

# Logibot: Investigating Engagement and Development of Computational Thinking Through Virtual Reality

Robert Sims  
School of Computing and  
Communications  
Lancaster University  
Lancaster, United Kingdom  
r.sims2@lancaster.ac.uk

Nathan Rutherford  
School of Computing and  
Communications  
Lancaster University  
Lancaster, United Kingdom  
NathanRutherford333@outlook.com

Prashanthi Sukumaran  
School of Computing and  
Communications  
Lancaster University  
Lancaster, United Kingdom  
PrashanthiSukumaran@gmail.com

Nikola Yotov  
School of Computing and  
Communications  
Lancaster University  
Lancaster, United Kingdom  
NikolaYotov109@gmail.com

Thomas Smith  
School of Computing and  
Communications  
Lancaster University  
Lancaster, United Kingdom  
ThomasSmithLU57@gmail.com

Abhijit Karnik  
School of Computing and  
Communications  
Lancaster University  
Lancaster, United Kingdom  
a.karnik@lancaster.ac.uk

**Abstract**—In this study we assess the effects of teaching fundamental programming concepts through a virtual reality (VR) block-based programming application and its impact on engagement. As a comparison study, participants played an existing desktop-based game (LightBot) and the developed VR game (LogiBot) covering similar gameplay mechanics and block-based programming. Initial results indicate that traditional desktop applications are currently more engaging than VR for teaching programming. We thus identify the need for careful design of interaction methods to support ease of use and reward factors to promote engagement in VR-based learning applications beyond the initial wow-factor.

**Index terms**—Virtual Reality, Engagement, Education, Visual Programming

## I. INTRODUCTION

A key parameter in learning is engagement. While e-learning tries to trigger psychological engagement through behavioural activities, such as selecting an answer from a list, this is not always successful [1]. On their own, traditional e-learning methods based purely on the dissemination of information are not enough to enable knowledge acquisition for learners. Instead, e-learning needs to provide an experience that participants can learn from.

Virtual Reality (VR) technologies offer a potential solution to this problem. This idea is supported by early research showing that VR provides students with successful experiences in a compelling environment [2], and that the experience of immersion offered by VR makes it an ideal solution for use in education [3]. There is evidence showing that users (particularly children) are more likely to engage with materials that offer a tangible object that they can interact with [4].

Learning programming as a key component in computer literacy is gaining traction within the education community. Consequently, institutions are attempting to identify methods for teaching programming to children. One such established method

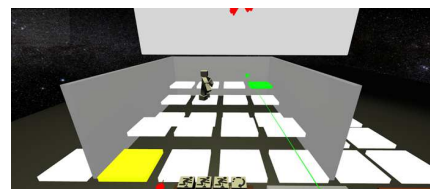


Fig. 1. Example level in LogiBot that requires a combination of command blocks used procedurally to complete the level.

[5] is using visual programming tools, such as Scratch [6], as an introduction to computational thinking. Other approaches use gamification [7] to apply game design elements to the process of learning to program/code. A prominent example of this is LightBot [8]. However, most current research only applies these models to children and novice learners and as a result, there is a gap in the literature in seeing how these concepts can also be taught to adult novices or those who identify as conversational programmers [9].

In this paper, we explore engagement through VR-based learning tools by contributing LogiBot (Fig. 1), a VR artefact that aims to teach programming concepts through gamification. We conducted a comparison study comparing LogiBot against another block-based programming learning environment, LightBot. This allowed us to investigate how this process transfers to adult novice learners as well as provide an application of the User Engagement Survey (UES) to LogiBot and LightBot.

## II. BACKGROUND

### A. Pedagogy

Constructivism views learning as an active building process based on experience, rather than a passive one, where information is transmitted from teacher to student [10]. However, one study [11] notes that students being taught introductory programming, found this approach "overwhelming and intimidating at times". Rahman et al [12] also note how such

students may feel demotivated when carrying out self-study exercises alone. The key to addressing this demotivation is to build in high levels of engagement with any self-study activities, including technology-based activities. We follow the definition of engagement as proposed by O'Brien and Toms [13]. They indicate that engagement is a process consisting of four stages; a point of engagement, sustained engagement, disengagement, and re-engagement, as well as attributes that relate to the quality of user experience such as endurance (the user will remember the experience), attention and feedback. They used this early research to develop the User Engagement Survey [14] as a measure of user engagement.

### B. Computing Education

As a key component of the National Curriculum in the United Kingdom, programming forms one of the key aims, with the ability to apply and understand fundamental concepts and principles including logic, algorithms and data representation detailed as critical components [15]. Teaching programming is a challenging task with syntax often being an early barrier to overcome. Teaching methods used to introduce programming often try to separate the syntax or language specific functions from more general logical reasoning. One approach to teaching programming to novices, particularly children, is visual, block-based programming languages such as Scratch [6]. Visual programming environments and games such as Scratch and LightBot are already well established as effective in teaching fundamental programming concepts [16]. Kalelioğlu and Gülbahar [17] identified that such tools are particularly effective in developing students' confidence in problem solving tasks. In a study of robot programming techniques, Weintrop, Afzal and Francis et. al [16] establish that block-based programming enables programming novices to quickly build complex programs with no loss of accuracy versus traditional text-based programming. Programming environments leveraging a block-based model support scientific enquiry, such as three-dimensional simulations programmed in AgentCubes [18]. This adoption of block-based programming is supported by the proliferation of learning spaces that attempt to teach programming via a block-based approach [8], [19], [20] which attempt to make programming accessible to a wider audience.

### C. VR in Education

Interactive and tangible objects within VR have been shown to promote engagement and when coupled with existing game-based learning provide an environment where effective learning can take place [21]. While not a VR application, this idea was explored by Melcer and Isbister [22], finding that tangibles had a positive impact on interest and enjoyment of programming. Bell et al [23] provides a comprehensive evaluation of VR as an educational tool. They point out how VR also provides the opportunity to improve on teaching methods where 'traditional' methods may not be as effective. Based on Felder and Silverman's [24] learning styles for example, they show how VR can help sensory learners by presenting "tangible representation of abstract concepts". Concurrently, falling costs of VR devices has lowered the entry barrier for the adoption of these devices into classrooms and as devices in the home. Recent studies [20] have explored the potential and effectiveness of leveraging VR-based teaching, with games being more effective than simulations and virtual worlds in improving learning outcomes.

Systems that use VR for programming education are rare. Chandramouli et al. [25] detail teaching programming concepts to engineering students via a VR system. This system uses the archaic VRML97 specification to provide limited interaction and graphics quality and targets young adults. A virtual 3D programming language for novice users is described by Ortega et al. [26]. As a practical 3D extension of Scratch, it uses a 3D-VPL environment to allow for creating relationships between boxes. Teaching of ordering algorithms is tackled by Grivokostopoulou [27], however it makes assumptions that learners understand the concepts of sequence, iteration and bifurcation thus is not targeted at novices. Computational thinking is commonly encountered in robotics, which Witherspoon et al. [28] explore by developing a series of problems to be solved via a 3D virtual environment game based on previous work by Carnegie Mellon University and the ROBomatter programming curriculum [29]. However, in comparison to existing desktop applications, there is a gap in understanding how the use of VR block-based programming games can help with understanding the foundational concepts of programming. This is the gap we choose to address with LogiBot.

### D. Motivation

We compare an existing block-based programming learning game, LightBot, and another, LogiBot, developed in a similar style to be broadly comparable. We choose to teach programming concepts using the gamification approach championed by LightBot that combines a simplified version of the visual block-based interaction style presented by applications such as Scratch. Our reasoning for this comes down to the philosophy of computational thinking, separating syntax from logical reasoning. We hypothesise: VR will be more engaging versus desktop learning applications (**H1**). The reason for this higher level of engagement will be attributed to the high interactivity with tangible objects within the VR game environment (**H2**).

## III. SYSTEM DESIGN

### A. Platform

Our chosen platform to develop LogiBot on was Unity [7], a cross-platform 3D game engine that supports scripting in the C# language as well as various virtual reality headset SDKs. Within Unity we also used a framework called Virtual Reality Toolkit (VRTK) [30]. VRTK is a general-purpose framework for developing virtual reality games, providing a number of 'out of the box' interactions for motion-tracked virtual reality controllers. We used Unity to develop the game whilst targeting the Oculus Go headset, which builds on top of Android as a development and delivery platform. For the purposes of development, we also used a Samsung Gear VR headset paired with a Samsung Galaxy S7 smartphone, which has the same hardware capabilities and SDK as the Oculus Go.

### B. Implementation

A key design decision was where possible to implement interactions via interactable objects and so minimize the use of GUI based elements (floating menus, start/stop buttons et cetera). As an initial prototype, we resorted to simplified graphical elements and thus the objects within game were not

textured and did not have complex lighting applied. Robot instruction types were defined as "Forward", "Rotate Clockwise", "Rotate Anti-clockwise", "Jump", and "None". Instructions include a 1-second wait between each executed instruction. An example is the forward function, which when executed will simply make the robot move one space forward in the maze. For this initial study, we implemented five hand-programmed levels in the game, declaring the positions of the tiles, walls, and objectives. We considered creating a more dynamic system that could read in level files (in JSON/CSV format), but for the purposes of our prototype, we decided to manually implement each one as it would be more time-efficient for only five levels.

### C. Interaction Design

The 3DoF controls of the Oculus Go necessitated certain choices being made for interaction techniques, e.g. for pointing and selection. Laser pointers (ray-casting) have been demonstrated as an efficient way of interacting with onscreen keyboards [31] so a similar approach was applied to selecting and grabbing interactable objects within LogiBot. A mechanism to move objects towards and away from the controller was implemented using the trackpad on the Oculus Go. A user alters the object distance by swiping up or down on the touchpad, although the requirement to adjust object distances was somewhat minimized by effective level design and careful consideration of object placement in the scene at run time.

Object manipulation was implemented in the form of pointing the controller at object (ray-cast), pressing the trigger to pick up the object, or clicking the touchpad down to "use" the object, which is a context-sensitive action (e.g. using the "trash" button next to the instruction block grid empties it of all blocks currently placed in it). Objects that have been picked up are suspended in the air and move relative to the controller when the player moves it.

Instruction grids are 4 by 3 grids of slots (Fig. 2) that users can place instruction blocks on to build their solutions to the robot maze. The main grid (that appears in every level) is used to create the program that the robot will execute, but another appears on the last level of the game - the procedure grid. Blocks that are dragged close to a grid will automatically snap to the first free slot on that board when the player releases their grip on the block. The grids build a list of instructions from the blocks by reading the blocks from the top-left to the bottom-right, which is then passed to either the robot directly, or the procedure block. Instruction blocks can be added to the main grid by either attempting to grab one of the "template" blocks, of which there is one for each instruction type. This creates a copy of that block type which the players grab instead, or by "using" the template block, which automatically inserts a copy of the block type in the first free slot on the board. Blocks can be removed from the grid they sit in by grabbing them and moving them. Blocks can be placed anywhere in the play area.

## IV. EXPERIMENT

### A. Apparatus

The control group were provided with a laptop with the required software installed and LightBot loaded, while the experimental group were provided with a VR headset (Oculus

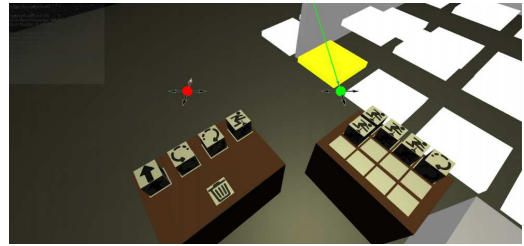


Fig. 2. Screenshot of instruction blocks table

Go) to complete the tasks, and a laptop to complete the User Engagement Survey (UES) [32]. The entire experimental task was designed to take 20-30 minutes, while the post-task questionnaire and interview took 30 minutes to complete.

### B. Participants

The study was run with 10 participants. We recruited from current students and alumni of Lancaster University. The sample was split into two groups for a between-subjects study, one group using LogiBot and the other using LightBot. Six participants completed the LogiBot tasks and four completed the LightBot tasks. The demographics of participants completing the study were 3F, 7M, age 18-55.

### C. Task

Participants in the LightBot condition were instructed to complete section 1 - basics and section 2 - procedures. The LogiBot participants were instructed to complete the five levels. 20 - 30 minutes was allowed for this section of the study. Observation notes were collected by the researchers for behaviours that were deemed interesting. On completing the task, both groups of participants were asked to complete the UES. Finally, we conducted short semi-structured interviews with participants to probe them on their motivations and insights when carrying out the tasks. Common questions revolve around the perceived learnability of the system, as well as comfort of the VR headset.

### D. Metrics

Engagement was measured using O'Brien et al.'s [14] User Engagement Survey, which consists of thirty 5-point Likert style questions measuring four latent variables – focused attention, perceived usability, aesthetic appeal and reward factor and has been used previously to measure engagement in VR games [33]. Each of these sub scores are averaged for each category and the totals summed together to derive an overall score. Further measurements were in the form of event sampled observations based on user behaviour during the experiment, accompanied by a semi-structured interview to explore thoughts and motivations of the users during the study.

## V. RESULTS

### A. UES

The survey results were analysed through independent sample t-tests. In comparing the LightBot and LogiBot survey responses, we found a statistically significant difference in focused attention for the applications, LightBot ( $\mu_{Li}=4.4$ ) was significantly higher than LogiBot ( $\mu_{Lo}=3.9$ ),  $t(8) = 1.91$ ;  $p < .05$ . We found a statistically significant difference in perceived usability for the applications, LightBot ( $\mu_{Li}=4.1$ ) was

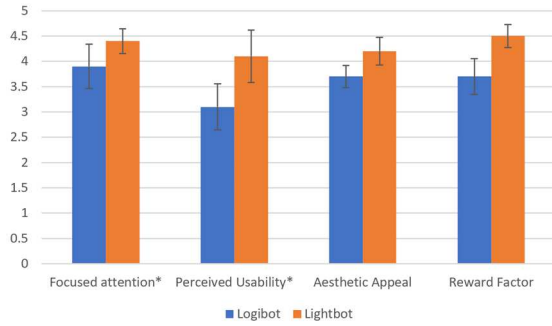


Fig. 3. Comparison between LightBot and LogiBot, mean responses grouped as latent variables. Reverse scale correction applied. \* indicates significant difference.

significantly higher than LogiBot ( $\mu_{Lo}=3.1$ ),  $t(8) = 2.28$ ;  $p < .05$ . There was no significant difference in the aesthetic appeal for LightBot ( $\mu_{Li}=4.2$ ) as compared to LogiBot ( $\mu_{T}=3.7$ ),  $t(8) = 1.01$ ,  $p > 0.05$ . There was no significant difference in the reward factor for LightBot ( $\mu_{Li}=4.45$ ) as compared to LogiBot ( $\mu_{T}=3.7$ ),  $t(8) = 1.8$ ,  $p > 0.05$ . These results are visualized in Fig. 3.

### B. Interviews

Interview responses indicated that participants for both LightBot and LogiBot were more likely to explore the interface independently as opposed to following on screen instructions. Both sets of participants recognized the value of such learning games in respect to learning programming and commented “bit like one of the programmable robot toys”. LightBot participants said it was good for practicing “solving structural problems”. For LogiBot a lot of the responses centred around the interaction methods and the clarity of the instructions provided, for example “vague instructions” and “instructions lacked context”. The VR headband impacted upon the comfort level of participants. They commented on how it can be loose at times during gameplay, which draws attention away from the game. The headband had a tendency to get hot over time, which participants noted may make them less likely to play the game for a long period of time.

For LogiBot, a common issue for participants was the way instructions were presented at the beginning of the game using text. They found this to be vague and confusing, and would rather have seen a visual example of how to solve a level before starting. For instance, one participant pointed out that they initially tried to place instruction blocks on the robots table, and tried to place the robot on the instruction table rather than the yellow block. They enjoyed the dragging interaction as it was easier to drag, due to the fact that they could not see the controller while playing the game. Some participants commented on the scaling of the levels, saying that the difficulty scales too quickly. One participant did not find LogiBot useful in teaching programming concepts. They argued that people will already have these skills before playing the game (structural problem solving). They also did not think that VR was a necessary tool for these kinds of problem solving exercises. Many participants noted frustration with trying to grab items such as blocks and the robot, sometimes picking up the wrong block, and placing the robot on the platform.

### C. Observations

Observations indicated that while the experience was a seated experience, participants still displayed significant ranges of movement. Correlation to interview responses indicate one of the reasons for this was that the table with the instruction blocks on it was too far to the left participants field of view, necessitating them to have to twist uncomfortably to see the instruction block table. In a couple of observations, participants did not locate the instructions block table for a considerable amount of time, leading to frustrations being evident.

## VI. DISCUSSION

The results show that the Perceived Usability and Focused Attention for LightBot scored significantly higher than LogiBot, indicating that the VR interface of LogiBot does not significantly contribute to use engagement. We can attribute LightBot scoring higher to interface familiarity for desktop applications. This adds interesting design questions for the VR interface and what implementations will allow the users to have a better engagement with the task and therefore the learning process. The neutral results for Aesthetic Appeal for LogiBot indicate that there is less evidence of the ‘wow-factor’ from using VR applications than is generally assumed. Perhaps with the proliferation of low-cost VR devices such as the Oculus Go or Quest, VR is treated as more common-place or expected. The users thus possibly expect better functionality and interaction mechanisms beyond typical ‘roller-coaster demos’ bundled with past VR headsets.

LogiBot implemented basic reward factors, limited to congratulation messages and sounds on level completion. Neutral results indicate reward mechanics did not dominate user experience over learning for both applications. There are opportunities to enhance engagement through gamification mechanics as a part of the learning process.

The results indicate that in its current form LogiBot is not more engaging than LightBot and by extension VR is not automatically more engaging than desktop applications, thus we reject **H1** while also making **H2** invalid. In analysing the interview responses, ease of use and interaction methods were identified as the main barriers in preventing the LogiBot application from being more engaging. If these issues are addressed, we believe that VR has the potential to be a key element of learning applications by increasing engagement.

Overall, the observations show that it may be necessary to think more deeply about interactions for VR applications versus traditional desktop applications, including identifying an acceptable range of motion for different experiences, i.e. seated or standing. The difficulty in implementing effective interactions can somewhat be ascribed to the limitations of the Oculus Go and its 3DoF tracking. This seems to support the notion that low-cost VR devices are more suited to content consumption rather than reflective interaction with the content.

### A. Limitations and Future Work

As a preliminary exploratory study, the number of study participants was low. We aim to conduct a thorough play test of LogiBot with additional participants to identify further interaction issues which we will look to solve. With the launch of the Oculus Quest (and Quest 2), we would like to port

LogiBot to this device, which will increase the options for interactions and it would be interesting to see if this has any impact on the level of engagement experienced by users.

## VII. CONCLUSION

We provide the preliminary results of a study to investigate engagement for VR-based block-programming games. Our results indicate that familiarity with VR by users no longer guarantees that VR is more engaging than similar traditional desktop applications. This places an ever greater emphasis on designing effective interactions within VR if VR-based learning tools like LogiBot are to show benefits of use over existing traditional desktop environments.

## REFERENCES

- [1] R. C. Clark and R. E. Mayer, *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. John Wiley & Sons, 2016.
- [2] M. Bricken and C. M. Byrne, *Summer Students in Virtual Reality*. Academic Press, Inc., 1993.
- [3] J. Psołka, "Immersive training systems: Virtual reality and education and training," *Instr. Sci.*, vol. 23, no. 5–6, pp. 405–431, Nov. 1995, doi: 10.1007/BF00896880.
- [4] M. S. Horn, E. T. Solovey, R. J. Crouser, and R. J. K. Jacob, "Comparing the use of tangible and graphical programming languages for informal science education," in *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, 2009, p. 975, doi: 10.1145/1518701.1518851.
- [5] M. Kalogiannakis and S. Papadakis, "A proposal for teaching Scratch/Jr programming environment in preservice kindergarten teachers," in *Proceedings of the 12th Conference of the European Science Education Research Association (ESERA)*, 2017, pp. 21–25.
- [6] M. Resnick et al., "Scratch," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, Nov. 2009, doi: 10.1145/1592761.1592779.
- [7] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness," in *Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11*, 2011, p. 9, doi: 10.1145/2181037.2181040.
- [8] Lightbot.com, "LightBot," 2017. .
- [9] A. Y. Wang, R. Mitts, P. J. Guo, and P. K. Chilana, "Mismatch of Expectations," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 2018, pp. 1–13, doi: 10.1145/3173574.3174085.
- [10] J. Piaget and E. Duckworth, "Genetic Epistemology," *Am. Behav. Sci.*, vol. 13, no. 3, pp. 459–480, Jan. 1970, doi: 10.1177/000276427001300320.
- [11] A. Amresh, A. R. Carberry, and J. Femiani, "Evaluating the effectiveness of flipped classrooms for teaching CS1," in *2013 IEEE Frontiers in Education Conference (FIE)*, Oct. 2013, pp. 733–735, doi: 10.1109/FIE.2013.6684923.
- [12] T. F. binti A. Rahman, N. Anuar, and R. F. M. Said, "How the nature of programming and learning materials affects novice learner's motivation and programming ability," in *Proceedings of the 6th International Conference on Information and Education Technology - ICIET '18*, 2018, pp. 124–128, doi: 10.1145/3178158.3178184.
- [13] H. L. O'Brien and E. G. Toms, "What is user engagement? A conceptual framework for defining user engagement with technology," *J. Am. Soc. Inf. Sci. Technol.*, vol. 59, no. 6, pp. 938–955, Apr. 2008, doi: 10.1002/asi.20801.
- [14] H. L. O'Brien and E. G. Toms, "The development and evaluation of a survey to measure user engagement," *J. Am. Soc. Inf. Sci. Technol.*, vol. 61, no. 1, pp. 50–69, Jan. 2010, doi: 10.1002/asi.21229.
- [15] Gov.uk, "National curriculum in England: computing programmes of study," National curriculum in England: computing programmes of study, 2013. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>.
- [16] D. Weintrop et al., "Evaluating CoBlox: A comparative study of robotics programming environments for adult novices," *Conf. Hum. Factors Comput. Syst. - Proc.*, vol. 2018-April, p. 2018, 2018, doi: 10.1145/3173574.3173940.
- [17] F. Kalelioğlu and Y. Gülbahar, "The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective," *Informatics Educ.*, vol. 13, no. 1, pp. 33–50, 2014.
- [18] A. Ioannidou, A. Repenning, and D. C. Webb, "AgentCubes: Incremental 3D end-user development," *J. Vis. Lang. Comput.*, vol. 20, no. 4, pp. 236–251, Aug. 2009, doi: 10.1016/j.jvlc.2009.04.001.
- [19] D. Bau, J. Gray, C. Kelleher, J. Sheldon, and F. Turbak, "Learnable programming," *Commun. ACM*, vol. 60, no. 6, pp. 72–80, May 2017, doi: 10.1145/3015455.
- [20] S. Cooper, W. Dann, and R. Pausch, "Alice: a 3-D tool for introductory programming concepts," *J. Comput. Sci. Coll.*, vol. 15, no. 5, pp. 107–116, 2000.
- [21] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The scratch programming language and environment," *ACM Trans. Comput. Educ.*, vol. 10, no. 4, pp. 1–15, 2010, doi: 10.1145/1868358.1868363.
- [22] E. F. Melcer and K. Isbister, "Bots & (Main)Frames: Exploring the Impact of Tangible Blocks and Collaborative Play in an Educational Programming," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 2018, pp. 1–14, doi: 10.1145/3173574.3173840.
- [23] J. T. Bell and H. S. Fogler, "The investigation and application of virtual reality as an educational tool," in *Proceedings of the American Society for Engineering Education Annual Conference*, 1995, pp. 1718–1728.
- [24] R. M. Felder, L. K. Silverman, and others, "Learning and teaching styles in engineering education," *Eng. Educ.*, vol. 78, no. 7, pp. 674–681, 1988.
- [25] M. Chandramouli, M. Zahraee, and C. Winer, "A fun-learning approach to programming: An adaptive Virtual Reality (VR) platform to teach programming to engineering students," in *IEEE International Conference on Electro/Information Technology*, Jun. 2014, pp. 581–586, doi: 10.1109/EIT.2014.6871829.
- [26] F. R. Ortega et al., "Towards a 3D Virtual Programming Language to increase the number of women in computer science education," in *2017 IEEE Virtual Reality Workshop on K-12 Embodied Learning through Virtual & Augmented Reality (KELVAR)*, Mar. 2017, pp. 1–6, doi: 10.1109/KELVAR.2017.7961558.
- [27] F. Grivokostopoulou, I. Perikos, and I. Hatzilygeroudis, "An Innovative Educational Environment Based on Virtual Reality and Gamification for Learning Search Algorithms," in *2016 IEEE Eighth International Conference on Technology for Education (T4E)*, Dec. 2016, pp. 110–115, doi: 10.1109/T4E.2016.029.
- [28] E. B. Witherspoon, R. M. Higashi, C. D. Schunn, E. C. Baehr, and R. Shoop, "Developing Computational Thinking through a Virtual Robotics Programming Curriculum," *ACM Trans. Comput. Educ.*, vol. 18, no. 1, pp. 1–20, Dec. 2017, doi: 10.1145/3104982.
- [29] Robotc, "Robotc: A C programming language for robotic," 2019. <https://www.robotc.net/> (accessed Aug. 03, 2020).
- [30] Vrtoolkit.readme.io, "VRTK - Virtual Reality Toolkit." 2021, [Online]. Available: <https://vrtoolkit.readme.io/>.
- [31] M. Speicher, A. M. Feit, P. Ziegler, and A. Krüger, "Selection-based Text Entry in Virtual Reality," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 2018, pp. 1–13, doi: 10.1145/3173574.3174221.
- [32] H. L. O'Brien, P. Cairns, and M. Hall, "A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form," *Int. J. Hum. Comput. Stud.*, vol. 112, pp. 28–39, Apr. 2018, doi: 10.1016/j.ijhcs.2018.01.004.
- [33] I. C. Landa-Avila and M.-L. Cruz, "Engagement in a Virtual Reality Game with Gesture Hand Interface. An Empirical Evaluation of User Engagement Scale (UES)," in *DUXU 2017: Design, User Experience, and Usability: Designing Pleasurable Experiences*, 2017, pp. 414–427, doi: 10.1007/978-3-319-58637-3\_33.