redeveloped primary curriculum would help create greater coherence between the curriculum and emerging and developing practice across schools.

[20] Audit of the Content of Early Years and Primary Curricula in Eight Jurisdictions. Desk study for the National Council for Curriculum and Assessment. www.ncca.ie/media/3927/ncca-key-findings-synthesis-breadth-and-depth.pdf

[21]Digital Strategy for Schools, 2015-2020 www.education.ie/en/publications/policy-reports/digital-strategy-for-schools-2015-2020.pdf

[22]Digital learning Framework for Primary Schools www.education.ie/en/Schools-Colleges/Information/Information-Communications-Technology-ICT-in-Schools/digital-learning-framework-primary.pdf

[23]Digital Learning Clusters initiative www.pdsttechnologyineducation.ie/en/Good-Practice/Projects/Clusters/

Conditions

Time

As a new field requiring the development of particular skills and competencies of teachers and children, integrating coding and computational thinking in the curriculum depends not only on the development of teachers' expertise, but also on their willingness to experiment and learn.[24] As reflected in the feedback from the teachers participating in Phase 2, time will be needed to embed the concepts of coding and computational thinking in classrooms. Having time would enable teachers to work collaboratively, to experiment, learn and reflect on classroom practice with colleagues within their own school and in other schools.

Professional development

The introduction of coding and computational thinking skills and a cross-curricular digital competency into a redeveloped primary curriculum would create a demand for large-scale in-service and continuing professional development. Teachers' current limited albeit expanding knowledge in the area, would demand the inclusion of both pre-service and in-service support. It would require that teachers have an open mindset and be willing to move outside their comfort zone to tackle new challenges.

Importantly, significant investment has already been made in the development of teachers' skills and expertise in this area. Since 2011, the Professional Development Service for Teachers (PDST), in collaboration with several national organisations such as Lero[25] and the ICS[26], have established various online and face-to-face professional development programmes. To date, approximately 3,180 primary teachers have participated in coding and computational thinking face-to-face workshops and, since 2014, a total of 3,463 teachers have completed the online 'Scratch for Learning course'[27], both facilitated by the PDST. Such

[24] Balanskat A., Engelhardt K., Licht A.H., (2018). Strategies to include computational thinking in school curricula in Norway and Sweden- European Schoolnet's 2018 Study Visit. European Schoolnet, Brussels.
[25] Lero the Irish Software Research Centre www.lero.ie/
[26] Irish computer Society www.ics.ie/
[27] www.pdsttechnologyineducation.ie/en/Training/Courses/Scratch-for-Learning-online-.html

numbers and training are considerable and provide a very important foundation from which to build going forward.

Due to the nature of the constant change within the technology sector, teachers' professional development in this area would need to be reviewed on a regular basis. Consequently, the development of communities of practice, which provide opportunities for teachers to generate, test and share ideas in a safe and collaborative way, should be encouraged.

School leadership

Research shows that leadership plays a key role in enabling schools to implement change in learning and teaching. Leadership is a social process of influence, mobilizing others' efforts to reach specific objectives serving a vision[28]. Schools within the initiative shared examples of this in action, where school leadership both enabled and supported the work taking place in classrooms. The importance of shared responsibility in the rollout and support of such teaching and learning within schools was highlighted as an important consideration by teachers. Teachers identified that the role of the school principal will be critical in this regard and school leaders will need to recognise the importance of developing their own skills and investing in continued professional development for themselves and their staff.

Resources and infrastructure

Teachers and principals identified the need for ongoing guidance as to the best possible solutions regarding digital technology infrastructure and classroom resources. Principals noted the need to establish a level playing field regarding technology effectiveness within schools. Enabling all schools to have adequate access to digital learning resources and an adequate technology infrastructure, must be seen as a priority if coding and computational thinking are to become part of a redeveloped curriculum.

[28] Leithwood K., Jantzi, D. & Steinbach R., (1999). *Changing Leadership for Changing Times*, Philadelphia: Open University Press.

Chapter 5: Conclusion and future directions

The report shows consensus across stakeholders that we are living in a world immersed in technology. Teachers and principals in the initiative stated clearly that they feel an obligation to enable children to become more digitally competent. There is a clear alignment of findings from Phases 1 and 2 of the NCCA's school-based initiative with the review of literature on computational thinking and the investigation of curriculum policy on coding in six jurisdictions. These findings include the identification of three aspects of digital competence—creating with technology, understanding technology, and using technology—as fundamental to the inclusion of coding and computational thinking in a curriculum.

Parents expressed the need for their children to be able to use and understand how technology works but also feared the potential negative impact of technology on children due to excess screen-time, online behaviour and threats to the child's personal safety. Due to the increased use and promotion of digital technology in classrooms schools are becoming increasingly aware of a shared responsibility with parents for the digital wellbeing of children. In 2001, Prensky[29] in his paper, *Digital Natives, Digital Immigrants*, claims that, *Our students have changed radically. Today's students are no longer the people our educational system was designed to teach* (p.1). He refers to children as digital natives: *the first generations to grow up with this new technology. They have spent their entire lives surrounded by and using computers, videogames, digital music players, video cams, cell phones, and all the other toys and tools of the digital age.*

Now, in 2019, we need to be cognisant that these 'digital natives' may not have all the necessary skills[30] to function in an ever-changing digital world, particularly in the mainly unregulated online environment in which most children find themselves. This then confronts us with new issues regarding the inclusion of coding and computational thinking in a redeveloped curriculum. The findings have shown that learning about the core concepts of

---

[29]Prensky M., (2001). Digital Natives, Digital Immigrant www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf

[30] Kirschner, P. and De Bruyckere, P., 2017. The myths of the digital native and the multitasker www.sciencedirect.com/science/article/pii/S0742051X16306692?via%3Dihub

coding and computational thinking would benefit from being taught separately and then integrated and developed across the curriculum. A playful, child-centred, project-based pedagogy incorporating the use of unplugged, plugged and physical/tangible device strategies is considered, by the teachers and principals, to be an effective approach. Finally, developing a whole school approach and having a shared vision and understanding are critical in the implementation of any curriculum change and special attention should be given to training and supporting teachers to meet this new challenge.

Coding and computational thinking were the focus of this initiative as requested by the then Minister for Education and Skills in 2016. However, as the digital world in which children live is constantly changing, and concerns intensify about the invasive nature of social media and the overuse of screen-time, concern for children's digital wellbeing also needs attention.

Future directions

The report has highlighted various important dimensions and approaches to teaching and learning coding and computational thinking in the primary classroom. The sections below chart short- and longer-term steps to integrating coding and computational thinking in the curriculum for all schools. These steps are shaped by the feedback from Phases 1 and 2 of the school-based initiative and by the findings from the international curriculum investigation (NCCA, 2018) and the review of literature on computational thinking (Milwood *et al*., 2018).

Short-term steps

The proposed short-term actions involve:

- revising the original learning outcomes (based on feedback received through Phase 2 of the initiative) to ensure that the progression from junior infants to 6th class is appropriate and clear.

- disseminating the findings from the initiative in order to encourage and support other schools' work in the area.
- publishing further classroom examples from the initiative on www.ncca.ie to support schools interested in finding out more about teaching computational thinking and coding.
- continuing with interagency collaboration, for example, with PDST, Education Centres, DES Digital Cluster Initiative and Webwise, to ensure a joined-up approach in the work with schools on digital technologies and to inform the ongoing redevelopment of the primary curriculum.

Longer-term steps

In addition, there are more systemic and longer-term actions for consideration. Some of these pertain specifically to the NCCA and its work in curriculum development, while others are beyond the curriculum itself. As noted earlier, the NCCA is currently reviewing and redeveloping the primary school curriculum. This is the first time, in almost twenty years, that those involved and interested in primary education have had an opportunity to consider what type of curriculum we need for the next decade. This work provides a timely opportunity to integrate coding and computational thinking skills in the curriculum. In progressing this, the initiative has highlighted the importance of

- introducing digital technologies to children from an early age.
- defining and describing play in the area of digital technology and exploring how it relates to computational skills.
- adopting and supporting a playful, child-centred, project-based approach to teaching coding and computational thinking.
- creating an explicit space in the curriculum, for example, in 'science and technology' to enable teaching and learning of the fundamental skills and concepts of coding and computational thinking, alongside integration across the curriculum.

- including digital competence as part of the key competencies in the redeveloped primary curriculum, and ensuring that this prioritises using, creating, understanding and interacting ethically and responsibly with technology.
- ensuring connections with current and future developments at preschool and post-primary sectors, so that progression in children's learning is clearly defined and developed.

Beyond the curriculum

As with most curriculum change, factors outside the curriculum play a key role in shaping implementation. Integrating coding and computational thinking in the primary curriculum, likewise, gives rise to other considerations. Arising from the initiative, these include the need for

- schools to have adequate time to introduce and embed curriculum change of this type, including time to reflect on what works and what doesn't.
- whole staff participation in continuing professional development including a foundational course for all teachers in digital technology.
- continuing professional development for teachers and school leaders which is hands-on and collaborative in nature and with a clear emphasis on practical classroom strategies.
- incorporating a stronger focus on digital technologies across the curriculum as part of Initial Teacher Education.
- schools to have continued access to IT funding, to advice and guidance on IT infrastructure and, where necessary, faster and more efficient access to the internet.

As was evident from the initiative and reflected in the steps mapped out above, integrating coding and computational thinking in the primary curriculum is challenging. Nonetheless, the current redevelopment of the curriculum provides an opportunity to revisit the purpose of a primary education. Today's and tomorrow's children live and interact in a digital world and as reflected by teachers, principals and parents alike, primary education can (and should) play an important role in equipping them to be ethical and responsible users of, and creators with technology.

References

Balanskat A., Engelhardt K., Licht A.H., (2018). Strategies to include computational thinking in school curricula in Norway and Sweden- European Schoolnet's 2018 Study Visit. European Schoolnet, Brussels.

Bell, T and Duncan, C. (2015) *Proposed digital technologies curricula up to NZC level 5*. Cantubury Uni.

Berk, L.E., and Meyers, A.B. (2013). *The Role of make-believe play in the development of excutive function: Status of research and future directions.* American Journal of Play, 6(1), 98.

Berry, M (2013). Computing in the national curriculum: A guide for primary teachers. NAACE. Available at www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf Accessed 29th August 2016

Bers, M. (2017) *Coding as a playground*. New York: Routledge.

Bers, M. U. (2012) Design digital experiences for positive youth development: From playpen to playground. Cary, NC: Oxford.

Bottino, R. (2015). National Report on Education and Technology: Italy. Available at http://link.springer.com/article/10.1007/s10639-016-9475-z Accessed: 26th August 2018

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77-101.

Coding/computer programming: European Schoolnet. (2015). Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe. Available at fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1- 8105-ea27c0d2bbe0

Computer science: Technology Schools.org. (2016). What is the Difference Between Computer science and Computer Information Systems Majors? Available at technologyschools.org/what-is-thedifference-between-computer-science-and-computer-information-systems-majors/

Computing at School Scotland, www.cas.scot/

Creswell, J. W., & Plano Clark, V. L. (2007). Designing and conducting mixed methods research. Thousand Oaks, CA: Sage.

CSTA standards, Computer Science Teachers Association, K-12 Computer Science Standards. Available at www.csteachers.org/page/standards

Dagiené, V. (2015). National Report on Education and Technology: Lithuania. Available at link.springer.com/article/10.1007/s10639-016-9475-z. European Schoolnet (2015).

Computing our future, Computer programming and coding Priorities, school curricula and initiatives across Europe. Available at

fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1- e1a6-4bf1-8105-ea27c0d2bbe0

Department of Education (2013). *Computing programmes of study: key stages 1 and 2*, National curriculum in England, Reference: DFE-00171-2013.

Digital Competence of Educators: DigComEdu. Available at ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/european-framework-digital-competence-educators-digcompedu

Digital Learning Clusters initiative (2018). Available at www.pdsttechnologyineducation.ie/en/Good-Practice/Projects/Clusters/

Digital Learning Framework for Primary Schools (2018). Available at www.education.ie/en/Schools-Colleges/Information/Information-Communications-Technology-ICT-in-Schools/digital-learning-framework-primary.pdf

Digital Strategy for Schools (2015-2020). Available at www.education.ie/en/publications/policy-reports/digital-strategy-for-schools-2015-2020.pdf

Duncan, C., Bell, T. and Tanimoto, S. (2014). *Should your 8-year-old learn coding?* In Proceedings of the 9th Workshop in Primary and Secondary Computing Education on - WiPSCE ’14, pages 60–69, New York, New York, USA, ACM Press.

Farrell, K. et al. (2017). Teach Computing Science, A Guide for Early Years and Primary Practitioners.

Finnish National Board of Education (2014). National Core Curriculum for Basic Education, The Scottish Informatics and Computer Science Alliance.

Futschek, G. (2015). National Report on Education and Technology: Austria. Available at link.springer.com/article/10.1007/s10639-016-9475-z

Grandbastien, M. (2015). National Report on Education and Technology: France. Available at link.springer.com/article/10.1007/s10639-016-9475-z

Grassel, E. & Schirmer, B., (2006). The use of volunteers to support family careers of dementia patients: results of a prospective longitudinal study investigating expectations towards and experience with training and professional support, 39 (3): 217-226 J.

Gulsecen, S. & Kubat, A., (2006). Teaching ICT to teacher candidates using PBL: A qualitative and quantitative evaluation. *Educational Technology & Society*, 9 (2): 96-106.

Harris, A. (2003). Teacher leadership as distributed leadership: heresy, fantasy or possibility? School leadership & management, 23(3), 313-324.

Johnson, J. S. and Newport, E. L. Critical period effects in second language learning: The influence of maturational state on the acquisition of English as a second language. Cognitive Psychology, 21(1):60–99, Jan. 1989.

Johnson, M.P., (2006). Decision models for the location of community corrections centers. Environment and Planning B-Planning & Design 33 (3): 393-412.

Keane, N. and NcInerney, C. (2015) *Report on the Provision of Courses in Computer Science in Upper Second Level Education Internationally.* LERO, The Irish software research centre, report commissioned by the NCCA.

Kimn, H.J. (2015). National Report on Education and Technology: Republic of Korea. Available at link.springer.com/article/10.1007/s10639-016-9475-z

Kirschner, P. De Bruyckere, P (2017) The myths of the digital native and the multitasker. Available at www.sciencedirect.com/science/article/pii/S0742051X16306692?via%3Dihub

Kloos, C. (2015). National Report on Education and Technology: Spain. Available at link.springer.com/article/10.1007/s10639-016-9475-z

Koivisto, J. (2015). National Report on Education and Technology: Finland. Available at link.springer.com/article/10.1007/s10639-016-9475-z

Koodi (2016), Learning paths Finland, koodiaapinen.fi/en/

Lee, I. (2012) Computer Science: Critical K-8 Learning. In P. Phillips, editor, Special Issue Computer Science K-8: Building a Strong Foundation, pages 10–11. Computer Science Teachers Association.

Leithwood K, Jantzi, D & Steinbach R (1999). Changing Leadership for Changing Times, Philadelphia: Open University Press

Magenheim, J. (2015). National Report on Education and Technology: Germany. Available at link.springer.com/article/10.1007/s10639-016-9475-z

Margolis, J., Fisher, A. (2013). Unlocking the clubhouse: Women in computing. MIT press.

McInerney, C., Tiziana, M. (2015). Software as a high-tech weapon in Ireland's strategy. 2015 IEEE 39th Annual International Computers, Software & Applications Conference

Mills et al, (2017). Case study research: foundations and methodological orientations. Available at www.ssoar.info/ssoar/handle/document/58015

Milwood et al, 2018. Review of Literature on Computational Thinking. www.ncca.ie/media/3557/primary-coding_review-of-literature-on-computational-thinking.pdf

National Council for Curriculum and Assessment (NCCA). *ICT (Information and Communications Technology) (2016)*. Available at www.ncca.ie/en/curriculum_and_assessment/ict/

NCCA, Applied Technology Specification, (2018). Available at nccacurriculum.azurewebsites.net/Junior-cycle/Junior-Cycle-Subjects/Applied-Technology/Rationale

NCCA, Coding short course, Junior cycle specification, (2018). Available at nccacurriculum.azurewebsites.net/Junior-cycle/Short-Courses/Coding/Statements-of-Learning

NCCA, Computer Science Leaving Certificate specification, (2018). Available at nccacurriculum.azurewebsites.net/Senior-cycle/Senior-Cycle-Subjects/Computer-Science/Introduction

NCCA, Engineering, Junior Cycle specification, (2018). Available at nccacurriculum.azurewebsites.net/Junior-cycle/Junior-Cycle-Subjects/Engineering/Rationale

New Zealand Ministry of Education, (2017) Digital Technologies Hangarau Matihiko, Draft for consultation. Available at www.education.govt.nz/ and www.tewhariki.tki.nz

Northern Ireland Council for the Curriculum, Examinations and Assessment, NI. Available at ccea.org.uk

Nowell, L., Norris, J., White, D., & Moules, N. (2017). Thematic Analysis: Striving to Meet the Trustworthiness Criteria. International Journal of Qualitative Methods, 16, 1–13.

Prensky, M 2001, Digital Native, Digital Immigrants. Available at www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf

Reynolds, N. (2015). National Report on Education and Technology: Australia. Available at link.springer.com/article/10.1007/s10639-016-9475-z

Sadler, D (2015). Queensland makes coding and robotics compulsory in schools. Available at www.startupsmart.com.au/advice/leadershipadvice/queensland-makes-coding-and-robotics-compulsory-in-schools/

Saito, T. (2015). National Report on Education and Technology: Japan. Available at link.springer.com/article/10.1007/s10639-016-9475-z

The Royal Society (2017). *After the reboot: computing education in UK schools,* royalsociety.org/computing-education. Available at royalsociety.org/-/media/policy/projects/computing-education/computing-education-report-summary.pdf

Unisco ICT Competency Framework for Teachers. Available at unesdoc.unesco.org/ark:/48223/pf0000213475

Viteli, J. and Tanhua-Piiroinen, E. (2017). *Comprehensive Schools in the Digital Age 2017 - Status review and recommendations for action*. Tampere Research Center for Information and Media, TRIM, University of Tampere, Finland, 2017.

Wing, J. (2014). Computational Thinking Benefits Society. 40th Anniversary Blog of Social Issues in Computing. Available at www.utad.pt/vPT/Area2/eventos/Documents/Artigo%203.pdf

Yin, R.K., (1984). Case Study Research: Design and Methods. Beverly Hills, Calif: Sage Publications.

Zainal, Z. (2007). Case study as a research method, J*urnal Kemanusiaan*, 9, 1-6. Available at www.researchgate.net/publication/41822817_Case_study_as_a_research_method

Appendices

Appendix A – List of Figures

Appendix B – List of Tables

NCCA

An Chomhairle Náisiúnta Curaclaim agus Measúnachta
National Council for Curriculum and Assessment