Purdue University
Purdue e-Pubs

ASEE IL-IN Section Conference

Recruiting VR Troopers: Bringing Introductory Programming Projects to Life in Virtual Reality

Stewart J. Thomas
Valparaiso University

Ruth E.H. Wertz
Valparaiso University

Jeffrey Will
Valparaiso University

Follow this and additional works at: https://docs.lib.purdue.edu/aseeil-insectionconference

Thomas, Stewart J.; Wertz, Ruth E.H.; and Will, Jeffrey, "Recruiting VR Troopers: Bringing Introductory Programming Projects to Life in Virtual Reality" (2018). *ASEE IL-IN Section Conference*. 2.
https://docs.lib.purdue.edu/aseeil-insectionconference/2018/innov/2

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

Recruiting VR Troopers: Bringing Introductory Programming Projects to Life in Virtual Reality

Stewart J. Thomas[1], Ruth E.H. Wertz[1,2], Jeffrey Will[1]
Dept. of Electrical and Computer Engineering[1], Dept. of General Engineering[2]
Valparaiso University
Valparaiso, IN USA 46383

Abstract

Classes in introductory programming often focus on solving small, succinct problems that can typically be completed in few lines of source code. While useful for learning the basics of algorithm implementation and language syntax, this method suggests to learners that all programming problems exist in isolation and are self-contained. In contrast, most programming assignments faced by fresh graduates are large in scope and require use of many pre-built libraries and extensions. As a result, students are not entirely prepared to write code that will function within a larger system.

To address this problem, an introductory C programming course at Valparaiso University has explored the use of virtual reality as a means to motivate students to have fun while practicing coding skills and showcase the power of working within constraints of a complex system. Students are provided a brief introduction to the OpenGL 3D graphics framework and then asked to design a small, optionally animated, scene using their current knowledge of the C programming language. Later in the semester, these same students are brought into a VisCube Virtual Reality system to experience their scenes in a fully immersive environment. The VisCube uses eight rendering paths and stereo displays to generate a 3D scene in a 10’x8’x6’ cube. This exercise serves to show students that even a simple scene can then easily expanded to display in a virtual reality environment. We discuss the project assignment and student impacts using assessment and provide a brief discussion of how this can be adapted to facilities with other visualization capabilities.

I. Introduction

Assignments in introductory programming courses are often small in scope and can typically be completed in a single file with a few lines of code. This structure is seen to facilitate learning the basics of not only a programming language, but program design itself. Using shorter assignments also easily lends itself quicker or easier grading for the instructor.

The instructional objectives of introductory programming courses tend to focus on learning fundamental concepts, problem-solving, algorithmic thinking, and the syntax of writing basic code [1]. The larger aspects of software design are not seen until later courses. A natural consequence of such objectives, is assignments that are able to test fundamental concepts. For instance, after discussing branching operations in lecture, student may be asked to write a program using an '`if`' statement to test the equality of two numbers. However, such a problem can be completed in just a few lines of code. Nonetheless, students are able to easily grasp the fundamental concepts of a language, and design a solution to the problem.

As more students enroll in programming courses and MOOC classes continue to flourish, writing assignments that can be automatically graded is a common goal for instructors [2]. This provides a variety of benefit to students, but also leads to smaller, succinct programs that can be easily assessed by a computer. The burden to write an easily-assessible task for a problem, becomes increasingly greater as program complexity increases.

For students pursuing a computer science degree, solving smaller-scale problems in introductory programming courses (i.e., CS1 and CS2) is not problematic. Computer-science students continue on to take other courses that expose the larger issues in software engineering and complex programming solutions. However, this is not always the case for other engineering disciplines. Many capable electrical engineering students may complete one or two introductory programming courses (such as MATLAB and C, or C and assembly) during their degree program and upon graduation accept a job with a heavy focus on software development. In jobs with a software focus, the scope of projects is much larger with many interconnected components. In such situations, the developer must quickly acclimate to a variety of libraries or proprietary code used by the project. Additionally, many companies have strict style and coding standards for writing code. A fresh graduate must quickly adapt to this style. For a student in such a position, their introductory courses have prepared them to write code, but has not necessarily prepared them to work in a large-scope, complex project.

In order to expose students to large-scale projects, this paper investigates using assignments involving a large-scale, complex 3D rendering system in an introductory programming class. We provide a background of the introductory programing class at Valparaiso University and the 3D rendering system. We describe the intervention and discuss the results. We end with a conclusion.

II. Background

At Valparaiso University, ECE251 is a mandatory introductory programming class for all electrical engineering and computer engineering majors. This course is typically taken during the third semester. Previously, the course was taught using the Java programming language, but has used the C language for the last five years. The course has two weekly 75-minute meetings throughout a 15 week semester. In a typical meeting, students a) participate in a short (typically 20 minute) lecture/discussion introducing a new topic, b) receive their assignment, and c) break up into pairs to work on the assignment for the balance of meeting time. During this last portion of the class meeting, the instructor walks around the room and answers student questions as they complete the assignment. The assignment is then due at the start of the next class meeting. In a semester students will complete over 20 assignments of varying difficulty and then complete a final project spanning the final two weeks of the course.

Virtual Reality (VR) is one example of a large, complex interconnected system. VR is a medium comprised of four key elements: a virtual world, immersion, sensory feedback, and interactivity [3]. The two main VR systems today are head-mount displays and projection-based systems. Head-mount systems have recently become quite affordable with the advent of consumer-grade products such as the Oculus Rift and the newer HTC Vive. These systems are attractive due to their low cost, but for education purposes suffer from the fact that they are single-user devices. Projection-based systems have the advantage of being large enough to be able to accommodate groups of users at a time, providing a venue for small-class teaching opportunities. Their drawback is their high cost of $400K to over $2M and requiring space and personnel resources.

Asking students to create a fully three-dimensional render of a scene offers several pedagogical benefits:

1) Drawing 3D scenes provide immediate feedback --- A significant advantage of automatic code grading systems is the immediate feedback for students [2]. Drawing and rendering your own 3D scene offers the same immediate feedback. Students know immediately if the scene is rendering correctly.
2) Creating 3D scenes encourage creativity --- Small, simplistic programming assignments can be restrictive with little room for student creativity. Designing one's own 3D scene, offers ample room for creativity and originality.
3) Rendering images goes beyond the console --- In teaching the basics of a language, most assignments focus on text-only streams viewed from a console (terminal) window. In contrast, 3D scenes and virtual reality push programming assignments well beyond a simple text-only interface, to interactive scenes.
4) Libraries such as OpenGL are complex --- Asking students to engage with a highly complex system such as OpenGL provides experience working within a system where everything cannot be completely understood. Students must become self-motivated learners and learn to work within uncertainty while focusing on the important aspects of their program.

III. Project Objectives

For this project, the OpenGL rendering engine was leveraged as a tool to engage and motivate students and provide experience using a complex system. The OpenGL assignment took place the course meeting immediately after the first exam. At this point, students have had experience with variables, branching, looping, bitwise and mathematical operations and functions --- the basic building blocks of the C language. While exposed to the basics of programming, students have not yet worked with external, compiled libraries, or used code that others have written as would be typical in most industry settings.

To expose students to the power in re-using code and following basic rules/templates, students were asked to draw and optionally animate a 3D scene using the OpenGL rendering engine. Students were provided a basic introduction and several example programs that demonstrate setup of an OpenGL window and adding a variety of colored 3D shapes to a scene. Students were also shown examples of using timing to animate various objects.

While students received immediate feedback on their own screens, our goal was to show students the power of working within a larger library and motivate students to further engage with the programming language. To this end, samples of student work were collected and converted to work in the fully-immersive Viscube VR system. Students were then taken inside the Viscube to experience their work in a VR context.

The OpenGL/VR intervention and the following research were developed to determine the affect the OpenGL assignment had on students' motivation and confidence in working with a complex problem in programming.

IV. Research Design

This project took place in two different semesters with the same professor at Valparaiso University. Valparaiso University is a small, private liberal arts university in the midwest. In the 2016 fall semester, there were 52 students enrolled in ECE251 across two sections. In the 2017 fall semester, there were 43 students enrolled across two sections. Of those 43 students, 32 (74%) were of sophomore standing, 9 (21%) were of junior standing, and 2 (4.6%) were of senior standing. The majority of students were either electrical or computer engineering majors. A small minority (3.8% in 2016 fall and 2.3% in 2017 fall) of students were majoring in either civil or mechanical engineering. In both years, all students participated in this project.

The OpenGL assignment asked students to render a 3D scene with their choice of a forest, an animal, or a house. The objective was to combine the necessary 3D primitives and polygon faces to represent the object. Extra credit was offered to students who animated their scene. Examples of student submissions are shown in Figure 1.

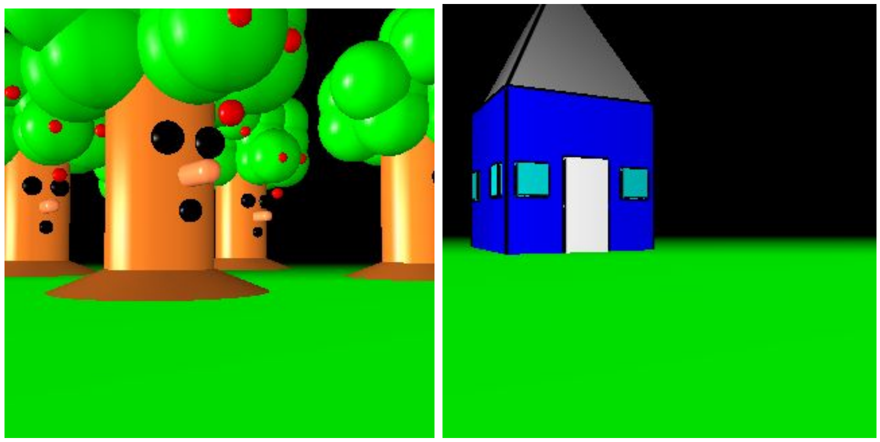

Figure 1: Student created renderings of a tree with an animated, falling apple (left) and a house (right).

The instructor selected exemplary student submissions and modified these programs to operate within the Viscube system. The Viscube uses multiple parallel rendering paths and requires animations and descriptions of 3D geometry to be programmed in a consistent manner. Of the programs selected for conversion, a majority (75%) student animations required minor modifications to be made by the instructor. The remaining functionality to setup the Viscube rendering pipeline and interactive camera tied to the 3D glasses worn by a user were performed by the instructor. After preparing the programs, groups of 4 to 8 students were taken into the Viscube and shown their peers' work. Because this is a fully interactive environment, students were able to literally walk around their virtual rendering. This led to a discussion of 3D visualization techniques and implementations for rendering and calculated light for all shapes --- whether or not they were obscured by other surfaces. For example, students "walked" inside a house only to see that windows were actually being created with large cubes offset in the walls.

To capture students' attitudes toward programming, virtual reality, and their self-confidence in being able to complete the assignment successfully, students were given a six-item Likert-scaled pre/post survey. The items listed in Tables 1 and 2 were generated by the authors for the purpose of this investigation. The selected items were chosen to reflect the expectancy/value theory framework. The selected questions are modified statements from Svinicki [4] . These statements consider a learner's motivation and items that influence the value of a targeted task.

Table I: Pre-Survey Items

Q1	I am interested in programming projects
Q2	I enjoy solving programming challenges
Q3	I enjoy showing others the programs I've written
Q4	I am excited to see class projects in Virtual Reality (VR)
Q5	I am interested in making my own program in Virtual Reality
Q6	I am confident I can make a Virtual Reality scene.

Table II: Post-Survey Items

Q1	I am interested in programming projects
Q2	I enjoy solving programming challenges
Q3	I enjoy showing others the programs I've written
Q4	I am enjoyed seeing class projects in Virtual Reality (VR)
Q5	I am interested in making my own program in Virtual Reality
Q6	I am confident I can make a Virtual Reality scene

The pre-survey was disseminated to students during week five prior to students having any exposure to a 3D rendering or virtual reality system. The post-survey was re-administered upon completion of the semester (week 15) and after the OpenGL intervention.

V. Results

Before the OpenGL assignment, in responses to the question "I am interested in programming projects", 77.5% of students agreed with this statement. Likewise, 70% of students agreed that they "enjoy solving programming challenges" while only 67% of student responses indicated they enjoyed showing others the programs they've written.

In stark contrast, 100% of student responses were excited to see class projects in virtual reality and only 2.5% indicated they were not interested in making their own program in virtual reality.

Students from the previous semester were asked a similar survey to see how the intervention affected their introductory programming experience.

Students surveyed after the Viscube demonstration revealed an increase in both motivation and confidence. Responses to surveys showed that students had a higher interest in programming projects. Likewise, confidence in their ability to create a virtual reality scene also reported higher.

Most interestingly, students surveyed after the Viscube demonstration reported a strong trend toward enjoying showing their work to others. This effect is seen in Figure 3.

Table III: Responses for survey prior to assignment

	Percentage of responses (%)				
	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Question #	(5)	(4)	(3)	(2)	(1)
Q1	50.0	27.5	12.5	7.5	2.5
Q2	40.0	30.0	17.5	10.0	2.5
Q3	37.5	30.0	25.0	5.0	2.5
Q4	77.5	22.5	0.0	0.0	0.0
Q5	52.5	27.5	17.5	0.0	2.5
Q6	20.0	30.0	35.0	12.5	2.5

Table IV: Responses for survey prior to assignment

	Percentage of responses (%)				
	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Question #	(5)	(4)	(3)	(2)	(1)
Q1	63.2	31.6	5.3	0.0	0.0
Q2	73.7	15.8	10.5	0.0	0.0
Q3	47.4	47.4	5.3	0.0	0.0
Q4	57.9	26.3	15.8	0.0	0.0
Q5	52.6	21.1	21.1	5.3	0.0
Q6	31.6	26.3	26.3	10.5	5.3
Q7	73.7	5.3	15.8	5.3	0.0
Q8	63.2	10.5	21.1	5.3	0.0

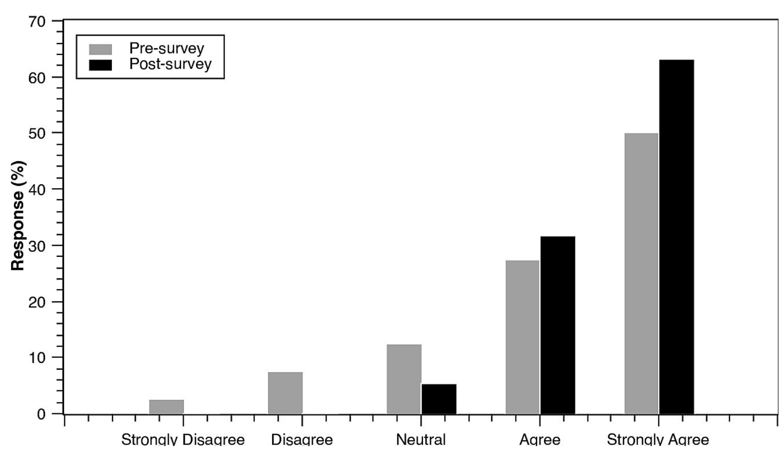

Figure 2: Responses to Q1, “I am interested in programming projects”

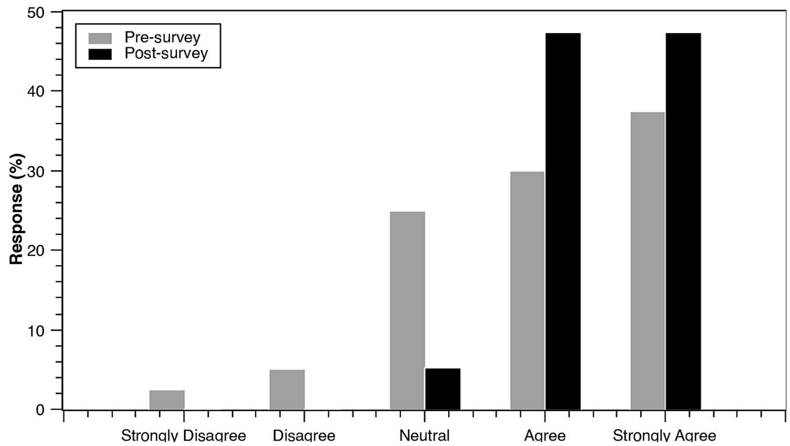

Figure 3: Responses to Q3, “I enjoy showing others the programs I’ve written”

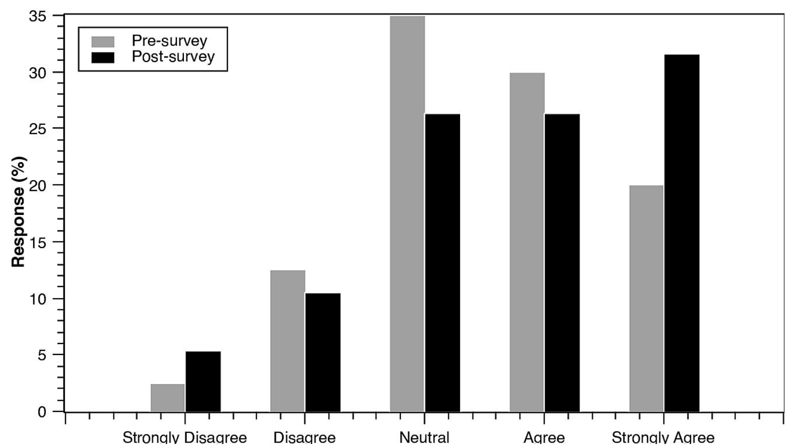

Figure 4: Responses to Q6, “I am confident I can make a Virtual Reality scene.”

VI. Discussion

The data suggests that an OpenGL virtual reality intervention improves both motivation and confidence of students in introductory programming classes. Of the students surveyed, only a small percentage (5%) did not enjoy the demonstration.

However, the data does have several limitations. The assignment and intervention was performed in two separate years. The pre-survey was issued to students in a fall 2017 class and the post survey was issued to students who took the class in fall 2016. This means that the same cohort of students were not surveyed. Further, since the initial assignment and survey took place in week five and the virtual reality demonstration takes place at the end of the semester, it is difficult to isolate effects of the intervention. This is an alternate explanation for the higher reported enjoyment of showing programming projects to others or interest in programming projects. As students have completed more tasks and delved deeper into the course, their confidence and enjoyment could have increased.

VII. Conclusion

This paper describes an intervention for improving student motivation and confidence in an introductory programming class. The project made use of the OpenGL toolset and a fully immersive, 3D virtual reality system. From the available data, the intervention can help to improve student motivation and confidence. This project also breaks the norm of typical, small programming assignments and introduces students to design within a larger, more complex system. A future study will work to isolate the effects of the OpenGL intervention and investigate the results at another institution.

References

[1] E. Murphy, T. Crick, and J. H. Davenport. "An Analysis of Introductory Programming Courses at UK Universities," *The Art Science and Engineering of Programming*, Vol. 1, Issue 2, Article 18, 2017.

[2] W. Sherman and A. Craig. *Understanding Virtual Reality*. San Francisco: Morgan Kauffman, 2003.

[3] V. Pieterse, "Automated assessment of programming assignments" in *Proc. of the 3rd computer science education research conference on computer science education research*, pp. 45-56. Open Universiteit, Heerlen, 2013.

[4] M. D. Svinicki, "A guidebook on conceptual frameworks for research in engineering education," *Rigorous Research in Engineering Education NSF DUE-0341127*, 2008.