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](#_Toc98168360)

[1.1 Justification 2](#_Toc98168361)

[1.2 Aims 3](#_Toc98168362)

[1.3 Objectives 3](#_Toc98168363)

[2 Literature Review 3](#_Toc98168364)

[2.1 Overview 3](#_Toc98168365)

[2.2 Euclidean Geometry 4](#_Toc98168366)

[2.3 Non-Euclidean Geometry 5](#_Toc98168367)

[2.3.1 What is Non-Euclidean Geometry 5](#_Toc98168368)

[2.3.2 Hyperbolic Non-Euclidean Geometry 6](#_Toc98168369)

[2.3.3 Spherical Non-Euclidean Geometry 7](#_Toc98168370)

[2.3.4 Rendering Non-Euclidean Geometry 8](#_Toc98168371)

[2.3.5 Examples of Non-Euclidean Games 11](#_Toc98168372)

[2.4 Summary 13](#_Toc98168373)

[3 Output Design 15](#_Toc98168374)

[3.1 Overview 15](#_Toc98168375)

[3.2 Possible Issues 15](#_Toc98168376)

[3.2.1 Level Creation 15](#_Toc98168377)

[3.2.2 Rendering 16](#_Toc98168378)

[4 References 17](#_Toc98168379)

[5 Glossary of Terms 19](#_Toc98168380)

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\CT6007---IRP\An%20investigation%20into%20the%20practical%20use%20of%20spherical%20non-Euclidean%20geometry%20in%20games.docx#_Toc98167848)

[Figure 2: Display to show Euclid's fifth postulates relation to triangles (Ravindran, 2007). 4](file:///C:\Users\iainf\OneDrive\Desktop\Uni%20Stuff\Year%203\CT6007%20-%20Indevidual%20Research%20Project\CT6007---IRP\An%20investigation%20into%20the%20practical%20use%20of%20spherical%20non-Euclidean%20geometry%20in%20games.docx#_Toc98167849)

[Figure 3: Poincaré Projection (Kallosh & Linde 2015). 8](file:///C:\Users\iainf\OneDrive\Desktop\Uni%20Stuff\Year%203\CT6007%20-%20Indevidual%20Research%20Project\CT6007---IRP\An%20investigation%20into%20the%20practical%20use%20of%20spherical%20non-Euclidean%20geometry%20in%20games.docx#_Toc98167850)

[Figure 4: Stereographic (left) and Gnomonic (Right) Projection (Coxeter, 1989). 9](file:///C:\Users\iainf\OneDrive\Desktop\Uni%20Stuff\Year%203\CT6007%20-%20Indevidual%20Research%20Project\CT6007---IRP\An%20investigation%20into%20the%20practical%20use%20of%20spherical%20non-Euclidean%20geometry%20in%20games.docx#_Toc98167851)

[Figure 5: Poincaré Disk (left) and Beltrami-Klein(right) projections of the Hyperbolic Maze (Madore, 2013). 12](file:///C:\Users\iainf\OneDrive\Desktop\Uni%20Stuff\Year%203\CT6007%20-%20Indevidual%20Research%20Project\CT6007---IRP\An%20investigation%20into%20the%20practical%20use%20of%20spherical%20non-Euclidean%20geometry%20in%20games.docx#_Toc98167852)

[Figure 6: Octahedral holonomy maze (Segerman, N.D). 12](file:///C:\Users\iainf\OneDrive\Desktop\Uni%20Stuff\Year%203\CT6007%20-%20Indevidual%20Research%20Project\CT6007---IRP\An%20investigation%20into%20the%20practical%20use%20of%20spherical%20non-Euclidean%20geometry%20in%20games.docx#_Toc98167853)

# Introduction

## Justification

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 game designers may 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 if it would benefit the demonstration of a spherically non-Euclidean geometry. Due to time considerations on the other hand, the main focus in the output Design will be that of the spherically non-Euclidean space meaning these further elements are not likely to be included.

## 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 implement 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. The world around us is mapped onto a plane with no curvature. With this “Each ordered pair (p1, p2) of real numbers determines exactly one point P of the plane. The point determined by (0, 0) is called the origin” (Ryan, 1986). This is the basic coordinate system that we interpret our flat geometry in, with a third real number being used to allow us to interpret in a third dimension. This allows for people to interpret the transform of something with relative ease due to the 2 / 3 axis’ being easily interpreted. This is especially the case when computers are involved, as the use of transformation matrices allow for the storing of data that not only stores the position of an object, but also its rotation and scale.

Another way in which people the layperson will find Euclidean geometry the least confusing simply revolves around the formation of a line. Defined by Euclid “A line which lies evenly between its extreme points is a straight or right line” (Casey, 2007). Two points have an obvious shortest distance. This definition, however, does not only apply to Euclidean geometry, but can also be applied in non-Euclidean (curved) geometry. Despite this, the definition is most easily understood in Euclidean space, due to the lack of curvature. This, in turn, makes rendering environments using Euclidean geometry far simpler than other geometries.

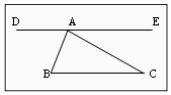
Another, and possibly one of the most important, aspect of Euclidean geometry is Euclid’s Fifth Postulate. This states “If a straight line falling on two straight lines make the alternate angles equal to one another, the lines will be parallel to one another” (Wolfe, 2012). This is a key aspect of Euclidean geometry as it forms the basis of many concepts in mathematics. One of these is that “The sum of the angles of a triangle equals two right angles” (Ravindran, 2007), see *figure 2.* Due to this being a key, base principal of geometry the requirement for a consistent idea such as all triangles having an interior angle of 180° is greatly important, and when considered in a flat Euclid plane can help solve other geometric calculations. This is then shown to be of great importance when it comes to rendering in a virtual environment. Typically, vertices will be used to make triangle that make up the rendered model. In a flat, Euclid space this will not be a problem due to the linear nature of the lines joining the vertices. Issues, however, begin to arise when rendering in non-Euclidean geometry due to the non-linear nature the curvature of these geometries introduces. This makes Euclidean geometry ideal for rendering as the nature flat nature of the screen and the linear nature of connecting vertices is not impacted.

Figure 2: Display to show Euclid's fifth postulates relation to triangles (Ravindran, 2007).

## Non-Euclidean Geometry

### What is Non-Euclidean Geometry

Non-Euclidean geometry exists in two main forms Hyperbolic and Spherical (Manning, 1963). The primary difference between two geometries is curvature. Euclidean geometry has a zero curvature nature people are accustomed to. However, Hyperbolic geometry is a geometry with negative curvature, whilst spherical is a geometry with positive curvature. This means that nothing is stopping the use of any non-Euclidean geometry in a virtual setting, Wolf (2018). Instead, the main thing stopping the use of these geometries in games, especially in a 3D environment, is whether the user can interpret the space. This is because interpreting the curvature on a 2D plane from the perspective of a 3D being is far different from interpreting from a 2D perspective, meaning an understanding of the geometry must be present to conceive a 3D curved geometry.

### Hyperbolic Non-Euclidean Geometry

Hyperbolic non-Euclidean geometry is one of the major forms of non-Euclidean Geometry found within the games industry. This is because it can have an interesting effect on how games are played. Especially when we look at 2D tiled games. “[It] would not be very interesting in a Euclidean tiling, but hyperbolic geometry makes it fun—not only because it feels very different, but also because it is possible to use the hyperbolic properties tactically, to deal with large groups of enemies” (Kopczynski et al, 2017). This can be credited to the larger areas the game plays out in, due to the curvature found within Hyperbolic geometry. The use of Hyperbolic, therefore, can show an instant use within the games industry. However, due to the increased space within the geometry when interpreted in a 3D format this form of geometry can result in the space seeming like an exaggerated Euclidean geometry. The similarity to non-Euclidean space is simply due to the curvature Hyperbolic geometry forms and the typical use of render distance within games, resulting primarily in the appearance of more space fitting into a smaller space. Despite this being a possible negative of Hyperbolic geometry, it does result in a positive as the less evident change in geometry means that users unfamiliar with non-Euclidean geometry will have a far easier time getting used to the new geometry.

Hyperbolic geometry has other aspects that deviate from Euclidean Geometry, one of these is that triangles will have a total of interior angles less than 180°, “Triangles in hyperbolic space have angle sum less than π; in fact, the area of a triangle with angles α, β, and γ is π − α − β – γ” (James et al, 1997). However, as stated by Euclid’s fifth postulate triangles should have a total interior angle of 180°, this confliction is one of the reasons Hyperbolic geometry is classified as non-Euclidean. This confliction will have a grave impact on how games are rendered using non-Euclidean geometry, as previously stated is due to the non-linear way in which vertices and their connections will be interpreted. This will mean that the type of projection used when displaying Hyperbolic geometry is extremely important as it must preserve, or counteract the non-linearity created by the curvature.

As a result of the triangles in Hyperbolic geometry having less than 180° as their total interior angle, Euclid’s fifth postulate does not always apply in Hyperbolic space. However, in Hyperbolic geometry lines can still be parallel. This is due to the different ways in which lines are considered. This is supported by the fact that a parallel ruler can be utilised in Hyperbolic space. “A parallel-ruler is, as in Euclidean geometry, an instrument for drawing a line through a given point and parallel to a given line. We shall also, as in Euclidean geometry, use the parallel-ruler as an ordinary ruler. THEOREM 1. Any construction in hyperbolic geometry that can be performed by means of a ruler and any of the three compasses, can be performed by means of a parallel-ruler.” (Handest, 1956). This shows that the fact that parallel lines appear to diverge in a Hyperbolic space is purely a result of perspective and not geometry. This means that Hyperbolic geometry can be very applicable in games, as forming simple maps that utilise straight lines is still possible.

### Spherical Non-Euclidean Geometry

One way that spherical geometry deviates from Euclidean geometry is that lines in Spherical geometry cannot be parallel. This is because any straight lines (using Euclid’s definition of a straight line) made in Spherical geometry will always intersect. This is attributed to them creating a greater circle. A greater circle is called this as it exactly bisects the sphere it is on (Kern & Bland, 1948). If we take a globe with lines marked on the longitude and latitude it is apparent that all the latitude lines intersect at the poles, also forming a greater circle. If we look at the longitude lines it looks like these lines are also straight but do not intersect, nor do they bisect the sphere. The issue with this is that the lines are not straight and are instead curved. As a result of this, they cannot be parallel (Whittlesey, 2019). This means that Euclid’s fifth postulate cannot apply in a Spherical geometry, making it non-Euclidean. Because all straight lines intersect Spherical geometry can have some issues when used in a virtual format. These develop since any straight environments such as hallways will converge and eventually meet, meaning that designing levels could present issues.

It is not just Euclidean geometry that can have practical applications in the real world. The opposite of this is the case as “we still cannot decide whether the real world is approximately Euclidean or approximately non-Euclidean” (Coxeter, 2019). A simple example of this is simply the curvature of the Earth, “Here is a very simple example: Consider the geographical triangle Amsterdam (latitude 528 210 north, longitude 48 540 east), Boston (428 200 north, 718 50 west), and Cape Town (358 560 south, 188 220 east). This triangle has an area of 38 700 000 km2. 70 J J Koenderink, A J van Doorn, J S Lappin From Amsterdam, Boston is 5568 km at 208 300 north of west, Cape Town is 9914 km at 108 520 east of south, etc. The angle sum of the triangle is 2348 28.30: This differs appreciably from the Euclidean 1808; thus the Earth is significantly curved.” (Koenderink et al, 1999). This presents a use for Spherical geometry in the real world, as the the area of the surface of the earth can only be truthfully represented in Spherical geometry. Much like Hyperbolic geometry, issues that may arise from this come from the primitive shapes used in rendering. When the geometry of models is designed in a Euclidean space the conversion from Euclidean to non-Euclidean can create depth issues due to the irregularity with lines the curvature creates.

A picture containing text, athletic game, sport, basketball

Description automatically generated

### Rendering Non-Euclidean Geometry

One of the primary methods of rendering a Hyperbolic geometry on a flat, Euclid, plane is by utilising the Poincaré projection. In this method, a line on the Hyperbolic plane is seen from point 0 on the flat plane. From this, the points on the Hyperbolic plane are then mapped to a flat plane forming a curve from what would have been a straight (Hyperbolic) line, see *figure 3*. However, this form of projection can lead to depth issues when resolving the rendered geometry in the fragment shader. This is because the curved line projected onto the Euclid plane will be interpreted linearly by the fragment shader. However, these lines should be interpreted non-linearly causing the depth of rendered models may appear incorrectly due to the depth of some rendered vertexes being interpolated be behind or in front of where they should be. This can create obvious graphical. One way to resolve this would be to subdivide the mesh of any models used within the game, reducing the obviousness of the graphical issue. Doing this can have a performance effect on the game, especially if done on a large scale, meaning that this form of rendering has the potential to be expensive when considering the resources that the game can access.

Figure 3: Poincaré Projection (Kallosh & Linde 2015).

However, the Beltrami Klein disk model can be used to prevent this depth issue as it is designed to preserve the distance between points. In this method, a similar system is used but instead of mapping the points as a curve, the points are used to construct a plane that is then used to create a chord in a flat plane (Francis, 2002). These points have then retained the scale of their distances and formed a straight line instead of a curved one. This retaining of distance prevents the depth issues mentioned previously, this does mean creating distortions in angles and circles. Despite this, the Beltrami Klein disk model is still a more ideal system for rendering Hyperbolic geometry since it overcomes the issues regarding the depth of vertices.

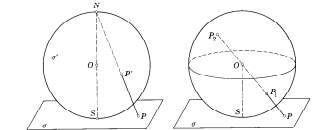
When rendering Spherical geometry, it is important to utilise the correct form of projection. One such form of projection is that of Gnomonic projection. In this form of projection, the centre of the sphere our geometry exists on is taken as the origin. From here a point on the flat plane we are projecting to is taken as the resulting point. A line is then created between the origin and the resulting point. The position this line then intersects on the sphere is the point in the geometry that will be rendered at the resulting point on the flat plane (Coxeter, 1989), see *figure 4 ‘Gnomonic Projection’*. There is an issue with this form of projection, it cannot render the equator of the sphere. This is due to the resulting line infinitely approaching the equator. If both hemispheres of the sphere are rendered a line will appear on the flat plane where nothing is rendered (what would be the equator of the sphere). This is due to the vertices of the rendered primitive being in different hemisphere and as each hemisphere is rendered separately the primitive shape with the missing vertices must be discarded. On the other hand, Gnomonic projecting does ensure that the interpolation is non-linear, ensuring the depth of vertices is correct in proportion to others. This means that in some cases this can be a very effective way to render Spherical geometry as it helps prevent any depth issues. The primary issue is that both hemispheres of the sphere are required to be rendered separately, this could be overcome by simply introducing a render distance that prevents the player from seeing further than the equator.

Figure 4: Stereographic (left) and Gnomonic (Right) Projection (Coxeter, 1989).

To project a Spherical geometry fully in one render pass, Stereographic projection can be used. This form of projection is very similar to that of Gnomonic projection. However, with Stereographic projection the origin point is not the centre of the sphere, instead, it is the point furthest away from the plane. Allowing for the mapping of all points, except the origin onto a flat plane from the Spherical geometry (Coxeter, 1989), see *figure 4 ‘Stereographic Projection’*. Utilising this form of projection can cause similar depth issues as within Poincare Projection, this will only begin to become apparent in the hemisphere closest to the origin. Therefore, if two render passes are used, one for the near hemisphere and one for the far, the two render passes can be combined to allow for the rendering of all geometry. This means that this is the most practical form of projection when it comes to Spherical geometry, as not only does it allow for both hemispheres to be rendered seamlessly but also helps to negate the depth issue found in most forms of rendering for non-Euclidean geometry.

### Examples of Non-Euclidean Games

Hyperbolica (CodeParade, 2022) is a game that utilises both Spherical and Hyperbolic geometry. This is an example of how games can work in a non-Euclidean format. Showing, that whilst the standards of the industry are to present games in Euclidean geometry, it is not only possible to make a game in non-Euclidean geometry but possible to have positive feedback from the community surrounding the game. Despite Hyperbolica’s small development team, the game shows gameplay in various situations such as shooting, flying and quests, proving that non-Euclidean geometry is practical in various situations. Due to the multiple forms of gameplay Hyperbolica demonstrates it evident that a non-Euclidean game can be created for most genres of game. Due to non-Euclidean geometry being applicable across all areas of the games industry it is evident that, despite possibly being a niche, as a large amount of untapped potential in the games market.

A highly successful example of a non-Euclidean game series is the Portal Game series. The Portal games offer the non-Euclidean experience using portals. These can be placed on surfaces and when passed through move the player to the other portal. This means that whilst the world is Euclid the use of portals means the game cannot be interpreted in a truly Euclid world. This is since in games such as Portal (Valve, 2007) or Portal 2 (Valve, 2011), the use of a portal can make distances and angles inconsistent, for example, if a point is twenty metres away from the player and they place a portal at that point, as well as one at one metre away from them, then they are both twenty metres and one metre away from the point. With these inconsistencies, it is believable that the non-Euclidean nature of the game could have caused audiences to find the game confusing, however, Portal (Valve, 2007) holds extremely good reviews. Metacritic has awarded the game a 90 / 100 based on 27 Critic reviews, as well as a 9.2 / 10 based on 5930 user reviews. This demonstrates that games using a non-Euclidean format can succeed in the games industry.

Another game that Utilises non-Euclidean geometry is the simple web game Hyperbolic Maze. This game shows how the use of different projections can change the resulting appearance of the game. In this game, the user can explore a 2D, hyperbolic maze, and choose between two different projections, Poincaré Disk and Beltrami-Klein, see *figure 5.* Each projection type represents the space differently on the screen. This demonstrates the importance of using the correct form of projection for the scenario, with some forms of projection being better for Spherical non-Euclidean geometry and others being better for Hyperbolic non-Euclidean geometry.

Figure 5: Poincaré Disk (left) and Beltrami-Klein(right) projections of the Hyperbolic Maze (Madore, 2013).

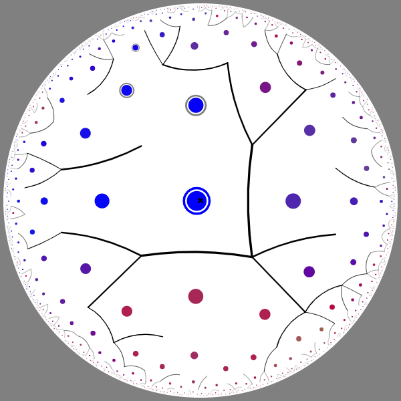
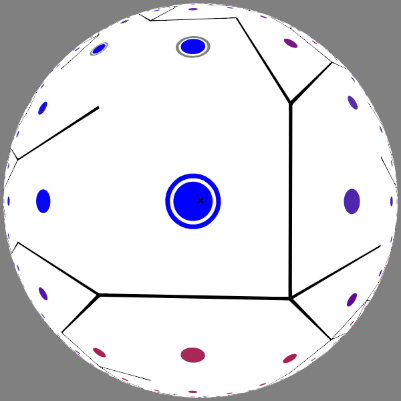




Figure 6: Octahedral holonomy maze (Segerman, N.D).

*Figure 6* shows a 3D printed maze that demonstrates the concept of holonomy, showing non-Euclidean geometry can also be represented in the real world. With this the green piece (rook) needs to be removed from the sphere, this can only be done by having it at the start point but with a different rotation. To achieve this the user moves the rook around the maze; due to the concept of holonomy, the rook will rotate based on its movement. With the addition of the blocking pieces on the maze, the user must move the rook in a specific way to achieve the required end rotation. This shows how non-Euclidean geometry causes rotation to be accumulated as things move along its surface, *figure 6* being equivalent to a positively curved 2D plane. Holonomy can be a unique mechanic that can be implemented into puzzle games or even exploration games. As a result of this element of non-Euclidean geometry being underused in the games industry, it is evident that further investigation is likely to reveal more interesting elements of non-Euclidean geometry that can be used as key gameplay mechanics.

## Summary

One of the key parts of this investigation is to analyse the uses of non-Euclidean geometry, namely Hyperbolic and Spherical geometry, regarding the games industry. The investigation has found many uses not only in the games industry but also for simulating real-world environments. This is becoming further apparent if we consider the applications of Space-Time curvature (Fiscaletti and Soril, 2014) as the bending of Space-Time can be considered as the warping of a flat plane. Whilst not having an impact on the games industry it is still worth considering that the uses of non-Euclidean geometry ventures out further than the games industry.

One of the most common uses of Hyperbolic geometry is in the creation of tile-based games due to the greater number of tiles Hyperbolic geometry can facilitate. This means that the interesting nature of the geometry can help to counter some of the less interesting natures of games. “That would not be very interesting in a Euclidean tiling, but hyperbolic geometry makes it fun—not only because it feels very different, but also because it is possible to use the hyperbolic properties tactically, to deal with large groups of enemies” (Kopczynski et al, 2017). This shows that Hyperbolic geometry can be used to make simple games far more interesting, in the case of HyperRogue (Kopczynski et al, 2017) this is done by having simple AI that moves towards the player, as a form of attack, but increases the number of enemies to account for their lack of depth. Due to the Hyperbolic geometry the player can still keep up with the hordes of enemies due to the increased range of movement Hyperbolic geometry provides.

Regarding Spherical geometry, the primary uses lie within simulating things that lie on a curved surface. Allowing the mapping of the curved nature of our planet to a 2D screen. Whilst not having large scale effects on the games industry this already has an existing presence within the world of navigation. As a result of Euclid’s definition “A line which lies evenly between its extreme points is a straight or right line” (Casey, 2007). Whilst navigating on large distances the effect of the Earth’s curvature must be considered. However, the application to games begins to show itself when considering the curvature a Spherical geometry provides. This is primarily due to the unique mechanics that could be created from seeing further away geometry as larger. This use of Spherical geometry is shown in the game Hyperbolica (CodeParade, 2020). With this form of geometry, it is practical to say that strategy games could succeed extremely well if this is utilised correctly.

This leads to the conclusion that both Hyperbolic and Spherical geometry have practical applications in games. Hyperbolic geometry has more practical uses in a 2D world, however, this is because it can provide an interesting space to be navigated when viewed from a top-down perspective. It is only when we enter the 3D world that Spherical geometry being to outshine Hyperbolic. In the 3D world, Hyperbolic geometry is simply a more exaggerated Euclidean one, with more space fitting in the distance to what would be possible. Whilst this has many applications, Spherical geometry allows for far more interesting uses. This is due to how all lines converge, this means that far away geometry will still appear to be large when viewed. As of this, the player can view the whole environment from the position in 3D space. This can lead to interesting mechanics that the player can take advantage of. Furthermore, the nature of the curvature found in Spherical geometry is far more evident in Spherical geometry when considered in 3D. As of this, it can be concluded that whilst both Hyperbolic and Spherical geometry are applicable in both 2D and 3D, Hyperbolic has far more uses in 2D, and Spherical has far more potential in 3D.

# 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 geometry in an interesting way.

## Possible Issues

### Level Creation

As a result of the Spherical 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 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 is because the Spherical geometry will have to have a different number of squares on each vertex to account for its curvature. 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 to 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 geometry in the game, the method in which the game is projected will have to be changed. Since the game will use Spherical geometry, Geometric projection will likely 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 the whole triangle will be discarded, resulting in nothing being drawn.

As a result of the issues mentioned, Geometric projection might not be used. This resulted in the conclusion that Stereographic projection must be used as it renders the whole geometry in one pass. However, depth issues will likely 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

Casey, J., Euclid, (2007) ‘The First Six Books of the Elements of Euclid: And propositions I. – XXI. Of Book XI,. And an Appendix on the Cylinder, Sphere, Cone etc.’, Cornell University

CodeParade, (2022) ‘Hyperbolica’ (To be released), Microsoft Windows, Linux [Game]. Available At: <https://store.steampowered.com/app/1256230/Hyperbolica/>

Connon, J., W., Floyd, W., J, Kenyon, R., and Parry, W., R., (1997) ‘Hyperbolic Geometry’, *Flavours of Geometry*

Coxeter, H., S., M., (1989) ‘Introduction To Geometry’. 2nd edn. Toronto. John Wiley & Sons, Inc.

Coxeter, H., S., M., (2019) ‘Non-Euclidean Geometry’. 5th edn. Toronto. University of Toronto Press.

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>

Fiscaletti, D., Sorli, A., (2014) ‘Space-Time Curvature of General Relativity and Energy Density of a Three-Dimensional Quantum Vacuum’, *Annales Universitatis Mariae Curie-Sklodowska Lubin – Polonia,* 115(AAA), pp. 54 - 78

Francis, G., (2002) ‘The Beltrami-Klein Model of the Hyperbolic Plane’

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>

Handest, F., (1956) ‘Construction In Hyperbolic Geometry’, *Canadian Journal of Mathematics.* Cambrige University Press, 8, pp. 389 – 394, DOI: <https://doi.org/10.4153/CJM-1956-043-0>

Kallosh, R., Lindre, A., (2015) ‘Escher in the Sky’, *Comutes Rendus Physique,* 16(10), pp. 914 – 927, DOI: <https://doi.org/10.1016/j.crhy.2015.07.004>

Kern,W. F., and Bland, J. R., (1948) ‘Solid Mensuration with Proofs’, 2nd edn. New York: Wiley

Koenderink, J., J., van Doorn, A., J., Lappin, J., S., (2000) ‘Direct measurement of the curvature of visual space’, *Perception*, 29(1), pp. 69 – 79, DOI: <https://doi.org/10.1068/p2921>

Kopczynski, E., Celinska, D., and Ctrnact, M., (2017) ‘HyperRogue: Playing with Hyperbolic Geometry’, *Bridges 2017 Conference Proceedings*

Madore, D., (2013) ‘Hyperbolic Maze’, Web Browser [Game]. Available At: <http://www.madore.org/~david/math/hyperbolic-maze.html#explanations>

Manning, H., P., (1963) ‘Introductory Non-Euclidean Geometry’, New York, Dover Publications, Inc.

Metacritic, Portal by Valve Software, Available At: <https://www.metacritic.com/game/pc/portal>

Ravindran, R., (2007) ‘Euclid’s Fifth Postulate’, *Resonance,*12(4), PP. 26 - 36

Ryan, P., J., (1986) ‘Euclidean and Non-Euclidean Geometry: An Analytic Approach’, The Press Syndicate of The University of Cambridge

Segerman, H., (N.D) ‘Octahedral holonomy maze 1’, [Product], Available At: <https://www.shapeways.com/product/GEXKSWT77/octahedral-holonomy-maze-1-rook-sold-separately?optionId=211502590&li=shops>

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)

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

Valve, (2007) ‘Portal’, Microsoft Windows, Xbox 360, PlayStation 3, OS X, Linux, Android, Nintendo Switch [Game]. Available At: <https://store.steampowered.com/app/400/Portal/>

Valve, (2011) ‘Portal 2’, Microsoft Windows, Xbox 360, PlayStation 3, OS X, Linux, Nintendo Switch [Game]. Available At: <https://store.steampowered.com/app/620/Portal_2/>

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

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)

Wolfe H., E., (2012) ‘Introduction to Non-Euclidean Geometry’, New York, Dover Publications, Inc.

Whittlesey, M., A., (2019) ‘Spherical Geometry and Its Applications’, Chapman and Hall/ CRC, p. 27

# Glossary of Terms

Transformation matrices – A matrix of values that represents a linear transformation

Interpolate – To smoothly transition from one value or position to another.

Chord – A straight line that lies on a circle with both endpoints sitting on the circumference of the circle.