

# Laura Lantz – Work Samples

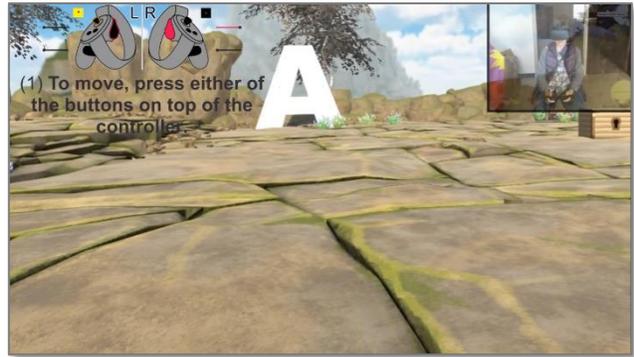
## 1. High-fidelity prototype (and user testing tools/workflow)

*Created for Summit, Virtual Therapeutics' in-development story-based game for stress resilience training.*

**Goal: To observe how novice VR users interact with the Summit navigation and inventory systems during a tutorial-like experience.**

### Design and development

I created a high-fidelity VR prototype in Unity using existing art assets (plus a small number of new assets: three alphabet letters, a treasure chest, and a key). I used the freely available Virtual Reality Toolkit (VRTK) and my own custom code to implement rough working versions of our planned navigation and inventory systems.



**Lo-fi tutorial instructions**—To support quick, easy iteration of the tutorial script, I chose to have the script be read aloud by the facilitator during user tests rather than recorded. Based on the tutorial instructions script I was given, I generated a non-linear facilitator script with additional supplemental instructions and clear criteria for script advancement. I also built a system into the prototype to play audio cues for the facilitator. Each cue either indicated that the player had met the current criteria (prompting the facilitator to read the next scripted instruction), or (based on player behavior and timing factors) would direct the facilitator to read a supplemental instruction or break script and check in with the player.

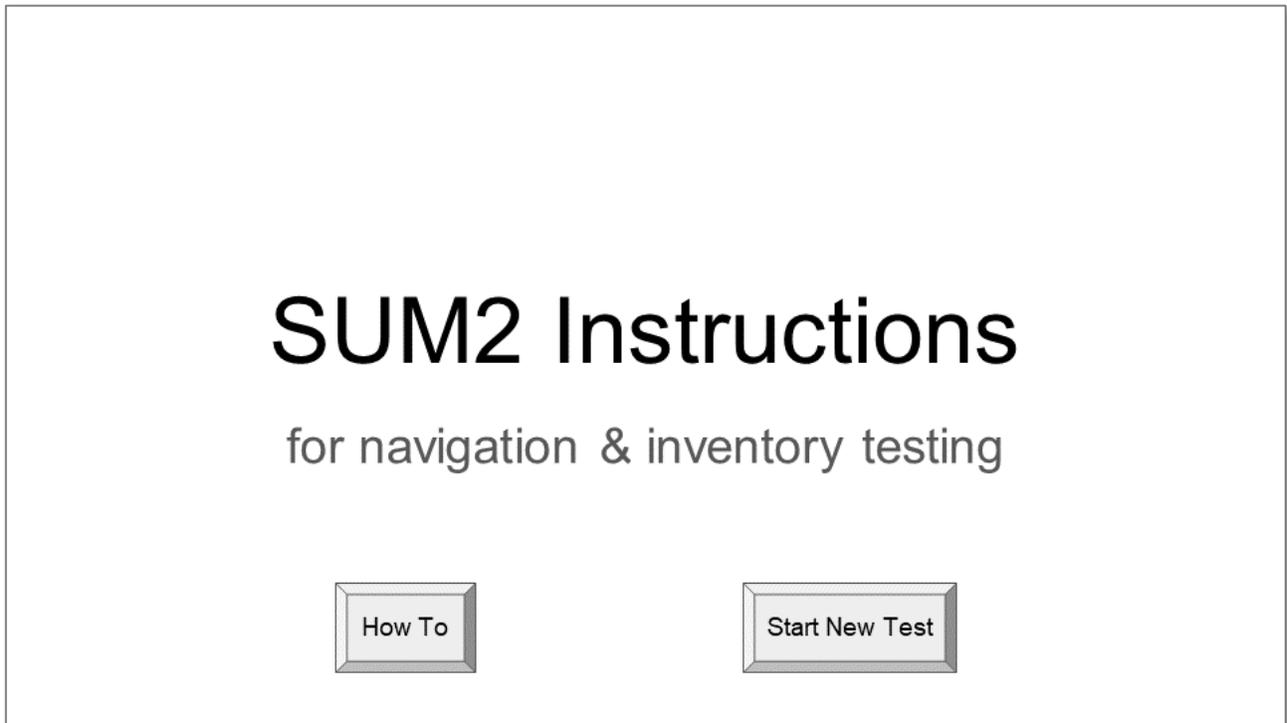
**Integrated script prompts and controller input visualization**—I quickly found that manually advancing the script in sync with audio cues was cumbersome and error-prone, and realized that the script could be integrated more directly into the prototype itself. In addition, I wanted to see exactly which buttons the player was pressing/touching at any given time during the session. I initially went about this by creating a standalone "input visualizer" in Unity but discovered that Oculus controller inputs are only detected by the application that currently has focus. An input visualizer app running in the background would never be able to respond to button presses while a player interacted in a separate VR experience.

I was able to integrate the facilitator script and solve the input visualization problem by using Unity's multi-display feature to add a second window display to my prototype. This second window shows the input visualizer, and below it, the current tutorial script instruction. I adjusted the audio cue system to update the instruction display appropriately, and immediately the facilitation role became easier and more reliable.

**Recording user testing sessions**—For recording, I created a custom Open Broadcaster Software layout. In this layout, the player's head-mounted-display (HMD) view takes up the full screen, with a webcam view of the seated player overlaying the upper right corner and the second window (showing the input visualizer and current instruction) overlaying the upper left. System audio output and HMD mic audio input are also recorded (in later layouts I added another input for clearer facilitator voice capture). Our recorded sessions are useful for reference, comparing and contrasting across sessions, and sharing research highlights.

## Early facilitator script (excerpt)

This version, from before I incorporated the script into the prototype, was designed to be viewed on tablet.



Give instructions exactly as they are written.

On each slide, read the instruction at the top of the slide aloud and then wait and listen for a sound.

Whenever you hear a...

- **Upbeat synth (progress):** Tap the **green highlighted text** to progress to the next slide.
- **“Brrring” (supplement):** On the current slide, identify which supplemental directions apply to the player’s situation and follow them. Tapping **yellow highlighted text** will return you to a previous slide.
- **Descending chime (break script):** Go off-script and answer questions/volunteer information as needed to help the player complete their current task. Once they trigger the chime, resume following the script.

Note any script changes you find yourself wanting to make during testing but wait to make these changes until you are between tests. This way it is clear which version of the script was used for each participant.



Non-linear links between slides let facilitators jump to the appropriate place in the script based on the audio cue, but if a player triggered multiple audio cues in rapid succession it was easy to fall behind.

(1) To move, press either of the buttons on top of the controller.

Tap here at progress SFX (player hits either button on top of either controller)

**Supplemental instructions (only give if you hear “brrring” sound)**

- Repeat the instruction at the top of this slide.

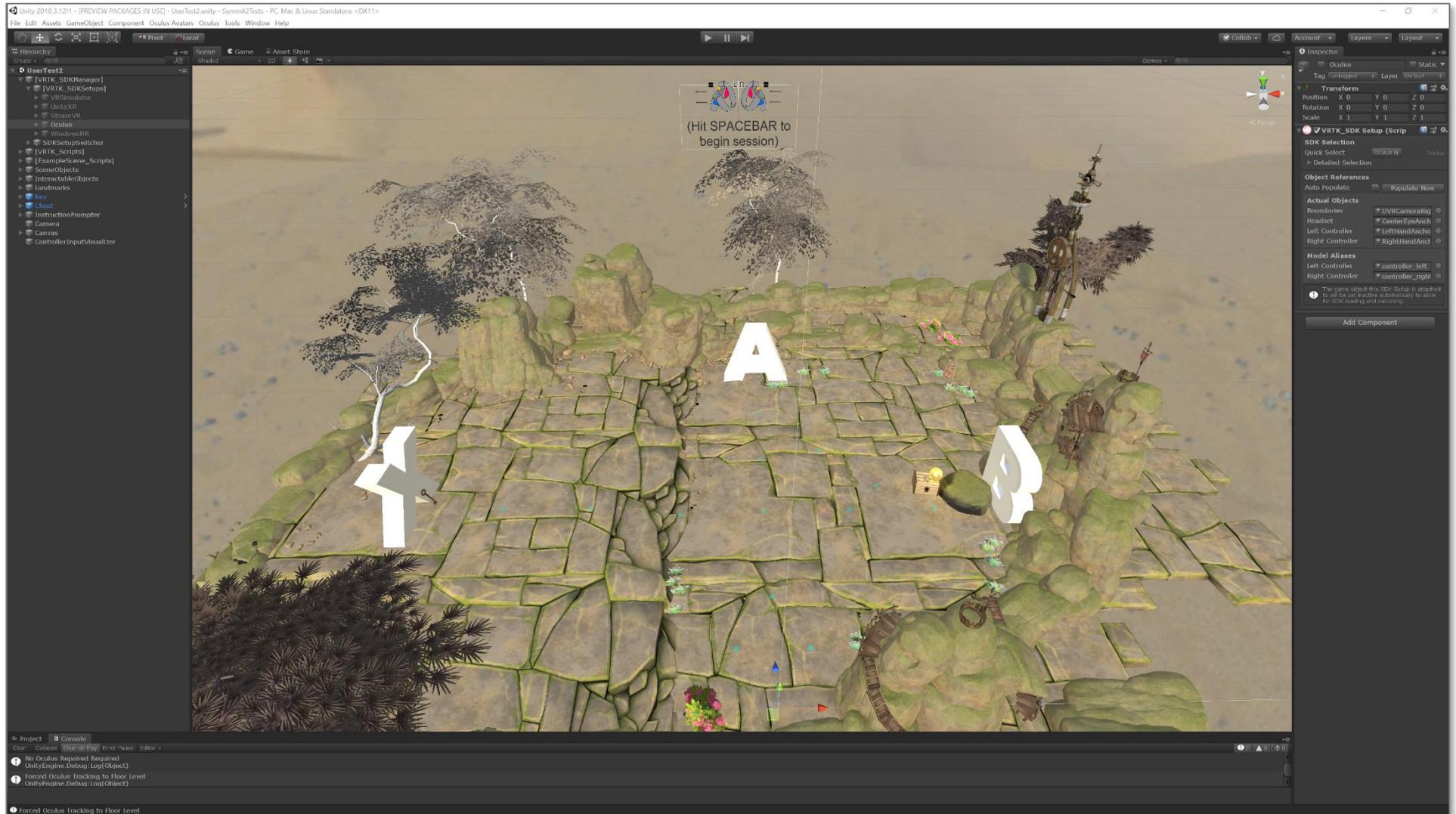
(2) Now aim the beam where you want to go. Those glowing circles show you where you can move.

Tap here at progress SFX (player's teleport stream locks on to a destination)

**Supplemental instructions (only give if you hear “brrring” sound)**

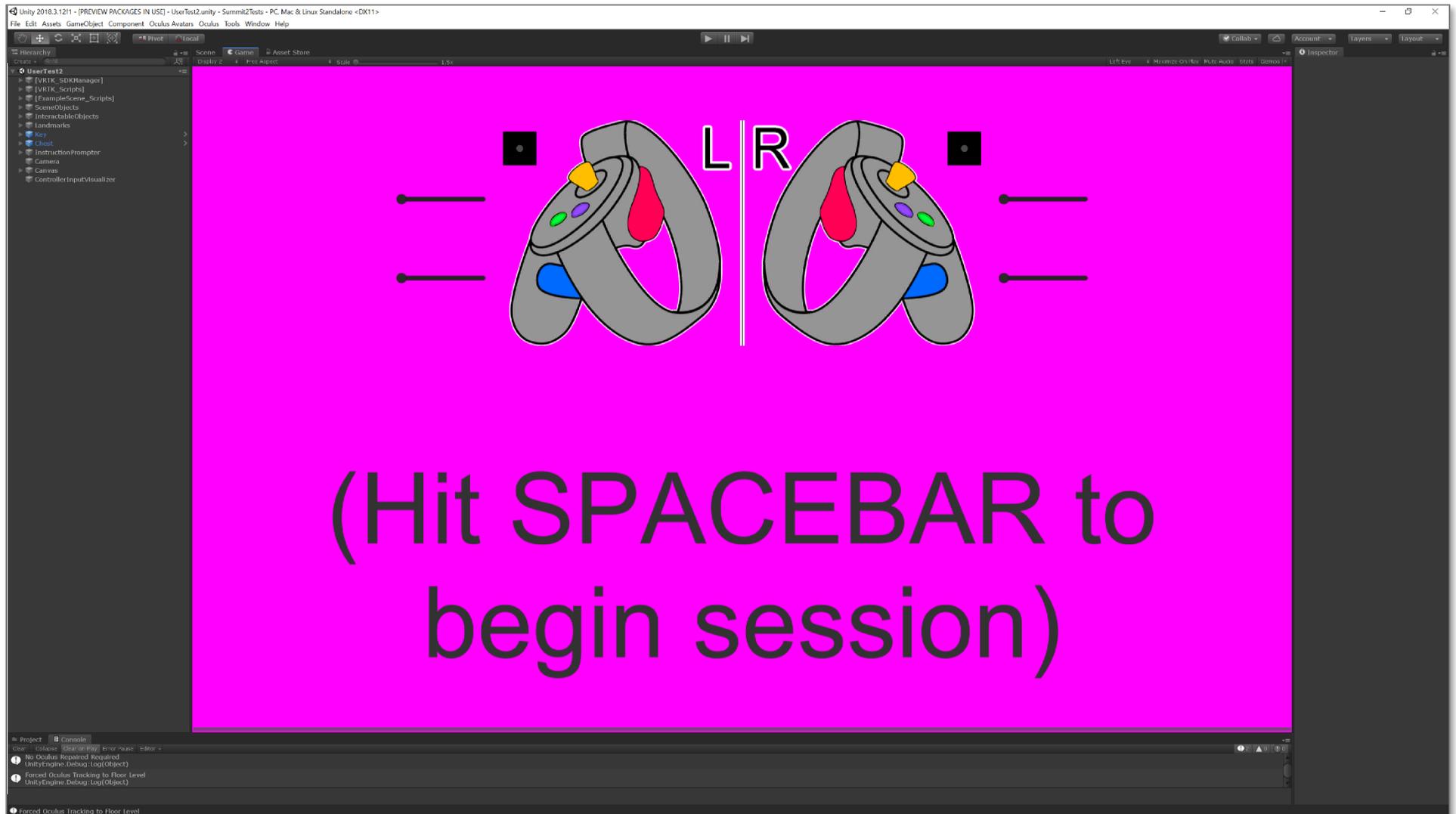
- If player is not pressing a teleport button: [Go back to instruction 1.](#)
- If player is pressing a teleport button, say: **“Point the beam at one of the glowing circles.”**

## Unity prototype – Editor view



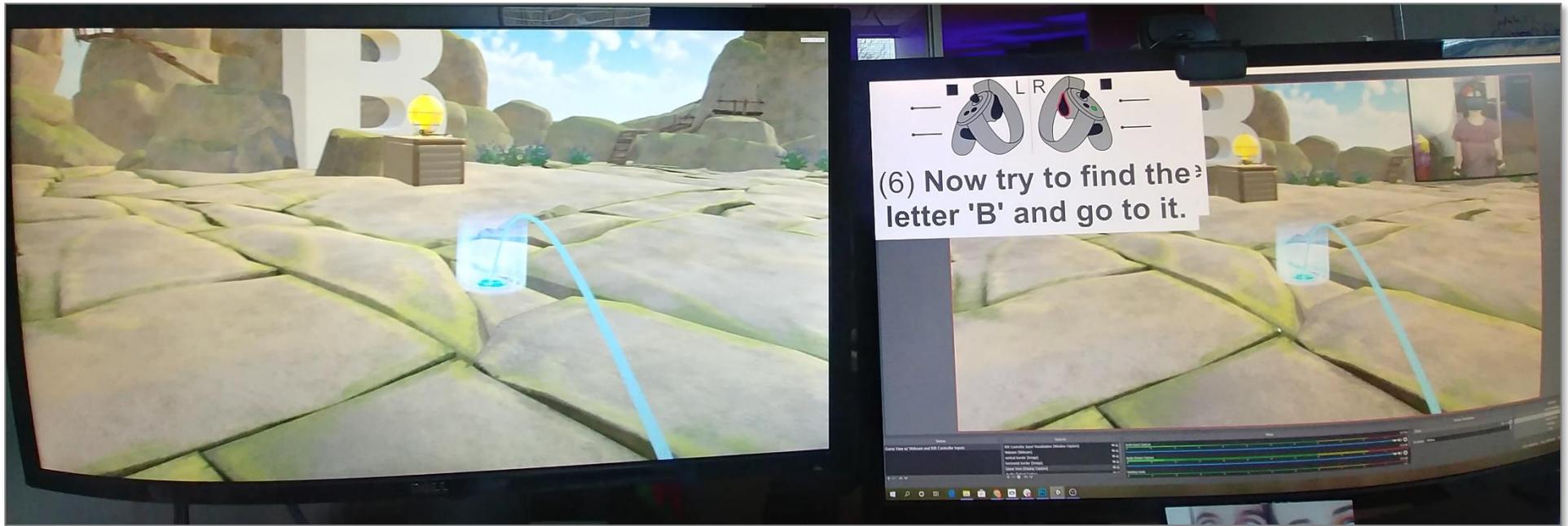
Although the second monitor display is visible beyond the VR environment in Editor view, during play it never renders to the HMD, only to the second monitor.

## Unity prototype – Second monitor display



The background color is easily changed; here I experimented with chroma keying using fuchsia, before realizing that I could simply chroma key using white.

## Unity prototype – Facilitator view



This is what a facilitator sees while running a user test. The left monitor shows the primary prototype screen, which mirrors the player's HMD view. The right monitor shows the prototype's second monitor display. Behind that, OBS is visible, showing a layout compositing HMD view, second monitor display, & webcam.

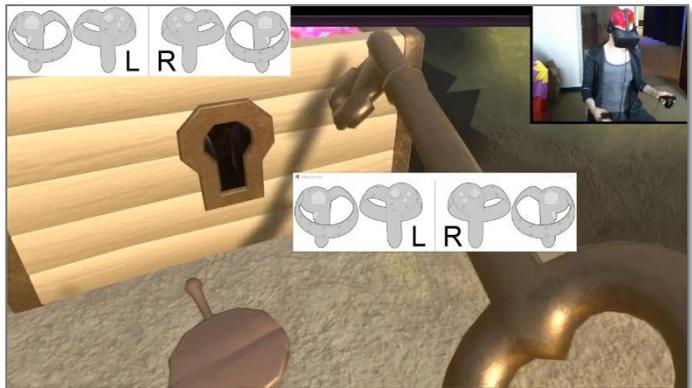
The facilitator still reads instructions to the player, but displayed instructions automatically advance based on the player's actions, and alert the facilitator should they need to break script to assist a player who has been stuck for too long a time. If a new instruction is triggered while the facilitator is still reading the previous one, they switch to reading the new instruction—the facilitator is simulating automated pre-recorded voice-over instruction, and the automated instruction is intended to behave the same way. In the meantime, fine-tuning script wording is as easy as changing a text file between user tests.

## OBS layout iteration

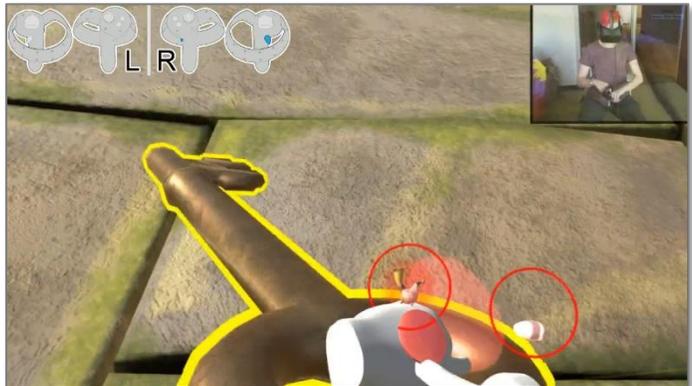
The initial OBS layout was simply the HMD view with picture-in-picture webcam footage.



The next pass incorporated the standalone input visualization app (visible in the screenshot in a window on top of the OBS window) as another picture-in-picture in the opposite corner.



Making the input visualization display background transparent and the webcam view semi-transparent reduces how much they infringe on the main HMD view.



Improved controller images show information more clearly while taking up less screen real estate, and instruction prompts are now integrated into the prototype and included in the layout.



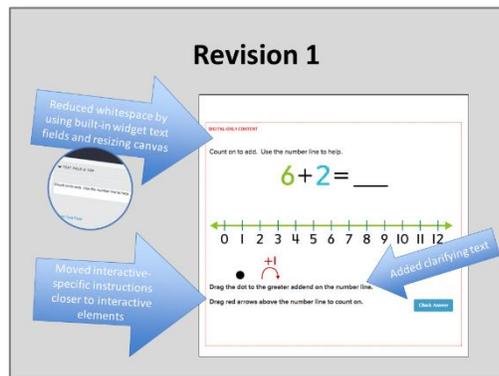
## User test videos

See **01-playtest-start.mp4**, **02-inventory-intro-a.mp4**, **03-inventory-intro-b.mp4**, and **04-inventory-task.mp4**.

## Results

The high fidelity VR prototype and user testing videos increased our company's internal interest in and openness to user research. In addition, I was asked to adapt the input visualizer for use across multiple VR projects as an aid in development, testing, and debugging. It's likely that in the future we'll further extend the input visualizer as a tool to assist with on-site and remote troubleshooting and IT support.

## 2. Design process presentation for fine-tuning layouts



While working as an instructional designer at McGraw-Hill Education, I was frequently consulted by subject matter experts who wanted to improve the usability of their interactive content. To help give insight into my process, I created and shared a presentation showing how I apply multimedia learning principles, highlighting some potentially-unknown features of our content authoring platform, and encouraging an iterative approach. (See **Lantz-InstructionEvolution.pdf** for full presentation.)

## 3. User research - Clinician interviews

At Virtual Therapeutics I am actively conducting user research with clinicians who work in a variety of healthcare roles and environments. Although I do not have permission to share materials, I'd like to highlight some aspects of my process.

Initially there was little internal buy-in, so my efforts focused on demonstrating the value and utility of user research. I was able to do this in a low-stakes way by interviewing stress-resilience-focused and biofeedback-focused clinicians from my own extended personal network. As a side effect of this approach, my small sampling of clinicians represented a broad range of healthcare environments—a small clinic, a large hospital, and a solo practice—which gave insight into how clinicians' needs and priorities are influenced by the particular environment in which they work. I am now seeking out stress-resilience-focused and biofeedback-focused clinicians who work in the Veterans Health Administration (America's largest integrated health care system, with its own unique system of funding) in order to round out my understanding of the variety of environments in which Virtual Therapeutics systems will be used.

This approach has also enabled me to develop a broad, foundational understanding through conversations with more-accessible clinicians, winning over our internal stakeholders with actionable insights. Now, as I gain access to meet with our partner clinicians, I'm able to conduct briefer, more targeted interviews, investigating whether my initial learnings apply equally for our partner clinicians and focusing on their unique knowledge and perspectives.