2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) | 978-1-6654-8402-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/VRW55335.2022.00182

A Tangible Augmented Reality Programming Learning Environment for Textual Languages

Dmitry Resnyansky*, Mark Billinghurst, Gun Lee

*School of Information Technology and Mathematical Sciences, University of South Australia, Mawson Lakes SA 5095, Australia

Abstract

We present a novel Tangible Augmented Reality Programming Learning Environment system that uses a head-mounted display (HMD) and physical manipulatives for teaching an Object-Oriented Programming (OOP) language. The system supports student understanding/recollection of terms, and construction of statements by enabling access to code components, terminology, and programming hints. It is designed to use the affordances of Augmented Reality (AR) and Tangible User Interfaces (TUIs) to provide a virtual workspace encouraging natural interaction with learning material. An interactive AR code template for physical manipulatives provides a building and testing environment for learners to practice statement construction and computational skills. The system bolsters active learning with a localised AR program visualisation and HMD-anchored AR glossary.

Index Terms: Applied computing—Education—Interactive learning environments; Human-centered computing—Interaction design

1 Introduction

A design of a Tangible Augmented Reality Programming Learning (PL) Environment (TARPLE) that uses a head-mounted display (HMD) and physical manipulatives for teaching basic Object-Oriented Programming (OOP) concepts to novice senior-secondary and junior-tertiary students is presented. TARPLE seeks to address the need to: create a learning environment that encourages active learning; and support a Computer Science Education (CSE) model of learning. According to a CSE-specific taxonomy of learning objectives [1], PL environments need to support multiple learning paths that traverse the *Producing* (*Coding*) and the *Learning* (*Explaining*, *Interpreting*) axes [2]. AR and ARTUI technologies support active learning [3] through provision of explanatory contexts, guidance, and feedback [2, 4].

AR has been effectively used to support instruction and learning [5]. Different AR methods' effectiveness and cognitive load experienced by users are determined by the task type and the instructional design. AR may enhance CSE by supporting visualisation of abstract concepts and complex dynamic processes [6] and interactivity with learning content [2]. In industrial settings, tangible block-based programming languages enable end-users to operate robots by including augmented annotation of objects, regions, and physical locations, without the need for expertise in advanced programming techniques [7]. AR GUI approaches can effectively encourage development of computational thinking skills in young learners [8], while customised tangible programming languages enable them to operate programs or robotic devices [2, 4]. However, the ease of transition to text-based languages remains an open question [9]. Conventional text-based languages taught using the combined affordances of AR and TUI mediums [6] as well as learning environments that bring the programming and learning dimensions together in a natural way are still uncommon.

TARPLE suggests a novel design that combines *programming* and *learning* dimensions through block-based TUI, marker-based, and localised (markerless) AR content and allows intuitive control over material by organising information spatially [10], which may help to reduce incidental cognitive load [11]. The system design outlined in this work seeks to address the following questions: 1) *how to structure the learning space; 2) how to enable easy switching between* Programming *and* Learning *activities; 3) how to scaffold learning yet also enable active explorative learning; 4) how to decrease cognitive load.*

2 System Design

TARPLE uses a Microsoft HoloLens2 HMD with sensors to track users' head and hand positions. Block markers with Vuforia tracking images enable natural interaction with virtual UI and display of marker-based augmentations inside the HMD optical system. The HMD's device pose tracking and optical capabilities allow users to interact with locally-anchored AR content and block-anchored augmented annotations through the HMD's expanded FoV. The system design comprises four key elements: a TUI-embedded AR annotation; localised AR; AR program visualisation; and HMD-anchored (AR) glossary.

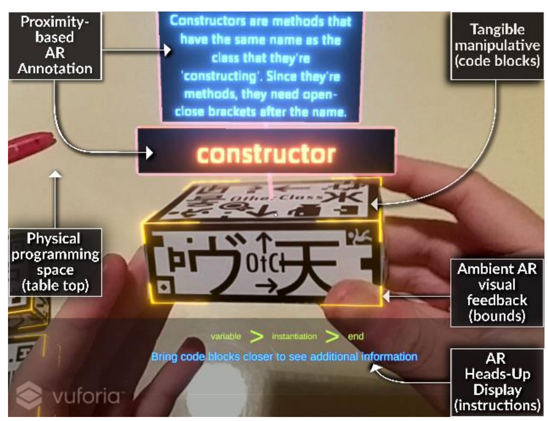

Figure 1: The *learning* and *programming* activity spaces supported by the TARPLE are shown in-situ, with Vuforia markers.

The TUI supports the *programming* activity by providing a natural interaction method for inputting, revising, fixing code, and viewing concurrent virtual information (Fig. 1). A text label with a part of a statement is provided on block sides to progressively develop independent programming skills in learners. Code blocks allow learners to switch between the AR-supported learning and tangible programming activities, and contextualise abstract information within the physical programming scenario. Active interaction with the learning content through block-anchored AR annotation provides access to three levels of information: the code

* dmitry.resnyansky@mymail.unisa.edu.au

978-1-6654-8402-2/22/$31.00 ©2022 IEEE
DOI 10.1109/VRW55335.2022.00182

component name; the component definition using programming-specific terminology; and detailed descriptions of the code component's meaning and explicit hint to its placement (Fig. 1).

Active learning is also supported by an AR workspace (Figs. 2, 3) with interactive visualisation assets (HMD-anchored glossary; localised virtual code template, program visualisation (PV), and progress indicator) that serve as additional guides for learners for constructing their program. Physical manipulatives are placed into the template slots, and a simplified parsing mechanism activated by a virtual button checks the statement and displays the result as a textual and visual cue. Immediate visual feedback is provided by colour-coding individual slots based on the state of the construction.

A static PV depicts the structure and meaning of the programming statement in a non-sequential form (Fig. 3). Static visualisations allow interaction with knowledge embodied inside the PV at learners' own pace and with minimal cognitive load [12]. Non-sequential deconstruction of the statement is designed to stimulate curiosity and solidify it by including an interactive, proximity-based annotation system. An interactive node-based "astronomy" metaphor is used for contextualising instantiation concepts, with the "planets" denoting primary concepts and "satellites" that orbit the planets indicating secondary concepts (expressions, operators, etc.) that bridge the primary concepts.

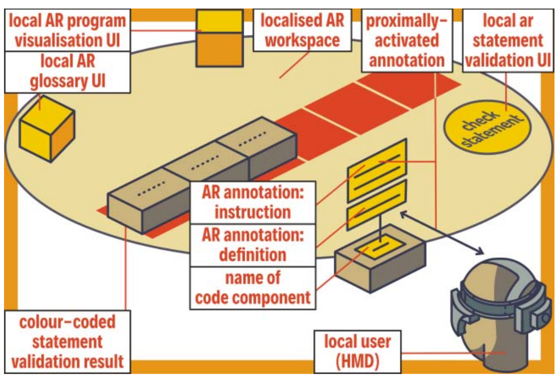

Figure 2: Schematic of the system's TUI with interactive AR tooltips and validation engaged in a statement construction scenario.

Figure 3: Localised AR components: contextual user-anchored glossary of programming terms; and static PV.

Additional information on relevant programming terminology used in AR tooltips for code blocks is provided as text cues in a gesture-toggled glossary (Fig. 3), provides contextual knowledge to learners.

3 Conclusion and Future Work

This project contributes to the development of PL environments by suggesting how TUI and AR (HMD) may support a CSE-specific model of PL [1, 2] (as a two-dimensional activity), and implement advanced pedagogical theories and principles (active guided learning; embodied learning; spatial learning; scaffolding; gamification; feedback) while considering the cognitive load aspect. The suggested system design situates and contextualises OOP concepts within the learners' physical environment through HMD-supported AR and tangible manipulatives. Key components of the system design include: block-anchored AR annotation; localised AR program visualisation; HMD-anchored AR glossary; and AR- and TUI-based interaction. The two-dimensional structure of the PL environment and the need to support active guided learning by embracing affordances of AR and TUIs determined the choice of a HMD (at this stage) as a method that best corresponds to the complexity of the learning content and activity, and a medium that may decrease cognitive load by enabling natural interaction with learning materials. Future work will involve a field study for comparative evaluation of HHD and HMD prototypes in terms of user experience, performance, cognitive load, and engagement.

References

[1] U. Fuller, C. G. Johnson, T. Ahoniemi, D. Cukierman, I. Hernán-Losada, J. Jackova, E. Lahtinen, T. L. Lewis, D. McGee Thompson, C. Riedesel and E. Thompson. Developing a computer science specific learning taxonomy. *ACM SIGCSE Bulletin*, 39(4): 152–170, December 2007.

[2] D. Resnyansky, M. Billinghurst and A. Dey. An AR/TUI-supported debugging teaching environment. In *Proc. OZCHI'19*, pp. 590–594. ACM, New York, 2019.

[3] A. N. Antle and A.F. Wise. Getting down to details: Using theories of cognition and learning to inform tangible user interface design. *Interacting with Computers*, 25(1), 1–20, 2013.

[4] A. Theodoropoulos and G. Lepouras. Augmented reality and programming education: A systematic review. *International Journal of Child-Computer Interaction*, 100335, December 2021.

[5] J. Buchner, K. Buntins, and M. Kerres. The impact of augmented reality on cognitive load and performance: A systematic review. *Journal of Computer Assisted Learning*, 2021. https://doi.org/10.1007/s10639-021-10445-y.

[6] C. De Raffaele, S. Smith and O. Gemikonakli. Teaching and learning queueing theory concepts using tangible user interfaces. In *Proc. 2016 IEEE TALE*, pp. 194–201. IEEE, 2016.

[7] M. Kapinus, V. Beran, Z. Materna and D. Bambušek. Spatially situated end-user robot programming in augmented reality. In *Proc. RO-MAN*, pp. 1–8. IEEE, 2019.

[8] K. Larson and R. Chambers. AR in the computer programming classroom: A review of the literature. In *2020 IEEE TALE*, pp. 436–443. IEEE, 2020.

[9] M. Dorling and D. White. Scratch: A way to Logo and Python. In *Proc. SIFCSE'15*, pp. 191–196. ACM, New York, 2015.

[10] G. E. Baykal, I. V. Alaca, A. E. Yantaç and T. Göksun. A review on complementary natures of tangible user interfaces (TUIs) and early spatial learning. *Int. J. Child-Computer Interaction*, 16: 104–113, June 2018.

[11] Chiang, T. H., Yang, S. J., & Hwang, G.-J. (2014). An augmented reality-based mobile learning system to improve students' learning achievements and motivations in natural science inquiry activities. *J. Educational Technology & Society*, 17(4), 352–365.

[12] A. B. Chikha, A. Khacharem, K. Trabelsi and N. L. Bragazzi. The effect of spatial ability in learning from static and dynamic visualizations: A moderation analysis in 6-year-old children. *Front. Psychol.,* 12(583968), June 2021.