

PICASSO VR

Ori Shem Tov

Eden Ben Oz

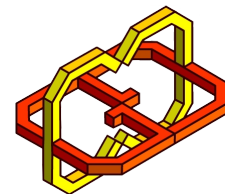
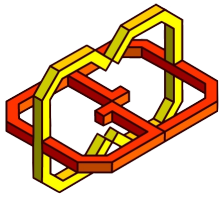


Table of Contents

Introduction	3
System & Technologies	4
Application Overview	5
Development Process	6-7
Proposed Next Steps	8
Screenshots & Video	9-11
Bibliography	
12	

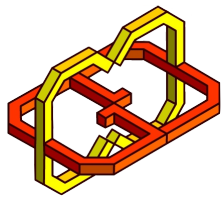


Introduction

We developed a VR environment for drawing in 3D.

The player can draw with both hands using controllers, and can edit their drawings by resizing, moving or deleting them.

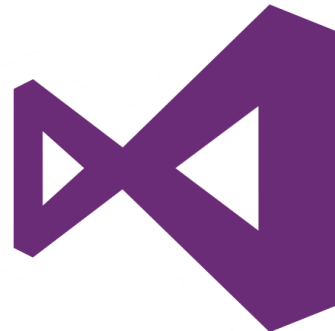
The player gets a unique drawing experience, such that they can draw “on air”, walk through their drawings, catch and move them, and so on.



System & Technologies

Our game runs on every standard computer (preferably a desktop), using HTC Vive VR equipment.

It was developed on Unity 2018.2.3f1 (64-bit) game engine, using VRTK features, and Visual Studio 2017 IDE.

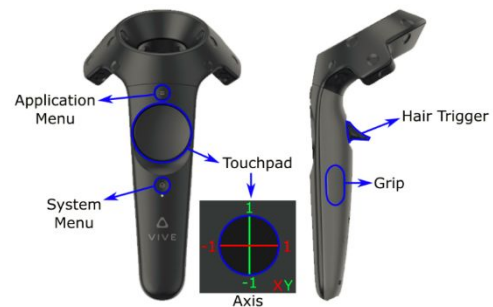


The required equipment is:

HTC Vive Headset



HTC Vive controllers

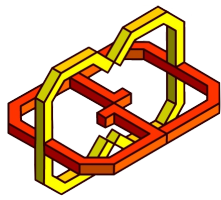


For more information please read:

<https://www.vive.com>

<https://www.unity.com>

<https://vrtoolkit.readme.io>



Application Overview

The game has 2 modes:

1. Start mode
2. Draw & edit mode

When running the game, the player enter the start mode, in which he can choose to start drawing or to read how to play.

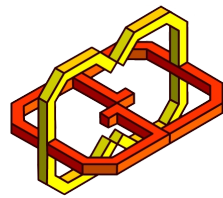
At any point of the game, the player can press on the 'Menu Button' and choose to either read again on how to play, or to quit (and some more options that will be elaborated later on).

After starting the game, the player can draw freely in both hands, using the 'Touchpad Button'. The color of the lines is chosen randomly, what makes the drawing colorful and pretty.

Whenever the player wants to edit their drawings, they press the 'Trigger Button' and initiate a laser ray, with whom they can select a drawing. On selection, the player can move the drawing (using a long press on the 'Trigger Button'), resize it (using the 'Touchpad Button') or delete it (by pressing the 'Menu Button' and choosing 'Delete').

While editing the drawings, the player cannot draw.

The player can clear all their drawings at any point by pressing the 'Menu Button' and choosing 'Clear All' (using the laser ray), and can quit the game in a similar way.



Development Process

The initial plan was to create a VR game in which the player can draw with their own hands, using Manus VR Gloves.

We worked on the integration between Unity, HTC Vive Headset, and Manus VR for a while - searched the internet and gathered information from colleagues. We reached a very nice result, where the player could have drawn with 2 fingers in each hand, but we've had many technical issues with the gloves, what made us switch to working with the HTC Vive Controllers.

At first we created a 'line' using multiple instances of a 'Sphere' game object. The result was impressive, but the performance was not good enough, so we needed to look for an alternative that would give better performance, with a similar look. After searching the internet, we found 'Line Renderer'.

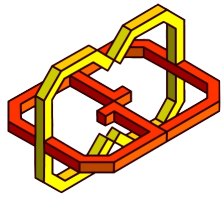
After improving the performance and finishing the work on the drawing itself, we started working on editing the drawings. This part was challenging since we implemented 3 different functionalities (move, resize and delete drawings).

Next was working on drawing selection. We investigated the sample scenes of VRTK to learn how to select objects using a laser ray. The integration between the 'Line Renderer' and the laser ray was not easy. We couldn't make the laser ray collide with the drawings, and it took some time to solve.

Last but not least is the UX.

We wanted the player to have an intuitive, easy and fun way to draw and edit, but we only had limited number of buttons to use, and so we needed to choose wisely.

At first, we separated draw mode from edit mode, in order to use the same buttons for different actions. We used all buttons on the

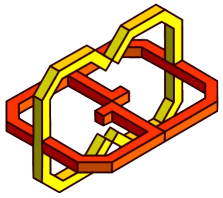


controllers, and most of them had double meaning, which made it difficult to understand and remember how to play.

After consulting, we decided to merge draw and edit mode, to achieve a more fluent experience. We did some research on UX for drawing games and applications, and adjusted our game accordingly.

In order to test our game (the whole experience) we let other students and colleagues play. We summarized their comments, prioritized it, and implemented small fixes accordingly.

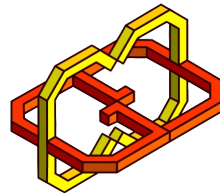
In conclusion, we had many ups and downs along the way, and even had a small pivot. We took notes, we were open for changes, and we did a lot of research, all helped us to solve our problems and reach better solutions.



Screenshots

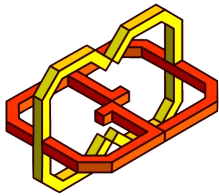
Start



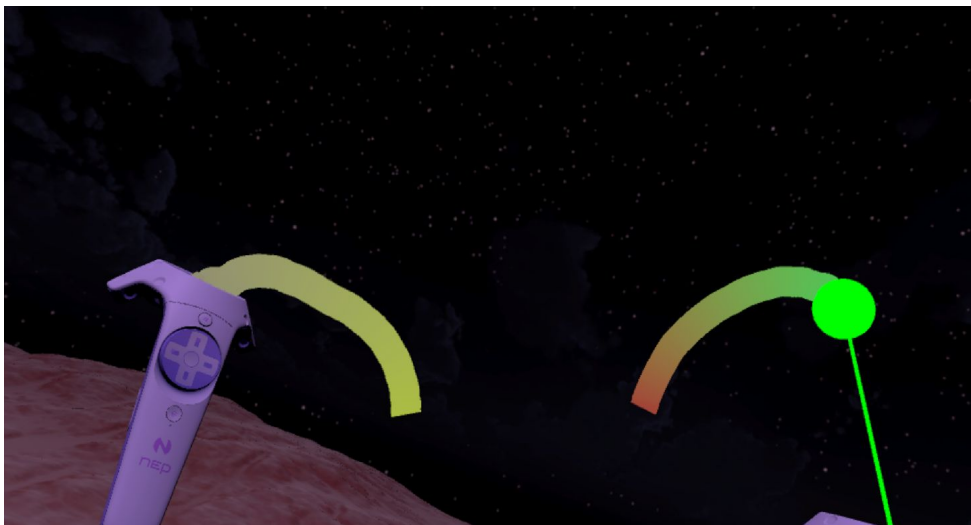
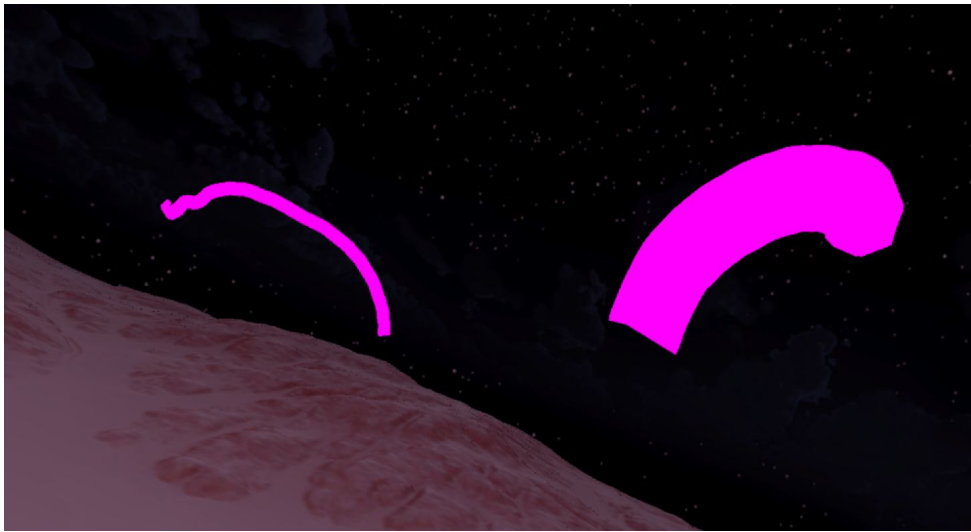


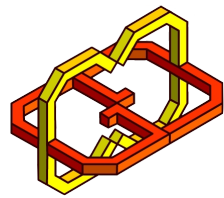
Draw



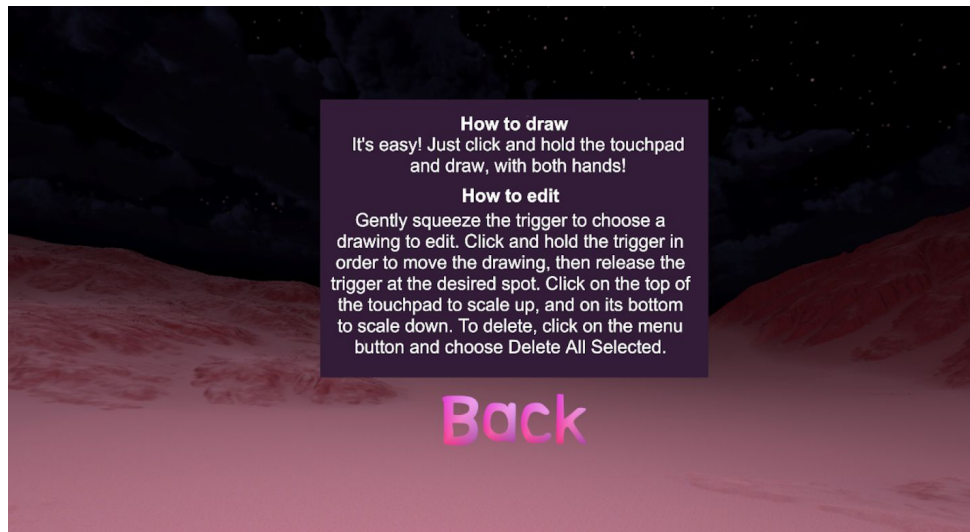


Edit



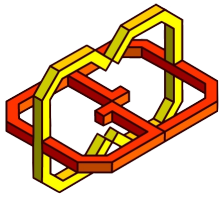


Menu



Demo video:

<https://youtu.be/eyESeW90vuk>



Bibliography

<https://docs.unity3d.com/Manual/class-LineRenderer.html>

https://vrtoolkit.readme.io/docs/vrtk_pointer

<https://docs.unity3d.com/Manual/class-MeshCollider.html>