

# Proposal - A Simulation of non-Euclidean Geometries in a 3D Environment

Jozef Carruthers

October 30, 2023

## Contents

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>Non-Euclidean geometry</b>	<b>3</b>
<b>4</b>	<b>Prior Approaches</b>	<b>3</b>
<b>5</b>	<b>Tools</b>	<b>4</b>
5.1	Unity (version 2022.3.11f1) . . . . .	4
5.2	C# . . . . .	4
5.3	Visual Studio Code . . . . .	4
5.4	GitHub . . . . .	4
<b>6</b>	<b>Goals</b>	<b>5</b>
6.1	Ideal goals . . . . .	5
6.1.1	An explorable 3D environment . . . . .	5
6.1.2	A good understanding of non-Euclidean space . . . . .	5
6.1.3	At least one non-Euclidean space . . . . .	5
6.2	Ambitious goals . . . . .	5
6.2.1	A deep understanding of non-Euclidean space . . . . .	5
6.2.2	An expansive non-Euclidean environment . . . . .	5
6.2.3	Project as a game . . . . .	5
<b>7</b>	<b>Milestones</b>	<b>6</b>

# 1 Abstract

This project aims at creating an explorative three dimensional environment featuring non-Euclidean geometries. The project will explore the realms of hyperbolic and elliptic/spherical geometry, challenging the conventional understanding of spacial interaction and giving intriguing visual experiences.

# 2 Introduction

The reason I am considering this as a project is that I find the idea of showing something that is inherently difficult to visualise interesting, as everthing about seeing the world and how things like physics work is based around Euclidean geometry. I also wrote a very simple pseudo 3D raycasting renderer for my A-Level CS project and found this very engaging.

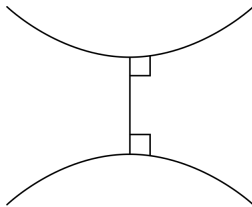
This document will discuss previous implementations of non-Euclidean geometry, giving links to examples that have explored the subject in a range of forms. The choice of tools I will need to use for the project is also detailed, including the use of Unity, C#, Visual Studio Code and GitHub.

I will explain the goals of my project, detailing my realistic and ambitious goals for the project, ranging from understanding non-Euclidean spaces to building a fully explorative environment, with various different implementations of non-Euclidean spaces and objects to interact with. The document will also include my milestones with estimated time frames for each step, guiding the project's development.

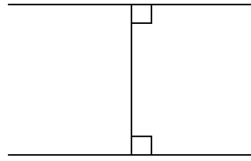
### 3 Non-Euclidean geometry

Non-Euclidean geometry is any geometry that does not follow the rules of Euclidean geometry. It often refers to hyperbolic and elliptic/spherical geometry. Euclidean geometry has zero plane/space curvature, whereas hyperbolic has negative curvature and spherical/elliptic geometry has positive curvature, other geometries may have varying curvatures.

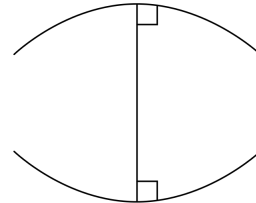
Imagine 3 lines, two parallel to each other and the third perpendicular to these two. In Euclidean geometry, the first two lines would stay at a constant perpendicular line distance from each other, however, in non-Euclidean geometry, this distance constantly changes. In hyperbolic geometry these lines will curve away from each other and in elliptic geometry the line will curve towards each other.



Hyperbolic



Euclidean



Elliptic

### 4 Prior Approaches

Looking into this area I have found many previous implementations of this idea often being a 3D rendered environment with areas of non-Euclidean geometry, including things like portals or tunnels that seem like the same length from outside but have different lengths on the inside using an area of stretched or compressed space. Below show a list of past approaches to this idea (clickable links):

- ["Non-Euclidean Worlds Engine" - CodeParade \[1\]](#)
- ["Non-Euclidean Geometry Explained - Hyperbolica Devlog #1" - CodeParade \[2\]](#)
- ["Spherical Geometry Is Stranger Than Hyperbolic - Hyperbolica Devlog #2" - CodeParade \[3\]](#)  
(the rest of the *Hyperbolica devlog series* [4] is a very interesting and deep look at implementing non-Euclidean geometry)
- ["Portals to Non-Euclidean Geometries" - ZenoRogue \[5\]](#)  
(other videos by *ZenoRogue* [6] are also very interesting)
- ["Hyperbolic maze" \[7\] - David Madore \[8\]](#)
- ["HyperRogue" - ZenoRogue \[9\]](#)

## 5 Tools

Below shows a list of tools I plan to use to create the project.

### 5.1 Unity (version 2022.3.11f1)

I plan on using Unity as I am familiar with using with this engine, I also believe it will be suitable for this project because Unity's rendering engine supports a variety of rendering techniques, including custom shaders and post-processing effects. Unity also allows for the writing of custom scripts using C# to implement complex mechanics. Unity's built-in physics engine is adaptable, meaning I can create realistic interactions with non-Euclidean space.

### 5.2 C#

C# is Unity's primary scripting language, and Unity has a specific scripting API to manage this. Along with these benefits, another reason to use C# would be my past experience, since I have been using C# for 7 years, and it is the programming language I am most familiar with. C# also has long term support, with a wealth of documentation and resources. C# is not only a well supported platform I am very familiar with, it also has great performance allowing me to make sure a computationally expensive task like rendering and simulating an environment runs smoothly.

### 5.3 Visual Studio Code

I have decided to use Visual Studio Code as my IDE for writing code in this project. It is a lightweight and powerful code editor that provides a good environment for C# development, making it a good choice for working with Unity projects. Visual studio code offers features like code completion, integrated version control and many extensions that allow a more enhanced development experience. Additionally, VS Code's user-friendly interface and customizable settings offer a suitable environment for coding, debugging and managing project files.

### 5.4 GitHub

In order to manage version control and project files across development on multiple devices, I have decided to use GitHub to create a centralized repository for my project files. GitHub provides a robust platform for version control, allowing me to track changes and make sure my project stays organized. By using this, I can take advantage of features like commit history and branching to create a more efficient development process.

## 6 Goals

By the end of this project I would like to achieve:

### 6.1 Ideal goals

#### 6.1.1 An explorable 3D environment

One of the primary objectives of the project is the creation of an immersive 3D environment. This environment should be engaging and navigable, but should offer a sense of unfamiliarity with the inclusion of non-Euclidean space. I aim to give an interactive experience, allowing users to explore the aspects of non-Euclidean geometry. The environment should serve as a stage to showcase the characteristics of non conventional spaces.

#### 6.1.2 A good understanding of non-Euclidean space

Another central goal of the project is to gain a comprehensive understanding of non-Euclidean geometries. This entails diving into the mathematical underpinnings, as well as building an intuitive grasp of how objects and spaces behave within non-Euclidean contexts.

#### 6.1.3 At least one non-Euclidean space

As another foundational accomplishment, the project aims to incorporate at least one region within the 3D environment where non-Euclidean geometry is explicitly demonstrated. It will be the main focus of the environment to serve as an interactive demonstration of the unfamiliarity of non-Euclidean space.

### 6.2 Ambitious goals

#### 6.2.1 A deep understanding of non-Euclidean space

One of my more ambitious goals of this project is to gain a deeper knowledge of non-Euclidean geometries. This involves exploring the aspects of different non-Euclidean spaces, such as hyperbolic and elliptic geometries, and comprehending their unique properties and implications.

#### 6.2.2 An expansive non-Euclidean environment

Building upon the basic explorative space mentioned previously, this ambitious goal involves the creation of an extensive 3D environment featuring multiple areas of varying space curvatures, presenting a captivating exploration of different geometries and their nuances.

#### 6.2.3 Project as a game

An overarching ambition is to transform the final project into a game-like experience, promoting exploration and understanding of the environments workings. This would help show how more trivial acts like solving a maze become much more interesting when done in non conventional spaces.

## 7 Milestones

### 1. Research and Experimentation:

Conduct further research into various different non-Euclidean geometries including how the maths works, and how these geometries may be visualised in three dimensions. Experiment with previous implementations of non-Euclidean geometries. Generally gain a deeper understanding of non-Euclidean geometry

### 2. Starting work on the environment basics:

Begin creating a basic 3D environment with a controllable observer, as mentioned previously I am leaning towards doing this in Unity as I have previous experience using it. The first things to do with this would be to create scripts for movement and viewing the environment for the observer.

### 3. Developing more of the environment:

Start developing the environment more, this could be adding more scripts to do with various aspects of a game if making this project as a game is not too ambitious, or simply adding obstacles and objects and making the environment more interesting looking.

### 4. Prototyping non-Euclidean space:

Begin prototyping non-Euclidean space in the environment, create various versions with different aspects of non-Euclidean geometry. Use these to assess the best way to implement non-Euclidean space.

### 5. Implementing non-Euclidean space:

Begin implementing non-Euclidean space/spaces with an understanding of how they will add to the environment and possibly affect how the game is played.

### 6. Testing, amending and evaluation:

Conduct self and peer user testing on the project, allow my peers to use the project, collect feedback from them and use this to finalise the project (this will also be done continuously throughout the project), finally evaluate the effectiveness of my project in meeting my goals.

## References

- [1] CodeParade. (2018). Non-Euclidean Worlds Engine. YouTube. Available at:  
<https://www.youtube.com/watch?v=kEB11PQ9Eo8> [Accessed 25 Oct. 2023].
- [2] CodeParade. (2020a). Non-Euclidean Geometry Explained - Hyperbolica Devlog #1 - YouTube. Available at:  
[https://www.youtube.com/watch?v=zQo\\_S3yNa2w](https://www.youtube.com/watch?v=zQo_S3yNa2w) [Accessed 25 Oct. 2023].
- [3] CodeParade. (2020b). Spherical Geometry Is Stranger Than Hyperbolic - Hyperbolica Devlog #2 - YouTube. Available at:  
<https://www.youtube.com/watch?v=yY9GAYJtuJ0> [Accessed 25 Oct. 2023].
- [4] CodeParade. (2022). Hyperbolica - YouTube. Available at:  
<https://www.youtube.com/playlist?list=PLh9DXIT3m6N4qJK9GKQB3yk61tVe6qJvA> [Accessed 26 Oct. 2023].
- [5] ZenoRogue. (2022). Portals to Non-Euclidean Geometries. Youtube. Available at:  
<https://www.youtube.com/watch?v=yqUv2J02BCs> [Accessed 25 Oct. 2023]
- [6] ZenoRogue. (2023). ZenoRogue YouTube Channel. YouTube. Available at:  
<https://www.youtube.com/@ZenoRogue> [Accessed 25 Oct. 2023].
- [7] Madore, D. (2016). Hyperbolic maze. Available at:  
<http://www.madore.org/~david/math/hyperbolic-maze.html#explanations> [Accessed 26 Oct. 2023].
- [8] Madore, D. (2023). David Madore. Available at:  
<http://www.madore.org/~david/> [Accessed 26 Oct. 2023].
- [9] ZenoRogue. (2023). HyperRogue - About. Available at:  
<https://www.roguetemple.com/z/hyper/> [Accessed 26 Oct. 2023].