Removing Reliance on Metaphors to Enable Cross Cultural Learning of Complex Programming

Keanna Wright
School of Electrical and
Computer Engineering
North Carolina A & T State University
Greensboro, North Carolina 27411
Email: kmwright1@aggies.ncat.edu

***Abstract*—A thorough understanding of Computer Science fundamentals is a necessity when solving problems requiring complex programming. However, introductory programming courses are taught inconsistently and inefficiently causing underrepresented students to be discouraged. Lectures are not giving students a deep understanding of runtime dynamics and primarily rely upon metaphors to explain abstract concepts. These metaphors are not received the same across cultures leaving minorities at a disadvantage. In this paper, we examine a more visual approach to teaching complexity. Additionally, we create an immersive virtual reality environment that models the dynamic features of code and how it effects computer memory. This research aims at observing how this virtual environment effects the understanding of programming abstractions. In the future more virtual reality curriculum modules will be created and data will be collected on the impact of these experiential learning environments.**

I. Introduction

When Computer Science students are learning about introductory programming a solid foundation is necessary. The current approach to teaching abstract programming concepts is lecturing while incorporating metaphors. These metaphors are not always received the same across cultures. This static approach to teaching coding does not show the dynamic features that take place when it is being run [3]. In this paper a new approach to teaching introductory programming will be introduced to combat the problem of students only understanding what is on the surface of code. This research will have an impact on all Computer Science students but specifically minority students who may struggle to adjust to some introductory programming courses. Professors, students, and corporations are interested in this research because a more in depth understanding of programming is important when creating new technologies. Without this research many students will continue to fall short of a deep understanding when it comes to the basics of programming. Lectures are not sufficient when having to introduce the dynamic concept of code. Using a more visual approach to teaching programming is necessary. The SIMR virtual reality environment presents the dynamic features of code and how it effects computer memory. This solution is a more visual approach to teaching introductory programming creating a mental model of how code runs. This method of learning benefits the students who are not accustom to lectures and unfamiliar metaphors. This research is different from current research due to the fully immersive nature of virtual reality being used to teach programming concepts.

II. Related Works

Using a more visual hands on approach to teaching Computer Science is not a new occurrence. Sharad Sharma used the concept of "Gamification" to help teach students about loops and arrays. The study conducted, showed that students believed that this method of instruction impacted their learning in a positive way [4]. Students additionally proved to be more motivated to learn about programming when interacting with visual code. An interactive environment fosters problem solving skills and also increased the competitive spirit of students [1]. Studies have been conducted associated with developing a new strategy of teaching programming apart from lectures. Instead of lecturing, finding a way to help students create satisfactory mental models [3]. Professors have different approaches to teaching students the basics of programming. The proper teaching of operating systems is constantly being debated [2]. These solutions do not ultimately solve the issue of students not having a deep understanding of introductory programming due to an auditory approach to teaching.

III. Approach

Current approaches to this problem of teaching Computer Science students abstract concepts do not provide a hands on or visual learning experience. Using a Virtual Reality environment that models the dynamic features of code will visualize how code is compiling and the inconsistencies that can take place. This new approach was inspired by women and minorities who are consistently left at a disadvantage when taking introductory programming courses due to inefficient teaching techniques. Virtual reality instructional modules designed to motivate students to learn about loops and arrays successfully demonstrated the effectiveness of visualizing programming concepts [4]. This new approach will provide an immersive learning experience leaving students with a greater understanding of introductory programming. This Virtual Environment was created using Unity which is a cross-platform game engine. The game development process uses C sharp to develop scripts. The SIMR virtual environment steps

through the compilation process of code visually showing any discrepancies that can occur.

IV. Methods

This research is concerned with both creating the SIMR virtual environment and collecting data on how virtual experiential learning impacts the learning of introductory programming. When creating the SIMR virtual environment the first step was to figure out which game development engine was to best to use. In a similar study done by an undergraduate researcher named Nyala Jackson, React360 was used to create the environment. However, in this research Unity was used to create the virtual environment. The first task at hand when creating the environment was to mimic a space that gives a futuristic computer feel. A plane with a neon green grid material was added to the scene. Next, digital circuit and digit line particle affects were added as well as neon panels for more visuals.

The translucent panel was then added to the scene. This panel will be responsible for displaying the code that will be broken down into snippets and visualized. This specific code that is presented has a memory-bug that's not very obvious. This bug results in "Hello World" being printed twice. However, this discrepancy will become apparent throughout the simulation.

Fig. 1. Translucent panel for code

The variable location table and the 10x10 memory table were then added. The grid will show the dynamics of the code, presenting how the computer runs each line of code and ultimately prints an output step by step.

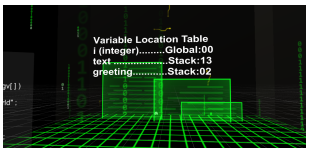

Fig. 2. Variable location table

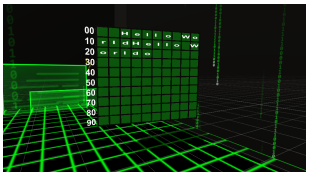

Fig. 3. Memory grid

The SIMR virtual reality environment allows the user to interact with the memory panel grid. This interaction involves a magnifying glass that when waved over the grid will display the ASCII code of each character.

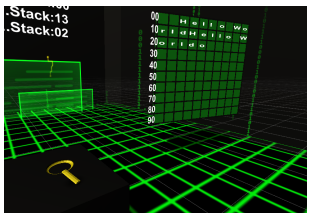

Fig. 4. Magnifying glass

Eight buttons were added to the environment that will allow the student to step through the code and simultaneously see the memory allocation on the memory grid as the code runs. This specific code has a logic error that results in "Hello World" being printed twice.

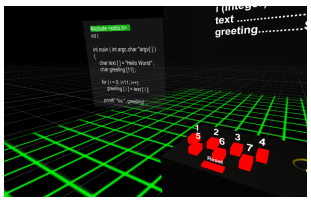

Fig. 5. Red buttons for stepping through code

The second part of this research is to collect data on how virtual experiential learning impacts the learning of introductory programming. Data will be both qualitative and quantitative. Interviews will take place where students that have experienced the use of the SIMR virtual learning environment state how it has influenced their understanding of memory reliant programming concepts. The results from the interviews will be analyzed and recorded. Questionnaires implementing a number based system to answer how effective this virtual module is will also be used to collect data. Students will state if their learning experience was excellent, above average, average, below average, or very poor.

V. Conclusion

We created a virtual reality teaching module for Computer Science students learning introductory abstract programming concept. Currently, students must learn the dynamic features of code based off lectures and verbal metaphors. This specific module focuses on the physical memory of code and any errors that can be produced. This visual teaching aid allows for students to make a mental model of programming abstractions. Due to the Covid-19 pandemic our experiment could not be executed. However, the results of this research will be both quantitative and qualitative. Student interview responses and surveys will be analyzed. In the future our team hopes to continue this research to analyze the impacts of virtual reality learning on student's understanding of computer science fundamentals. Furthermore, This research has potential to impact the Computer Science learning community in a positive way

and provides a blueprint for future immersive visual education possibilities.

ACKNOWLEDGMENT

This research opportunity would not have been made possible without the Computing Research Association, Distributed Research Experiences for Undergraduates Program. The author would like to acknowledge the contributions of Dr. Monica Anderson and undergraduate researcher, Nyala Jackson during the research process.

REFERENCES

[1] Khaled Asad, Moanis Tibi, and Jamal Raiyn. "Primary School Pupils' Attitudes toward Learning Programming through Visual Interactive Environments." In: *World journal of education* 6.5 (2016), pp. 20–26.

[2] Bernard Doyle and Raymond Lister. "Why teach unix?" In: *Conferences in Research and Practice in Information Technology Series*. 2007.

[3] Jeisson Hidalgo-Céspedes, Gabriela Marın-Raventós, and Vladimir Lara-Villagrán. "Understanding notional machines through traditional teaching with conceptual contraposition and program memory tracing". In: *CLEI Electronic Journal* 19.2 (2016), pp. 3–3.

[4] J. Stigall and S. Sharma. "Virtual reality instructional modules for introductory programming courses". In: *2017 IEEE Integrated STEM Education Conference (ISEC)*. 2017, pp. 34–42.