FIVE - Flexible Immersive Virtual Environments

A New Framework for Designing Environments for VR

Brett Schumacher, Malik Khadar, Evan Suma Rosenberg

University of Minnesota, Department of Computer Science

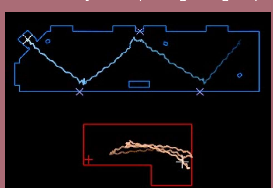
Introduction

Virtual Environments (VEs) – digital spaces, in our case created for Virtual Reality (VR).
Flexible/Immersive – VEs scale flexibly to increase immersion for users

FIVE is meant to be a framework for VR developers to use to create environments that scale dynamically to a user's physical space to increase immersion by allowing them to navigate the entire space.

Motivation

Physically navigating a VE requires physical space, if the VE is too big, or the physical space too limited, then workarounds are required. Currently the most popular is Redirected Walking (shown below with physical space in red and the VE in blue) where users' paths in in the real world are subtly altered to make them feel as if they are exploring a larger space.



This technique is limited due to the disconnect between real vs perceived movement causing motion sickness in users (aka Cybersickness) [1,2].

Inspiration

In this 3-part system of a physical environment, virtual environment, and the user's locomotion, rather than aiming at the user's locomotion, like redirected walking, we decided to focus on the Virtual Environment itself.

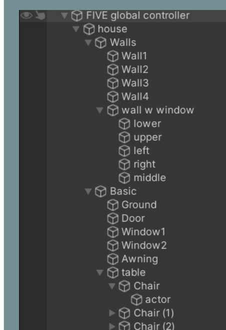
When looking at a webpage on the internet, the page will rearrange and resize elements on the page to match the user's screen size (shown in picture below). This is known as Responsive Web Design. For our approach, we decided to take inspiration from this technique when approaching our system.

Implementation

To develop our framework, we decided to use the Unity Game Engine to develop a tool for VR developers to utilize when designing environments.

The framework involves assigning behaviors to each object in the scene to dictate how they will respond to different physical space sizes. These behaviors can be passed down throughout the Hierarchy to streamline development.

The Hierarchy:



The Unity Hierarchy (shown on the left) displays all the objects inside the Virtual Environment. These objects can be nested inside each other (known as "parenting") to allow more general behaviors to be described and passed down through the hierarchy, setting more specific behaviors as you go down. This allows less time spent defining all parameters for each object.

In the example hierarchy on the left, the logic controller for the framework is at the top, and then below that you have increasingly specific categories.

Walls describes the general behavior for a wall, and nested inside of it are all the walls of the room, thus preventing developers from having to specify the same behavior 4+ times.

The Behaviors:

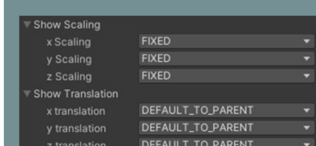
The behavior or an object inside the framework is determined by a set of parameters that the developer defines. These are largely split between Translation and Scaling parameters, both of which have similar options:

Relative – if an objects translation is set to relative it will move to always be in the same relative position inside the room. If it's scaling is relative it will stretch to take up the same relative space in the room.

Static – if an object's translation is set to static it will not move as the room changes size. Similarly, it will not change size if it's scaling is static.

Bounded – bounded will function the same as relative until an objects offset/scale reach a specified limit at which point it will begin behaving statically.

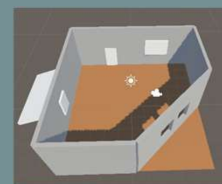
All of these behaviors are additionally specified separately for an object's x/y/z axes, so, for example, walls will shrink/grow along their major axis, but will not get extremely thick/thin as the room changes size.



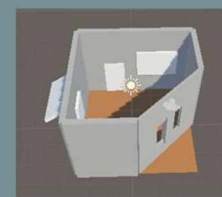
In the example on the left, these are parameters for a door. A door should always be the same size so it's scaling is set to fixed,.

Its parent has everything set to relative, so it's translation is left unchanged (the default value for all parameters is default_to_ parent).

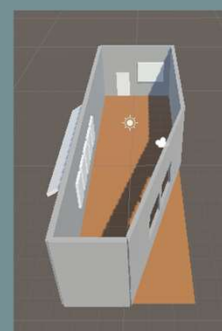
Results



This is the normal size of the room, there is a door, and then various windows



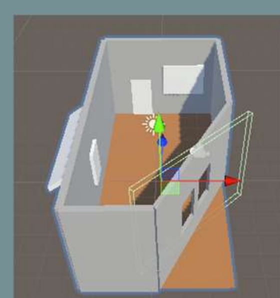
This is a shrunken version of the room, the windows and awning have gotten smaller, but the door is the same size



This room has one shrunken dimension, and one larger one. The angle of the diagonal wall has adjusted.

Additionally, the "windows" on the left wall have duplicated. This is what we named Tiling, where an object will duplicate to fill space, to prevent large empty walls. If this is enabled there will be additional parameters to set.

Limitations



Due to how Unity handles scaling and its physics engine, if an object is at an angle, then its colliders (used for physics) will become misaligned (shown below, where the green box is the diagonal wall's collider).

This means that objects at an angle will not be modeled properly by Unity's built-in physics engine. Causing, for example, objects or the player to pass through them, or collide with them in unusual places.

Future Work

Going forward, we hope to address the issues with angled objects, as well as implement the framework to have native support of the Oculus Quest Guardian (A user-defined physical space boundary for the Oculus Quest VR headset, shown below).

Additionally, we wish to merge this technique with existing redirected walking techniques to greater increase the possibilities of user locomotion inside Virtual Environments.

Conclusion

When looking at the current state of VR, if one wishes to walk around naturally, they either must have ample physical space, or use redirected walking, which has its limitations. We decided to pursue another alternative solution, adjust the Virtual Environment itself. By using Unity to develop the framework we were able to make a fully functional system, limited mainly due to quirks with Unity's scaling on angled objects.

References

[1] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe, "Estimation of Detection Thresholds for Redirected Walking Techniques," in IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 1, pp. 17-27, Jan.-Feb. 2010, doi: 10.1109/TVCG.2009.62.
[2] E. Langbehn, P. Lubos, G. Bruder, and F. Steinicke, "Bending the Curve: Sensitivity to Bending of Curved Paths and Application in Room-Scale VR," in IEEE Transactions on Visualization and Computer Graphics, vol. 23, no. 4, pp. 1389-1398, April 2017, doi: 10.1109/TVCG.2017.2657220.

Photos:
- Redirected Walking Diagram: https://www.semanticscholar.org/paper/Redirected-Walking-in-Virtual-Environments-Walker/6daceaebf0bc569621163ea15ff166c3f05b3c00/figure/0
- Responsive Web Design: https://medium.com/@madelinecorman/responsive-web-design-1748f1e6d781
- Oculus Guardian: https://www.ultimatepocket.com/oculus-quest-la-zone-guardian-maximum-passera-bientot-de-100-a-225m%C2%B2/
- All other photos were screenshots of the project itself on Unity