An investigation into the practical uses of spherical non-Euclidean geometry in games

Is spherical non-Euclidean geometry in the Unity game engine practical?

January 30, 2022

Iain Farlow, s1903434

University of Gloucestershire

Contents

[1 Introduction 2](#_Toc96700657)

[1.1 Justification 2](#_Toc96700658)

[1.2 Aims 3](#_Toc96700659)

[1.3 Objectives 3](#_Toc96700660)

[2 Literature Review 3](#_Toc96700661)

[2.1 Overview 3](#_Toc96700662)

[2.2 Euclidean Geometry 3](#_Toc96700663)

[2.3 Non-Euclidean Geometry 3](#_Toc96700664)

[2.3.1 Examples of Non-Euclidean Games 3](#_Toc96700665)

[2.3.2 Hyperbolic Non-Euclidean Geometry 4](#_Toc96700666)

[2.3.3 Spherical Non-Euclidean Geometry 4](#_Toc96700667)

[2.4 What can Spherical Non-Euclidean Geometry bring to the Games Industry 4](#_Toc96700668)

[2.5 Summary 4](#_Toc96700669)

[3 Output Design 4](#_Toc96700670)

[3.1 Overview 4](#_Toc96700671)

[3.2 Possible Issues 4](#_Toc96700672)

[3.2.1 Level Creation 4](#_Toc96700673)

[3.2.2 Rendering 5](#_Toc96700674)

[4 References 5](#_Toc96700675)

Table of Figures

[Figure 1: The Difference Between Euclidean and Non-Euclidean Geometry (Kurniadi, 2011). 2](file:///C:\Users\iainf\OneDrive\Desktop\Uni%20Stuff\Year%203\CT6007%20-%20Indevidual%20Research%20Project\An%20investigation%20into%20the%20practical%20use%20of%20spherical%20non-Euclidean%20geometry%20in%20games.docx#_Toc94469190)

# Introduction

## Justification

The problem that is being investigated is that of ‘Can non-Euclidean geometry be practical in the creation of a 3D game?’. Euclidean geometry is defined by a collection of rules as set out in “The Elements” (Euclid, 300 bce) and has formed the basis of geometry in all modern society, see *figure 1 ‘zero curvature’*. As a result of Euclidean geometry being a key aspect of society, it has an influence on video games as well, “Since the conception of the first video game, its visual representation has always been done mainly in the fashion of the Euclidean geometry.” (Guimaraes, Mello & Velho, 2015).

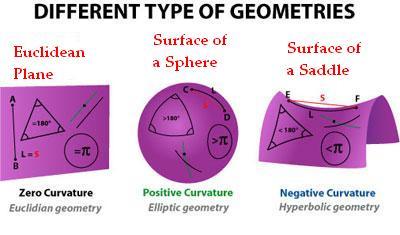


Figure 1: The Difference Between Euclidean and Non-Euclidean Geometry (Kurniadi, 2011).

Wolf (2018) states that “video games can easily allow for non-Euclidean configurations”. However, Taylor (2003) also says “In keeping with these cultural prejudices, descriptions of the optical space of video games presume an uncomplicated optical scheme, founded on traditions of linear perspective.” This suggests that, despite non-Euclidean geometry being, theoretically, easy to implement, a complicated form of non-Euclidean geometry needs to be done to a high quality to be successful. Currently, to make games intuitive and familiar to players the industry heavily relies on the conventions set out in film and television (Wolf, 2004) this means that games designers have neglected the possibilities that abstract design methods provide. This has the potential to be one of the key aspects of why there is a lack of games that utilise non-Euclidean geometry.

The output design will involve, but not be limited to, an environment containing a procedurally generated maze that is created in a spherically non-Euclidean world. In this a basic character controller will be present, allowing for the movement within the maze. However, there is no intent to have any gameplay elements present, such as UI or a formal gameplay loop. However, this is anticipated to change as development progresses.

## Aims

This paper aims to investigate the uses, and potential uses, of spherical non-Euclidean geometry, research why non-Euclidean geometry is typically not used in video games, and try to establish a viable method for the implementation of a spherically non-Euclid geometry in the Unity game engine (Unity Technologies, 2020).

## Objectives

* Conduct a review of academic literature regarding non-Euclidean geometry.
* Research and evaluate the uses, and potential uses, of spherical non-Euclidean geometry in the industry.
* Determine a system for simulating spherical non-Euclidean geometry.
* Create a program in the Unity game engine (Unity Technologies) using said system.

# Literature Review

## Overview

This literature review will investigate how computers can simulate Non-Euclidean Geometry. With this, the methods of simulating Non-Euclidean Geometry will be compared whilst also analysing their practical uses within video games. This will then be used to justify the use of Spherical Non-Euclidean Geometry as a viable form of Geometry in games.

## Euclidean Geometry

Euclidean Geometry is the typical form of geometry most people are used to. In this form of geometry, the world around us is mapped onto a plane with the axis mapped at 90 angles to each other.

## Non-Euclidean Geometry

### Examples of Non-Euclidean Games

### Hyperbolic Non-Euclidean Geometry

### Spherical Non-Euclidean Geometry

## What can Spherical Non-Euclidean Geometry bring to the Games Industry

## Summary

# Output Design

## Overview

Using the information gathered in the literature review of this paper a game will be created that utilises spherical non-Euclidean geometry. This geometry will contain a simple randomly generated maze that will allow the user to experience the spherical non-Euclidean Geometry in an interesting way.

## Possible Issues

### Level Creation

As a result of the Spherical Non-Euclidean nature of the game the issue will arise regarding the creation of levels may arise. At this current point the idea behind the game is that the levels will be procedurally generated, to ensure the mazes are different every time the game is played. However, due to the positive curvature found within Spherical Non-Euclidean Geometry simply designing the game in Euclidean Space and then rendering it Spherical Space issues may arise. If this is the case, then it is likely that a tile map like system will have to be created.

If a tile made is to be used, the common method of simply using x, y and z coordinates is not practical. This because the spherical plane in geometry will have to have a different number of, in this case, squares on each vertex to account for its curvature, three for spherical when flat would have four. This will in turn result in a custom coordinate system being created that will simply start at the origin and use up, down left and right to determine specific points on the geometry. If this is the case then it is likely that simple vectors will not be practical to use, due the holonomy that this form of geometry causes. If this is the case, a simple fix can be created by introducing gyrovectors to the game, which will fix these issues, primary since their translations are non-commutative.

### Rendering

As a result of the Spherical Non-Euclidean geometry in the game the method in which the game is projected will have to be changed. Since the game will use Spherical geometry, it is likely that Geometric projection will be used. The issue with this form of projection lies in rendering both hemispheres, this is because one render pass can only render one half of the geometry. If two render passes were used, one for each half of the geometry, the nature of Geometric projection will cause the line where these two render passes meet to not render, as both passes can only render up to one half of the geometry. If two vertices of each triangle are in different hemispheres to each other then the whole vertex will be discarded, resulting in nothing being drawn.

As a result of the issues mentioned Geometric projection cannot be used in this case. This resulted in the conclusion that Stereographic projection must be used as it renders the whole geometry in one pass. However, it is likely that depth issues will arise if this method is used, with these issues becoming worse the further the geometry is from the camera. To fix this, two render passes can be used, both stereographic, one rendering the closer geometry and one rendering the further.

There is still one issue that may arise when rendering the geometry for the game. This is the case due to the stereographic projection that will be used. If this form of projection does have to be utilised then the depth buffer will have to be altered to compensate for issues in the far hemisphere, as well as the joining of the two projections. To do this the two projections will be in one depth buffer, the near occupying 0 – 0.5 (mapped 0 to 0 and 1 to 0.5), whilst the far occupies 0.5 to 1 (mapped 0 to 1 and 1 to 0.5 to account for depth issues).

# References

Euclid. (300 bce) ‘The Elements’, Available At: <https://www.google.co.uk/books/edition/The_Elements_of_Euclid/5lN1sy51SwYC?hl=en&gbpv=1&printsec=frontcover>

Guimaraes, F., D., Mello V., M., & Velho, L. (2015) ‘Geometry independent game encapsulation for non-Euclidean geometries’, [Online], Available at: <https://www.semanticscholar.org/paper/Geometry-independent-game-encapsulation-for-Guimar%C3%A3es-Mello/1a14f10aa957acfbf28f31eeb69657a976cce92c#references>

Taylor, L. (2003) ‘When Seams Fall Apart, Video Games Space and the Player’, Game Studies, vol. 3, issue. 2, [Online], Available At: <http://www.gamestudies.org/0302/taylor/> (Accessed: 25 January 2022)

Wolf, M., J., P. (2004) ‘Abstraction in the Video Game’, in Wolf, M., J., P. & Bernard Perron (eds.) The Video Game Theory Reader, 1st edn. Routledge, pp. 47 – 53

Unity Technologies, (21 October), Unity3D (Version 2019.4.13f1) [Software], Available At: <https://unity.com/> (Accessed: 30 January 2022)

Wolf, M., J., P. (2018) ‘Theorizing navigable space in video games’ Potsdam University, issue. 6, pp. 18 – 49, [Online], Available At: <https://publishup.uni-potsdam.de/frontdoor/index/index/docId/5043> (Accessed: 25 January 2022)