

# **OPTIMIZED LEARNING WITH VIRTUAL REALITY**

SUMMER INTERNSHIP PROJECT REPORT

Codey Winslow

Dr. Philip Howard

Oregon Space Grant Consortium

Oregon Institute of Technology

July 6 – September 4, 2020

## Introduction

### Project Goal

The goal of the Optimized Learning with Virtual Reality project was to gather evidence on whether skills can be better acquired using virtual reality technology over conventional media.

### Experiment Design

In order to gather concrete data, it was planned that I would develop a virtual learning environment for a group of subjects to learn one or more tasks, create videos to teach those same tasks to another group of subjects, and test both groups by having them perform the tasks in reality and by recording their performance.

After my mentor and I evaluated the technical scope of the project, it was decided that the tasks would be a set of 3 procedures performed on a touchscreen user interface, each with increased difficulty.

The virtual form of this would be a virtual room with a screen containing the same user interface as the actual test but including tutorials to teach the procedures. The video would teach these procedures by explaining the goal of each and showing their completion. A laptop with a touchscreen would contain the testing version of the user interface for subjects to perform the procedures and either pass or fail. I would observe and record in writing their actions and results.

### Procedure Descriptions

*Level 1: Rote Memorization* – This procedure requires a specific order of actions to be performed on the controls. The user must memorize and repeat them.

*Level 2: Enumerated Rote Memorization* – This procedure contains a Rote Memorization procedure for each of the three indicator lights. The user must remember all three and perform them based on which light is shown. Five lights are shown at random for the duration of this procedure.

*Level 3: Skill and Judgement* – This procedure requires understanding of the interface's use in the context of a problem, and for the user to solve the problem using the interface.

## Technical Details

### UI Design

After a few iterations of mocking up the UI design, a final version was settled on with the following:

- 3 indicator lights (green, yellow, red)
- 3 3-digit meters

- 2 3-option setting controls
- 4 toggles
- 9 buttons
- 3 vertical sliders



*Fig 1. Final UI design with 24 controls.*

### UI Implementation

A base control class along with derived classes for each control were written to wrap the functionality for all UI control behavior.

Since the runtime instances of these controls are separate from the code in scripts, a class called `UIControlCenter` was written to hold references to these instances. It also holds references to all other runtime instances used for the interface, like menus and tutorial messages.

Similar to the control classes, a base class as well as derived classes were written for procedures. These procedures subscribed listener methods to the events triggered by the control classes when the UI controls are used. This allows them to inspect what the user is doing and decide if it is valid. The base class includes logic for starting, stopping, and ending the procedure as well as its success status.

A procedure controller class was written to control the flow of procedures in the program. A list of procedure class instances is used to cycle through the procedures. When a procedure is finished, success or failure messages are displayed to the user.

### VR Implementation

In order to introduce virtual reality capabilities, the SteamVR plugin for Unity developed by Valve Corporation was imported into the project, and the Player prefab was instantiated in

front of the virtual screen. This allows users to use a virtual reality headset to view and move about the world as well as controllers to represent virtual hands.

To enable touchscreen-like behavior for the virtual screen, collider boxes were added to all UI controls so that collision with the virtual hands' index fingers are captured. Logic was added to the controls so that when a hand collider enters the control's collider, then exits, a click event is fired.

Touchscreen behavior for sliders proved to be far more complicated, so instead I utilized the SteamVR plugin's Linear Drive component class to allow the user to "grab" the slider and move it.

For tutorials, modified versions of the procedure scripts they teach were written to utilize tutorial messages.

### Training Video

For the video, a brief explanation of the UI layout is used as an introduction. This precedes in-depth explanations of each procedure along with demonstrations of completing them.

I used OBS Studio to record my screen and voice as I used a mouse to demonstrate use of the testing program.

It is arguable that the procedure explanations in the video are more elaborative than the tutorials in the VR training program.

## **Experiment Progress**

### IRB Approval Pending

For the IRB approval process to acquire permission to use human subjects in these trials, a completed IRB application and an Informed Consent Form were submitted to the IRB for consideration.

As of September 4, 2020, approval is still pending and so experiment trails have not begun.

### Future Plans

Once approval is granted by the IRB, I intend to carry out trials to gather data for this study, then synthesize the data into a report.

## **Lessons Learned**

### Technical Contributions

### Where Virtual Reality Makes Sense

It has become clearer through working on this project that there are times that virtual reality makes sense, and times that it doesn't. VR can require substantially more energy to interact with an interface when compared to traditional I/O methods. Specifically, simulating touchscreen behavior in virtual reality is not only much more work to implement, but also more work for the user to perform with the precision it requires. This project centered wholly on this mechanic, which made the cost of developing in VR and the cost of how difficult the program is to use much higher than the benefits of VR.

From the information gathered through developing this project, it can be said that activities with a primary focus on crude physical movement or immersion make the most sense for a VR application. Activities that focus more on abstract ideas or strategy, for example, make the cost of VR too high for what it provides to the user.

Crude physical movement is a good fit for VR because the movement is monitored and replicated in the virtual space, which allows the user to quickly make the connection between their actions and the outcomes. For this aspect of VR, users don't have to learn abstract controls to interact with the application's environment. Instead, they use the bodily interface they've learned all their lives to interact with the real world, and the application does the work to interpret them. If user's actions in an application map directly to real-world actions, and if those actions don't require precise movements, it is likely beneficial to make the application in VR.

Immersion alone has a much lower cost of developing in VR, but the VR aspect may make a drastic difference in the experience a user has with the application if immersion is a primary focus. Allowing a user to look around a virtual space with six degrees of freedom is quite a different experience than with traditional 2D screen-based applications, and implementing this ability into an application that already utilizes a 3D space can be quite easy when compared to the more complex systems used for interaction.

### Real World Mimicry in Virtual Reality

The 3D, physics-based nature of a typical VR environment makes developing it quite an involved task. If one was to model a VR application after a real-world physical task like tossing a bean bag, a tremendous amount of time would be spent mimicking real-world behavior while keeping it performant enough to satisfy VR comfort requirements. These goals seem almost contradictory, but a great deal of knowledge in computer science and physics would allow these goals to be met.

### Using Engines for Virtual Reality

I believe using an engine as the backbone for virtual reality simulation is necessary for small projects such as this one, but perhaps Unity was not the best choice for this kind of project. For the results of studies like this to be as accurate as possible, it is best that the engine used to simulate is one that is highly accurate. The engine I used is optimized for gaming, which does

not typically require very accurate physics simulation. It only has good enough physical behavior to be believable to a player. A sufficient engine may not exist yet since runtime optimization is still crucial for the usability of virtual reality, but in the meantime, Unity engine is a good choice being fairly capable of sophisticated physics simulation while being optimized enough for virtual reality.

## Personal Lessons Learned

### Planning

My initial plan for project entailed physical tasks like throwing a bean bag or bowling to highlight the strengths of virtual reality. This was ideal, but discussing this with my mentor Dr. Philip Howard revealed that nine weeks was too little time for me to achieve this alone, at least realistically.

Discussing alternatives with Dr. Howard showed me how much having just one other person to help brainstorm can make a difference. He suggested things I hadn't considered but which made a large impact on the future of the project, such as the importance of a 1:1 correspondence between virtual reality interfacing and real-world interaction.

Planning for the project took at least two weeks out of the internship, which was much longer than I had expected. It is likely that my project scope would have been remarkably different from the beginning had I anticipated the lengthy planning process.

### Working Alone

Working from home presented its own set of challenges apart from the typical software internship experience. One of the biggest challenges of this experience was staying motivated. Being a self-directed project, all milestones, expectations, and next steps were up to me. This puts extra emphasis on one's degree of self-drive.

There were many times I finished a task and was unsure of what to do next. Instead of going to a manager or team lead to retrieve my next task, I had to evaluate my goals, my current progress, and invent my own next step.

Experiencing this over the summer has allowed me to improve my work ethic and self-starting skills in a way that I think a traditional internship would not.

After weeks of working on a program to challenge the user, I decided I could use another set of eyes to see if my development direction needed adjusting. This proved to be one of the more stressful parts of this experience since it forced me to confront my severe misjudgment of the users.

I had some family and friends try to use a primitive version of my project to see how they fare in the procedure trials. Compared to the performance I expected during the development of

the procedures, their performance was abysmal. I had to completely reevaluate my plans for the procedures and change where I spent my energy.

Looking back, I wish I had involved others in the development much sooner than I did, as I would not have wasted as much time developing something so far off base from what I wanted to achieve.

### New Technology

Virtual reality technology is relatively new, and prior to this internship, I had no experience developing for it. Before pitching an idea involving VR development, I did do a bit of research, and found that the Unity game engine was a popular development environment for VR applications, and there exist plugins for the engine to make basic VR application development a breeze.

I did not own any virtual reality hardware prior to this project, so some funding from the internship was used to buy an HTC Vive Cosmos Elite kit.

Due to the nature of the final project, less emphasis was put on virtual reality development than other areas. Almost all my development time was spent creating and improving a traditional 2D user interface and its functionality. When porting it to the virtual learning environment, I found that trying to mimic the behavior of a touchscreen UI accurately in VR would be incredibly tedious, and likely requires rewriting much of the existing code behind modern touchscreens. Additionally, trying to accurately select items on a virtual screen with a virtual hand is tedious and difficult to get right. I believe this speaks to how differently virtual reality applications approach user interfaces. Nevertheless, I managed to make the UI useable in VR, and I believe I modelled it close enough that subjects who learn the procedures in VR would have no problem utilizing their newfound skills on the real application.

Much of the VR logic in the virtual learning environment I created was made possible by Valve Corporation's SteamVR plugin for Unity. Scripts to handle many common VR behaviors are included in the plugin, and it streamlined my VR development. However, because the technology is fairly new, and it is a third-party library, I encountered many issues with stability and misalignment with Unity features. I often had to resolve errors that occur because outdated library calls were used or because some initialization steps did not complete properly upon launching the development software. Such is the experience of working with new technology.

### My Takeaways

I learned a myriad of valuable things from this experience. Being put in an unexpected and unusual position such as this with so much freedom but also accountability allowed me to really stretch and fill many positions I hadn't planned for. For readability, here is the list form of some lessons I learned from this opportunity:

- **Project flexibility** – When planning a project, keep flexibility in mind because there are plenty of unknowns that could destroy too rigid of ideas.
- **Deadlines are crucial** – This may be obvious, but if you get too comfortable with freedom, you can ignore setting deadlines for yourself. Deadlines encourage a steady pace of progress and allow one to properly assess their priorities.
- **Test early** – Although basic, this lesson cannot be stressed enough. Start with small prototypes and then start testing. Rapid prototyping is effective because it minimizes waste. Even if you think you know the right development path, getting the product into the hands of your customers, or something that represents them, will reveal the truth.
- **Working with others is invaluable** – Having coworkers to help with a project provides more than just extra hands. The set of ideas and encouragement available from others makes for a team that is more than just the sum of its parts.
- **Everyone’s job is equal** – By this, I mean everyone has an equal share of importance in what they contribute and they each have their own set of challenges. Don’t take others’ work for granted, and don’t over-emphasize your own.
- **Have a plan for motivation** – This is more specific to remote work, but working alone and in the comfort of your own home can do wonders to crush your motivation. Having some way to stay focused and keep your energy to work high is critical to making your goals a reality.

One last thing I learned from this experience is my own strengths and weaknesses. Because I had to fill so many roles to see this project through from beginning to end, I could see where my skills and interests lie. I am most effective at brainstorming and planning when I work with others, and my skills are more aligned with focused programming for hard goals over the more managerial responsibilities. These discoveries will shape how I improve myself for the future, and I owe them all to the opportunity provided to me by the Oregon Space Grant Consortium.

### **Direction for the Future**

The vision I initially had for this project was severely scaled down due to the short time I had, my inexperience, and my lack of help in development. In the future, I believe allotting more time and engineers to develop a highly sophisticated virtual learning environment would be necessary to sufficiently answer the question, “does virtual reality allow people to learn skills more quickly?” This would allow more emphasis to be placed on the physical skills that I am convinced virtual reality teaches best.

Additionally, using a different engine that is more suited for physics simulation rather than gaming would be a better fit for the project. Research into various options for this choice is advised.

Making a quantifiable way of describing task difficulty as well as attributing types of learning used for each procedure would be beneficial to the project for testing and analysis.



Finally, a solid integration of preliminary software testing during development would make for a more reliable and error-free final product to be used in subject testing.