How easy is it for players to complete a navigational task within a Non-Euclidean virtual environment?

Oliver Toby Tennant

1 Proposal

Within the games industry, there is a concept called Euclidean and Non-Euclidean geometry. Euclidean geometry is used in the study of flat surfaces. Meaning, a straight line is the shortest possible line between two points. Whereas Non-Euclidean geometry uses curved surfaces or spherical shapes. Which then allows for the measurement of the shortest distance between two points whilst also using a spherical object [2]. An example of a game that does this technique is "**Portal**" and how it creates the look of its dual portal system. It gives the player the ability to places down two portals at two different points of the level and allows them to see through the primary portal and see the surroundings of the secondary portal whilst being at the angle of the player's perspective. Another example of how it can be utilised can be seen in the game "**Antichamber**" with how it generates looping corridors, portals and having multiple scenes existing in the same space through the use of stencil buffers. I'll explain their purpose later in the paper. But for now, I shall point your focus more toward the more psychological effects that Non-Euclidean geometry may provoke and how these games make the average user feel when playing them.
Referring back to **Antichamber**, this game most certainly toys with the idea of generating abstract ways of making the player feel uneasy and disorientated. But also gives the player a sense of wonder when discovering the mechanic of the puzzle as they literally make you think outside (or inside?) the box.

2 Research Question

I intend on this paper to delve further into Non-Euclidean techniques and levels of Agency that the user possesses when they are given these such aspects. I'm also considering that the data acquired from the experiment be used to help in the production of other applications be that, Virtual Reality or in other games that wish to use Non-Euclidean elements. So with the experiment, I aim to test two hypotheses:

2.1 Hypotheses

1. There are a set amount of tasks that the user can complete before they become disorientated;
2. Users that have less experience with games will perform more efficiently in the experiment compared to those that do;
These hypotheses were tested in the experiment using the **Anova** method to see if there were a substantial difference between the vast amount of players. Participants would each be tested on their efficiency within each of the different Non-Euclidean levels. The way to prove these hypotheses would be through the results on the pre-test and the post-test questionnaires. This data would then be used to form a conclusion on whether these hypotheses had been proven or disproven.

3 Background

3.1 Metric Cognitive Map Testing

When looking at the more psychological effects with using Non-Euclidean geometry, there has only been a very small handful of academic research tests that had been conducted, however not by using a video game as a ground to work on. Within the paper "Journal of Experimental Navigation: Non-Euclidean navigation" [5], it is mentioned that they conducted a test which involved an assortment of rats and a maze that was constructed on an elevated, spherical surface. This test were to gather behavioural evidence for **metric cognitive maps**.

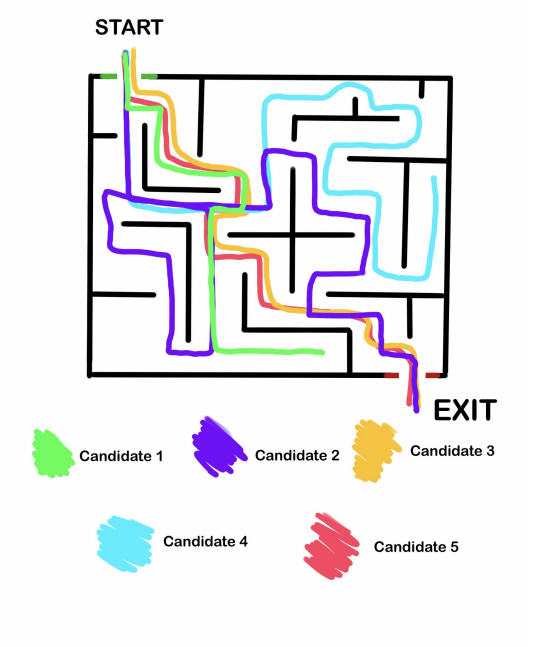

Figure 1: Shows a cognitive map that represents a test conducted on 5 individuals and their progression through a simple maze. 3 of which made it to the end and the other 2 failed.

The aim of the experiment were to train the rats to go a certain path to acquire a food package. Then, when the rats memorised the path, they would change the scenario in a way that corresponds to the previous layout but blocks off some of the original paths. Not only this, the end goal had been lit with a lamp to act as a beacon to give the rats a sense of direction. After the experiment concluded, it was recorded that only 36 percent of the test subjects subconsciously went towards the familiar path and couldn't acquire the food. Which didn't really prove the theory of metric cognitive mapping, but instead showed more signs of stimulus response learning.

A similar experiment was later conducted in humans and had proved that humans possess the ability to point out key features of familiar or previously traversed terrain. Though, it is rather an estimate and does have a tendency to be incorrect. As there is a bracket of **20-100 deg** in which the person could potentially deviate from their path. Resulting in the person unintentionally missing their end goal.

3.2 Stencil Shader Buffers

Stencil Buffering is a technique used to create a mask on an object that will completely disregard pixels and only show what is necessary. The stencil can be used as an alpha mask on the face of a shape. The example given on the web page [1] describes it being similar to looking through a window and seeing a mountainscape. What it does it is basically hides everything that is outside the view of the window by the marking pixels between the frame with an arbitrary value of 1. This value will be used to signify that anything marked with the value of 1 within the frame will be kept and anything else will be disregarded.

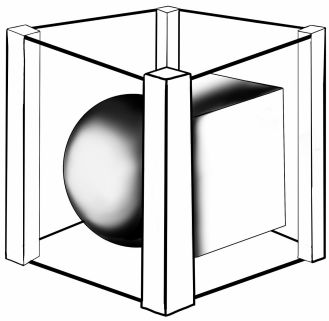

Figure 2: A visual representation of a Stencil Buffer working on a cube object.

With the artefact being created in the Unity Game Engine and being coded in the C language, looking for implementation techniques is actually rather simple as Unity comes with a plethora of documentations that help in achieving multiple different effects. The one document that will be focused on is heavily is the **ShaderLabs: Stencil** [4]. This is due to the fact that it explains what procedures need to be taken in order to get the desired effect.

4 Artefact

4.1 What is going to be created?

The intended artefact that I'm creating will consist of a linear-esque level design where the player is tasked to pick up an object and place it onto a button at another point within the level. However, the more the player progresses through each of the levels, the more Non-Euclidean elements are revealed with the intent on slowing their progression. Some of the Non-Euclidean aspects I intend on developing will have near similarities to the method in which the Portal games use their seamless transitions through each of the portals. However instead of creating a small surface that the player can see visually, I intend on having these transitions placed within the game's

environment at all times that will not be noticeable until the player has a moment of realisation. This can be done within Unity rather simply. As it uses two separate cameras (let's say cameras **A** and **B**) and places them so they are looking towards a screen at two separate points within the level. Both of these cameras will orbit their screen as they essentially replicate the player's camera movements. Camera **A** will project the perspective of camera **B** onto a screen that is situated with **A** to get the image of the opposite side. As the player moves and looks through the screen, they should see the current perspective of camera **B**. (To see how this process works in the code, it can be seen in figures 10, 11, 12 and 13.)

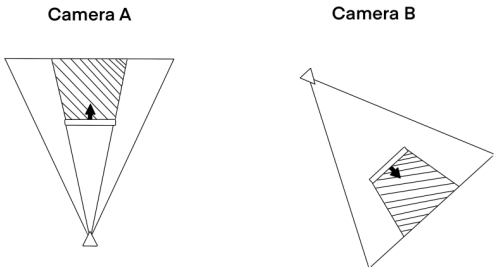

Figure 3: Shows the positions of cameras A and B.

Using this method of simple camera trickery will allow me to replicate the idea of looping corridors, infinitely ascending/descending staircases, portals and much more. To be able to get a thorough test that works as intended, I'll need to use the Non-Euclidean formula, or as it is better known as the Hyperbolic Geometry formula, which can be viewed here [6].

4.2 How is it going to be created?

I aim to use the **Prototyping Life Cycle**[3] as I felt as though the Agile methodology would work substantially well. This is due to the fact that I will be ensuring a consistent level of work is being completed over the duration of a 16 week period. This allows for a game level to be created every 2 weeks as part of a sprint plan I'm devising. Furthermore, within this time I will be running QA tests frequently to ensure that the Non-Euclidean geometry is working as intended and gives me more of an idea on how the project can be improved or changed in anyway.

4.3 Test Plan

I aim for my experiment to take these feelings of confusion to test a user's efficiency within a simple navigational task. I will begin the experiment by handing them a questionnaire which will ask if they've ever had any experience with games before and to see their overall experience with games. Then after they've finished the questionnaire and the task has been completed, I'll hand them a second questionnaire which will ask how they found the experiment. On completion, they will then be asked to go through a different version of the same level but with more Non-Euclidean elements and then complete a third questionnaire. Again, asking how they found the test.

5 Research Methodology Acknowledgements

5.1 Data Collection

To gather the data needed for this experiment, the aforementioned questionnaire will be the primary source of this research as I have tailored all of the questions to be relevant towards my hypotheses. I have not included anything that is personal to the participants as the research does not require them.

5.2 Storing Collected Data

The data will be stored within a simple Excel spreadsheet document as it keeps everything together and makes it easier to calculate the percentages when the experiment has concluded. This data will then be used to make a conclusion between the two hypotheses in the form of a scatter graph that shows the amount of time played along with the efficiency of the user. I will then run this data through **R Code** using **NUnit** and **RStudio** to get a fully analysed set of data. For the type of test I wanted to run, I chose the **Anova Test** as it deals with multiple participants and finds out if the results are significant. So in this case, I wanted to see the correlation between those who don't play games compared to those that do to a varying degree. These results can be seen within figures 6 and 7.

5.3 Ethical Considerations

With the experiment being a psychological test and the safety of the participants and faculty being the most high priority. I have taken consideration of numerous risks and have thought of ways that

can reduce the level of likely-hood towards both parties. The following subsections will address these and explain how they will be followed within the experiment.

5.3.1 Nausea and Anxiety

With the experiment dealing with the disorientation of participants and how they process the Non-Euclidean elements. My main worry is that the candidate may be prone to nausea and the test might induce anxiety. I aim to alleviate some of these feelings by giving the participant an in depth explanation on Non-Euclidean geometry beforehand and allow for them to leave if they do not feel comfortable with participating.

5.3.2 Illness Associated with Consistent Computer Use

A common form of illness that may occur when using a computer for an extended period of time is either RSI or eye straining. Either of which are deemed a main cause for concern when conducting the experiment as eye straining can lead to visual impairment and high levels of pain where as RSI may possibly lead to a posture deformity. I hope to alleviate this by giving the candidate a brief moment to take a break in which they can recuperate themselves and come back to the test and continue. Another cause for concern is the matter of headaches or migraines. To combat this, I'm allowing for the candidate to withdraw from the experiment if they are starting to feel these such sensations. A member of staff will also be around if they require any immediate help.

5.3.3 Withdrawal from the Experiment

As mentioned previously, the candidate is free to leave the experiment at anytime without consequence if they feel uncomfortable at anytime during the study.

5.3.4 Covid-19 Saftey

Covid-19 is still being considered a threat at the current moment. Precautionary measures will be taken very seriously when running the experiment to avoid any infections towards the candidates and those in close proximity to the research station. So to make sure everything runs perfectly:

- Face coverings must be worn at all times;
- A risk assessment of the equipment will be completed;
- Social distancing will be enforced before anything proceeds;
- The equipment will be sanitised after every use.

6 Results and Data

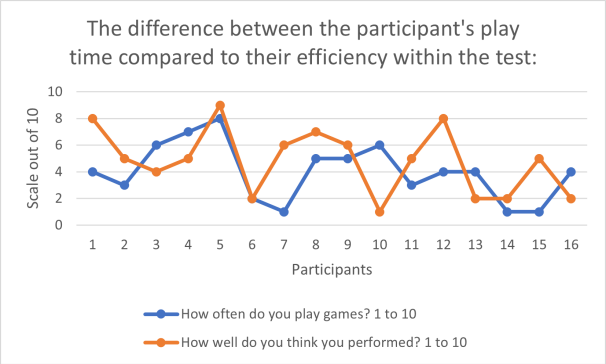

Figure 4: This graph shows the difference between the 16 participants who took part within the test.

To illustrate the difference in each of the results, they have been placed into this scatter graph that shows both of the results for the amount of time they play games weekly and how well they thought they did on the test. Another thing to note is that both of the questions were asked within a scale out 1 to 10. In regards to the user's efficiency factor, the scale was set at 1 being a very low score, 10 being the highest that they could give themselves.

Amount of time playing games on average.	Scale from 1-10.
Not at all	1
Under an Hour	2
1hr	3
2hrs	4
3hrs	5
5hrs	6
7hrs	7
10hrs	8
12hrs	9
15hrs+	10

Figure 5: Table for reference.

Sadly, I couldn't get more than 16 participants as the Covid restrictions in place at the time of me conducting the experiment made gathering data and volunteers a rather difficult task. Nevertheless, the amount of people who took part within the experiment gave rather correlated data. Which was fortunate in getting an answer for the hypotheses.

7 Discussion

As you can see from the graph shown in **figure 4**, it's evident that out of the 16 participants, 3/4 of the group said that they didn't play video games for over 3 hours. Which showed to be rather interesting at the time. Then after the experiment had been concluded, only 6 gave themselves an efficiency ranking of over 5. To go further in depth, 5 out of these 6 people said that they only play games for an average of 1 to 3 hours per week. Whereas the remaining person said that they only play video games for around 10 hours on average.
These results suggest to me that out of the majority of those that had a lower playtime, a third of those performed better within the test. Comparing this to the higher scoring participants, only one of those people gave themselves a higher score.

8 Conclusion

So with the results gathered and analysed, it's clear to me that the data shown does prove my hypotheses but it can't be completely finite due to the heavy bias towards the lower playtime participants as they're the majority within the experiment. However, there is quite a clear definition with how well they did perform. One thing to consider is that the higher playtime users may have potentially missed obvious ques that may have impeded their progress in some way. This could be due to the fact that gamers in general typically share a linear-esque state of mind whilst playing video games. A prime example would be games in the first-person shooter genre as they're mainly linear in the way the player would progress through the story. So with this in mind, I believe that non-gamers or low playtime gamers are more inclined to think outside the box whilst they play which would allow them to potentially have a more versatile play style as they don't have this restraint of thinking linearly.

8.1 R Code

```
> anovaTukeyHsdOutput
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Non_Euclidean_Questionnaire$Performance ~ as.factor(Non_Euclidean_Questionnaire$Time))

$`as.factor(Non_Euclidean_Questionnaire$Time)`
          diff        lwr       upr     p adj
2-1 -2.3333333 -13.718348  9.051681 0.9872078
3-1  0.6666667  -8.333977  9.667311 0.9999792
4-1  0.6666667  -6.863812  8.197146 0.9999311
5-1  2.1666667  -6.833977 11.167311 0.9699878
6-1 -1.8333333 -10.833977  7.167311 0.9876355
7-1  0.6666667 -10.718348 12.051681 0.9999958
8-1  4.6666667  -6.718348 16.051681 0.7297201
3-2  3.0000000  -9.075631 15.075631 0.9648350
4-2  3.0000000  -8.023493 14.023493 0.9453785
5-2  4.5000000  -7.575631 16.575631 0.8020547
6-2  0.5000000 -11.575631 12.575631 0.9999996
7-2  3.0000000 -10.943738 16.943738 0.9833288
8-2  7.0000000  -6.943738 20.943738 0.5391670
4-3  0.0000000  -8.538761  8.538761 1.0000000
5-3  1.5000000  -8.359712 11.359712 0.9977513
6-3 -2.5000000 -12.359712  7.359712 0.9611004
7-3  0.0000000 -12.075631 12.075631 1.0000000
8-3  4.0000000  -8.075631 16.075631 0.8721948
5-4  1.5000000  -7.038761 10.038761 0.9946787
6-4 -2.5000000 -11.038761  6.038761 0.9238131
7-4  0.0000000 -11.023493 11.023493 1.0000000
8-4  4.0000000  -7.023493 15.023493 0.8197949
6-5 -4.0000000 -13.859712  5.859712 0.7381910
7-5 -1.5000000 -13.575631 10.575631 0.9993675
8-5  2.5000000  -9.575631 14.575631 0.9864769
7-6  2.5000000  -9.575631 14.575631 0.9864769
8-6  6.5000000  -5.575631 18.575631 0.4680483
8-7  4.0000000  -9.943738 17.943738 0.9304416

> summary( anovaOutput )
                                              Df Sum Sq Mean Sq F value Pr(>F)
as.factor(Non_Euclidean_Questionnaire$Time)  7  42.77   6.110   0.984  0.502
Residuals                                    8  49.67   6.208
```

Figure 6: Shows the participants in an Anova test.

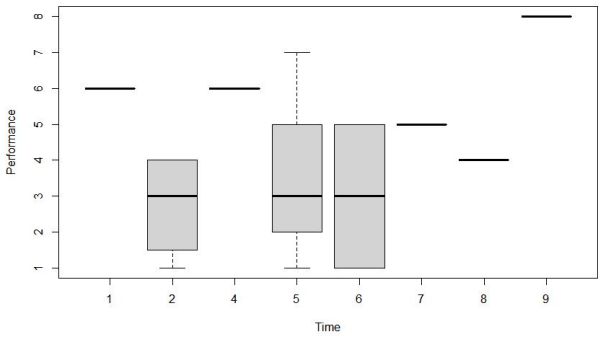

Figure 7: Shows the participant's data in a boxplot graph.

9 Critical Addendum

9.1 What went well?

When I first started this project, I felt quite alien to the Non-Euclidean side of the geometry. But after delving deep into how these systems work and seeing how I could potentially achieve some of these aspects on my own gave me the inspiration to carry out this project. So during the course of the project, I would quite like to admit that I really enjoyed developing the portal system and achieving the "mirror into another world" kind of look I was going for, albeit not to my expectation. Nevertheless, I feel as though the final version of my portal system did attain the visuals and some of the functionality that I'd hoped for when starting this project.

9.2 What didn't go well?

As I started working on this project, there were quite a plethora of problems that started to arise. Some were out of my control and the rest could've been managed a lot better. For this portion of the the critical addendum, I'd like to take some time to reflect on some the issues I faced during the development cycle.

Firstly, as mentioned previously, COVID-19 was one of the main causes for concern as it's still quite a global problem that did host many situations that couldn't be avoided. One of these such issues include a hardware issues I had which caused my machine to randomly turn off due to a faulty power supply. This problem couldn't be resolved for quite some time as I couldn't seek the professional support I needed as we were forced into lockdown. This did get resolved however, but at the cost of two months work. Also during this time, I also faced mental health issues that lowered the motivation I had for the project. But after speaking to my personal tutor and my fellow peers about the current occurrences. It certainly helped in getting the project back on track.

Another problem I ran into was that the pipeline version of Unity I was using (the Universal Rendering Pipeline) didn't support the rendering materials I needed for the portal screens. So a work around had to be put in place that got the desired effect.

9.3 How could it be done differently in the future?

If I were to take this project again, I certainly would change a considerable amount of factors in the project that may certainly help in getting more desirable figures for the data and having a more polished final product.

I'd personally like to start off with how I structured my time during the development cycle as I had other projects to work on that also needed quite a lot of attention. To combat this, I think I would've made the weekly HacknPlan updates a lot more compulsory as I ended up just forgetting to use it after the early weeks of development which definitely skewered the final product in the long run. Either this, or finding an alternative Agile method that does require much more attention. My personal tutor did mention a daily task notification software that they use on the daily to help keep structure to their work schedule.

On another note, another item I'm keen on improving would be that I often had a tendency to get stuck on a particular task and tended to avoid asking for help when needed. So, I think this would be a perfect opportunity to ask my fellow peers or personal tutor how they would deal with the issue at hand and not to be afraid of criticism when necessary. I believe that if I asked for help at the time of the problem, some of the issues present would've been resolved sooner.

10 Computer Artefact Addendum

To run my unit test for the artefact, I used Unity's built in unit testing framework which shows substantial results as it shows in figure 7 and 8. The reason for me choosing this over any other framework would be it's ease of use and that it didn't require any external source.
For the test shown, I wanted to ensure that the "Portkey" script passed as this script caused quite an abundance of issues during development. It would randomly either let the player pass through the portal's rendering screen or just work as intended.

```
namespace Test
{
    public class Portkey
    {
        [Test]
        public void PortkeyWithPlayerTest()
        {
            Vector3 playerPos = new Vector3(0, 0, 0);
            Vector3 targetPos = new Vector3(0, 0, 1);

            float portalToPlayer = 5f;
            float dotProduct = -0.3f;

            if (dotProduct < 0f)
            {
                playerPos = targetPos;
            }

            Assert.That(playerPos, Is.EqualTo(targetPos));
        }
    }
}
```

Figure 8: Unit test of the Portkey script which is being used to make sure the player's rotation and position are aligned with the opposite portal correctly.

On a side note, I feel running these test is highly beneficial to the user as it can root out bugs incredibly quickly as appose to hunting them individually. After trying out and using unit testing for this artefact. I can definitely say that I will be using this in future projects.

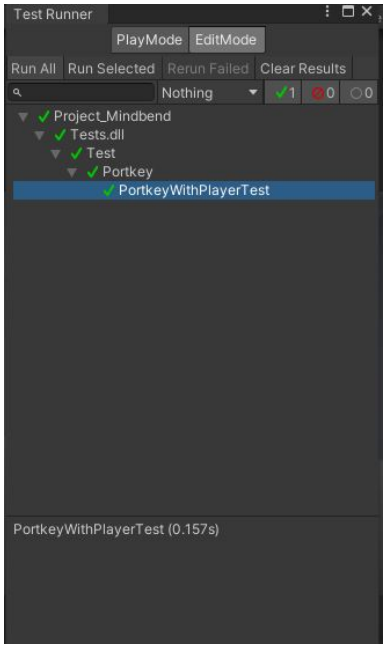

Figure 9: Evidence of the test passing.

11 GitHub Repository

https://github.com/Lieu-Tennie/Non-Euclidean-Virtual-Environment-Artefact

References

[1] David Gavilan. *The Stencil Buffer and how to use it to visualize volume intersections*. URL: https://tech.metail.com/the-stencil-buffer-and-how-to-use-it-to-visualize-volume-intersections/. (accessed: 14.12.2020).

[2] A. Glassner. "Going the distance [2D geometry]". In: *IEEE Computer Graphics and Applications* 17.1 (1997), pp. 78–84. DOI: 10.1109/38.576861.

[3] P. Isaias T. Issa. *Information System Development Life Cycle Model*. URL: http://www.Springer.com/987-1-4614-925395. (accessed: 14.07.2020).

[4] Unity. *ShaderLab: Stencil*. URL: https://docs.unity3d.com/Manual/SL-Stencil.html. (accessed: 14.12.2020).

[5] William H. Warren. *Journal of Experimental Navigation: Non-Euclidean navigation*. URL: https://jeb.biologists.org/content/222/Suppl_1/jeb187971. (accessed: 01.09.2020).

[6] Eric W. Weisstein. *Hyperbolic Geometry*. URL: https://mathworld.wolfram.com/HyperbolicGeometry.html. (accessed: 10.12.2020).

12 Code Snippets

```
void Update ()
{
    Vector3 playerOffsetFromPortal = playerCamera.position +- otherPortal.position;
    transform.position = portal.position + playerOffsetFromPortal;

    float angularDifferenceBetweenPortalRotations = Quaternion.Angle(portal.rotation, otherPortal.rotation);

    Quaternion portalRotationDifference = Quaternion.AngleAxis(angularDifferenceBetweenPortalRotations, Vector3.up);
    Vector3 newCameraDirection = portalRotationDifference * playerCamera.forward;
    transform.rotation = Quaternion.LookRotation(newCameraDirection, Vector3.up);
}
```

Figure 10: This will port the player to the corresponding portal with the same rotation and positioning as when they went through the entry portal.

```
// This is called before the player camera has been rendered
public void Render()
{
    screen.enabled = false;
    CreateViewTexture();

    //Make portal cam position and rotation the same relative to this portal as the player cam is relative to the linked portal
    var m = transform.localToWorldMatrix * linkedPortal.transform.worldToLocalMatrix * playerCam.transform.localToWorldMatrix;
    portalCam.transform.SetPositionAndRotation(m.GetColumn(3), m.rotation);

    //Render the camera
    portalCam.Render();

    screen.enabled = true;
}
```

Figure 11: Renders the linked portal's camera view and rotation onto the entry portal screen and vice versa. This also manages how the camera orbits the portals.

```
void CreateViewTexture()
{
    if (viewTexture == null || viewTexture.width != Screen.width || viewTexture.height != Screen.height)
    {
        if (viewTexture != null)
        {
            viewTexture.Release();
        }
        viewTexture = new RenderTexture(Screen.width, Screen.height, 0);
        // Renders the view from the portal camera to the view of the texture
        portalCam.targetTexture = viewTexture;
        //Displays the view texture of the linked portal onto the screen
        linkedPortal.screen.material.SetTexture("_MainTex", viewTexture);
    }
}
```

Figure 12: This will create a texture on what the camera sees to place onto the screen. This function will then be referenced within the Camera Render.

```
    void Update()
    {
        if (playerIsOverlapping == true)
        {
            Vector3 portalToPlayer = player.transform.position - transform.position;
            float dotProduct = Vector3.Dot(transform.forward, portalToPlayer);

            // If this is true: The player has moved across the portal
            if (dotProduct < 0f)
            {
                // Teleport
                float rotationDiff = -Quaternion.Angle(transform.rotation, reciever.rotation);
                rotationDiff += 180;
                player.transform.Rotate(Vector3.forward, rotationDiff);

                Vector3 stepOffset = Quaternion.Euler(0f, rotationDiff, 0f) * portalToPlayer;
                player.transform.position = reciever.position + stepOffset;

                playerIsOverlapping = false;
            }
        }
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.tag == "Player")
        {
            playerIsOverlapping = true;
        }
    }

    void OnTriggerExit(Collider other)
    {
        if (other.tag == "Player")
        {
            playerIsOverlapping = false;
        }
    }
}
```

Figure 13: This makes it so the player will be teleported to the exact point on the receiving portal through which they entered.