# Development of a multipurpose sensor tracking script for evaluating adaptation to new dynamics with virtual reality technology

By Erik Watterson

Mentored by Dr. Kristen Macuga

## Abstract

When people first learn how to use a tool or when they switch from one tool to another, they must adapt to the new dynamics of the human-tool system. To measure the adaptation within a virtual space, the software must first be coded. A virtual reality script was created that will be used as a multipurpose template for experiments that will evaluate these adaptive processes in virtual space. The script that was created tracks hand movements by the time the participant enters and exits sensors, the sequence of sensors they entered, and the hand that they used to enter them. This script bore its challenges along the way, such as a difficulty in programming different logic for each individual hand. This script will be used by the CARVE lab to test various adaptation paradigms using human participants. The plan is to continue and extend this project in the lab by adding additional features like foot tracking capability to the script as well.

## Introduction

As stated in the URSA-ENGAGE proposal, video gamers can adapt to new controls that drive their simulated movements. Athletes can learn the correspondence between the forces applied to their sports equipment and the resulting actions. For instance, when using a new bat, tennis racket, or set of golf clubs, athletes can adapt and recalibrate their predictions. In more general terms, when people first learn how to use a tool or when they switch from one tool to another, they must adapt to the new dynamics. Some of the things that the Cognition and Action in Real and Virtual Environments Lab (CARVE Lab) tries to examine are these processes by using virtual reality technology to selectively manipulate sensory feedback and measure changes in movements.

One benefit of using virtual reality technology here is that it allows us to create endless tools for the participant to use. Just about any tool one can imagine can be created using 3D models. Another benefit is that it allows us to selectively manipulate limb and tool dynamics. Finally, it allows us to track participants' movements and record their accuracy and response times. This third component is the one that I worked on for my URSA-ENGAGE award.

A Python script was produced using the Vizard Virtual Reality Developer Toolkit to be used as a template for various future lab experiments investigating adaptation processes. The idea was that this template script could be modified to test how various stimuli or tools could be used in different ways or combinations of ways. The developments for this script accomplished in the past year include: Placement of sensors and visual boxes that are dynamically anchored to the virtual space based on a few physical attributes of the participant; Data tracking and record keeping that encompasses which hand has moved into a particular sensor box, the time it took to move to those locations, and an overall time for the trial; Easy to use trial setup functionality so that any changes to any constant variable that need to be made by the research assistants are encapsulated away from the rest of the logic.

## Method

The general method that was used to construct the script started with using Vizconnect, a built in tool inside of the Vizard Developer tool kit that produces a secondary script that allows the connection between hardware trackers and virtual trackers. Once the trackers were working, a Vizard manager class object was created that associates the two types of trackers together and allows for them to interact with associated sensors. Once the connections were in place, coding of the virtual environment was started. I created four standard, two start test, and one stop test box objects and placed the boxes throughout the virtual space according to the distance between the shoulder and the foot, the shoulder and the elbow, and one shoulder to the other shoulder. After completing the distance logic, I created sensors that attach to the boxes. These sensors were then associated to the manager class, so that participants can interact with the boxes. Once the boxes and sensors were placed, tracking logic for the sensors could be created. The logic for entering the four standard sensors checks the current hand that has entered it and tracks the time of when it was entered. Next, I adapted the entering logic for the two start sensors so that they only allow the participant to start the trial if both hand trackers are inside of the sensors, and I created exiting logic for the two start sensors. This logic starts the trial timer base on when the first hand leaves the first start box. The final logic created was for entering the stop sensor, which ends the trial and prints the data to a text file.

## Results

Once it was completed, the script was implemented with the logic to track what sensor was activated, how long it took to activate it, and with what hand a participant entered a non-start sensor after a trial starts. Once the trial ends, the data that has been recorded is written to a text file in four lines. The first line is in the text file is used for what sensor was activated, the second is used for what hand activated it, the third is used for the time it took to reach it from the last sensor activated, and finally the last line is for the total time it took to complete the trial. When a participant ends a trial, the data is separated by two new line characters, and the data write logic is repeated.

The code was easier to produce thanks to refactoring previously programed virtual lab tests. This improved the implementation of the manager class, the boxes and sensors, the attachment of the sensors to the boxes, and the sensors' enter and exit functionality.

Some issues arose when multiple trackers were implemented. This caused issues in that sensors could be entered and exited twice. This doubled any effects that would occur with any of the entering or exiting logic on any of the sensors. To accommodate this, most of the script was developed and tested with a single hand tracker then refactored to encompass a second hand tracker. The doubling effect was eventually overcome by calling a manager class function that returned a list of trackers that are located inside of a specified sensor. If no sensor was specified, a full list of the program's trackers would be returned. By knowing the list of hand trackers, a comparison can be made to see if any of the trackers in the list have entered any of the sensors. This allows the ability to verify if a particular tracker or multiple trackers have entered any sensors.

We had originally envisioned the script to allow foot trackers as well, but due to time limitations and my level of experience, the foot trackers have not yet been implemented. However, we have formulated a way to implement these, and we plan move forward with this extension, as I'll be continuing work on this project in the lab whenever my schedule allows.

## Discussion

This script will be used and modified to collect data from human participants for research experiments on adaptation to novel limb and tool dynamics. At present, the script can be used to evaluate time intervals between box changes, the sequence in which the sensors were activated, and the associated sequence of the hand trackers, but it is flexible with regard to stimuli and experimental procedure. This makes it a useful template for collecting and recording hand movement data. The script can be easily modified to change the orientation and visualization of the boxes. The box visuals can also be turned on and off for testing visually-based versus nonvisually-based adaptation processes. In summary, I have created a flexible, multipurpose script for measuring changes in hand movements. It will be implemented with virtual reality hardware and used for experiments that will evaluate a variety of sensorimotor adaptive processes.