Manifold Racquetball

Horalia Armas David Dumas Brandon Reichman Hai Tran
University of Illinois at Chicago

Mathematical Computing Laboratory

UIC

Summary

The goal of this project was to create a virtual reality game that gives the user a first-person, interactive experience of geometries different from the Euclidean geometry we live in. We wanted the user to be able to compare these spaces with familiar real-world experiences, so we created a simple racquetball game to allow the user to interact with objects within multiply connected spaces, like the three-dimensional torus.

In order to make this game, we used the development environment and 3D graphics engine *Unity* [4] and the *Oculus* virtual reality system, including the *Rift* headset and real time hand position tracking *Touch* controllers. The game itself was written in the C# programming language.

The game was created in the Spring semester of 2017 as a semester project with faculty advisor David Dumas and graduate mentor Hai Tran in the Mathematical Computing Laboratory.

Motivation

Because the space we live in is Euclidean, it is difficult to imagine what it would be like to be inside a curved or multiply connected space such as hyperbolic space or a three dimensional torus.

With the use of virtual reality, we can create worlds that cannot normally be experienced. This allows a user to step out of our normal space and experience, for example, a three-dimensional torus. Virtual reality also allows the user to interact with simulated objects in such a space. This type of experience can be a useful tool for anyone studying curved or multiply connected geometries.

Tutorials

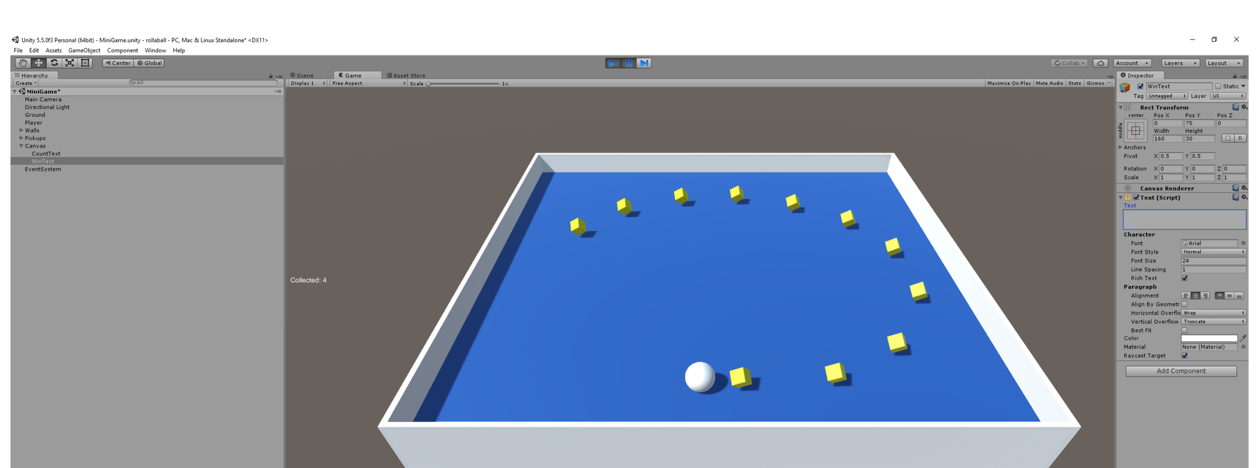
Since everyone on our project team was new to both the Unity engine and to virtual reality programming, we took several beginner steps to become familiar with the development toolkits.

Game Programming Concepts in Unity

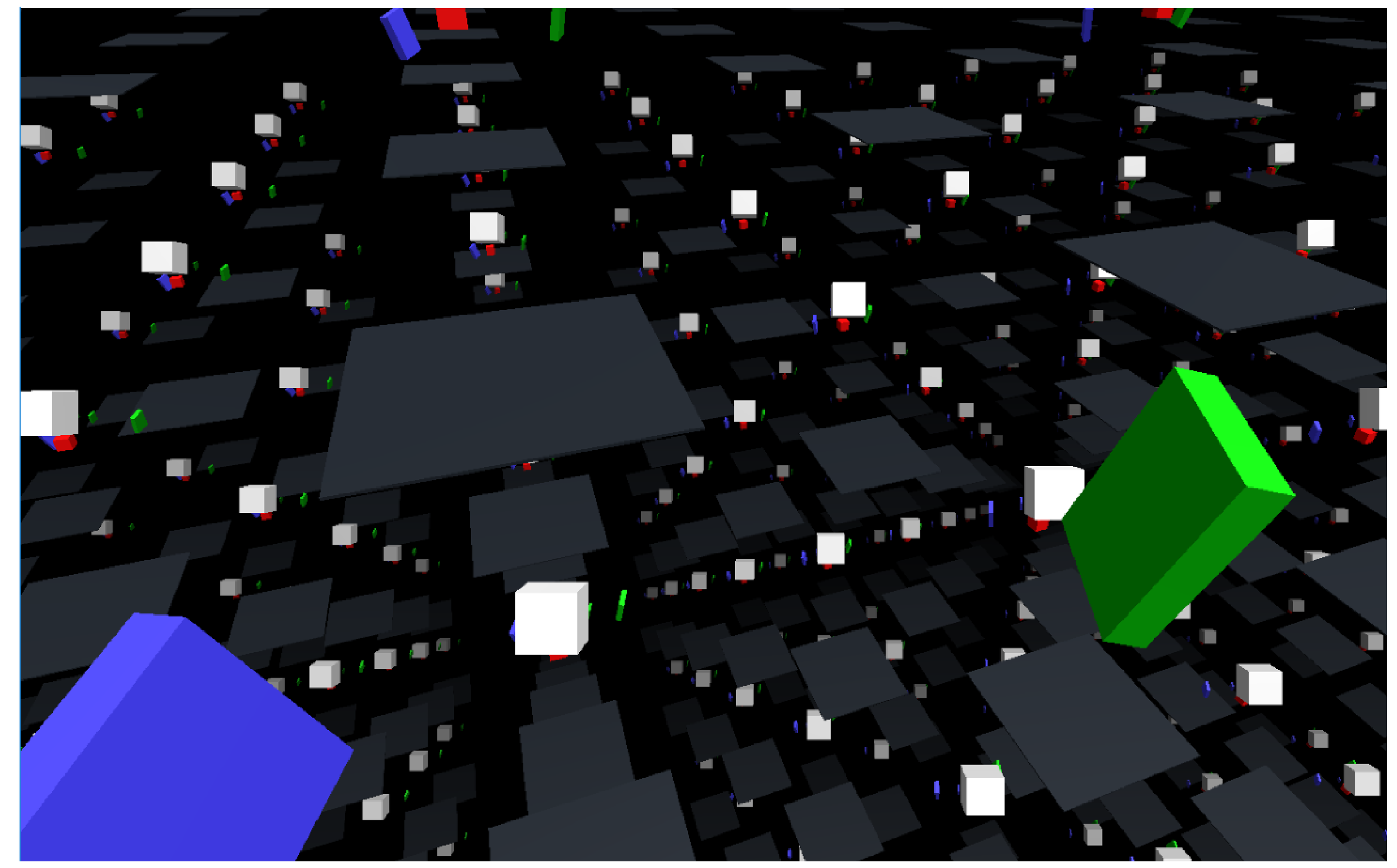
- Setting up a game
- Moving the player
- Moving the camera
- Detecting collisions
- Interacting with objects

Oculus Programming Concepts

- Hand position and rotational tracking
- Implementing grab and throw mechanics
- Integration with Unity 3D



A Simple Game in the Three-Dimensional Torus



The three-dimensional torus is a multiply connected geometric space where moving in any direction will eventually bring you back near your starting position. A person standing in the three-dimensional torus will see copies of themselves in every direction they look, seemingly arranged in a lattice or tiling. This is similar to a “hall of mirrors” effect without the reflections.

Playing a racquetball game in such a space allows for the user to hit the ball back and forth to themselves. The multiply-connected nature of the space gives the player many possibilities for where they can hit the ball.

Working Around Unity’s Limitations

In the first prototype of our virtual reality game, we used Unity’s built-in transformation systems to change object position and rotation, perform collision detection, and simulate Newtonian mechanics. Since Unity does not support non-Euclidean geometries and does not allow us to directly change its internal geometric transformations, we had to implement a method to manually control this aspect of the game.

We developed our own *shaders* which allowed us to position the objects by ourselves and implement our own version of physics interactions. In order to do that, we had to figure out the calculations behind every event within the environment. Each object is associated with a 4×4 matrix that determines its position, orientation, and scale. Modifying this matrix allows us to have complete control of our objects and sets up the framework for hyperbolic geometry.

Euclidean Geometry Matrix. For a translation t units along the x-axis, Unity uses the Euclidean isometry:

$$\begin{pmatrix} 1 & 0 & 0 & t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

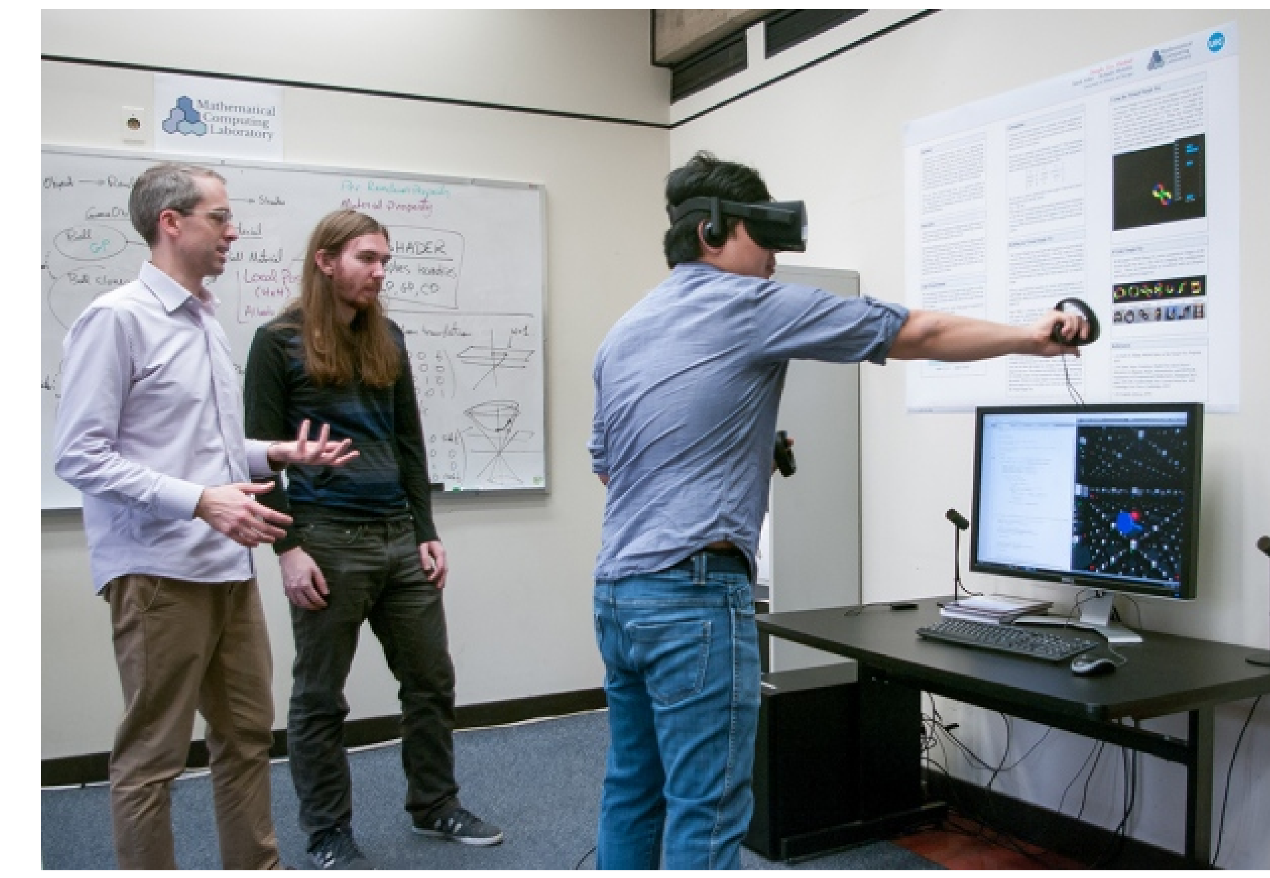
Hyperbolic Geometry Matrix. To perform a translation by t on the on the x-axis in hyperbolic geometry, we must instead use the isometry:

$$\begin{pmatrix} \cosh t & 0 & 0 & \sinh t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sinh t & 0 & 0 & \cosh t \end{pmatrix}$$

Replacing Euclidean transformations with their hyperbolic counterparts is possible in our manual object positioning system.

Honorable Mention

Our project was featured in Nature [1], a scientific journal from the United Kingdom. Below is a photo of our game in action that appeared in the article. The article talked about mathematicians using virtual reality to create “warped worlds,” which in our case refers to the geometry of the three-dimensional torus.



Related Projects

HypVR [2][3] is a webVR hyperbolic geometry project by Vi Hart, Andrea Hawksley, Elisabetta A. Matsumoto, and Henry Segerman that puts the user in a tiling of hyperbolic space. Instead of being able to interact with objects, HypVR gives the user the ability to continuously move through the space.

Jeff Weeks [5] has created many real time 3D graphics programs for exploring different spaces, including hyperbolic space and the torus. Part of the motivation for this project was to combine some of the visualization features of Weeks’ programs with the interactivity of virtual reality.

Future Objectives

Next we would like to make the game more user friendly by adding a short user tutorial and adding a menu for navigation or to change user preferences.

We would also like to add other locally Euclidean spaces besides the three-dimensional torus, for example a reflection orbifold.

While we did not implement a hyperbolic game this semester (as was originally planned), the framework we created will allow hyperbolic geometry to be used with the Unity game engine in the future.

References

[1] Davide Castelvecchi. *Mathematicians Create Warped Worlds in Virtual Reality*. ⟨http://www.nature.com/news/mathematicians-create-warped-worlds-in-virtual-reality-1.21689⟩ (2017), March 21

[2] Vi Hart, Andrea Hawksley, Elisabetta A. Matsumoto, and Henry Segerman. *Non-Euclidean Virtual Reality I: Explorations of* $\mathbb{H}^3$. ⟨https://arxiv.org/abs/1702.04004⟩ (2017), Feb. 13

[3] Vi Hart, Andrea Hawksley, Elisabetta A. Matsumoto, and Henry Segerman. *HypVR*. ⟨https://github.com/hawksley/hypVR⟩ (2017), Mar. 20

[4] *Unity (version 5.5)*.⟨https://unity3d.com⟩ (2016)

[5] Jeff Weeks. *Topology and Geometry Software*. ⟨http://www.geometrygames.org⟩ (2017)