The Australian Educational Researcher
https://doi.org/10.1007/s13384-024-00742-z

Moving beyond the rhetoric: integrating coding into the English curriculum in Australian primary schools

Karen Woo[1] · **Garry Falloon**[2]

Received: 8 November 2023 / Accepted: 1 June 2024

Abstract

Coding and computational thinking are being hailed as the new literacy for the twenty-first century, and have become increasingly common in schools world-wide. At the same time, the sector is facing a global shortage of technology teachers, and technology lessons are frequently being delivered by non-specialist teachers. Therefore, some have suggested integrating coding into other curricula as a practical solution for non-specialist teachers to deliver outcomes in both Digital Technologies and their own subject areas, whilst also developing general capabilities. To better understand the benefits and practicality of integrating coding into the English curriculum, we present a detailed case study of an Australian Year 5/6 classroom where students engaged in learning units integrating both Digital Technologies and English curriculum outcomes. We explored the nature of students' interdisciplinary learning and general capabilities development through two learning units in which they coded animated narratives (CANs). We also built understanding of how non-specialist teachers in regular classrooms can develop the necessary technological, pedagogical and content knowledge (TPACK) to facilitate cross-curricular learning involving coding, and at the same time, promote general curriculum capabilities. Findings challenge the commonly held assumption that integrating coding can be a universal solution to specialist staff shortages and an overcrowded curriculum, and reveal the challenges faced by non-specialist teachers and school organisations that need to be overcome for successful implementation. However, they also indicate that when these challenges are met, integrated approaches can result in interdisciplinary learning, high levels of student engagement, and provide effective environments for general capability development.

Keywords Coding · General capabilities · Collaboration · Problem solving · TPACK · School environment

Extended author information available on the last page of the article

Published online: 12 July 2024

Introduction

Block-based programming, or coding, has become increasingly common in Australian schools. Although coding is not mandated in the Australian Curriculum, it is often seen as a natural way to expose students to computational thinking (Lye & Koh, 2014), which is present in many state syllabi and curricula (e.g., NSW, Victoria and Western Australia) and internationally (e.g., UK, US). In this article, we consider how an integrated curriculum in the form of a design-based coding task might deliver outcomes for both Digital Technologies and English disciplines, and also facilitate students' general capabilities, specified in the Australian Curriculum (ACARA, 2023). These capabilities are recognised internationally as crucial for students' success, and are also known as key competencies, competences, or twenty-first century skills, and are widely adopted by countries such as Spain, New Zealand, and the European Union (Tahirsylaj & Sundberg, 2020). We utilise an ecological system theoretical lens to reveal how these outcomes and capabilities can be fostered in regular classrooms (Falloon et al., 2023), and how non-specialist teachers can develop the necessary technological, pedagogical and content knowledge taking into account the affordances and constraints of regular school environments (Author1 & Author2, 2023; Koehler & Mishra, 2009). The study was undertaken in a Year 5/6 classroom in an Australian city, where the teacher and students participated in two integrated English—Digital Technologies units focused on coding animated narratives (CAN). It was part of a 3-year Australian Research Council (ARC) Discovery Project entitled *Coding Animated Narratives as Multimodal Authorship in Schools,* (referred to as the CAN project hereafter), that aimed to develop a new theoretical and practical foundation for the integrative teaching of coding and multimodal authorship for middle-school students. The CAN project was a cross-university collaboration, where researchers worked with primary and secondary schools in New South Wales and Victoria. A CAN is a short (1–2 min) story that incorporates features of multimodal semiotics such as changing camera angles, facial expressions, and gestures (Mills & Unsworth, 2018; Unsworth, 2013) that are coded using a block-based programming language, Scratch (Resnick et al., 2009). Previous research in the project has reported how CANs may facilitate computer science knowledge development (see Woo & Falloon, 2023), so for the purpose of this article, we focus on teacher and school attributes and characteristics that facilitated the development of general capability and interdisciplinary learning.

Literature review

The research foundation for general capabilities such as those included in the Australian Curriculum, can be traced back to the work of the OECD undertaken at the turn of the century. The *Definition and Selection of Competencies* project, also known as DeSeCo, was tasked with defining a set of key competencies 'necessary for individuals to lead a successful and responsible life and for society to face the challenges of the present and near future' (Rychen & Salganik, 2001, p. 7). The key competencies referred to 'multifunctional and trans-disciplinary competencies that are useful for achieving many important goals, mastering different tasks, and acting in unfamiliar situations' (Weinert, 2001, p. 52). This endeavour recognised that the knowledge and skills obtained at school are context dependent, but that individuals need to function in many different contexts and situations throughout their lifetime. The essence of the original key competencies manifest in schools across the globe through adaptations such as '21st Century' or 'future-focused' skills (e.g., Dede, 2010; Griffin & Care, 2015). In Australia, one of the goals of the Alice Springs (Mparntwe) Education Declaration is that 'all young Australians become: confident and creative individuals, successful lifelong learners, active and informed members of the community' (Education Council, 2019). This goal is supported by the Australian Curriculum's general capabilities, that comprise critical and creative thinking, digital literacy, ethical understanding, intercultural understanding, literacy, numeracy and personal and social capability (ACARA, 2023).

In this study we focus on students' development of 'digital literacy', and 'personal and social capability' (ACARA, 2023). The Australian Curriculum *digital literacy* capability states that students in Years 5–6 should 'select and use the core features of digital tools to efficiently complete tasks', and 'troubleshoot basic problems and identify repetitive tasks to automate' (ACARA, 2023). With regards to general curriculum capabilities, two components of *personal and social capability* are 'self-management' and 'social management'. *Self-management* is closely aligned with self-regulatory behaviours, such as planning, task management and persistence (ACARA, 2023), and these behaviours are known to relate to student motivation and engagement (Collie & Martin, 2019). *Social management* includes communication, collaboration, leadership, decision-making and conflict resolution. Specifically, students in Years 5 and 6 are expected to 'coordinate contributions of group members, suggesting improvements to ways of working and collaborative outputs' (ACARA, 2023). Furthermore, Drake and Reid (2018) argue that general capabilities 'transcend the disciplines. Thus, it makes sense that curriculum integration offers an effective way to teach twenty-first century capabilities' (p.35). *Personal and social capability* and *digital literacy* were focused on as they were considered integral to achieving successful outcomes from the CAN units.

Similarly, the creators of Scratch also argue that schools should give students opportunities to engage in cross-curricular, design-based coding projects, where they can integrate multiple knowledge disciplines, reason systematically, think creatively, and work collaboratively (Resnick & Rusk, 2020). In this article, integrated coding curriculum refers to cross-curricular learning that involves teaching coding in a subject area other than/in addition to Digital Technologies. Here, interdisciplinary learning may be evidenced by students' integrating knowledge from multiple disciplines in their CANs, which is often beyond that defined by predetermined syllabus outcomes. While integrated approaches are already common in schools as a practical response to a crowded curriculum and a lack of specialist teachers (Rich et al., 2019; von Wangenheim et al., 2017), empirical evidence of how students engage in interdisciplinary learning and develop general capabilities through curriculum of this nature is scarce.

Theoretical framework

This study utilises ecological systems theory (EST) (Falloon et al., 2022; Bronfenbrenner, 2005) and the technological, pedagogical and content knowledge (TPACK) framework (Koehler & Mishra, 2009), to build understanding of factors that contributed to the successful delivery of integrated coding curricula. Combining both theoretical referents facilitated close interrogation of how activities and decisions occurring in the wider school environment (mesosystem) affected the development of the teacher's TPACK within the classroom (microsystem), and in turn, how this influenced curriculum design, pedagogy and the delivery of targeted outcomes. Using Koehler and Mishra's (2009) TPACK framework as a theoretical referent is appropriate for this study, because it addresses the different knowledges teachers need for successful integration of digital technology into classroom curriculum.

The TPACK framework was developed from the work of Shulman (1986), who argued that effective teachers not only require content knowledge (CK) of the subject area and general pedagogical knowledge (PK), but also pedagogical content knowledge (PCK) to transform and deliver subject content in ways most aligned with the learning needs and preferences of their students. In this study, the subject content focused on multimodal artefact creation and coding knowledge developed through authoring animated narratives, while pedagogical content knowledge comprised understandings and methods associated with how to best teach these to students. Koehler and Mishra (2009) extended Shulman's (1986) model by adding technological knowledge (TK), which they identify as deep understanding of information technology for productive application in everyday lives. For this study, TK refers to coding as a new form of *computational fluency* to support student expression (Resnick, 2017). Teachers' understanding of how coding may be used to support computer science concept development and multimodal expression is considered technological content knowledge (TCK), and how to teach coding in a way that supports students' collaboration and problem solving is referred to as technological pedagogical knowledge (TPK). Santos and Castros (2021) suggested that TCK and TPK have the most impact on teachers' overall TPACK application and the achievement of students' learning outcomes, so we have paid particular attention in our study to how these were developed.

Adopting an EST perspective supported understandings about how student learning can be influenced by factors such as the professional learning their teacher receives, school culture and values, planning, staffing, resourcing, leadership, infrastructure, and the school community, and how this, in turn, manifests in teachers' TPACK. Understanding such factors is important because, as Resnick and Rusk (2020) noted, organisational and structural factors (such as rigid separation of subjects, timetables, age group, and space) present significant barriers to meaningful integration of coding in mainstream schools. This perspective is further supported by Mishra (2019), who argued in his recent update to the TPACK framework, that 'conteXtual knowledge' (XK) is needed, 'to go beyond seeing teachers as designers of curriculum within their classrooms but rather as intrapreneurs—knowing how their organization functions, and how levers of power and influence can effect sustainable change.' (p. 77).

Research questions

We adopted case study methods to build a thick description (Stake, 1995) of how the CAN units were implemented in this classroom, how the teacher's TPACK was developed and applied in the learning units, and how school factors influenced these processes and outcomes. Data collection and analysis were guided by these questions:

1. How do general curriculum capabilities and interdisciplinary learning manifest and develop in an integrated CAN curriculum?
2. How did the teacher's TPACK develop to support the integrated CAN curriculum?
3. What aspects and characteristics of the school supported the teacher's TPACK development and successful integrated CAN curriculum?

Research design

The broader CAN project within which this study took place followed a design-based methodology (Anderson & Shattuck, 2012) and was conducted over three iterations in two Australian states. For each iteration, a primary and a secondary school was recruited in each state, and in each school, two Year 6 or Year 7 teachers volunteered to participate in the project. To support their TPACK development, teachers participated in professional learning focused on multimodal semiotics (use of facial expressions, gestures and meanings to convey meaning) and Scratch coding. Scratch was chosen because it supports multimedia production and has a 'low floor' (Resnick, 2017) i.e., it is suitable for teachers and students without any prior coding knowledge. After participating in the professional development, the teachers planned their CAN units and taught them to their students.

Data for this article were collected from a teacher working in a public, coeducational primary school, who participated in the second iteration of the CAN project. The teacher was the home class teacher for a Year 5/6 composite class, meaning that he had the responsibility of teaching most of the subjects and providing pastoral care for these students. Therefore, he had an intimate knowledge of the students' individual strengths and weaknesses as well as their personalities and friendships, which he used as the basis of pairing the students for the projects.

Data were collected from the teacher and six student pairs. Limiting the number of student pairs supported deeper analysis of their learning processes, and helped build more detailed understandings of the factors that influenced these. Pairs were purposively selected by their teacher to reflect gender balance (6 boys and 6 girls) and a range of Scratch coding experience. These were broadly classified as low, mid, and high. 'Low' experience students (n = 6) were those who only started learning Scratch at the beginning of the school year, while 'mid' experience students (n = 3) had participated previously in out-of-school coding activities. Students who had more than 1 year of coding experience at school and reported learning by themselves at home, were identified as ‘high’ experience (n = 3). In reporting findings, pairs are referred to by the first initial of their names, and their profiles are summarised in Table 1.

The class was taught by non-specialist teacher, Mr S. (pseudonym used), who had 5 years of teaching experience. Prior to the CAN project, he had had no experience with Scratch or coding more generally. Supporting teacher technological knowledge (TK) development, Mr S. participated in four 1.5 h Scratch coding workshops to learn key code patterns used in creating animations. He also engaged in three, 1 h online seminars focused on multimodal authoring, to build the content knowledge (CK) to support the CAN units. These explored the use of facial expressions, gestures, and camera angles used in creating meaningful narratives using animations. He was given access to an online resource hub created by the first author, comprising video explanations of essential code patterns and ‘test your skill’ exercises. The exercises were designed to address the most commonly encountered problems faced by students, as identified during the first project iteration.

Mr S. coded his own animated narrative as an example, before he planned and taught the two CAN units. Unit 1 focused on students choosing ‘an important message for the world’, and then creating a narrative to communicate that message multimodally with emphasis on characterisation and use of camera angles. Weekly 60-min lessons were scheduled for the duration of Term 2 (10 weeks) and extended 3 weeks into Term 3. Unit 2 focused on creating a historical narrative about a famous Australian. This unit involved students researching and choosing a historical figure, and then scripting and recording a monologue in pairs and coding an animation to complement their monologue. Again, for the second unit, the classes were scheduled for 60 min each week for the duration of Term 4 (10 weeks). Staffing organisation in the school meant that Mr S. taught the CAN unit to his home class in addition to four other Year 5/6 classes, as a semi-specialist teacher.

Data collection and analysis

The six student pairs and the teacher were interviewed three times: once before project commencement (pre-interviews), after Unit 1, and again after Unit 2. The purpose of the student pre-interview was to collect data about their prior experiences and level of interest in coding, and to ask any questions they had about the research. The two post-interviews were approximately 30 min in duration and designed to gather students’ reflections about their learning, their perceptions of their Scratch projects, the problems they encountered while developing their projects (and how they solved them), and how effectively they worked together. The teacher interviews were between 30 and 60 min in duration and focused on understanding the teacher’s TPACK development, and identifying mesosystem factors that may have influenced this. The interview schedules are included in Appendices A-F.

All interviews were verbatim transcribed and imported into NVivo for analysis. Using Corbin and Strauss’s (2015) inductive coding method, interview data were

Table 1 Student pairs and links to their projects

Student pair	Gender	Scratch experience	Unit 1 project	Unit 2 project
EC	MM	High–Low	http://codingstories.com.au/CANSB3/EC_anon.sb3	http://codingstories.com.au/CANSB3/EC3_anon.sb3
GM	FF	Low–Low	http://codingstories.com.au/CANSB3/GM.sb3	https://scratch.mit.edu/projects/770100570
HN	FF	Low–Mid	https://scratch.mit.edu/projects/720726288	https://scratch.mit.edu/projects/774497747
LJ	MM	Low–High	http://codingstories.com.au/CANSB3/LJ.sb3	http://codingstories.com.au/CANSB3/LJ3_anon.sb3
NJ	MM	Mid–Mid	https://scratch.mit.edu/projects/720195621	http://codingstories.com.au/CANSB3/NJ3_anon.sb3
SS	FF	High–Low	http://codingstories.com.au/CANSB3/SS_anon.sb3	http://codingstories.com.au/CANSB3/SS3_anon.sb3

Links to the original projects have been provided wherever possible. Some projects have been modified to de-identify the students. The.sb3 files can be saved onto the local computer and uploaded to the Scratch website for viewing first manually coded line-by-line. Similar concepts were grouped to establish first-order categories. The two authors discussed the categorisation and conducted several iterations of data alignment and realignment within the first-order categories These generated subthemes detailing (1) students' general capability and knowledge outcomes; (2) teacher TPACK development and application, and (3) the influence of the mesosystem factors on the CAN curriculum. Samples of data coded under each of these are provided in Table 2.

The first author also observed 11 lessons and took field notes documenting the teacher's pedagogical practices and students' demonstration of general capabilities, particularly digital literacy (including problem-solving strategies), and personal and social capabilities (such as collaboration). Students' Scratch projects were also collected and used as prompts for questioning in the interviews. The researcher field notes and Scratch projects were crossed-referenced to strengthen interpretation of interview data.

Findings

The findings are organised around the three research questions.

RQ1: How do general curriculum capabilities and interdisciplinary learning manifest and develop in an integrated CAN curriculum?

Development of problem-solving skills as part of digital literacy

Interview data indicated that the problems students encountered during CAN authoring were wide-ranging—comprising both technical and nontechnical problems. The most commonly cited were *debugging* problems, which generally related to the timing of sprite appearances and confusion caused by inconsistent numbering of 'broadcast messages' when the students combined their files (e.g., NJ, SS, Unit 1 interview). Other general technical problems included difficulties with drawing (NJ, Unit 1 interview), inserting sound files (HN, Unit 1 interview), and using the Scratch backpack function for sharing sprites (HN, Unit 1 interview). Although some difficulties were apparent throughout both CAN units, data indicated that students transferred effective problem-solving strategies between units, so by the end of the second unit, most pairs reported they did not experience unsolvable problems when coding their second CAN. This is evidenced by some students described authoring the second CAN as 'smooth sailing' (MG, Unit 2 interview), while others commented, 'We knew what we were doing' (NJ, Unit 2 interview).

In their reflections, some students cited learning to troubleshoot when problems occur as a principal benefit from the CAN units. NJ commented: 'I feel like coding gives you skills and improves your skills at problem solving, because, if you make a mistake, you have to know where to go' (Unit 2 interview). Some students also indicated they made use of knowledge from other subject areas to inform their problem-solving strategies, as pair LJ reported:

Table 2 Themes, first order categories, codes, and sample data

Themes	First order categories	Subthemes	Sample data
Students' development	Digital literacy (problem solving)	Debugging (coding problems)	'And then the lady halfway she kept gliding towards the end without [unclear] walking animation. So we had to fix that.' (NJ, Unit 1 interview)
		Troubleshooting (non-coding problems)	'We had this weird backpacking issue where I just could not get the stuff out of my account history, put the sounds on one account and the designs on the other.' (HN, Unit 1 interview)
	Personal and social capabilities (collaboration)	Working together	'Mostly together because one day she would do it and I'd help her and then the next day I would do the coding and she would help me.' (GM, Unit 1 interview)
		Specialised roles	'H did the designing part. I was like, you can have the designing part, I'm just going to code. Well, we made the first view [scene] together so that I could start coding.' (HN, Unit 2 interview)
		Division of labour	'She would do one scene, I would do the other scene.' (GM, Unit 2 interview)
	Interdisciplinary learning	Apply to other subject areas	'I think if I read a book on this. I have no idea. Because sometimes when you're having fun you also learn things, like in coding, I don't know why, I find it fun. And I sometimes read something or watch something and it just sticks in my brain and I can't get it out of it.' (EC, interview, Unit 2)
		Artistic *and* technical	'I found it enjoyable because I just like making costumes and coding too… it's just fun' (NJ, Unit 2 interview)

Table 2 (continued)

Themes	First order categories	Subthemes	Sample data
Teacher Attributes & Knowledge[a] (microsystem)	Attributes	Commitment to learning	'I would say that, because I think that's what I was prepared to do myself, in learning. Was to commit to learning it really well.' (Mr S, Unit 1 interview)
		Personal interest	'Scratch is awesome, [laughs], um, I, I really have fun, even as an adult, it's a very basic level of coding, but I love putting things together.' (Mr S, pre-interview)
		Positive about challenges	'As soon as you overcome a seemingly unsolvable problem it just opens up a whole other world of opportunity, which I think is really cool. That's the essence of the project, really.' (Mr S, Unit 2 interview)
	Technological knowledge (TK)	No prior technical knowledge	'It helped me being a beginner myself' (Mr S., Unit 1 interview)
	Technological content knowledge (TCK)	Developing TCK	'It helped me being a beginner myself … so it was really easy for me to identify, okay, these are the different things that you need to learn in order to create a movie [CAN].' (Mr S., Unit 1 interview)
	Pedagogical content knowledge (PCK)	High initial PCK	'I feel like I'm slowly becoming a bit of a guru, at least in the style that, um, I've, I've learnt over the last couple of years.' (Mr S, Pre-interview)

Table 2 (continued)

Themes	First order categories	Subthemes	Sample data
	Technological pedagogical knowledge (TPK)	Tinkering with pedagogy	'in terms of pedagogy, I'm changing it class-to-class to see what works' (Mr S, Pre-interview)
		Teaching troubleshooting skills	'We spoke about the way they set out the code. Even though I taught that explicitly the first time around, there were still some very loose elements there. And when it came to trying to fix problems with the code it was very hard to identify where the problem had occurred.' (Mr S, Unit 2 interview)
		Supporting independent problem solving	'we're able to develop a checklist that the students followed explicitly for the first lesson or two, and then after that it just became automatic. So, developing that sense of autonomy, routine was really impressive to see, and I think that aided the problem solving.' (Mr S, Unit 2 interview)
	Pedagogical knowledge (PK)	Pairing decision	'I do think someone like [SS] were the perfect combination, which you're not going to find in every classroom. You might find a few, but if you pair all those really committed students together, then are you leaving your [student who lack focus] behind? Because someone needs to be able to help him learn his craft as well.' (Mr S, Unit 1 interview)

Table 2 (continued)

Themes	First order categories	Subthemes	Sample data
School characteristics (macrosystem)	School organisation	Stage/interdisciplinary team	'Our way of operating here, the fact that we're teaching five different classes on a rotation of five different key learning areas, the way that we do, certainly helps.' (Mr S, Unit 1 interview)
		Home class	'There was some from the other classes that we saw which, granted, I didn't spend anywhere near as much time with them as I did with my own, where they were using some of the sprites that existed in the library already. And you just have less capability to manipulate those sprites. So, I think that harms the final product, to be brutally honest.' (Mr S, Unit 1 interview)
	Syllabus	Compliance	'Knowing that we could tick those boxes simultaneously. I think that's also what got it across the line in terms of my teaching team as well… It's not that they wouldn't have seen the value, but they might not have seen the practicality of doing something like this, without us being able to tick that box in the English syllabus.' (Mr S, Unit 1 interview)

[a]Data from the three teacher interviews were coded separately, but are combined here to conserve space

L: It involves a lot more math, like a lot of timing, and a lot more attention.
J: Yes, like focus; you have to be focused a lot. And a lot of problem solving as well. (LJ, Unit 2 interview).

Such comments indicate that students understood the problem-solving processes they engaged when authoring their CANs, and that they were useful for advancing their general problem-solving capability whilst drawing on their knowledge from other disciplines.

Development of collaboration skills as part of social capabilities

Interview data indicated three main collaboration strategies were being operationalised: *working together* (sharing a laptop while working on the same task); *division of labour* (dividing the task equally, often by working on alternate scenes); and *role specialisation* (one student responsible for coding and the other for drawing).

In the first of these demonstrated in Unit 1, GM and HN reported that they were 'working together' within their pairs, and that this strategy related to their 'low experience' with Scratch. This approach enabled them to learn with and from each other, effectively bringing 'two brains to bear' on difficult problems. However, as they gained more knowledge and confidence, data also indicated that these students transitioned onto a different collaboration strategy that would support greater efficiency. To achieve this, GM adopted a *division of labour* approach, while HN opted for *role specialisation*. *Division of labour* for GM took the form of 'she would do one scene, I would do the other scene' (GM, Unit 2 interview). By creating alternate scenes, these students had equal input to the project, while also improving efficiency by allowing them to work independently on different sections at the same time.

A similar approach was also adopted by NJ, who were identified as having 'mid-level' experience with Scratch. However, a shortcoming of the 'creating alternate scenes' strategy was subsequent difficulties combining the scenes together in Scratch. In Unit 1, pair NJ were coding their scenes independently without realising they had different scene numbering systems. This caused major confusion at the end when they had to combine their scenes into a single narrative. Therefore, in the second unit, NJ transitioned to *role specialisation* by having one person focus on drawing (N) and the other on coding (J). In J's words, 'because he did most of the costumes and I did most of the coding … we worked together better because we learned our lesson [from Unit 1]'. (NJ, Unit 2 interview).

Role specialisation ended up as the preferred collaboration strategy for most pairs, although not all pairs began by using this approach, as illustrated by pair HN who transitioned from *working together* to *role specialisation*. As N commented, 'H did the designing part. I was, like, you can have the designing part, I'm just going to code' (HN, Unit 2 interview). They indicated this approach allowed them to work more efficiently. Another pair, SS, who adopted specialised roles in both units, decided to swap roles in the second unit, 'to give each other another opportunity to learn [other knowledge and skills]' (SS, Unit 2 interview).

Other pairs adopted specialised roles from the beginning, based on preferences and skillsets. For example, pair EC acknowledged from the beginning that E was knowledgeable in coding and C was a gifted artist. As a result, C completed most of the drawing and E did all the coding. In this example, specialisation contributed to the authoring of a highly sophisticated CAN, both in terms of technical coding and use of visual elements (see Fig. 1). The disadvantage of this approach was that C did not gain much coding knowledge, despite the pair successfully completing two CANs.

In summary, data indicated that all pairs (with varying effectiveness) self-organised and self-determined how they worked together across the two units. None of the strategies were perfect, but most students communicated sound reasons for why they chose to work in the way they did, and were prepared to modify their strategies where improvements could be identified.

Learning engagement and the CAN curriculum

Interview data indicated high levels of engagement by all six student pairs across both units, and they all reported enjoying the mix of technical and artistic aspects of the task. For example, student N reported, 'I found it enjoyable because I just like making costumes and coding too… it's just fun' (NJ, Unit 2 interview). The 'fun' these students referred to related to the process of authoring a CAN, and also satisfaction with the CAN they had produced. Student G expressed enjoyment in watching their own creation: '… to see what we've done was really fun' (GM, Unit 1 interview), and student E communicated pride in their achievement: 'It was fun … I feel really proud of both of us for making it possible' (EC, Unit 1 interview).

Furthermore, data suggested that the processes involved in authoring the CAN promoted student engagement with the subject matter being learnt. In the second unit, all student pairs created coherent, informative, original, and entertaining narratives about their chosen historical figure, with their background research

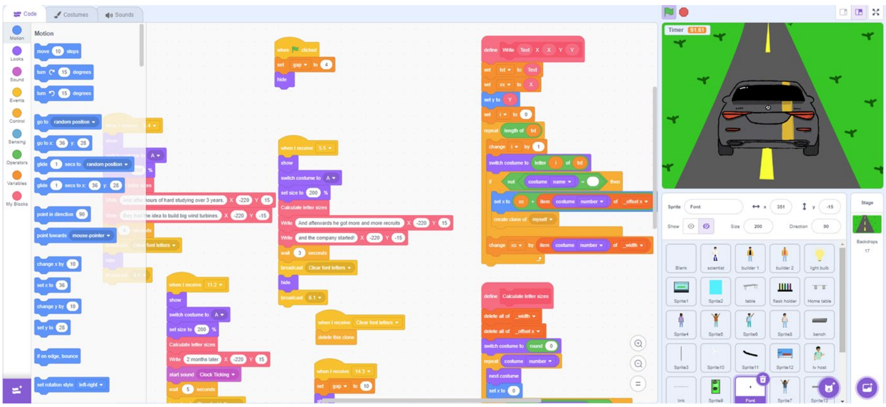

Fig. 1 Screenshot of student pair EC's project which demonstrates complexity of their coding and original drawing

indicating robust engagement with a range of historical sources. Figure 2 illustrates pair SS's well researched script for their chosen historical character, Mary Wade.

Some student pairs, like HN and EC, expressed that they do not normally enjoy learning about historical figures. However, student N explained that she found that the process of coding an animated narrative helped her better engage with and understand key concepts and information that were essential for creating a quality artefact:

> … when it comes to non-fiction, I don't like doing that. But technically, this is non-fiction. So if I had to write a report on Mary Bryant [the historical figure], I would hate doing that. But I loved doing this. It sort of keeps it more interesting, next level, it's teaching, like… the style. (HN, Unit 2 interview).

Such data indicate that students found the CAN format motivating, and that helped them productively engage with content that does not normally interest them, to create an original artefact.

RQ2: How did the teacher's TPACK develop to support the integrated CAN curriculum?

Data coded under this sub-theme were drawn from field notes and three interviews with Mr S. at three key data points: before the units (pre-interview), after Unit 1 implementation, and after Unit 2 implementation. Each interview was coded separately and is reported sequentially to build understanding on how his TPACK evolved over the course of both units.

Chapter	Title	Narration
1	Legacy	Which Australian convict was the youngest aboard Lady Juliana, has a book written about her and a prison named after her? Mary Wade, of course!
2	Early life	The star of our story was born on the 5th of October, 1777. She had three siblings, Henry Wade, Elizabeth Wade and Henrietta Wade. Her parents were Lawrence Wade and Mary Smith. Young Mary spent the majority of her childhood sweeping the streets, begging for money for her struggling family.
3	Crime	In 1788 aged 11, it was the final straw for Mary. She couldn't take poverty any more. So Mary and her friend, Jane Whiting, stole a hat, a scarf and a dress from Mary Phillips, a poor 8 year old. The girls then sold the hat and the dress to a pawnbroker.
4	How she was caught	**dramatically Alas, things started to go downhill for Mary. Another child reported her to a police officer, who found the scarf under her bed.
5	Consequences	Mary was then arrested and placed in Bridewell Prison. She was found guilty at her trial and was sentenced to death by hanging. Urgh! However fortunately for Mary, the king had just recovered from an illness, and in celebration all women holding death penalties were converted to transportation. Yippee!
6	Before transportation	Mary was placed in Newgate prison while she was waiting for the ship to arrive. The conditions were horrible. 12 year old Mary was chucked into a cell... with 49 other women! **shocked voice They had no toilets or beds, and they were fed nothing but bread and water. Bleh.
7	Transportation	The journey on Lady Juliana took 11 months. Mary was the youngest there.
8	Life in Australia	Upon arriving, Mary was moved to Norfolk Island, to relieve pressure on the overcrowded colony of NSW. In Australia Mary lived with Jonathan Brooker near the Hawkesbury River from 1809. It was here that Mary raised her family which numbered 7 children. Mary received her Certificate of Freedom on the first of September 1812. Her husband owned 30 acres in 1822 until bushfires destroyed their property. Mary and Jonathan went on to own 62 acres in Illawarra in 1828. Here, Mary lived till Jon's death on 14 March 1833. RIP, Jonathan, you were a good guy. **sigh dramatically
9	Death	Mary died at the ripe old age of 82, on the 17th of December, 1859.
10	Outro	Mary made some mistakes in her life, that's for sure. Stealing is bad, kids. There is no doubt that Mary Wade lived an eventful and fulfilling life. But the real takeaway lesson is, don't steal, or you might end up on the other side of world! No really, you will!

Your whole script should take between 1:30 and 2 minutes to read aloud.

Fig. 2 Sample student script for the historical monologue (Student pair SS)

Pre-interview

Pre-unit interview data indicated that Mr S. started the project with only tentative ideas of what learning outcomes the CAN units would yield for his students. He mentioned hoping to see high levels of student 'engagement', having a 'sense of pride', 'produc[ing] some quality work', and even just meeting the '90-s mark [for CAN duration]… for some students, that'd be a really big achievement'.

He concluded that he had only 'tempered expectations at the moment', because he was also new to coding. Pre-unit interview data also indicated Mr S. had high CK in English but low CK in Digital Technologies, and how he considered this reflected in his teaching practice. He characterised himself as a 'guru' in teaching narratives, indicating high PCK in English, and described how he planned to support students to construct their narrative by aiming for stories with 'big picture ideas', possibly 'related to the experiences that the students have had'—so they could 'explore it with a bit more depth'. However, in terms of coding, he admitted he was a complete beginner with Scratch. Often in the interview he described an experimental approach to developing his coding TPK, in that he was 'figuring it out' [as the units developed] and 'in terms of pedagogy, I'm changing it class-to-class to see what works'. His early strategy concentrated on teaching his students to use the art tools and expecting his colleague, Mr D., to teach the coding aspects. Unfortunately, for personal reasons, Mr D. had long absences in Terms 1 and 2, so in Term 2, Mr S. was forced to teach the coding aspects as well.

Unit 1

Unit 1 comprised four technical lessons on (1) using *broadcast messages* in coding; (2) creating camera angles; (3) code organisation, and (4) sharing assets using Scratch backpack. These were followed by two CAN planning sessions comprising a brainstorming session on themes for the students' stories, and narrative planning using storyboards. Students were required to work in pairs to plan their CAN using the storyboard template provided, and they could only start coding after Mr S. approved their storyboards. Figure 3 illustrates the whiteboard which resulted from their brainstorming session, and the smartboard displays the storyboarding template. Figure 4 shows a sample student storyboard from Unit 2.

A prevailing theme in interview data was Mr S.'s understanding of the importance of learning to code well, and his commitment to developing high TCK in this essential skill. However, he also indicated an important strategy in developing this was being prepared to learn from students, and being open to the prospect of failure:

> That's what I was prepared to do myself, in learning… was to commit to learning it really well… taking the time to get to know it [coding], working out what we need to learn in terms of the skillset that we need to develop, mapping that. I can do that as a teacher, but there are some technical skills that you just [need to learn] you need to be open to failure, to learn from your students as well. (Mr S., Unit 1 interview).

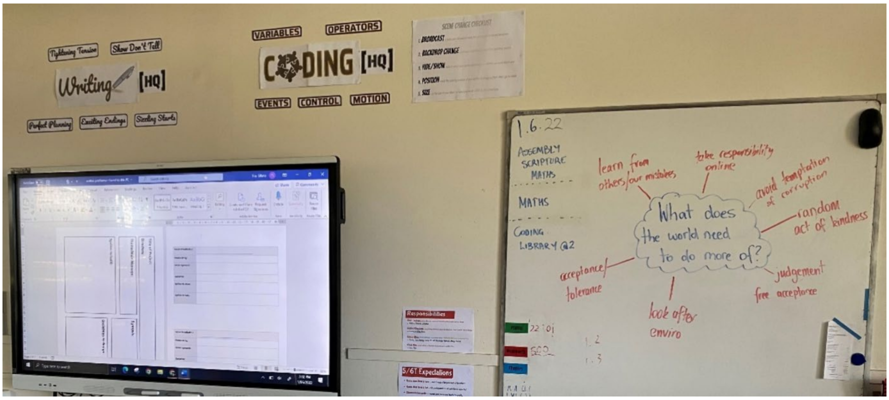

Fig. 3 Brainstorming and storyboarding session (Unit 1)

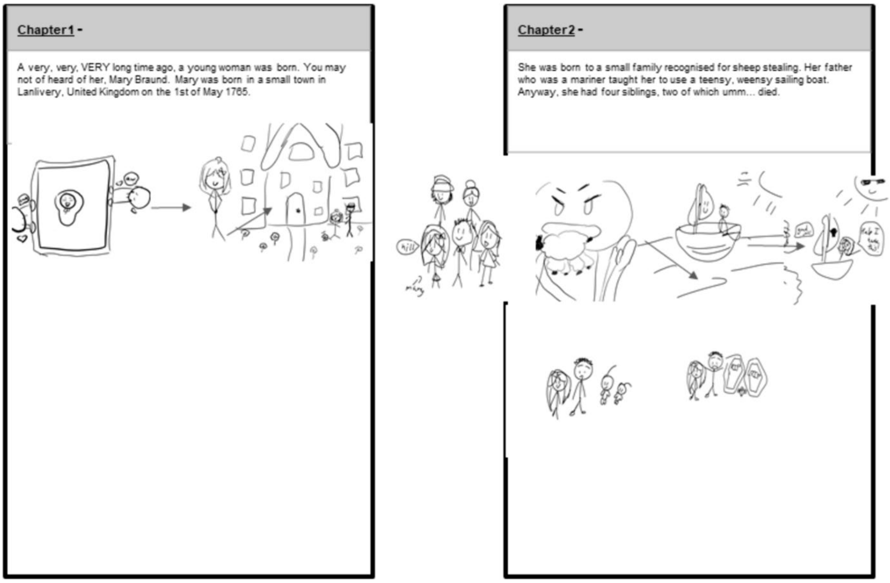

Fig. 4 Sample storyboard for Unit 2 (student pair HN)

Interview data indicated Mr S. drew heavily from his personal learning experiences, and that these informed his TPK: ‘It helped me being a beginner myself … so it was really easy for me to identify, okay, these are the different things that you need to learn in order to create a movie [CAN].’ He described the process as open and iterative: ‘We were being open-minded to adding things right up to the end’, and this approach reflected in, and was supported by, his growing knowledge and what he called his ‘retrospective planning and programming’.

Unit 1 data also indicated that Mr S.'s understanding and expectation of student outcomes evolved in response to his own learning: 'The more I learnt and the more that I could do, I started to expect the students to be able to do the same.' Those expectations mainly related to strategies used to solve debugging problems. As he explained, 'if there's a big glitch somewhere a minute into your movie, I expect you [students] to be able to go back and find it.' He commented that in Unit 1 most students in his home class were able to solve debugging problems, and he attributed this to their code organisation skills, which he explicitly taught. These data suggest Mr. S's development of TPK directly influenced his TCK. By reflecting on his own learning and identifying the important knowledge and skills required for the task, he was able to determine his pedagogical strategies to help students avoid common pitfalls and reach their goals through systematic methods.

Another theme present in Unit 1 data was the effect of the teacher's PK on the nature of student collaboration. Mr S. attributed successful collaboration to how he selected the pairs. As their home class teacher, he knew who would work well together, which was one of the first criteria he considered in selecting the pairs. However, apart from social compatibility, Mr S. indicated the pairings were diverse in nature, comprising pairs where both students displayed strong commitment to learning; pairs where there were complementary skills, and pairs where both students were lacking in skills or experience. The teacher's notes on the factors he took into consideration when pairing students are presented in Table 3.

While Mr S. commented that he needed to have conversations with some pairs about compromise and time management, overall, he considered most students worked effectively together without much monitoring. In his own words, 'we saw how it worked out, how collaborative most of them were… you couldn't do that blind… matching people off a piece of paper.'

Table 3 Mr S.'s notes for pairing students

Pairing	Student names (Year)	Notes
1	E (6)	Highly capable coder/diligent worker
	C (6)	Highly capable artist/lacks focus
2	S (6)	Highly capable artist/diligent worker/fast learner
	S (6)	Highly capable artist/diligent worker/fast learner
3	N (6)	Highly capable coder/diligent worker
	J (5)	Capable coder/diligent worker/fast learner
4	M (5)	Minimal experience/diligent worker
	G (5)	Capable coder and artist/diligent worker/fast learner
5	L (5)	Minimal experience/reasonably diligent worker
	J (5)	Minimal experience/lacks focus
6	H (5)	Highly capable artist/occasionally lacks focus
	N (5)	Minimal experience/reasonably diligent worker

Unit 2

Unit 2 interviews occurred after students had completed their project to develop a CAN about a historical figure incorporating an original recorded monologue. When asked what changes he made for Unit 2, Mr S. emphasised the greater time and importance placed on storyboarding. He explained that by requiring students to pre-plan using storyboards, students worked more efficiently in the coding phase:

> … it eliminates another complication that you run into when you begin coding. So, rather than having to think 'I want this character to move here at this time, and then following that we'll change the backdrop' and whatever it might be, that thinking has already been done, and you just simply worry about bringing it to life. (Mr S., Unit 2 interview).

The second change he made between units was spending more time on explicit teaching (i.e., giving direct and clear instructions) on some aspects of the task, for example, creating a monologue and simplifying some of the visual design elements, including the creation of stick figures, as well as basic props and backdrops. He explained that his increased explicit teaching about visual design was in response to pedagogical shortcomings he identified in Unit 1: '[while] it's important to give the students that creative licence and to be able to take ownership of their work, but … [in Unit 1] it became a bit loose'. He had observed that students' use of the Scratch library in Unit 1 had a detrimental effect on their work, because the visual design elements 'just didn't marry up on screen'. The weighting he placed on visual quality and accuracy was consistent with his classroom practice, where he frequently supported students to create original drawings. Sometimes it involved him drawing on a whiteboard to show the students the outline (Fig. 5); sometimes it involved him showing how it could be done using the Scratch art tools (Fig. 6); and sometimes students would commission him to create a sprite for their project (Fig. 7).

Fig. 5 Mr S. demonstrating drawing on a whiteboard

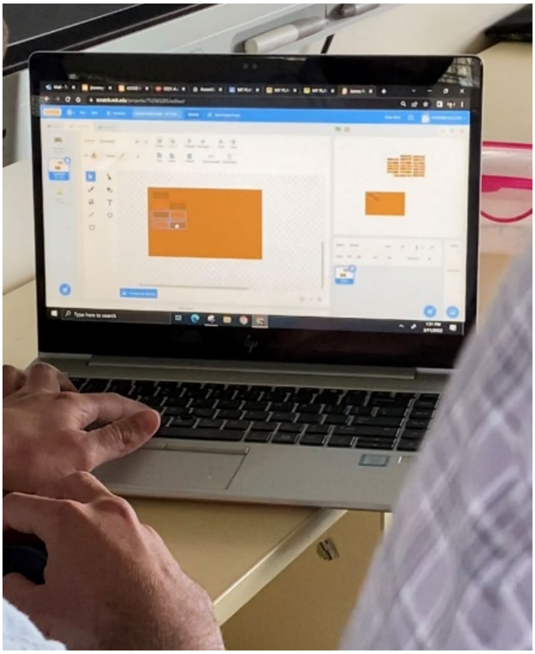

Fig. 6 Mr S. demonstrating to a student how to use the art tools to quickly create a brick wall by copy and pasting a repeated pattern

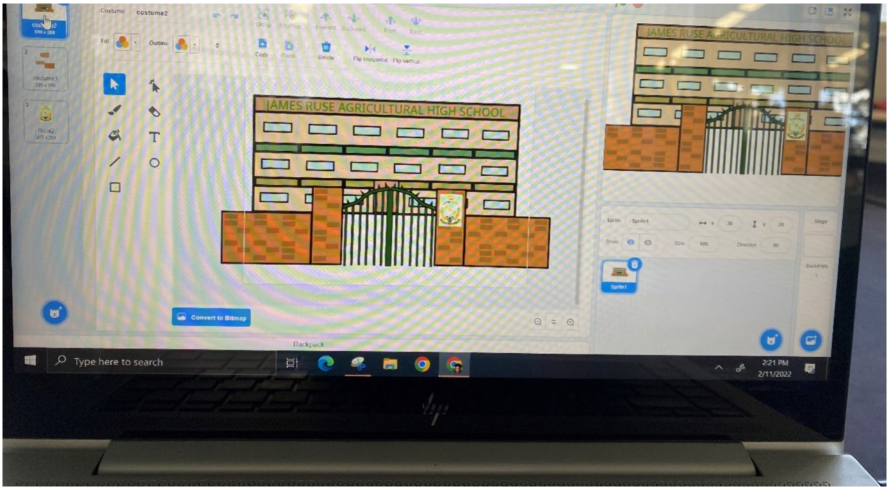

Fig. 7 Mr S. was commissioned by a student group to create a school building for their project

Mr S noted the influence of improved visual design knowledge on the overall quality of the student CANs, and in Unit 2, specifically asked students to simplify their designs for three reasons. First, simplifying visual design enabled students to spend less time creating finicky and complex designs and more time coding, in order to create smoother sequences and more seamless transitions. Second, simplifying the characters into stick figures allowed them to be more easily manipulated as sprites. Third, simplifying visual design aided the multifaceted purpose of the texts, primarily to inform, but also to entertain. Whilst these changes might, at a surface level, appear to be limiting students' creativity, data suggests they were effective for supporting students to become more fluent in their multimodal expression. As student pair LJ reflected:

> L: I think the second one [the Unit 2 project] looks better.
> J: The second one is just a bit more clean…Like it's just the designing is so much easier with the one we have now. And it's much more simple, it's not all complicated. Like last time some backdrops took me two sessions to make. This time I can make three in one session. (LJ, Unit 2 interview)

When asked what he understood to be the main arguments for teaching coding in schools, Mr S. found it difficult to provide a definitive response: 'I would've hoped that by the end of this year … I'd have a very succinct answer for you, but I don't, in all honesty.' Instead, he shared his observed benefits from the CAN units, which he summarised as three skills: 'Problem solving, building routine, and collaboration. [These] were the three skills that really presented themselves throughout this project.' Elaborating further, he regarded the first two skills as being closely related. In his observation, an important way to support problem solving was through developing systematic processes and routines. He gave as an example the checklist students used that helped them to create a scene change (Fig. 8). The checklist provided an initial scaffold that students 'followed explicitly for the first lesson or two, and then after that it just became automatic'. He further commented that 'developing that sense of autonomy and routine was really impressive to see, and I think that aided the problem solving'. Moreover, he considered his home class developed more robust and effective problem-solving strategies, largely because he worked more closely with them and spent more time supporting them between lessons.

Fig. 8 Mr S.'s scene change checklist

In terms of the extent of observed collaboration, Mr S. suggested results were mixed: ‘Some that we intentionally paired because we knew that they were going to work well together and others that we thought wouldn’t necessarily work as well… they probably yielded results that we expected.’ However, in further analysing this, he highlighted a range of student characteristics that appeared to influence the quality of their collaboration. These included the students’ levels of perseverance, their delegation and time management skills, experience with Scratch, friendship, patience (to help each other), and speed to learn new skills. Reflecting on the problem-solving and collaboration skills exhibited in his classroom, Mr S. believed that the CAN units had provided his students with very accessible and highly important opportunities for building these curriculum general capabilities:

> These [skills] that we know that these children will need to be able to access when they’re out in the workforce, eventually, doing these jobs that we’ve heard don’t even exist yet … I know deep down that we’ve done a really good job and I know that the students have benefited greatly. (Mr S., Unit 2 interview)

RQ3: What aspects and characteristics of the school environment supported the teacher’s TPACK development and successful integrated CAN curriculum?

Interview data coded under the ‘the influence of the school context on the CAN units’ sub-theme indicated several factors that Mr S. noted in the school environment that influenced his students’ outcomes and his own professional learning. These included class arrangements, his stage team, and syllabus and cross-curriculum links.

Class arrangements and time constraints

For his home class, Mr S. observed advantages from him teaching both the writing and coding unit requirements, which helped mitigate the time constraints imposed by the school-wide timetable. This provided the additional time needed to create original characters and backdrops, further enhancing the accuracy and quality of the CANs. However, with regards to his other classes that only had limited and timetabled lesson times, he commented on the effect this had on the quality of their CANs:

> There was some from the other classes that we saw which, granted, I didn’t spend anywhere near as much time with them as I did with my own… where they were using some of the sprites that existed in the library already. And you just have less capability to manipulate those sprites. So I think that harms the final product… (Mr S., Unit 1 interview).

The role of the teaching team to support concept development

Mr S. was a member of a teaching team that worked collaboratively in programming and curriculum delivery across the entire cohort of Year 5/6 students. He mentioned several times that although he was coordinating the CAN units, it was a collaborative effort to make CAN work in the school. Within this team, Mrs K. taught visual literacy throughout the year where students made slideshows and discussed different visual literacy techniques. Another team member, Mr D., had been teaching the students since their third year, using 'Grok challenges, Bebras challenges developing computational thinking skills often [through] unplugged [activities]' (Mr S., Unit 2). These earlier experiences meant the students came to the CAN units with related conceptual understandings that, in all likelihood, they applied to designing and coding their CANs.

Cross-curriculum links important to justify the CAN program

Mr S. emphasised the importance of cross-curriculum links between narrative writing (English) and coding (Digital Literacy) in making it possible to teach the CAN units in his school:

> Knowing that we could tick those [curriculum] boxes simultaneously. I think that's also what got it across the line in terms of my teaching team as well … it's not that they wouldn't have seen the value, but they might not have seen the practicality of doing something like this, without us being able to tick that box in the English syllabus. (Mr S., Unit 1 interview).

In summary, data highlight the complexities non-specialist teachers can face when attempting integrated units of this nature. The need to learn sufficient technical coding skills and understand how these are best used to communicate meaning though a multimodal representation, is a demanding task. However, when alignment exists that combines high levels of teacher commitment with responsive pedagogies, reflexive curriculum design, and supportive school structures, it can result in high levels of student interdisciplinary learning and capability development.

Discussion

The discussion is organised around the research questions to address how general capabilities in the Australian Curriculum and other outcomes were developed in these CAN units, and the teacher and school attributes that were fundamental to supporting integrated coding curricula of this nature.

Microsystem factors impact on student capability development and interdisciplinary learning

Digital literacy

Several attributes of Mr S.'s practice influenced students' development of digital literacy, especially problem-solving ability, during the CAN units. Through reflecting on his own learning process, Mr S. recognised the need to support students in developing systematic strategies for solving often complex coding and design problems. He identified frequently encountered problems such as timing and backpacking (i.e., code and sprite sharing) issues, and explicitly taught his students systematic strategies for solving them. In computer science terms, Mr S. taught logical and efficient debugging and code organisation skills that enhanced readability (see Basu, 2019). He did this recognising that his students' projects each contained hundreds, and sometimes over a thousand, code blocks, and that the code would be impossible to debug without logical organisation. As additional scaffolds, Mr S. also provided checklists to guide debugging (see Fig. 6), which helped students focus on the important elements to solve common code problems.

In summary, Mr S.'s pedagogy could be described as systematic and explicit when it came to teaching technical skills, both in terms of coding (e.g., setting up procedures and checklists) and in generating visual elements (e.g., insisting on original characters). This practice challenges the commonly-held view that creativity in coding is achieved by allowing students to discover technical knowledge primarily through tinkering (e.g., Resnick, 2017). Instead, our study indicates that explicitly teaching technical skills (whether it is for coding or for visual elements) enabled students to focus their attention on communicating a message with flair and originality, which we argue is a more meaningful manifestation of creativity in CANs.

Furthermore, an unanticipated and extended outcome of the project was that some Year 6 students decided to create a CAN as their graduation memoir, to mark their transition into high school. This voluntary use beyond the set classroom tasks demonstrates developing computational fluency in this new form of multimodal authorship (cf. Resnick & Rusk, 2020).

Personal and social capability

To support personal and social capability, Mr S. provided planning tools (e.g., storyboard templates) and gave occasional reminders to groups to help them with time management. However, what we do not have direct evidence of, is how Mr S.'s personal motivation and enthusiasm for the project may have influenced students' motivation. Regardless, there is emerging evidence in psychology that teachers' and students' motivation are interconnected, noting the importance of factors like classroom climate and teachers' autonomy-supportive practices for supporting students' motivation and engagement outcomes (Collie & Martin, 2019). In our study, Mr S expressed that he was 'intrinsically motivated to learn something new' (Mr S, Unit 2 interview), and that he was 'committed to learning [CAN] really well' (Mr S, Unit 1 interview). Moreover, he expected the same from his students. When asked what he considered was the most important element for students to succeed, he commented, 'commitment is the key'. We would suggest that his high expectation of students and his own motivation and enthusiasm were likely to have had a positive influence on his students' motivation and engagement, which in turn, led to enhanced self-management and task orientation.

While some researchers suggest that creative coding tasks can promote collaboration (e.g., Resnick & Rusk, 2020), other classroom-based research has demonstrated that this is not necessarily the case (e.g., Arawjo & Mogos, 2021; Spieler, 2018). In the current study, collaboration was deliberately designed by the teacher with thorough consideration of each pair's compatibility in terms of each person's strengths and weaknesses, as well as friendship and learning characteristics. Moreover, Mr. S established an environment where students were comfortable trying different strategies to organise themselves in ways that both partners made meaningful contributions to developing high-quality artefacts. This adds to the substantive body of research on the central role that teachers play in managing groupwork (see Blumenfeld et al., 1996; Falloon, 2024).

Interdisciplinary learning and higher order thinking

CAN provided a viable curriculum and structure supporting students' animation authoring, elevating the cognitive dimension of the task from 'analyse' to 'create' on Anderson and Krathwohl's (2001) Revised Bloom's Taxonomy. Specifically, students were *applying knowledge* in computer science (see Woo & Falloon, 2023), narrative structures (Christie & Derewianka, 2010) and visual literacy (Mills & Unsworth, 2018), and in their Unit 2 historical narrative, *creating factual knowledge* in the form of an animated narrative. Moreover, at the intersection between the technical and artistic aspects of the task, students expressed having 'hard fun' (Resnick, 2017). The multifaceted nature of the task allowed different student interests and talents to make meaningful contributions to the project, and also provided them with opportunities to draw on their knowledge from a range of disciplines.

Non-specialist teachers' TPACK development

Teachers' knowledge and efficacy is known to be influential to student learning (Falloon et al., 2022; 2023). Using the TPACK theoretical framework outlined in 'Theoretical framework' Sect., we analysed Mr S.'s development of TPACK that led to successful implementation of the CAN curriculum. This is presented graphically in Fig. 9. Initially, Mr S. brought to the project strong content knowledge (CK) and pedagogical content knowledge (PCK) in English. This was evident in his early decision to centre students' narratives around a meaningful theme that allowed students to bring in their own personal experiences. The importance of a meaningful theme has long been advocated by integrated curriculum scholars, as it allows students to integrate their learning experiences into their personal schemes of meaning (Beane, 1995).

Second, although Mr S. started as a non-specialist in Digital Technologies he was committed to the deep learning of coding—specifically, becoming highly proficient in coding animations (TCK), as well as learning effective methods for the teaching of coding (TPK). The trajectory of his knowledge development is indicated in Fig. 9 by the arrows. First, he participated in the professional learning sessions to develop basic technological knowledge (TK) (1). He then further developed in TCK, principally through creating his own CAN (2). After that, his reflection on his learning process informed his development of TPK. Furthermore, having access to five classes across the age level allowed him to try different pedagogies with different classes, and thus he refined his skills and curriculum design over the course of the units (3). It was his newly developed deep TPK in combination with his existing

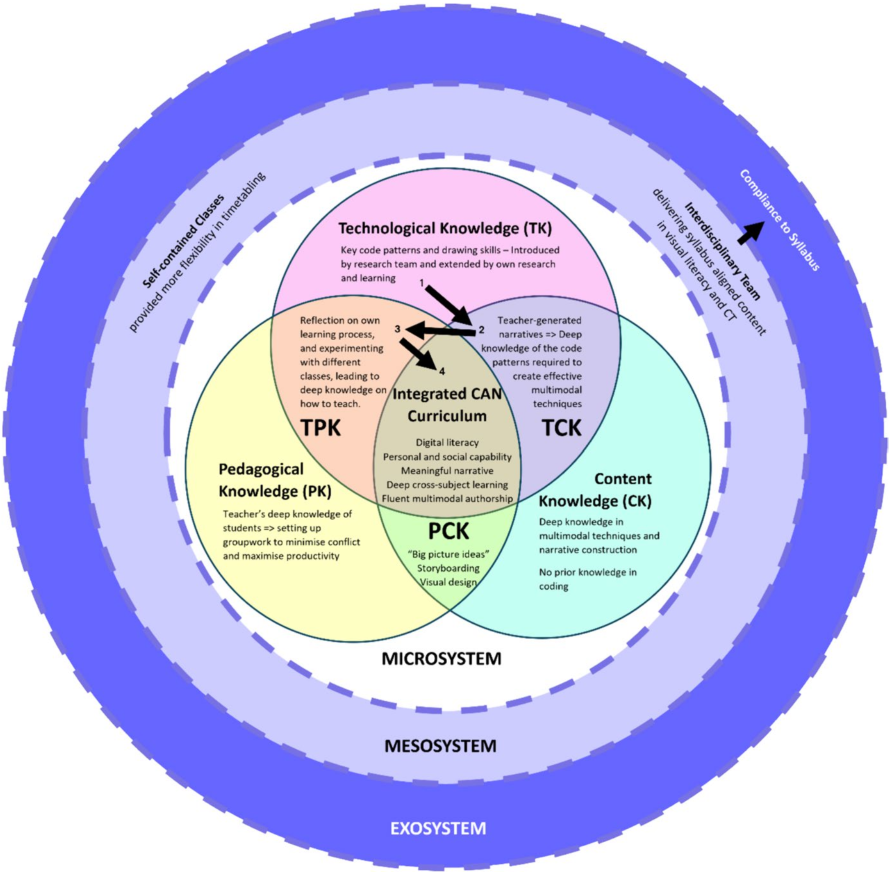

Fig. 9 Teacher's knowledge development and school context in integrated CAN curriculum (adapted from Koehler & Mishra, 2009)

PCK that enabled him to support students' development of general capabilities, cross-subject knowledge, as well as fluent multimodal authorship in an integrated CAN curriculum (4).

Mesosystem factors impacting on implementation of integrated coding curriculum

Finally, teaching an integrated curriculum involving coding (or not), requires a supportive school context. In this case, these principal supportive attributes are recorded in Fig. 9 as existing in the mesosystem. Vars (1991) noted three forms of school organisation to deliver an integrated curriculum—namely, a total staff approach, an interdisciplinary team approach, and making use of block-time and self-contained classes. In this study, the CAN program was supported by an interdisciplinary team approach in which other teachers in the team 'front-loaded' students with the necessary disciplinary knowledge in visual literacy and in coding. They also fulfilled the compliance mandates imposed by the state syllabus, so that those requirements were not necessarily imposed on the CAN unit. At the same time, Mr S.'s home class was an example of 'block-time and self-contained classes', which gave him maximum flexibility to work around the school timetable. Within this environment, students benefited from access to a well-planned and supported integrated coding curriculum, and were able to build their digital literacy and personal and social capabilities whilst developing fluency in multimodal authorship.

Limitations and further research

The limited generalisability of single case studies is acknowledged, and in this case, some of the success factors—such as a highly motivated and skilled teacher, staff arrangement and flexible timetable—may not be readily replicated in other schools. Furthermore, readers are cautioned against interpreting the findings of this study as evidence that integrated coding curriculum is a 'silver bullet' solution to technology teacher shortages. Instead, we have endeavoured to provide a detailed analysis of teacher and school environment factors supporting successful integrated coding curriculum, rather than presenting a 'model' applicable across all contexts. It is up to readers to assess the relevance of the study's findings and conclusions to their own contexts. To improve the study's trustworthiness, multiple data sources (interviews, classroom observations, student projects) have been used to strengthen its results, and member checking with the teacher participant has been conducted to minimise researcher bias.

It is anticipated that future iterations of the larger CAN project will further refine our understanding of school environment and pedagogical and curriculum design attributes essential to successful implementation of integrated coding curriculum (Barab & Squire, 2016). We also look forward to analysing any gender differences across different classrooms using data from the full project, as well as conducting fine-grained analysis of the screen recordings collected from the pairs in this class, to deepen our understanding of student collaboration and problem-solving processes.

Conclusion

This study introduced authoring coding animated narratives (CANs) as an innovative approach to integrating English-Digital Technologies curricula. Through integrating coding into English, students in our study developed digital literacy and personal and social capabilities, which are important general capabilities included in the Australian Curriculum. Furthermore, this study contributed to the literature on TPACK by demonstrating that TPACK is not a static quality of teachers, but may be developed 'on-the-job'. This is especially important when many schools rely on non-specialist teachers to teach coding curriculum. Furthermore, we applied Falloon et al.'s (2022) approach to using EST and Mishra's (2019) addition of XK to

TPACK, to provide a detailed analysis of the school contextual factors that can support non-specialist teachers' TPACK development.

However, despite the generally favourable outcomes this study reports, it also challenges prevailing assumptions about the benefits of integrating coding across the curriculum. First, for most schools, integrating coding is not a simple solution for meeting outcomes from Digital Technologies and English curricula, or addressing the pressures of an overcrowded curriculum. For students to reap the benefits of interdisciplinary learning and general capability development, it is likely more time will be needed than traditional single-subject curricula, because students need to draw on multiple new and existing disciplinary knowledges to inform their project. Given increasing moves towards standardisation and compliance, schools will likely find themselves under pressure to fulfil syllabus requirements *before* students can attempt integrated tasks of this nature. Second, integrating coding into another discipline is challenging for non-specialist teachers. It demands that non-specialist teachers be committed to learning coding and the associated CS concepts and practices, which, for most, is more than what other forms of ICT integration requires. In other words, coding is not simply a set of procedures or steps, but effective implementation requires understanding of CS discipline-specific knowledge. Moreover, teachers need to develop CS-specific pedagogical knowledge if they are to help students deliver quality outcomes. In Beane's (1996) words, 'curriculum integration is not for the faint-hearted or the marginally competent' (p. 9). Third, school organisation has an important role to play in supporting an integrated coding curriculum. A supportive interdisciplinary team is required, and teachers having their own (home) classes appears to support this type of curriculum, as it enables flexibility in implementation.

In conclusion, schools should be aware of the substantial investment of time required to nurture non-specialist teachers to become effective in teaching coding using integrated curriculum. It requires high levels of commitment from both the school and the teacher, if the intent is to teach coding as part of the regular school curriculum. The challenges of delivering an effective integrated coding curriculum are considerable, and the learning benefits may not be easily measurable by standardised descriptors. However, the learning that results if challenges such as those detailed in this study can be overcome, can be broadly-based and highly relevant to students' futures.

Appendix A: Student pre-intervention interview schedule

1. When did you start coding?
2. Do you like coding? What do you like about it?
3. What have you made with code? With Scratch? Other things? (Robots, Minecraft, Roblox, Makey Makey, Microbit)
   a. What motivated you to make it?
   b. What do you like most about that project?
   c. What was the hardest part? How did you deal with it?

   d. What do you do when you're stuck?
   e. Did you work with other people on it? Who would you talk to about coding? Do you like working with other people or prefer to work alone?

4. With Scratch, have you ever looked inside other people's projects to learn how they code?
5. Some people say coding makes you a better thinker. Do you think that's true? Why or why not? In what ways?
6. Some people say coding makes you more creative. Do you think that's true? Why or why not? In what ways?
7. Do you think these qualities carry over from coding to other areas? Can you give examples?
8. How important do you think it is to learn to code? Why?

Appendix B: Student post-intervention interview schedule for Unit 1

We'd like to look at your animation with you and ask you about the story and how you created it and then we'd like to ask you about how you managed the coding to create the characters and what happens in the story.

The students' coded animated narratives and the crafting of the story and the characters

1. Let's look at your animation. How would you describe your animation to someone who could not watch it?
2. What was it like for you making the animation?
3. What were you wanting to communicate in the story?
4. How did you try to show the viewers what you wanted them to understand about the story?
5. How do you use the animated images to show the ideas that are important in your story?
6. Tell me about how you composed the language parts of the animation.
7. What other ways did you create meaning in your story?
8. What do you think is the best (coolest) thing about your animated narrative?

The students' experience of coding and computational thinking

1. How did you manage the coding for your story?
2. What were the easy things to code and what things were difficult?
3. Were there some particular problems you had? How did you manage to deal with these? What kind of help were you able to get?
4. What is it like for you working with others to create your animation?
5. What would you like to be able to do with coding that you have not learned yet?
6. What would you have liked to put into your story that you were not able to do?

7. How did you feel about doing the work on this project?
8. What could we do to make this a better project for you?

Appendix C: Student post-intervention interview schedule for Unit 2

1. Between your Term 2 and Term 4 project, which one do you like better, why?
2. What has really made a difference between your Term 2 and Term 4 project?
3. What new knowledge or skills have you picked up for Term 4?

The students' coded animated narratives and the crafting of the story and the characters

4. Let's talk about your Term 4 animation, can you summarise the story for us?
5. What is the main message you're trying to convey?
6. Who is the main character? Tell us all the ways you have used to help us get to know this character. How do you want us to feel about this character? How did you manage to do that? (speech/thought/narration, attitudinal language; gesture, posture, movement; colour)
7. Tell us about the setting of your story. Can you tell us how tone and mood is conveyed in your story? (backdrops, music, sound effects)
8. Have you considered how you positioned the audience in designing the narrative? (camera angles)

The students' experience of coding and computational thinking

1. Were there some particular problems you had this time? How did you manage to deal with these? What kind of help were you able to get?
2. What is it like for you working with others to create your animation? How did that compare with Term 2?
3. What would you like to be able to do with coding that you have not learned yet?
4. What would you have liked to put into your story that you were not able to do?
5. What do you think is the best (coolest) thing about your animated narrative?
6. How did you feel about doing the work on this project?

General

1. How would you compare this with other tasks you've had in English? Writing a narrative/comic strip?
2. What have you been doing in Digital Technologies this year? Can you see any connection between what you're doing here and what you're doing in Digital Technologies?
3. Thinking a bit more broadly, beyond the particular work you did coding these stories, what is your feeling about this kind of school work for the things that are

important to you about learning for school and for learning more generally in your life?

4. How important do you think it is to learn to code? Why?

Appendix D: Teacher pre-intervention interview schedule

Demographics

- Name
- Gender
- Secondary/primary
- Teaching experience—years
- Teacher training—specialties/subject areas
- Other non-teaching degrees

The structure of narrative and use of Image to portray characters' feelings and attitudes

1. How would you describe the organisation or main structure of a narrative or story?
2. What different types of feelings or emotions might be shown through the visual and verbal depiction of characters in animated narratives?
3. If you were making judgments about the qualities, behaviour and attitudes of a character in an animation, what types or categories of attributes would you comment on?
4. If an animated character is expressing his/her feelings, how can different kinds of feelings be expressed by facial expression?
5. How can gestures of animated characters express different kinds of feelings?
6. What about posture and movement? How can they express different kinds of feelings?
7. What if one character were making judgments about another character's qualities, behaviour or attitudes? Can you suggest some particular kinds of judgements and how they might be expressed through one of more of the following:

   a. facial expression?
   b. gesture?
   c. posture and movement?

8. If you think of making an animated movie, how can you use different points of view or camera positions/camera angles to influence how the viewer relates interpersonally to the character(s) in the animation?
9. In making an animated movie, what would you consider in deciding what kinds of meanings to communicate through:

a. images,
b. language as dialogue or narration, and
c. images and language as dialogue or narration.

10. What teaching strategies can be used to support students in creating animated narratives using computational thinking, coding and multimodal authoring techniques?

Coding and computational thinking in the curriculum

1. What is your understanding of the sort of student knowledge and capabilities targeted in the Digital Technologies learning area of the Curriculum?
2. In what ways, and to what extent, do you consider the current digital technologies program in your school is successful in delivering on the goals of the curriculum?
3. The Australian Curriculum emphasises elements of computational thinking as integral to learning in Digital Technologies. What is your understanding of computational thinking and why it features so prominently in the curriculum?
4. Often coding is taught in schools as a means of delivering curriculum outcomes related to CT. Are you aware of any relationship existing between coding and computational thinking development, as indicated in the curriculum?
5. Do you think it is important for students to be taught to code?
6. Are you aware of any specific approaches to teaching coding—for example, learning design and pedagogy, that are particularly effective in delivering curriculum outcomes?

Appendix E: Teacher post-intervention interview schedule for Unit 1

1. Can you describe how you went about planning your CAN unit? What considerations influenced how you went about your planning?
   Ext: What planned outcomes were there for the students? What did you expect the students to learn from the unit? (knowledge, skills, competencies).
2. How did you decide your lesson sequence? (structure, systematic approach). Did you consider this effective? Why?
3. Observationally, each lesson appeared to follow a similar structure or format. Can you comment on the reasons for this?
4. While all groups developed their narratives based on a common theme, their outcomes were quite different. In terms of planning and teaching the unit (pedagogy), what sort of environment did you seek to establish in the classroom to encourage diverse outcomes, and what strategies did you adopt to achieve this?
5. Were there any ‘meso-environmental’ (school, logistical/organisational, resource, syllabus, assessment, policy etc) issues or challenges that influenced the trajectory of your CAN unit? Can you describe these and their effects?

6. If you were to comment from a teacher's perspective on 'what it takes' to successfully implement units like a CAN at stage 3 or 4, what would you comment or advise on?
7. Finally, thinking about the planned student outcomes from question 1, were these met? (How did you know?)
   Ext: Did other outcomes become apparent as the unit progressed? What were these, and how did they become apparent?

Appendix F: Teacher post-intervention interview schedule for Unit 2

1. TPACK
   a. What existing knowledge about coding, narratives or multimodal techniques did you bring into the CAN project? (Narrative, cinematic techniques, group-work, etc.?)
   b. Did you develop any new knowledge/s during the CAN project? What were these? (CK, PK, TK, TCK, etc.)?
   c. Was there anything surprising that you learnt as a result of the CAN project?
   d. Do you think there are any advantages or disadvantages to learning coding by integrating its use with other KLAs, such as English?
   e. When you were planning your CAN unit, did you place priority on meeting outcomes from the Digital Technologies or English curricula—of both more or less equally, or some other emphasis, such as Curriculum General Capabilities? Why?
   f. Can you tell us about the resources (e.g., Scratch tutorials) that you've used to deliver the CAN project. How did you select those resources? How helpful have they been?
   g. Can you tell us about any resources you created to support your CAN unit (storyboard templates, lesson plans, worksheets). How were they used, and what were the benefits or limitations?

2. Macrosystem
   a. Do you think learning to code will help the students in the future? How? (literacy/economic benefits)?
   b. Would you support a program like CAN being rolled out more broadly to other schools? Why or why not? (Equity)

3. Exosystem
   a. Has assessment and reporting requirements influenced how you designed and delivered this unit? How?
   b. Has the Australian Curriculum or NSW syllabus influenced how you structured or delivered this unit? (Curriculum and syllabus)

c. Are you aware of any authorities or central agencies (e.g. Department of Education), who may be interested in the outcomes this project? Why might they be interested?

4. Mesosystem

a. Have you had any feedback from parents or the community about the CAN project? What has that feedback been, and has this influenced the project in any way (How)?
b. Have you connected with any teachers in other schools or within your school about the CAN project (via networks etc.). What has been their response to what you are doing? Do you think they can see its value? Would you like to develop those connections further? If so, how?
c. How would you describe the response you have received from school leadership about the project and your participation in it? (leadership)
d. Do you think this project is compatible with your school's vision for the type of education it wants to provide for its students? How? (vision and commitment)
e. Has your and your students participation in the project been supported by any other staff at the school? (IT staff, colleagues). How?
f. Do you think any of your students' previous experiences with coding has influenced the project? (Peer-learning? Extra support for teachers? Intimidating?) (student expertise). How?

5. Microsystem

a. How would you compare the Term 2 program with the Term 4 program? Did you make any changes between the two iterations? What were these, and why did you make them?
b. What do you consider were the most important outcomes for your students from completing the CAN unit? Were there any differences between the outcomes in Term2 compared to Term 4?

6. Chronosystem

a. Do you see any longer term benefits or advantages for the students from undertaking units like this?

7. Overall

a. What has participating in this research project meant for you professionally, and for your school?

Author contributions Karen Woo: Conceptualization, Methodology, Data curation, Formal Analysis, Writing—Original draft preparation. Garry Falloon: Supervision, Writing—Reviewing and Editing.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. The research in this article was supported by an Australian Research Council Discovery Project Grant [DP190100228] and an Australian Government Research Training Program (RTP) Scholarship. The views expressed herein are those of the authors and are not necessarily those of the Australian Research Council.

Data availability Due to participant confidentiality, the involvement young children, and the presence in data of potentially identifying information including names and images of students, ethical permission was not granted for data to be openly shared.

Declarations

Conflict of interest There are no conflicts of interest involved in this research.

Ethical approval This research was approved by the Australian Catholic University Human Research Ethics Committee [2019-105H] and the New South Wales Education Research Application Process [SERAP #2019301].

References

ACARA. (2023). Australian curriculum—General capabilities (version 8.4). Retrieved 13 Jun 2023, from https://www.australiancurriculum.edu.au/f-10-curriculum/general-capabilities/

Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learninig, teaching and assessing—A revision of bloom's taxonomy of educational objectives*. Addison Wesley Longman Inc.

Anderson, T., & Shattuck, J. (2012). Design-based research: A decade of progress in education research? *Educational Researcher, 41*(1), 16–25. https://doi.org/10.3102/0013189x11428813

Arawjo, I., & Mogos, A. (2021). Intercultural computing education: Toward justice across difference. *ACM Transactions on Computing Education, 21*(4), Article 30. https://doi.org/10.1145/3458037

Woo, K., & Falloon G. (2023). The search for computer science concepts in coding animated narratives: Tensions and opportunities. *Journal of Educational Computing Research, 61*(7), 1335–1358. https://doi.org/10.1177/07356331231174929

Barab, S., & Squire, K. (2016). Design-based research: Putting a stake in the ground. *Design-based research* (pp. 1–14). Psychology Press.

Basu, S. (2019) Using rubrics integrating design and coding to assess middle school students' open-ended block-based programming projects. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education Minneapolis, MN*. https://doi.org/10.1145/3287324.3287412

Beane, J. A. (1995). Curriculum integration and the disciplines of knowledge. *The Phi Delta Kappan, 76*(8), 616–622.

Beane, J. A. (1996). On the shoulders of giants! The case for curriculum integration. *Middle School Journal, 28*(1), 6–11. https://doi.org/10.1080/00940771.1996.11496182

Blumenfeld, P. C., Marx, R. W., Soloway, E., & Krajcik, J. (1996). Learning with peers: From small group cooperation to collaborative communities. *Educational Researcher, 25*(8), 37–39. https://doi.org/10.3102/0013189x025008037

Bronfenbrenner, U. (2005). Ecological systems theory. In U. Bronfenbrenner (Ed.), *Making human beings human* (pp. 106–173). Sage.
Christie, F., & Derewianka, B. (2010). *School discourse: Learning to write across the years of schooling*. Bloomsbury Publishing.
Collie, R. J., & Martin, A. J. (2019). Motivation and engagement in learning. *Oxford research encyclopedia of education.* Oxford University Press.
Corbin, J., & Strauss, A. L. (2015). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (4th ed.). Sage.
Dede, C. (2010). Comparing frameworks for "21st Century skills". In: J. A. Bellanca & R. Brandt (Eds.), *21st century skills: Rethinking how students learn* (pp.51–76). Solution Tree Press. Retrieved from http://sttechnology.pbworks.com/f/Dede_(2010)_Comparing%20Frameworks%20for%2021st%20Century%20Skills.pdf
Drake, S. M., & Reid, J. L. (2018). Integrated curriculum as an effective way to teach 21st century capabilities. *Asia Pacific Journal of Educational Research, 1*(1), 31–50. https://doi.org/10.30777/apjer.2018.1.1.03
Education Council. (2019). *Alice Spring (Mparntwe) Education Declaration.* Retrieved 20 Feb, from https://docs.education.gov.au/system/files/doc/other/final_-_alice_springs_declaration_-_17_february_2020_security_removed.pdf
Falloon, G., Powling, M., Fraser, S., & Hatisaru, V. (2022). Shaping science, technology, engineering and mathematics curriculum in Australian schools: An ecological systems analysis. *Australian Journal of Education, 66*(2), 171–195. https://doi.org/10.1177/00049441221083347
Falloon, G. (2024). Advancing young students' computational thinking: An investigation of structured curriculum in early years primary schooling. *Computers & Education, 216*, 105045.
Griffin, P., & Care, E. (Eds.). (2015). *Assessment and teaching of 21st century skills*. Springer.
Koehler, M., & Mishra, P. (2009). What is technological pedagogical content knowledge (TPACK)? *Contemporary Issues in Technology and Teacher Education, 9*(1), 60–70.
Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K–12? *Computers in Human Behavior, 41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012
Mills, K. A., & Unsworth, L. (2018). iPad animations: Powerful multimodal practices for adolescent literacy and emotional language. *Journal of Adolescent & Adult Literacy, 61*(6), 609–620. https://doi.org/10.1002/jaal.717
Mishra, P. (2019). Considering contextual knowledge: The TPACK diagram gets an upgrade. *Journal of Digital Learning in Teacher Education, 35*(2), 76–78. https://doi.org/10.1080/21532974.2019.1588611
Resnick, M. (2017). *Lifelong Kindergarten*. MIT Press.
Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J. S., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60–67. https://doi.org/10.1145/1592761.1592779
Resnick, M., & Rusk, N. (2020). Coding at a crossroads. *Communications of the ACM, 63*(11), 120–127. https://doi.org/10.1145/3375546
Rich, P. J., Browning, S. F., Perkins, M., Shoop, T., Yoshikawa, E., & Belikov, O. M. (2019). Coding in K-8: International trends in teaching elementary/primary computing. *TechTrends, 63*, 311–329. https://doi.org/10.1007/s11528-018-0295-4
Rychen, D. S., & Salganik, L. H. (2001). *Defining and selecting key competencies*. Hogrefe & Huber.
Santos, J. M., & Castro, R. D. (2021). Technological pedagogical content knowledge (TPACK) in action: Application of learning in the classroom by pre-service teachers (PST). *Social Sciences & Humanities Open, 3*(1), 100110. https://doi.org/10.1016/j.ssaho.2021.100110
Shulman, L. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher., 15*(2), 4–14. https://doi.org/10.3102/0013189x015002004
Spieler, B. (2018). Reinforcing gender equality by analysing female teenagers' performances in coding activities: A lesson learned. *Proceedings of the 4th Conference on Gender & IT, Heilbronn, Germany*. https://doi.org/10.1145/3196839.3196871
Stake, R. E. (1995). *The art of case study research*. Sage.
Tahirsylaj, A., & Sundberg, D. (2020). The unfinished business of defining competences for 21st century curricula—A systematic research review. *Curriculum Perspectives, 40*(2), 131–145. https://doi.org/10.1007/s41297-020-00112-6

Unsworth, L. (2013). Point of view in picture books and animated movie adaptations. *Scan, 32*(1), 28–37. Retrieved from https://education.nsw.gov.au/content/dam/main-education/teaching-and-learning/professional-learning/scan/media/documents/archive-pdfs/vol31-40_2012-2021/Scan_2013_32_1_red.pdf

Vars, G. F. (1991). Integrated curriculum in historical perspective. *Educational Leadership, 49*(2), 14–15.

von Wangenheim, C. G., Alves, N. C., Rodrigues, P. E., & Hauck, J. C. (2017). Teaching computing in multidisciplinary way in social studies classes in school—A case study. *International Journal of Computer Science Education in Schools, 1*(2).

Weinert, F. E. (2001). Concept of competence: A conceptual clarification. In D. S. Rychen & L. H. Salganik (Eds.), *Defining and selecting key competencies* (pp. 45–66). Hogrefe & Huber.

Karen Woo is a Ph.D. candidate at Macquarie School of Education at Macquarie University. Karen is researching the development of computational thinking and twenty-first century competencies through coding. She is an experienced coding and robotics educator working with pre-service teachers and primary-age students. She also consults with schools on integration of technology in the classrooms.

Garry Falloon is Research Professor of Education in the Institute of Education, Arts and Community at Federation University (Mt. Helen) and Honorary Professor in the School of Education at Macquarie University (Sydney). Before joining Federation in March 2023, he was Professor of Education and Director of International Engagement in the Macquarie University School of Education in Sydney, NSW (2017–2023) and Associate Professor and Professor of Digital Learning in the Faculty of Education at Waikato University in Hamilton, New Zealand (2010–2017). His research interests are STEM-focused and include mobile learning, digital learning in primary and middle schools, online and blended learning, curriculum design, pedagogy and assessment in digitally-supported innovative learning environments, learning in primary science and technology, and educational research methods.

Authors and Affiliations

Karen Woo[1] · Garry Falloon[2]

✉ Karen Woo
karen.woo@mq.edu.au

Garry Falloon
g.falloon@federation.edu.au

[1] Macquarie School of Education, Macquarie University, Macquarie Park, Australia

[2] Institute of Education, Arts and Community, Federation University Australia, Mount Helen, Australia