Non-Euclidean Video Games: Exploring Player Perceptions and Experiences inside Impossible Spaces

Daniil Osudin, Alena Denisova and Christopher Child

***Abstract*—Non-Euclidean geometry has the potential to be used for novel interactions in video games and create virtual spaces that are not physically possible in the real world. To explore how players perceive and experience them in video games, we have adapted two well-known 2D games, *Snake* and *Asteroids* to create two versions in addition to the conventional virtual space – with hyperbolic and spherical environments – and conducted a within-subject design user study on all three versions of these games. The results show that experienced Mastery and Control are lower when playing the two non-Euclidean versions while perceived Immersion and Challenge do not differ significantly between these three conditions. We also report on the qualitative findings from our participants, which provide further insights into the perception and experiences of these environments.**

***Index Terms*—Non-Euclidean Geometry, Video Games, Player Experience, Impossible Spaces**

I. Introduction

Video game engines streamline the complex process of video game development by shifting the focus away from the components, which are made decoupled and reusable, to the code and assets specific to the game being developed. A game engine consists of a range of different features and often includes a physics engine that simulates the complex physical characteristics of a virtual world, such as soft body dynamics (e.g. cloth modelling), rigid body dynamics (e.g. collision detection) and fluid dynamics. These are mostly modelled on and simulate the properties of the real world.

In recent years, however, some attempts have been made to explore physics and virtual spaces that are not modelled on the real world, including using non-Euclidean geometry to create 'impossible' spaces and novel game mechanics. A non-Euclidean geometry encompasses any geometry that arises from either changing the parallel postulate (Euclid's fifth postulate) or the metric requirement. This article will be focusing solely on traditional non-Euclidean 2D geometries: Spherical and Hyperbolic geometry, illustrated in Figure 1.

In *Spherical* geometry, all geodesics (straight lines in a non-planar space) intersect, so there are no parallel lines. Even if the lines start parallel, they do not preserve the same distance along their length and, instead, appear to 'bend' towards each other. In *Hyperbolic* geometry, any line can have an infinite number of parallel lines, as the lines appear to bend away from each other.

Daniil Osudin and Christopher Child are with the Department of Computer Science, City, University of London, UK, EC1V 0HB.

Alena Denisova is with the Department of Computer Science, the University of York, YO10 5GH, UK (email: alena.denisova@york.ac.uk).

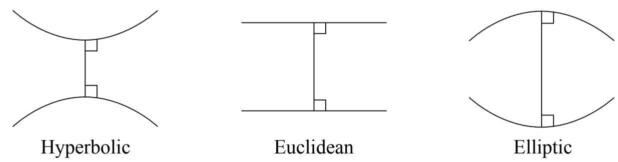

Fig. 1. Comparison of parallel lines in 2D spaces of Hyperbolic (left), Euclidean (middle), Elliptic/Spherical (right) curvature.

Implementing non-Euclidean geometry elements within a video game could present interesting, new challenges to the player that Euclidean spaces are not capable of. Despite the evident benefits of such an approach, non-Euclidean geometry has not been explored extensively in game design and no standalone game engines, to our knowledge, use non-Euclidean geometry to model virtual space and physics.

In this paper, we report on the creation of Spherical and Hyperbolic spaces in two classic 2D video games, *Snake* and *Asteroids* and a user study to explore players' experiences and perceptions of such spaces in games with different core mechanics. To create non-Euclidean versions of the games, we used bespoke software capable of rendering 2D non-Euclidean geometry in real time. This software is created using a method described in [1] for calculating the position and movement of shapes and their vertices in polar coordinates using spherical or hyperbolic trigonometry. This software can calculate and render arbitrary shapes in curved space in real-time. It is also possible to define the parameters of the objects, allow for object interactions, create physical environments in a curved space and render curved shapes on screen using a projection. This study has been undertaken to research the interest in the features developed in the described software. Additionally, the aim was to gauge how intuitive and easy-to-use the non-Euclidean versions of the games are compared to the regular versions of those games.

Our contributions are as follows:

1) We report on the creation of bespoke software for simulating and rendering 2D non-Euclidean (curved) environments and the adaptation of *Snake* and *Asteroids* to operate in a non-Euclidean environment using the created software.
2) We present the findings from a user study which, for the first time, explores how players perceive games in non-Euclidean environments and evaluated player experience when interacting with the two games in curved spaces as opposed to a Euclidean space.

3) We provide a discussion of these types of games and players' experiences of playing them and highlighting some potential areas for future work in this domain.

II. Related Work

Several recent video games have been described as non-Euclidean either by the players/reviewers of these games or their creators largely because the geometry of the game environment appears to be 'impossible' when compared to the real world. Some better-known commercial examples of these types of games are *Fragments of Euclid* [2] and *Manifold garden* [3]. Most of these types of games, however, were not created using non-Euclidean geometry to create these 'impossible' spaces. Instead, they use Euclidean manifolds – a technique that 'glues' together certain fragments cut out of Euclidean space in a non-standard way. To the naked eye, the transition between these fragments appears to be seamless, resulting in an optical illusion of an 'impossible' space.

In a Euclidean environment, if one walks forward 5m, turns 90 degrees right, and repeats these two steps three more times, they will return to the original point and orientation. While in a manifold, it is possible to end up at a different point. Such an approach can be particularly useful for virtual reality (VR) experiences, where the player is constrained by the limitations of the physical space around them, but can travel long distances inside a large virtual world thanks to the manifold of overlapping spaces. One example of such an experience is the VR demo *Tea for God* [4]. Research into non-Euclidean immersive environments began in the 1990s [5] and the model used has been developed by Charlie Gunn and Mark Phillips in 1992 [6]. Research in this area is still ongoing, for example, a recent VR video game, *Non-Euclidean Billiards in VR* [7], can be played in 3D spherical, Euclidean and hyperbolic spaces. Non-Euclidean ray tracing has been proposed as an alternative approach for immersive visualisation of non-Euclidean spaces [8].

The approach of using manifolds changes the topology of the space, but it does not change its geometry. In a truly non-Euclidean environment, however, the geometry of the virtual world is created programmatically. This approach in creating 'impossible' spaces is, however, much less prevalent in game development and, therefore, there are only a limited number of examples of games which use this geometry. Just to name a few, *Warped Mines* is a non-Euclidean version of the Minesweeper [9]; *Hyperbolica* is a hyperbolic 'walking simulator' with mini-games in hyperbolic and spherical geometry [10], and one of the oldest commercial examples is *Hyperrogue* – is a turn-based rogue-like exploration game [11]. The latter game uses a hyperbolic plane, making the virtual world in this game larger than *No Man's Sky* [12], *Minecraft* [13], or any other examples of procedurally generated games created using Euclidean geometry.

These types of games and applications, however, have value not only from the educational viewpoint (learning about non-Euclidean geometry) but also from an entertainment point of view (exploring spaces that are not possible in the real world). Additionally, as mentioned above, this approach could be used to create large worlds and help with locomotion in VR. Despite the evident appeal of using non-Euclidean geometry to create novel interaction techniques and mechanics, the experiences and perception of these kinds of spaces remain largely overlooked.

Understanding how different aspects of games affect player experience (PX) is important as this knowledge affords more informed design decisions for the game creators and refined models and theories for games researchers. To date, research efforts have focused on the aspects of games like, to name a few, leaderboards [14], camera perspective [15], avatars [16], music and sound [17], [18], difficulty setting and adjustment [19], [20], [21], [22], power-ups [23], locomotion in VR [24] and how these aspects influence PX. The types of experiences that have been studied include immersion [25], [26], challenge [25], [27], uncertainty [28], controls [25], [29], and mastery [25] amongst many others – in most cases, these PXs are assessed using validated questionnaires. Understanding how players perceive and experience non-Euclidean environments has the potential to provide new knowledge to enrich the existing body of games research.

III. Non-Euclidean Games

The first step in this investigation is the creation of non-Euclidean games and their Euclidean counterparts. Rather than creating bespoke games, we decided to adapt two existing well-known games, *Asteroids* and *Snake*, to create non-Euclidean versions of them, as this would allow for a higher external validity. Adapting existing games to work in non-Euclidean spaces has been explored by several researchers already, e.g. [30], [31]. In our research, we are also adapting two classic 2D games to work in non-Euclidean geometry, but with a focus on the exploration of different player experiences and perceptions of these games.

As mentioned in the introduction, to create non-Euclidean versions of the games, bespoke software has been used. This software uses a polar coordinate system ((r, θ)) for all calculations rather than Cartesian coordinates. Point $O(0, 0)$ is a reference point for the distance coordinate, r, and is usually displayed in the centre of the screen, while eastbound is the reference direction for the bearing coordinate, θ. This allows the same coordinate system to be used irrespective of the curvature of the space. Azimuthal equidistant projection[1] is used to render the curved space onto a flat 2D screen. Other models and projections were evaluated, but the azimuthal projection was selected due to the study's focus on exploring the influence of non-Euclidean geometry on user experience. Consequently, it was crucial to minimize the visual distinctiveness between Euclidean, spherical, and hyperbolic spaces. In the chosen projection, the main visual distinction among these three spaces is in the radial distances between points.

The approach is split into two elements: the movement of the objects and the rendering of the shapes. The objects are

[1]Azimuthal equidistant projection – a map projection of the surface of a sphere so centered at any given point that a straight line radiating from the centre to any other point represents the shortest distance and can be measured to scale [32]

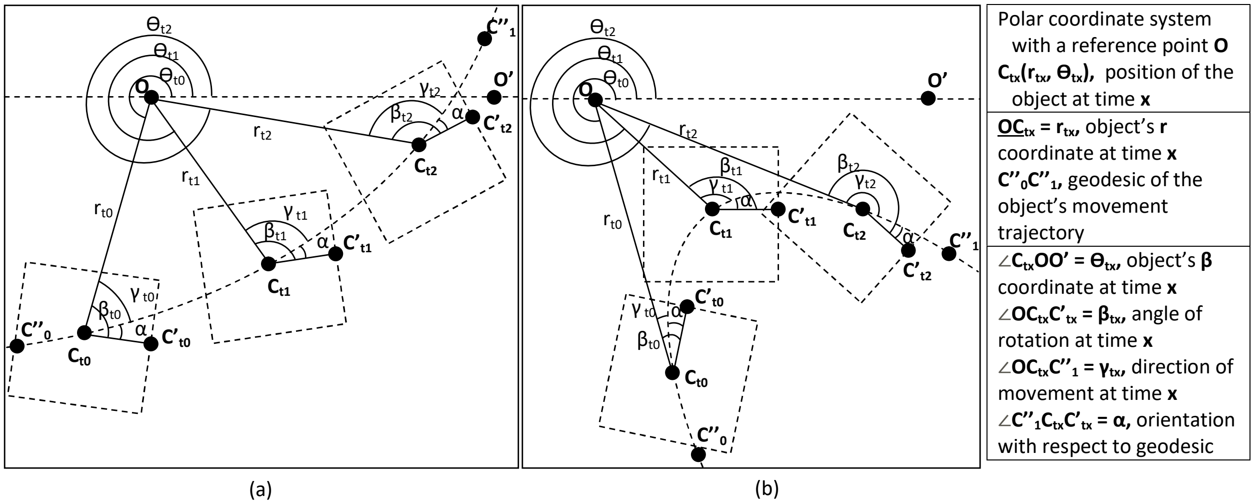

Fig. 2. Movement of the object along a hyperbolic in Spherical (a) and Hyperbolic (b) space. Orientation with respect to the geodesic is kept the same (angle α is constant) if the object is not rotating.

moved continuously by translating them to their position at the next time instance (frame in the engine) as shown on Figure 2. This is unlike the approach used in the games like *Hyperrogue* and *Hyperbolic Maze* [33]. Continuous movement was preferable, because then the curvature could also be modified continuously in real-time.

The trajectory of movement in non-Euclidean space is a geodesic and the object has to keep the same orientation with respect to this geodesic when moving. This trajectory is stored as the direction of the velocity vector of an object. In addition to updating the global position vector of an object, the velocity vector and rotation angle have to be updated accordingly in order to keep their orientation towards the geodesic consistent for a non-rotating object. For a rotating object, extra rotation over time should be added after the position of the object at the next time instance is calculated.

So, the known values are: $C_{t0}(r_{t0}, \theta_{t0})$, position of the object at time $t0$; β_{t0}, object's rotation angle at time $t0$; γ_{t0}, object's velocity vector direction at time $t0$.

The unknown values are: $C_{t1}(r_{t1}, \theta_{t1})$, position of the object at time $t1$; β_{t1}, object's rotation angle at time $t1$; γ_{t1}, object's velocity vector direction at time $t1$.

First the angle α should be found in order to have the preliminaries for calculating r and θ coordinates of the point C_{t1}. α is the difference between object's local reference direction and its geodesic of movement ($C_{t0}C_{t1}$). This angle has to stay constant if the object is not rotating over time. The value is found using object's parameters at time $t0$:

$$\alpha = \beta_{t0} - \gamma_{t0} \tag{1}$$

α should be in the range 0 to π, take explementary angle if $\alpha > \pi$. This will determine the direction of the movement with respect to the reference point (needed to calculate θ_{t1} later).

Then using the spherical or hyperbolic law of cosines the following formulas are derived to find the properties of the object at time $t1$. When Gaussian curvature, K, is positive, the geometry is spherical; when K is negative, the geometry is hyperbolic. Let $d = C_{t0}C_{t1}$, distance the object travels between two frames.

If $K > 0$ and $r = \frac{1}{K}$, then:

$$r_{t1} = \cos^{-1}(\cos\frac{r_{t0}}{r}\cos\frac{d}{r} + \sin\frac{r_{t0}}{r}\sin\frac{d}{r}\cos\gamma_{t0}) \tag{2}$$

$$\Delta\theta = \cos^{-1}(\frac{\cos\frac{d}{r} - \cos\frac{r_{t0}}{r}\cos\frac{r_{t1}}{r}}{\sin\frac{r_{t0}}{r}\sin\frac{r_{t1}}{r}}) \tag{3}$$

$$\gamma'_{t1} = \cos^{-1}(\frac{\cos\frac{r_{t0}}{r} - \cos\frac{d}{r}\cos\frac{r_{t1}}{r}}{\sin\frac{d}{r}\sin\frac{r_{t1}}{r}}) \tag{4}$$

If $K < 0$ and $k = \frac{1}{-K}$, then:

$$r_{t1} = \cosh^{-1}(\cosh\frac{r_{t0}}{r}\cosh\frac{d}{r} - \sinh\frac{r_{t0}}{r}\sinh\frac{d}{r}\cos\gamma_{t0}) \tag{5}$$

$$\Delta\theta = \cos^{-1}(\frac{\cosh\frac{r_{t0}}{r}\cosh\frac{r_{t1}}{r} - \cosh\frac{d}{r}}{\sinh\frac{r_{t0}}{r}\sinh\frac{r_{t1}}{r}}) \tag{6}$$

$$\gamma'_{t1} = \cos^{-1}(\frac{\cosh\frac{d}{r}\cosh\frac{r_{t1}}{r} - \cosh\frac{r_{t0}}{r}}{\sinh\frac{d}{r}\sinh\frac{r_{t1}}{r}}) \tag{7}$$

Now the rest of the unknowns can be found.γ'_{t1} and γ_{t1} are supplementary angles, so:

$$\gamma_{t1} = \pi - \gamma'_{t1} \tag{8}$$

β_{t1} is found using the angle α:

$$\beta_{t1} = \gamma_{t1} + \alpha \tag{9}$$

To find the θ coordinate of C_{t1}, add $\Delta\theta$ to the θ_c. If explementary angle of α was used, subtract $\Delta\theta$ from θ_c instead.

To make the space represented in the engine be more usable for testing and rendering interactive scenes, a cut-off distance has been implemented. This distance is set at a distance of r=N from the reference point of the global polar coordinate system. N is set to be half the window size used in the application and N=500 is set to be the default.

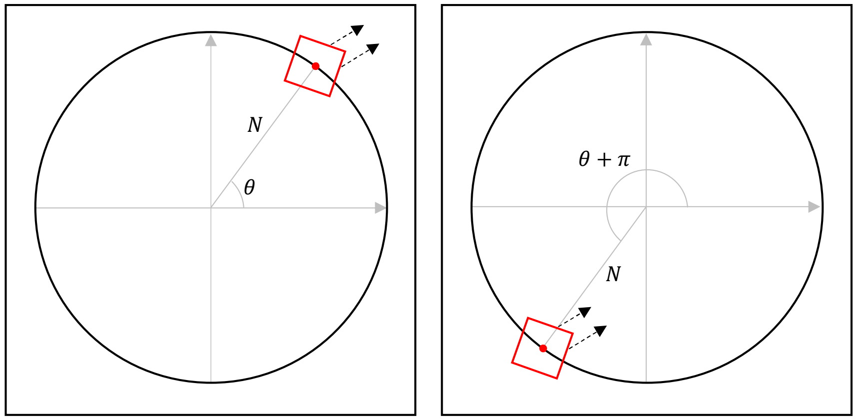

Fig. 3. A quadrilateral object moves further than distance N away from the reference point and is moved to the antipodal point maintaining the orientation, velocity and rotation values.

When the object's centre point moves further away from the origin than the distance N, it is moved to the antipodal point of the limiting circle. This happens in Euclidean and hyperbolic space, as well as spherical space when K¡1. When K¿1 the centre of the shape never goes past distance N, as the whole spherical space is contained within this radius.

There are different ways to accomplish this limitation. In this engine, only the object's position is changed, while orientation, rotation and velocity are unchanged, as illustrated in Figure 3. This means that the space simulated is not a closed manifold. This was the option chosen to preserve similarity in appearance to what is done in classic 2D games, like *Asteroids* and *Pacman*.

If the object moves further away from the origin than the distance N, it is moved onto the screen from the antipodal point at the same distance. Because of the coordinate system used, it is easy to set or lift this limiting distance: the object's theta coordinate is increased by π and then standardised to be in range 0 to 2π. This makes it appear on the antipodal point of the limit circle with preserved velocity and orientation.

We recreated two well-known 2D games to research player experiences, perceptions and preferences of the features developed in this software: *Asteroids* and *Snake*.

We chose *Asteroids* because of the straightforward mapping of the core game mechanics and features (i.e., spaceship, lines and asteroids) to the capabilities of the software (Figure 4 shows time-lapses of the *Asteroids* game in the three conditions). Note that the grid lines are rendered as separate objects and trace the geodesics through the curved space. They help visualise the curvature of the space and choose the correct trajectory when playing the game.

Some additional game logic had to be implemented to develop the game. Proximity-based collision detection was achieved using the bounding circle method: A bounding circle radius is assigned to each object, i.e. spaceship, asteroids and bullets. Circle-circle collision detection works by finding the distance between the centres of two bounding circles and comparing it to the sum of these radii of the circles. Let $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ be centre points of two bounding circles with radii R_1 and R_2 respectively, the collision is found if:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} < R_1 + R_2 \tag{10}$$

This is often optimised by squaring the sum of radii instead of taking the square root on the left-hand side of the inequality:

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 < (R_1 + R_2)^2 \tag{11}$$

Due to the software working in the polar coordinate system, the distance calculation had to be derived from the law of cosines. Let $P_1(r_1, \theta_1)$ and $P_2(r_2, \theta_2)$ be centre points of two bounding circles with radii R_1 and R_2 respectively, in Euclidean space the collision is detected if:

$$r_1^2 + r_2^2 - 2r_1 r_2 \cos \Delta\theta < (R_1 + R_2)^2 \tag{12}$$

Likewise, collision detection in spherical and hyperbolic spaces uses equations derived from the spherical and hyperbolic cosine laws respectively.

If $K > 0$ and $r = \frac{1}{K}$, then:

$$\cos^{-1}\left(\cos\frac{r_1}{r}\cos\frac{r_2}{r} + \sin\frac{r_1}{r}\sin\frac{r_2}{r}\cos\Delta\theta\right) < \frac{R_1 + R_2}{r} \tag{13}$$

If $K < 0$ and $k = \frac{1}{-K}$, then:

$$\cosh^{-1}\left(\cosh\frac{r_1}{k}\cosh\frac{r_2}{k} - \sinh\frac{r_1}{k}\sinh\frac{r_2}{k}\cos\Delta\theta\right) < \frac{R_1 + R_2}{k} \tag{14}$$

In addition to collision detection, the logic for shooting with the player's spaceship, splitting the asteroids into smaller asteroid shards has been implemented and a simple scoring system based on the number of asteroids destroyed.

The bullets a player shoots are displayed as green dashes, which follow the geodesic passing through the player's position in the direction the player is facing. The movement of projectiles is calculated using the same method as for other objects in the simulation. The asteroid is split into two smaller asteroid shards when a collision is detected between the said asteroid and the bullets the player shot. There are two levels of smaller asteroids, like in the original *Asteroids* game. Also, each time a collision between an asteroid (or an asteroid shard) and bullets is detected, the score is incremented. If a player collides with the asteroid (or an asteroid shard), a game over screen is displayed.

This article has been accepted for publication in IEEE Transactions on Games. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TG.2024.3386816

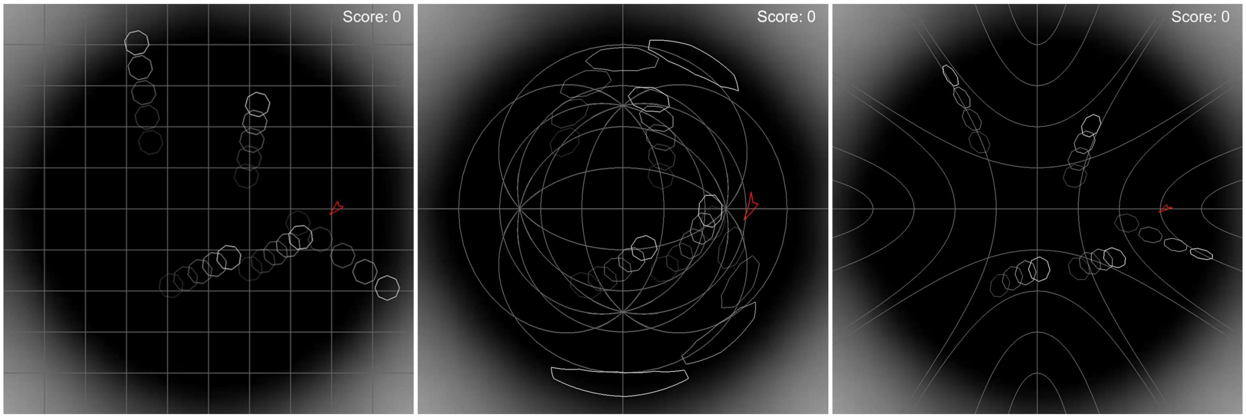

Fig. 4. Time-lapse of the Asteroids game in Euclidean (left), Spherical (centre) and Hyperbolic (right) space. 5 Asteroids move at different speed. The player's spaceship (red) remains stationary.

The second game implemented within the non-Euclidean engine was *Snake* – a popular 2D game with a top-down view, which is a good fit for the capabilities of the software. Some of the features implemented for the *Asteroids* game have been used to build *Snake*, including collision detection, object position randomisation and score functionality.

However, Snake does not move in a continuous simulation, instead, it moves in increments of the size equal to the single section of the snake.

This step-based simulation has been implemented to update the snake's head position following the same method as for the continuous movement simulation. Distance travelled equal to the snake's section size is passed in, rather than calculating it from the object's velocity. Instead of running the update every game loop iteration, it is run once the trigger timer runs out. This trigger time determines the difficulty of the game, making the snake move faster or slower depending on its value. The trigger timer is reset once the step update is run.

A crucial design choice needed to be made when implementing the snake's movement was game controls. In the classic implementation of the *Snake*, up, down, right and left keys are used to turn the snake with respect to the screen. Thus, regardless of whether the snake is moving up or down the screen, the left key input turns the snake towards the left edge of the screen. There are only 4 directions the snake can move in, making it intuitive for the player to control the snake.

However, in the curved environments, due to the shape of the space, the snake could move in any direction on the screen. So, to keep the controls consistent between the different spaces in the game, it was decided to turn the snake with respect to itself, rather than the screen. Regardless of where the snake is on the screen, the left key input would turn the snake 90 degrees to the left. This is illustrated in Figure 5 for Euclidean, spherical and hyperbolic spaces.

A snake is made up of multiple game objects, one for each section of the snake. To reduce the number of movement calculations, the position update function is run for the head section of the snake. After that, the previous positions are propagated through the snake's sections. Meaning that the previous position of the head is assigned to the next snake's section and so on until the tail section is updated. The snake grows in size by 1 section when a collision between the snake's head and the food is detected. The score is updated when the food is consumed and a new position is randomised for it.

Figure 6 shows the time-lapse of the non-Euclidean *Snake* game's execution in the three versions. Each image within a time-lapse is generated every 3 position updates of the snake. After the fifth position update (the second update after the second screenshot is taken) the input has been generated to turn the snake to the right.

IV. User Study

This research aims to explore how players perceive and experience games with different non-Euclidean environments. To do this, we conducted a within-subject design user study with 22 players to assess the PX of mastery, controls, immersion and challenge when playing three versions of *Snake* and *Asteroids*. We also collected qualitative responses from players to learn about their preferences and experiences of these novel game environments.

The following hypotheses were derived from the literature and the assumptions made about the effects of curved spaces on the PX of mastery, challenge, control and immersion:

- **H1:** Games with non-Euclidean curvature are more difficult to master than games with Euclidean curvature.
- **H2:** Games with non-Euclidean curvature are more difficult to control than games with Euclidean curvature.
- **H3:** Games with non-Euclidean curvature are more immersive than games with Euclidean curvature.
- **H4:** Games with non-Euclidean curvature are more challenging to play than games with Euclidean curvature.

A. Experimental method

1) Materials: To assess PX, we used four out of the ten dimensions of the Player Experience Inventory (PXI) [25] that were deemed most relevant in the context of our research: Ease of Control, Challenge, Immersion and Mastery. Each question in all sub-scales was assessed on a 7-point Likert scale with 1 being anchored to "Strongly Disagree" and 7 - to "Strongly Agree". The PXI was chosen amongst other existing

This article has been accepted for publication in IEEE Transactions on Games. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TG.2024.3386816

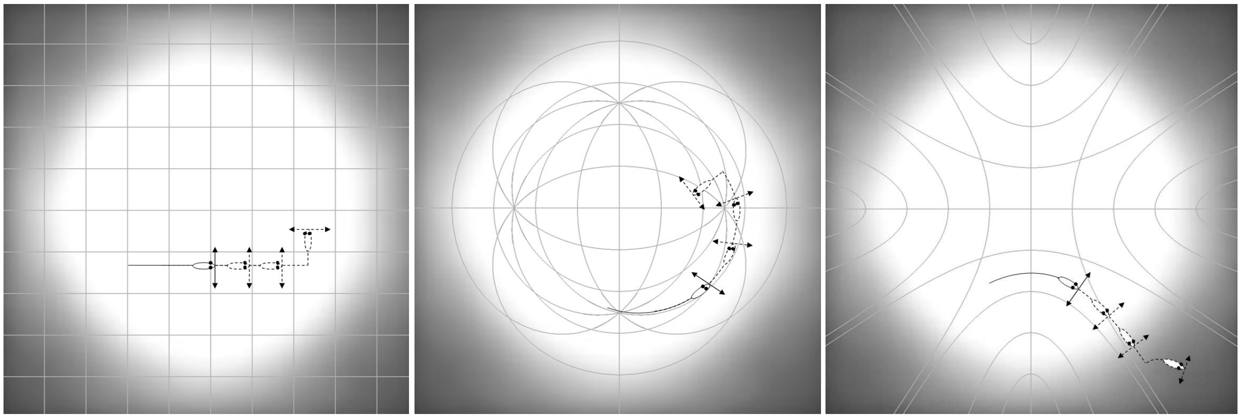

Fig. 5. Snake's movement on a non-curved 2D plane (left), positively curved 2D plane (centre) and negatively curved 2D plane (right). Forward movement trajectory followed by a turn to the left.

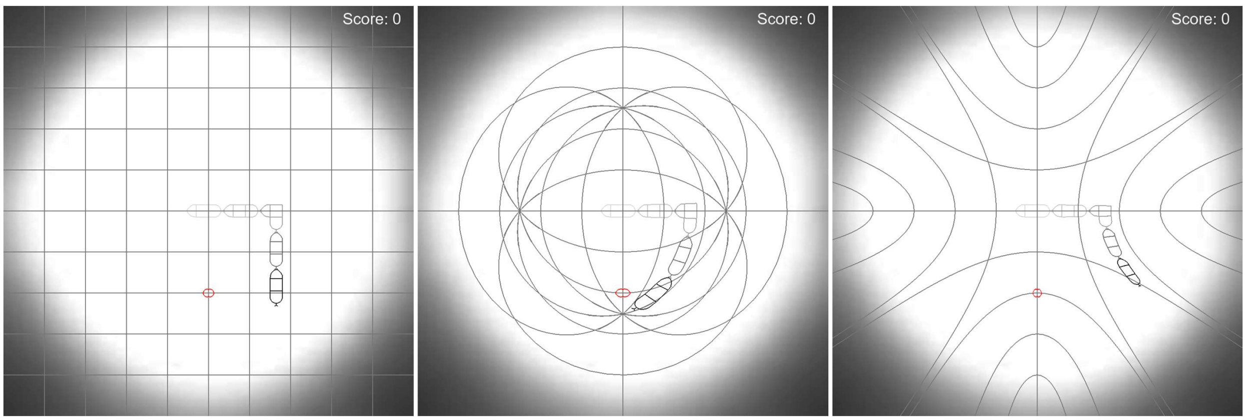

Fig. 6. Time-lapse of the snake game in Euclidean (left), Spherical (centre) and Hyperbolic (right) space. The snake moves forward and makes a turn to the right, toward the food (red shape).

PX questionnaires as it has been validated and used in several recent studies and it allows for the measurement of different facets of PX in a broad sense.

The two games used in the study were the aforementioned *Snake* and *Asteroids*. Each game had three versions: one Euclidean and two non-Euclidean versions (hyperbolic and spherical). The supplementary materials for this paper contain an archive, "*curved space game.rar*", with an executable used for the survey.

2) Procedure: To compare the experiences between three different conditions (Euclidean, Spherical and Hyperbolic) in the two games, we opted in for a within-participant study which would allow players to compare their experiences across games and conditions. We also wanted to know whether players had any preferences towards specific conditions, so being able to compare across these conditions was essential.

For this, we created and distributed a Qualtrics survey that contained: 1) An information sheet and a consent form, 2) Instructions for the study and the games that could be downloaded and played on the participants' personal computers; 3) Demographics questions; 4) Six copies of the four dimensions of the PXI (controls, immersion, challenge and mastery) for each condition in each game; 5) Questions about players' preferences for any of the versions in both games; and 6) Open-ended questions to collect comments from players about the games and their experiences with them. The order in which participants played the games was randomised between games and between the conditions for each of the games to avoid order effects.

We used a repeated-measures ANOVA to test for the effects of the manipulations on the four experiences as measured by the PXI. Bonferroni post hoc test was used for multiple comparisons at a significance level of $\alpha = 0.05$. The qualitative responses were analysed using thematic analysis [34] with short phrases forming the smallest coding units. We introduce the themes accompanied by illustrative quotes from text-based responses, which were kept in the original spelling.

3) Participants: Overall, 48 people took part in the study. After reviewing and cleaning the data to remove 1) incomplete responses, 2) insincere responses and 3) responses which were completed too quickly to be genuine, we included 22 legitimate responses for the analysis. This sample included 16 men, 5 women and a non-binary participant. The average age of the sample was 30.35 ($SD = 6.80$). Most participants (12) said that they played video games daily, 6 participants played games several times a week, 3 - several times a month and one several times a year. Five players said that they had played *Snake* regularly, 12 had played it several times and 4 said

This article has been accepted for publication in IEEE Transactions on Games. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/TG.2024.3386816

that they had never seen this game before. As for *Asteroids*, players were somewhat less familiar with the game with 6 people saying that had never seen the game before, 2 played it regularly and 14 had played the game several times.

B. Results

1) Quantitative findings: For both *Snake* and *Asteroids*, mastery and controls differed significantly between the three conditions: Euclidean, Spherical and Hyperbolic geometry (Table I).

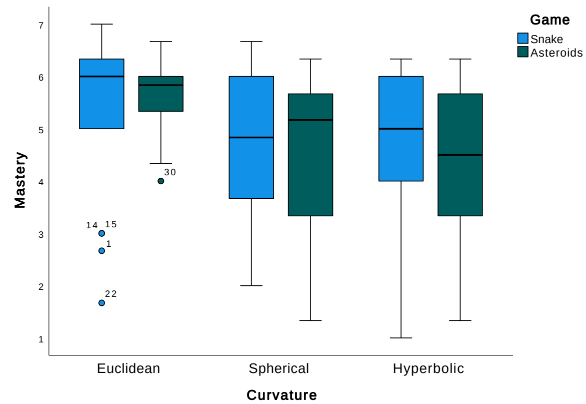

Fig. 7. Experience of Mastery in Euclidean, Spherical and Hyperbolic versions of Snake and Asteroids.

H1: Accepted Participants felt more masterful when playing *Snake* with a Euclidean curvature than with Spherical ($p = 0.041$) but not with Hyperbolic ($p = 0.072$) curvature. For *Asteroids*, players felt more masterful when playing the Euclidean version than the Spherical version ($p = 0.043$) as well as the Hyperbolic ($p = 0.015$).

H2: Accepted Overall, ease of control was significantly different between the three conditions in both *Snake* and *Asteroids*. There were no significant differences between the experiences of control in the three versions of *Snake* but in *Asteroids*, players felt more in control in the Euclidean version than in the Spherical ($p = 0.009$) and the Hyperbolic ($p = 0.012$) versions.

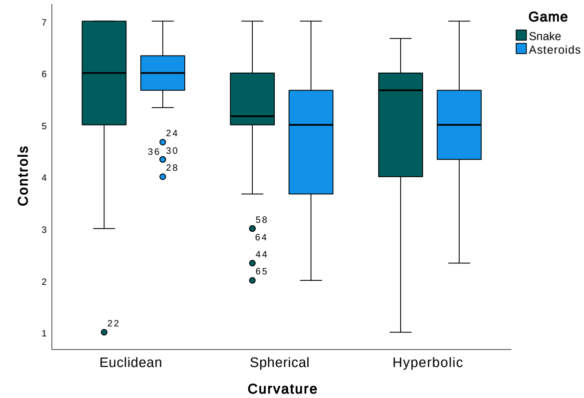

Fig. 8. Experience of Control in Euclidean, Spherical and Hyperbolic versions of Snake and Asteroids.

H3 & H4: Rejected The two non-Euclidean versions of both *Snake* and *Asteroids* did not differ significantly in the experiences of challenge (Figure 10) or immersion (Figure 9).

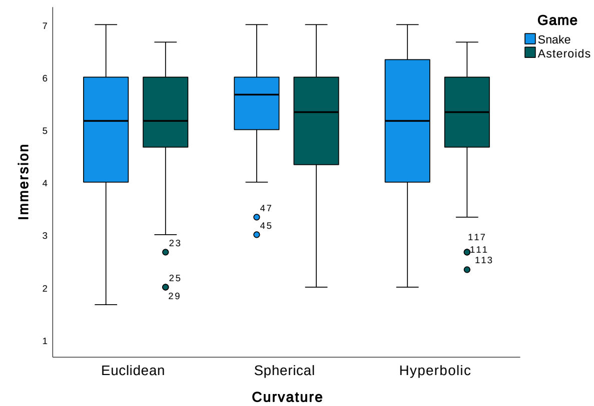

Fig. 9. Experience of Immersion in Euclidean, Spherical and Hyperbolic versions of Snake and Asteroids.

		Snake					Asteroids				
		M	SD	$F(2,19)$	p	$\eta^2_{partial}$	M	SD	$F(2,19)$	p	$\eta^2_{partial}$
	Euclidean	5.47	1.51				5.31	1.27			
Mastery	**Spherical**	4.71	1.43	4.797	**0.013***	0.186	4.47	1.56	7.021	**0.002****	0.251
	Hyperbolic	4.64	1.67				4.38	1.41			
	Euclidean	5.68	1.47				5.67	1.25			
Controls	**Spherical**	5.03	1.41	4.243	**0.021***	0.168	4.76	1.38	8.808	**<0.001*****	0.295
	Hyperbolic	4.76	1.79				4.85	1.22			
	Euclidean	5.08	1.42				4.85	1.40			
Immersion	**Spherical**	5.38	1.09	1.066	0.353	0.048	5.05	1.45	0.463	0.633	0.022
	Hyperbolic	5.06	1.51				5.06	1.31			
	Euclidean	5.20	1.26				5.35	1.47			
Challenge	**Spherical**	4.80	1.26	0.818	0.448	0.038	4.64	1.56	2.506	0.094	0.107
	Hyperbolic	4.91	1.54				4.77	1.71			

TABLE I
PLAYER EXPERIENCE (MASTERY, CONTROLS, IMMERSION, AND CHALLENGE) IN EACH OF THE THREE VERSIONS (EUCLIDEAN, SPHERICAL AND HYPERBOLIC) OF SNAKE AND ASTEROIDS. SIGNIFICANT RESULTS ARE SHOWN IN BOLD (* FOR < 0.05, ** FOR < 0.01 AND *** FOR < 0.001).

This article has been accepted for publication in IEEE Transactions on Games. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TG.2024.3386816

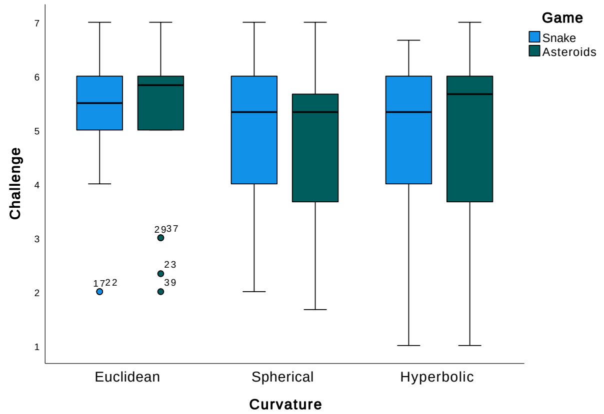

Fig. 10. Experience of Challenge in Euclidean, Spherical and Hyperbolic versions of Snake and Asteroids.

With regards to the preferences for specific curvatures, 12 participants said that the spherical curvature worked better in *Snake*, 7 said hyperbolic, and 3 did not have a preference. For *Asteroids*, 9 players preferred the spherical curvature, 5 - hyperbolic and 8 had no preference. This difference was not significant neither for *Snake* ($\chi^2(2) = 5.545, p = 0.06$) nor for *Asteroids* ($\chi^2(2) = 1.182, p = 0.56$).

2) Qualitative findings: Overall, most participants left positive comments about their experience of the non-Euclidean versions of both games. They described their experiences as *"interesting"* (P4, P5, P16), *"fun"* (P10, P11, P12, P13, P14, P17), *"enjoyable"* (P10, P12, P17), *"stimulating"* (P17) and *"engaging"* (P12, P13), but also *"weird"* (P15, P16) and *"trippy"* (P13, P17, P18). Some also remarked upon the novelty of this system positively impacting their experience: *"it was new [. . .] not like anything I have tried before"* (P17). Most participants noted that the curved versions of the games improved their experiences of the *"conventional and otherwise quite boring game[s]"* (P13) – the curved version was *"more fun than the original"* (P12), making the games *"more enjoyable [. . .] even though it was harder"* (P12).

Challenge Challenge specifically was one of the most frequently mentioned experiences: the 'new' versions of *Snake* and *Asteroids* were *"more challenging"* (P2, P7, P14, P17) or *"harder"* (P12, P15) than the original ones or had an *"added level of difficulty"* (P12, P14). When talking about challenge, however, the players did not refer to challenge as a player experience (c.f. [35], [27]) but they used the term in the contexts of 1) the difficulty of navigating the game world because of the controls; or 2) an increased cognitive involvement as a result of having to *"think many more steps ahead"* (P17).

Controls Several players mentioned that part of the challenge was trying to figure out what they were looking at to begin with and trying to *"get a grip on how to play the game well enough"* (P12). Nonetheless, participants noted that they got used to the controls fairly quickly: *"as i played the game it made more sense"* (P12). Some participants noted that while the controls worked well for the Euclidean version of the games, these could be potentially adapted to make them more intuitive and more suitable for the curved spaces: *"trying some other types of controls and see which ones work better for the curved versions? Like the arrows are meant for the Euclidean spaces, so it would be interesting to see what kind of controls would be more appropriate for the curved ones"* (P17).

Planning There was also the challenge of *"planning"* (P15) and thinking several moved ahead, trying to predict the snake's and asteroid's/bullets' movements around the curved/hyperbolic edges. Specifically, in *Snake*, it was *"difficult to predict how different curves would effect the snakes movement"* (P9) – one needed to *"adapt the movement of the snake and control according to the curvature and to some degree try to predict where the snake would end up based on the location on the map"* (P14), and yet *"it's very satisfying when you "hit" the right curve"* (P13). In *Asteroids*, participants mentioned that the mechanic made the *"entire aiming more challenging, as we have to take into account the curvature while at the same time predict where the asteroid will end up"* (P14) and the players had to *"worry about the trajectories that your bullets will follow"* (P18). They noted that both the hyperbolic and spherical versions were more confusing when it came to navigating around the edges of the map, as it was *"hard to predict where you'd end up once you'd gone round the edge"* (P12). This meant that they spent more time in the centre of the screen: *"I found myself moving around in the center of the map more regularly, just to avoid going into the extrememities as the unpreditable movement of the asteroids often lead to disaster"* (P14).

Thinking differently The comments about trying to think several steps ahead were often made with relation to a change in perspective and the player's abilities to think differently and think outside of the 'norm'. Participants noted that they were *"forced to think in terms of the curvature"* (P13), which *"adds complexity and unpredictability, and trains your brain to think in a different way"* (P15). *"[I]t require[d] a different set of skill or at least more refined spatial reasoning from the player than the normal version"* (P17) as *"[the mechanic] forces you to understand the dynamics of the geodesics [and] forces you to think hyperbolically since L/R turns onto different geodesics can have totally different effects"* (P10).

Suggestions for future work Players had several suggestions for potential future expansions to the games they played. These included improving the graphics and adding textures. Mechanics-wise, participants mentioned *"show[ing] the vector or predicted location where my bullets will end up when shooting"* (P14), adding new maps or *"more levels with double torus, or even a mobius strip – [. . .] changing topology throughout the levels"* (P10). To encourage players to 'brave' the edges more in the *Asteroids* game, it was suggested to *"find a way for players to "boost" out of corners in the hyperbolic version"* (P14) and to add *"a motivation to explore the edges"* (P15). In-game instructions for first-time players were also mentioned as well as potential changes to the controls to make them more intuitive and to add *"more granularity"* (P14). Finally, some participants also suggested adding a difficulty adaptation: *"as you do well, it stays flat and as you miss shots, or a long time between hits, the map distorts itself. Only "repairing" as you start doing better"* (P12). It was also suggested to *"tweak the curvature for more exaggerated/less exaggerated effects while a player attunes to the changes"* (P10).

Many participants expressed their interest in playing more games with a curved space mechanic and provided several suggestions for the types of games that, in their opinion, could work with it. They named platformers, puzzle games, *"physics game[s] with trajectory-based aiming"* (P15) (e.g. 'Angry Birds' type games), strategy games, racing games and potentially *"FPS game[s], like* Unreal Tournament*"* (P12). Our participants also mentioned remakes of 'classic' games like *Breakout*, *Pacman*, *Pong*, *Elite* (1984), *Super Mario Galaxy*, *Space Invaders*, and *"sonic the hyperbolic hedgehog"* (P13). New bespoke games exploring this space were also welcome: *"something that truly exploits the surfaces for their own right would force new creative challenges"* (P10).

V. Discussion

In this work, we demonstrate that non-Euclidean geometry can be used in video games to provide enjoyable novel interactions and to allow players to explore spaces that are not physically possible in the real world.

Game controls were perceived as easier in the Euclidean version of the games by our participants. Familiarity with the original games could have played a role in how easy the players perceived the controls as the 'arrow' mapping of the in-game controls to the familiar 'flat' space of the Euclidean versions could be considered more intuitive. However, it is also possible that if the controls that worked well for the Euclidean version, they may not naturally translate to curved spaces. As one participant suggested, alternative control designs could unveil more intuitive mappings of real world controls to the impossible spaces in the game.

Familiarity of some players with at least one of the games would also mean that they would likely approached gameplay with preconceived expectations, strategies, and knowledge of game mechanics. This could potentially affect their engagement and decision-making within the game. This familiarity could have affected the evaluation of these novel mechanics too though we lack data to confirm or refute this assumption – future work could focus on the qualitative evaluation of non-Euclidean spaces by the players with different prior experiences of games with this implementation. While familiarity could result in quicker mastery of gameplay mechanics and better understanding of objectives, it is important to note that all participants played both games in all conditions, which minimises the potential impact of prior familiarity on the overall results. Any observed differences in PX between the games are likely attributed to their inherent characteristics rather than prior experience with the games.

In non-Euclidean game worlds where the camera follows the player, (e.g. in mobile games like *Monsters*), changes in camera positioning and control may have a less noticeable effect on player experience compared to top-view games like Snakes and Asteroids. This is because the closely tracked camera perspective in these games creates a more dynamic and immersive experience, where alterations in camera angles or movements may be subtler and have a less immediate impact on the player. In contrast, in top-view games where the player's viewpoint is fixed relative to the game world, shifts in camera perspective may be more noticeable and potentially influencing spatial awareness, navigation, and overall gameplay. Therefore, while camera positioning and control are important considerations, their impact may vary depending on the specific mechanics and camera perspectives employed.

Being unfamiliar with the object movement through curved space could have created a perception of a game being harder to master, which also became apparent in the qualitative responses from the players describing their experience when first seeing the curved space. Multiple participants mentioned that they did not immediately understand how to control the player character in the curved space, but were able to figure this out shortly after. This could also impact the perceived challenge of the game. In a spherical space, the snake follows the great circle geodesics, so it ends up going in a circle if not controlled and thus can collide with itself easier. On the other hand, in a hyperbolic space, the snake will go off-screen (and re-enter the screen from the antipodal point) following the geodesics. The following comment highlights how it was harder to adjust to the controls in a spherical space over hyperbolic: *"I noticed I was still trying to apply a Cartesian style of movement to reach the object. I began using the geodesics on the Spherical surface to reach the object instead of turning so much, but I think the Hyperbolic surface really shines here – it forces you to think hyperbolically since L/R turns onto different geodesics can have totally different effects, at least compared to what you expect in a Cartesian system."* (P10). This comment refers to the fact that when turning 90 degrees with a right or left turn, one switches the geodesic the snake is following, but this new geodesic will not continue orthogonally on the screen, it will also curve away from the origin due to exponentially expanding amount of space. If one wants to return the snake towards the origin, multiple 90-degree turns might be required, which could potentially require more concentration from the player.

Hyperbolic movement is potentially more intuitive in a space setting. This could be due to controls and movement in the *Asteroids* being less constrained than in *Snake*. Snake movement is step-based and turns are only in 90-degree angles, while the spaceship in the *Asteroids* game has free range of movement, so the motion likely looks more natural.

Interestingly though, players did not rate their experience of challenge as significantly higher in the curved spaces than in the Euclidean version of the two games but the comments suggested that the non-Euclidean versions were 'harder' or 'more challenging'. Looking closer at the context in which our participants described their experiences as challenging, they either referred to the controls of the game (as discussed above) or the increased cognitive involvement when playing the games with curved spaces as this required a change in perspective, thinking differently and planning ahead. While the PXI does not capture such nuanced experiences, it is possible to further investigate if players' increased cognitive involvement contributed to their experience of cognitive and performative challenge using the Challenge Originating from Recent Gameplay Interaction

Scale (CORGIS) [27] and explore the cognitive involvement aspect (as well as other components) of immersion using the Immersive Experience Questionnaire (IEQ) [26]. These questionnaires can provide a more granular breakdown of experiences that players can have when playing games with curved spaces and would allow to accurately capture these nuanced experiences. Triangulating self-reports with objective measures like players' in-game performance (e.g. completion time, scores, or achievements to evaluate players' actions and progress in the game) and behaviour analysis (e.g. player movement patterns and interaction frequency) could offer a more comprehensive understanding of this player experience and potentially enhance the validity and reliability of findings.

A. Limitations and Future Work

The games discussed in this paper are currently only available to Windows users but future iterations of the software will aim at making these games accessible on other platforms. We also believe that the suggestions from the participants noted in Section IV-B2 are realistically possible to implement. For instance, adding textures could make the visualisation of shape distortion more understandable and realistic, as well as making the games more visually appealing. A dynamic difficulty adjustment will also be explored in future iterations of the games, as well as creating a bespoke game with a gameplay focus on the curved space mechanic. To make the curved space more intuitive and the movement more natural, a free range of motion as well as gradual acceleration can be used instead of step-based movement.

Non-Euclidean versions of both games allow for a useful visualisation of the spherical and hyperbolic projections and can help players understand the object movement in such environments. As the properties of these environments are not taken into account when designing in-game challenges, it would be an interesting direction for future work. Finally, to further our understanding of player experiences of games with non-Euclidean mechanics and widen the possibility space of game design, we propose to run co-design sessions with game designers to explore how such mechanics can be used to create novel experiences and interactions in video games. It also would be interesting to see if this software can be used for educational purposes, for instance, to train the spatial reasoning and navigation skills of players.

VI. Conclusions

In this work, we explored how players perceive and experience video games created using non-Euclidean geometry. For this, we created clones of two well-known 2D games, *Snake* and *Asteroids*, and adapted these games to make two non-Euclidean versions of these games: hyperbolic and spherical. We evaluated the experiences of players of these two games in the three conditions and found that players experience higher mastery and ease of control in the Euclidean version of the two games and perceived challenge and immersion do not differ significantly between the three versions of the games. We provide a discussion of the meaning and the significance of these results and propose future work in this domain to deepen our understanding of impossible spaces and motivate the creation of novel interactions and experiences using non-Euclidean geometry in game design.

References

[1] D. Osudin, C. Child, and Y.-H. He, "Rendering non-euclidean space in real-time using spherical and hyperbolic trigonometry," *International Conference on Computational Science, Springer*, pp. 543–550, 2019.

[2] NuSan, "*Fragments of Euclid*," Windows, macOS, Linux, March 2017. [Online]. Available: https://nusan.itch.io/fragments-of-euclid

[3] W. C. Studio, "*Manifold Garden*," iOS, macOS, Windows, PlayStation 4, Nintendo Switch, Xbox One, Xbox Series X/S, PlayStation 5, October 2020.

[4] V. Room, "*Tea for God*," Windows, 2023. [Online]. Available: https://store.steampowered.com/app/1764400/Tea_For_God/

[5] R. Hudson, C. Gunn, G. K. Francis, D. J. Sandin, and T. A. DeFanti, "Mathenautics: using vr to visit 3-d manifolds." *In Proceedings of the 1995 symposium on Interactive 3D graphics*, pp. 167–170, 1995.

[6] M. Phillips and C. Gunn, "Visualizing hyperbolic space: Unusual uses of 4x4 matrices," *In Proceedings of the 1992 symposium on Interactive 3D graphics*, pp. 209–214, 1992.

[7] J. Weeks, "Non-euclidean billiards in vr." *In Proceedings of Bridges 2020: Mathematics, Art, Music, Architecture, Education, Culture*, pp. 1–8, 2020.

[8] L. Velho, V. da Silva, and T. Novello, "Immersive visualization of the classical non-euclidean spaces using real-time ray tracing in vr." *In Graphics Interface*, 2020.

[9] S.-T. B. L. Co., "*Warped Mines: Minesweeper Game*," Android, Web, iOS, 2022. [Online]. Available: https://cdn.warpedmines.com/

[10] CodeParade. (2022) Hyperbolica. [Online]. Available: https://www.roguetemple.com/z/hyper/

[11] Z. R. Games. (2011) Hyperrogue. [Online]. Available: https://www.roguetemple.com/z/hyper/

[12] H. Games, "*No Man's Sky*," PlayStation, Windows, Xbox, Nintendo Switch, macOS, August 2016.

[13] M. Studios, "*Minecraft*," Windows, macOS, Linux, November 2011.

[14] J. T. Bowey, M. V. Birk, and R. L. Mandryk, "Manipulating leaderboards to induce player experience," in *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, ser. CHI PLAY '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 115–120. [Online]. Available: https://doi.org/10.1145/2793107.2793138

[15] A. Denisova and P. Cairns, "First person vs. third person perspective in digital games: Do player preferences affect immersion?" in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 145–148. [Online]. Available: https://doi.org/10.1145/2702123.2702256

[16] M. V. Birk, C. Atkins, J. T. Bowey, and R. L. Mandryk, "Fostering intrinsic motivation through avatar identification in digital games," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 2982–2995. [Online]. Available: https://doi.org/10.1145/2858036.2858062

[17] K. Rogers, M. Jörg, and M. Weber, "Effects of background music on risk-taking and general player experience," in *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, ser. CHI PLAY '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 213–224. [Online]. Available: https://doi.org/10.1145/3311350.3347158

[18] T. Farkas, A. Denisova, S. Wiseman, and R. Fiebrink, "The effects of a soundtrack on board game player experience," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3491102.3502110

[19] J. D. Smeddinck, R. L. Mandryk, M. V. Birk, K. M. Gerling, D. Barsilowski, and R. Malaka, "How to present game difficulty choices? exploring the impact on player experience," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 5595–5607. [Online]. Available: https://doi.org/10.1145/2858036.2858574

[20] A. Denisova and P. Cairns, "Player experience and deceptive expectations of difficulty adaptation in digital games," *Entertainment Computing*, vol. 29, pp. 56–68, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1875952118300521

[21] P. Moschovitis and A. Denisova, "Keep calm and aim for the head: Biofeedback-controlled dynamic difficulty adjustment in a horror game," *IEEE Transactions on Games*, 2022.

[22] A. Denisova and P. Cairns, "Adaptation in digital games: The effect of challenge adjustment on player performance and experience," in *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, ser. CHI PLAY '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 97–101. [Online]. Available: https://doi.org/10.1145/2793107.2793141

[23] A. Denisova and E. Cook, "Power-ups in digital games: The rewarding effect of phantom game elements on player experience," in *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, ser. CHI PLAY '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 161–168. [Online]. Available: https://doi.org/10.1145/3311350.3347173

[24] J. Frommel, S. Sonntag, and M. Weber, "Effects of controller-based locomotion on player experience in a virtual reality exploration game," in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, ser. FDG '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3102071.3102082

[25] V. V. Abeele, K. Spiel, L. Nacke, D. Johnson, and K. Gerling, "Development and validation of the player experience inventory: A scale to measure player experiences at the level of functional and psychosocial consequences," *International Journal of Human-Computer Studies*, vol. 135, p. 102370, 2020. [Online]. Available: https://doi.org/10.1016/j.ijhcs.2019.102370

[26] C. Jennett, A. L. Cox, P. Cairns, S. Dhoparee, A. Epps, T. Tijs, and A. Walton, "Measuring and defining the experience of immersion in games," *International journal of human-computer studies*, vol. 66, no. 9, pp. 641–661, 2008. [Online]. Available: https://doi.org/10.1016/j.ijhcs.2008.04.004

[27] A. Denisova, P. Cairns, C. Guckelsberger, and D. Zendle, "Measuring perceived challenge in digital games: Development & validation of the challenge originating from recent gameplay interaction scale (corgis)," *International Journal of Human-Computer Studies*, vol. 137, p. 102383, 2020. [Online]. Available: https://doi.org/10.1016/j.ijhcs.2019.102383

[28] C. Power, P. Cairns, A. Denisova, T. Papaioannou, and R. Gultom, "Lost at the edge of uncertainty: Measuring player uncertainty in digital games," *International Journal of Human–Computer Interaction*, vol. 35, no. 12, pp. 1033–1045, 2019.

[29] R. M. Ryan, C. S. Rigby, and A. Przybylski, "The motivational pull of video games: A self-determination theory approach," *Motivation and emotion*, vol. 30, no. 4, pp. 344–360, 2006.

[30] F. D. Guimarães, V. M. Mello, and L. Velho, "Geometry independent game encapsulation for non-euclidean geometries," *In Proceedings of SIBGRAPI Workshop of Works in Progress*, vol. 1, 2015.

[31] S.-K. László and M. Magdics, "Adapting game engines to curved spaces," *The Visual Computer*, pp. 1–13, 2021.

[32] (2009) Azimuthal equidistant projection. [Online]. Available: https://www.merriam-webster.com/dictionary/azimuthal\%20equidistant\%20projection

[33] D. Madore. (2013) Hyperbolic maze. [Online]. Available: http://www.madore.org/~david/math/hyperbolic-maze.html

[34] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa

[35] A. Denisova, C. Guckelsberger, and D. Zendle, "Challenge in digital games: Towards developing a measurement tool," in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 2511–2519. [Online]. Available: https://doi.org/10.1145/3027063.3053209

Dr Daniil Osudin is a Lecturer in Computer Science at City, University of London, UK. Until November 2022, he was a PhD student at City, University of London, UK. Daniil's research was in developing a novel method of rendering non-Euclidean geometry in real-time using Spherical and Hyperbolic trigonometry; implementing it within a framework which allows the creation of custom environment; and gauging the interest in non-Euclidean games.

Dr Alena Denisova is a Lecturer (Assistant Professor) in Computer Science at the University of York, UK. Until January 2022, she was a Lecturer at City, University of London, UK. Alena's research focuses on conceptualising and measuring user experiences of video games and designing and building educational and persuasive interactive media. Her work explores the role of the 'placebo effect' of technology in shaping player experiences, perceived challenge and uncertainty in video games, and, more recently, emotionally impactful player experiences – understanding how these experiences are shaped with the view to inform the design of games that promote these experiences.

Dr Christopher Child has been an Associate Dean for Employability and Corporate Relations at the School of Science and Technology at City, University of London, UK since 2020. He has been a games technology lecturer at City, University of London, UK since 2005, and became course director for Computer Games in 2008. Chris is also a researcher in the Department of Computer Science developing cutting edge game agent AI using techniques such as reinforcement learning, probabilistic planning, environment modelling and approximate dynamic programming.