# Intro (10 s)

Hello everyone, thank you for coming!

My name is Codey Winslow, and during Summer 2020, I worked on Optimized Learning with Virtual Reality thanks to the Oregon NASA Space Grant Consortium.

# The goal (10 s)

The purpose of this project was the investigate the viability of VR technology as an educational device for industry skills. More specifically, to teach skills that require some kind of environmental interaction that would be expensive to mock. I would do this by setting up an experiment where subjects would learn using a VR application I develop.

# The initial idea / final (40 s)

Originally, I wanted to model a tactile skill like playing cornhole and a procedural task in VR. In the experiment, one group would learn using this model while another would learn by watching a video. Both would then be tested on the real thing to measure how well they learned each.

As it turns out, my mentor had some issues with this, particularly the accuracy with which I would be able to model throwing a bean bag for the cornhole model. Instead, we settled on a much more approachable project: 3 levels of procedures for a touchscreen UI. This would allow me to develop one application usable for the test and the VR learning environment.

# Creating UI / Code arch. (3 min)

[How I wrap functionality in custom classes]

Each UI control has some functionality based on whether the pointer is hovering over it or if it is clicked. To achieve more control over how this behavior communicates with my procedure framework, I wrapped each type of control on the interface in my custom classes that all inherit from a ‘BaseControl’ class. This way, all controls trigger an event when they are interacted with, passing on information about which control it was. This notifies any other object that may be subscribed to this event that a control was used. This is known as the observer pattern.

[How procedures work]

With this system in place, building a procedure is somewhat trivial. At a high level, I just need to define my expectations for the procedure, and whenever a control is used, I inspect the behavior to see if it matches my expectation.

For the level 1 procedure, I created a list of functions known as “Steps.” I used an enumerator object to indicate my current step, and I move through the list as the user completes steps successfully. These step functions are subscribed to the control events in order to inspect the user’s actions. If the step is successful, the enumerator is moved to the next step, but if they make a wrong move, the procedure is marked as failed and it is ended.

The level 2 procedure was very similar to the level 1, except a set of steps is only tied to its respective stimulus, or in other words, a specific light being shown. To implement this, I created a class for each stimulus and set of expected steps. This allowed me to define any kind of stimulus action if that class was chosen and then expect any number of actions until success or failure.

Creating the level 3 procedure was quite different. In some ways, it was easier. I only had to define success and failure conditions, and if either was met, the procedure was ended. The nature of this procedure was to use the controls to turn on and off a set of systems without running out of power or overloading the current limit. This just required some backend system to keep track of values and allow the user to turn systems on and off. The most challenging part was ensuring that out of the randomly generated options, there was a guaranteed way to succeed.

[How I manage procedures]  
The last important part of getting this procedure system up and running was to have the procedures reliably execute in succession. This was done in the ProcedureController class shown here. It just runs the current procedure’s update loop if it is running, prints the success status of the previous procedure if it finished, and starts the next one when the Begin button is pressed. For the purpose of testing, extra functionality was added to allow random access to procedures.

[Tutorials]

This was really all the code we needed for the testing part of the experiment, but the virtual reality part needs Tutorials to actually teach the procedures to the user. This turned out to be one of the hardest parts of this project! Conceptually, it didn’t seem too difficult: make a version of each procedure where messages are shown explaining what to do. This is actually exactly what I did, but not only ensuring the tutorial didn’t break, but also that the tutorials were effectively teaching the procedures were what took the most time. More on that later.

# VR in Unity (1 min)

[SteamVR plugin]

So how did I implement VR into this application? Luckily, there is a very popular platform for VR applications and games called SteamVR. The company that runs SteamVR, Valve, created a plugin for Unity that makes developing for this platform trivial. After wrestling with version discrepancies between the plugin and Unity, I was able to insert a prebuilt object representing the player into the scene and BOOM, I have VR capabilities.

[Touchscreen in UI]

Getting the UI to play nice with VR gave me quite a bit of trouble, though. The SteamVR plugin has a component I can add to my controls which detect hand collisions and trigger that control’s click event, which worked perfectly with things like buttons and toggles. Sliders, on the other hand, don’t work so nicely with this system. At this point, I was running out of time and couldn’t afford to write a whole system to calculate position deltas from the VR hands and place the slider accordingly. With another stroke of luck, the plugin contained a Linear Drive component which allows a user to “grab” an object and move it between two points: much like a slider! This worked great, you just need to use the grip controls to move the sliders instead of sliding your finger against them.

# Testing / adjustments (1 min)

[Family testing performance]

Now that I had a functional application, the next step was to test it to make sure it not only worked as expected, but also was clear in its teaching procedures.

I had family members who vary in tech skills test my application, but they all failed miserably.

[Adjustments made]

After this round of testing and feedback, I changed some of the wording and added extra hints in the tutorials. The next round I made using my family members went much better, but what seemed challenge them was the difficulty of the procedures. I revised each procedure to require less steps and, for the final procedure, display what each control represented. My final round of testing went much smoother and with hardly any time left, I was prepared to polish my experiment design and perform it.

# Getting approval / didn't test (1 min)

[Why approval was pursued]

My mentor, Dr. Phil Howard, brought up the question of getting approval for using human subjects, which hadn’t even crossed my mind. I was advised to contact the Institutional Research Board to pursue approval for my experiment.

[Process delayed testing]

This process required a lot of documents to be written and verbose information. I had to fill out and revise an application a few times, as well as write an Informed Consent Form for participants to sign. I recorded my training video and continued to polish my application while awaiting the verdict, but my 9 week internship window came to an end before receiving an answer.

[Verdict]

# My experience (1 min)

[What I gained from the experience]

I want to take a minute to highlight some of the valuable experience I gained from this project. Being a remote internship, I didn’t have the luxury of working in a typical work environment or next to a team of experienced professionals, but the experience I had just working with my mentor to solve problems and come to agreements on design decisions was a great learning experience.

Designing and developing a project like this helped me build the skills necessary to work on large code bases. It’s so important to be descriptive in your code, comment sufficiently, and design with the intention of modification and extensibility in the future.

Perhaps the most valuable experience I gained from this internship was being able to work alone, at home, and stay focused. When you’re at home with people and things that can distract you easily, those skills are challenged, and you are forced to hone them if you wish to succeed. And who knows, maybe this kind of experience will be a new normal for development positions, so it doesn’t hurt to strengthen these skills.

# Where to go from here (1 min)

[Strive to perform experiment, what else should be done to answer the question]

So even though I didn’t receive a response from the IRB in the internship window, I did eventually get approval. I intend to perform this experiment on my own time.

I think the question this experiment will answer follows more closely with “can learning be done effectively in VR?” If the answer turns out to be yes, I think the next step would be to test whether or not physical skills can be learned better in VR. This was what I intended to ask in the beginning, but more time and perhaps more engineers would be better suited for that project.

[My thoughts on the matter]

I think world of VR is important to explore and use as an educational tool. VR is already being used on a large scale as an experiential tool, but such a groundbreaking facet of technology has a world of potential, we just have discover and perfect that potential.