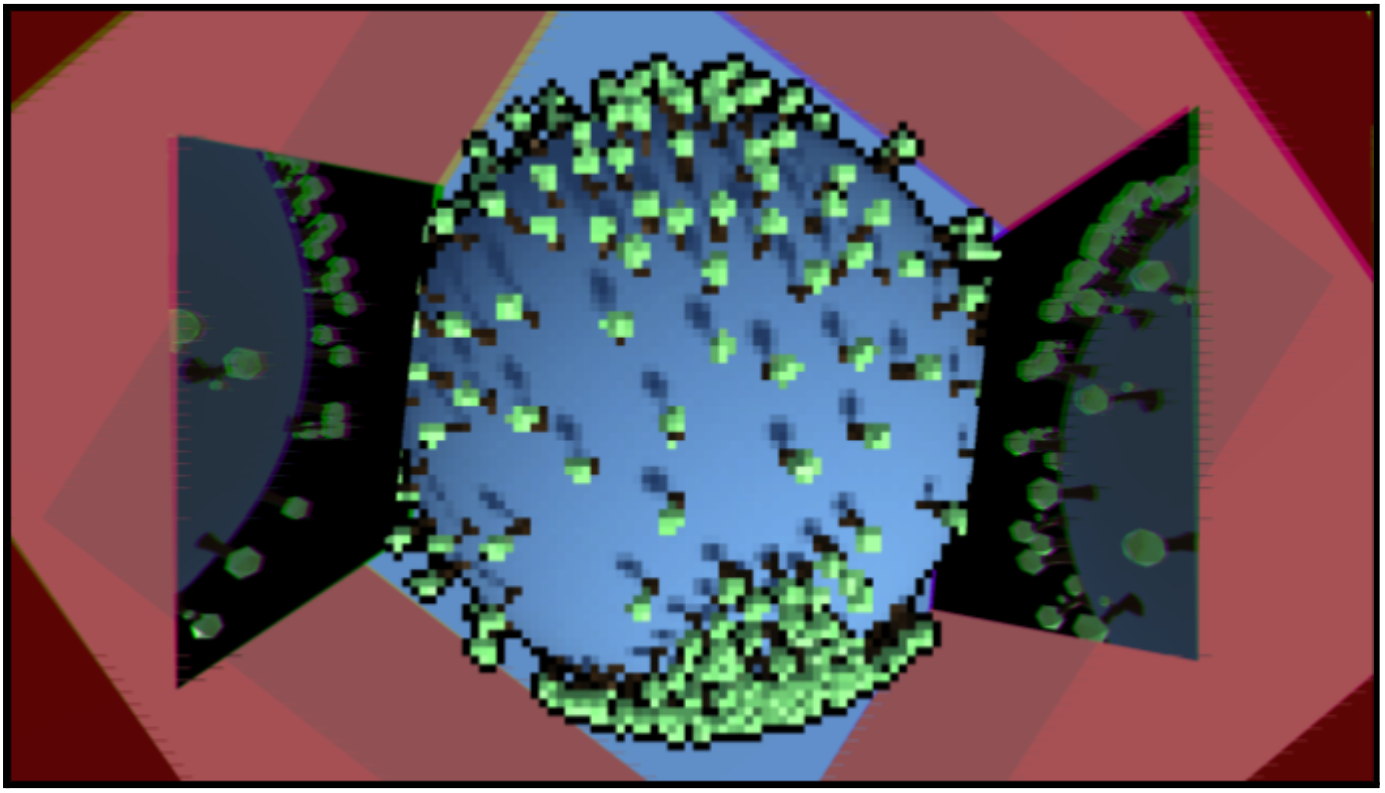
Comprehensive Creative Technologies Project: Evolving Video Games with Complementary Geometries: Non-Euclidean Level Design

Connor Easterbrook
connor2.easterbrook@live.uwe.ac.uk
Supervisor: Marcus Lynch

Department of Computer Science and Creative Technology
University of the West of England
Coldharbour Lane
Bristol BS16 1QY



Abstract

Non-Euclidean geometry is frequently discussed but rarely implemented in the video games industry. Using the Unity Game Engine, experimentation was carried out with non-Euclidean geometry in different game genres. The aim of this project was to investigate if and how gameplay is affected by different non-Euclidean mechanics

Keywords: Non-Euclidean; Euclidean; Hyperbolic; Spherical; Rigidbody; Unity; Physics

Brief biography

Non-Euclidean geometry is a subject of personal curiosity. The opportunity to delve into this topic arose through an observation that although the topic is often discussed, non-Euclidean geometry is rarely utilised within video game development. After conducting research and examining video games featuring non-Euclidean elements, a decision was made. The project examined combinations of game genres and different non-Euclidean mechanics rather than developing a single non-Euclidean video game. It was believed that conducting a project such as this will offer me skills that would serve well in several career choices while satiating a personal curiosity.

Link to portfolio: https://exoweb.dev/portfolio/

How to access the project

To access the different parts of the project, please visit the relevant links below:

Final subproject source code and files:

https://github.com/GoatBandit/UFCFHQ-45-3-Comprehensive-Creative-Technology-Project-Final-Subproject

Alpha and Beta Stage Subprojects' source code and files:
https://github.com/GoatBandit/UFCFHQ-45-3-Comprehensive-Creative-Technology-Project-Subproject-Beginning

Final video:
https://youtu.be/cTUx0pBz8Mw

ReadMe files are included within the Github repository links containing information on the repository and accessing the relevant files. The first link provided is for the final developed subproject, including all relevant files, while the second link provided offers access to the alpha and beta stage subprojects' code.

---

1. Introduction

Non-Euclidean geometry has been a topic of research since the year 1832, and yet it continues to be of great interest in the games industry, in particular. Yet, there is a disparity between this curiosity and the number of developed games featuring non-Euclidean geometry. This disparity is the core inspiration behind this project. There are also questions to be answered through experience of working with non-Euclidean game mechanics.

- Is there a reason behind this disparity?
- Can non-Euclidean mechanics enhance the gameplay of video games both passively and actively?
- What new considerations should be taken when working with different geometries?
- What are the best combinations of non-Euclidean mechanics and video game genres?

The Non-Euclidean mechanics featured within the project are members of both of the primarily discussed non-Euclidean geometries, hyperbolic and spherical, as well as geometries that contradict Euclid's postulates in other ways. The focus of these mechanics was on their visual representation rather than the mathematical calculations that enforce them.

This project was developed within the Unity Game Engine (version 2020.3.30f1) using C# for the script programming and a mixture of HLSL and Shader Graph (Universal Render Pipeline) for the shader files. Most assets were designed for full independence to ease the transfer between subprojects and other Unity projects.

The objectives of this project were:

- To investigate possible combinations of video game genres and non-Euclidean geometries that could be commercially successful.
- To create a professionally handled and well-documented project that covers a wide range of skills.

This project's deliverables are as follows:

- A portfolio consisting of seven pre-produced video game subprojects.
- A final Unity project that functions as the final artefact, subproject, an interactive showcase of the subproject portfolio, and a standalone video game.
- Documentation of the project's research, pre-production, production, and two time logs of contrasting detail.

2. Literature review

Due to the broad nature of this project and the desire to focus on the most relevant research sources used, this section is divided into three specific groups encompassing the theoretical and practical requirements for the project.

2.1. Geometric theory

Geometric theory allows for the knowledge to create and detail the non-Euclidean mechanics used. Regarding Euclidean geometry, Artmann (2019) discusses the history, mathematics, and significance of Euclidean geometry. This created a basic comprehension of Euclid's postulates within the project.

Wolfe (2013) states the core of non-Euclidean geometry and details the more prominent geometries. This information served to expand the project's possibilities. Henderson (2020) details how a geometry can be considered non-Euclidean as long as it differs from Euclidean geometry. This ensured that the visual mechanics in the project are non-Euclidean.

One non-Euclidean geometry is spherical geometry which can be simple to grasp but with many intricacies. Papadopoulos (2014) discusses the history and mathematics of spherical geometry, expanding the project's knowledge of spherical geometry, keeping its depiction accurate. Gowers (2002) explains how spherical geometry differs from Euclidean geometry and which of Euclid's postulates it violates thereby showcasing how geometries remain mathematically correct even though they differ from others.

Another non-Euclidean geometry is hyperbolic geometry which plays a small but pivotal role within the project. Comment (2016) offers a simplified view into hyperbolic geometry while covering its mathematics, history, and visual theory which helped significantly in the understanding and creation of the portal mechanic within the project.

2.2. Video game theory

Video game theory helps to make the subprojects into effective and enjoyable minigames. Swink (2008) covers topics such as the role of sound, ancillary indicators, metaphor importance, and perception. This supports the subprojects' development cycles.

Learning about the average video game user is an important part of planning. Greenberg et al. (2008) detail the statistics and correlations of video game preference with gender and age. Limelight (2021) delves into comprehensive detail on the present video game industry and the average video game user's preferences and play time. These sources guided the plans for the subprojects to ensure they were enjoyable.

CodeParade (2022) showcases non-Euclidean geometry being used within a video game made with the Unity Game Engine. The development logs made in relation to the video game aided in non-Euclidean theory within Unity.

2.3. Video game development

Sources focusing on video game development kept the project focused. Laramée (2007) breaks down the core concept of a game development project into "creating what the game is going to be out of thin air." Before discussing the stages of development in thorough detail. This guided the development process of this project and the subprojects featured.

This project is not wholly unique in its endeavours. Guimaraes et al. (2015) discuss the gameplay possibilities of non-Euclidean geometries in video game design. The most difficult mechanics within the project were due to the portals. Lague (2020) showcases an advanced method of smooth portals within Unity which helps break down each step required. Lague (2016) also developed a project following simple spherical geometry which also aided in development.

Ilett (2021) teaches the use of every Unity Shader Graph node within Unity's Render Pipelines. This was very useful within the project's development.

HyperRogue (Zeno Rogue, 2021), *Tea For God* (Void Room, 2021), *Non Euclidean Room* (Nusan, 2016), *Antichamber* (Bruce, A., 2013), and *Knossu* (Whiting, J., 2015) have each made unique video games with non-Euclidean level design that was referred to throughout the project's development. Doucette (2017) showcased simple non-Euclidean visuals within Unity.

3. Research questions

The question that started this project's aim was: *"To what extent does geometry affect a video game's synergy between game genre and level design? How does this affect the video game's creation?"* This led to more concise questions:

- How does non-Euclidean geometry change a video game's dynamics?
- What new considerations are to be taken when factoring in different geometries?
- What game genres are the most compatible with different geometries?

It was important to keep to the concept that video game development is most effective when developing for others rather than oneself within the project. To go about this, one final research question was asked.

- Would non-Euclidean video games be appealing to the average video game user?

With these questions at the core of the research process, the project could remain focused on its original aim while also factoring in game development techniques and processes. These would keep the created subprojects feeling like video games, as opposed to feeling like presentations.

4. Research methodology

This section explains the methodologies used in the gathering of research references relevant to the development of the project as well as their reliability.

4.1. How and why were the research sources chosen?

Research sources were discovered through browsing Google search engine and Google Scholar for relevant materials. These options were selected to allow users access to many research sources. These sources would have their reliability questioned through the four-step checklist (see Table 1).

Keyword	Description
Relevance	How relevant is this to the current question? Does it suit the project's needs?
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?

Table 1. Source reliability checklist. From 'UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document'

Only if each check was confirmed to be true was a research source included in the project's research.

4.2. Are there any research sources that can be considered "secondary research?"

Due to this project's focus, most geometry-based research is considered "secondary research." This is because they comprise modern research on the work of previous mathematicians. Meanwhile, the video game development sources can be primary research, as they were created by the referenced party. Regarding the video game theory research, they can be considered as being a mixture of primary and secondary research because of some collecting the data from other sources.

4.3. Could there be a bias in research outcomes?

Whether because of choice or application, there is always a chance that bias could lead to one research source being selected over others. The source checklist is in place to mitigate this risk as much as possible. However, even if there was research bias, it would not damage this project's validity because of the project's practical nature. The end product would rely on the knowledge gained through each previous subproject, as opposed to the research sources.

5. Ethical and professional principles

It is important to clarify that this project is ethical, and that all research methods were applied professionally in research and practice. For this, this section is split into three subsections.

5.1. Could this project generate any ethical concerns?

It is not believed that this project could generate any ethical concerns, as there are no parties involved other than the project creator. Besides this, the project is not breaching upon any topics that could be deemed offensive, nor is it involving any information regarding to an individual.

5.2. Could this project negatively impact individuals?

It is not believed that this project could negatively affect an individual, as it is not producing anything contradictory to any beliefs. The only disagreeable aspect within the project would be the project's conclusion of compatible video game genres and non-Euclidean mechanics. This conclusion would solely be of opinion and not passed off as factual.

5.3. Research sources that include media

It was a rule within this project to seek any licences pertaining to the media referenced from a source. In cases of there being a licence or, if unknown, the project requested permission from the research source's author before using the piece of media. These media references are present as referenced figures in this report. If an author could not be contacted, then the media was deemed unusable. This occurred although most licences allow for educational work because of the desire to keep standards professional.

6. Research findings

This section discusses this project's research findings, ordered in the same manner as section 4.

6.1. Geometric theory

Artmann's (2019) discussion of Euclidean geometry allowed for a foundational understanding within the project of Euclid's postulates. Following this, Wolfe's (2013) statements about the core of non-Euclidean geometric theory made clear the direction of the project. An example of this would be how an infinite number of hyperbolic right-angled squares can be connected on the same vertex, creating a map that defies the seemingly Euclidean geometry. Henderson (2020) clarified this with teachings that a non-Euclidean geometry exists so long as an appropriate postulate of Euclidean geometry is broken. This project relies on this teaching, changing the geometry of level design either locally or globally. See Figure 1 for an example.

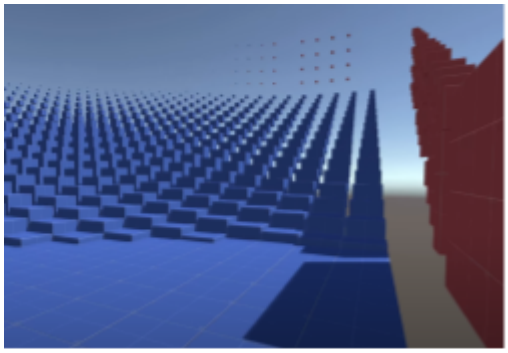


Figure 1. "Mostly Euclidean" game showcasing a non-Euclidean game mechanic in Euclidean space. (Doucette, 2017. Used with permission)

Papadopoulos (2014) taught applications with spherical geometry and the mathematics that would be required to develop sphere-based mechanics. This aided in visualising and explaining the project's spherical subprojects. Gowers (2002) allowed for an understanding of spherical mathematics, which aided subprojects featuring a spherical scene. Specifically, the discussions from both mathematicians on antipodal points aided in understanding spherical gravity within video games as seen in Figure 2.

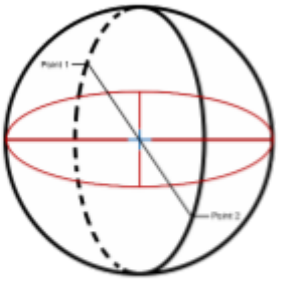


Figure 2. Example of antipodal points in spherical geometry.

Comment's (2016) discussion on projective geometry was useful in the understanding of hyperbolic portals. The visuals were depicted with conic sections as seen in Figure 3.

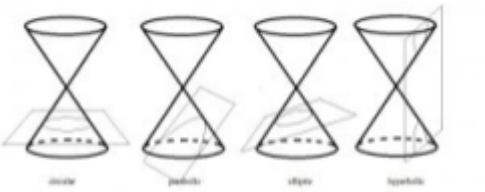


Figure 3. Conic sections cross-sectioned by a plane. (Comment, S., 2016).

6.2. Video game theory

Swink (2008) was a great learning material for understanding how to plan and execute a video game project. Leading to a reliance on convergent iterations for successful development. This aided in the understanding of considerations that should be taken within game development. With the first half being covered through the geometric theory references.

The work by Greenberg et al. (2008) made clear the type of video game that the subprojects should aim to mimic based on their core gameplay. Meanwhile Limelight (2021) guided this project's plans through data on what types of video games would increase the interested user audience.

The video game by CodeParade (2022) focuses on changing global geometries within the exploration genre. This provided a key example of how a non-Euclidean video game should be created. This allowed for a professional view of the considerations that should be made with this style of video game.

6.3. Video game development

Laramée's (2007) book was a great learning experience, teaching the step-by-step process of video game project development. This aided in the project's planning process and discovering considerations to take when

implementing different play styles through non-Euclidean mechanics.

Guimaraes et al. (2015) allowed insight into how non-Euclidean mechanics may alter a video game's gameplay dynamics. Meanwhile Lague's (2020) discussion of portals within the Unity Game Engine taught about approaching non-Euclidean mechanics practically within the engine. A similar teaching was found from Lague's (2016) spherical work, but instead of hyperbolic geometry, it allowed for a glimpse into required considerations for spherical geometry.

Ilett's (2021) work aided in an understanding of how Shader Graphs' work in the Unity Game Engine. As well as revealing how effects that suit this project's nature might be created.

The publications from Zeno Rogue (2021), Void Room (2021), Nusan (2016), Bruce (2013), and Whiting (2015), showed what video game genres are compatible with non-Euclidean mechanics. Also, allowing for an understanding of what non-Euclidean video games are appealing to a general user audience through their reviews and publicly recorded sales.

7. Practice

Due to this project's nature, the discussed practice will comprise concise sections for an appropriate document length. These sections discuss the project's pre-production, subproject production, and then an in-depth look at the artefact. In that order. The artefact section will be split into two sections.

First, it is important to disclose the tools used within the project. This project was developed within the Unity Game Engine (version 2020.3.30f1); Programmed using Visual Studio Code (version 1.66.0). The graphical assets were created using the modelling program Blender (version 3.1.0) as seen in 'Figure 4'. Finally, the audio files used in the project were created in the digital audio workstation FL Studio (version 20.7.2).

Figure 4. A plant model used within the project being created in Blender.

Second, the limitations within the project. As this project was conducted by a student, the project's boundaries were required to be specified to avoid becoming too complex. Decided by the overseeing advisors, the project was limited to focusing on the visual effects of the non-Euclidean mechanics, as opposed to the mathematical. This means the mechanics are more simple and appear more as "illusions." Mathematics will only be discussed at a superficial level.

Third, this project was developed through a subproject-based system in order to keep each experimented combination unique. Each subproject was required to be an individual investigation in order to reliably answer the project's questions and achieve its objectives.

7.1. Pre-production and planning

Upon starting, several factors had to be considered. The combination of game genre and non-Euclidean mechanics were required to be chosen, alongside a plan for approaching each subproject. Please see Appendix E for a more in-depth look at the pre-production and planning in this project.

7.1.1 Game combinations

Deciding on combinations of game genre and non-Euclidean mechanics was essential to this project. Observations were made of the referenced video games, and plans were drawn up discussing the advantages and disadvantages of genres with the subproject's mechanics. Following this, a subjective view was written up on the subproject's scalability, interest value, entertainment value, accessibility, and originality. Through this write-up, the subproject's were graded on their success.

7.1.2. Convergent iteration

Once combinations were selected, it was decided that convergent iteration was the most appropriate method of development. Each iteration would be a new subproject that is built from the experience gained in creating the previous. This method made it easier to spot risks within the project and the selected combination's compatibility.

7.2. Alpha-stage subprojects

Alpha stage subprojects within this project comprise combinations developed with the mechanic in mind, so gameplay is superficial. This section will focus on the developments made when working on the subprojects.

7.2.1. Horror with portals

This subproject's concept was the horror game genre mixed with portals to achieve a hyperbolic square-like effect. The belief was that this combination would work well because of the natural fear response to unfamiliar environments (Grupe & Nitschke, 2013). This subproject's mechanic breaches Euclid's fourth postulate, as all right angles are not equal to each other. If the user were to walk straight, turn right and walk straight again, through different rooms, they would not end up in the same room. Therefore, the local right angle within the user's movement is not equal to a global right angle. This specific example with portals would fall into hyperbolic geometry. See Figure 5 and Figure 6 for visual examples.

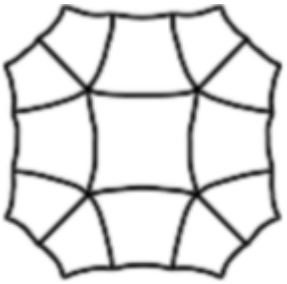


Figure 5. A drawing of a hyperbolic square. Each corner is a right angle but more than four squares are connected at a vertex.

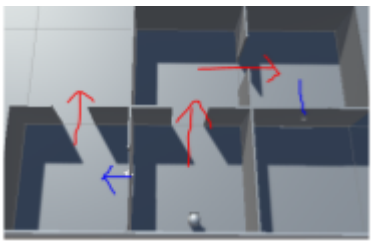


Figure 6. A small portion of the Horror With Portals subproject map, showcasing the hyperbolic square in Euclidean space.

For the portal mechanic to be developed, the portal work by Lague (2020) and Ilett (2021) was heavily referenced. These were used to troubleshoot any issues that arose throughout the mechanic's development. The mechanic works visually by checking for the linked portal. Then moving its camera to the linked portal's location, calculating the clipping plane and creating a render texture. Smooth movement through the portal is done by getting the user's transform, calculating its position in the linked portal and then teleporting the player to the other portal while keeping the movement velocity untouched. The user-specific teleportation code can be found in Appendix F.

The clipping planes caused issues throughout the project's development. The issue was that upon user camera movement, the clipping plane would move out of position briefly. The programming for the clip plane was to get the distance between the portal plane and the player (clipPlaneWorldSpace) and then using this to calculate the position of the portal camera. With the portal camera position, the clip plane can be drawn based on the portal plane bounds (clipPlaneCameraSpace) and then rendered (cameraMatrix). An example of this code is shown below.

```
if (Mathf.Abs
(planeToPlayerDistance) >
(nearClipOffset))
{
        Vector4 clipPlaneCameraSpace
        = Matrix4x4.Transpose
        (Matrix4x4.Inverse
        (portalCamera.worldToCameraM
        atrix)) *
        clipPlaneWorldSpace;

        var cameraMatrix =
        mainCamera.CalculateObliqueM
        atrix
        (clipPlaneCameraSpace);
```

```
        portalCamera.projectionMatri
        x = cameraMatrix;
}
else
{
        portalCamera.projectionMatri
        x =
        mainCamera.projectionMatrix;
}
```

7.2.2. Dissolving floor puzzle

Regarding the dissolving floor, the best genre was the puzzle genre. This is because the global visual change would aid a genre that involved visual challenges. This subproject infringes upon Euclid's second postulate as a straight line is not indefinite in global space as the world only exists around the user and specified objects, as seen in Figure 7.

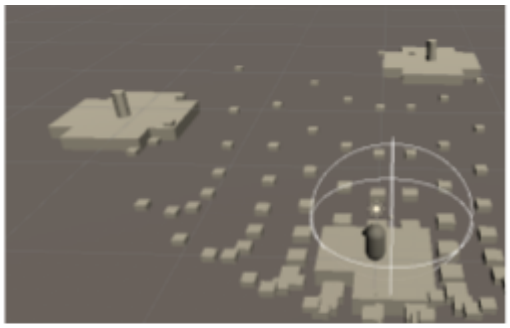


Figure 7. Visual example of dissolving environment based on location of player and objects.

Doucette's (2017) work aided in the inspiration and visualisation. Though, it was produced within this subproject inefficiently at runtime. This was due to each child object of a transform having its tag read and added to a list. Then each frame this list would be run through with a for loop. Each time the for loop progressed, the transform of the child object was changed. Frames were low. This was improved drastically within the interactive showcase.

7.2.3. Spherical exploration

Exploration was chosen for this subproject as the effect gained from the spherical world is only showcased if the user is travelling around it. Euclid's fifth postulate is being broken in this subproject as parallels can not be drawn on a sphere, as seen in Figure 2. This mechanic was chosen due to Lague's (2016) spherical work showcasing how the user's perspective of objects changes based on their spherical position. This can be seen in Figure 8.

Figure 8. Two identical towers appearing differently sized due to the user's spherical position

Getting the correct spherical rotation worked simultaneously with setting up gravity within the user controller script. This was done by calculating the target direction from the player to the centre of the planet, and ensuring the user's rotation matched the line drawn, as shown below.

```
Vector3 targetDirection =
(playerRigidbody.position -
planetGameObject.transform.position
).normalized;

playerToPlanetRotation =
Quaternion.Slerp
(
        transform.rotation,
        Quaternion.FromToRotation
        (playerRigidbody.transform.u
        p, targetDirection) *
        playerRigidbody.rotation,
        cameraRotationSmoothTime * 2
);

playerRigidbody.MoveRotation
(playerToPlanetRotation);
```

The Subproject Beginning repository was created at this point to comprise subproject mechanics. Through managing this repository and making the user controller script the manager for the mechanics, discoveries were made in project management. It was found that having a base level and necessary mechanics easily accessible for new subprojects allowed for an efficient start.

7.2.4. Inside sphere explorer

This subproject utilises a shader to create a spherical mask that surrounds the user, masking everything outside of the radius. An example can be seen in Figure 9. Regarding Euclid's postulates, this subproject is more similar to the Dissolving Floor Puzzle than Spherical Exploration, as a straight line is not indefinite.

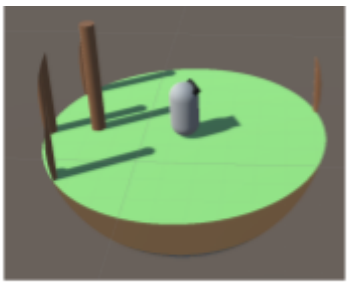


Figure 9. Inside-Sphere visual effect showcased within the Unity Editor.

This subproject required knowledge with High-Level Shader Language (HLSL). Alisavakis (2018) guided the shader's development as it aimed to achieve an effect similar to his. The result was a material-based shader. Transparent if a certain distance away from the user while visible if close to the player. The code below showcases how the user position (_PlayerPos) changes the transparency.

```
float distanceFloat =
distance(_PlayerPos, IN.worldPos);

float effectAmount = saturate
((distanceFloat - _MaskRadius) /
-_MaskSoftness);

clip(effectAmount - 1);

float4 lerpColor = lerp
(transparentColour, colourFloat,
effectAmount);

float4 lerpTransparency = lerp (0,
1, effectAmount);
```

7.2.5. Gravity changing scale puzzler

This subproject focuses on two mechanics due to them not being non-Euclidean, instead warping the user's visual experience. The gravity changing effect was implemented to add another layer to the puzzle genre, while the scaling effect alters the geometry of objects based on the user's perspective. An example can be seen from the code below.

```
RaycastHit hit;
if (Physics.Raycast
(transform.position,
transform.forward, out hit,
Mathf.Infinity, scaleLayer))
{
        targetObject.position =
        hit.point -
        transform.forward *
        offsetFactor *
        intendedScale.x;
        float currentDistance =
        Vector3.Distance(transform.p
        osition,
        targetObject.position);

        float distanceCalc =
        currentDistance /
        initialDistance;

        intendedScale.x =
        intendedScale.y =
        intendedScale.z =
        distanceCalc;

        targetObject.localScale =
        initialScale *
        intendedScale;
}
```

This code casts a ray forward from the camera position onto appropriate surfaces. Upon raycast hit, the object's scale is multiplied by the length of the ray compared to the user's initial position. The object is scaled halfway between the user and the surface in order to keep the object the same size in the user's vision.

7.3. Beta-stage subprojects

Beta stage subprojects within this project comprise combinations that use already developed mechanics alongside improved gameplay. This section will focus on the developments made when working on the subprojects. These were created with the intention of investigating the mechanics more thoroughlyl.

7.3.1. Gravitational tactical scaling

Utilising the scaling, gravity, and inner-sphere mechanics, this subproject aims to create a tactical video game. To enhance gameplay, a field of view script was created for guard-type characters. Large objects were used to obstruct the user and the guard's vision, while a low ceiling allows for risky traversal. See Figure 10 for a visual example.

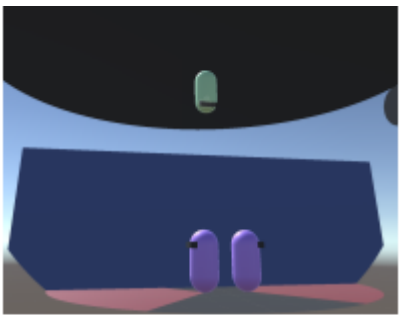


Figure 10. Visual example of the Gravitational Tactical Scaling subproject showing the user on the ceiling while guards are below.

7.3.2. Portal sphere shooter

Featuring the portal and spherical exploration mechanics, this subproject focused on the shooter game genre. To add a deeper level of gameplay, a target system was implemented. These targets were placed in specific locations to utilise exploration and the portals to get to the correct shooting positions, as seen in Figure 11.

Figure 11. Visual showcase of portal sphere shooter's target being shot at.

7.4. Artefact

The deliverable artefact features both an interactive showcase of the previously mentioned subprojects and the final subproject. Due to this, this section will be more in-depth than the previous subproject discussions.

7.4.1. Interactive showcase

The interactive showcase part of the artefact features the previous subprojects altered to fit into a linked, single playable experience. Starting the interactive showcase is a menu to function as the landing page for the user. Its function is simple and allows the user to disable or reduce sounds.

Following this, the user is placed into a tutorial environment to introduce movement and a couple of mechanics. See Figure 12 for an example. These are the scaling, field of view, and Simon Says mechanics. The aim is for the user to avoid guards, enter the large building, and complete the Simon Says puzzle.

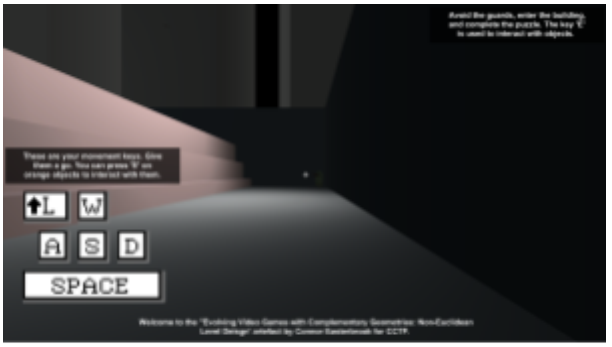


Figure 12. Showcase of the introductory 'Outside' scene that explains controls and the project's context to the user.

Upon completion of this level, the user is placed in a white room that allows access to all appropriate subprojects. This is called the white room, and it works as the core of the artefact. It can also be quickly returned to through the in-game menu. Regarding the subprojects featured within the white room, they were all overhauled.

Horror With Portals comprises two new stages. A whimsical introduction that allows the user to play with the portals in a controlled setting, and the gameplay. Gameplay comprises several identifiable rooms connected by portals, allowing for endless movement. The user must enter the correct portals to reach the end of the level, keeping track of room identifiers to ensure they do not move incorrectly, as seen in Figure 13. Trigger-based jump scares bring the horror aspect.

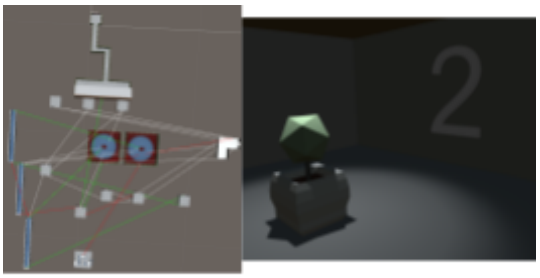


Figure 13. The Horror with portals subproject's map with drawn portal connections.

An issue that arose previously with the portals was an incorrect rotation of the user momentarily before it being corrected. This was fixed by interpolating the rigidbody, allowing it to be updated at the correct moments.

Dissolving Floor Puzzle had riddles implemented to enhance gameplay with some tiles being removed for added challenge. The previously mentioned optimization fault was fixed by removing the list aspect within the code and allowing each environmental object to function individually.

In Spherical Exploration, no mechanical improvements were required, but the core gameplay aspect was changed to feature objectives the user must collide with. This came in the form of cubes that the user must find in scene landmarks, as seen in Figure 14.

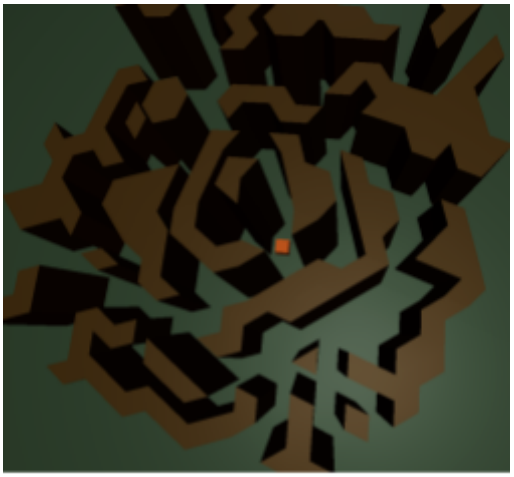


Figure 14. Objective object hidden at the centre of a landmark within the Spherical Exploration subproject.

Inside Sphere Explorer's mechanic received a complete overhaul through implementation of a Shader Graph, replacing the HLSL Shader. This was done to correct the spherical shape of the shader, as seen in Figure 9, the shader script did not match. The gameplay was changed by increasing the map size and implementing characters that hint at the user's required position.

Gravity Changing Scale Puzzler received the Gravitational Tactical Scaling level with more objects and puzzles implemented. No changes were made to the subproject mechanics but lighting was implemented and the guards received patrol scripts through transform changes.

Gravitational Tactical Scaling received a complete overhaul in the level design with shaders to guide the player. The new level is designed to have the user thinking about their approach to certain areas, avoiding guards, and using scaling to climb up walls that block them. This new level can be seen in Figure 15.

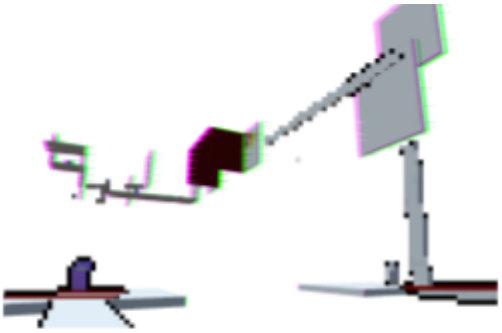


Figure 15. Visual showcase of the Gravitational Tactical Scaling subproject from the user's perspective.

Portal Sphere Shooter also received a complete gameplay overhaul, becoming a timed shooting platform challenge. The user must jump on platforms and shoot the circling targets within the time limit. Each portal sends the user to a new stage.

Upon completion of all subprojects, the user is sent to the final subproject, featuring the best combination of game genre and non-Euclidean mechanics.

7.4.2. Final subproject development

This section is to discuss the final subproject, built from the experience from all the previous subprojects.

The planned design of the level was to be split into two main areas and a single bridge between them. One main area was designed to match Escher's *Relativity* (1953) as it is compatible with the gravity shifting mechanic. The other main area was designed as a tower the user must climb using the scaling mechanic and correct portal usage. The connector utilises portals in an illusionary way.

The gameplay aim of the subproject is to solve five Simon Says puzzles. Upon the completion of a number, the inner-sphere visual effect will take effect, obscuring the user's vision. This was the decided upon combination with a puzzle genre. The level design is basic beyond the initial primary area for the subproject to remain as an interactive showpiece, and game.

While the first primary area was designed with the gravity changing mechanic in mind, the remaining level design was created before mechanical implementation. This was decided in

order to avoid overcomplication in areas other than the first primary area.

7.5. User controller

A core component within this project is the user controller script, which functions as the control centre for all the mechanics, and features an editor that gives it editor functionality. The variables of the script can be seen in Appendix G. The development of this controller was an unexpected outcome within the project but it eventually became a complex modular component. The player object began as a Character Controller component but, due to the requirements of the project, it became reliant on the Rigidbody component. Through this rigidbody component gravity, collision bounds, velocity, etc. was able to be specified to match each subproject's requirements.

All elements of the user's gameplay can be modified, including the camera pitch limitations and visual effects. As the final subproject was being developed, an audio control system was also implemented within the script. This allows for further audio work.

8. Discussion of outcomes

This section details what has been achieved within the project and discusses improvements and other approaches that could have been made.

8.1. Professional context

In its current state, including Appendices C and E, this project functions as a source of research on non-Euclidean mechanic implementation within video games. There are three primary reasons for this.

First, the project's nature means that non-Euclidean experience is gained through each subproject's development. This experience allows the project to be a reliable source in the discussion of non-Euclidean implementation within video games, as there are eight subprojects, each with differing mechanics, allowing for comparison between them. This comparison can not only be made by the project but also by professionals seeking a reference through the project's artefact.

Second, Guimaraes et al. (2015) discusses the mathematical approach of non-Euclidean geometries in video games while serving as a crucial research material for this project. However, this project comprises visual implementations with minimal mathematical requirements, leaving the topic with lower mathematical requirements. Sacrificing more complex mechanics for educational purposes was one of this project's objectives upon proposal.

The third primary reason is the accessibility of the project's mechanics. The modular design of the mechanics mixed with the user controller script (see Appendix G) allows for them to be easily utilised in other projects. This allows others to not only reference the project, but to use the same mechanics to come to their own conclusions. The code and assets will remain public for this purpose.

The project, as previously discussed, was conducted by a student, meaning programming proficiency is lacking. Despite this, the project remains educational to any who are curious about non-Euclidean implementations within video games. But, convergent iteration within the project could have been approached differently. Instead of utilising the previous subproject, which often comprised unrelated mechanics to the current, it may have been more productive to develop each subproject independently. Also, rather than overhauling all subprojects with experience gained within the deliverable artefact, it may have been more prudent to get the current subproject to that stage before moving on to the next.

8.2. Project discoveries

It was discovered that non-Euclidean geometry could hold no effect on a video game's dynamics if sought. This is because the mechanical implementations, if compatible with the chosen genre, seamlessly enhance gameplay. An example of this can be seen via the portals where the user can utilise them to create a seamless addition to the level. A spherical world is the unique case in this statement, as level design and level mechanics have to be designed around it (see Figure 8).

Regarding the portals, in their state as of the production of this report, they allow for seamless teleportation, but the clipping plane is unsteady. This was previously discussed in the Horror With Portals subsection of the Practice section. The smooth teleportation on alternating surfaces was a big achievement in the project, as it allowed for another layer of portal-based puzzles and gravity shifting. See Appendix F for the user-specific teleportation code.

Considerations were required within level design when factoring in differentiating geometries in order to keep gameplay consistent throughout a level. If there was an implemented mechanic that altered gameplay in some way, then the level should accommodate it rather than just ignore it. An example of this would be if the level featured gravity-changing walls, but the shifting gravity was not required to complete the level. Here, the

user would avoid using the mechanic and it would detract from the gameplay experience. Within the final subproject, the first primary area was designed solely around the gravity-changing mechanic, allowing for the user to explore the level through several means.

8.3. Subproject discoveries

Each subproject allowed the project to learn and evolve. These lessons were in level design, general programming, non-Euclidean visualisation, and game project management.

Horror With Portals allowed for insight into how visuals within level design can create an atmosphere independently. This atmosphere can excite the user through human fear of the unknown (Grupe & Nitschke, 2013). Two advantages of using portals to connect separate level areas is designing levels in a modular fashion, and utilising the fear of the unknown to enhance the horror genre. This can also be a disadvantage, however. If the level is modular, then it can be difficult, when designing in the editor, to get an understanding of how the level will look to the user. This was found to be a highly compatible combination.

Dissolving Floor Puzzle taught about the Unity Game Engine, specifically how it is more optimised to change many objects individually instead of throughout a single manager. Regarding the project, though, this subproject showcased how destructive an effect like this is to any gameplay outside of a user's local area, including non-player characters. This is because the world does not exist outside of certain bounds and the effect is lost if there are many objects that manifest it. This means that the combination of mechanics and genre is compatible but only on a small scale. The ideal implementation of this combination would be as an individual puzzle.

Spherical Exploration showcased the requirement of script-based transform changes with game objects in order to keep them at the appropriate rotation and distance from the spherical object. The combination of this mechanic with the exploration genre was compatible, but did not enhance gameplay much. This is likely because the mechanic does not affect the user's gameplay experience, so the subproject felt like a regular exploration video game. This subproject is considered the weakest non-Euclidean level design example.

Inside Sphere Explorer's mechanic, much like Dissolving Floor Puzzle's, was destructive to gameplay. This effect is useful for creating a darkness-like effect, but this should only be utilised temporarily. Experiencing this effect over a long period can heavily detract from the gameplay experience as the user cannot experience the level design, as seen in Figure 9. This combination of mechanic and game genre was found to be highly incompatible unless under specific circumstances. This is because exploration games rely on the user experiencing the world as they explore it, while this mechanic obstructs the user from doing so.

Gravity Changing Scale Puzzler, while not fitting the project, was an essential subproject. The mechanics created within this subproject were relied on throughout the project, as they both add a layer of depth to gameplay. These mechanics were compatible with the puzzle genre, as they both introduced new puzzles. Due to the gravity-changing mechanic, the player object required changing due to Unity's Character Controller component being unable to change its direction of gravity.

Gravitational Tactical Scaling showcased the unique scenario in which a darkness-like effect would not be destructive towards level design. Here, it allows for an added depth of gameplay because of its compatibility with the tactical genre. This compatibility has been utilised in many stealth-type video games. Also, this subproject featured the field of view mechanic, allowing for insight into triangular mesh creation within certain bounds.

Portal Sphere Shooter's mechanics, while being compatible with the game genre, did not affect the user's experience much. As discussed with the Spherical Exploration subproject, the spherical world alone does not impact the user experience outside of visual alterations that the user quickly becomes used to. Meanwhile, portals, while creating a sense of unknown, do not impact the gameplay much outside of placement changes. These placement changes can be quickly disregarded by the user as there is no depth to the teleportation outside of getting to the visible target. Thus, the shooter genre was compatible with mechanics but gameplay was not enhanced through them, meaning this subproject could be considered the second weakest non-Euclidean level design example.

The final subproject utilised the most successful genre from the subprojects alongside mechanics believed to be compatible. From this, it was found that the portal mechanic was highly compatible with the puzzle genre, alongside the gravity-changing, inner-sphere, and scaling-effect mechanics. With the inner-sphere and scaling mechanics utilised in only certain circumstances, while the gravity-changing mechanic exists throughout the project via select locations, a high level of compatibility was found with no mechanics feeling intrusive. The portal mechanic was used in a supportive way by

connecting areas. Through this usage, it was found that non-Euclidean mechanics can be especially effective if used sparingly and under specific circumstances. These circumstances could occur when the user does not expect it while traversing Euclidean space to disorientate the user or increase gameplay depth.

8.4. Project originality

While this project does not cover a unique question, it is believed to go about answering it originally. Implementing visual non-Euclidean mechanics while disregarding the complex mathematics allowed for mechanics that do not cause the user to find the video game complex. The effect created from this project that answers the question allows for the bending of Euclidean space while keeping visuals simple for the user. To explain, if the user was placed within hyperbolic space, they could traverse the world but they may not understand it; The same scenario exists within spherical space. *Hyperbolica* (CodeParade, 2021) showcases these geometries within local Euclidean space, achieving a similar effect of simple visualisation for the user, just on a grander scale while seeking answers for a different question.

8.5. User controller

Through the creation of the user controller within the subproject, many discoveries were made regarding general programming and the Unity Game Engine. An example of the discoveries is how to optimise input to movement calculations with Vector3 variables. Also, with implementing the user's rigidbody component, a great deal was learnt as it required it to be compatible with all mechanics. Though, it is intended that the user controller script be altered and improved to allow the rigidbody to be kinematic, opening more alterations with user movement.

This user controller will be utilised in future projects and further developed, likely becoming a great resource.

9. Conclusion and recommendations 300

Via the subproject-based project development, a conclusive answer was made regarding the effect of non-Euclidean level design within gameplay. It can enhance gameplay through appropriate usage. This project can be considered as a success with all objectives achieved. Though, the subprojects could have been developed further, exploring multiple genres with each mechanic rather than specified ones. Below, I will conclude on this project's potential future, third-party recommendations, beneficial impact, and the potential future.

Further work on this project would involve optimising and polishing all the featured mechanics. Following this, more depth can be added to the project's playability, making it into a commercially viable product. An example of this would add more horror mechanics to the Horror With Portals subproject. The music and sounds would need to be recreated at a higher quality, enhancing the gameplay experience. Last, an overhaul on the user controller script to allow for the rigidbody to be kinematic could be implemented in order to modify more user factors and solve any current issues such as interpolated collisions.

This project allows for any curious parties to explore non-Euclidean video game mechanics and level design. This can be done through either their own subprojects made from the project's public source code or first-hand via the project executable build. Also, if each mechanic within the project was optimised and polished, then each one could become an asset utilised by developers in other projects. If deemed reliable, this project's findings could even be referenced in similar projects.

All things considered, this project has formed an accessible entry for third-parties to explore non-Euclidean mechanics and level design within video games.

10. References

Alisavakis, H. (2018) *My take on shaders: Spherical mask dissolve*. [online]. Available from: https://halisavakis.com/my-take-on-shaders-spherical-mask-dissolve/ [Accessed 10 April 2022].

Artmann, B. (2020) Euclidean Geometry [online]. *Encyclopedia Britannica*. Available from: https://www.britannica.com/science/Euclidean-geometry. [Accessed 06 April 2022].

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

CodeParade (2021) *Hyperbolica*. [Video game] Available from: https://store.steampowered.com/app/1256230/Hyperbolica/ [Accessed 06 April 2022].

Doucette, M. (2017) *Unity - Non-Euclidean Geometry - Test 1.03*. YouTube.com [online]. Available from: https://www.youtube.com/watch?v=ExpFTbJE6Yg [Accessed 07 April 2022].

Escher, M. (1953) *Relativity*. Not on view.

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

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 06 April 2022].

Grupe, D. & Nitschke, J. (2013) Uncertainty and anticipation in anxiety: an integrated neurobiological and psychological perspective. *Nature Reviews Neuroscience*. 14 (7), pp. 488-501.

Guimaraes, F., Mello, V. and Velho, L., (2015) Geometry independent game encapsulation for non-euclidean geometries. In *Proceedings of SIBGRAPI Workshop of Works in Progress* (Vol. 1).

Henderson, D.W. (2020) *Non-Euclidean Geometry* [online]. Encyclopedia Britannica. Available from: https://www.britannica.com/science/non-Euclidean-geometry. [Accessed 06 April 2022]

Ilett, D. (2021) *How To Use Every Node in Unity Shader Graph*. Daniel Ilett: Games | Shaders | Tutorials. [online] Available at: https://danielilett.com/2021-05-20-every-shader-graph-node/ [Accessed 6 April 2022].

Ilett, D. (2021) *GitHub - daniel-ilett/shaders-portal.* [online] Available at: https://github.com/daniel-ilett/shaders-portal [Accessed 09 April 2022].

Lague, S. (2020) *GitHub - SebLague/Portals: Portals in Unity*. [online] Available at: https://github.com/SebLague/Portals/tree/master [Accessed 06 April 2022].

Lague, S. (2016) *GitHub - SebLague/Spherical-Gravity*. [online] Available at: https://github.com/SebLague/Spherical-Gravity [Accessed 06 April 2022].

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

Limelight (2021) *THE STATE OF ONLINE GAMING – 2020*. [online] Available from: https://www.limelight.com/resources/white-paper/state-of-online-gaming-2020/ [Accessed 06 April 2022].

Nusan (2016) *Non Euclidean Room.* [Video game] Available from: https://nusan.itch.io/non-euclidean-room [Accessed 06 April 2022].

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 06 April 2022].

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

Void Room (2021) *Tea For God.* [Video game] Available from: https://void-room.itch.io/tea-for-god [Accessed 06 April 2022].

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

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

Zeno Rogue (2021) *Hyperrogue.* [Video game] Available from: https://roguetemple.com/z/hyper/ [06 April 2022].

11. Bibliography

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 06 April 2022].

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

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

Wolf, M. & Perron, B. (2013) *The Video Game Theory Reader*. Hoboken: Taylor and Francis

Appendix A: Brief Project Log

The full document can be found in the "final subproject source code and files" repository linked at the top of the report, within the documentation folder. The document is called 'UFCFHQ-45-3 Comprehensive Creative Technology Project Progress Log.' This shown section is the second page, "Progress Log." For a more in-depth look at production, see the repository and find the document is called 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document.'

UFCFHQ-45-3 Comprehensive Creative Technologies Project Progress Log	Connor Easterbrook 18015101 THE UNIVERSITY OF THE WEST

	OF ENGLAND

Project Progress Log

Date	Set Target	Progress Log	Personal Thoughts
11/10/2021 - 17/10/2021	**N/A.**	'UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document' started. **(A)**	**N/A.**
18/10/2021 - 24/10/2021	**N/A.**	'UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document' progress. **(A)**	**N/A.**
25/10/2021 - 31/10/2022	Begin 'UFCFHQ-45-3 Comprehensive Creative Technology Project Proposal Document' to be submitted by 04/11/2021. **(C)**	Started and worked on 'UFCFHQ-45-3 Comprehensive Creative Technology Project Proposal Document.' **(C)(A)** 'UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document' progress. **(A)**	The project proposal should match the project's document theme and contain a fair combination of theoretical and practical research.
01/11/2021 - 07/11/2021	Finish Project Proposal document and hand in early. **(C)** Finish 'UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document' by 07/11/2021 **(F)**	UFCFHQ-45-3 Comprehensive Creative Technology Project Proposal Document' submitted early. **(C)** 'UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document' progress. **(A)**	I believe I lacked enough research on game-development and coding. Besides that, I am happy with the outcome.
08/11/2021 - 14/11/2021	Begin a subproject. **(F)**	**N/A.**	**N/A.**
15/11/2021 - 21/11/2021	Create 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' **(F)**	UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document' progress. (15/11/2021 & 17/11/2021)	**N/A.**
22/11/2021 - 28/11/2021	Finish a subproject. **(F)**	UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document' progress. (23/11/2021) 'UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document' progress. (26/11/2021 - 28/11/2021)	**N/A.**

29/11/2021 - 05/12/2021	Begin 'UFCFHQ-45-3 Comprehensive Creative Technology Project Research Document' by 04/11/2021. **(C)** Begin 'Horror with portals' subproject by 05/12/2021. **(C)**	UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document' completed. (30/11/2021) **(F)** 'UFCFHQ-45-3 Comprehensive Creative Technology Project Research Document' started and worked on. (03/12/2021 - 05/12/2021) **(C)** 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' Started. (05/12/2021) **(F)** Began the development of 'Horror with portals' subproject. (05/12/2021) **(C)** Created a subproject template for use in all subprojects. (05/12/2021) 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on (05/12/2021).	This proved to be a much longer document than initially planned but it now contains more than enough resources to comfortably support the project's development. **(05/12/2021)** Initial confusion in the writing of this document but I believe I am off to a good start. It has been productive to go over the possible ethical concerns for the project. It was deemed that there was no cause for concern on this issue. This document will probably be longer than the pre-production document but due to all the research already having been acquired, this document should develop at a smooth rate. The subproject was decided to be a combination of horror and non-Euclidean portals. Through the process of starting this subproject, it was found that I should establish a template that project's can build from in order to increase subproject development efficiency. This template should only feature the minimal features that each subproject will require, so there will be no level-design or specific scripts. Just a framework that leads to a quicker subproject start.
06/12/2021 - 12/12/2021	Finish 'Horror with portals' subproject's development by 12/12/2021. **(F)**	UFCFHQ-45-3 Comprehensive Creative Technology Project Research Document' worked on. (06/12/2021 - 12/12/2021) 'Horror with portals' Hyperbolic subproject worked on. (06/12/2021) 'UFCFHQ-45-3 Comprehensive Creative Technology Project Research Document' worked on and completed (06/12/2021 - 10/12/2021) **(C)** 'Horror with portals' Hyperbolic subproject worked on. (10/12/2021) 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on (06/12/2021 & 10/12/2021)	**(06/12/2021)** Choosing between what reading materials are immediately helpful, and which could be considered extended reading, has proven to be difficult but serves an important function. It will lower the list of options for reference when in a tough situation. Specifically developing the hyperbolic portals. These will be moved into the subproject template for all subprojects that utilise hyperbolic portals. **(07/12/2021 - 09/12/2021)** The writing within the document had to be trimmed down in order to achieve a concise point. (07/12/2021) Following the concise writing style, advised by the advisor, the 'Research findings' section was written out. (08/12/2021) Document finished in first draft version. (09/12/2021) There has been an issue with the portal rendering. I am unable to see the rendering through the player camera. I will have to work to fix this in the next session. (10/12/2021)
13/12/2021 - 19/12/2021	UFCFHQ-45-3 Comprehensive Creative Technology Project Research Document' to be ready for submission by	UFCFHQ-45-3 Comprehensive Creative Technology Project Research Document' submitted. (16/12/2021)	**N/A.**

	16/12/2021. **(C)**		
10/01/2022 - 16/01/2022	**N/A.**	Horror with Portals' Hyperbolic subproject worked on. (13/01/2022) 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on (13/01/2022)	It is believed that five subprojects can be completed as groundwork for the subproject that will be further developed, totalling to six. This is due to the portal work currently being one of the heavy support systems for non-Euclidean video game design, alongside spherical-geometry-based gameplay.
17/01/2022 - 23/01/2022	**N/A.**	Development on 'Horror with Portals' subproject. (17/01/2022 - 20/01/2022) Prototype demonstration as 2-min video ready for submission by 24/01/2022 complete. (20/01/2022) **(C)** Initial planning and development started on 'Dissolving floor puzzle' subproject. (21/01/2022) 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on (17/12/2021 - 21/01/2022)	The portal work being achieved in this subproject is able to scale and comfortably work with other subprojects. A basic foundation has been established for a future planned subproject that involves using items within portal-like systems.
24/01/2022 - 30/01/2022	Prototype demonstration as 2-min video ready for submission by 24/01/2022. **(C)**	Presented a demo video and held a 15 minute Q&A to discuss it. (24/01/2022). 'Dissolving Floor Puzzle' subproject worked on and completed. (25/01/2022). 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on. (25/01/2022).	From the Q&A in my demo presentation I learnt that I should include a viewable criteria for how I rate my subprojects, with breakdowns as to why they have been rated what they have been. This will be contained within a new document that will contain all the ratings of all the subprojects.
14/02/2022 - 20/02/2022	Have idea of development for 'Spherical exploration game' subproject by 15/02/2022. **(C)**	Initial planning and development started on 'Spherical exploration' subproject. (15/02/2022). **(C)** 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on (15/02/2022)	As the CCTP progresses it is normal for things to evolve. Later subprojects may be better detailed and planned than previous. This creates a requirement at some point to revisit old subprojects and add information previously left out.
21/02/2022 - 27/02/2022	Finalise development of 'Spherical exploration game' subproject by 26/02/2022. **(C)** Have idea of development for 'Inside sphere explorer' subproject by 27/02/2022. **(C)**	Spherical exploration' subproject development and completion. (22/02/2022 - 25/02/2022). **(C)** Initial planning and development started on 'Inside sphere explorer' subproject. (27/02/2022). **(C)** 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on (22/02/2022 - 25/02/2022 & 27/02/2022)	**N/A.**
28/02/2022 - 06/03/2022	Finalise development of 'Spherical exploration game' subproject by 05/03/2022. **(C)**	'Inside sphere explorer' subproject development and completion. (01/03/2022 - 02/03/2022). **(C)** Initial planning, development and completion on 'Gravity changing scale puzzler' subproject. (02/03/2022 - 03/03/2022). **(C)** All alpha stage subprojects completed. (04/03/2022). **(C)** 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on (01/03/2022 - 04/03/2022) 'UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document' Document worked on. (04/03/2022)	All alpha stage subprojects have finished development this week. Before beginning the development of beta stage subprojects it seems important to first go back over the alpha stage documentation, within the production document, and ensure all sections are written in a concise and consistent manner. Once all the documentation is checked over, planning on the beta stage subprojects will begin. These beta stage subprojects will consist of several of the alpha stage subprojects but built into an actual game environment.

07/03/2022 - 13/03/2022	Finalise development of 'Inside Sphere explorer' subproject by 12/03/2022. **(C)**	'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on (10/03/2022 - 13/03/2022). Initial planning, development and completion on 'Gravitational tactical scaling' subproject. (10/03/2022 - 12/03/2022). **(C)** Meeting with project advisor. (11/03/2022).	The meeting with the project advisor helped clear some worries and get the project back on track. There was an issue of how the final submission will play as it is intended as both a playable game and an interactive showcase of the subprojects. Ultimately it was decided that the player will go through the subprojects in chronological order with context and information shown and played audibly. Once the player reaches the end, there will be a single level to showcase the final subproject. This would form the artefact.
14/03/2022 - 20/03/2022	All alpha stage subprojects finished development by 17/03/2022. **(C)**.	'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on (14/03/2022 - 18/03/2022) Initial planning, development and completion on 'Portal Sphere Shooter' subproject. (15/03/2022 - 17/03/2022). **(C)** All beta stage subprojects completed. (17/03/2022). **(C)** Initial planning started on the final subproject & deliverable artefact. (18/03/2022).	**N/A.**
21/03/2022 - 27/03/2022	Develop and complete the two selected beta stage subprojects by 25/03/2022. **(C)** Create a project plan by 27/03/2022. **(C)**	'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on. **(A)** Development of the final subproject & deliverable artefact. **(A)** Basic project plan created. (22/03/2022). **(C)**	**N/A.**
28/03/2022 - 03/04/2022	**N/A.**	'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on. **(A)** Development of the final subproject & deliverable artefact. **(A)**	**N/A.**
04/04/2022 - 10/04/2022	**N/A.**	'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document' worked on. (04/04/2022 - 05/04/2022) Meeting with project advisor. (05/04/2022). Development, and completion of the final subproject & deliverable artefact. (04/04/2022 - 05/04/2022 & 08/04/2022 - 10/04/2022). **(C)** 'UFCFHQ-45-3 Comprehensive Creative Technology Project Report Document' worked on. (05/04/2022 - 10/04/2022).	With the final subproject and interactive showcase complete, any logged work on it from this point on will be polish and minor additions. The focus can now shift on to the report. Once the report is through its second draft, the primary focus can move on to the video. Hopefully this can all be done to a high level by 22/04/2022. (05/04/2022).
11/04/2022 - 17/04/2022	Complete development on final subproject as artefact by 16/04/2022. **(C)**	UFCFHQ-45-3 Comprehensive Creative Technology Project Report Document' worked on. (15/04/2022 - 17/04/2022).	**N/A.**

18/04/2022 - 24/04/2022	Have report and video complete by 22/04/2022. **(C)**	UFCFHQ-45-3 Comprehensive Creative Technology Project Report Document' worked on and completed. (18/04/2022 - 19/04/2022). **(C)** Project deliverable video recorded, edited, and produced. (19/04/2022 - 20/04/2022). **(C)**	**N/A.**
25/04/2022 - 01/05/2022	7000 word report, artefact, and final 2-min video ready for submission on 28/04/2022. **(C)**	**N/A.**	**N/A.**

Appendix B: Project Timeline
The full document can be found in the "final subproject source code and files" repository linked at the top of the report, within the documentation folder. The document is called 'UFCFHQ-45-3 Comprehensive Creative Technology Project Progress Log.' This shown section is the first page, "Introduction."

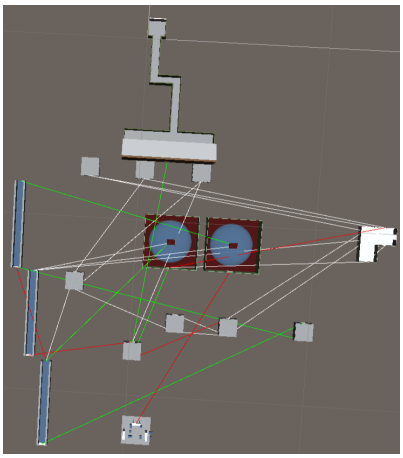
Pre-production Estimate Timeline		
Month	**Task**	**Estimated Duration**
October 2021	Write a project proposal to be submitted by 04/11/2021.	11 days (24/10/2021 - 04/11/2021).
November 2021	Finish 'CCTP Pre-Production & Research Document', Create 'CCTP Production Document', Develop one subproject* (artefact & 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.

Appendix C: Development Journal
The full document can be found in the "final subproject source code and files" repository linked at the top of the report, within the documentation folder. The document is called 'UFCFHQ-45-3 Comprehensive Creative Technology Project Production Document.' Below is a snippet of section "5.2. Development

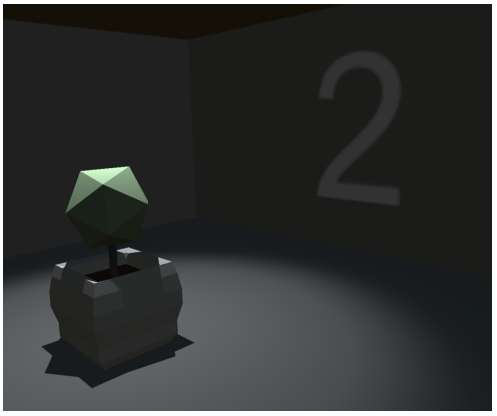
journal.” This shown section discusses part of the development process of the final version of the Horror With Portals alpha-stage subproject and the completion of the dissolving floor subproject.

25/03/2022

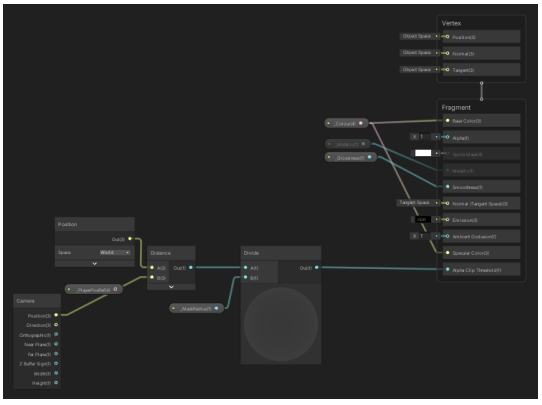
The map was completed with a number of different rooms that have yet to be furnished and numbered. So long as the user selects the correct portal in ‘Room 3’ then they will be able to just walk straight and reach the exit easily. This was done for marking and easy progression reasons.



The drawn lines have their own meanings. White means that the user has been brought into the maze and sent to a room where progression will not occur. Red means that the user has regressed and has been sent to a room that is connected to a progressive portal. Green means the correct path is chosen and progression will occur. Lighting, furniture and room identification was then implemented.

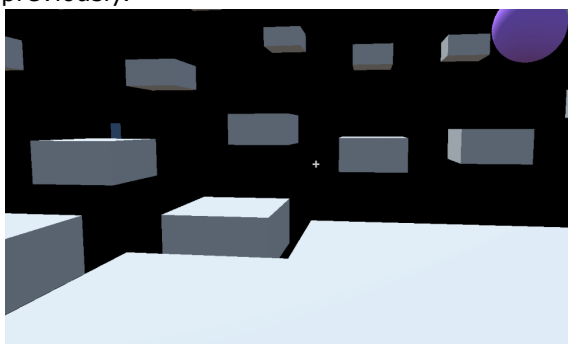


The render pipeline used within the subproject was changed for performance and shader-based reasons. This led to the ‘Inside Sphere’’ shader requiring a change and caused the portal shader to form a couple visual problems. Below is the shader graph for the ‘Inside Sphere’ effect.



26/03/2022
The 'Universal Render Pipeline' package being used within Unity caused a major memory leak in the fixed version of the portal script. This was solved by adding a proximity detector and rendering the cameras through Unity. The lighting also needs to be adjusted within the portal level.

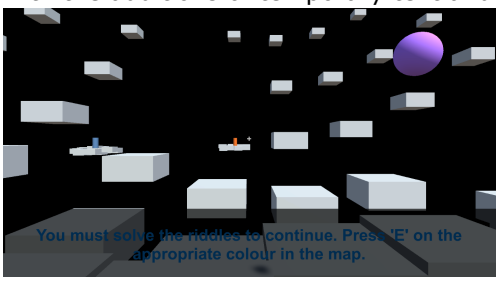
In the 'Dissolving floor' level, the spirit object has received an update to its movement code to allow it to always aim to sit at the top-right of the user's screen unless thrown. This coincided with changes that caused floor creation based on distance to look a lot smoother than it did previously.



More changes were made to allow for the user to remain on an island of land that will reduce in size if the user moves.

Since the dissolving floor effect was created with a puzzle game in mind, a lot of thought went into a puzzle idea that would suit the effect. The end result was that the user should explore to find clues and then using those clues, they find the exit. This also led to a realisation that no horror gameplay elements were added into the horror portal level.

With the additions of temporary text and a win objective, the dissolving level is done.



Appendix D: Assets used in the Project
All assets were created by the project developer specifically for the project. Audio files were made in FL Studio. Model assets were created in Blender.

Appendix E: Pre-Production and research document
The full document can be found in the "final subproject source code and files" repository linked at the top of the report, within the documentation folder. The document is called 'UFCFHQ-45-3 Comprehensive Creative Technology Project Pre-Production & Research Document.' Below is a snippet of section "3.2. Genre selection."

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.

Appendix F: User Teleportation Code
Below is the teleportation code called upon the user passing halfway through a portal. It overrides the usual function called because the user game object has specific components that need to be matched.

```
// Function that handles player teleportation within portals
1 reference
public override void Teleport (Transform inPortal, Transform outPortal, Vector3 teleportPosition, Quaternion
teleportRotation)
{
    // Rotate the player game object to match the outPortal's X & Z rotations. Doesn't work if LateUpdate() lock is being
    called.
    if (outPortal.eulerAngles.x > 1 || outPortal.eulerAngles.z > 1 || outPortal.eulerAngles.x < -1 || outPortal.eulerAngles.z
    < -1)
    {
        Vector3 relativeRot = (outPortal.rotation.eulerAngles * -1) + transform.rotation.eulerAngles; // Get the opposite
        rotation of current rotation
        if (relativeRot.y < 1 && relativeRot.y > -1) relativeRot.y += 180;
        else if (relativeRot.y < -179 && relativeRot.y > -181) relativeRot.y += 180;
        else if (relativeRot.y < 181 && relativeRot.y > 179) relativeRot.y += 180;
        Debug.Log (relativeRot);

        Vector3 cameraRot = (outPortal.rotation.eulerAngles * -1) + playerChild.transform.rotation.eulerAngles; // Get the
        opposite rotation of current rotation
        Quaternion teleRot = outPortal.rotation * Quaternion.Euler (cameraRot); // Establish rotation variable for correct
        way to face after teleportation

        float shortestDistance = Mathf.DeltaAngle (cameraPanSmooth, teleRot.y); // Calculate the shortest distance between
        player's camera rotation and desired teleportation rotation

        cameraPan += shortestDistance; // Establish correct rotation for left & right camera movement

        cameraPanSmooth += shortestDistance; // Correct Yaw Smooth with calculated shortest distance to allow for continuity

        playerRigidbody.transform.rotation = Quaternion.Euler (relativeRot);
        playerChild.transform.rotation = Quaternion.Euler (transform.up * cameraPanSmooth); // Set player rotation to correct
        rotation. It should match the implied direction before entering portals
    }
    else
    {
        Vector3 eulerRotation = teleportRotation.eulerAngles; // Create a Vector3 of quaternion for transform calculation

        float shortestDistance = Mathf.DeltaAngle (cameraPanSmooth, eulerRotation.y); // Calculate the shortest distance
        between player's camera rotation and desired teleportation rotation

        Vector3 correctRotation = new Vector3 (0, 0, 0);
        transform.rotation = Quaternion.Euler (correctRotation);

        cameraPan += shortestDistance; // Establish correct rotation for left & right camera movement

        cameraPanSmooth += shortestDistance; // Correct Yaw Smooth with calculated shortest distance to allow for continuity

        playerChild.transform.rotation = Quaternion.Euler (transform.up * cameraPanSmooth); // Set player rotation to correct
        rotation. It should match the implied direction before entering portals
    }

    velocity = outPortal.TransformVector (inPortal.InverseTransformVector (-velocity)); // Move player off the dotProduct
    (mid point) of the portal and match velocity of entering

    playerRigidbody.transform.position = teleportPosition; // Set player position to the calculated teleport location

    Physics.SyncTransforms(); // Sync physics to stop drifting
}
```

Appendix G: PlayerController.cs Variables
Below are the variables used within the user controller script. These were used in tandem with an editor script that helps keep the Unity Editor's Inspector clean when working. This can be found within the source code of the final subproject and interactive showcase.

```
    // Character Variables
    private Rigidbody playerRigidbody;
    private CapsuleCollider playerCollider;

    // Camera Variables
    Camera playerCamera;

    [Header ("Camera Movement")]
    [Tooltip ("Set whether the Cursor is hidden or shown.")]
    public bool lockCursor = false;
    public float mouseSensitivity = 10.0f;

    [Header ("Camera Controls")]
    [Tooltip ("Length of camera movement smoothing. Lower values = sharper stops. 0.1f
offers a realistic feel.")]
    [Range (0.0f, 0.4f)]
    [SerializeField] private float cameraRotationSmoothTime = 0.1f;
    private float cameraPan; // Looking left and right
    private float cameraPanSmooth; // For smoothing the pan movement
    private float cameraPanSmoothVelocity; // Pan smoothing speed
    private float cameraTilt; // Looking up and down
    private float cameraTiltSmooth; // For smoothing the tilt movement
    private float cameraTiltSmoothVelocity; // Tilt smoothing speed
    [Tooltip ("Control the camera tilt range. X = Up. Y = Down. +-40 = A good range.")]
    [SerializeField] private Vector2 cameraTiltRange = new Vector2 (-40.0f, 40.0f); //
Control how far player can look (up, down)

    [Header ("Sound Options")]
    public bool useSound;
    public AudioSource playFootsteps;
    public AudioClip[] walkingSounds;
    public AudioClip[] sprintingSounds;
    public float footstepVolume = 0.05f;
    private float stepTimer;

    // Movement Variables
    private Vector3 moveDirection;
    [Header ("Player Movement")]
    [Tooltip ("Walking speed. 5.0f feels good for shooter-like movement.")]
    [Range (2, 8)] public float walkSpeed = 5.0f;
    [Tooltip ("Sprinting speed. Usually 1.5x faster than walking speed for smooth
movement change.")]
    [Range (4, 10)] public float sprintSpeed = 7.5f;
    [Tooltip ("Jump height. 10.0f feels good for arcade-like jump.")]
    private float currentSpeed; // For determining our speed in code
    [Tooltip ("Smooths player movement. Lower values = sharper stops. 0.1f feels
cinematic.")]
    [Range (0.0f, 0.4f)]
    public float movementSmoothTime = 0.1f;
    [Tooltip ("Jump height. 7.5f feels good for arcade-like jumping (10.0f gravity).
10.0 for realistic jumping (20.0f gravity)")]
    public float jumpForce = 10.0f;
    public bool allowJumping = true;
    private bool jumping;
    [Tooltip ("Amount of gravity. 10.0f feels good for arcade-like gravity. 20.0f for
realistic gravity.")]
    public float gravityForce = 20.0f;
    private float fallingVelocity = 0.0f; // Keep track of falling speed
    private float yCollisionBounds = 0.0f; // Variable used in raycast to check if
grounded
    private float lastGroundedTime = 0.0f; // Keep track of when last grounded
    private Vector3 velocity;
    private Vector3 currentVelocity;
```

```
    private bool sprinting;
    private bool moving;

    // Dissolving Floor Variables
    [HideInInspector] public bool dissolvingFloor;

    // Spherical Variables
    [HideInInspector] public bool sphericalMovement;
    [HideInInspector] public GameObject planetGameObject;
    [HideInInspector] public bool isModel;
    private Quaternion panRotation;
    private Quaternion playerToPlanetRotation;
    private GameObject playerChild; // Child object of the player

    // Wall-Walk Variables
    [HideInInspector] public bool wallWalk; // Enable wall walking
    [HideInInspector] public float gravityRotationSpeed = 4.0f; // How quickly should
the player rotate
    [HideInInspector] public float wallWalkDetection = 1.5f; // How long should the
raycast be
    [HideInInspector] public LayerMask groundLayers; // What layers are floor objects
set as
    private Vector3 groundDirection; // What direction is the ground
    private bool wallWalkRotate;

    // Inner-Sphere Variables
    [HideInInspector] public bool insideSphere;
    [HideInInspector] public float insideSphereRadius = 5.0f;

    // Shooting Variables
    [HideInInspector] public bool enableShooting;
    [HideInInspector] public Rigidbody projectileRigidbody;
    [HideInInspector] public float projectileSpeed = 20;

    // Spherical world variables
    [HideInInspector] public bool sphericalWorld;
    [HideInInspector] public float sphereXAxis;
    [HideInInspector] public float sphereYAxis;
```