

Investigating the Use of TUIs and GUIs for 6- to 7-Year-Old Programming Learning

Abrar Almjally
University of Sussex
Brighton, UK
A.Almjally@sussex.ac.uk

There is persistent interest in tangible approaches to supporting young children's learning of programming, but there has been mixed evidence about its benefits, and few studies have investigated the specific features that offer learning benefits in classroom contexts. This research will involve the use of programming blocks that can be instantiated in two contrasting types of interface: a tangible user interface (TUI) and a graphical user interface (GUI). The system will be designed to minimize extraneous differences between the two interfaces in order to isolate the variables of interest. Using a between-subjects design, the study will investigate the impact of interface type on learning and on attitudinal outcomes for children aged 6 to 7. From an embodied interaction perspective, the study will analyse the cognitive advantages of each interface, including identifying how and why each interface type might affect learning outcomes. The study will also investigate children's spontaneous gestures as indicators of understanding. Finally, the research will explore the relationships between interface types, attitudes towards computing, engagement and gender.

Tangible interaction; multi-touch interaction; embodied interaction, user interfaces.

1. CONTEXT AND MOTIVATION

There is growing interest in supporting both boys' and girls' Computer Science (CS) education when they are young [1], [2]. To increase the interest, engagement, and participation in CS, different approaches, including CS Unplugged (<https://csunplugged.org/en/>), and novice programming environments such as Alice [3], Scratch [4], and Green foot [5], all of which help to encourage youth to choose CS majors. However, few studies have focused on tools for children at the levels of nursery and primary education. There is evidence that children can learn programming from early childhood [6], and that this can be beneficial for children's cognitive development [7]. There is particular interest in encouraging positive engagement with computing for girls at this age, with a view to improving the gender balance in the field. Limited research has been carried out on the effects of different types of programming interface, such as TUIs vs GUIs, for younger children, and on the particular features of such interfaces that affect learning and enjoyment [6]. In this context, TUI and GUI refer to programming environments that use blocks-based programming. Specifically, a TUI programming interface has physical blocks that users can manipulate by hand, and the output is demonstrated by the physical movement of a robot. A GUI version has virtual blocks that can be manipulated by touchscreen or mouse, and produces virtual action on the screen. Although there are many studies comparing tangible

programming blocks and graphical programming blocks for learning programming [6], there has been little focus on unpicking exactly how interacting with different types of interface fosters children's understanding. Research underpinning embodied interaction theory suggests that actions such as object manipulation or using spontaneous gesturing have potential for supporting learning [8]. An embodied interaction is an interaction that centers on embodied cognition, based on the idea that cognition is related to the perceptual and physical interaction of the human body and the environment [8]. Embodied interaction can support the learning process because it emphasizes understanding the interaction between learning behaviors and cognitive development and what this looks like in the learning environment [8]. For example, even the understanding of abstract symbols used in everyday education, such as numbers in math, and function words in programming, can be developed via embodied interaction to foster learning. Tangibles might trigger affordances for action that facilitate retrieval from memory [8] and at the same time inspire spontaneous gestures that accompany verbal communication which exposes mental understanding of abstract concepts. Furthermore, there has been little empirical work that explores why and how different interfaces may or may not enhance learning [6], particularly the role of using embodied interaction such as object manipulation and spontaneous gesturing. Thus, this work will investigate the differential impact on girls and boys of TUI vs GUI programming environments on their

Abrar Almjally

learning outcomes, engagement, attitude, and perceptual and mental imagery (represented by spontaneous gesturing) in the experimental setting.

2. RESEARCH OBJECTIVES AND QUESTIONS

This research will compare a TUI with a GUI for learning programming skills by young girls and boys aged six to seven years old, investigating the role of embodied interaction theory, and any gender effects on learning outcomes, engagement, attitude, and spontaneous gestures. The following research questions will be addressed:

How can TUIs, especially tangible programming languages (TPLs), support the development Primary computing skills?

- (i) How does interface type (TUI or GUI) affect children's learning outcomes?
- (ii) How does interface type (TUI or GUI) affect children's attitudes towards computing?
- (iii) How does interface type (TUI or GUI) affect children's engagement with computing?
- (iv) What is the effect of interface type (TUI or GUI) on children's use of spontaneous gestures?
- (v) What is the effect of spontaneous gestures on children's learning outcomes?
- (vi) Will there be any differential gender effects in the above measures?

3. BACKGROUND & RELATED WORK

The use of computers has been taught for more than three decades, and several countries have introduced CS and Computational thinking CT(more in the next section) into their curricula [9]. Few studies focus on children at the level of preschool or primary education (e.g.[10], [11]) and more empirical research is needed to investigate younger children's learning experiences using programming tools. Starting early has advantages, as children are often eager to learn. In a recent study, [12] investigated children's knowledge of computers. A total of 18 boys between 5 and 8 years old from two Scottish primary schools were interviewed about understanding what computers are, how computers are programmed, and their beliefs about them. The children demonstrated intelligent answers about computer usage, but they were unable to explain how computers work, and they had their own theories about if computers could think. The authors proposed fundamental concepts that young children could learn to enhance their knowledge of computers, as it is important for children to know the limits and capabilities of technology. Current knowledge of computing functionalities might benefit children's everyday lives or even help them to manage their lives more effectively. It is thus important for children to know the advantages and

the disadvantages of being technology dependent (Robertson, Manches & Pain, 2017). In addition, children need to use technology in a way that is suitable for their cognitive development and that allows them to engage with their environments. Integrating CS with their school or everyday teaching materials and curricula, for example, is vital. One of the famous studies done in this area is that of [13]. They looked at the possibility of engaging young children from 4 to 6 (n=53) in learning computer concepts. These researchers developed the TangibleK Robotics curriculum (<http://ase.tufts.edu/DevTech/tangiblek/>), which contains 20 hours of instruction and a final project to assist children in understanding CS concepts. The study revealed that when provided with age-suitable technologies, curricula and education, young children could keenly participate in learning computer programming and take the early steps needed to develop computational thinking. They emphasized that teaching CS in a way that is suitable for children's development levels is important. Additional research is required to investigate the affordances of different programming tools on children learning and their effect on understanding computing concepts and cognitive abilities such as problem-solving tasks.

3.1 Computer education

In the field of teaching computer skills, the term computational thinking, first introduced by Papert in 1980, is increasingly popular. Jeannette Wing more recently revised and developed the term from Papert's (1980) earlier ideas and popularized the term more widely, using it to refer to handling complex problems by breaking them down into small parts, tracing errors, debugging them iteratively and trying to refine and enhance the design over time, including developing the understanding of human behaviour [14]. Since 2006, disagreement about CT has emerged, such as a lack of agreement about its usefulness and definitions, how it can be transferred to other domains, the fact that it is difficult to assess, the limited learning resources available for teaching it and unconfident teachers. Many efforts have been made to address these issues[13], [15]–[20]. Researchers have reached a consensus on some CT concepts, such as abstractions, decompositions, debugging, algorithms and pattern recognition, which indicates some overall agreement on the main concepts that computer scientists use to solve real-world issues. CT has been adopted in the primary computer curricula such as in the UK, allowing potential for engaging with the basic computer concepts without the need to tackle the often complex syntax of the programming language, for example, to ask students to decompose a big task like organizing a school trip, which can be decomposed into smaller pieces such as booking coach, staffing, checking weather, etc. [21] Young

Abrar Almjally

children from 5 to 7 need to learn how to develop and debug a simple programming project and understand simple algorithms and how they can operate and implement them [12].

3.2 Girls in computing

Although vast work has been completed to inspire women to be involved in the computer domain, they remain underrepresented [22]. For example, robotic competitions appeal to a higher number of young men than women, predominantly in free-choice learning environments, such as after-school courses and museum programs. Only 30% of the FIRST LEGO League are girls [23]. However, the lack of girls entering the computing domain is not limited to after-school programs. For instance, at the General Certificate of Secondary Education (GCSE) level, the number of girls choosing to study computing was only 17% in 2015 (Joint Council for Qualifications, 2015b) and 16.1% in 2016. Even though girls exhibit less confidence in their skills and their numbers are fewer in CS, they generally demonstrate better CS skills [24]. Researchers have indicated a gender gap in the general participation rates at both the school level and the university level [23][25]. There are many different reasons why this is happening, such as girls' lack of knowledge about CS application, the mediums and style of teaching that are used to teach computing or other physiological reasons which are not addressed in this research. Females might be more interested in computing if they were introduced to it and explored its functionality in other domains [26]. Doing this at younger ages might encourage them to pursue CS because they become familiar with it as a domain. Regarding the learning tools, there is an agreement of the effect of using different learning mediums such as tangible user interfaces (TUI) or screen-based interfaces (SBI) to foster girls' computer education. Several studies, such as [17], [27], [28], considered exploiting CT concepts to encourage females to take part in the computer field. In [29], they found that both boys and girls ages 15 to 18 years old reach the same CT skill levels after additional training for girls. [30] Found that young females (aged 11–14 years) spend expressively more time to learn after school that reflects better performance than boys. However, there is no difference among genders stated by [31] ages 10 to 14 years and by [32] ages 6 to 12 years. Another inconclusiveness is the preferences of types of interfaces that are more appealing to girls. In research by [33] conducted in a museum of science to examine the possibility of different interfaces to create computer programs, they found that tangible interfaces seem to attract both girls and boys, while graphical user interfaces (GUI) appeal more to boys who are younger than 16 years old. Similar results were found for children ages 5 to 8 years old [34]. In contrast, in a study aimed to investigate the role of user interfaces on children's (between 4 and 7 years

old) mastery of CT, they found that TUI sessions attracted more boys than girls who registered to the Scratch Jr session (GUI) [35]. Therefore, more investigation is needed of the gender influence on children's preferences of diverse interfaces and abilities to develop CS concepts. We will address the influence of gender on preferences of different interfaces and their ability to learn computing concepts such as programming.

3.3 Action-based learning

Theories of embodiment, for example [36], and developmental psychology such as Thelen (1995) are defined in detail that link behavioural activity with basic cognitive and interactional processes and provide a theoretical foundation for theorising the usefulness of behavioural activity in structuring knowledge. Theories of embodiment can be used in several ways to shape the design of novices' interaction and learning using TUI [37]. First, individuals or learners use complementary strategies to adopt the environment in order to improve their cognitive strategies for problem-solving. In other words, complementary strategies are strategies through which a mental task is dynamically distributed to action in the surroundings [38], for example, organizing activities such as arranging the position of objects. Complementary strategies involve actions of two types: epistemic and pragmatic. Epistemic is an action that changes the situation to decrease cognitive work. Epistemic actions support cognition to decrease the memory involved, the number of steps or the possibility of failure in mental computation [36]. In contrast, pragmatic actions are actions that bring individuals closer to their goals, such as winning a game. [39] Described how children were physically turning and rotating jigsaw pieces to solve the puzzle problem, which is an example of epistemic action. Thus, action-based learning supports epistemic action such as TUI, which support physical manipulation, and provides a better learning experience when it's used for specific context and users because it supports the cognitive [39]. Two other studies [40],[41] compared children's effectiveness, efficiency and satisfaction using TUI and CUI jigsaw puzzles, coding each child's sequence of epistemic and pragmatic complementary actions on puzzle pieces. They found that TUI for both puzzle pieces and table edges afforded more illustrations of epistemic problems of space exploration such as grouping edge pieces, adding to the development of mental skills. Moreover, a positive correlation was found between successful puzzle completion and the number of times pieces were manipulated. Similarly, it was stated by [42] that particular gestures or actions on tangibles could offload some part of the cognitive process to actions in the world, aligning with complementary actions. Thus, these actions could have the possibility to support

Abrar Almjally

memory, develop perception or simplify cognitive computation needed to solve a task. Further research is required to investigate how to design the mapping between action, objects and digital representation to enhance the offloading of different kinds of cognitive tasks and improve cognitive tasks such as learning[8].

3.4 Different TUI APPLICATIONS

There have been some previous studies comparing TUIs with GUIs for children's programming (e.g. [6]), which suggest that TUIs are more appealing to children under 16 years old. The results showed that both interface types are equally easy to use and that there is not much difference in terms of users' performance and learning gains. However, [1] found that TUIs are more enjoyable, attractive, support collaboration, and have a positive effect at the initial stages of learning. Additionally, in another study mentioned in [1] where video recording was used to track time taken to complete tasks and the number of errors, tasks were completed more quickly, except for the oldest children (11 to 12) and there were fewer errors with more successful debugging using TUIs [6]. Gender preference results have so far been inconclusive. Some studies found that TUIs seemed to attract both girls and boys equally while GUIs appealed more to young boys. Another study found that girls considered TUIs more attractive [6]. Therefore, more investigation is needed of the influence of gender on children's interface preferences and their ability to learn computing concepts. One of the few experiments addressing the learning benefits of TUIs and GUIs with young children from 4 to 6 the pilot study described in [43]. The researchers found an association between interfaces and learning outcomes, but the result was not statistically significant, and they did not use a programming blocks representation. The pilot study did not look for other learning benefits such as collaboration, engagement. Theories of embodied interaction can be used to frame and inform educational tools in terms of understanding children's interactions with their surroundings and how these tools can support such interactions [3]. Even though research has found positive results from the manipulation of TUIs in problem-solving and reading comprehension [3], there is limited research that examines the cognitive advantages of TUIs compared with GUIs in terms of learning abstract concepts [3].

4. RESEARCH APPROACH &METHODOLOGY

The first study will use a between-subjects design, with pre- and post-test measures. In order to eliminate confounding factors as far as possible, functionally identical TUI and GUI environments will be designed. The experimental methodology will make use of both qualitative and quantitative

methods. The quantitative methods will focus on system usability and user performance data, such as the manipulation rate, the error rate and the number of times children ask for help. These will all help to indicate the effect of the two interfaces on overall performance. Data will also be collected on enjoyment and attitude towards computing. Spontaneous gesture scores will be collected by counting and cataloguing the number of times each child uses gestures while completing the task. Qualitative methods will also be used, such as observation and video recording of children's interaction with the two interfaces These methods should provide more insight on user engagement and children's use of spontaneous gestures. Data will be obtained from children's discussions and general behaviours observed during the session. Overall evaluation of any changes in children's learning outcomes will be measured using a pre-and post-test.

5. DISSERTATION STATUS

By the time of the conference, I will have submitted the literature review and the official proposal of the thesis. I will be finalizing the design of the TUI and GUI environment prototype to use in my first study and will have made an application for ethical review.

6. EXPECTED CONTRIBUTIONS

This work will give insight into how TUIs might provide benefits for children's learning and motivational outcomes. The research aims to provide empirical evidence, which might explain how and TUIs or GUIs could be more beneficial in a school setting. Additionally, the work will contribute to knowledge of how to design interfaces that are optimized for both learning and engagement. Finally, the research will add to the literature on the use of spontaneous gestures in computer science education.

7. ETHICS

The evaluation of the study will be approved by the University of Sussex Sciences & Technology Cross-Schools Research Ethics Committee (SCITEC C-REC).

8. REFERENCES

- [1] L.-K. . Soh, A. . Samal, S. . Scott, S. . Ramsay, E. . Moriyama, G. . Meyer, B. . Moore, W. G. . Thomas, and D. F. . Shell, "Renaissance computing: An initiative for promoting student participation in computing," SIGCSE'09 - Proc. 40th ACM

Abrar Almjally

- Tech. Symp. Comput. Sci. Educ., pp. 59–63, 2009.
- [2] D. C. Webb, A. Repenning, and K. H. Koh, "Toward an emergent theory of broadening participation in computer science education," Proc. 43rd ACM Tech. Symp. Comput. Sci. Educ. - SIGCSE '12, p. 173, 2012.
- [3] Carnegie Mellon University, "Alice – Tell Stories. Build Games. Learn to Program.," Alice 2.X, 1999. [Online]. Available: <http://www.alice.org/>. [Accessed: 02-Nov-2017].
- [4] MIT, "Scratch," MIT, 2003. [Online]. Available: https://wiki.scratch.mit.edu/wiki/Development_of_Scratch_1.0. [Accessed: 31-Jan-2018].
- [5] P. Henriksen and M. Kolling, "Greenfoot," Companion to 19th Annu. ACM SIGPLAN Conf. Object-oriented Program. Syst. Lang. Appl. - OOPSLA '04, p. 73, 2004.
- [6] T. Sapounidis and S. Demetriadis, "Educational robots driven by tangible programming languages: A review on the field," Advances in Intelligent Systems and Computing, vol. 560. Springer Verlag, pp. 205–214, 2017.
- [7] A. Manches and L. Plowman, "Computing education in children's early years: A call for debate," Br. J. Educ. Technol., vol. 48, no. 1, pp. 191–201, Jan. 2017.
- [8] A. N. Antle and A. F. Wise, "Getting down to details: Using theories of cognition and learning to inform tangible user interface design," Interact. Comput., vol. 25, no. 1, pp. 1–20, Jan. 2013.
- [9] F. Heintz, L. Mannila, and T. Farnqvist, "A review of models for introducing computational thinking, computer science and computing in K-12 education," Proc. - Front. Educ. Conf. FIE, vol. 2016–Novem, pp. 1–9, Oct. 2016.
- [10] M. Bers, "The TangibleK Robotics Program: Applied Computational Thinking for Young Children," Early Child. Res. Pract., vol. 12, no. 2, 2010.
- [11] G. Elsevier Science (Firm), G. Fessakis, E. Gouli, and E. Mavroudi, "Problem Solving by 5-6 Years Old Kindergarten Children in a Computer Programming Environment: A Case Study," Comput. Educ., vol. 63, no. April 2013, pp. 87–97, 2013.
- [12] J. Robertson, A. Manches, and H. Pain, "It's Like a Giant Brain With a Keyboard': Children's Understandings About How Computers Work," Child. Educ., vol. 93, no. 4, pp. 338–345, 2017.
- [13] M. U. M. U. Bers, L. Flannery, E. R. E. R. Kazakoff, and A. Sullivan, "Computational thinking and tinkering: Exploration of an early childhood robotics curriculum," vol. 72, pp. 145–157, Mar. 2014.
- [14] J. M. Wing, "Computational thinking," Commun. ACM, vol. 49, no. 3, 2006.
- [15] V. Barr and C. Stephenson, "Bringing computational thinking to K-12," ACM Inroads, vol. 2, no. 1, p. 48, Feb. 2011.
- [16] C. Artym, M. Carbonaro, and P. Boechler, "Evaluating pre-service teachers' computational thinking skills in scratch," Ubiquitous Learn., vol. 10, no. 2, 2017.
- [17] S. Grover and R. Pea, "Computational Thinking in K-12: A Review of the State of the Field," Educ. Res., vol. 42, no. 1, pp. 38–43, 2013.
- [18] S. Psycharis and M. Kallia, "The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving," Instr. Sci., vol. 45, no. 5, pp. 583–602, 2017.
- [19] A. Repenning, A. Basawapatna, and N. Escherle, "Computational thinking tools," in Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC, 2016, vol. 2016–Novem.
- [20] V. J. Shute, C. Sun, and J. Asbell-Clarke, "Demystifying computational thinking," Educ. Res. Rev., vol. 22, pp. 142–158, 2017.
- [21] S. P. Jones, COMPUTING AT SCHOOL Computing A CPD toolkit for primary teachers. 2014.
- [22] Royal Society, After the reboot : computing education in UK schools. 2017.
- [23] A. Melchior, F. Cohen, T. Cutter, T. Leavitt, and F. Manchester, "More than Robots: An Evaluation of the FIRST Robotics Competition Participant and Institutional Impacts," 2005.
- [24] A. Merkouris, K. Chorianopoulos, and A. Kameas, "Teaching programming in secondary education through embodied computing platforms: Robotics and wearables," ACM Trans. Comput. Educ., vol. 17, no. 2, 2017.
- [25] N. Rusk, M. Resnick, R. Berg, and M. Pezalla-Granlund, "New pathways into robotics: Strategies for broadening participation," J. Sci. Educ. Technol., vol. 17, no. 1, pp. 59–69, 2008.
- [26] M. McLean and D. Harlow, "Designing

Abrar Almjally

- inclusive STEM activities: A comparison of playful interactive experiences across gender,” in IDC 2017 - Proceedings of the 2017 ACM Conference on Interaction Design and Children, 2017.
- [27] C. Kazimoglu, M. Kiernan, L. Bacon, and L. Mackinnon, “A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming,” *Procedia - Soc. Behav. Sci.*, vol. 47, pp. 1991–1999, Jan. 2012.
- [28] A. Repenning, D. C. Webb, K. H. Koh, H. Nickerson, S. B. Miller, C. Brand, I. H. M. Horses, A. Basawapatna, F. Gluck, R. Grover, K. Gutierrez, and N. Repenning, “Scalable game design: A strategy to bring systemic computer science education to schools through game design and simulation creation,” *ACM Trans. Comput. Educ.*, vol. 15, no. 2, 2015.
- [29] S. Atmatzidou and S. Demetriadis, “Advancing students’ computational thinking skills through educational robotics: A study on age and gender relevant differences,” *Rob. Auton. Syst.*, vol. 75, pp. 661–670, 2015.
- [30] S. Grover, R. Pea, and S. Cooper, “Designing for deeper learning in a blended computer science course for middle school students,” *Comput. Sci. Educ.*, vol. 25, no. 2, pp. 199–237, Apr. 2015.
- [31] L. Werner, J. Denner, S. Campe, and D. C. Kawamoto, “The fairy performance assessment: Measuring computational thinking in middle school,” in *SIGCSE’12 - Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 2012.
- [32] T. Sapounidis, S. Demetriadis, and I. Stamelos, “Evaluating children performance with graphical and tangible robot programming tools,” *Pers. Ubiquitous Comput.*, vol. 19, no. 1, pp. 225–237, Jan. 2015.
- [33] M. S. Horn, R. J. Crouser, and M. U. Bers, “Tangible interaction and learning: the case for a hybrid approach,” *Pers. Ubiquitous Comput.*, vol. 16, no. 4, pp. 379–389, Apr. 2012.
- [34] T. Sapounidis and S. Demetriadis, “Tangible versus graphical user interfaces for robot programming: exploring cross-age children’s preferences,” *Pers. Ubiquitous Comput.*, vol. 17, no. 8, pp. 1775–1786, Dec. 2013.
- [35] A. Pugnali, A. Sullivan, and M. U. Bers, “the Impact of User Interface on Young Children’S Computational Thinking,” *J. Inf. Technol. Educ. Innov. Pract.*, vol. 16, no. 16, pp. 171–193, 2017.
- [36] G. W. Fitzmaurice, “Ph.D. Thesis: Graspable User Interfaces (George Fitzmaurice, 1996),” University of Toronto , 1996.
- [37] A. N. Antle, “Embodied child computer interaction: Why embodiment matters,” *interactions*, vol. 16, no. 2, p. 27, Mar. 2009.
- [38] A. Clark, *Being there: putting brain, body, and world together again*. MIT Press, 1997.
- [39] A. Antle, “The CTI framework: informing the design of tangible systems for children,” ... 1st Int. Conf. Tangible ..., pp. 15–17, 2007.
- [40] A. N. Antle, M. Droumeva, and D. Ha, “Hands on what?,” in *Proceedings of the 8th International Conference on Interaction Design and Children - IDC ’09*, 2009, p. 80.
- [41] A. N. Antle, G. Corness, and A. Bevans, “Balancing justice: comparing whole body and controller-based interaction for an abstract domain,” *Int. J. Arts Technol.*, vol. 6, no. 4, p. 388, 2013.
- [42] S. Goldin-Meadow, *Hearing gesture: how our hands help us think*. Belknap Press of Harvard University Press, 2003.
- [43] A. Strawhacker and M. U. Bers, “‘I want my robot to look for food’: Comparing Kindergartner’s programming comprehension using tangible, graphic, and hybrid user interfaces,” *Int. J. Technol. Des. Educ.*, vol. 25, no. 3, pp. 293–319, 2015.