ScratchThAI: A conversation-based learning support framework for computational thinking development

Kantinee Katchapakirin1 · Chutiporn Anutariya1 · Thepchai Supnithi2

Received: 8 July 2021 / Accepted: 14 December 2021 / Published online: 9 March 2022 © The Author(s) 2022

Abstract
Computational Thinking (CT) has been formally incorporated into the National Curriculum of Thailand since 2017, where Scratch, a block-based visual programming language, has been widely adopted as CT learning environment for primary-level students. However, conducting hands-on coding activities in a classroom has caused substantial challenges including mixed-ability students in the same class, high student-teacher ratio and learning-hour limitation. This research proposes and develops ScratchThAI as a conversation-based learning support framework for computational thinking development to support both students and teachers. More specifically, it provides learning experiences tailored to individual needs. Students can learn CT concepts and practice online coding anywhere, anytime. Moreover, through its ScratChatbot, students can ask for CT concept explanations, coding syntax or practice exercises. Additional exercises may be assigned to students based on the diagnosed individual learning difficulties in a particular topic to provide possible and timely intervention. Teachers can track learning progress and performance of the whole class as well as of individuals through the dashboard and can take suitable intervention within limited school hours. Deploying ScratchThAI to several Thai schools has enabled this research to investigate its effectiveness in a school setting. The obtained results indicated positive teacher satisfaction, better learning performance and higher student engagement. Thus, ScratchThAI contributes as a possible and practical solution to CT skill development and CT education improvement under the aforementioned challenges in Thailand.

Keywords Computational thinking · Conversational agent · Adaptive and personalized system · Automated assessment · Technology-enhanced learning · Scratch

Kantinee Katchapakirin
kantinee.katchapakirin@ait.ac.th

Extended author information available on the last page of the article
1 Introduction

Computational thinking (CT) is an essential skill that is increasingly important in today’s digital world and is closely related to digital competency and 21st century skills. Wing (2014) defined CT as “the thought process to formulate a problem, express solutions in a form that can be effectively carried out by an information-processing agent”. Besides, Wing (2017) emphasized that nowadays coding is equally essential compared to reading, writing, and arithmetic skills. Therefore, CT is a basis of computational problem-solving skills which incorporates computer and human to solve complex problems.

With its significance, many countries worldwide have introduced CT as part of their formal national education (Heintz et al. 2016). Students can learn, and practice CT through coding which is a new form of expression and a new context for learning (Grover et al. 2013; Djambong and Freiman 2016; Kwon 2019). Scratch is one of the most widely used coding environments in primary education (McGill et al. 2020). It is a visual block-based programming language, designed especially for learners between 8 to 16 years old, to learn fundamental computational concepts by connecting graphical blocks of computer codes in a manner similar to how they snap Lego bricks in the physical world. This mechanism can help minimize the cognitive load of remembering commands, syntax, and punctuation.

However, it is challenging to effectively conduct hands-on coding activities for many students in a regular classroom (Tissenbaum et al. 2018; Sentance and Csizmadia 2017). Students are individually different in terms of knowledge background, cognitive abilities, learning pace, and learning styles. Neglecting individual differences and providing similar learning experience for every student may not lead to desired outcomes (Cook et al. 2018). Hence, teachers are required to pay attention to individual progression, to diagnose individual strengths and weaknesses, and to provide assistance when needed.

Considering a traditional school setting, the following two challenging factors: a high student-teacher ratio, and a limited instructional time, have added extensive complexity for teachers to deal with an overwhelming range of student diversity. Assignments are a useful formative assessment instrument to determine for each student, the topics (s)he can master well and the topics (s)he struggles; hence helping teachers to early identify at-risk students. However, assignment grading is a time-consuming task which prevents teachers to frequently implement at scale (Alves et al. 2019). Regardless of available diagnosis information, precise intervention is hard to implement. As a result, at-risk students would get lost and lose learning interests, and thus leading to ineffective outcomes and educational waste (i.e., time, money, and resources that produces no change in outcomes) (Basawapatna et al. 2015).

This paper proposes ScratchThAI\(^1\), a conversation-based learning support framework for computational thinking development as the main contribution (Katchapakirin and Anutariya 2018). The framework not only supports teachers to conduct hands-on coding activities in the classroom, but also assists students

\(^{1}\) https://www.scratchthai.com
for outside classroom practices. It adds a virtual conversational chatbot, automated assessment, and adaptive features to the existing learning environment, i.e., Scratch coding website\(^2\). The chatbot motivates students to further practice CT problems, and provides timely personalized learning assistance. Automated assessment gives instant feedback to adjust their understanding. Adaptive instruction (i.e., exercises and learning materials) is suggested to intervene and fill the student’s knowledge gap. With a cloud-based design, teachers can monitor through a dashboard how satisfactorily the whole class and each individual student progresses both in-class learning and outside classroom practices. Furthermore, the framework observes student-system interaction during a practical session which is hard to accomplish in practice (Katchapakirin and Anutariya 2019). Coding progression, learning preferences, learning behaviors, and learning activities are maintained as learner profile for teachers to manage and provide precise intervention.

Based on the developed ScratchThAI framework, the paper also presents a pilot study which explores the framework’s effectiveness in enhancing student outcomes, as well as in supporting teachers to improve their instruction and to better facilitate students.

The paper is organized as follows. Section 2 discusses important challenges faced in CT education worldwide, and in Thailand, reviews existing works and identifies the research gaps, Section 3 elaborates ScratchThAI framework, Section 4 presents its pilot study in a primary school, and Section 5 concludes.

2 Literature review

2.1 Challenges in CT education worldwide and in Thailand

Introducing CT as part of national basic curriculum in many countries worldwide has imposed various challenges, which may differ from country to country depending on their education systems, infrastructure, available resources and supporting systems, as well as teacher and student related factors, etc. Table 1 summarizes important challenges, categorized into the following three aspects: teachers, students, and environment, and faced by different countries.

In Thailand, CT was introduced as part of the national standard curriculum in 2017. Katchapakirin and Anutariya (2019) identified important challenges encountered by various Thai primary schools after the implementation of 3-year roll-out of the first Thailand Computing Science Curriculum in 2018. The research confirmed that the following challenges related to teachers, students and environment were priority ones: (1) teachers’ lacking CT fundamental knowledge and confidence in teaching CT, (2) mixed-ability class in terms of computing literacy, mathematical and/or problem-solving skills as well as learning and achievement gaps, and (3) infrastructure problem and limited school hours as compared to the curriculum’s expected outcomes.

\(^2\) https://scratch.mit.edu
| Teacher's challenge                                      | Description                                                                                                                                                                                                 | Country/ Region           |
|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Limited Fundamental and Pedagogical Knowledge          | Lack of understanding in digital literacy, programming, and CT; Limited teaching strategies to engage students (Saidin et al. 2021; Sentance and Csizmadia 2017; Ribeiro et al. 2013; Lockwood and Mooney 2017; Mouza et al. 2017; Yadav et al. 2014; Black et al. 2013; Schulte et al. 2012). | Europe, Brazil, USA       |
| Lack of Experience                                     | Lack of experience in teaching the subject (Sentence and Csizmadia 2017; Lockwood and Mooney 2017; Heintz et al. 2016).                                                                                           | Europe                    |
| Lack of Confidence                                     | Lack of confidence in current CT knowledge (Saidin et al. 2021; Sentance and Csizmadia 2017; Lockwood and Mooney 2017; Bower and Falkner 2015).                                                                  | Australia, UK, Ireland    |
| Lack of Time Management                                | Limited preparation time to refine lesson plans and to develop quality materials (Cho et al. 2014; Sentance and Csizmadia 2017).                                                                             | UK, USA, Uganda           |
| Less Popularity of ICT Teachers                        | Lack or inadequate ICT teachers which lead to high student-teacher ratio and large class-size.                                                                                                              | Europe, USA, Uganda, Turkey|
| Low Quality of Teaching Degrees for CS Educators       | Existing undergraduate teaching degrees for ICT teacher careers mostly focusing on teaching how to use computers, but not CT development skills (Ribeiro et al. 2013).                 | Brazil                    |

| Student’s challenge                                    | Description                                                                                                                                                                                                 | Country/ Region           |
|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Mixed-ability students in the same class               | Students with different ability in mathematics, computing literacy, and/or problem-solving skills. Basu et al. (2016); Sentance and Csizmadia (2017); Grover et al. (2015); Schulte et al. (2012) | Europe, USA               |
| Low Engagement, Motivation, and Low Students' Resilience | Students having low interest and engagement in classrooms. Not able to develop their mastery after class; Giving up easily when code not working. Saidin et al. (2021); Hooshayar et al. (2021); Grover et al. (2019); Sentance and Csizmadia (2017); Mooney et al. (2017). | Ireland, USA, UK          |

| Environment-related challenge                          | Description                                                                                                                                                                                                 | Country/ Region           |
|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Material                                               | Lack of good quality material and lack of material to contextualize CT with other disciplines (Sentance and Csizmadia 2017; Heintz et al. 2016; Schulte et al. 2012).                                         | Europe, Australia, New Zealand, South Korea |
| Curriculum                                             | Not enough school hours to achieve the expected learning outcomes of the curriculum; No detailed guidance on CT assessment provided in the curriculum. Sentance and Csizmadia (2017); Van Gorp and Grissom (2001); Román-González (2017); Tissenbaum et al. (2018); Schulte et al. (2012). | Europe, USA               |
| Infrastructure                                         | Lack of IT resources (e.g. hardware, software, and networks), and IT supports (Sentance and Csizmadia 2017; Ribeiro et al. 2013; Carvalho et al. 2013; Lockwood and Mooney 2017).                                                | Ireland, Brazil, UK       |
In this study, we focus on supporting CT education in young learners under these three challenges: P1: Mixed-ability students in the same class, P2: High student-teacher ratio and P3: Learning-hour limitation.

2.2 Existing work in teaching and learning support system for CT development

This section reviews recent personalized and adaptive systems. Tikva and Tambouris (2021) provided a broad overview of teaching and learning support systems for CT development through programming in K-12 education. AutoThinking (Hooshyar et al. 2019; Hooshyar et al. 2021) is an adaptive CT game to instill primary and secondary students with three important CT concepts: sequence, conditional, and loop and CT skills. Adaptivity within a gameplay includes adaptive hints, feedback, and tutorials. CTSiM (Basu et al. 2014, 2016) contextualizes CT with Science and Ecology for middle school students. It embeds a virtual agent to provide feedback to struggling students, and keeps them motivated through predefined agent-student conversations. AgentSheets and AgentCubes (Repenning et al. 2010) introduces an adaptive scaffolding framework with CT pattern graphs. The framework can track, interpret students’ computational artifacts in games and science simulations and present in knowledge gaps using CT pattern graph. Flowchart-based Intelligent Tutoring System (FITS) (Hooshyar et al. 2015) is an adaptive intelligent tutoring system that utilizes flowchart development to help students conceptualize their solution design. The system provides adaptive guidance by dynamically defining Sub-flowcharts for scaffolding low-performance students. Although the system can facilitate conceptualization, it cannot visualize the result of the designed solutions. REACT (Koh et al. 2014) is a Cyberlearning tool for computer science education with a teacher dashboard that visualized learning progression of students in real-time, allowing teachers to give support to students who significantly need help. The system merely monitors students learning and informs the teachers of the students who are in trouble but does not give feedback directly to the students.

Although various existing research reviewed above can provide personalized and adaptive features to assist learners in CT skill development, Sentance and Csizmadia (2017) found that a computing course still had the largest achievement gaps among students when compared to other subjects. In addition, Montiel et al. (2021) reported a present need for frameworks to support teaching CT in the classroom. That is, there was a lack of the following important capabilities in existing teaching-learning support tools to effectively assist a teacher to conduct hands-on coding activities under the three important challenges in a regular classroom: (i) learning progression tracking, (ii) diagnosis and early identification of at-risk students with slow learning, low performance, or disengagement, and (iii) collection and analysis of learning activities and learning behavior for both in-class and outside-school-hours.

This study aims to fill such gaps by employment of technology-enhanced learning to deal with CT development to enhance student learning and teacher teaching quality. Table 2 compares existing research and the proposed research based on the supporting CT skill development.
| Research/Learning System | Key System | Diagnosis | Intervention | Learning System | Context | Feature | Performance | Diagnosis | Criteria | Techniques | Based on | Framework |
|--------------------------|------------|-----------|--------------|-----------------|---------|---------|-------------|-----------|----------|------------|----------|-----------|
| Laboratory/Classroom     | Personlaized | Learning | Performance | Programs/CT | Fixed | Teacher-based | Learning | Adaptive | Adaptive | Learning | Knowledge | Student | Kid-friendly | Customizable | Integratable |
| Support School and Adaptive Learning | Activity | Tracking | Codes | Thought | Criteria defined | Material | Learning | Feedback | Style | Level | Preference | Conversational | Theme | Other |
| ScratchThAI (Proposed research) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | * | ✓ | * | ✓ | ✓ | ✓ | * |
| AutoThinking             | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Hooshayar et al. (2021)  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CTSiM                    | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Basu et al. (2016)       | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FITS                     | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Hooshayar et al. (2015)  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| AgentSheets              | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Repenning et al. (2010)  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| REACT                    | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Research/ Learning System | Key System Feature | Diagnosis Diagnosed Objects | Based on Framework Learning Techniques Used | Based on Framework Learning Criteria Used |
|---------------------------|-------------------|-----------------------------|------------------------------------------|------------------------------------------|
|                           | Laboratory/ Classroom | After- Personalized Learning Performance |                           |                           |
|                           | Experiment Support School and Adaptive Activity Tracking | Codes Thought Criteria defined Material Learning Feedback Style Level Preference |                           |                           |
|                           | Support Features Analysis & Visualization | Process Criteria Support Path User Interface |                           |                           |

Koh et al. (2014)

Remark: ✓ means the feature is supported by the research/system; * means the feature is part of future work.
3 Proposed research

This section presents the research questions, the research design overview and the design and development of the proposed framework, ScratchThAI, which employs innovative technology-enhanced learning (TEL) as well as validated educational approaches to promote computational thinking development for young learners under the complex Thai school context influenced by the three important challenges: P1 (mixed-ability students in the same class), P2 (high student-teacher ratio) and P3 (learning-hour limitation).

The paper formulates the following research questions:

\[ R1: \] How can we construct a TEL-based framework for CT skill development in young learners and dealing with the three important challenges?

\[ R2: \] How effectively can the proposed framework improve learning performance and enhance teaching-learning activities?

\[ R3: \] What are the students’ and teachers’ perceptions toward the proposed framework’s usage experience regarding the learning motivation and engagement as well as the learning flexibility dimensions?

3.1 Research design overview

Based on the defined research questions R1-R3, this section proposes ScratchThAI framework and elaborates its design overview using influenced educational approaches and informed-empirical studies as a theoretical basis. Fig. 1 illustrates for each CT challenge, the adopted design strategies & TELs, the implemented ScratchThAI modules and the evaluation criteria.

Concerning the challenge P1 (mixed-ability students), ScratchThAI employs the learner-centered approach and personalized learning approach to address the challenge where students are able to learn at their own knowledge level, pace and style (Bjork and Bjork 2011; Deunk et al. 2018; OECD 2012; Schleicher 2016; Peng et al. 2019). Personalized & Adaptive Technology (T1), and Automatic CT Assessment (T2) are the key enabling technologies that can help improve students’ progress and engagement (Campos et al. 2012; Hattie and Timperley 2007; Marwan et al. 2020).

To overcome the challenge P2 (high student-teacher ratio), important strategies include: (i) Tracking of at-risk students, (ii) Provision of assistance to teachers, and (iii) Provision of instructional support to students. The enabling TELs are T1, T2, T3 and T4 where Conversational Agent Technology (Chatbot) (T3) can be adopted as a personal teaching assistant and Cloud-based & Web Technology (T4) can improve the system usability and availability for both teachers and students (Hailikari et al. 2008; Ssharratt 2017).

The challenge P3 (learning-hour limitation) is overcome by the provision of flexible learning in time and place strategy. With Internet access, the Cloud-based
and Web technology (T4) allows ubiquitous learning, and enable teachers to monitor students’ progress both in and outside the classroom practices.

The proposed ScratchThAI framework is designed and constructed to provide learning experiences tailored to individual’s needs. Students can learn CT concepts and practice online coding anywhere, anytime. Teachers can track learning progress and performance of the whole class as well as of individuals through the dashboard and can design suitable intervention within limited school hours. Details are described in Section 3.2.

Figure 1 further denotes that to address the research questions R2-R3, the research performs both quantitative and qualitative analysis with the following defined evaluation criteria: (1) Learning Performance Improvement, by determining pretest posttest scores, (2) Teaching and Learning Enhancement by determining the number of students who can complete all assigned exercises as well as the average learning time and teaching time to complete each exercise, (3) Student
Satisfaction and Perception and (4) Teacher Satisfaction and Perception. Details are elaborated in Section 4.2.

3.2 System use cases and architecture

Based on the design overview elaborated earlier, ScratchThAI framework is constructed. Figure 2 illustrates the system architecture of ScratchThAI, which supports the following four student use cases: U1: Request Personalized Practices, which assigns exercises based on the current student’s performance and the teacher’s predefined lesson plan, U2: Request Learning Material, which prompts a list of learning materials such as Scratch block information and CT concepts, U3: Practice and Submit Code, which allows a student to perform a coding practice, submit the code and obtain the automatic grading and feedback, and U4: View Student Progression and Performance, which displays the student’s individual progression and performance report. Furthermore, it supports two teacher use cases: U5: View...
Dashboard, for teachers to display the whole class progression and learning status, and hence enabling them to identify and monitor potential at-risk students easily. U6: View Report, to support teachers to view learning progression and performance of each individual student in details.

The system architecture organizes the six designed modules of ScratchThAI into three layers: User Interface, Precision Education, and Data Layers, and decomposes into components denoted by rectangle shape. Components with blue solid line represents the completely developed software. While gray solid line components represent utilized existing tools i.e., MIT Scratch, Chatbot Engine. The components denoted by a dashed line border means future components which are parts of future work.

3.2.1 User interface layer

This layer consists of two modules: (i) ScratChatbot & CT Practice, and (ii) Visualization Modules. The ScratChatbot & CT Practice Module comprises four components: ScratChatbot, Chatbot Engine, Connector, and and CT Practice Environment i.e., MIT Scratch Environment in this study. The ScratChatbot component, designed as a browser extension (aka. add-on or plugin), allows seamless integration with the MIT scratch environment. Figure 3 shows the ScratchThAI extension, circled by green color, the ScratChatbot UI for the student use cases U1 – U3 and MIT Scratch Environment. The bot-student conversations are in a kid-friendly manner through simple language, avatars, emoticons, and images to attract primary-level students. To send a message, students can simply select one of the dynamic buttons/messages suggested based on the current context. In this layer, the Chatbot Engine component identifies students’ intentions, extracts relevant information. The Connector component provides interoperability with the other modules in the Precision Education Layer.

The Visualization Module consists of Dashboard and Report components to supports the teacher use cases U5: View Dashboard and U6: View Report. As shown in Fig. 4, the dashboard presents the real-time class progression for a teacher to identify potential at-risk students having slow progress or low performance. To be specific, the bar graph shows the whole class progression with respect to the number of practice exercises completed by each student and grouped by the exercise difficulty levels (Level 0 to Level 5). The teacher could, thus, notice that the students st110005 and st140027 progressed slwlier than peers, i.e., completing only 2 exercises of Level 0. The coding practice logs then show in detail, for each individual student, the in-progress exercise, practice history, elapsed time, and start time; allowing the teacher to further inspect that the students had difficulty in EX105 exercise and had been practicing for 20 minutes. Thus, the teacher could provide a timely assistance and a closer monitor on the students’ learning performance. Figure 5. shows learning performance reports of specific students with respect to their CT concept understandings. From such reports, the teacher could see that the students st140040 and st140048 had limited understanding in the loop concept, while the student st140048 also struggled with the condition concept.
3.2.2 Precision education layer

This layer has three modules: (i) Diagnosis, (ii) Treatment & Prevention, and (ii) Learning Insight & Analytics. The first module, Diagnosis, focuses on automatically diagnosing students’ CT skill development during the learning to provide precise and reliable information for adaptive instruction and possible intervention. It has two components: Programming Assessment and Problem-Solving Assessment. The first component automatically checks and grades a student’s
programming code which corresponds to particular CT concepts by evaluating the code against a teacher’s predefined correct coding specification. The specification can be described using a combination of correct coding patterns, must-have or must-not-have blocks, and a maximum number of blocks allowed in the students’ code. The output score is computed as a weighted summation of the coding similarity score, which uses Jaccard’s similarity coefficient, together with the coding satisfaction score. The Problem-Solving Assessment is a future component which evaluates the student’s CT thought process, comprising: problem decomposition, abstraction, pattern recognition, and algorithmic thinking for a more precise and comprehensive diagnosis.

Fig. 4 Teacher dashboard: real-time class progression and practice history (Use Case U5)
The Treatment & Prevention Module comprises two components: Personalized Assignment and Material Support to support an adaptive learning path for students. The first component suggests an appropriate personalized practice for a student, based on the teacher’s lesson plan and the student’s learning status and progression which includes his/her understanding level/scores for each CT concept, measured by the Diagnosis Module, described earlier. The other component, Material Support, facilitates the learning of challenging topics by allowing students to ask ScratChatbot for learning materials related to specific CT concepts or Scratch coding blocks. The materials are available in various formats for students to choose according to their preferences including textual explanations, infographics, coding examples, and video clips.

The Learning Insight & Analytics Module comprises two components: Collector and Analyzer. The former is responsible for logging all learning activities and related data for a data-driven approach, which are maintained in the Data and Knowledge Layer and can be categorized into (1) Coding practice logs: source code evolution, the number of runs, edit distance from run by run, and time spent on each exercise; (2) Learning activity and behavior logs: practice timestamps, frequency of practices, materials learned, time spent on each material, and any web addresses students open while practicing; (3) ScratChatbot conversation logs. The Analyzer component is responsible for in-depth analyzing the collected logs using learning analytics and educational data mining techniques to gain more insight into individual variability in learning behavior, learning style, and learning preference of each student. Hence, proper intervention and prevention can be provided as the aims of the precision education approach. Note that, this Analyzer component is part of future work.

3.2.3 Data and knowledge layer

This layer maintains Data Store and Knowledge Base for the system and comprises the following:

Fig. 5 Individual student learning performance reports (Use Case U6)
A relational database and a document store are used for this layer. In the future, we also plan to incorporate ontology and knowledge graph to formally model the system’s knowledge base.

3.3 System implementation

For this section, we describe the implementation of ScratchThAI system architecture in this study. Table 3 summarizes techniques, programming languages, tools, and production detail for the components in the system architecture.

4 ScratchThAICamp: Adopting and evaluating ScratchThAI in computing science classroom

ScratchThAICamp, a 2-day scratch coding camp, was organized to perform a pilot study of adopting ScratchThAI system as part of the Grade 5’s computing science course in a provincial primary school in Thailand. The study explored the effectiveness and practicability in applying ScratchThAI to promoting student engagement, improving learning performance, and to enhancing teaching-learning activities. In particular, the study was conducted to answer the defined research questions R2 and R3, using the defined evaluation criteria as introduced in the Section 3.1.

4.1 Lesson plan, material and assessment design

The camp was designed to introduce CT concepts and coding experience using Scratch. It consisted of four important CT lessons: sequence, event handling, conditional statement and looping. For the designed lessons, fundamental knowledge of the four CT concepts, basic block programming concepts as well as related mathematics knowledge, e.g., geometry coordinate and angle, were developed and provided as supporting learning materials in various formats, e.g., textual explanations, infographics, and video clips. In addition, practice exercises were designed with six difficulty levels (Level 0 to Level 5) and were defined as either mandatory or optional. An optional exercise would be assigned as an extra practice by the ScratChatbot to the students who could not master certain concepts and are required to
| Modules                      | Components              | Implementation Details                                                                                                                                                                                                 |
|------------------------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| **User Interface Layer**     |                          |                                                                                                                                                                                                                       |
| ScratChatbot                 | ScratChatbot UI          | Consisting of the Client-sided UI and the Server-sided UI:                                                                                                                                                           |
|                              |                          | – The Client-sided UI is an extension UI pages in Chrome Extension platform, containing HTML and JavaScript.                                                                                                           |
|                              |                          | – The Server-sided UI is developed as a REST API for message translation between the Client-sided UI and the Chatbot Engine.                                                                                             |
| Chatbot Engine               |                          | Utilizing Google Dialogflow for intent classification.                                                                                                                                                                |
| Connector                    |                          | Developed using the REST API architecture as a Webhook in Dialogflow to pass intent information to corresponding components in the Precision Education Layer.                                                           |
| CT Practice                  | MIT Scratch              | Utilizing the existing coding environment, i.e., MIT Scratch.                                                                                                                                                           |
| Visualization                | Dashboard                | Developed as a Web component using PHP programming language.                                                                                                                                                         |
| Report                       |                          |                                                                                                                                                                                                                       |
| **Precision Education Layer**|                          |                                                                                                                                                                                                                       |
| Diagnosis                    | Programming Assessment   | Developed as REST APIs using Python and Node.JS programming languages to communicate with the Connector component via HTTP POST with JSON encoded messages.                                                            |
| Treatment & Prevention       | Personalized Assignment  |                                                                                                                                                                                                                       |
|                              | Material Support         |                                                                                                                                                                                                                       |
| Learning Insight & Analytics | Collector               | Developed using JavaScript as background script of the Chrome Extension.                                                                                                                                               |
| **Data & Knowledge Layer**   |                          |                                                                                                                                                                                                                       |
| Data Store & Knowledge Base  | Data Store               | Developed using relational and document database technology: MongoDB and PostgreSQL.                                                                                                                                     |
repeat the learning. This personalized alternative exercise was determined and suggested by ScratchThAI’s adaptive feature. In addition, students could interact with the ScratChatbot to get helps and request for materials or explanations related to unclear CT concepts at any time. As a result, students could learn at different paths and paces depending on their prior knowledge and abilities to learn new concepts.

To measure a student’s prior knowledge, a pretest was utilized and comprised 20 multiple-choice questions. A student could choose “I do not know” or leave blank if they did not know the answer. To assess the learning outcomes, a posttest was taken as a summative assessment. The posttest was identical to the pretest for reliability and validity.

Note that all learning materials, exercises and assessment specifications for this pilot study were co-developed and validated by ten computing science teachers from three primary schools in Thailand.

4.2 Evaluation design

The camp, was organized in June 2020 at a provincial primary school in Thailand with a face-to-face learning experience. A total of 45 5th-graded students participated in this camp and was divided into two groups: an experimental group and a control group. The former practiced their CT skills with ScratchThAI, while the latter learned in an ordinary environment. Figure 6 illustrates that the camp was
designed with eight learning activities: pretest, tutorials, coding practices, posttest and interviews with students and teachers.

To answer the defined research questions R2 and R3 regarding the three important dimensions: learning performance improvement, teaching and learning enhancement, and student/teacher perceptions toward their ScratchThAI usage experience, corresponding quantitative and qualitative research methods were employed as elaborated below.

For performance improvement analysis, a quasi-experimental research with non-equivalent control group pretest-posttest design was adopted. It compared the performance improvements of both within groups and between groups. The score differences between the pretest and posttest within group could identify how scores changed from pretest to posttest. On the other hand, the analysis of gain scores could show the effectiveness of practicing with ScratchThAI. Hence, the hypothesis statements for the study are:

- $H_{S1}$: The mean for posttest scores is greater than the mean for pretest scores in the experimental group.
- $H_{S2}$: The mean for posttest scores is greater than the mean for pretest scores in the control group.
- $H_{S3}$: The mean of gain scores for the experimental group is greater than the mean of gain scores for the control group.

To analyze how ScratchThAI can better enhance and support teaching and learning, the study compared interesting descriptive analytic between students of the two groups on their learning behavior, number of practices and time taken to accomplish an exercise, extra personalized learning support needed.

Lastly, to perform an initial investigation on the student/teacher satisfaction and perceptions toward the adoption of ScratchThAI, two qualitative techniques were utilized. First, a focus group discussion with the students of the experimental group was conducted to enable students to share their opinions or on top of their peers’ views on the following predefined topics: (1) Learning Motivation and Engagement, (2) Learning Flexibility, Efficiency and Effectiveness, and (3) Additional Features Suggestions. Second, a semi-structured interview was performed to elicit the perspectives from the teachers who observed the entire learning activities of both groups related to the following points: (1) Learning Behaviors, Motivation and Engagement, (2) Learning Flexibility and Learning Improvement, (3) ScratchThAI Adoption by Teachers, and (4) Limitations and Suggestions.

### 4.3 Results

#### 4.3.1 Learning performance improvement

Figure 7 illustrates the distribution of the pretest, posttest and gain scores of the experimental group and the control group. Table 4 summarizes key descriptive
statistics, the resulting normality data distribution tests, and the obtained t-Tests using the significance level of 0.05, which can be interpreted as follows:

- For HS₁, the mean of the posttest scores of the experimental group was greater than the mean of the pretest score (p-value = 0.00 < 𝛼).
- For HS₂, the mean of the posttest scores of the control group was not significantly higher than the mean of the pretest scores (p-value = 0.06 > 𝛼).
- For HS₃, the mean of the gain scores of the experimental group was significantly higher than the mean of the gain scores of the control group (p-value = 0.03 < 𝛼).

Hence, from the pilot study, it was found that practices with ScratchThAI could lead to learning performance improvement in computational thinking ability (HS₁, HS₃) and impact on CT education (effect size= 0.8).
Based on the learning logs collected and maintained automatically by ScratchThAI for students of the Experimental Group as well as the data recorded manually for the Control Group, Table 5 compares interesting descriptive statistics between the two groups with regard to their learning behavior, practice accomplishment, personalized learning support needed. The table shows that there were the total of 10 mandatory and 4 optional exercises (2 of which were challenging ones) as part of the designed lesson plan with 6 difficulty levels covering the important 4 CT topics. Important findings drawn follow. For the control group, the learning activities were led by a teacher. That is, based on the designed lesson plan, the teacher

### Table 4 Learning performance improvement: hypothesis testing

| Statistical Measurement | Hypothesis 1 | Hypothesis 2 | Hypothesis 3 |
|-------------------------|--------------|--------------|--------------|
|                        | Experimental | Control Group | Experimental | Control Group’s Group’s Gain |
|                         | Pretest | Posttest | Pretest | Posttest | Gain |
| Mean                    | 5.86 | 8.95 | 5.24 | 6.52 | 3.09 | 1.29 |
| Median                  | 5.5 | 10 | 5 | 7 | 3 | 2 |
| Mode                    | 5 | 10 | 5 | 7 | 1 | 2 |
| Standard Deviation      | 2.4 | 2.54 | 2.62 | 0.62 | 0.02 | 3.0 |
| Variance                | 5.74 | 6.43 | 6.89 | 8.16 | 4.08 | 9.01 |
| Observation(n)          | 22 | 22 | 21 | 21 | 22 | 21 |
| Data Normality Test     |            |              |              |              |              |
| Chi-square p-Value      | 0.99 | 0.83 | 0.93 | 0.11 | 0.95 | 0.35 |
| Data distribution       | Normal | Normal | Normal | Normal | Normal | Normal |
| t-Test                  | Paired t-Test | Paired t-Test | Unpaired t-Test |
| Pearson Correlation     | 0.67 | 0.40 | – |
| Hypothesized Mean Difference | 0.00 | 0.00 | – |
| Degrees of freedom      | 21 | 20 | 35 |
| p-value                 | 0.00 | 0.06 | 0.03 |
| Hypothesis              | Reject $H_0$ | Fail to reject $H_0$ | Reject $H_0$ |
|                         | Accept $H_1$ | Not accept $H_1$ | Accept $H_1$ |
| Statistical difference  | Significant | Not Significant | Significant |

**Remark:** The criterion for acceptance $\alpha = 0.05$ (p-value < 0.05); The total scores of pretest and posttest were 20; Gain = Posttest Score - Pretest Score.
Table 5  Teaching-learning enhancement: descriptive statistics

| Practice Exercises | Experimental Group (n=22) | Control Group (n=21) |
|--------------------|---------------------------|----------------------|
|                    | Passing \* | Repeating \* | Accomplished \* | Average Time to complete (mins) | Accomplished \* | Time to complete (mins) |
| Level | Key CT Topic | Exercise ID | | | | |
| 0 | Sequence | EX101 | 16 | 6 | 22 | 12 | 21 | 16 |
| | | EX102 | 19 | 3 | 22 | 28 | 21 | 26 |
| 1 | Sequence | EX103 | 11 | 10 | 21 | 16 | 21 | 24 |
| | | EX104 | 18 | 3 | 21 | 25 | 19 | 37 |
| | | EX105 | 18 | 3 | 21 | 20 | Skip | Skip |
| | | EX106 (Optional) | 11 | 0 | 11 | 36 | Skip | Skip |
| 2 | Event Handling | EX107 | 12 | 7 | 19 | 21 | 19 | 62 |
| 3 | Condition | EX108 | 15 | 4 | 19 | 17 | 18 | 50 |
| | | EX109 | 13 | 6 | 19 | 21 | Skip | Skip |
| | | EX110 | 13 | 6 | 19 | 14 | Skip | Skip |
| 4 | Loop | EX111 | 12 | 5 | 17 | 21 | 18 | 92 |
| | | EX112 (Optional) | 8 | 0 | 8 | 18 | Skip | Skip |
| 5 | Loop | EX113 (Challenge) | 11 | 4 | 15 | 7 | 17 | 105 |
| | | EX114 (Challenge) | 2 | 11 | 13 | 12 | 5 | 75 |

*Remark*: \*Passing means the number of students whose codes were correct in the first submission; \*Repeating means the number of students who could not attempt the exercise and were assigned as an extra practice by the ScratChatbot; \*Accomplished means the number of students who could accomplish the exercise (in any number of submissions)
selected practice exercises and assigned to the whole class using the same teaching-learning pace. Thus, fast-learning and average-learning students must wait for some slow-learning students to finish their works, and then the whole class would move together to the next topic/exercise. In addition, the teacher had to grade the student submissions manually to check their understanding of the learning topics.

On a contrary, students in the experimental group had more flexibility in learning at different pace and did not need to wait for their peers to accomplish the practices. Therefore, fast-learning students could progress their practices more efficiently, while slow-learning students could get help from the teacher directly or from the system (in term of extra practice or extra learning materials). In addition, based on the automatic code checking and grading functionality of the system, the learning of the whole class progressed much faster, since it did not need to wait for the teacher to manually check the correctness of each student’s code. That is, the table shows that for the topics of the level 1 and above, students of the control group took much longer time to accomplish than the experimental group. Furthermore, the teacher of the control group had to skip some mandatory exercises, e.g., sequence/Ex104, condition/Ex109/Ex110 due to slow class progression.

The number of exercises practiced by students of the experimental group (control group) were: 10 (resp. 7) mandatory, 2 (resp. 0) optional, and 2 (resp. 2) challenge exercises. In total, the number of students from the experimental group who could successfully accomplish all 10 mandatory exercises and the two challenge ones was 13 (dropout rate 41%), whereas that of the control group was 5 (dropout rate 76%).

4.3.3 Student satisfaction and perceptions

The students of the experimental group participated in a focus group discussion to share and discuss their satisfactions and perceptions regarding their experience in learning and practicing with ScratchThAI. Important findings are threefold:

1. Learning Motivation and Engagement: With ScratchThAI, the students were motivated, inspired, excited and had less pressure to obtain feedback and to get their exercises graded by ScratChatbot. Most students preferred to learn with ScratchThAI over the traditional style as well as to ask for help and to request for learning materials from ScratChatbot rather than by searching and reading books by themselves. All students agreed that ScratchThAI could motivate and engage them to practice and to put more effort in learning with the goal to achieve higher scores and to rank higher in the class leaderboard.

2. Learning Flexibility, Efficiency and Effectiveness: Most students agreed that through the Request Learning Materials (U2) function, ScratchThAI could support them to understand the fundamental CT concepts as well as related topics such as geometry, coordinate, and angle better. They also had flexibility to select the types of learning materials they preferred at their own pace and their learning style. Considering coding practices, since ScratChatbot could assign students to practice additional exercises on the topics that they could not complete or did not pass, the students agreed that this iterative practice with immediate feedback could effectively strengthen their competency and understanding.
3. **Additional Features**: Students suggested to have a feature to re-submit and ask for more exercises based on their interests to improve their overall scores and their ranks in the leaderboard. Additional gamification features could also help engage and motivate them more.

### 4.3.4 Teacher satisfaction and perceptions

From the close observation and from the pilot study during the entire learning activities, the teachers recorded and reported the following important points:

1. **Learning Behaviors, Motivation and Engagement**: In the experimental group, students spent some time at the beginning to understand the mechanism of ScratchThAI for learning material request, coding practice, exercise submission, and personalized practice request. After that they were motivated to complete their personal learning plan, stayed engaged and interacted actively with the ScratchThAI. Moreover, with ScratchThAI’s score announcement, the leaderboard and the immediate feedback obtained after each practice submission, the experimental group students were motivated to self-learn, while the control group students had to wait for the teacher assistance and code verification when they faced learning difficulty.

2. **Learning Flexibility and Learning Improvement**: The teachers observed that ScratchThAI helped promote effective and flexible learning in the following ways: First, students interacted with ScratchThAI to progress their learning at their current knowledge level, and to practice coding exercises from the basic level to a more advanced one. Furthermore, ScratchThAI provided personalized learning assistance for diverse students to improving their learning through its additional practice assignment, automatic assessment and immediate feedback; hence, enabling mixed-ability students in the same class to learn at different paces. In other words, fast learners could continue with their exercises and quickly increase their progress, whereas slow learners or struggling students could get help either from ScratchThAI or from the in-class teachers.

3. **ScratchThAI Adoption by Teachers**: Teachers agreed that adopting ScratchThAI as a supporting learning tool in the computing science subject could not only enhance students learning experience, but also reduce teachers workload, and enable them to track and develop students’ learning performance individually. Moreover, since ScratchThAI can promote flexible learning regarding time and place, it can overcome the problem of limited school-hours. Students can use ScratchThAI at home as part of the assigned homework. Furthermore, several co-curricular activities can be organized.

4. **Limitations and Suggestions**: Even though ScratchThAI could alleviate the problem of mixed-ability students in both computational thinking and related mathematical skills, one concern was that the learning and achievement gaps may expand if low-performing and low-engaging students do not give extra effort and/or teachers do not provide timely intervention to assist their learnings.
4.4 Discussion

Interesting findings from the pilot study show that compared to the traditional learning method, ScratchThAI could improve learning performance, enhance teaching-learning activities, gain satisfaction and obtain positive perceptions from both teachers and students under the aforementioned challenges i.e., P1 (mixed-ability students in the same class), P2 (high student-teacher ratio) and P3 (learning-hour limitation). Survey feedback along with the observed learning behavior have pointed to the following factors:

1. ScratchThAI helps enhance teaching-learning activities for mixed-ability students by shifting the classroom from a teacher-led one into a more learner-focused one, where each student has more flexibility in learning at their own pace. Hence, fast learners can progress their learning and practices more efficiently, while weaker learners can seek help from the teacher or from the system. In addition, since the system can check the correctness of student codes automatically, the learning of the whole class can also progress faster. That is, the number of exercises and topics learned in ScratchThAI classroom were higher than that of the traditional Scratch classroom.

2. ScratchThAI helps teachers to manage a classroom of students with different learning abilities, to monitor progresses and performance, as well as to identify learning gaps. With ScratchThAI, teachers can classify students by their learning abilities and identify required assistance accordingly. Good performance students can self-learn new concepts through the system and gradually gain more understanding through exercises. On the other hand, at-risk students and their learning and achievement gaps will be identified so that teachers can effectively plan interventions and provide specific assistance within limited school hours.

3. The integrated ScrapChatbot keeps students engaged and motivated to practice. Based on its design specifically for young learners in the digital generation, who are familiar with technology, ScratchThAI provides interactive learning through engaging chatbot and gamification, i.e., progressive exercises and friendly competition.

4. As the effort in Q&A and in giving and grading assignments are offloaded to ScratchThAI, teachers can focus on course material preparation, paying attention to students’ progression and individual assistance offering.

5 Conclusion

This paper designed and proposed ScratchThAI, a conversation-based learning support framework, which adopts relevant educational approaches and related technology-enhanced learning (TEL) in its design and architecture to address the three critical CT development challenges in Thailand, namely, mixed ability students in the same class, high student-teacher ratio and limited school hours. Specifically, it incorporates the following approaches and design elements: (i) learner-centered and personalized learning approach, (ii) tracking of at-risk students, (iii) provision
of assistance to teachers to support and reduce teachers workload, (iv) provision of instructional support to students, and (v) flexible learning support in time and place. The key enabling TELs employed and developed include T1: personalized and adaptive technology, T2: automatic CT assessment/grading, T3: conversational agent (chatbot), and T4: cloud-based and web technology. By adopting ScratchThAI as part of Grade 5’s computing science course in a primary school in Thailand, the results revealed that ScratchThAI could alleviate the aforementioned challenges with the following supports: (i) achieving better learning performance, (ii) enhancing teaching and learning activities, (iii) promoting student engagement and motivation, (iv) positive students’ and teachers’ satisfaction.

Acknowledgements This research and innovation activity is funded by National Research Council of Thailand (NRCT) and Thailand Graduate Institute of Science and Technology (TGIST).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Alves, N. D. C., Von Wangenheim, C. G., & Hauck, J. C. (2019). Approaches to assess computational thinking competences based on code analysis in K-12 education: A systematic mapping study. *Informatics in Education, 18*(1), 17. https://doi.org/10.15388/infedu.2019.02

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Basawapatna, A. R., Repenning, A., & Koh, K. H. (2015). Closing the cyberlearning loop: Enabling teachers to formatively assess student programming projects. In *Proceedings of the 46th ACM technical symposium on computer science education* (pp. 12–17). https://doi.org/10.1145/2676723.2677269

Basu, S., Sengupta, P., & Biswas, G. (2014). A scaffolding framework to support learning of emergent phenomena using multi-agent-based simulation environments. *Research in Science Education, 45*(2), 293–324. https://doi.org/10.1007/s11165-014-9424-z

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students’ challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning, 11*(1), 1–35. https://doi.org/10.1186/s41039-016-0036-2

Bjork, E. L., & Bjork, R. A. (2011). Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning. *Psychology and the Real World: Essays Illustrating Fundamental Contributions to Society, 2*, 59–68.

Black, J., Brodie, J., Curzon, P., Myketiak, C., McOwan, P. W., & Meagher, L. R. (2013). Making computing interesting to school students: teachers’ perspectives. In *Proceedings of the 18th ACM conference on innovation and technology in computer science education*, (pp. 255–260). https://doi.org/10.1145/2462476.2466519

Bower, M., & Falkner, K. (2015). Computational thinking, the notional machine, pre-service teachers, and research opportunities. In *ACE* (pp. 37–46)
Campos, D. S., Mendes, A. J., Marcelino, M. J., Ferreira, D. J., & Alves, L. M. (2012). A multinational case study on using diverse feedback types applied to introductory programming learning. In 2012 Frontiers in education conference proceedings (pp. 1–6). IEEE. https://doi.org/10.1109/FIE.2012.6462412

Carvalho, T., Andrade, D., Silveira, J., Auler, V., Cavaleiro, S., Aguiar, M., & Reiser, R. (2013). Discussing the challenges related to deployment of computational thinking in brazilian basic education. In 2013 2nd workshop-school on theoretical computer science (pp. 111–115). IEEE. https://doi.org/10.1109/WEIT.2013.27

Cho, S., Pauca, P., & Johnson, D. (2014). Computational thinking for the rest of us: A liberal arts approach to engaging middle and high school teachers with computer science students. In Society for information technology & teacher education international conference (pp. 79–86). Association for the Advancement of Computing in Education (AACE).

Cook, C. R., Kilgus, S. P., & Burns, M. K. (2018). Advancing the science and practice of precision education to enhance student outcomes. Journal of School Psychology, 66, 4–10.

Deunk, M. I., Smale-Jacobse, A. E., de Boer, H., Doolaard, S., & Bosker, R. J. (2018). Effective differentiation practices: A systematic review and meta-analysis of studies on the cognitive effects of differentiation practices in primary education. Educational Research Review, 24, 31–54. https://doi.org/10.1016/j.edurev.2018.02.002

Djambong, T., & Freiman, V. (2016). Task-based assessment of students’ computational thinking skills developed through visual programming or tangible coding environments. International Association for Development of the Information Society.

Van Gorp, M. J., & Grissom, S. (2001). An empirical evaluation of using constructive classroom activities to teach introductory programming. Computer Science Education, 11(3), 247–260. https://doi.org/10.1076/csed.11.3.247.3837

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38–43.

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. Computer Science Education, 25(2), 199–237. https://doi.org/10.1080/08993408.2015.1033142

Grover, S., Lundh, P., & Jackiw, N. (2019). Non-programming activities for engagement with foundational concepts in introductory programming. In Proceedings of the 50th ACM technical symposium on computer science education (pp. 1136–1142). https://doi.org/10.1145/3287324.3287468

Habibu, T., Abdullah-Al-Mamun, M. D., & Clement, C. (2012). Difficulties faced by teachers in using ICT in teaching-learning at technical and higher educational institutions of Uganda. International Journal of Engineering, 31(7), 1–10.

Hailikari, T., Katajavuori, N., & Lindblom-Ylanne, S. (2008). The relevance of prior knowledge in learning and instructional design. American Journal of Pharmaceutical Education, 72(5). https://doi.org/10.5688/aj7205113

Hattie, J., & Timperley, H. (2007). The power of feedback. Review of Educational Research, 77(1), 81–112. https://doi.org/10.3102/003465430298487

Heintz, F., Mannila, L., & Färnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education, 2016 IEEE Frontiers in education conference (FIE), 2016, pp. 1–9, https://doi.org/10.1109/FIE.2016.7757410

Hooshyari, D., Ahmad, R. B., Yousefi, M., Yusop, F. D., & Horng, S. J. (2015). A flowchart-based intelligent tutoring system for improving problem-solving skills of novice programmers. Journal of Computer Assisted Learning, 31(4), 345–361.

Hooshyari, D., Lim, H., Pedaste, M., Yang, K., Fathi, M., & Yang, Y. (2019). AutoThinking: An Adaptive Computational Thinking Game. In L. Roñningsbakk, T.-T. Wu, F. E. Sandnes, & Y-M. Huang (Eds.), Innovative technologies and learning - 2nd international conference, ICITL 2019, proceedings (pp. 381–391). (Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics); Vol. 11937 LNCS). Springer. https://doi.org/10.1007/978-3-030-35343-8_41

Hooshyari, D., Malva, L., Yang, Y., et al. (2021). An adaptive educational computer game: Effects on students’ knowledge and learning attitude in computational thinking. Computers in Human Behavior, 114, 106575. https://doi.org/10.1016/j.chb.2020.106575

Nizamettin, K., & Bekir, C. (2015). The impact of number of students per teacher on student achievement. Procedia-Social and Behavioral Sciences, 177(1), 65–70. https://doi.org/10.1016/j.sbspro.2015.02.335
Katchaphakin, K., & Anutariya, C. (2018). An architectural design of scratchthai: A conversational agent for computational thinking development using scratch. In Proceedings of the 10th International Conference on Advances in Information Technology (pp. 1–7).

Katchaphakin, K., & Anutariya, C. (2019). Computational thinking development challenges: Case studies in Thai primary education. In Proceedings of the 27th International Conference on Computers in Education (pp. 362–371).

Koh, K. H., Basawaputra, A., Nickerson, H., & Repenning, A. (2014). Real time assessment of computational thinking. In 2014 IEEE Symposium on visual languages and human-centric computing (VL/HCC) (pp. 49–52). IEEE. https://doi.org/10.1109/VLHCC.2014.6883021

Kwon, K., & Cheon, J. (2019). Exploring problem decomposition and program development through block-based programs. International Journal of Computer Science Education in Schools, 3(1), n1.

Lockwood, J., & Mooney, A. (2017). Computational thinking in education: Where does it fit? A systematic literature review. arXiv:1703.07659

Marwan, S., Gao, G., Fisk, S., Price, T. W., & Barnes, T. (2020). Adaptive immediate feedback can improve novice programming engagement and intention to persist in computer science. In Proceedings of the 2020 ACM conference on international computing education research (pp. 194–203). https://doi.org/10.1145/3372782.3406264

McGill, M. M., & Decker, A. (2020). Tools, languages, and environments used in primary and secondary computing education. In Proceedings of the 2020 ACM conference on innovation and technology in computer science education (pp. 103–109). https://doi.org/10.1145/3341525.3387365

Montiel, H., & Gomez-Zermeño, M. G. (2021). Educational challenges for computational thinking in K-12 Education: A Systematic Literature Review of Scratch as an Innovative Programming Tool. Computers, 10(6), 69. https://doi.org/10.3390/computers10060069.

Mooney, A., Duffin, J., Naughton, T. J., Monahan, R., Power, J. F., Maguire, P. (2014) PACT: An initiative to introduce computational thinking to second-level education in Ireland. In International conference on engaging pedagogy 2014 (ICEP), 5th December 2014, athlone institute of technology.

Mouza, C., Yang, H., Pan, Y. C., Ozden, S. Y., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). Australasian Journal of Educational Technology, 33(3). https://doi.org/10.14742/ajet.3521

OECD (2012) Equity and quality in education: Supporting disadvantaged students and schools. OECD

Peng, H., Ma, S., & Spector, J. M. (2019). Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment. Smart Learning Environments, 6(1), 1–14. https://doi.org/10.1007/978-981-13-6908-7_24

Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In Proceedings of the 41st ACM technical symposium on Computer science education (pp. 265–269). https://doi.org/10.1145/1734263.1734357

Ribeiro, L., Nunes, D. J., da Cruz, M. K., & de Souza Matos, E. (2013). Computational thinking: Possibilities and challenges. In 2013 2nd Workshop-School on Theoretical Computer Science (pp. 22–25). IEEE. https://doi.org/10.1109/WEIT.2013.32

Román-González, M., Moreno-León, J., & Robles, G. (2017). Complementary tools for computational thinking assessment. In S.C. Kong, J. Sheldon & K.Y. Li (Eds.), Proceedings of international conference on computational thinking education (CTE 2017). The Education University of Hong Kong (pp. 154–159).

Saidin, N. D., Fariza, K., Martin, R., et al. (2021). Benefits and challenges of applying computational thinking in education. International Journal of Information and Education Technology, 11, 248–254. https://doi.org/10.18178/ijiet.2021.11.5.1519

Schleicher, A. (2016). Teaching excellence through professional learning and policy reform. International Summit on the Teaching Profession: Lessons from Around the World.

Schulte, C., Hornung, M., Sentance, S., Dagiene, V., Jevsikova, T., Thota, N., & Peters, A. K. (2012). Computer science at school/CS teacher education: Koli working-group report on CS at school. In Proceedings of the 12th Koli Calling international conference on computing education research (pp. 29–38). https://doi.org/10.1145/2401796.2401800

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher’s perspective. Education and Information Technologies, 22(2), 469–495. https://doi.org/10.1007/s10639-016-9482-0
Ssharratt, L. (2017). Scaffolded literacy assessment and a model for teachers’ professional development. In Perspectives on transitions in schooling and instructional practice (pp. 138–155). University of Toronto Press. https://doi.org/10.3138/9781442667105-011

Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. Computers & Education, 162, 104083. https://doi.org/10.1016/j.compedu.2020.104083

Tissenbaum, M., Sheldon, J., Sherman, M. A., Abelson, H., Weintrop, D., Jona, K., Horn, M., Wilensky, U., Basu, S., Rutstein, D., Snow, E., Shear, L., Grover, S., Lee, I., Klopfer, E., Jayathirtha, G., Shaw, M., Kafai, y., Mustafaraj, E., Temple, W., Shapiro, R. B., Lui, D., & Sorensen, C. (2018). The State of the Field in Computational Thinking Assessment. In J. Kay & R. Luckin (Eds.) Rethinking learning in the digital age: Making the learning sciences count, 13th international conference of the learning sciences (ICLS) 2018, Volume 2. London, UK: International Society of the Learning Sciences. https://doi.org/10.22318/cscl2018.1304

Wing, J.M. (2014) Computational thinking benefits society. 40th Anniversary Blog of Social Issues in Computing, 26

Wing, J. (2017). Computational thinking’s influence on research and education for all. Italian Journal of Educational Technology, 25(2), 7-14. Ortona, Italy: Edizioni Menabó - Menabó srl. Retrieved September 10, 2021 from https://www.learntechlib.org/p/183466/.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. ACM Transactions on Computing Education (TOCE), 14(1), 1–16. https://doi.org/10.1145/2576872

Authors and Affiliations

Kantinee Katchapakirin1 · Chutiporn Anutariya1 · Thepchai Supnithi2

Chutiporn Anutariya
chutiporn@ait.ac.th

Thepchai Supnithi
thepchai@nectec.or.th

1 Department of Information and Communications Technologies, School of Engineering and Technology, Asian Institute of Technology, Pathum Thani, Thailand

2 National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency, Pathum Thani, Thailand