UFCFHQ-45-3 Comprehensive Creative Technologies Project Pre-Production & Research Document

Connor Easterbrook
THE UNIVERSITY OF THE WEST OF ENGLAND

CONTENTS

UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document	
Student Name:	Connor Easterbrook
Student Number:	18015101
Project Title:	Evolving Video Games with Complementary Geometries: Non-Euclidean Level Design

Abstract

This document's purpose is threefold:

- Collect found research materials that relate to the project and compile them into a single document.
- Write down the key information that the research reference provides and discuss its relevance to the project.
- Be presented as a project pre-production document.

The research material provided will be relevant to my Comprehensive Creative Technologies Project. This project has two main goals: finding the best combinations of non-Euclidean geometry within game genres and finding which combinations work best when utilized on a large map designed to "fit" in a small area. An example of this would be a large "haunted house" map with hyperbolic geometry, or the illusion of it, to give the player a feeling of moving around the house whilst locally being contained within a static five metre-squared area.

Keywords:

Euclidean geometry; Spherical geometry; Hyperbolic geometry; Video game design; pre-production.

1. Introduction

The purpose of this document is to establish the beginning of the project and ensure that there has been an appropriate amount of research conducted in order for the project to be developed in a stable way.

1.1. Project basis

Non-Euclidean geometry is a popular topic within the scope of video game development. This is possibly due to its contradictory nature of what we, as third-dimensional beings, see as “normal”. The popularity surrounding the topic would insinuate a correlating amount of projects containing non-Euclidean geometry but, in truth, the number of projects containing it in even the most superficial way contrasts the popularity.

This project was created when an inquisitiveness began towards the perspective change and the following change in gameplay, that could be implemented through non-Euclidean geometry. Then upon beginning the initial research about the topic, and finding the contrasting project count to popularity, the curiosity towards this project grew. It was found that there is a lot of technical discussions surrounding it but not a lot of practical, design-based discussions and projects. So it was decided to approach this topic from a design perspective and my initial question was “Could a large map be created in a small space while catering to a specific genre, and how would that impact the playability?”

Other questions then began to flood my mind. “How would a claustrophobic horror game feel in a large house? Could a shooter be made in the same house-type map?” There are many ways to implement a non-Euclidean design into a game and my project will be focused on how each type of geometry would affect the following gameplay on the same maps. This project will be focusing on the Spherical and Hyperbolic non-Euclidean geometric theorems.

1.2. Core concept

The core concept of a game development project is described by [1]Laramée (2007, p.191), as “creating what the game is going to be out of thin air”. In other words, this is the section in which the task is identified and forms an idea that will accurately showcase the direction of the project. The task for this project would be to explore different geometries within multiple game genres to figure out which combination creates the greatest gameplay experience with a compact map design. Or, in plain English, which genres work best with a large map fit into a small space with a different perspective

The idea for creation would be a build consisting of all the attempted genres and map designs in a level-based style with a running explanation as to why a particular combination was or was not selected. This would run similar to an interactive presentation including the cut gameplay aspects. This would allow for an experience where the player can see where each concept would have gone, had it been continued. Other builds would contain selected combinations of the genre, non-Euclidean geometry, and map. Alongside the two build ideas would be the documentation discussed in section ‘6.2. Deliverables’ of this document.

1.3. Initial concerns

There will always be initial concerns going into a project but there are a couple of large concerns. The most prevalent concern is being able to use a familiar software application to achieve the project’s goal. This is because popular video game engines do not get designed to accommodate such a niche topic as non-Euclidean geometry in a true way. Another large concern is having the ability to comprehend non-Euclidean geometry at more than just a superficial level. A solution to these two concerns would be to achieve the illusion, or design, of non-Euclidean space instead of achieving “true Euclidean space”. More detail on this can be found in section 3.4. ‘Non-Euclidean Implementation’.

1.4. Objectives

There are a number of objectives that are aimed to be achieved by this project. It will be easier to break them down into three sections, each with its own primary focus.

1.4.1. Project objectives

What is this project intending to achieve?

- Develop a video game prototype that could be developed further into commercial release,
- Present possible combinations for non-Euclidean video games that can be used by other game developers,
- Create an important piece for a professional portfolio that covers a wide range of skills.

To summarise, this project is intended to create a foundation for possible future work, whether it be a continuation of the prototype developed, allowing access to better employment opportunities, or a source of inspiration for third parties to develop their own non-Euclidean video games.

1.4.2. Research objectives

What are the intended discoveries of this project?

- Discover how changes in perspective can affect gameplay based on its genre,
- Explore the history behind non-Euclidean geometry,
- Deepen understanding of non-Euclidean geometry both visually and mathematically,
- Discover the best combinations of geometry and genre.

To summarise, the research objectives of this project prioritise deepening the understanding of both video game development and non-Euclidean geometries, both individually and in tandem.

1.4.3. Learning objectives

What is intended to be learned, personally, from this project?

- Further develop game development skills in a commonly-used engine,
- Advance ability to manage a project efficiently when working as an individual.

To summarise, the personal learning objectives from this project intend to establish a foundation of skills that would be helpful both inside and outside the games development industry.

1.5. Reference checklist

To ensure that all references are reliable and fit-for-purpose, each source will go through a four-step checklist to determine its validity.

Keyword	Description
Relevance	How relevant is this to the current question? Does it suit the requirements?
Accuracy	How accurate is this information? Is it backed up by other sources?
Trustworthy	Is the source known for factual information? What is the academic view of the source?
Purpose	For what reason does this source exist? Is it to inform?

2. Geometry discussion

The historical background behind Euclidean and non-Euclidean geometry is important to know to appropriately utilize and understand the topic. It is also interesting to learn how these geometries came to be theorised, and the Mathematicians' thought processes behind them. In this section, the focus will be on both of these things and discussing the geometry's compatibility with this project. A final section will also be added to discuss, in brief detail, additional non-Euclidean geometries. The research provided, in this section, will be focused on the discussion, details, and historical background of the geometries.

2.1. Euclidean

Euclid, the 'founder of geometry', and whilst living between the mid-4th and mid-5th centuries BC, deduced the theorems to current-day Euclidean geometry from a small set of axioms and wrote them in his mathematical treatise consisting of thirteen books [2]'The Elements' (Artmann, 2020).

Euclid theorised five postulates for Euclidean geometry. They are as followed:

1. To draw a straight line from any point to any point.
2. To extend a finite straight line continuously in a straight line.
3. To describe a circle with any centre and radius.
4. That all right angles are equal.
5. That, if a straight line falling on two straight lines make the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which the angles are less than the two right angles (the parallel postulate).

Euclidean geometry is the most common geometry taught in schools, and until the 20th century Euclidean geometry was just 'geometry'. This is due to no popular contesting theories being voiced until non-euclidean geometries began to offer other forms for mathematicians to focus on.

There is nothing inconsistent with Euclid's postulates but his fifth postulate, the 'parallel' postulate, was controversial. This is because Euclid's 'parallel' postulate could not be proven. While not overly substantial, it is also worth noting that Euclid's postulates do not meet the standards of modern axiom systems. Euclid's postulates contain the concepts of point, line, angle, and circle but these can only be realized by the reader. There is no set size for points, nor length for lines. Modern Axiom systems do not attempt to define their main concepts but, instead, describe their behaviour ([3] Wolfe, 2013. p.1 - p.21).

Non-Euclidean geometries offer alternative postulates for the situations where Euclid's theorems were not adequate. One example of this would be using a sphere, where parallels are non-existent. [4]Gowers (2002) discusses how spherical geometry violates three of Euclid's postulates whilst remaining "correct" mathematically. Another example is in hyperbolic geometry where Euclid's 'parallel' postulate is replaced with a new postulate that states lines both converge and diverge in different locations in the same situation while remaining true to Euclid's remaining four postulates.

2.2. Spherical

Greek mathematicians, two centuries after Euclidean geometry was theorised, began to work on spherical geometry. This was possibly due to the Earth having been recently discovered to be rounded rather than flat, requiring navigators to recreate maps with the help of mathematicians. Regardless of the reasoning, it is known that spherical geometry was used for its practical applications in navigation and astronomy. A popular character in the history of spherical geometry was Leonhard Euler, whose work benefited many mathematicians and continues to do so.

The history behind spherical geometry is complicated to briefly explain due to the consistent additions made to the theorems by mathematicians. [5]Papadopoulos (2014), states that spherical geometry, “as the study of the figures made by intersections of planes with the sphere”, was developed by Theodosius. Following Theodosius of Bithynia, Menelaus of Alexandria, and Claudius Ptolemy’s work, Michel Chasles is quoted to have said that “this doctrine, which is almost similar to that of straight lines, is not all of spherical geometry.” It is mentioned how Fuss, a “follower” of Euler, who studied spherical ellipses, the location of points on a sphere where the distance between two fixed points remains constant, deepens the theories surrounding spherical geometry with his work.

A sphere is defined as a closed surface in three-dimensional space, formed by a set of points an equal distance from the centre of the sphere and its radius is calculated as the distance between its centre and the surface. A straight line within a three-dimensional space with the sphere could lead to one of three outcomes.

- An intersection at two points when the line passes through the sphere.
- An intersection at one point on the sphere when the line is tangent to the sphere at the intersection point.
- No intersection occurs.

In a case where a line passes through the centre of the sphere, the first outcome, the points of intersection form the antipodes of the sphere. Like lines and spheres, a plane and sphere in three-dimensional space can have the outcomes of:

- No intersection
- An intersection at one point on the sphere when the plane is tangent to the sphere at the intersection point.

But the third outcome would be different.

- An infinite number of intersection points when the plane cuts through the sphere and forms a circle of intersection.

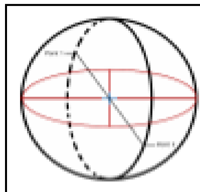


Figure 1. Example of antipodal points.

The focus would then not be on the line in the figure but instead on the great circle with the plain and dashed lines. A great circle is defined as the circle of intersection which shares the same radius and centre as the intersected sphere. A small circle is a title for plane intersections where the plane is not intersecting with the sphere at a central location ([6] Szecsei, 2004).

When three great circles intersect on the sphere’s surface a spherical triangle is formed. An example of a spherical triangle being different from a Euclidean triangle could be if you drew each great circle at a right angle from the previous. Let us use the Earth as an example, pick a point on the equator, Point A, draw a line a quarter of the way around the circle. From the end of this line, Point B, draw a line to the north pole, Point C. Connect Point C to Point A and you’ll have your triangle. To see the image with great circles, just continue each line of the triangle around the globe. With its three right-angles, this triangle will equal two-hundred and seventy degrees because spheres have positive curvature.

The mathematics discussed offers us a key insight into how spherical geometry can be portrayed visually within a video game. With a spherical triangle having three right-angles, the player’s movement on a spherical map would have to accommodate this. Each ninety-degree angle would represent a ninety-degree player rotation, which gives us our basis for spherical geometry-based movement. Fortunately, an anchoring script should allow for easy access into this spherical movement as the sphere’s positive curvature dictates the spherical triangles’ angles. With spherical movement accomplished, the next step would be to allow for a perspective that either relates to spherical geometry or can coincide with the map design.

In Euclidean geometry, two parallel lines will never intersect, but these lines will intersect twice in spherical geometry. This difference is known as the Riemannian Postulate. Spherical geometry bears many similarities and connections with Euclidean geometry but also contains key differences. Despite these differences, spherical geometry is not sufficient to resolve Euclid’s ‘parallel’ postulate controversy ([7] Maltby et al., 2021). This controversy was instead resolved in hyperbolic geometry.

2.3. Hyperbolic

Nikolai Ivanovich Lobachevsky and János Bolyai, are two key figures in the creation of hyperbolic geometry. Lobachevsky published the complete system of hyperbolic geometry in 1829, while Bolyai discovered it independently and published his findings in 1832. These were not popularised until after 1862 when another mathematician Carl F. Gauss' private letter to Heinrich Schumacher about his thoughts on hyperbolic geometry was published. ([3] Wolfe, 2013. p.45 - p.56). The published work of the three mathematicians put to rest the controversy of Euclid's parallel postulate while proving that the remaining four postulates remain consistent.

Visualising hyperbolic geometry is certainly not simple for those who have not extensively looked into it. There are multiple ways to present hyperbolic geometry but if the calculation was visualised then the parallel postulate will first have to be completely negated. To do this, avoid making every line intersect with every other line because this would be spherical geometry and the interior angles of triangles would become more than one-hundred and eighty degrees. The correct option contains infinite lines that intersect the other line while never intersecting with the original line; this would be hyperbolic geometry as it leads to the interior angles of triangles summing up to less than one-hundred and eighty degrees. This hyperbolic triangle's interior angle sum would depend on the area, and there would be a larger triangle within the amount made from the geometry ([8] Comment, 2016). A Poincaré disk best shows off hyperbolic geometry as all lines, even the ones that appear to be curved, are straight due to the difference in the definition of distance to Euclidean geometry.

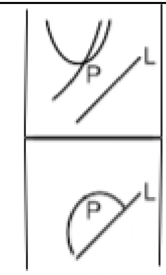


Figure 2. Alternatives to the Parallel Postulate

So, to summarise so far, hyperbolic geometry has three distinct features.

- The sum of interior angles in a triangle is less than one-hundred and eighty degrees.
- The sum of interior angles within a triangle depends on the triangle's area.
- There is a larger triangle than others when creating a triangle.

Outside of the calculation visualisation, a real-world view of what a hyperbolic plane would look like can be found in any hyperbolic paraboloid shape. These would contain areas that bend away from a point and others that bend towards a point; an example would be a saddle, where two sides are lowered to allow for leg comfort and the perpendicular sides are bent upwards for grip. This is known as negative curvature and is the core of hyperbolic geometry. In other words, in hyperbolic geometry, distance is measured differently as the distance to the edge of the plane increases the closer you get to it.

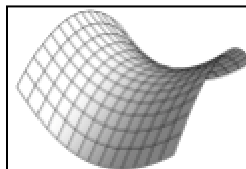


Figure 3. Saddle shown as a hyperbolic paraboloid shape

The implementation of hyperbolic geometry into a video game ties into this next section of explaining how hyperbolic geometry works as it is the main concept of this project. In hyperbolic geometry, you can create shapes and allow each angle to be any angle, due to how the plane is warped the curvature allows for it. An example of this in practice would be a plane that consists of hexagons, and each angle within the shape's interior is ninety degrees; this would allow for four hexagons to be connected at each vertex whereas in Euclidean geometry you could only have three. This fits into the core objective of this project as it allows you to manipulate surfaces to fit more in what seems to be a smaller area, even though planes are much larger in hyperbolic geometry than Euclidean; directly fitting in with the presented idea of a large map set in a seemingly small space.

The method of presentation for hyperbolic geometry is currently a mystery as it is yet to be decided on a method of approach for this project. Currently, implementing a type of teleporting window that offers the illusion of hyperbolic movement within three-dimensional space is the only primary focus. There is also a possibility of creating the map within a tile-like system that allows changes to occur to the map in a way that befits hyperbolic geometry. Two-dimensional applications do not match with this project's intentions of "large maps in small spaces" as length is a requirement alongside height and width.

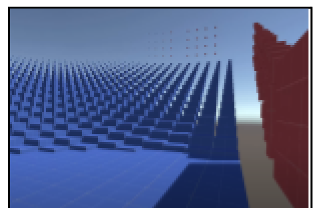
2.4. Elliptical

It is not currently intended to utilise elliptical geometry within this project, it would be remiss not to discuss it and the reasoning behind my not implementing it. Elliptical geometry is similar to spherical in many ways, it is appropriate to describe it using the spherical geometry section above. If a plane description was taken from spherical geometry and changed to elliptical, then take the spherical plane and remove one of the two points that are directly opposite each other. That is to say that, in spherical geometry, all lines intersect in two points, whereas in elliptical geometry lines intersect at only a single point. The reason for doing this is to allow elliptic geometry to satisfy the axiom of a unique line passing through any two points ([9] Sommerville, 1914. p.88 - p.101). Every point corresponds to a line and is the absolute pole. Any point on this line can form a pair with the pole which is orthogonal. The distance between the pair is a quadrant.

The reason why elliptical geometry was not considered within this project is due to its ambiguity within the gaming industry. There is just not much exploration into an elliptically geometric gamespace. Also, it is somewhat redundant to create an example in both spherical and elliptical geometries due to their similarity.

2.5. "Mostly Euclidean"

Whilst it is difficult to offer historical information about Non-Euclidean geometry in general, this section is equally as appropriate to include, due to modern issues. There is a common belief, aided by popularised videos and gameplay, that if there is anything Euclidean within a project then it cannot be classified as "non-Euclidean". As [10]Henderson (2020) discusses, a 'non-Euclidean' geometry is simply a geometry that differs from Euclidean geometry. Though some differ much more than others, it was decided to describe the geometries that are very similar as "mostly Euclidean."



[11] Figure 4. "Mostly Euclidean" game (Doucette, 2017. Used with permission)

An example of what can be considered a "mostly Euclidean" game would be [11]Doucette's (2017) provided image. The image shows the player walking on Euclidean geometry that shrinks and disappears based on distance. Even though the gameplay is Euclidean and the local geometry is Euclidean, when perceived from a global perspective, it can be said that this is a non-Euclidean game.

"Mostly Euclidean" geometry will be included in my project artefacts as it provides a nice middle-ground between Euclidean geometry and the previously mentioned non-Euclidean geometries. It offers a great amount of flexibility in level design in the event of an artefact based on the previously mentioned non-Euclidean geometries being unachievable or uninspiring. One method of implementation would most likely consist of a Euclidean level with portals that offer a hyperbolic-like effect.

3. Non-Euclidean Game Design

It is important to discuss and compare the many different implementations that this project offers so that it achieves the best possible result from the deliverable artefacts. To accomplish this, more focus should be on the design aspect of the game development, as opposed to the technicalities of it. This section will be more opinionated but the research provided will be related to video game examples and game marketability.

3.1. Video game examples

While there is not a large enough range of non-Euclidean games for it to be a specified genre, there are still games focused on it and the gameplay it brings. This section is for the discussion of these games in brief detail - the geometry, how it affects gameplay, and the genre.

3.1.1. HyperRogue

Developed by Zeno Rogue, and released in January 2015, 'HyperRogue' is a third-person roguelike based on a hyperbolic plane ([12] Zeno Rogue, 2021). The map is procedurally generated as the player moves while already loaded tiles remain saved and untouched while being deactivated to save memory. While the visuals are stylised, they do limit the marketability. It is presented in a two-dimensional spherical plane with hexagonal tiles, meaning that all visuals are presented from a birds-eye view. Due to the infinite nature of the game, there is a lot to do in one playthrough. The game has a wide range of enemies and locations, meaning that each playthrough will be unique.

The hyperbolic geometry within this game is presented through a tile-based movement system and a matching perspective. The tile-based movement system works with a turn-based action system, with each action causing an enemy action. Hexagonal tiles were used as they allow movement in more than just four directions and can be placed adjacent to each other without any gaps. This game is designed for very specific players.

3.1.2. Tea For God

Developed by Void Room, and currently unreleased, 'Tea For God' is a first-person virtual reality rogue-lite shooter based on portals ([13] Void Room, 2021). Portals are used to create an illusion of hyperbolic space within Euclidean geometry by making the player feel like they haven't teleported when they have actually been transported to a different location, which is a different size. The map is procedurally generated as the player moves while already loaded tiles remain saved and untouched while being deactivated to save memory. The visuals of the game appear to be a mixture of retro, low-poly, and rounded cylinders, a modern look meaning that this game is appealing to a large demographic. While the game is infinite, it is not yet fully developed so each playthrough will be unique but gameplay will feel repetitive at times.

The only aspect of non-Euclidean geometry within this game is the portals. These allow for more map design options while keeping the aesthetic of the map pure. More space inside allows for more diverse maps. The procedural generation is interesting for a shooter as provided it is optimised well enough, it can allow for changes in gameplay pacing. This game is designed for a large range of players.

3.1.3. Non Euclidean Room

Developed by 'NuSan' in forty-eight hours, and released for Ludum Dare 37, 'Non Euclidean Room' is a first-person exploration game that relies on portals ([14] Nusan, 2021). Portals are used to create an illusion of hyperbolic space within Euclidean geometry by making the player feel like they haven't teleported when they have actually been transported to a different location, which is a different size. The map has been constructed and repeats each level with visual changes that change the atmosphere. The visuals of the game follow basic shapes but there is a sketch effect that enhances the experience of the game and makes it appealing to a large demographic. The game is not very replayable as each playthrough would not be unique.

The only aspect of non-Euclidean geometry within this game is the portals. These allow for more map design options while keeping the aesthetic of the map pure. More space inside allows for more diverse maps. The constructed map allows for a more specialised experience. This game is designed for a large range of players.

3.1.4. Hyperbolica

Developed by ‘CodeParade’, and currently unreleased, ‘Hyperbolica’ is a first-person exploration game that utilises many types of non-Euclidean geometry ([15] CodeParade, 2021). Hyperbolic and spherical geometry are heavily used in the game alongside visuals that match the geometry but still look appealing to the average player. Each map has been constructed and the game follows a typical level-based system. The visuals may differ each level but there is an overall low-poly art style that can be appealing to a large demographic. The game has varied replayability as each playthrough is not unique but the non-Euclidean visuals may invoke a desire to revisit it.

The non-Euclidean implementations within this game are the main selling points of this game. Casual players are able to experience non-Euclidean geometry without knowing the mathematics behind it. Due to the streamlined nature of the game, the player experience can be enhanced through specific level designs and scripted events. This game is designed for a large range of players.

3.1.5. Antichamber

Developed by ‘Alexander Bruce’, and released in January 2013, Antichamber is a first-person puzzle game that uses non-Euclidean elements ([16] Bruce, 2013). The elements include, but are not limited to, forced perspective and portals in Euclidean space, allowing for simplicity. Each level is constructed and follows a typical level-based system. The visuals consist of black and white, with bold colours only being used under specific circumstances meaning that the style will be appealing to a more specific audience. The game is not very replayable as each playthrough would not be unique.

The non-Euclidean elements in this game are used as a supporting force to the general gameplay. The visuals of the game limit the game's audience as it takes a more artistic approach which could deter the casual player. The constructed level designs allow for puzzles to be used in optimal locations and create a streamlined gameplay experience. This game is designed for a specific audience.

3.1.6. Knossu

Developed by ‘Jonathan Whiting’, Knossu is a first-person horror game that uses non-Euclidean elements. The elements include, but are not limited to, portals and non-Euclidean visuals, both of which enhance the disorientating nature of the horror game ([17] Whiting, 2015). Each level is constructed and does not follow a level-based system. The visuals are designed to look like the classic ‘DOOM’ game without the monsters and shooting, which limits the possible audience. The game has varied replayability as each playthrough is not unique but the experience of playing the game may bring players back for a second time.

The non-Euclidean elements in this game are used as a supporting force to the general gameplay. The visuals of the game limit the game's audience as it takes a more artistic approach which could deter the casual player. The constructed level designs allow non-Euclidean elements to strongly support the map design in specific locations, enhancing the gaming experience. This game is designed for a specific audience.

3.2. Genre selection

Genre selection is an important part of video game creation as it not only sets the overall tone of the game, it affects all aspects of development in order to make the experience more polished. This section discusses each of the chosen video game genres, how compatible they are with non-Euclidean space, and how non-Euclidean geometry can enhance the experience.

3.2.1. Horror

Non-Euclidean geometry appears to have a high level of compatibility with the horror genre as it can feed into the disorientating and uneasy atmosphere that horror games strive for. The unfamiliar nature of non-Euclidean geometry could also enhance the tension within the game. Specific music would need to be created for the game as each horror game holds a unique feel to it. This means that extra work would be required on audio creation and scripting, especially because the music should develop in synchronisation with the gameplay.

Non-player characters may not work well with non-Euclidean geometry as the ai pathing may be unable to keep up with the different map designs. This is not too big of an issue, however, as horror games can work with limited ai and a general uneasy atmosphere. For example, an infinite hyperbolic loop that slowly becomes eerier and features scripted events on specific loops.

3.2.2. Action

Non-Euclidean geometry appears to have a high level of compatibility with the action genre as it can create environments that match the video game's intensity. The unfamiliar nature of non-Euclidean geometry could enhance energetic level designs as the player would be required to focus more on the gameplay and controlling their movement. General fast-paced music can be used for an action game as the atmosphere is not a key aspect of immersion. This means that more effort can be placed on gameplay and level design.

Stationary non-player characters could be implemented to diversify gameplay aspects. Only stationary as moving characters may not be able to path correctly in unusual geometric environments. An example of matching gameplay with this genre would be a fast-paced survival shooter where the character must fight their way through rooms to reach a location before the time limit ends.

3.2.3. Adventure

The adventure genre is an odd choice for this project as it is a genre that non-Euclidean games frequently move towards. The reason it has been included though is due to the calming nature an adventure game can offer and the diversity that it allows. Adventure games can cross into most genres so it is a safe option to have. Specific music would only be required for this genre if the atmosphere requires it, otherwise, general music can be used. The most effort with an adventure game would be with level design and not gameplay aspects, meaning that the focus would be on creating a nice visual experience.

Mobile non-player characters could be implemented in an adventure game as the map designs would usually allow for larger areas where ai pathing would have an easier time. An example of matching gameplay with this genre would be an exploration game that visits different non-Euclidean elements with a storytelling aspect behind it.

3.2.4. Puzzle

Puzzle games rely on subverting expectations, so non-Euclidean geometry would be highly compatible. Elements like forced perspective, portals, and hyperbolic spaces can be used to force the player to think outside of Euclidean geometry and match the tone that the game would go for. There would be no requirement for non-player characters as the level designs would be the main focus of gameplay. Specific music would not need to be created as the atmosphere is not a key focus for pure puzzle games.

With the focus on level design and puzzle implementation, visuals would be a very important part of the game. This means that a lot of work would be required to make the game visually appealing. An example of matching gameplay would be a game where players must solve issues in rooms with the threat of death if they guess incorrectly.

3.3. Marketability

Marketability, in this case, means the available audience of video game players that would be interested in a non-Euclidean-style game. This section is important as the possible demographic of a game dictates all aspects of development in order to appeal to the selected audience as much as possible. There are a few demographics to target with regards to non-Euclidean gameplay. This section will discuss those in brief detail with research being provided by [18]Greenberg et al. (2008) and [19]Limelight (2021). Age group data will be focused on those under 18 as the produced games will not feature content that will give them an adult rating.

3.3.1. Age

Age is an important factor in game development. Countries all over the world restrict the accessibility of video games based on their content so that a more vulnerable audience does not get affected by mature themes.This section will be focusing on the average weekly hours played in video games based on age groups. The study conducted by [18]Greenberg et al. (2008) contains a number of statistics that discuss age.

Age Group (years)	**Time Spent (hours)**
7 - 10	18.71 (18 hours 42 minutes) [Male] 6.70 (6 hours 42 minutes) [Female] 12.7 (12 hours 42 minutes) [Average]
11 - 14	22.94 (22 hours 56 minutes) [Male] 11.54 (11 hours 32 minutes) [Female] 17.24 (17 hours 14 minutes) [Average]
15 - 18	16.30 (16 hours 18 minutes) [Male] 7.19 (7 hours 11 minutes) [Female] 11.74 (11 hours 44 minutes) [Average]

Table 1. Weekly Hours of Video Game Gameplay by Age.

From this table, it can be deduced that children play video games more often in young adolescence than at any other point. There is also a larger number of females who become more interested in video games than males in the fifteen to eighteen age group. The reason this is important is that it lets us know which age group to market towards for optimal commercial efficiency when attempting a “PEGI” age rating target.

[19]Limelight (2021) discusses how males would play more frequently and for longer periods of time than females. Females have a higher percentage of video game players who play for less than four hours a week, while men have increased percentages of players with higher playtimes.

3.3.2. Gratification

Gratification within video games is highly important as players are more likely to spend more time on an activity that makes them feel happy or fulfilled. This section will discuss the genres of games that incite the most gratification in players. [18]Greenberg et al. (2008) ‘Table 4’ discusses mean gratification by age group (1-7, 7 being highest), appropriately matching the previously referenced table. Only previously mentioned genres will be prioritised, however.

Genre	Age Group (years)		
	7 - 10	11 - 14	15 - 18
Strategy, Fantasy, and **Adventure** games (“imagination”)	4.54 (64.86%) [Male] 4.45 (63.57%) [Female] 4.50 (64.29%) [Average]	5.33 (76.14%) [Male] 4.86 (69.43%) [Female] 5.10 (72.86%) [Average]	4.68 (66.86%) [Male] 4.12 (58.86%) [Female] 4.40 (62.86%) [Average]
Arcade, card, trivia, board, and **Puzzle** games (“traditional”)	3.63 (51.86%) [Male] 4.60 (65.71%) [Female] 4.12 (58.86%) [Average]	3.55 (50.71%) [Male] 4.91 (70.14%) [Female] 4.23 (60.43%) [Average]	3.81 (54.43%) [Male] 4.96 (70.86%) [Female] 4.39 (62.71%) [Average]
Sports, Fighting, **Shooters**, and Racing games (“physical”)	5.04 (72.00%) [Male] 3.80 (54.29%) [Female] 4.42 (63.14%) [Average]	5.55 (79.29%) [Male] 4.43 (63.29%) [Female] 4.99 (71.29%) [Average]	5.01 (71.57%) [Male] 4.09 (58.43%) [Female] 4.55 (65.00%) [Average]

Table 2. Video Game Genre Preferences by Age.

This table suggests a number of things which, for the sake of simplicity, will be broken down through bullet points.

- Age groups 7 - 10 had the second-highest approval rating of “imagination” games.
    - The overall low approval ratings suggest that this age group audience is the least agreeable of the age groups.

- Age groups 11 - 14 had the highest approval ratings of “imagination” and “physical” games while having the second-highest approval rating of “traditional” games.
    - The overall high approval ratings suggest that this age group is the most agreeable of the age groups.

- Age groups 15 - 18 had the highest approval ratings of “traditional” games while having the second-highest approval rating of “physical” games.
    - The overall approval ratings suggest that this group is average in terms of agreeability.

- The “traditional” games received the lowest approval ratings overall by a large margin.
    - This suggests that “traditional” games would be the least effective in regards to commercial release.

Overall, the gratification data provided suggests that the eleven to fourteen years age group would be the best target audience. While also regarding “traditional” games to be the least gratifying option of the three video game groups.

3.3.3. Summary

Marketability is a key part of the video game production process, so data related to it will always be relevant. The referenced data in this section suggests that, when making a video game avoiding an adult rating, the best target audience is the eleven to fourteen years age group. When targeting this audience, it would be best to create a video game in the “imagination” or “physical” game groups.

The data also suggests that “traditional” games are considered more viable for an adult audience, alongside unexplored genres like ‘Horror.’

3.4. Implementation

As briefly discussed, the implementation of non-Euclidean geometry within this project will avoid major technicalities and focus more on visual design. As such, this section will discuss how to achieve a non-Euclidean implementation that looks non-Euclidean, as opposed to programming a script that follows non-Euclidean physics. The two major implementations would be ‘hyperbolic portals’ and ‘spherical visual effects’ as they cover both simple and advanced non-Euclidean effects.

3.4.1. Hyperbolic portals

Featured in the modern games Splitgate,’ and ‘Portal,’ hyperbolic portals feature a window-like viewport into connected areas and smooth teleportation that feels akin to walking through a doorway. They essentially stir the dynamics of maps by allowing transportation over large areas and expanding smaller areas to be more interactive.

Hyperbolic portals require a fair deal of mathematics to achieve a seamless implementation but fortunately, the coding is only intermediate-level. They can be used in numerous ways, from deployable portals to creating illusions of space not matching the initial perception (elongating corridors would be the most simple example.) The method of achieving this would be by creating two floor planes very close to one another, surrounded by walls that make up a door frame. Attach a camera to the center of each plane, and then bind the render texture of each plane to the camera on the opposite plane. Then, write a script that teleports the player to a matching position at the second portal. More work will be required to create a seamless portal but this alone will serve as a basic non-Euclidean simulation. [20]Sebastian Lague (GitHub - SebLague/Portals: Portals in Unity, 2020) has a Github repository that is focused on achieving the “portal” effect and within it the need for matrices manipulation and Unity’s shader coding ‘HLSL.’

Regarding portals, they can only be considered as a hyperbolic implementation if the rooms are connected in a way a doorway connects rooms. In this case, they can only be used to mimic connecting rooms, as opposed to, teleporting the player across an unconnected distance.

3.4.2. Spherical visual effect

Spherical visual effects can be achieved by manipulating the viewport within game engines to almost bend the camera view around the spherical object the character is on. These allow for players to have a normalised experience when in non-Euclidean space. Alternatively, a visual effect that has the opposite effect and explores the abnormal reality within non-Euclidean geometry could be created.

Spherical visual effects require a large amount of specified knowledge with the engine’s visual scripting, but fortunately, the coding is only intermediate-level. These effects could achieve a number of things, from traversing small planets that have been normalised to creating a map that restricts how far the player can see. The method to doing this would be through the use of a shader. Everything would exist in Euclidean space while appearing to be non-Euclidean. For example, a sphere in Euclidean space would be taken to spherical space by warping each object on the sphere with an offset through the translation. Then a non-collidible sphere would be instantiated and made to an equal scale to the world space. After, a volumetric shader that skews lighting into rotating around the center of the new sphere would be created. The issue with this is that characters would be required to be rotated consistently, with an artificial gravity that centers on the new sphere. These would form the spherical visual effects detailed in section ‘3.4.1.’ [21]Sebastian Lague (GitHub - SebLague/Spherical-Gravity, 2016) has a Github repository that is focused on achieving the “portal” effect and within it the need for matrices manipulation and Unity’s shader coding ‘HLSL.’

4. Engine discussion

To be able to create these projects, the selected engine will have to allow for both efficiency and simplicity in achieving the end goal. This section will be discussing possible engine options, the engine's compatibility with the project, and why or why not to use them. The research in this section will be focused on each engine and examples of the engine in use in a similar situation.

4.1. Engines

4.1.1. Godot

The Godot engine is a lightweight, free-to-use, 3D game engine that allows for coding in either 'GDScript,' its own coding language, or 'C#.' This allows the engine to be more open to a wider audience. Godot would not be a good option for non-Euclidean environments as engine access is somewhat constrained, causing geometric work to have to be written purely with code that's compatible with Godot.

This means that the only achievable non-Euclidean, without getting too advanced, would be the hyperbolic portals discussed in section '3.4.1.'

4.1.2. Unreal Engine 4

Unreal Engine 4 is a good option for non-Euclidean effects as it allows for a fair amount of freedom through coding, while blueprints simplify desired movement. The engine has not been featured much in the 'Games Technology' course though, so the necessary experience may not be provided. There are a large amount of learning materials for 'UE4', so even beginners can figure out how to achieve tasks.

The 'C++' coding language may serve as a barrier to project completion as it is believed to be a far more complex programming language to learn, as opposed to 'C#.' The compatibility of Unreal Engine 4 and non-Euclidean geometry is somewhat close. This is due to the lack of flexibility towards simulating non-Euclidean .

4.1.3. Unity

Unity is the most appropriate engine selection and the engine that will be used for this project. This is due to its unconstrained nature, allowing even engine code to be rewritten. Unity has been a large focus of the 'Games Technology' course, so necessary experience would already be provided. There are also a large amount of learning materials for Unity, so even beginners can figure out how to achieve tasks. Two of which, the previously cited references from Sebastian Lague. [20]'Portals' (GitHub - SebLague/Portals: Portals in Unity, 2020) and [21]'Spherical Gravity' (GitHub - SebLague/Spherical-Gravity, 2016).

The programming language of unity being 'C#' allows for an easier achievement of goals as it is believed to be more simple to learn than other comparable languages. The compatibility of Unity and non-Euclidean geometry is likely as close as you can get with a standard video game engine. This is due to the given flexibility that Unity provides.

5. Pre-development process

The pre-development process of this project will be where this project reaches a level that allows for the beginning of development. By utilising the research gathered within this document to guide the project limitations, choices and goals, the project will go into more depth and form the [1]“conceptual bones”(2007, p.192) of the project. A visual demonstration will be created of the project to test each considered option, alongside conceptual ideas and technical information in order to answer any remaining questions.

5.1. Macro design

Macro design is the activity of fleshing out the original idea, focusing on the first systems that will be required, and working from there to the small details that the project relies on. These small details could be game objects, level designs, player mechanics, enemies, etc. This is to allow all members participating in a project to understand what needs to be implemented and explain it to those outside of the project. Macro design also includes creating mockups and working prototypes to demonstrate the visuals and sound design of the project in order to test whether a concept will work in practice [1](Laramée, F. 2007, p.192). When macro designing, it is important to keep in mind video game theory, keeping a game feeling good. [22]Swink’s (2008) book on video game theory ‘Game Feel: A Game Designer’s Guide to Virtual Sensation’ is a good example of research material that covers this.

This section will receive more details in the ‘production’ document for this project, as that document is intended to be a report of all actions that produce deliverables, which the macro design prototypes would be. For now, this section can only discuss the superficial expectations and details of each project generally.

The expected macro design of each deliverable within this project are:

- Showcase of non-Euclidean level design
    - The level must have non-Euclidean elements
        - Hyperbolic portals, spherical visual effects, etc.
    - Gameplay should lean into non-Euclidean elements
        - non-Euclidean elements should have some level of effect on the gameplay
- Showcase of the game genre
    - The game genre should be known from playing the game as a standalone
        - An action-shooter should be action-packed, an exploration-thriller should build tension as you explore, etc.
- Delve into game genre mechanics
    - Have the game feel unique through developed mechanics that compliment the game genre
        - Shooter slow-mo, Puzzle POV, Horror psychological health, etc.

5.2. Convergent iteration

Convergent iteration is a process that begins through macro design when the mockups, prototypes, and ideas begin to form. It is with this that ideas, designs, implementation, etc. are elaborated. Each iteration would be a new prototype that is built from the experience gained in the creation of the previous. In the case of this project, each subproject would be built off of knowledge gained in the previously attempted. This method makes it easier to spot risks within the project and how compatible the selected combination would be.

5.3. Estimate timeline

Month	Task	Estimated Duration
October 2021	Write a project proposal to be submitted by 04/11/2021.	11 days (24/10/2021 - 04/11/2021).
November 2021	Finish ‘CCTP Pre-Production & Research Document’, Create ‘CCTP Production Process Document’, Develop one subproject* (artefact & documentation).	20 days (24/10/2021 - 12/11/2021), N/A 15 days (13/11/2021 - 28/11/2021).
December 2021	Research documentation ready for submission by 16/12/2021, Develop one subproject* (artefact & documentation), Create a roadmap for project development in 2022, including the previous work completed.	N/A, 15 days (01/12/2021 - 16/12/2021), 3 days (17/12/2021 - 20/12/2021).
January 2022	Prototype demonstration as 2-min video ready for submission by 24/01/2021 (exact date TBD), Develop one subproject* (artefact & documentation).	N/A, 15 days (01/01/2022 - 16/01/2022).
February 2022	Develop one subproject* (artefact & documentation), Develop one subproject* (artefact & documentation).	15 days (01/02/2022 - 16/02/2022). 15 days (17/02/2022 - 04/03/2022).
March 2022	Write a project post-production document, Work on final hand-in.	20 days (05/03/2022 - 25/03/2022), N/A.
April 2022	Artefact, final report, and final video to be completed for hand-in by 28/04/2022 via Blackboard and Github.	N/A.
May 2022	Viva submission completed for hand-in by 16/05/2022 (exact date tbd).	N/A.

*One subproject will be selected to be developed further into the primary deliverable.

6. Deliverables

The deliverables would be one of the most important items as they are the culmination of all the work and research throughout the project. Appropriate deliverables are required in order for the end result of this project to be considered complete. This section will discuss planned deliverables, possible additions and the focus of each deliverable. This section is mostly opinion-based and may not match the final result as the project may evolve from pre-production to post-production.

6.1. Portfolio

A deliverable portfolio of tests and projects, both successful and unsuccessful, is most likely the optimal way to present my findings and end result. It will allow me to construct a detailed timeline, in tandem with documentation, to accurately show my thought process, and end it with a strong conclusion. In this case, a successful project would be one that showed the ability to become a promising video game; while an unsuccessful project would be any started project that did not show enough promise to be developed further - this could range from pre-production to mid-production.

As of now, the deliverable portfolio appears it will be:

- A range of successful games, each with its own build.

    - These games do not have to be complete, full games.
    - There should be a minimum of two unique games.
        - A “unique game” constitutes different genres and geometries.
    - In total, there should be a minimum of three successful projects.

- A showcase of unsuccessful projects, each being presentable in some way.
    - This could be showcased in a game build where each scene or level is its own failed project.

6.2. Documentation

Documentation is a key process throughout a project’s lifespan. It can range from the creator’s thoughts throughout the project to in-depth analyses on what went wrong in a particular test. Appropriate documentation throughout this project will allow me to portray all relevant information in an accessible format and add depth alongside a portfolio submission.

As of now, the deliverable documentation appears it will be:

- A document dedicated to discussing the pre-production and research of the project.
    - This document should include both opinion-based and factual writing to appropriately discuss the development decisions and research in detail.

- A document dedicated to discussing the production of the project.
    - This document should include both opinion-based and factual writing to appropriately discuss why a subproject was or was not successful.

- A development diary in the form of a spreadsheet.
    - This diary will become a timeline that showcases my thought process, project development progress, and links together all deliverables.

- Both research and proposal documents.
    - The marked deliverables that were requested in the assessment specifications.

- A debriefing document to clearly discuss ownership of any used assets, and authorship of this project.

Keeping documentation specific to certain areas, while it may increase workload and deliverable count, will allow them to showcase their area much more effectively and allow for easier document navigation.

7. Bibliography

All research carried out for this document can be found appropriately referenced here following The University of the West of England's Harvard referencing system. Although it is not standard procedure, each reference has been numbered in order to make document navigation more streamlined.

[1] Laramée, F. (2007). *Secrets of the game business*. 1st ed. Hingham, Mass.: Charles River Media.

[2] Artmann, B. (2020) Euclidean Geometry [online]. *Encyclopedia Britannica*. Available from: https://www.britannica.com/science/Euclidean-geometry. [Accessed 05 August 2021].

[3] Wolfe, H. (2013) *Introduction to Non-Euclidean Geometry.* Newburyport: Dover Publications.

[4] Gowers, T. (2002) *Mathematics: A Very Short Introduction*. Oxford University Press.

[5] Papadopoulos, A. (2014) *On the works of Euler and his followers on spherical geometry* [online]. HAL Archives-Ouvertes. Available from: https://hal.archives-ouvertes.fr/hal-01064269/document [Accessed 10 August 2021].

[6] Szecsei, D. (2004) *The Complete Idiot's Guide to Geometry.* Alpha Books.

[7] Maltby, H., Thomas, J., Dash, S., Khim, J. & Ross, E. (2021) *Spherical Geometry* [online]*. Brilliant Math & Science Wiki.* Available from: https://brilliant.org/wiki/spherical-geometry/ [Accessed 25 August 2021].

[8] Comment, S. (2016) The Geometric Viewpoint [online]. *History of Hyperbolic Geometry*. Available from: https://web.colby.edu/thegeometricviewpoint/2016/12/08/history-of-hyperbolic-geometry/ [Accessed 25 August 2021].

[9] Sommerville, D.M.Y. (1914) *The Elements of Non-Euclidean Geometry.* George Bell & Sons.

[10] Henderson, D.W. (2020) Non-Euclidean Geometry [online]. *Encyclopedia Britannica.* Available from: https://www.britannica.com/science/non-Euclidean-geometry. [Accessed 29 August 2021]

[11] Doucette, M. (2017) *Unity - Non-Euclidean Geometry - Test 1.03.* YouTube.com [online]. Available from: https://www.youtube.com/watch?v=ExpFTbJE6Yg [Accessed 06 August 2021].

[12] Zeno Rogue (2021) *Hyperrogue.* [Video game] Available from: https://roguetemple.com/z/hyper/ [Accessed 28 September 2021].

[13] Void Room (2021) *Tea For God*. [Video game] Available from: https://void-room.itch.io/tea-for-god [Accessed 29 September 2021].

[14] Nusan (2016) *Non Euclidean Room.* [Video game] Available from: https://nusan.itch.io/non-euclidean-room [Accessed 29 September 2021].

[15] CodeParade (2021) *Hyperbolica.* [Video game] Available from: https://store.steampowered.com/app/1256230/Hyperbolica/ [Accessed 29 September 2021].

[16] Bruce, A. (2013) *Antichamber.* [Video game] Available from: http://www.antichamber-game.com/ [Accessed 29 September 2021].

[17] Whiting, J. (2015) *Knossu.* [Video game] Available from: https://jonathanwhiting.com/games/knossu/ [Accessed 29 September 2021].

[18] Greenberg, B., Sherry, J., Lachlan, K., Lucas, K. and Holmstrom, A. (2008) Orientations to Video Games Among Gender and Age Groups. *Simulation & Gaming*, [online] 41(2), pp.238-259. Available at: https://www.researchgate.net/publication/247740438_Orientations_to_Video_Games_Among_Gender_and_Age_Groups [Accessed 29 September 2021].

[19] Limelight (2021) *THE STATE OF ONLINE GAMING – 2020*. [online] Available from: https://www.limelight.com/resources/white-paper/state-of-online-gaming-2020/ [Accessed 17 November 2021].

[20] Lague, S. (2020) *GitHub - SebLague/Portals: Portals in Unity*. [online] Available at: https://github.com/SebLague/Portals/tree/master [Accessed 2 December 2021].

[21] Lague, S. (2021). *GitHub - SebLague/Spherical-Gravity*. [online] Available at: https://github.com/SebLague/Spherical-Gravity [Accessed 3 December 2021].

[22] Swink, S. (2008) *Game Feel: A Game Designer's Guide to Virtual Sensation.* Elsevier.