Implementation Of U.I. in V.R.

Mehrzad Bazhdanzadeh
Programmer & Concept Designer
(647)784-0474
mehrzadb@outlook.com

Abstract

This documentation provides you with the concepts and methodologies taken to design, create relations, and improve interactions within Virtual Reality projects.

Keywords

V.R. denotes virtual reality; G.O. denotes Game object; A.R. denotes Augmented Reality; HMD denotes Head Mounted Displays; U.I. denotes User Interface;

1. Introduction

The world around us has been driven mostly by a unique twenty first century invention which shaped our daily work flow, interactions, and communications. This little device called smart phone has grown to be so necessary for us that most people cannot go about without using them.

The world around us has been driven mostly by a unique twenty-first-century invention which shaped our daily workflow, interactions, and communications. This little device called smartphone has grown to be so necessary for us that most people cannot go about without using them.

However, even an older technology existed before the introduction of smartphones - in 1978 - where it took the man to a digitized universe of its creation, virtual reality. The introduction of this technology, even though began with military use, it was a big step towards many of its uses for teaching, simulating, and gaming.

But recently, we are seeing different approaches to V.R. which may as well be focused more towards productivity and daily use. Microsoft and its hololens have been recently catching our eyes as in what would be a near future possibility of V.R. and A.R. use for our daily usage. But will it be sufficient to say that our ways of using cellular devices would be the same as using Head Mounted Displays (HMD) and two joysticks?

Certainly, before we mix the two concepts of A.R. and V.R. we must attempt to simulate a sample world within V.R. to demonstrate our powers and potential use. As such this project has aimed to immerse the user to play around with a sample world to which he could interact with attached data and possibly manipulate them.

2. Project's Goals and Achieved Implementations

While creating a User Interface (U.I.) is the main problem to tackle, we need to define how the world is laid out, what defines a data relation within a Game Object (G.O.), and what is necessary for users to understand while interacting with this 3D environment. As such the parts to address are broken down as following:

2.1. G.O. and their relations

Every G.O. may or may not have a data relation. Lets us imagine the following: while searching for Tim Horton's location we will find a specific shop with the following details on Google Maps.

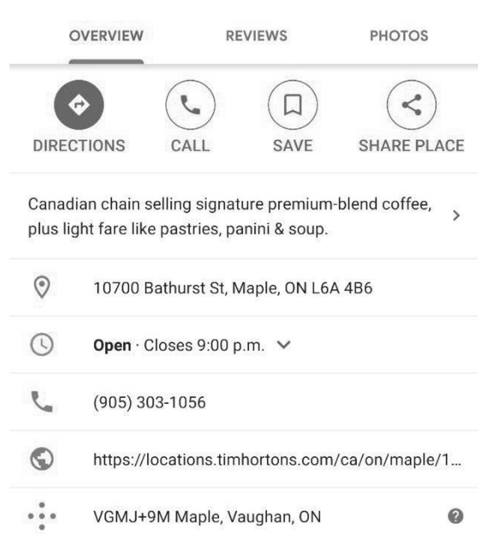

Fig 1. ***Sample of Google Maps' location data***

Now imagine in your virtual world you are close to that specific location/store, and as you are viewing the store you are able to pull the information from that virtual object. What would be a reliable way to get and display this information?

The challenging step for this small demo was to find a valuable source to test this proposition with a different approach. Instead of places, we could possibly attempt to profile people in V.R. using any social media platform. And for that Twitter was chosen to be working with for the purposes of this project for various reasons:

1. Easy to get and access profiles
2. There are a different type of profiles
3. Each profile has a dedicated feed
4. There are trending and focused tweets and topics shared by the community

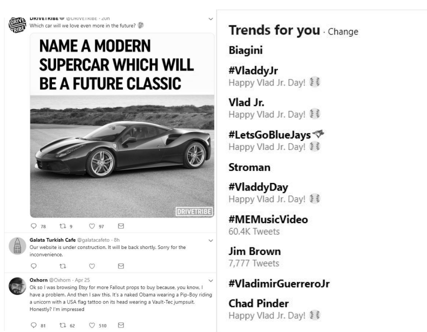

Fig 2. ***On left: tweets from different account types; On right: trending topics globally***

As such, the goal is to create two G.O. in this project to represent the binding of data within the world perfectly.

- ✓ A pillar G.O. figure represents the recent trending topics

- ✓ A person G.O. figure represents an individual, organization, or group

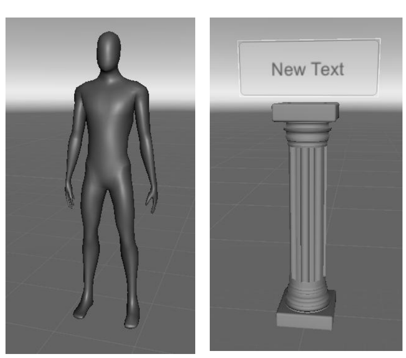

Fig 3. ***On left: Person G.O.; On right: Pillar G.O.***

2.2. User Interaction + Travel

The problem of travel arises where the need of movement becomes intertwined with the need to view from close distances. Every day we move towards objects to feel, grab, read, and analyze them. However, in V.R. there are limitations into how far away from the center point of your own room - the physical world - you can move.

As such the travel mechanism was decided to become much like point and travel. It is sufficient enough for users to understand this small concept of teleport like travel when it comes to V.R. and to combine it with touch and grab.

However, what are the objects that you could possibly grab and interact with? What would be the physical indications of “this is an object that you are able to move, and hold.” To make it easier for a normal user to understand what can be held and what not, we may use what makes the most sense to us in real life: weight and size.

Bigger objects tend to be heavier and harder to move, while smaller objects tend to be lighter. In this respect, a paper is much lighter than a statue, and as such you are able to hold a piece of paper, throw it at any object but in contrast are not able to move or push the statue.

Now, this may be reasonable enough to go through with, yet with the further development of this V.R. environment, there will be restrictions which are needed to be altered. For example, say a user attempts to open a G.O.’s door or lid. While the G.O. is not moveable by the user overall, the hinge mechanism should give the capability to the user to move that object in restricted perspective.

So, there is an interface rule introduced to the game. Only objects which certain local weight limits are to be manipulated by the users if such manipulations are intended and are easily understood in real life.

2.3. Displaying Data

The biggest and most complicated problem which one may have in V.R. is to display properties and data of a G.O. to the player of the V.R. Respectively it is not easy to simple cast windows and canvases to the world and hope users are able to travel the closest to the screen to be able to see them. But this is not a viable solution.

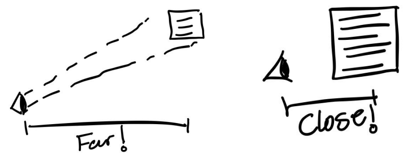

Fig 4. ***Moving in closer would allow to visualize the screens better***

The first idea was to cast the canvases as close as possible so users can have them in a certain proximity of themselves, where they can manipulate, interact, and read them as easily as they could within the V.R. world. This meant the screens are directly attached to the user, would not move about or away from the user, and are big and close enough for him to read.

But this was not reliable enough for the purposes of the project as it meant so much collusion at a time for the user, and it simply castes the flat monitor screen in a circular manner around the user. So what became the second approach was to cast and bring forth methodology of homer select.

Fig 5. ***Ray cast, select, bring forth***

This method is effective enough to have the canvases cast anywhere near the desired G.O. and to bring it forth while necessary. However, this may as well have a greater downside when introduced among multiple users at a time. Which user will have the priority and capability of grabbing that canvas? One, or all? And if it is all, what happens where one user grabs it for himself?

As such, a simpler methodology was introduced which would restrict the content to a smaller and more connect place for a user and reduce the need of creating multiple canvases at once for multiple uses, a wristwatch screen.

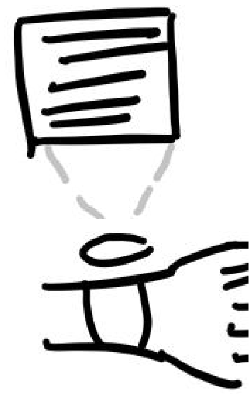

Fig 6. ***Wrist watch will be placed over the OVR hand models***

This screen type would generally be constricted to a single user and is easily scalable for the user as they can bring the screen closer or further away from themselves by moving in and out their hands away from their face. This also removes the need of having multiple screens at a time in space and instead of having them cast over the wrist in relative positions.

As such, the watch functionality is a more sensible user interaction element to add to this virtual environment. It is reliable, extendible, restricted, and expandable enough to be used for the purposes of U.I. interaction design. The manipulation of screens while being able to cast and pull information from objects is a methodology which is easy to understand for any user. And as such it has been implemented in the project.

2.4. Responsiveness Grabbing User Focus

The process of showing data may have been easy to solve, but in reality, humans are looking for responses and impulses generated by their surrounding to get validated feedback and notifications. It is as essential to know if you have pressed a button on a physical keyboard as it is on a virtual keyboard on a flat screen. So, what are the feedback and type of responses necessary within this V.R. world for users to be able to understand?

One traditional way of getting a notified is by receiving a sound from an interaction. Much like a button on a virtual keyboard, the responses generated from users interaction may result in a feedback sound generated by G.O. which was interacted with.

However, there are other possibilities for feedback functionalities. Another great method may be a color change in G.O. upon interaction. These results are best in effect where they are being ray cast and are being selected from afar. Because of the openness of the world, users need to have other methods of defining which G.O. they are pointing and selecting.

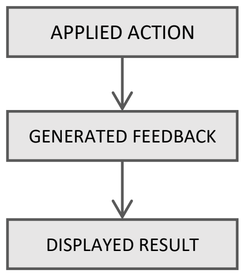

Fig 7. ***Methodology of feedback generation***

In addition, rays pattern itself is a viable solution within feedback functionality. A ray only appears to assist with the selection of G.O. and is simple and understanding enough for any user to get valid feedback from. As such this was the first option to be implemented within the project. However, it is not the best by itself.

A complex need between different components of responsive feedback system may affect how users proceed to collect data from surrounding. The more detailed, unified, and to the point this integration, the more easily understood.

2.5. Ease of Use & Combining All

As much as one would like to create it, humans are driven by simplicity in design. As the famous father of phones said, "*People don't know what they want until you give it to them.*" This famous motto from Steve Jobs has driven the core of this project.

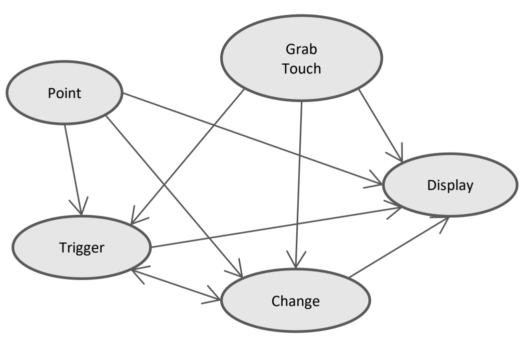

Fig 8. ***Relation between actions and result between user and G.O.***

The decisions to make while challenging the idea of relations and combinations of G.O. and elements in this V.R. world has to be so easily understood that even a child could play around and pick up on simple clues and patterns. As such, this project proceeded with little but combined everything.

The first step was to define what was the best method of interaction which is plausible to expand further. Three methods were chosen:

1. Touching with OVR hands
2. Ray casting with hands
3. Grabbing with hands

These are so easy and simple to understand for general public use.

Next step was to choose a viable pattern to display our data. To do so, this project demonstrates watch display functionality to display necessary information to users who are engaged in it.

And finally, understanding materials. As explained in G.O. and their relations, objects should be easily identifiable for users to know how to interact or even if they possibly could. This means in the future expansion of this project, relations will be so complex that a unique universal identifier is necessary to assist users to better understand the V.R. world capabilities.

3. Release and Results

With the recent release of version 0.0.1 - in April 22nd, 2018 - one can simply interact with the world defined interaction material explained in **2.5. Ease of Use and Combining All**. However, there were challenges to take care of while developing this project.

The Twitter API has been a great success in loading and sharing information from their database. It was a very reliable solution to be shared by its author *Craig Tinney* on YouTube for Unity developers. However, it was limited to specific pulls and data types that Twitter provides out of the box. Further expansion on this API will result in more complex data relation on the Twitter Application preview of this project.

To be able to correctly set up ray casting components and watch, I have generated certain prefabs which contain all the functionalities of these objects and their relation to the user. Most of the work done in this project was designed and made in house to better experiment with possible outcomes of each design. Some like cast and pull were scratched as other reliable and more understandable solutions were found. But that does not mean these functionalities would not be used further. It simply did not serve its purpose in the current version of this project.

As for set-up the following are the work environment that was used to create this project:

1. Unity 2019.1.0f2
2. Ryzen 7 2700 and Radeon 7990 (CPU & GPU)
3. Oculus Rift
4. Oculus OVR Unity Integration

4. Conclusion

The goal of this project is to show the complexity in V.R. and its workarounds to simplify user interaction in a better integration for future productivity for work and daily use. It demonstrates simple but effective interactions in a possible future life demonstration.

This project was successful to create a V.R. world where simple users are present and are tagged with their profile information. Furthermore, users are able to find and pull related data from G.O. user and to achieve the simple task of connecting to a user without using a cellular device.

Does this effectively mean that it is plausible that V.R. and A.R. would completely change how we look at our surroundings? Yes and no. If the technology is not pushed for lighter and more carriable HMD devices we may not be able to use this capability. However, this does not mean that it does not exist and cannot be developed. A future where humans and computers are more interconnected is where our aim should be focused on. And V.R. and A.R. are where this connection lies.

Resources and Research

- ✓ Craig Tinney. 2017. Twitter API Access Tools for Unity Demonstration. (November 2017). Retrieved April 26, 2019 from <**link**>

- ✓ Google Developers. 2017. Designing Screen Interfaces for VR (Google I/O '17). (May 2017). Retrieved April 27, 2019 from <**link**>

- ✓ Anon. 2017. Beyond Flatland: User Interface Design for VR. (June 2017). Retrieved April 27, 2019 from <**link**>

- ✓ Brandon Vigliarolo. Microsoft HoloLens: Cheat sheet. Retrieved April 27, 2019 from <**link**>

- ✓ Bernard Marr. 2018. 9 Powerful Real-World Applications Of Augmented Reality (AR) Today. (July 2018). Retrieved April 26, 2019 from <**link**>

- ✓ Isl and Isl. 2018. Augmented Reality, In Real Life. (July 2018). Retrieved April 26, 2019 from <**link**>