SoftwareX 27 (2024) 101737

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Virtual CAT: A multi-interface educational platform for algorithmic thinking assessment

Giorgia Adorni [a,*], Simone Piatti [b], Volodymyr Karpenko [b]

[a] *Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI, Lugano, Switzerland*
[b] *Università della Svizzera italiana (USI), Lugano, Switzerland*

ARTICLE INFO

Keywords:
Algorithmic thinking
Educational assessment
Multilingual learning tool
Child-computer interaction
Gesture-based interface
Visual programming interface

ABSTRACT

The virtual Cross Array Task (CAT) is an educational platform designed to evaluate algorithmic thinking (AT) skills among students within Swiss compulsory education. This tool introduces an adaptable multi-interface system, enabling users to interact via intuitive gesture-based commands or through a visual programming interface that uses drag-and-drop blocks, facilitating a versatile approach to constructing and understanding algorithms. The platform encompasses a comprehensive training module for skill acquisition and a validation module for assessment. The system offers real-time feedback to users during activities, adjusting dynamically based on their actions, providing insights into progress and areas for improvement, thereby facilitating learning and performance enhancement. With multilingual capabilities extended to English, German, French, and Italian, the virtual CAT is intricately crafted to meet the diverse needs of educational contexts across various regions. Preliminary application and evaluation through a small-scale study indicate the virtual CAT's potential to offer scalable assessment and a robust platform for integrating AT into broader educational and research methodologies, setting the stage for its integration into academic research and daily pedagogical practice.

Code metadata

Current code version	Version 1.0.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-23-00771
Permanent link to Reproducible Capsule	https://github.com/GiorgiaAuroraAdorni/virtual-CAT-app/tree/main
Legal Code License	Creative Commons Attribution 4.0 International
Code versioning system used	git
Software code languages, tools, and services used	Flutter, Dart, Java
Compilation requirements, operating environments & dependencies	Flutter 3.13.8, Dart 3.1.4, Java 18.0.2.1, DevTools 2.25.0, Gradle 7.6, Android Studio/XCode
If available Link to developer documentation/manual	http://https://giorgiaauroraadorni.github.io/virtual-CAT-app/
Support email for questions	giorgia.aurora.adorni@gmail.com

1. Motivation and significance

Algorithmic Thinking (AT) involves a systematic approach to breaking down complex problems into manageable steps and devising sequential actions to solve them [1–5]. This skill is becoming increasingly critical in a world that is constantly being reshaped by digital advancements, extending beyond computer science to become foundational for problem-solving, logical analysis, and creative thinking in various contexts [1,6–9]. In education, AT facilitates a comprehensive understanding of algorithmic concepts and data structures, which is essential in developing the next generation of thinkers and innovators [1,2,10,11].

The rapid evolution of technology, coupled with the integration of computational concepts into diverse fields, underscores the pressing need for robust and adaptable assessment tools that are capable of delivering objective evaluations of AT proficiency in diverse educational settings [3,6,12–20].

* Corresponding author.
E-mail addresses: giorgia.adorni@idsia.ch (Giorgia Adorni), simone.piatti@usi.ch (Simone Piatti), volodymyr.karpenko@usi.ch (Volodymyr Karpenko).

https://doi.org/10.1016/j.softx.2024.101737
Received 12 November 2023; Received in revised form 14 March 2024; Accepted 11 April 2024
Available online 22 August 2024

While traditional assessment methods, such as closed-ended questions and multiple-choice tests, have faced criticism for limiting real-world problem-solving evaluation [15,21–24], open-ended questions, albeit showcasing critical thinking, tend to be time-consuming [23]. Programming assignments emphasise practical application but can be labour-intensive [16]. Automated systems, although offering prompt and consistent feedback and suitability for large-scale educational programs, are still evolving in their ability to comprehensively assess AT [18,18,25,26].

These assessment methods often neglect developmental aspects, social and environmental contexts [6,19,27–34]. Additionally, it is crucial to consider the impact of artefacts and the autonomy level of students engaged in the assessment, as these factors contribute to a nuanced and differentiated evaluation. The challenges posed by diverse evaluation methods and the lack of a standardised tool for assessing AT underscore the critical need for a comprehensive, reliable, and objective assessment tool. Such a tool is essential for establishing a baseline of proficiency, ensuring consistency, and facilitating meaningful measurements of progress across diverse age groups and educational settings [31,35–43].

The virtual Cross Array Task (CAT) [44], derived from an unplugged assessment activity [45], emerges as a response to the gaps identified in current AT assessment tools. This platform, conceived within the Swiss educational context, accommodates the diversity of this multilingual educational landscape, adeptly facilitating AT assessments for large, varied student populations. As a forward-looking resource in educational technology, the virtual CAT's automated, uniform assessment capabilities offer a foundation for broad-based research into teaching methods. Its effectiveness and adaptability have been validated in a study [46,47], affirming its potential for widespread AT evaluation.

The virtual CAT platform is designed to assess pupils' proficiency in formulating algorithms to describe and replicate visual configurations known as cross arrays. These structures consist of 20 dots forming a 2-thick cross, with colour choices from options like yellow, green, blue, or red, and exhibit various regularities and patterns. It caters to a diverse student population with varied backgrounds, profiles, and learning styles. This tool stands out for its multi-lingual capabilities and flexible interfaces, which offer the choice of gesture-based controls and visual programming, with commands in symbolic or textual representations.

A structured learning module, packed with engaging exercises and video tutorials, introduces users to AT fundamentals, ensuring a solid grasp of the software's functionalities and educational content. A validation module challenges users to apply their AT skills practically, through a set of challenging tasks.

A feedback system is integral to the virtual CAT, offering real-time responses to user inputs and interactions, and generating extensive performance reports. Such feedback is crucial for steering learners towards a nuanced comprehension and proficiency in AT, thus showcasing the platform's dedication to enhancing computational education.

By bridging the digital and educational divide, the virtual CAT not only aligns with the pedagogical needs of Swiss compulsory education but also extends to an international academic audience, potentially informing future developments in AT assessment.

2. Software description

This section provides a detailed overview of the virtual CAT software, including its architecture, user interface and functionalities, with an exploration of the technical composition and the underlying technological stack supporting its operation.

The software's design, following user-centric principles, involved setting objectives, analysing requirements, and drawing insights from the unplugged CAT version [45]. Prototypes centred on user experience underwent thorough testing, including expert inspections. Children actively participated in the design process, ensuring alignment with their needs. A practical test in a real educational setting identified areas for improvement, enhancing the tool's readiness for broader use. Further design and usability assessment details can be found in [46].

2.1. Software architecture

The virtual CAT application [44] has been developed using the Flutter framework [48], chosen for its cross-platform support. While optimised for iPad uses, Flutter ensures a uniform user experience on different devices. The application's architecture is structured into three main layers, programming language interpreter, data infrastructure, and user interface, creating an engaging, secure, and intuitive platform.

Programming language interpreter. The virtual CAT's programming language interpreter is a Dart package [49] developed to interpret and translate user interactions with the application interface into machine-readable instructions [50]. It ensures accurate execution of algorithmic commands and immediate feedback and error handling. The interpreter interacts with the CAT programming language, a standardised code that formalises user interactions into algorithms, enabling users to solve tasks within the app's virtual environment.

A detailed list of commands for board navigation, dot colouring, and complex operations is available in Table 1, equipping users with a versatile toolkit for algorithmic problem-solving.

Data infrastructure. In line with the strict privacy and security demands of Swiss educational environments, we developed a secure data infrastructure that ensures a safe flow of data during the collection and assessment process between the devices used and the central data collection hub [51]. To achieve this, we established a dedicated local network by deploying a router that interconnects all the participant devices, and directs the information to a central computer that acts as the nerve centre for data collection, ensuring that the data remains within a controlled and safeguarded environment.

Gradle is an integral part of this framework, acting as the ignition system for our data process. It initiates the server and establishes the necessary network connections, enabling a seamless data exchange from the user devices to our central database. This setup allows the data to be synchronised in real-time, ensuring that all information is current and accurately reflected across the system.

We leverage the H2 database system for data management and organisation. In particular, Java is used for its database engine execution, and SQL is used for data manipulation and interaction.

The database schema, illustrated in Fig. 1, features interconnected tables designed for specific roles within our educational assessment framework. Notably, the `ALGORITHMS` table catalogues algorithms based on complexity levels and specific commands, contributing to the algorithm dimension metric. Simultaneously, to compute the interaction dimension metric, the `RESULTS` table records information such as the artefact type and the level of autonomy.

Application interface. The final layer of the virtual CAT's architecture is the application interface, which presents the tangible interaction space for the end-users. To design the user interface (UI), we adhered to standard mobile application design principles, focusing on simplicity and ease of use, to accommodate the diverse technological proficiency of our target audience [52–55].

The app is designed to be simple and intuitive, allowing users to interact by touching and moving items directly on the screen, which is great for kids or those new to programming. It immediately shows visual cues that help users recognise if they are on the right track. The built-in feedback system responds to users' actions in real time, pointing out mistakes, confirming correct moves, and providing helpful tips.

Users can interact with the platform in two different modalities, each supporting different user preferences and learning styles: a gesture interface (CAT-GI) for direct interaction (see Fig. 2) and a visual programming interface (CAT-VPI) for creating sequences of commands (see Fig. 3).

The common top toolbar allows users to switch between interfaces, view results for completed schemas (success, failure, surrender), and check the ongoing schema score.

Table 1
List of commands in the CAT programming language.

Category	Command	Description
Movements	`goCell(String cell)`	Move directly to a specific cell coordinate on the board.
	`go(String move, int repetitions)`	Moves in one of the eight possible directions, including cardinal and diagonal movements, of a specified number of cells.
Colouring	`paintSingleCell(String color)`	Colour the current cells with the specified colour.
	`paintPattern(List<String> colors, String repetitions, String pattern)`	Colour multiple cells according to predefined patterns (cardinal, diagonal, square, L, zigzag) each with various directions. If a sequence of colours is specified, they will alternate following the selected pattern.
	`paintMultipleCells(List<String> colors, List<String> cellsPositions)`	Colour multiple dots with custom patterns, defined by specifying the coordinates of the cells to be coloured.
	`fillEmpty(String color)`	Colour all uncoloured dots on the board uniformly with the specified colour.
Loops	`repeatCommands(List<String> commands, List<String> positions)`	Repeats a sequence of commands (e.g., a series of go and paint) at specified coordinates.
	`copyCells(List<String> origin, List<String> destination)`	Copies colours from one set of coordinates (origin) to another (destination).
Symmetry	`mirrorBoard(String direction)`	Reflects the coloured dots on the board onto the non-coloured ones, in accordance with the principle of symmetry, along the specified direction (horizontally on the x-axis or vertically on the y-axis).
	`mirrorCells(List<String> cells, String direction)`	Applies symmetry to specific dots across a specified direction.
	`mirrorCommands(List<String> commands, String direction)`	Reflects a sequence of commands across a specified direction.

The right column is shared between both interfaces. It features a reference schema at the top and a cross representing the current progress at the bottom. Navigation arrows, a restart button, and the visual feedback icons are available above the reference schema. Users can confirm completed tasks with the green checkmark or surrender with the red cross.

The left sidebar of CAT-GI houses easily selectable buttons for colour selection and actions, while CAT-VPI's left sidebar contains pre-defined building blocks grouped within menus, corresponding to different sets of commands that can be used within the interface. A colour-coding system enhances clarity, ensuring blocks within the same menu share the same colour.

In CAT-GI, the main workspace allows users to directly touch and manipulate dots to colour the cross, while in CAT-VPI, users arrange and combine visual coding blocks from the left sidebar.

The CAT-VPI offers two versions with two different command representations to accommodate learners from different backgrounds, ages, and cognitive abilities. One version is textual and uses explicit linguistic expressions for commands and parameters, while the other is symbolic and allows non-literate learners to interact effectively with the interface.

2.2. Software functionalities

The virtual CAT is a multifaceted educational platform featuring a suite of functionalities that support the assessment of AT skills.

- **Training module**: The training module offers 15 sample tasks to solve, along with supplemental video tutorials, to help users familiarise themselves independently with the app and its functionalities.
- **Validation module**: The validation module assesses students' AT through 12 specialised tasks. Through the CAT score metric, students' performance is evaluated across two key dimensions, the type of interaction used and the type of algorithm produced, providing a nuanced understanding of their abilities. The interaction dimension gauges the complexity of artefacts used during the activity, such as the CAT-GI or the CAT-VPI, and the level of autonomy displayed by students, measured by their reliance on visual feedback. The interaction dimensions, given by the combinations of artefacts and autonomy levels, are scored based on their complexity. Simultaneously, the algorithm dimension assesses the complexity of students' generated algorithms, focusing on the operations involved. These operations range from 0D (colouring individual dots) to 1D (colouring multiple dots based on patterns like rows and diagonals) and 2D (involving intricate patterns like alternating colours or mirroring operations).
- **Diverse interaction and language support**: The multi-interaction methods (gesture-based and visual programming block interfaces) accommodated different learning styles. The multilingual options (English, German, French and Italian) cater to the diverse linguistic preferences.
- **Real-time feedback**: The immediate visual feedback and comprehensive performance reports support students' learning journeys and provide valuable insights for continuous improvement.
- **Data privacy and security**: The local network infrastructure ensures anonymity and security, assigning unique identifiers to each user to separate personal identity from performance data.
- **Scalable assessment capabilities**: Designed for both individual and large-scale assessments, the platform's robust architecture supports extensive data collection and analysis, critical for educational research and policy development.

These functionalities collectively underscore virtual CAT's commitment to providing an educational platform that is scalable, versatile, user-friendly, secure and responsive to the needs of a modern learning ecosystem.

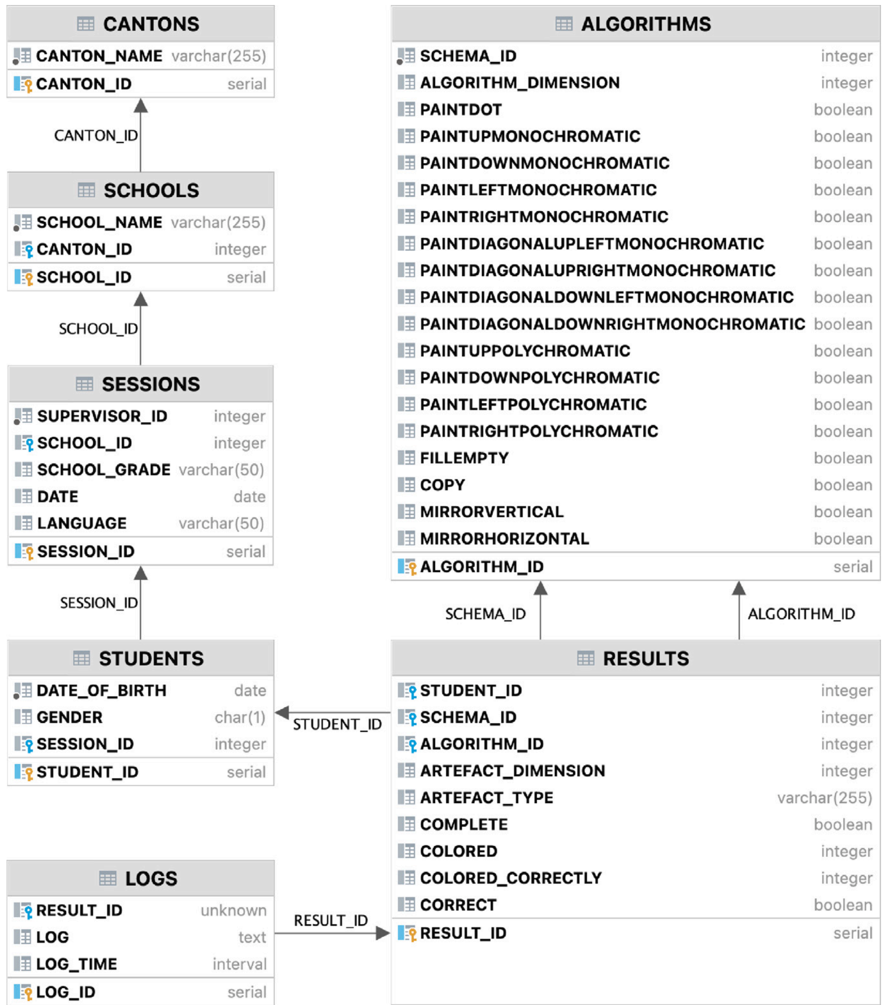

Fig. 1. Database schema. The diagram depicts the logical structure of the database, including tables such as CANTONS, SCHOOLS, SESSIONS, STUDENTS, ALGORITHMS, RESULTS, and LOGS, along with their attributes and the relationships between these entities.

3. Illustrative examples

This section provides a detailed guide through the virtual CAT application, showcasing its features and functionalities with illustrative images and a comprehensive screencast.

Initial interaction with the application presents users with a language selection screen. Users are to select their preferred language flag (for example, English) to ensure all instructions and content are subsequently displayed in the chosen language. Following this, users are prompted to choose between the training and validation modules.

3.1. Training module

The training module is designed to introduce users to the virtual CAT application. For this purpose, a generic introductory video and a set of 15 tasks are provided, each complemented by three in-app videos that explain each interface (see Fig. 4). When users select a task, they are guided by a tutorial video illustrating how to accomplish it. Subsequently, users are encouraged to apply the instructions provided in the video to solve the schema using the designated interaction interface. A green check mark on the video icon indicates successful task completion. The primary goal of this training process is to equip users with the necessary skills to navigate the application effectively and confidently, utilising its features. By completing the training tasks, users gain a solid foundation in using the virtual CAT software, preparing them for a seamless transition to the validation module. This structured approach ensures that users understand the software's functionalities and feel proficient in AT before progressing to more complex assessments.

3.2. Validation module

Upon transitioning from the training module, users enter the validation module, where the AT skills are tested. To initiate this process, a local network infrastructure is established using a router, connecting user devices, typically iPads, to a central computer with a configured database for secure data reception and storage.

The validation process begins by creating a session and inputting specific details such as the date, canton, school, class grade level, and supervisor information into the application. Student registration follows, recording only gender and date of birth. Unique identifiers are automatically assigned to each student and session to ensure privacy, facilitate re-entry and registration of new users to the same session and enabling individual progress tracking.

In the validation module, users encounter a series of 12 tasks, with the flexibility to select their interaction mode, navigate between tasks or restart them, confirm completion, or skip tasks as necessary.

On CAT-VPI, users use drag-and-drop interactions to assemble the programming commands in the central workspace and replicate the reference pattern, as illustrated in Fig. 5 where a student is attempting task 2 by placing a command to select the starting point for colouring and another to colour a blue square.

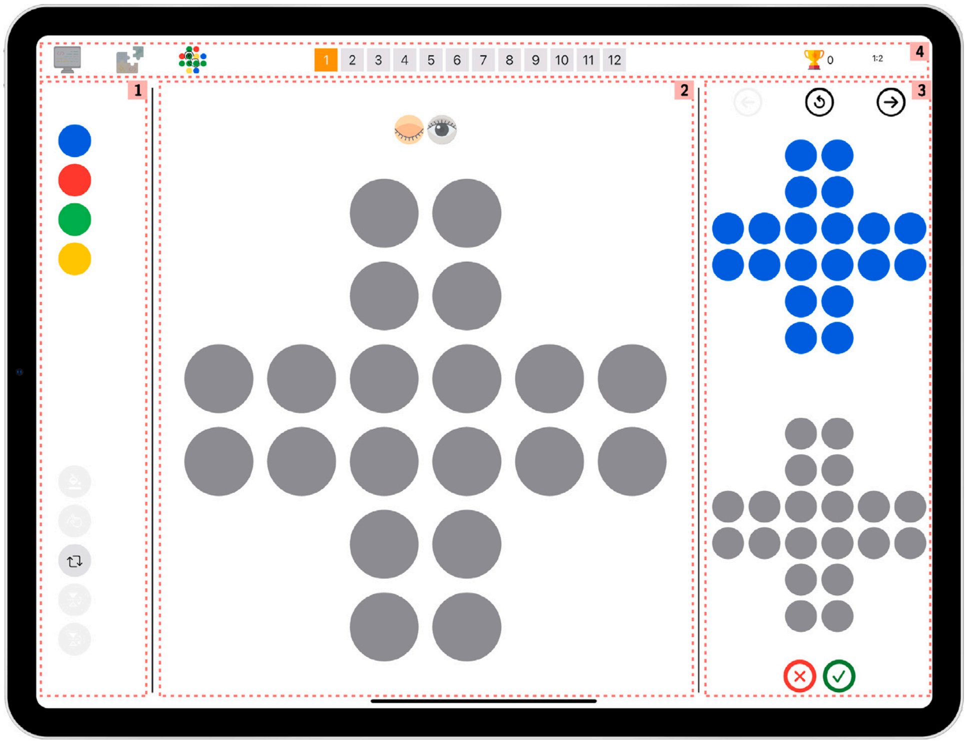

Fig. 2. CAT gesture interface (CAT-GI).

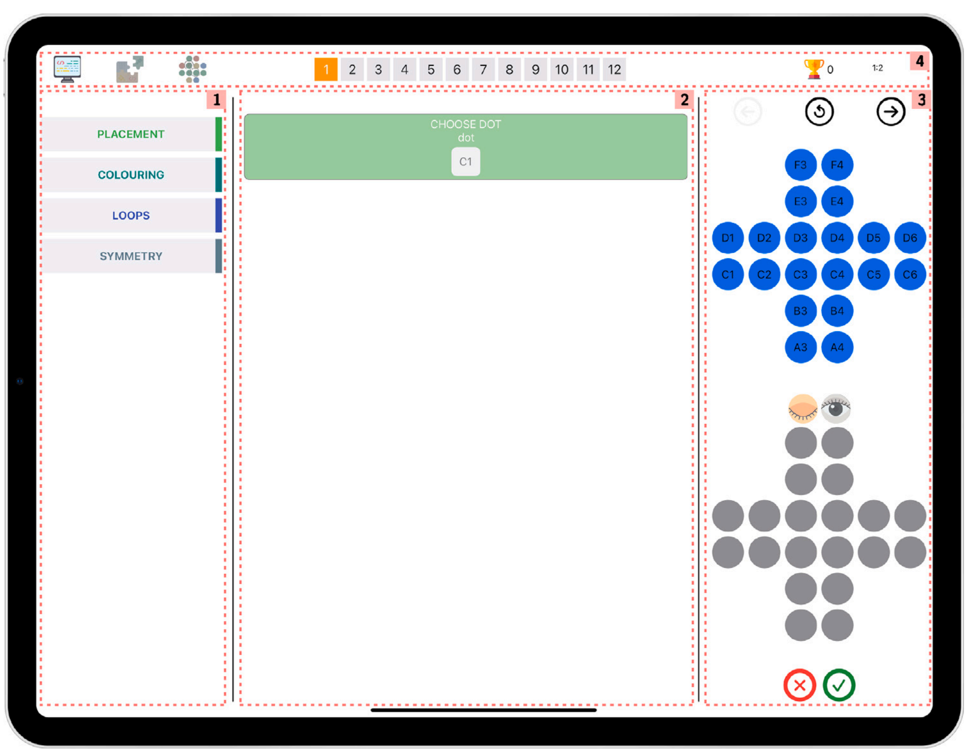

Fig. 3. CAT visual programming interface (CAT-VPI).

Fig. 4. Training module.

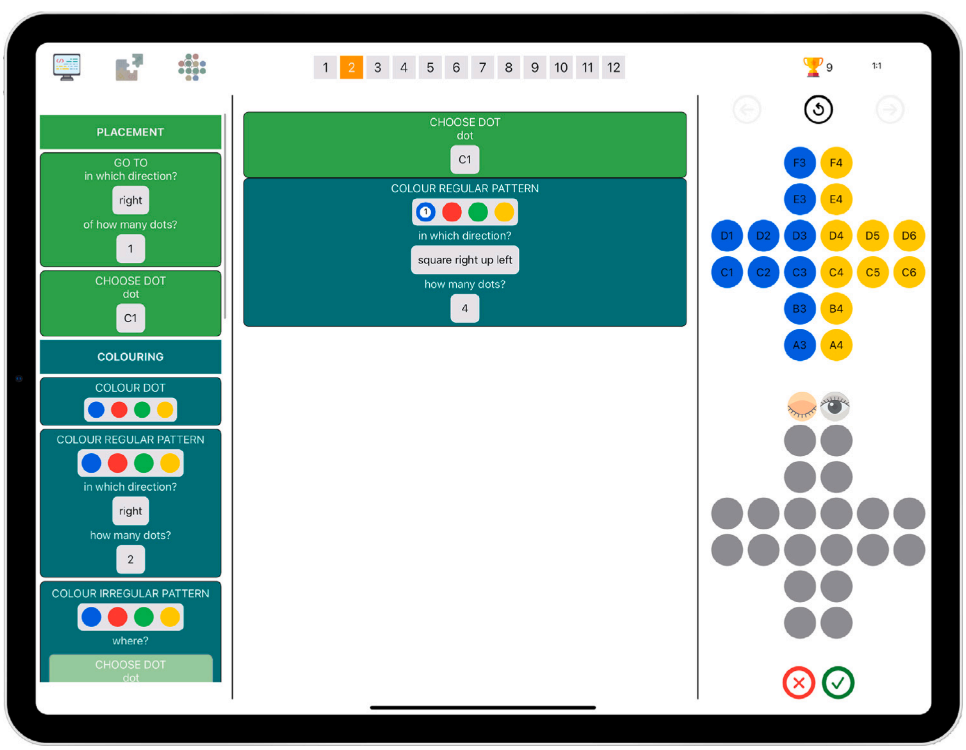

Fig. 5. Example of usage of the CAT-VPI with textual commands.

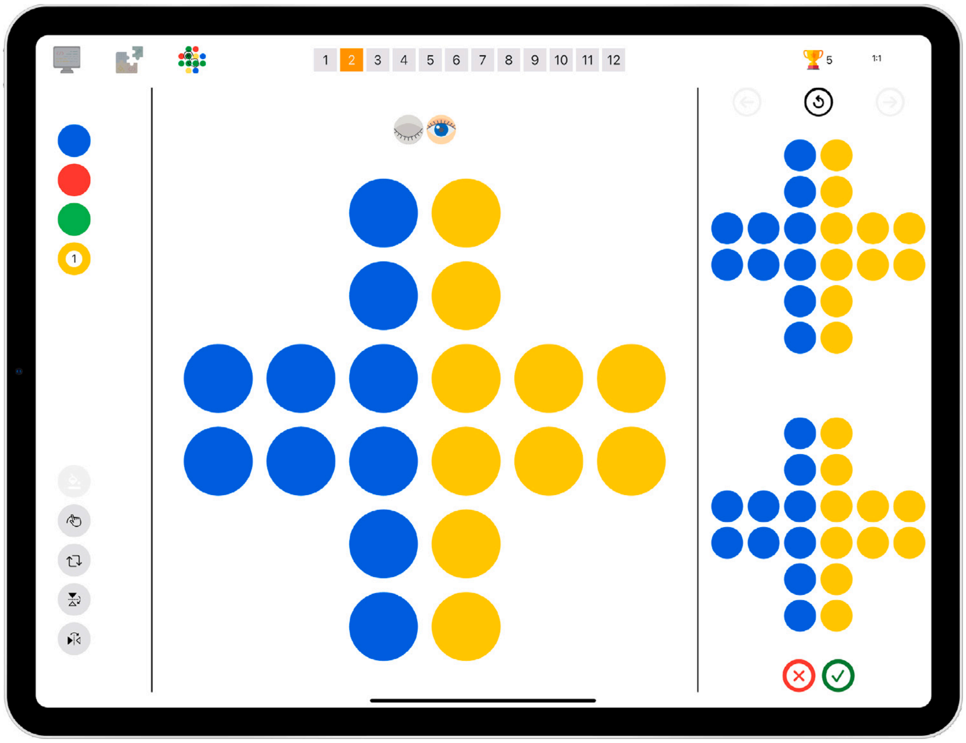

Fig. 6. Example of usage of the CAT-GI.

Users can freely switch between interaction interfaces at any time, such as transitioning to the CAT-GI, allowing direct manipulation of elements on the screen, as shown in Fig. 6, highlighting a switch to CAT-GI, colouring actions, and enabling visual feedback for result visualisation.

When the user believes the task has been completed, they can confirm completion by selecting the appropriate option. However, if the user chooses not to complete the task, they can bypass it and move on to the next one.

Completing all tasks leads to a results dashboard, which provides a comprehensive summary of the user's performance. This includes visual representations of attempted tasks, scores, completion status, and time taken.

Simultaneously, all session data are acquired and managed on the central computer where the database is configured, ensuring accessibility at all times.

4. Impact

The virtual CAT platform is an effort to contribute to the fields of education and algorithmic skills assessment. In this section, we explore its potential impact, recognising the preliminary nature of our work and the extensive research conducted in these areas.

We aim to offer a fresh perspective on the intricate dynamics of virtual environments and how they can support the assessment and enhancement of AT skills. Particularly, the virtual CAT catalyses studies investigating how various user interfaces might influence the development of AT competencies across diverse user groups. With features such as multilingual support and diverse interaction methods, it invites exploratory research into the effects of language and cultural backgrounds on learning algorithmic problem-solving.

Moreover, the virtual CAT introduces a method for capturing detailed user performance data, providing preliminary insights into learners' progress and interaction with the tool. This capability offers a potentially valuable resource for studies seeking to track educational outcomes over time or promptly evaluate the impact of instructional strategies.

As the virtual CAT is currently in the early stages of its life cycle, its adoption is not yet widespread, and we remain cautious but optimistic about its future role. Encouraging preliminary findings from a small-scale study have highlighted the tool's potential to support AT learning and assessment [46,47]. Examining user engagement statistics, analyses of interaction data revealed a significant level of involvement. On average, users spent approximately 13 min per session engaging with the platform, with an engagement rate of 83%. By comparison, traditional unplugged activities boasted an average session duration of 30 min. Furthermore, the success rate within the virtual CAT platform, defined by the completion of assigned tasks, stood at an impressive 80%. Additionally, our analysis found that the CAT-GI demonstrated quicker task completion times than its CAT-VPI counterpart. This observation suggests that the gesture interface, characterised by its simplicity, proves to be more intuitive and efficient for students when navigating the virtual CAT platform. Moreover, age emerged as a significant factor in user success rates, with older students generally achieving higher success rates than their younger counterparts. Such insights provide a robust foundation for ongoing inquiry and discussion within the academic community, driving forward the evolution and refinement of the virtual CAT platform.

5. Conclusions

The virtual CAT represents a stride towards enriching the Swiss compulsory education system with an assessment tool for AT skills. Throughout this paper, we have outlined its conceptual framework, design considerations and provided insights into its potential use from a small-scale study.

The virtual CAT's introduction as an educational platform has been met with an initial positive reception, indicating a promising avenue for enhanced interactive assessment of AT. The adaptability of its interface to cater to a multilingual audience reflects a conscientious effort to embrace the cultural and linguistic diversity of the student population. Its multi-interface design, which accommodates users of varied ages and diverse programming backgrounds, including those with no prior experience, demonstrates an inclusive approach. This flexibility not only aids in personalising the experience but also ensures that the virtual CAT is accessible and engaging for a broad user base.

The dual-module system, comprising both training and validation components, embodies a holistic approach that can be used for learning. It ensures that users not only engage with algorithmic concepts through a guided discovery process but also have their skills rigorously evaluated, providing a comprehensive educational experience.

Looking ahead, the virtual CAT holds great potential for impacting AT in education. With continued refinement and community participation, it could become a critical resource, bridging the gap between technology and education for a digitally fluent generation.

While the platform's innovative features set a precedent for future educational assessment tools, the true impact of the virtual CAT will only be measurable through extensive real-world application and research. The preliminary study has laid a foundational stone, and the insights garnered from these investigations are crucial stepping stones towards ongoing enhancement [46].

Our vision for future developments involves continuous refinement and expansion of the platform. To formally assess user experience, we have designed a brief survey to be incorporated at the end of the validation module to collect subjective impressions from pupils about how they perceive the tool and their overall experience. Additionally, we aim to integrate an intelligent assessment system (IAS) that performs inferences on the students' performance based on Bayesian networks with noisy gates. This system will provide an automatic and more comprehensive evaluation of students' skills and knowledge, enabling educators to tailor instruction more effectively to individual needs. We anticipate significant improvements in the efficiency and accuracy of assessment processes, ultimately enhancing the educational experience for all users.

Software availability

The software components discussed in this study are open-source and available on Zenodo:

- Virtual CAT platform: [44]
- Programming language interpreter: [50]
- Data infrastructure: [51]

CRediT authorship contribution statement

Giorgia Adorni: Conceptualization, Data curation, Methodology, Software, Supervision, Writing – review & editing, Validation, Visualization, Writing – original draft. **Simone Piatti:** Software. **Volodymyr Karpenko:** Software.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Giorgia Adorni reports financial support was provided by Swiss National Science Foundaton. Volodymyr Karpenko reports financial support was provided by Swiss National Science Foundaton. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset supporting this research, collected in the small-scale study, can be accessed in a Zenodo repository: [47].

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT and Grammarly to enhance language and readability. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Acknowledgements

We would like to express our appreciation for the contributions of Lucio Negrini, Francesco Mondada, Francesca Mangili, Dorit Assaf, and Luca Maria Gambardella, members of this research project, and Jérôme Guillaume Brender and Giovanni Profeta. They generously offered their time and expertise to test the application, providing valuable feedback throughout its refinement and usability.

Funding

This research was funded by the Swiss National Science Foundation (SNSF) under the National Research Program 77 (NRP-77) Digital Transformation (project number 407740_187246).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2024.101737.

References

[1] Wing JM. Computational thinking. Communications of the ACM 2006;49(3):33–5. http://dx.doi.org/10.1145/1118178.1118215.
[2] Seehorn D, Carey S, Fuschetto B, Lee I, Moix D, O'Grady-Cunniff D, Owens BB, Stephenson C, Verno A. CSTA K–12 computer science standards: Revised 2011. Association for Computing Machinery; 2011, https://dl.acm.org/doi/book/10.1145/2593249.
[3] Barr V, Stephenson C. Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads 2011;2(1):48–54. http://dx.doi.org/10.1145/1929887.1929905.
[4] Poulakis E, Politis P. Computational thinking assessment: Literature review. In: Research on e-learning and ICT in education: technological, pedagogical and instructional perspectives. Cham: Springer International Publishing; 2021, p. 111–28. http://dx.doi.org/10.1007/978-3-030-64363-8_7.
[5] Futschek G. Algorithmic thinking: The key for understanding computer science. In: Informatics education – the bridge between using and understanding computers. Berlin Heidelberg: Springer; 2006, p. 159–68. http://dx.doi.org/10.1007/11915355_15.
[6] Yadav A, Mayfield C, Zhou N, Hambrusch SE, Korb JT. Computational thinking in elementary and secondary teacher education. Acm Transactions on Computing Education 2014;14(1):1–16. http://dx.doi.org/10.1145/2576872.
[7] Yildiz CD. Ideal classroom setting for English language teaching through the views of English language teachers (a sample from Turkey). English Language Teaching 2020;13(3):31. http://dx.doi.org/10.5539/elt.v13n3p31.
[8] Amini R, Stolz L, Kartchner J, Thompson M, Stea N, Joshi R, Adhikari S, Hawbaker N. Bedside echo for chest pain: An algorithm for education and assessment. Advances in Medical Education and Practice 2016;293. http://dx.doi.org/10.2147/amep.s103083.
[9] Jocz J, Peterson KA, Pfeif D. Motivating youth to learn STEM through a gender inclusive digital forensic science program. Smart Learning Environments 2023;10(1). http://dx.doi.org/10.1186/s40561-022-00213-x.
[10] Kules B. Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes. Proceedings of the Association for Information Science and Technology 2016;53(1):1–6. http://dx.doi.org/10.1002/pra2.2016.14505301092.
[11] Olkhova NV. Development of algorithmic thinking in primary school students when studying computer science. Scientific Bulletin of Mukachevo State University. Series Pedagogy and Psychology 2022;8(2):25–32. http://dx.doi.org/10.52534/msu-pp.8(2).2022.25-32.
[12] Jiménez Toledo JA, Collazos CA, Ortega M, Ramos DX. Algorithmic thinking and extension of its definition for trainee software developers: A systematic literature mapping. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje 2023;18(4):331–43. http://dx.doi.org/10.1109/rita.2023.3323784.

[13] Grover S, Pea R. Computational thinking in K–12: A review of the state of the field. Educational Researcher 2013;42(1):38–43. http://dx.doi.org/10.3102/0013189X12463051.
[14] Voogt J, Roblin NP. A comparative analysis of international frameworks for 21st century competences: Implications for national curriculum policies. Journal of Curriculum Studies 2012;44(3):299–321. http://dx.doi.org/10.1080/00220272.2012.668938.
[15] Wickey da Silva Garcia F, Ronaldo Bezerra Oliveira S, da Costa Carvalho E. Application of a teaching plan for algorithm subjects using active methodologies: An experimental report. International Journal of Emerging Technologies in Learning (Ijet) 2022;17(07):175–207. http://dx.doi.org/10.3991/ijet.v17i07.28733.
[16] Sun D, Ouyang F, Li Y, Zhu C. Comparing learners' knowledge, behaviors, and attitudes between two instructional modes of computer programming in secondary education. International Journal of STEM Education 2021;8(1). http://dx.doi.org/10.1186/s40594-021-00311-1.
[17] Keith PK, Sullivan FR, Pham D. Roles, collaboration, and the development of computational thinking in a robotics learning environment. In: Computational thinking education. Springer Singapore; 2019, p. 223–45. http://dx.doi.org/10.1007/978-981-13-6528-7_13.
[18] Qian Y, Lehman JD. Using technology to support teaching computer science: A study with middle school students. Eurasia Journal of Mathematics Science and Technology Education 2018;14(12). http://dx.doi.org/10.29333/ejmste/94227.
[19] Román-González M, Moreno-León J, Robles G. Complementary tools for computational thinking assessment. In: Kong SC, Sheldon J, Li KY, editors. Proceedings of international conference on computational thinking education. The Education University of Hong Kong; 2017, p. 154–9, https://api.semanticscholar.org/CorpusID:208192014.
[20] OECD. PISA 2018 results (Volume VI): Are students ready to thrive in an interconnected world? OECD: Organisation for Economic Co-operation and Development; 2020, p. 426. http://dx.doi.org/10.1787/d5f68679-en.
[21] Oyelere SS, Agbo FJ, Sanusi IT. Developing a pedagogical evaluation framework for computational thinking supporting technologies and tools. Frontiers in Education 2022;7. http://dx.doi.org/10.3389/feduc.2022.957739.
[22] Simmering VR, Ou L, Bolsinova M. What technology can and cannot do to support assessment of non-cognitive skills. Frontiers in Psychology 2019;10. http://dx.doi.org/10.3389/fpsyg.2019.02168.
[23] Csernoch M, Biró P, Máth J, Abari K. Testing algorithmic skills in traditional and non-traditional programming environments. Informatics in Education 2015;14(2):175–97. http://dx.doi.org/10.15388/infedu.2015.11.
[24] Campbell-Barr V, Lavelle M, Wickett K. Exploring alternative approaches to child outcome assessments in Children's centres. Early Child Development and Care 2012;182(7):859–74. http://dx.doi.org/10.1080/03004430.2011.590937.
[25] Romero M, Lepage A, Lille B. Computational thinking development through creative programming in higher education. International Journal of Educational Technology in Higher Education 2017;14(1):1–15. http://dx.doi.org/10.1186/s41239-017-0080-z.
[26] Stanja J, Gritz W, Krugel J, Hoppe A, Dannemann S. Formative assessment strategies for students' conceptions—the potential of learning analytics. British Journal of Educational Technology 2022;54(1):58–75. http://dx.doi.org/10.1111/bjet.13288.
[27] Brennan K, Resnick M. New frameworks for studying and assessing the development of computational thinking. In: Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada, vol. 1, 2012, p. 25, https://scratched.gse.harvard.edu/ct/files/AERA2012.pdf.
[28] Korkmaz Ö, Bai X. Adapting Computational Thinking Scale (CTS) for Chinese High School students and their thinking scale skills level. Participatory Educational Research 2019;6(1):10–26. http://dx.doi.org/10.17275/per.19.2.6.1.
[29] Korkmaz Ö, Çakir R, Özden MY. A validity and reliability study of the Computational Thinking Scales (CTS). Computers in Human Behavior 2017;72:558–69. http://dx.doi.org/10.1016/j.chb.2017.01.005.
[30] Martínez ML, Lévêque O, Benítez I, Hardebolle C, Zufferey JD. Assessing computational thinking: Development and validation of the algorithmic thinking test for adults. Journal of Educational Computing Research 2022;60(6):1436–63. http://dx.doi.org/10.1177/07356331211057819.
[31] Román-González M, Moreno-León J, Robles G. Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In: Computational thinking education. Springer Singapore; 2019, p. 79–98. http://dx.doi.org/10.1007/978-981-13-6528-7_6.
[32] Tsai MJ, Liang JC, Hsu CY. The computational thinking scale for computer literacy education. J Educ Comput Res 2020;59(4):579–602. http://dx.doi.org/10.1177/0735633120972356.
[33] Román-González M, Pérez-González J-C, Jiménez-Fernández C. Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. Computers in Human Behavior 2017;72:678–91. http://dx.doi.org/10.1016/j.chb.2016.08.047.
[34] Polat E, Hopcan S, Kucuk S, Sisman B. A comprehensive assessment of secondary school students' computational thinking skills. British Journal of Educational Technology 2021;52(5):1965–80. http://dx.doi.org/10.1111/bjet.13092.
[35] Ezeamuzie NO, Leung JW. Computational thinking through an empirical lens: A systematic review of literature. Journal of Educational Computing Research 2021;60(2):481–511. http://dx.doi.org/10.1177/07356331211033158.
[36] Scherer R, Siddiq F, Viveros BS. The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. Journal of Educational Psychology 2019;111(5):764–92. http://dx.doi.org/10.1037/edu0000314.
[37] Pilotti M, Nazeeruddin E, Mohammad N, Daqqa I, Abdelsalam H, Abdullah MM. Is initial performance in a course informative? Machine learning algorithms as aids for the early detection of at-risk students. Electronics 2022;11(13):2057. http://dx.doi.org/10.3390/electronics11132057.
[38] Adorni G, Piatti A, Bumbacher E, Negrini L, Mondada F, Assaf D, Mangili F, Gambardella LM. A theoretical framework for the design and analysis of computational thinking problems in education. 2024, http://dx.doi.org/10.48550/arXiv.2403.19475.
[39] Moreno-León J, Robles G, Román-González M. Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. RED. Revista de Educación a Distancia 2015. https://www.redalyc.org/articulo.oa?id=54741184010.
[40] Ota G, Morimoto Y, Kato H. Ninja code village for scratch: Function samples/function analyser and automatic assessment of computational thinking concepts. In: 2016 IEEE symposium on visual languages and human-centric computing. IEEE; 2016, p. 238–9. http://dx.doi.org/10.1109/vlhcc.2016.7739695.
[41] Koh KH, Basawapatna A, Bennett V, Repenning A. Towards the automatic recognition of computational thinking for adaptive visual language learning. In: 2010 IEEE symposium on visual languages and human-centric computing. IEEE; 2010, p. 59–66. http://dx.doi.org/10.1109/vlhcc.2010.17.
[42] Koh KH, Basawapatna A, Nickerson H, Repenning A. Real time assessment of computational thinking. In: 2014 IEEE symposium on visual languages and human-centric computing. IEEE; 2014, p. 49–52. http://dx.doi.org/10.1109/VLHCC.2014.6883021.
[43] Ortega Ruipérez B, Asensio Brouard M. Evaluar el Pensamiento Computacional mediante Resolución de Problemas: Validación de un Instrumento de Evaluación. Revista Iberoamericana de Evaluación Educativa 2021;14(1):153–71. http://dx.doi.org/10.15366/riee2021.14.1.009.
[44] Adorni G, Piatti S, Karpenko V. virtual CAT: An app for algorithmic thinking assessment within Swiss compulsory education. 2023, http://dx.doi.org/10.5281/zenodo.10027851, (1.0.1). Zenodo Software.
[45] Piatti A, Adorni G, El-Hamamsy L, Negrini L, Assaf D, Gambardella L, Mondada F. The CT-cube: A framework for the design and the assessment of computational thinking activities. Computers in Human Behavior Reports 2022;5:100166. http://dx.doi.org/10.1016/j.chbr.2021.100166.
[46] Adorni G, Piatti A. The Virtual CAT: A tool for algorithmic thinking assessment in Swiss compulsory education. 2024, http://dx.doi.org/10.48550/arXiv.2408.01263, arXiv:2408.01263.
[47] Adorni G. Dataset for algorithmic thinking skills assessment: Results from the virtual CAT pilot study in Swiss compulsory education. 2023, http://dx.doi.org/10.5281/zenodo.10018292, (1.0.0). Zenodo Dataset.
[48] Google and community. Flutter framework. 2017, https://flutter.dev/.
[49] Bak L, Lund K. Dart programming language. 2011, https://dart.dev/.
[50] Adorni G, Karpenko V. virtual CAT programming language interpreter. 2023, http://dx.doi.org/10.5281/zenodo.10016535, (1.0.0). Zenodo Software.
[51] Adorni G, Karpenko V. virtual CAT data infrastructure. 2023, http://dx.doi.org/10.5281/zenodo.10015011, (1.0.0). Zenodo Software.
[52] Nayebi F, Desharnais J-M, Abran A. The state of the art of mobile application usability evaluation. In: 2012 25th IEEE Canadian conference on electrical and computer engineering. IEEE; 2012, p. 1–4. http://dx.doi.org/10.1109/ccece.2012.6334930.
[53] Coursaris CK, Kim DJ. A meta-analytical review of empirical mobile usability studies. Journal of Usability Studies 2011;6(3):117–71, http://uxpajournal.org/a-meta-analytical-review-of-empirical-mobile-usability-studies/.
[54] Tidwell J. Designing interfaces: Patterns for effective interaction design. O'Reilly Media; 2010, https://dl.acm.org/doi/10.5555/1965339.
[55] Hartson R, Pyla PS. The UX book: Agile UX design for a quality user experience. Morgan Kaufmann; 2018, http://dx.doi.org/10.1016/c2013-0-19285-9.