The Clamshell: Rethinking the Virtual Reality Interface

Justin Berry
Yale School of Art
Yale University
New Haven, Connecticut, USA
justin.berry@yale.edu

Lance Chantiles-Wertz
Mechanical Engineering
Yale University
New Haven, Connecticut, USA
lance.chantiles@yale.edu

Isaac Shelanski
Physics and Philosophy
Yale University
New Haven, Connecticut, USA
isaac.shelanski@yale.edu

***Abstract*— This project seeks to challenge the dominance of button-based interfacing with Virtual Reality by creating a platform with which anyone can customize their own virtual reality controller based on any sensors of their choosing. After two prototypes and as many demonstrations, the project has successfully yielded a proof of concept that showed results both in the technical creation of a custom VR controller and in the effects of creative rethinking of the physical interface on the VR experience.**

Keywords—virtual reality, controller, haptic, interface, design

I. Introduction

An ongoing concern in creating virtual reality experiences is that the current paradigm for controllers is based on leveraging the same interactions inherent to gaming. Insofar as virtual reality is used to make games this is acceptable, though still restrictive, but if we challenge the hypothesis that virtual reality is a medium ideally suited to gaming then we arrive at the conclusion that different forms of interaction might be needed for it to reach its full potential. Abstracted controls are ideal for interacting with abstract mediums, such as a flat television screen or computer monitor. We interact well with a symbolic and representational space via symbolic representations such as buttons, triggers, or joysticks, but those same representations are less intuitive in an immersive environment that appears to exist concretely to the end user. If I want to grab an apple in virtual reality, I want to reach out and grab it the same way that I would in the real world rather than clicking the grab button while pointing at the object or selecting it from a menu. In order to test this hypothesis, we have sought to create a custom controller that does not use buttons but relies on simple physical gestures such as grasping. By removing the abstract interactions that are borne out of the gaming paradigm the goal is to create more intuitive interactions that are accessible to a broader range of people from more diverse backgrounds. In order to maximize accessibility, the controller was not designed after a glove that had to fit or an object with buttons that users had to be able to reach. Instead we opted for a clamshell inspired design that could be handled by anyone able to grasp from opposing directions. The primary point of interaction is reduced to the intuitive action of opening or closing the controller. Additionally, by removing interfaces such as the trigger, we are able to avoid cultural and contextual implications, such as using gun-like interfaces to engage the virtual world. By retaining concrete interaction but moving away from a skeletal hand based model we aspire to make interactions accessible to differently-abled individuals.

Concurrently in our process we realized that the clamshell might not necessarily be the end-goal for virtual interactions. Challenging the role of buttons and triggers as the primary physical interaction with virtual reality opened up numerous possibilities for creativity on the physical aspect of virtual reality. We therefore decided to build a framework that would allow anyone developing in virtual environments to be able to simply communicate inputs and outputs between the physical and digital world. Our groundwork and research in creating the software and hardware for the "clamshell" was extrapolated to make a device that anyone could attach sensors (force, infrared, air quality) or outputs (lights, motors, speakers) to that would communicate directly with a game engine. Our objective became to create a device that would allow anyone, regardless of technical background, to engage in creative modification of the physical interface with virtual reality.

II. BACKGROUND

One of the first anecdotal inspirations for this project came from the challenge of teaching people to use Tilt Brush [4]. Tilt Brush is a spatial drawing program produced by Google that was one of the earliest well-developed applications for mass market VR and that continues to be a useful program for on boarding novice users to the medium of VR. However, even simple instructions such as telling people to 'pull the trigger' often leads to confusion for first time users. We noticed that the amount of time required to grasp the interface seemed longer than expected, even for people familiar with similar interfaces. Given that the program does a good job of guiding users and implements a straightforward design it seemed that the confusion was potentially not a design problem but an interface problem.

The general strategy we employ to teach and explore emerging mediums at the Yale Center for Collaborative Arts and Media is to regard them from an art historical perspective that sees creative mediums as having internal properties that are distinct. By understanding those unique qualities of the medium we better understand how it mediates the content it conveys and shapes the interpretation of that content. Often, we explore the primary verbs of a given medium, such as 'to listen' for music, and 'to watch' for film, or, in the case of VR, what we hypothesize as 'to be'. After numerous conversations, a pattern of observations emerged that people described VR as being a concrete rather than abstract medium [9]. Though virtual simulations are inherently representations and symbolic, the user does not perceive them as such. We do not think of a flower in VR as a drawing of a flower or as a model of one, but simply as a flower. When we are walking across a plank in virtual reality we do not fear that we will lose the game if we fall, but rather that we will actually fall; this feature of the medium is what makes it so effective at treating phobias and helping people with PTSD [7]. Observing this quality of the medium led us to believe that an interface utilizing more concrete interactions would allow for more intuitive and accessible experiences.

As it is described here, a concrete form of interaction means one that does not rely on symbolic logic but functions as 'the thing itself'. A concrete interaction for a door is when you push on the door and it opens, an abstract interaction would be to point and the door and select the 'opening' option via a button press or a menu selection. In the context of gaming, abstract controls are required because we are not interacting directly with the virtual world but rather navigating it through the abstract two-dimensional lens of the screen. Abstract values such as points or character levels take on immediate relevance and the use of buttons or other interfaces is intuitive exactly because it is necessary. In a game where one needs to acquire a certain number of apples the specific features of the apples are generally irrelevant, only the metric they engage has value. Compare this with a person shopping for apples at a grocery store. In a grocery store, many people will inspect each apple that they buy, inspecting them for soft spots or ripeness and often use subjective evaluations such as looking for aesthetically pleasing features.

Fig. 1 Examples of interface models produced by group workshop.

III. METHODOLOGY

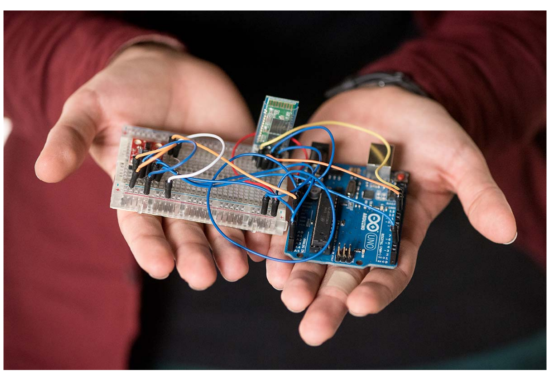
The "clamshell" is of particular relevance when the predominant method of interacting with a virtual world is to pull a trigger while pointing at something. A form of violence as a basis of interaction is especially troubling in a concrete medium where we experience the world with a deeper emotional connection than we do an abstracted world. With this in mind the clamshell is intended to only offer a form of input but also a form of haptic feedback that uses a magnetic field to alter the attraction or repulsion of the two facing halves of the device. The goal is to have a mechanism by which the virtual world can be 'felt' or touched. The act of clasping is one of the first actions we instinctively know in order to understand the physical world. The "clamshell" is uniquely situated to allow us to explore the new virtual world as we would naturally as children.

We began our design process by holding an open brainstorming session which was attended by faculty and students from a variety of backgrounds, ranging from artists to engineers to entrepreneurs. We provided them modeling materials, showed a selection of example interfaces, and asked them to model what they would be perceive to be an ideal interface for a virtual world. It was as a result of this session, and the diversity of results, that the project shifted focus to create a platform for getting data from different types of non-button sensors into unity. Our objective shifted from focusing on a single controller that subverted the dominance of gun-based VR interactions to creating an open hardware platform on which anybody could create a controller with any form of interaction that they chose.

Our new approach to design was therefore to first create the framework for new types of interactions between the digital and physical worlds before moving onto the specifics of the clamshell. This base framework could be expanded upon to create new models of interactions that could be easily created by anyone with a basic sense of programming - nothing past the skills needed to use the Unity game engine. The two axes of the work were the software framework and the device itself.

For the hardware, our first prototype used an Arduino Uno and HC-05 Bluetooth Module to put together a rough proof of concept. The sensors that we used were purchased from various vendors such as SparkFun Electronics and Adafruit Industries, and included a force sensor, a light sensor, a 9-axis IMU, and a motion sensor. We connected the various components of this early prototype on a breadboard, also using a SparkFun Electronics 8 channel I2C multiplexer to communicate with multiple sensors at the same time. This initial prototype served as a medium to test if our code could send various sensor values into Unity and also retrieve data

a.



b.

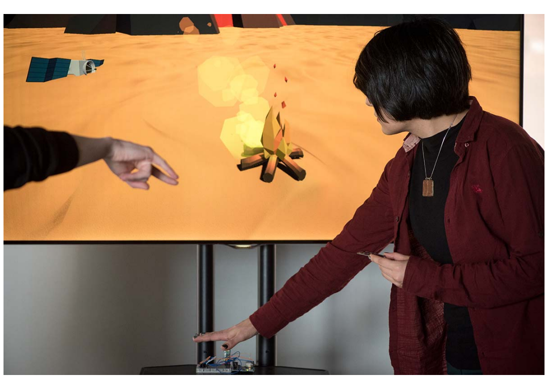
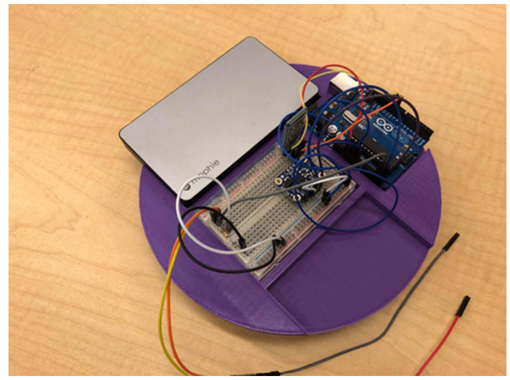


Fig. 2a. Board layout for Firepit Demonstration.
Fig. 2b. Interaction example for Firepit Demonstration.

a.



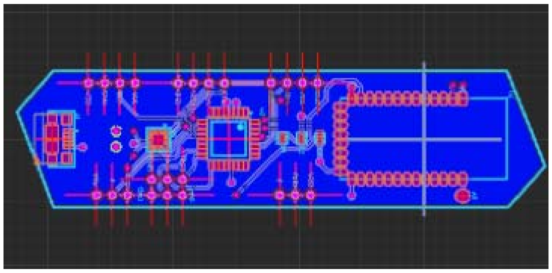
b.

Fig. 3a. Bowl demonstration internal configuration.
Fig. 3b. Bowl prototype build.

back out from Unity onto the device. The prototype at this stage did not provide the ease of use that we eventually aim for as each new sensor attached had to be rewired on the breadboard. Most of the sensors we used however used the I2C protocol which slightly simplified the process of connecting them across the breadboard, as they were centralized around the multiplexer's input channels.

The software on our early prototype had two sides to it – the code on the controller itself and the code on the computer running a Unity experience. The code on the controller only pulled data from the sensors and had very little logic or processing performed on the sensor data. The majority of data manipulation took place on the computer running the experience. Each frame of data was combined into a new string which was sent over a Bluetooth serial port. On the computer, a C# script in Unity created a serial port on the controller's Bluetooth connection port. Each line of the string was read and parsed to retrieve the data from the individual sensors. This data was then manipulated in whichever way it needed to be used in the Unity experience.

We used this early stage Arduino-based prototype to put together two demonstrations. The first demonstration involved connecting a SparkFun Electronics motion sensor to a.



b.

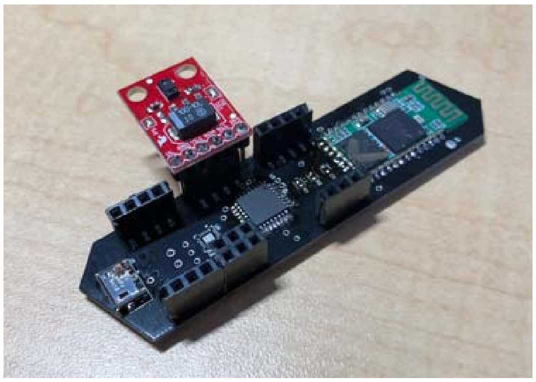


Fig. 3a. Layout for Clamshell V1.
Fig. 3b. Test build for Clamshell V1.

the Arduino and sending the data to Unity over Bluetooth. In Unity, we created a virtual environment lit by a central campfire. By passing one's hand back and forth over the motion sensor, one could change the direction of the wind in the environment, blowing the flames in various directions.

Having succeeded in getting data from one sensor into Unity, we aimed to create a second demonstration that allowed us to manipulate features of a virtual environment using multiple sensors at once, ideally with a fair amount of ease when it came to changing the sensors. For our second demonstration, we added the multiplexer to the breadboard and connected a light sensor, an IMU, and a force sensor to the Arduino and Bluetooth module.

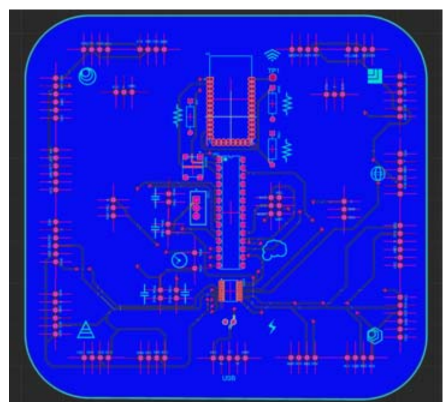
We also sought to demonstrate the ability of this controller "black box" to turn ordinary objects into controllers that have very different physical interfaces than the standard trigger-based gun-shaped VR controllers. We therefore placed our apparatus in a wooden salad bowl and built a lid to place over it, with the force sensor on the lid and an opening for the light sensor. We used this bowl-controller to manipulate a virtual ocean environment. By turning the bowl, one could change the weather in the landscape from calm and sunny to cloudy to stormy and back again. By tapping on the force sensor, one could disturb the water, creating waves and occasionally splashes in the water. Finally, by covering the light sensor or shining light on it, one could change the time of day in the landscape. The idea was that one could sit in front of this virtual ocean landscape holding the bowl in their lap and use all three interactions to modify the landscape to represent whatever qualities they preferred.

Having completed two proof of concept demonstrations with our first prototype, we felt confident enough to move onto the next step, which we saw as designing a chip that condensed our complex breadboard wiring into an easy to use plug and play chip that provides a simplified platform for sending data from various sensors into Unity. We quickly came up with two designs, using the online program Upverter. Both designs include an Atmega328P microcontroller and an HC-05 Bluetooth module. The later design also adds a TCA9544A multiplexer, along with various capacitors, resistors, power and charging modules, and a reset button.

The first of the two chips, though somewhat flawed and certainly missing several of the components necessary to perform all of the functions we desired, was enough to test the concept of streaming data from a sensor (in this case an IMU) into Unity without the help of an Arduino and a breadboard.

The second chip has been designed, but not yet tested. This second chip should allow us to test the process of streaming multiple data streams into Unity in real time over Bluetooth using nothing but our device. We anticipate designing a third and final chip that is much smaller than the

a.



b.

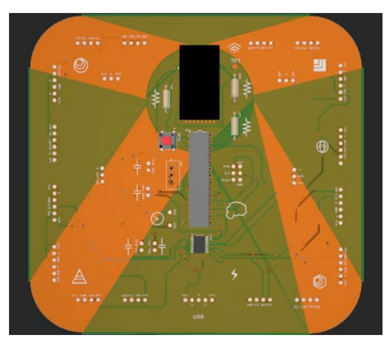


Fig. 4a. Layout for Clamsell V2.
Fig. 4b. 3D Rendering of Layout for Clamshell V2.

4.5x4.5 inches that our current design takes up and that boasts a far more elegant design for switching out different sensors, making it easy for anyone, regardless of technical expertise, to create their own custom controller by routing diverse sensor information through the neutral “black box” interface. We are also considering a further version that includes micro-USB ports through which the chip can receive data from sensors, and perhaps building our own sensors that will work when plugged into these ports.

As we look to the next steps in our project, our objective is to update, streamline, and simplify this apparatus in order to make a black box for our controller that allows any sensor to stream data into Unity. We aim to make a device so open and simple that the clamshell controller is just one of its many possible applications, allowing for myriad creative uses of the interface through which one interacts with the virtual world.

IV. Results and Conclusion

Our first prototype yielded two demonstrations, both of which provided valuable results for our project. The first demonstration, the campfire in the desert, allowed us to successfully show that we could use data from one sensor to control a virtual environment in Unity. Our second demonstration, the bowl on the ocean, revealed that we could successfully stream data from multiple different sensors into a Unity experience in real time, creating a multi-faceted VR controller.

The interaction with the virtual environment reflected the vigor of the physical action of the person holding the bowl: the weather would change faster the faster the bowl was turned, the waves and splashes were larger the harder and faster the force sensor was tapped, and the time of day more extreme the more completely one covered the light sensor. The result was a concrete correlation between the physical interface of the controller and the virtual landscape.

Our project has evolved from designing a specific VR controller to building an easy-to-use platform for creating creative and imaginative VR controllers, while maintaining the original objective of challenging the standard of gun-like controllers that operate through buttons and triggers. So far, we have built two prototypes of the device and designed and ordered a third. We have used the prototypes we built to carry out two demonstrations, which have revealed results that indicate that a creative approach to the physical interface with a virtual environment can generate a completely new experience with VR. While we have much left to do in terms of refining our device, making it easier to use, more compact, and more robust, we have successfully carried out proofs of two concepts: first, we have demonstrated that there is value in opening up creative possibilities for VR controllers that challenge the dominance of button-based interactions, and secondly we have demonstrated that such a device is possible to construct and functional. While it is not novel to have sensors communicate directly with a simulation the difference in our approach is that we are striving to make a universal sensor device capable of being customized by novice users and allowing non-specialists access to an otherwise daunting interaction with emerging technology.

References

1. Choi, Inrak; Hawkes, Elliot W.; Christensen, David L.; Ploch, Christopher J.; Follmer, Sean, 2016, “Wolverine: A wearable haptic interface for grasping in virtual reality” IEEE Xplore Digital Library
2. Choi, Inrak; Culbertson, Heather; Miller, Mark R.; Olwal, Alex; Follmer, Sean (2017); “Grabity: A Wearable Haptic Interface for Simulating Weight and Grasping in Virtual Reality,” UIST '17 Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, Pages 119-130
3. Drummond, Katie (2014). "The Rise and Fall and Rise of Virtual Reality". *The Verge* 2014.
4. Google VR, *Google Tilt Brush Website,* https://www.tiltbrush.com/
5. Hatzfeld, Christian; Kern, Thornsten A. (2016) “Engineering Haptic Devices” Springer
6. Leap Motion, Inc, 2018, *Leap Motion website*, https://www.leapmotion.com/.
7. Mura, Gianluca (2011) “Metaplasticity in Virtual Worlds: Aesthetic and Semantic Concepts” Information Science Reference, Hershey, NY
8. Mychilo Stephenson Cline (2005), “Power, Madness, and Immortality: The Future of Virtual Reality,” University Village Press
9. Sherman, William R. & Craig, Alan B. (2018), “Understanding Virtual Reality: Interface, Application, and Design” Morgan Kaufmann Publishers, Second Ed. 2019