pARt Blocks: Programming with Physical Tangible Blocks and AR

Karolina Trajkovska[1], Klen Čopič Pucihar[1,2], Matjaž Kljun[1,2], Jordan Aiko Deja[1,3] and Maheshya Weerasinghe[1,4]

[1]University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies, Koper, Slovenia
[2]Faculty of Information Studies, Novo Mesto, Slovenia
[3]De La Salle University, Manila, Philippines
[4]University of St Andrews, St Andrews, United Kingdom

Abstract

Computer programming is a demanding task requiring users to understand a syntax of a programming language, logic flows, and complex abstract concepts. Adult users commonly start learning how to program in a text-based programming environment. However, such tools are not optimal for children as they require understanding of a high level of abstraction. Instead, visual programming languages were developed to hide the syntax and error messages, but are still capable to teach concepts such as parallelism and event handling. Despite, these languages still require children to handle virtual block-like elements on a computer screen. In this research, we examine the feasibility of teaching young children programming using physical blocks coupled with Augmented Reality (AR). To this end, we conducted a between subject design study with 8 participants. We compared a traditional 2D desktop application for visual programming to a 3D setting with tangible physical programming blocks coupled with a Head Mounted Device displaying programming results as AR content. The preliminary results show that the immersive 3D learning environment scored better in mental effort, task completion time, performance and subjective user satisfaction. The results should be validated with a larger sample size.

Keywords

learning, mixed reality, tangible interfaces, programming environments

1. Introduction and Background

The most common and time-efficient approach to programming is using high-level text based programming languages (PL). However, understanding the meaning behind written lines of code is difficult for beginners and in particular for children. Learning programming concepts can be made easier with visual representation of the syntax. First, visuals are known to enhance learning [1], and second, visuals can hide syntax and error messages, allowing users to focus on programming concepts. Such programming languages are called visual programming languages (VPL).

"Scratch" [2] is one of the popular VPL. It features a block paradigm as a programming interface, which inspired several subsequent projects and studies [3, 4]. In block programming

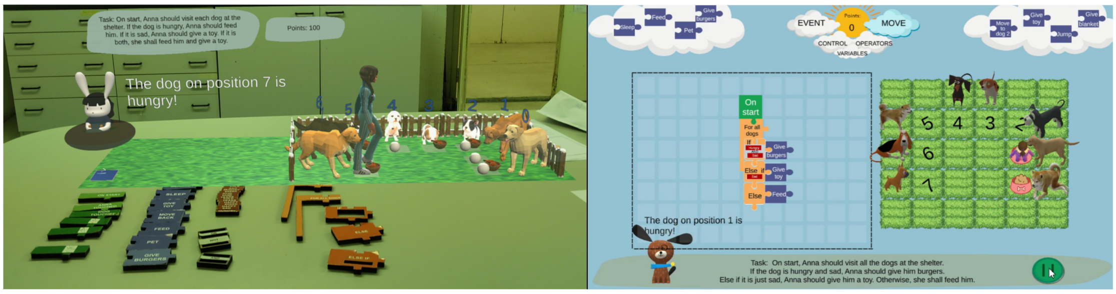
Human-Computer Interaction Slovenia 2022, November 29, 2022, Ljubljana, Slovenia
trajkovskakarolina@gmail.com (K. Trajkovska); klen.copic@famnit.upr.si (K. Čopič Pucihar); matjaz.kljun@upr.si (M. Kljun); jordan.deja@famnit.upr.si (J. A. Deja); maheshya.weerasinghe@famnit.upr.si (M. Weerasinghe)

interface the programming syntax of a high-level programming language is replaced by blocks that can be joined only in ways that are syntactically correct, and their incompatibility can be used to infer errors [5]. In addition, hiding the syntax has been shown to decrease cognitive load [6]. Nevertheless, programming still happens using traditional input methods including the display, a mouse and a keyboard.

Moving beyond traditional interfaces, researchers explored novel programming systems for children with tangible interfaces where programming results with the physical blocks or cards are either visible on the screen [7] or in the augmented reality (AR) [8, 4]. Multi sensory experiences of these systems can include sight, sound, touch, and smell, which help children to encode more information in their brains while learning [7] and consequently better support recall. However, to the best of our knowledge, no study explored how tangible block programming interface could be augmented with a Head Mounted Display (HMD) showing the results of users' actions in the physical environment of actual blocks. To address this, we have implemented *pARt Blocks*, an AR application for learning **p**rogramming by combining **AR** and **t**angible physical **blocks** [9].

2. pARt Blocks Concept and Prototype

The key features of the application developed are the *blocks*, a *programming field* and a *game field* for two different applications we planned to compare – Tangibles+AR and Desktop – as seen in Figure 1. These elements are based on the prior work from Weerasinghe et al. [10].



(a) (b)

Figure 1: pARt Blocks (a) Tangibles+AR and (b) Desktop prototypes. Both show the *blocks*, a *programming field* (where users program with blocks), and a *game field* (where the results of programming are visible).

The idea is to visualise the execution of the program in the *game field* by linking the *blocks* in the *programming field* to complete a task. The task involves the character Anna that has to interact with dogs. At the beginning the dogs are in one of these states: sad, hungry or both. Users need to write a program that moves Anna around the field from dog to dog and either feed, pet or perform both actions on dogs. To complete the task all dogs should turn in the *happy* state. The learning concepts consist of *events*, *loops* and *conditional assignments*. Accordingly, we have designed the blocks as described in Table 1.

Both applications were designed in Unity. In addition, the Tangibles+AR application uses Vuforia Engine to recognise tangible blocks and was deployed to Hololens 2.0 HMD. In both

Table 1
Taxonomy of programming blocks and their purpose.

Block Type	Color	Purpose
Event	Green	Specify when something should occur.
Control	Orange	Create loops and if/else expressions.
Operator	Grey	Specify the logical and comparison operators.
Variables	Red	Replicate states of the dogs. Predefined to reduce complexity.
Movement	Blue	Define the movements/actions of the character Anna.

applications the same block types are used. In Desktop application users need to drag-and-drop blocks, while in the Tangibles+AR application users interact with physical blocks. The physical blocks were laser-cut from wood. The two applications also differ in presentation. For Desktop application we used a standard 2D presentations for the game field as is common in other visual programming languages, while for the Tangibles+AR application we use a 3D game field as seen in Figure 1.

Both applications have a play button; when pressed and if the program is correct, the execution starts. Otherwise, the character Anna reports an unsuccessful attempt. The task was the same for both conditions. Users had to write a program with blocks that makes the character Anna traverse through the abstracted array – i.e. the line of dogs.

3. Research Method and Initial Results

We developed two conditions based on each application: Desktop and Tangibles+AR condition. Our goal was to determine which would prove better in teaching kids how to program.

At the beginning of the study participants had to read and sign a consent form. After a brief introduction to programming with blocks and using the application they were asked to complete the task. Next, the NASA Task Load Index (NASA-TLX) [11] questionnaire was completed by participants in order to measure the mental effort invested. Participants were also asked to complete a post-test questionnaire to assess their performance, a System Usability Scale (SUS) [12], and User Experience Questionnaire (UEQ) [13]. At the end, participants completed a post-questionnaire about user demographics and prior experience with programming, AR and HMD.

The research used a between-subject design and the study took between 30 and 45 minutes for each condition. Both conditions included four (4) participants. They had no programming experience beforehand. The participants were aged between 10 and 24 years, with the mean of $\overline{x} = 17.1$ and $SD = 4.82$ (and with a 2:6 female to male ratio).

The preliminary results show that the Tangibles+AR condition required less mental effort to complete the task, resulted in higher performance score, lower time needed to complete the task, and higher SUS score (Table 2). The results of the UEQ questionnaires are shown in Figure 2. The Tangibles+AR application scored higher again in all dimensions. However, we are unable to draw any significant conclusions regarding the effectiveness of the learning process with the present number of participants. In addition, we acknowledge that the 3D interface might have affected learning as well as the results. This should be investigated further by comparing the same game field in both desktop and AR interfaces (either both 2D or 3D) and exploring possible effects of a 3D game field.

Table 2
Descriptive statistics of desktop and AR conditions for all dependant variables.

Metrics	Conditions	mean	median	stdev	min	max
Mental Effort	Desktop	34.4	37.1	10.03	20.0	43.3
(NASA-TLX Scores)	AR	27.7	32.1	9.61	13.3	33.3
% Task Score	Desktop	67.5	85.0	47.17	0	100.0
	AR	85.0	85.0	17.32	70	100.0
Time in seconds	Desktop	490.2	600.0	219.54	160.9	600.0
	AR	474.5	512.5	157.64	273.0	600.0
SUS Score	Desktop	66.3	70.0	17.38	42.5	82.5
	AR	80.0	82.5	18.71	55.0	100.0

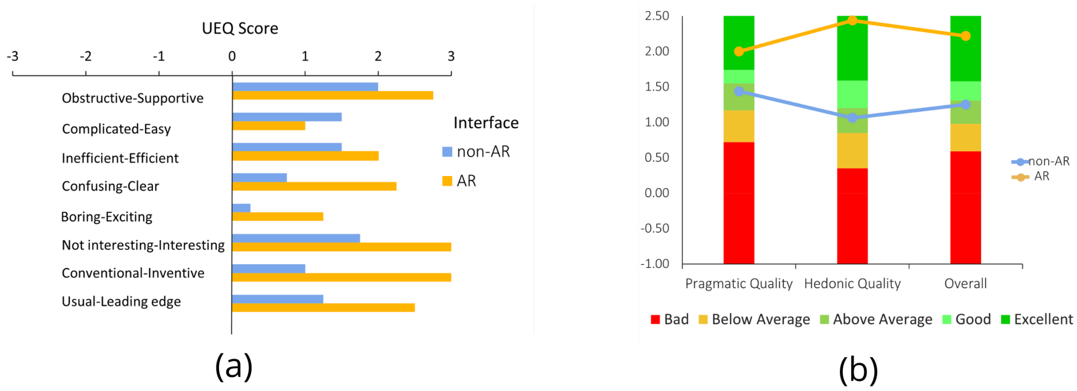


Figure 2: User experience questionnaire: (a) Mean values per item/question; (b) Mean values of UEQ factors (pragmatic and hedonic) and overall.

4. Discussion and Conclusion

The *pARt Blocks* prototype was developed to explore the potential of tangible interfaces and augmented reality (AR) technology in teaching programming. We have built two prototypes (Desktop and Tangibles+AR) and conducted initial evaluation. Due to time constraints, we have been unable to complete the entire study, which is our next objective. As the students are back in school, recruiting and testing individuals of the desired age group will be easier. The preliminary results of the study are promising; however, still incomplete. According to current data, both techniques support creativity and are helpful when learning how to program. However, the tangible-AR prototype indicated this to a greater extent.

In the future we plan to make some adjustments to our applications (using the same game field in both) and test more participants. In addition, we plan to expand our prototypes with aspects of collaborative programming, dynamic implementations and personalised guidance systems for assisting users in learning.

References

[1] R. E. Mayer, Multimedia learning, in: Psychology of learning and motivation, volume 41, Elsevier, Radarweg 29, 1043 NX Amsterdam, The Netherlands, 2002, pp. 85–139.

[2] MIT, Scratch developers, 2006. URL: https://scratch.mit.edu/.

[3] J. M. S. III, G. Nodalo, J. Valenzuela, J. A. Deja, Explore, edit, guess: Understanding novice programmers' use of codeblocks for regression experiments, in: Proceedings of the 6th HCI Slovenia Conference 2021, 2021, pp. 3–17.

[4] I. Radu, B. MacIntyre, Augmented-reality scratch: a children's authoring environment for augmented-reality experiences, in: Proceedings of the 8th International Conference on Interaction Design and Children, 2009, pp. 210–213.

[5] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, et al., Scratch: programming for all, Communications of the ACM 52 (2009) 60–67.

[6] R. Pelánek, T. Effenberger, Design and analysis of microworlds and puzzles for block-based programming, Computer Science Education 32 (2022) 66–104.

[7] G. Futschek, J. Moschitz, Learning algorithmic thinking with tangible objects eases transition to computer programming, in: International conference on informatics in schools: Situation, evolution, and perspectives, Springer, 2011, pp. 155–164.

[8] B. Cleto, J. M. Moura, L. Ferreira, C. Sylla, Codecubes-playing with cubes and learning to code, in: Interactivity, Game Creation, Design, Learning, and Innovation, Springer, 2018, pp. 538–543.

[9] K. Trajkovska, J. A. Deja, M. Weerasinghe, pARt blocks: Programming in AR with tangible blocks, in: Poster Presentations of the 9th ACM Celebration of Women in Computing womENcourage, 2022, pp. 1–2. doi:10.13140/RG.2.2.10986.59846.

[10] M. Weerasinghe, A. Quigley, J. Ducasse, K. Čopič Pucihar, M. Kljun, Educational Augmented Reality Games, Springer International Publishing, 2019, pp. 3–32. doi:10.1007/978-3-030-15620-6_1.

[11] S. G. Hart, Nasa-task load index (nasa-tlx); 20 years later, in: Proceedings of the human factors and ergonomics society annual meeting, volume 50, 2006, pp. 904–908.

[12] J. Brooke, et al., Sus-a quick and dirty usability scale, Usability evaluation in industry 189 (1996) 4–7.

[13] M. Schrepp, Ueq-user experience questionnaire, 2019. URL: https://www.ueq-online.org/.