Holonomy: A Virtual Environment based on Hyperbolic Space

Martin Skrodzki*
Comp. Graphics and Vis., TU Delft
Delft, Netherlands
M.Skrodzki@tudelft.nl

Scott Jochems
TU Delft
Delft, Netherlands
S.A.Jochems@student.tudelft.nl

Joris Rijsdijk
TU Delft
Delft, Netherlands
J.A.Rijsdijk@student.tudelft.nl

Ravi Snellenberg
TU Delft
Delft, Netherlands
R.Snellenberg@student.tudelft.nl

Rafael Bidarra
Comp. Graphics and Vis., TU Delft
Delft, Netherlands
R.Bidarra@tudelft.nl

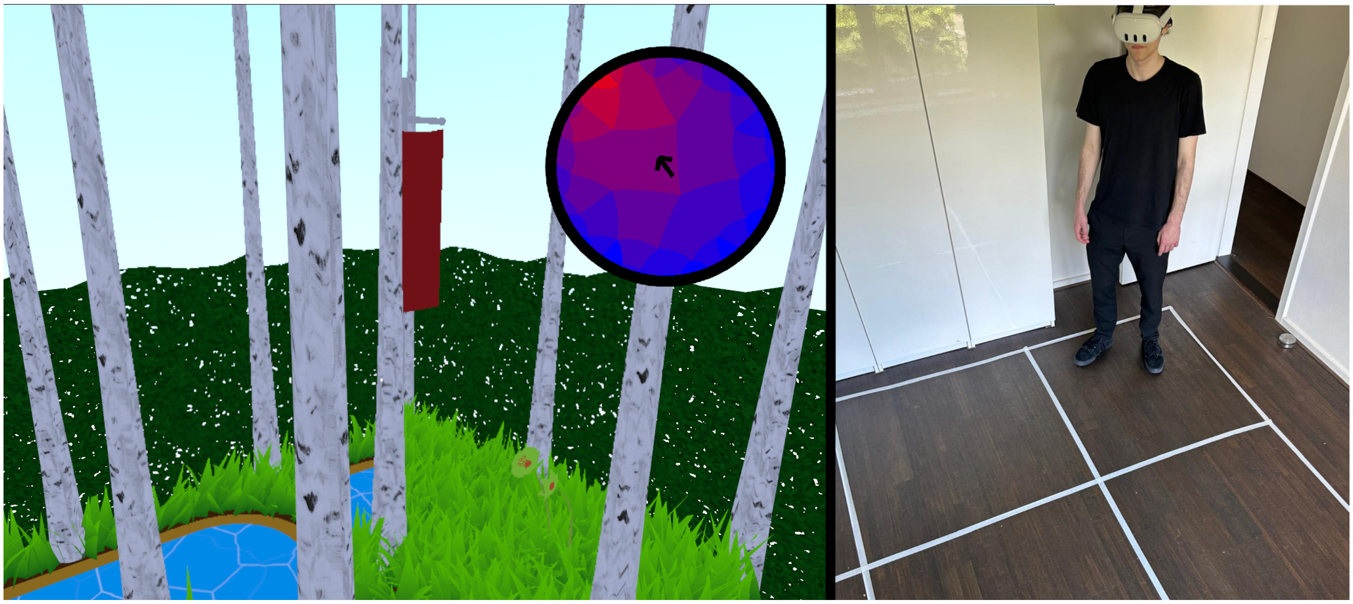


Figure 1: Left: Rendering of the hyperbolic virtual environment *Holonomy,* including a mini-map in the HUD. Hyperbolic effects are visible in the flag and the creeks partially vanishing behind the trees. Right: A player in the physical move area, with the corresponding square tiles marked on the floor.

ABSTRACT

Holonomy is a virtual environment based on the mathematical concept of hyperbolic geometry. Unlike other environments, *Holonomy* allows users to seamlessly explore an infinite hyperbolic space by physically walking. They use their body as the controller, eliminating the need for teleportation or other artificial VR locomotion methods. This paper discusses the development of *Holonomy*, highlighting the technical challenges faced and overcome during its creation, including rendering complex hyperbolic environments, populating the space with objects, and implementing algorithms for finding shortest paths in the underlying non-Euclidean geometry. Furthermore, we present a proof-of-concept implementation in the form of a VR navigation game and some preliminary learning outcomes from this implementation.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **User interface toolkits**; *Visualization toolkits.*

KEYWORDS

Virtual Environments, Virtual Reality, Hyperbolic Geometry, Embodied Control, Infinite Exploration

ACM Reference Format:
Martin Skrodzki, Scott Jochems, Joris Rijsdijk, Ravi Snellenberg, and Rafael Bidarra. 2024. Holonomy: A Virtual Environment based on Hyperbolic Space. In *The 29th International ACM Conference on 3D Web Technology*

---

[*] Corresponding author: mail@ms-math-computer.science

WEB3D '24, September 25–27, 2024, Guimarães, Portugal

ACM ISBN 979-8-4007-0689-9/24/09…$15.00
https://doi.org/10.1145/3665318.3677149

(WEB3D '24), September 25–27, 2024, Guimarães, Portugal. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3665318.3677149

1 INTRODUCTION

In an increasingly interconnected and digitized world, people communicate and interact in various ways and via multiple different platforms. In this context, virtual environments rapidly gain importance for both professional and leisurely exchanges. Recent hardware developments like Apple's *Vision Pro* seek to integrate virtual content into the user's physical surroundings, merging the two in the amalgam of Spatial Computing. Similarly, the Metaverse movement aims to integrate everyday processes within a virtual environment. Using virtual, artificial, and mixed reality blurs the lines between physical and virtual worlds. In the context of these developments, we present a virtual environment based on the mathematical concept of hyperbolic space.

Hyperbolic geometry, a non-Euclidean geometry pioneered by mathematicians such as Nikolai Lobachevsky and János Bolyai in the 19th century, offers a fascinating alternative to the familiar principles of Euclidean space. In hyperbolic geometry, the parallel postulate is replaced by the concept that through a given point, an infinite number of lines can be drawn parallel to a given line, leading to properties such as space being negatively curved and exhibiting unique geometrical constructions. Understanding and visualizing hyperbolic geometry can be challenging due to its abstract nature and departure from our intuitive understanding of space. However, the unique geometric properties of hyperbolic spaces provide creative opportunities for creating new virtual environments.

Hyperbolic spaces have been used very successfully, specifically regarding data representation. For example, it is possible to embed hierarchical structures like trees into two-dimensional hyperbolic space with arbitrarily low distortion [Sarkar 2011]. By this property, previous works have successfully embedded social networks [Verbeek and Suri 2014] or the Internet [Boguná et al. 2010] into hyperbolic space. This usage as an exploration space for virtual data further motivates our creation of a new virtual environment based on hyperbolic spaces.

The advent of virtual reality (VR) technology has opened up new avenues for exploring and experiencing hyperbolic geometry in immersive and interactive ways. VR provides a platform where users can not only passively observe hyperbolic spaces but actively engage with and navigate through them. We apply specific terminology in the following discussion on navigating VR and throughout the paper. The *physical space* is the real-world space the user occupies and where they perform their movements, see Figure 1, right. In contrast, the *virtual space* refers to the virtual environment in which the user is immersed, the virtual world rendered via the VR headset, see Figure 1, left. Within the physical space, the user has a dedicated *move area*: ideally, a square shape, physically unobstructed free space, indicated on the floor in Figure 1, right. As the user moves through this area physically, we refer to it as *walking*. Every walking motion will be translated into movement in the virtual environment. Those parts of the virtual environment shown to the user will be called *accessible*, while hidden areas are *inaccessible*.

There are several ways of allowing users to navigate through the virtual environment. One option is to allow them to walk within the move area. The benefit of this method is the intuitive understanding and freedom of movement. This method is not without flaws, as it either requires expensive equipment—such as specialized treadmills [Iwata 1999] or a CAVE® [Razzaque et al. 2002]—or the use of redirected walking techniques in a large move area [Fan et al. 2022] to give users the required freedom of movement. Aside from the user's walking, the most common way is some form of controller input to glide or teleport the user, which in turn can create VR sickness [Monteiro et al. 2021]. The benefit of this method is that it requires only a minuscule move area to work. However, it reduces the feeling of presence and thus immersion within a virtual environment [Clifton and Palmisano 2020]. In contrast, the virtual environment presented in this paper provides an infinitely explorable space that can be traversed solely by walking in a relatively small move area without additional hardware requirements.

Creating a virtual environment for VR comes with its own set of crucial technical challenges compared to a 3D environment explored via a display. The virtual environment must perform at a stable high frame rate, high resolution, and very low input latency [Wang et al. 2023]. To achieve this on widely available consumer-level hardware implies a sacrifice in visual quality, paired with a lower capacity for overall scene complexity. This is an additional challenge in a virtual space based on hyperbolic geometry, as classical rendering techniques, pipelines, and hardware are optimized for processing Euclidean scenes. The same holds for representations of the environment and their object population, as well as navigational cues for the user, for instance via the display of the shortest path to a point of interest. In this paper, we tackle these challenges and offer some solutions. Note that while we developed VR first, our environment does not exclude other modes of exploration. Users can also explore the environment with a traditional screen and controller setup.

In summary, this paper introduces *Holonomy*, a new virtual environment based on principles of hyperbolic geometry. We discuss its development choices, technical challenges, and potential applications. Through the fusion of hyperbolic geometry and VR technology, Holonomy offers a novel approach to virtual exploration in non-Euclidean spaces. The specific contributions of this work are:

- A virtual environment, based on the mathematical concept of hyperbolic spaces, Section 4.
- Potentially infinite exploring of a virtual world solely by physically walking in a move area, without the need for teleportation or other artificial locomotion, Section 4.
- Efficient rendering techniques for the virtual environment to maintain high frame rates suitable for VR, Section 5.2.
- Adaptation of a wave function collapse algorithm to populate the virtual environment, Section 5.3.
- Adjustment of shortest path algorithms for user guidance in navigating the environment, Section 5.4.
- A proof-of-concept implementation of our environment as a VR navigation game, Section 6.

We see two immediate applications of our work. First, the interactive teaching of aspects of hyperbolic geometry. Second, usage of

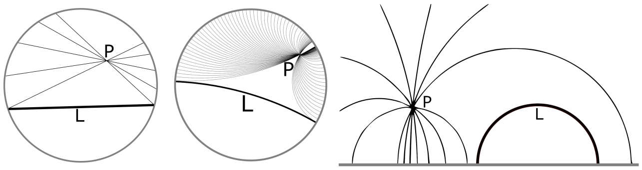


Figure 2: A line L and several parallels to L through a point P in different models of the hyperbolic plane. Left: Beltrami-Klein model in a disk, straight hyperbolic lines remain straight; Center: Poincaré disk model, straight hyperbolic lines become curved; Right: Poincaré half-plane model, straight hyperbolic lines are either vertical or half circles, but angles are kept intact. Reproduced with permission [Skrodzki 2021].

the system for the presentation of unique navigational challenges, for instance, within cognition studies. We will elaborate on these applications in Section 7.

2 BACKGROUND

Before discussing our virtual environment, we set the stage by introducing some necessary background. First, we cover basic notions of hyperbolic geometry, highlighting the notable difference to Euclidean geometry that through a given point P off a given line L, there are infinitely many lines that do not intersect L, see Figure 2. Second, we introduce the notion of a *hyperbolic tiling*, which will provide the main mode of connection between the user's move area and the virtual environment.

2.1 Visualizing Hyperbolic geometry

Various visualization models have been developed to help conceptualize hyperbolic space. The Beltrami-Klein model, for instance, renders the hyperbolic plane into a disk in Euclidean space, see Figure 2, left. It maps straight hyperbolic lines to straight lines in the model but distorts their angles. The namesakes are Eugenio Beltrami and Felix Klein. Another common approach is the Poincaré disk model, named after the French mathematician Henri Poincaré. In this model, hyperbolic space is again represented within a Euclidean disk but straight hyperbolic lines are rendered as circular arcs orthogonal to the disk's boundary, see Figure 2, center. Another visualization model is the Poincaré half-plane model, which utilizes the entire upper half of the Euclidean plane to map the hyperbolic plane. It renders straight hyperbolic lines as vertical lines or half circles while preserving angles, see Figure 2, right. All three models offer distinct perspectives on hyperbolic geometry, aiding comprehension and study of its unique properties and structures [Kinsey et al. 2011, Chapter 3.6].

Maurits Cornelis Escher, the renowned Dutch artist, played a pivotal role in popularizing the Poincaré disk model through his iconic *Circle Limit* series. Inspired by a diagram, which the British mathematician Donald Coxeter used in a 1957 mathematical paper on hyperbolic geometry [Coxeter 1957], Escher created intricate woodcut prints depicting tessellations of hyperbolic space, where angels, demons, and other figures form repeating patterns that fill the entire plane. Tesselations, also called tilings, of the hyperbolic plane are the foundation for the following discussion. For a more general introduction to hyperbolic tilings, see [Conway et al. 2008].

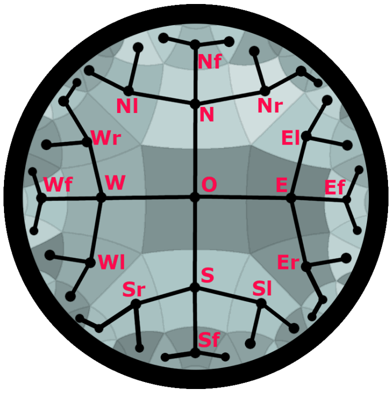


Figure 3: The spanning tree of the graph. Nodes represent the hyperbolic squares underneath. Labels indicate the indices of the nodes.

Note that a hyperbolic space is a homogeneous space with constant negative curvature. This property is exhibited for instance in the fact that the sum of angles of a triangle is less than 180 degrees. Another way to express this fact via tilings is to count the number of tiles that meet at a vertex. Consider an ordinary bathroom tiling of square tiles, see Figure 8, left. Here, four squares meet at a vertex, giving an angle sum of 360 degrees. In a corresponding hyperbolic square tiling, more than four squares can meet, see Figure 8, right, for an example of five squares meeting.

Having more than four squares meet at a point enables the effect of *Holonomy*—after which our project is named—by which a closed circular walk in the physical space causes a rotation or displacement in the hyperbolic virtual space. This was first presented as a phenomenon in VR by Henry Segerman[1] as follows: A user walks in a 2×2 Euclidean square grid, marked on the floor of their move area. Here, they have to take a step and perform a 90-degree left turn exactly four times to return to the square where and orientation in which they started, see Figure 4, top row. Simultaneously, they see themselves navigating a hyperbolic square tiling in VR with five squares meeting at a vertex. That is, after four consecutive steps and turns, they are not yet back to the starting square, see Figure 4, bottom row. It takes them another step and a 90-degree turn to return to their virtual starting position, causing a discrepancy between their physical and virtual presence. This discrepancy will allow for an infinite exploration within our virtual environment solely based on the user walking within their move area.

2.2 Representing hyperbolic tilings

In our virtual environment, we will focus on the hyperbolic tiling where five squares meet at any vertex. See Section 4 for reasoning regarding this design choice. This hyperbolic tiling has to be represented by a suitable data structure. This is a vital design choice because it immediately determines the difficulty of implementing

---

[1] See the corresponding video https://www.youtube.com/watch?v=ztsi0CLxmjw.

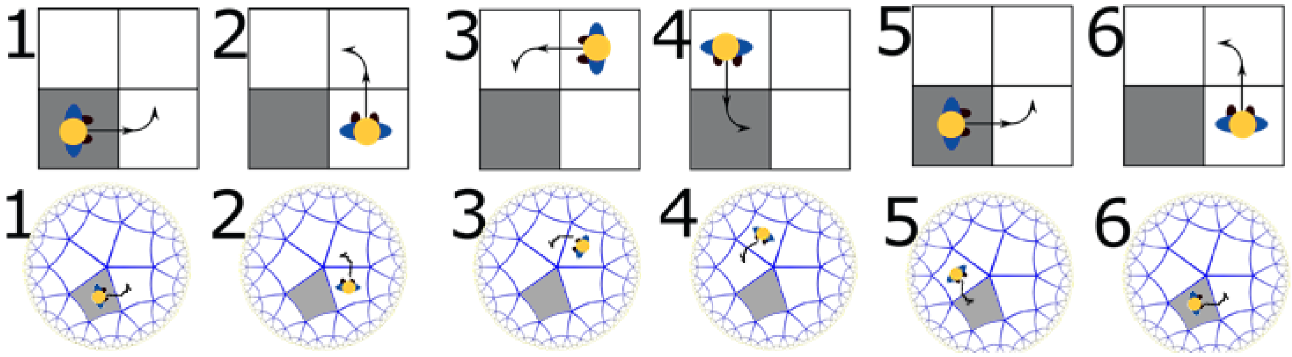


Figure 4: Top row: A user moving their body through a physical 2×2 Euclidean square grid. Bottom row: The movements are mapped to a square tiling in the hyperbolic plane, with five squares meeting at a vertex. Reproduced with permission [Jochems 2023].

other features such as a map of the environment. The representation used throughout this paper has been presented in previous work [Celinska-Kopczynska and Kopczynski 2021] and we give a brief overview here.

First, note that the tiling can be reduced to a graph representation with each square represented by a node, and two nodes connected if their squares are adjacent. A spanning tree can be built on this graph, see Figure 3. As the tree structure guarantees a unique shortest path to any node, this allows an indexing of the nodes and thus easy access to a square via the index of its node. A discrete representation like this is preferred, because numeric coordinate representation in one of the two-dimensional hyperbolic models of Figure 2 introduces numeric precision issues.

Following [Celinska-Kopczynska and Kopczynski 2021], the representation of a step sequence is as follows: squares are first indexed by a choice of branch from the starting point: North, West, South, or East. This is followed by a series of directions: Forward, Left, or Right. Here, moving backward corresponds to deleting the last step, see Figure 3. As no shortest path includes a backward step and the squares are indexed by their respective shortest path, we exclude a specific backward step. However, this means, that the first step needs a different representation since there are four valid directions of movement from the origin rather than three for all subsequent steps.

In building the data structure, two algorithms are used to ensure that the sequence of steps remains unique. For instance, taking a right step from the node Nr in Figure 3 will switch to the E branch and collapse the invalid index Nrr to the valid index El, indicated in the figure by a small red arrow.

The first algorithm determines whether a step is valid. That is, it decides whether a step stays on the current branch of the spanning tree rather than moving to a different branch. This decision works as follows. A forward step is always valid. A right step is invalid if and only if the previous step was also a right step. A left step is invalid if and only if there is no right step between it and the last left step.

The second algorithm normalizes an invalid sequence of steps into a valid one, by switching to a different branch of the tree if necessary. The normalization algorithm applies the following transformations:

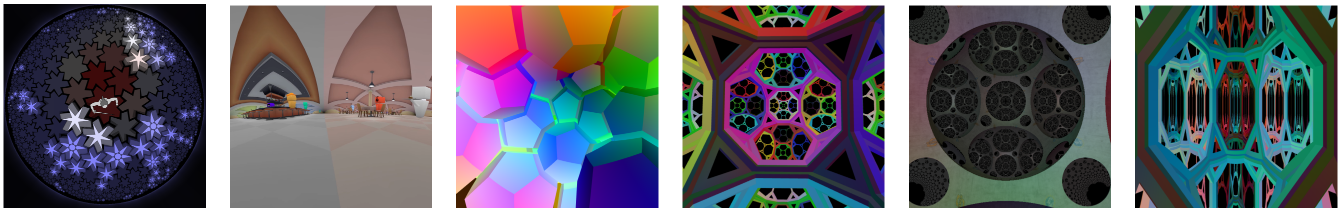
$$\begin{aligned} \{x, r, r\} &\rightarrow \{r(x), l\}, \\ \{x, l, l\} &\rightarrow \{l(x), r\}, \\ \{x, \{r, f\}_n, r, r\} &\rightarrow \{r(x), l, \{f\}_n\}, \\ \{x, l, \{f\}_n, l\} &\rightarrow \{l(x), \{r, f\}_n, r\}, \end{aligned}$$

where $r(x)$ rotates the previous step, either branch or direction, to the right, and $l(x)$ is a rotation to the left. This normalization ensures that the sequence of steps corresponds to a connected and unique series of edges of the spanning tree, starting at the point of origin. It allows for instance the population of the virtual environment, see Section 5.

3 RELATED WORK

Virtual environments including hyperbolic spaces have been included in a few previous works, see Figure 5. Those are set in two-dimensional hyperbolic space $\mathbb{H}^2$, three-dimensional hyperbolic space $\mathbb{H}^3$, or a combination $\mathbb{H}^2 \times \mathbb{E}$, where the floor is a two-dimensional hyperbolic space and the height-dimension is regular Euclidean. Skrodzki [Skrodzki 2021] provides a general review of illustrations of non-Euclidean geometry in VR.

HyperRogue is a game set on the hyperbolic plane [Kopczyński et al. 2017]. The player is shown a top-down view of the game world, rendered using the Poincaré disk model, see Figure 5(a). It is a rogue-like, turn-based game in which the player moves the character from one tile to a neighboring tile every turn or executes an attack on an adjacent tile. *HyperRogue* gives the player an idea of what it is like to navigate tilings of the hyperbolic plane. While a VR mode for *HyperRogue* exists using $\mathbb{H}^2 \times \mathbb{E}$, it only allows for teleportation as a locomotion technique. This keeps the user static and does not use the move area to connect physical and virtual space.



(a) Hyperrogue (b) Hyperbolica (c) HyperNom (d) HypVR (e) HypVR (RayMar.) (f) HypVR2

Figure 5: Screenshots of VR environments, experiences, and games using hyperbolic geometry.

Hyperbolica is a game set in a continuous three-dimensional hyperbolic space [CodeParade 2022]. It is a 3D game that supports VR, see Figure 5(b). While *Hyperbolica* is set in $\mathbb{H}^3$, it uses $\mathbb{H}^2 \times \mathbb{E}$ for the underlying physics. In the game, movement by walking is limited and locomotion is realized via teleportation for larger distances. This disconnects the user and hinders full immersion. *Hyperbolica* illustrates effects such as holonomy when the user walks around but does not use it to connect physical walking with movement in the virtual world.

The task in the VR experience *Hypernom*—see Figure 5(c)—is to 'eat' all cells of a regular 4D polytope, which is radially projected into the three-sphere $\mathbb{S}^3$ [Hart et al. 2015]. The user moves through the environment by rotating their head in the physical space. Based on the unit quaternions representing the headset's three-dimensional rotation, the player's physical movements are translated to moves in the virtual space. While this provides a high degree of immersion, no movement with the entire body is possible and the experience is limited to 'eating' one polytope.

HypVR [Hart et al. 2017a,b] is a hyperbolic space simulator incorporating several aspects that we will use within *Holonomy*. In *HypVR*, users can explore both $\mathbb{H}^3$ and $\mathbb{H}^2 \times \mathbb{E}$, see Figures 5(d) to 5(f). However, it shows a continuous space the user cannot fully explore. Movement in the world is possible and the user can utilize holonomy similar to the usage in our environment—see below—to access the entire hyperbolic disk or sphere. However, the spatial limitations of the physical space are not reflected in the virtual display, causing a disconnect between virtual movement possibilities and physical walking limitations. As all user movement happens in a pre-rendered cube, the population of the worlds by different objects becomes impossible, as shown by Skrodzki [Skrodzki 2021].

As stated in Section 1, our virtual environment overcomes the limitations of previous work. In particular, we enable users to freely walk for infinite exploration, eliminating the need for teleportation. Furthermore, we provide a populated space that can be used for various application scenarios.

4 *HOLONOMY*—THE VIRTUAL ENVIRONMENT

In this section, we describe several design choices made in the development process of *Holonomy*. First, we assume the move area to be roughly a square of at least 3×3 meter unobstructed physical space, subdivided into nine 1×1 meter cells, confer the floor in Figure 1, right. The virtual environment is designed with these proportions in mind and it currently does not include support for other grid sizes. The user's walking in this physical space is mapped to movement in the virtual hyperbolic space.

Find the flag.
#Steps:2
Path: Sr
Time:00:09

Figure 6: The user interface shows the current status and the mini-map, helping with navigation.

We have chosen to use a hyperbolic square tiling with five squares meeting at every vertex. This has two major benefits over other tiling options. First, squares map nicely to Euclidean space. Hence, the 3×3 squares in the move area can be easily aligned with the virtual ones. Second, five meeting squares is the smallest number to exhibit hyperbolic geometry. Thus, the user can more easily perceive the holonomy effect. With more squares meeting at a vertex, the user has to run through several extra rooms around a vertex to return to the original square. This would make it less likely that they actively experience holonomy, as the phenomenon cannot be perceived immediately but only by keeping track of movements.

As only 3×3 squares in the move area are physically accessible, we also only render these accessible squares virtually. While the mini-map indicates that the user is moving in an infinite environment, see Figures 6 and 12, limitations are shown in the virtual environment. As an application scenario, we chose to render a virtual, hyperbolic forest, therefore, the limitations are rendered in the form of hedges, see Figure 11. These coincide with the move area's boundaries, providing a safe exploration environment.

To test the capabilities of our virtual environment, we confront users with objectives to follow. One objective prompts the users to navigate to certain squares. Usually, these squares are initially out of reach, as they are outside the accessible 3×3 grid. However, applying the holonomy concept shifts different squares into the accessible area. Users can visit any cell in the hyperbolic grid by continually applying these transformations, including their objective.

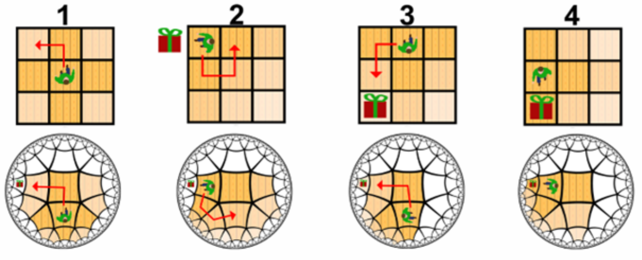


Figure 7: The top row indicates the user walking in the move area. The bottom row indicates how the user and the objective move within the virtual hyperbolic space, those squares accessible by the user are shaded.

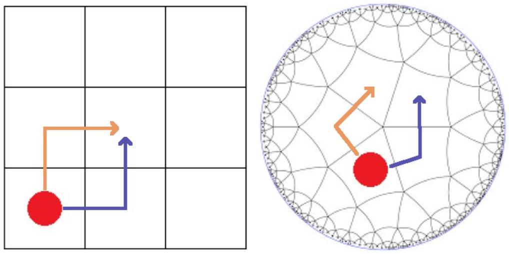


Figure 8: Moving along the blue and orange arrow in the physical space leads to the same position in the physical space (left), but not to the same position in the virtual space (right). Replicated with permission [Slotboom 2023].

Consider Figure 7 for an example of applying holonomy to reach an objective. Here, the objective for the user is to access the square marked with the icon showing a present. Initially, this square is out of reach. By walking a full circle in the move area, the holonomy concept causes a discrepancy between the physical and virtual position of the user as illustrated in Figure 4. This shifts previously inaccessible virtual squares into the accessible area for the user to reach them. Repeated application of this principle lets the user reach any position in the infinite virtual hyperbolic plane, while only walking within their 3 × 3 move area.

To support the users in fulfilling an objective, we provide them with navigational tools. Namely, we use a Poincaré disk model to show the user a larger part of the virtual, hyperbolic world expanding from the currently accessible area. Each square in the mini-map is colored according to its distance to the closest objective, where red colors are close to (hot) and blue colors are far from (cold) the goal, see Figure 12, second from right. This coloring represents a form of an unobstructed Manhattan distance on the tiling. Additionally, we display the shortest path the user needs to walk to return to a reasonable distance to the objective whenever they stray away too far. This prevents users from becoming completely lost, see Figure 12, rightmost.

For the positioning of the mini-map, we offer two different views: One in the HUD and one on the VR controllers, as a map to be held by the users. In an initial user study comparing these different views, no link between them and the navigational performance was found [Jochems 2023]. However, different users did prefer one mode over the other. Hence, we offer both map views as a choice.

5 CHALLENGES AND SOLUTIONS

Implementing the environment with the features described above poses several technical challenges. In the following, we list these and our respective derived solutions. We developed the environment within the *Unity* game engine and tested it in VR on an HTC *Vive* and a Meta *Quest* headset. Additionally, the environment offers a pancake mode to explore with a screen and a controller. However, the main feature of having the user use physical walking as a control mode to navigate the virtual environment naturally only comes within the VR setup.

Note that all challenges contribute to the usability of this versatile environment. As it is based on the unfamiliar geometry of hyperbolic space, indicating direction helps new users to navigate the virtual environment. A correct rendering allows us to capture and visualize the effects of hyperbolic geometry. A solution for the population of the virtual environment makes it a versatile tool, which can then be used to, for instance, represent information, provide a space for digital co-creation, or play games. We will discuss these aspects in detail in the following.

5.1 Indicating directions

Note that for a given objective, the target tile might not only lie outside the currently accessible area but even outside of the tiles currently displayed on the mini-map. Thus, it is necessary to indicate the general direction of such objectives to the user. To do this, the user's and the target tile's locations are converted into two points in Minkowski hyperboloid coordinates [Kinsey et al. 2011, Chapter 3.6]. Using these coordinates, the direction can be found by taking the x and y coordinates of both points and computing the angle between these two points in two dimensions. Since the angle is only used as a visual indicator on the mini-map, it does not have to be very precise. Therefore, the usual mathematical inaccuracy of this process is not an issue.

5.2 Rendering

The process of rendering a hyperbolic tiling using an Euclidean render engine presents challenges arising from the need for diverse tiles to coexist within the same Euclidean space, as depicted in Figure 8. Namely, two different tiles in the hyperbolic world can occupy the same coordinates in the Euclidean representation. Following the orange or blue arrow, respectively, lets the users stand in the same physical square, but has them end up in different virtual tiles. Going around the vertex via the left (orange arrow) should thus show a different tile than going around via the right (blue arrow). This can be seen in Figure 11, where the red flag is visible around the right view of the tree, but not around the left.

Our first solution to this problem was to divide the grid into *sub-grids* and to establish rendering *portals* that connect the sub-grids to the primary grid, see Figure 9. The view according to the orange arrow of Figure 8 would thus show a portal rendering a 2 × 2 geometry somewhere far away in the scene, shown in the second column of Figure 9. The same would happen for the blue route. As the phenomenon is also present in the 2 × 2 grids, these would be

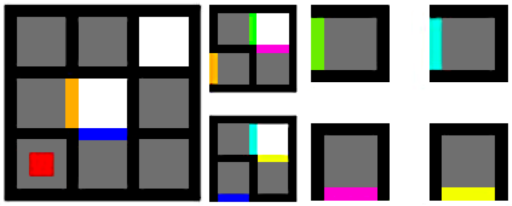


Figure 9: Splitting the 3×3 grid into subgrids, with the portals to connect them shown as colored edges. Replicated with permission [Slotboom 2023].

split again, leading to four more portals and geometries, see the right of Figure 9. Overall, depicting the hyperbolic space in this fashion calls for rendering not one, but seven views. This quickly becomes a performance issue in VR, where high frame rates are required for a comfortable user experience.

To overcome this problem, instead of using expensive render textures, we use *stencil polygons* [Neerdal et al. 2019] to create the portals. Stencil polygons are transparent objects that write a value to the stencil buffer, which can be read to determine whether an object placed behind the stencil should be rendered. This allows an object to be placed in the same position as another while still only being visible when looking through one stencil polygon and not the other—that is, being visible through the orange but not the blue stencil in the context of Figure 9. This reduces the render load drastically as only one render queue is necessary, instead of one render queue per portal. For all technical details of this implementation and a benchmark of time game in rendering, we refer to a corresponding tech report [Slotboom 2023].

5.3 Populating the virtual space with objects

An empty virtual environment is not of much use, as all tiles look the same, aside from their colors, see Figure 5, four rightmost images. This also hinders effective navigation due to the absence of meaningful landmarks. Only population with objects representing, for instance, data, introduces the possibility of integrating the environment in a specific application. It will help to differentiate the different tiles.

The general challenge for populating is, however, regarding previous work, that adding "such objects would call for tremendously different rendering algorithms" [Skrodzki 2021, p. 212]. Specifically, recall the exponential growth of a hyperbolic disk regarding its radius, which can lead the user to potentially visit thousands of distinct tiles within ten steps from the origin. Manual design becomes infeasible for such numbers of tiles. Therefore, we employ a *Wave Function Collapse* (WFC) algorithm [Gumin 2016] to procedurally populate the virtual environment.

The specific algorithm we implement is similar to the WFC algorithm examined by Karth et al. [Karth and Smith 2017] and presented as Algorithm 1. We choose it, as it has been proven to have great potential for generating populated environments [Cheng et al. 2020]. Our algorithm first collects all tiles around the user. It is then executed on all uncollapsed tiles, that is, those tiles without an object. While we do not use a grid, we can get the neighbors per tile based on the underlying graph structure, see Figure 3. This enables us to impose constraints on the neighbors of each tile.

Algorithm 1 Wave Function Collapse (WFC)

```
while an uncollapsed tile exists do
    Tile $T$ ← GET any tile with lowest entropy
    Object $O$ ← Collapse $T$ to show any of its possible objects
    for all neighbors of $T$ do
        Remove objects incompatible with $O$
    end for
end while
```

Once a tile has collapsed into an object, it will impose constraints specific to that object onto its neighbors. Constraints can limit the set of possible objects within a tile or force neighboring objects to connect in a certain orientation, like a jigsaw puzzle. These objects together form a more immersive environment for the user than an empty world [Rijsdijk 2023].

Constraint solving conflicts could prevent stock WFC to finalize, and good implementations aim to mitigate them as much as possible. They arise whenever a tile has no possible objects to collapse into, as too many constraints have been imposed upon it by its neighbors. We reduce the chances of a conflict by collapsing tiles according to increasing entropy [Karth and Smith 2017]. If one occurs, we backtrack.

Using this algorithm creates an initial population of the virtual environment. See Section 6 for an illustration within an application example. Naturally, this initial population can be manually edited by a designer, who can easily add, move, or change specific highlights or points of interest, without going through the laborious task of designing "all tiles" for the environment.

5.4 Navigational Guidance

Since navigation in a hyperbolic space can be confusing, it is convenient to provide ways to guide the user in the desired direction. This also aids the users in learning the mechanics of the hyperbolic virtual environment. One way is to show them the shortest path to the objective. Yet, the computation of the shortest path has to take into account both the currently accessible tiles and the entire hyperbolic virtual space. Deriving the shortest path under only one of these circumstances is relatively simple. However, combining both of them results in a more complex problem that requires a new way of finding the shortest path. The main obstacle here is that the user's orientation has to be taken into account as it shifts the accessible tiles, confer Figure 7.

Holonomy models the problem as a graph so existing graph traversal algorithms can be used. The attributes used to define each node encompass all possible locations in the virtual space, the move area, and the user's respective rotation. They allow a node to calculate what its neighboring nodes are lazily, without requiring any past knowledge. Storing a graph of all locations with a decent radius around the starting point is not feasible because of the exponential growth of the hyperbolic space regarding the distance from the origin. Therefore, lazy generation of the graph is essential.

We use the A^* algorithm as the shortest path algorithm on our graph. The A^* algorithm is a variation of the Dijkstra shortest path algorithm that adds a heuristic to the traversal process to guide the node expansion. The performance of A^* depends on the heuristic that is used [Hart et al. 1968]. The heuristic we employ is the length of the shortest path in the two-dimensional hyperbolic space, without regarding the currently accessible tiles. This is a relaxed version of our problem and is thus admissible. It guarantees that A^* returns an optimal path [Hart et al. 1968].

To ensure quick user feedback, *Holonomy* keeps the runtime low by switching the shortest path algorithm depending on the user's distance to the nearest objective. When the user is close to the objective it uses the A^* algorithm. If the user is far from the nearest objective, it switches to an *anytime* variant of the A^* algorithm. This variant tries to approximate the start of the shortest path but does not guarantee that the returned path will end at the objective. However, complete paths are unimportant for larger distances, as the user cannot see further than three tiles away from their position on the mini-map, confer Figure 12, rightmost. This variant still uses the same A^* algorithm at its base but has an extra parameter representing a time limit. The algorithm keeps track of the path that gets closest to the objective according to the heuristic. When the time limit is reached and the algorithm has not found the optimal path, it returns the saved path. In our preliminary experiments, for a returned path of length n, the algorithm has always returned the first n steps of the optimal path [Snellenberg 2023].

A user of the environment might want to navigate to several objectives within the environment. Calculating the shortest path between multiple locations in *Holonomy* is not as straightforward as a normal implementation of the *Traveling Salesperson Problem* (TSP). The shortest path algorithm does not consider what rotation (both in the virtual environment and the physical move area) and position in the move area the user reaches the location in. However, those properties influence the paths that can be taken from that location. Consequently, if the shortest path from point A to point B goes through C, then the shortest path to C is not necessarily contained in the shortest path from A to B. Thus, the described shortest path algorithm violates the optimal substructure property which is a requirement to use existing TSP solvers. This issue can be solved by modeling the problem as a set TSP problem, where all possible states of a certain objective location constitute a set. In the case of *Holonomy*, each location is linked to 144 states since the user can be in one of 9 move area states and one of 4 rotations in both the move area and the hyperbolic plane. However, the size of a set can be reduced to 9 by taking advantage of the symmetries present in the environment.

6 PROOF OF CONCEPT: A NAVIGATION GAME

To illustrate the capabilities of our virtual environment, we implemented a VR treasure hunt game. The game is set in a forest and asks users to navigate to certain objectives. To reach these, the users get varying degrees of support. We decided to use simplified natural-looking objects to populate the world, examples of which are shown in Figure 10.

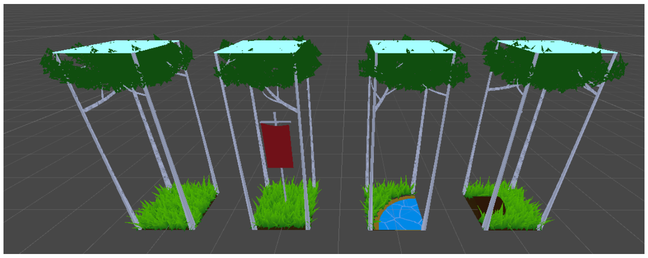


Figure 10: Four examples of natural objects used to populate the world. The tree trunks in each corner of each square serve as natural separating pillars.

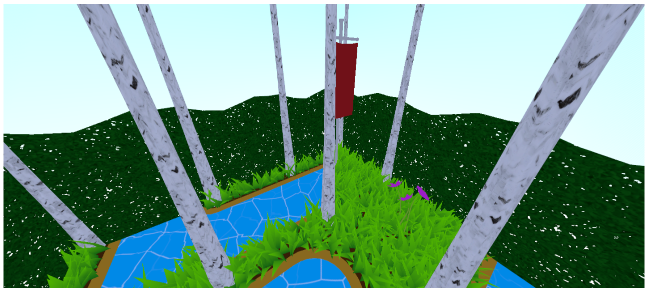


Figure 11: An Example of the world as seen in VR.

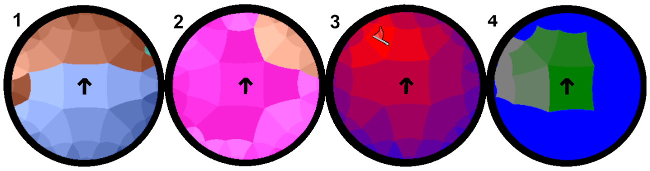


Figure 12: Images 1 and 2 display two different cases of biome propagation in the hyperbolic grid. Image 3 shows an instance of hot-cold coloring in the minimap. Image 4 shows an instance of the shortest path to the closest objective.

Within this forest environment, the objects serve as orientation landmarks. Aside from flowers occupying a single tile, as shown in Figure 1, logs occupy two neighboring tiles, while creeks continue over several tiles. This contributes to highlighting the hyperbolic effects of the environment, as creeks and logs suddenly vanish behind one side of a tree, confer Figures 1 and 11. Furthermore, the trees we place at the vertices where five squares come together serve as natural separating pillars with diverging views around their left or right side, see Figure 11.

In the WFC algorithm, we can introduce a hierarchy [Alaka and Bidarra 2023] to group similar objects. Here, we limit the hierarchy to one high-level layer on top of the objects, representing *biomes*. We use a simple propagation algorithm to generate biomes, the result of which is shown in the left two images of Figure 12: If a square has no biome, we assign one and set its propagation depth to a value greater than zero. If a square collapses and its depth is greater than zero, it propagates its biome to neighbors without a biome and decreases its propagation depth by one. The type of biome then determines the possible objects for the tile.

Figure 13: Left: a chest; Right: the required key.

The navigation aspect discussed in Section 5.4 provides a tutorial tool for users to learn how to navigate in the virtual environment. However, it also allows the users to compare the number of steps taken toward the goal with the optimal number represented by the shortest path. It motivates them to improve their navigational skill and to re-play a level.

Two game modes are currently implemented. The first asks a user to reach a flag, as shown in Figures 1, 10, and 11. This serves as a first-stage navigational task, aided by navigational cues. The second mode requires the user to collect several keys and bring them to a chest, see Figure 13. This requires finding the optimal way including several objectives, along a TSP problem as discussed in Section 5.4. To learn the basic functionality of the game, users are walked through a tutorial level.[2]

We let several users interact with our virtual environment based on this proof of concept. One user study with 23 participants explored the impact of the procedurally generated environment on the user's immersion [Rijsdijk 2023]. This study reports an increasing immersion due to the population of the environment. A second study with 30 participants investigated the placing of the mini-map on either the HUD or the controller of the user [Jochems 2023]. The findings show that the optimal position mostly depends on the preference of the individual user. Furthermore, the prototype was shown at the *Dutch Game Gardens* event, where feedback from 12 more users was gathered qualitatively based on the following questions:

1. What is your spatial perception of the environment? How does it feel?
2. How did you approach the navigation task?
3. When navigating: how did the mini-map help (or not) in completing this task?
4. When navigating: how did the colored poles help (or not) in completing this task?
5. What applications do you see for this game?
6. What other suggestions or comments do you have?

All user groups were comprised of people with (extensive) prior VR experience and people that never used VR before.

From these interactions, aside from the validated findings reported [Jochems 2023; Rijsdijk 2023], we made several anecdotal findings, that warrant to be followed up on. First, it does take the users some engagement with the platform to overcome their initial confusion, which is rooted in the underlying hyperbolic geometry. Three participants from the qualitative study described the environment as "confined" or "claustrophobic". One of the participants stated regarding the first question: "It feels claustrophobic and a bit disorienting or even frightening, but also really intriguing and gives a great curiosity for exploration." Three other participants described their spatial perception of the environment as "good" with one user specifying: "Intriguing as soon as I saw the effect of items 'around the corner'."

In contrast to the initial spatial perception, controlling movement in the environment came naturally to all participants, as they walked and moved to explore it. Most noteworthy is that due to the natural walking movements, no participant had to abort testing the prototype due to VR sickness.

Navigation in the environment seems to work well for some users, others need more time. However, most players improve their navigational abilities through a few initial interactions. Overall, the proof of concept showed the feasibility of navigating and exploring a potentially infinite virtual world via physical movement in a restricted move area.

7 CONCLUSIONS: IMPACT AND OUTLOOK

We have presented *Holonomy*, a virtual environment based on the mathematical concept of hyperbolic space. Unlike other virtual environments, it allows users to explore a potentially infinite virtual space solely by walking in a confined physical move area, without the need for artificial locomotion such as teleportation. We described how to overcome several technical challenges related to rendering and populating the space and how to compute the shortest path to an objective to initially guide the user before they master navigation in the space. In addition, we give two examples where the application of such an environment has an impact.

First, hyperbolic geometry, leaving behind the familiar Euclidean principles, presents a significant challenge for learners due to its unintuitive nature [Armand et al. 2019, Chapter V]. Aspects such as holonomy, the concept of how a geometric object evolves as it moves along a curve, can be difficult to grasp without tangible visualizations or hands-on experiences [Shapovalova et al. 2020, p. 79]. Traditional educational methods often struggle to convey these abstract ideas adequately. The virtual environment presented in this paper offers a promising base for building future learning techniques. We hope learners can develop a deeper intuition for the underlying principles by navigating through hyperbolic spaces and observing how shapes and objects behave within these environments. The educational value of our platform was already exemplified by incorporating a corresponding task in the 2023 edition of the "Mathekalender," a project for mathematics education[3]. Thus, we see great potential impact in implementing our platform within an educational context.

Second, integrating representations of hyperbolic geometry into VR environments offers a novel approach to studying navigational challenges in psychological research. Unlike traditional Euclidean maps, which study how participants are accustomed to navigating, hyperbolic spaces present entirely new and unfamiliar landscapes. By immersing participants in hyperbolic environments through VR,

---

[2] Find a video of the tutorial with explanations here: https://youtu.be/wL37nyJ_ECM.

[3] Find the task formulation here: https://www.mathekalender.de/wp/calendar/challenges/2023-10-en/.

researchers can create unique navigational challenges that require individuals to adapt their spatial reasoning and cognitive strategies. Observing participants in their learning process on navigating might offer insights into how navigational strategies are formed and what role human cognition and perception play in these strategies. In these experiments, the immersive nature of VR allows for controlled setups with various hyperbolic geometries, enabling researchers to investigate how different features, such as tiling variations, influence navigational performance and cognitive load. This opens avenues for studying spatial cognition and advancing our understanding of human navigation in unfamiliar and complex environments.

Finally, considering these and other potential use cases, our virtual environment has technical challenges that remain unanswered. One example is the rendering of light. While hyperbolic space is a well-described mathematical concept, light propagation in a virtual environment context has not been explored. Regarding our setup, spotlights in one square tile can affect neighboring tiles in non-euclidean ways, requiring a novel technical implementation currently unsupported by the rendering pipelines.

Another challenge is the incorporation of further physical boundaries. In our proof of concept, the physical boundaries of the move area are rendered as hedges within the virtual space. However, there are no measures in place that prevent the user from walking through these virtual boundaries, leaving the move area. Placing physical hedges, or dummies that mimic such, on the boundary of the move area would allow the users to reach out to hedges they see in the virtual world and touch them physically. The same could be done for the trees on the vertices between square tiles. This would contribute to a significantly stronger connection between the virtual and physical worlds, as suggested in previous research [Melo et al. 2020]. All these challenges are left as future work.

REFERENCES

Shaad Alaka and Rafael Bidarra. 2023. Hierarchical Semantic Wave Function Collapse. In *Proceedings of the 18th International Conference on the Foundations of Digital Games, FDG 2023 (ACM International Conference Proceeding Series)*, Phil Lopes, Filipe Luz, Antonios Liapis, and Henrik Engstrom (Eds.). Association for Computing Machinery (ACM), 10. https://doi.org/10.1145/3582437.3587209

Armand, André Totohasina, Daniel Rajaonasy Feno, and Ambeondahy. 2019. Relevance and feasibility of early teaching of hyperbolic geometry according to Henri Poincaré's model. *Educational Research (IJMCER)* 1, 4, 15–27.

Marián Boguná, Fragkiskos Papadopoulos, and Dmitri Krioukov. 2010. Sustaining the internet with hyperbolic mapping. *Nature Communications* 1, 1 (2010), 62. https://doi.org/10.1038/ncomms1063

Dorota Celinska-Kopczynska and Eryk Kopczynski. 2021. Generating Tree Structures for Hyperbolic Tessellations. *CoRR* abs/2111.12040 (2021). arXiv:2111.12040 https://arxiv.org/abs/2111.12040

Darui Cheng, Honglei Han, and Guangzheng Fei. 2020. Automatic Generation of Game Levels Based on Controllable Wave Function Collapse Algorithm. In *Entertainment Computing – ICEC 2020*, Nuno J. Nunes, Lizhuang Ma, Meili Wang, Nuno Correia, and Zhigeng Pan (Eds.). Springer International Publishing, Cham, 37–50.

Jeremy Clifton and Stephen Palmisano. 2020. Effects of steering locomotion and teleporting on cybersickness and presence in HMD-based virtual reality. *Virtual Reality* 24, 3 (2020), 453–468.

CodeParade. 2022. Hyperbolica. https://store.steampowered.com/app/1256230/Hyperbolica/

John Conway, Heidi Burgiel, and Chaim Goodman-Strauss. 2008. *The Symmetries of Things*. Vol. 10. A. K. Peters. xviii+426 pages. https://doi.org/10.1201/b21368

HSM Coxeter. 1957. Crystal symmetry and its generalizations. *Royal Society of Canada,(3)* 51 (1957), 1–13.

Linwei Fan, Huiyu Li, and Miaowen Shi. 2022. Redirected walking for exploring immersive virtual spaces with HMD: a comprehensive review and recent advances. *IEEE Transactions on Visualization and Computer Graphics* (2022).

Maxim Gumin. 2016. *Wave Function Collapse Algorithm*. personal repository. https://github.com/mxgmn/WaveFunctionCollapse

Peter Hart, Nils Nilsson, and Bertram Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4, 2 (1968), 100–107. https://doi.org/10.1109/tssc.1968.300136

Vi Hart, Andrea Hawksley, Elisabetta Matsumoto, and Henry Segerman. 2017a. Non-euclidean Virtual Reality I: Explorations of H^3. In *Proceedings of Bridges 2017: Mathematics, Art, Music, Architecture, Education, Culture*, David Swart, Carlo H. Séquin, and Kristóf Fenyvesi (Eds.). Tessellations Publishing, Phoenix, Arizona, 33–40. http://archive.bridgesmathart.org/2017/bridges2017-33.html

Vi Hart, Andrea Hawksley, Elisabetta Matsumoto, and Henry Segerman. 2017b. Non-euclidean Virtual Reality II: Explorations of $H^2 imesE$. In *Proceedings of Bridges 2017: Mathematics, Art, Music, Architecture, Education, Culture*, David Swart, Carlo H. Séquin, and Kristóf Fenyvesi (Eds.). Tessellations Publishing, Phoenix, Arizona, 41–48. http://archive.bridgesmathart.org/2017/bridges2017-41.html

Vi Hart, Andrea Hawksley, Henry Segerman, and Marc ten Bosch. 2015. Hypernom: Mapping VR Headset Orientation to S^3. In *Proceedings of Bridges 2015: Mathematics, Music, Art, Architecture, Culture*, Kelly Delp, Craig S. Kaplan, Douglas McKenna, and Reza Sarhangi (Eds.). Tessellations Publishing, Phoenix, Arizona, 387–390. http://archive.bridgesmathart.org/2015/bridges2015-387.html

Hiroo Iwata. 1999. Walking about virtual environments on an infinite floor. In *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*. IEEE, 286–293.

Scott Jochems. 2023. *Mini-map positioning for Virtual Reality environments in hyperbolic space.* Technical Report. http://resolver.tudelft.nl/uuid:9b3a815e-7366-4b08-9544-cb1ae6b78070

Isaac Karth and Adam M. Smith. 2017. WaveFunctionCollapse is constraint solving in the wild. In *Proceedings of the 12th International Conference on the Foundations of Digital Games* (Hyannis, Massachusetts) *(FDG '17)*. Association for Computing Machinery, New York, NY, USA, Article 68, 10 pages. https://doi.org/10.1145/3102071.3110566

L. Christine Kinsey, Teresa E. Moore, and Efstratios Prassidis. 2011. *Geometry & Symmetry*. John Wiley & Sons, Inc, 111 River Street, Hoboken. 186–197 pages.

Eryk Kopczyński, Dorota Celińska, and Marek Čtrnáct. 2017. HyperRogue: Playing with Hyperbolic Geometry. In *Proceedings of Bridges 2017: Mathematics, Art, Music, Architecture, Education, Culture*, David Swart, Carlo H. Séquin, and Kristóf Fenyvesi (Eds.). Tessellations Publishing, Phoenix, Arizona, 9–16. http://archive.bridgesmathart.org/2017/bridges2017-9.html

Miguel Melo, Guilherme Gonçalves, Pedro Monteiro, Hugo Coelho, José Vasconcelos-Raposo, and Maximino Bessa. 2020. Do multisensory stimuli benefit the virtual reality experience? A systematic review. *IEEE transactions on visualization and computer graphics* 28, 2 (2020), 1428–1442.

Diego Monteiro, Hao Chen, Hai-Ning Liang, Huawei Tu, and Henry Dub. 2021. Evaluating Performance and Gameplay of Virtual Reality Sickness Techniques in a First-Person Shooter Game. In *2021 IEEE Conference on Games (CoG)*. 1–8. https://doi.org/10.1109/CoG52621.2021.9619145

Jannik AIH Neerdal, Thomas B Hansen, Nicolai B Hansen, Kresta Louise F Bonita, and Martin Kraus. 2019. Navigating procedurally generated overt self-overlapping environments in VR. In *International Conference on ArtsIT, Interactivity and Game Creation*. Springer, 244–260.

Sharif Razzaque, David Swapp, Mel Slater, Mary C Whitton, and Anthony Steed. 2002. Redirected walking in place. In *EGVE*, Vol. 2. 123–130.

Joris Rijsdijk. 2023. *Exploring the Impact of a Procedurally Generated Environment on Immersion in Virtual Hyperbolic Space.* Technical Report. https://repository.tudelft.nl/islandora/object/uuid:b2bcf835-240d-420e-b2f3-79aa433dbc21?collection=education

Rik Sarkar. 2011. Low Distortion Delaunay Embedding of Trees in Hyperbolic Plane. In *Int. Symp. on Graph Drawing*. Springer, 355–366.

Natalia Shapovalova, Larisa Panchenko, and Olena Bashchuk. 2020. Hyperbolic geometry in the course of professional training of future mathematics and physics teachers. *Modern Science–Moderní věda.–Praha.–Česká republika, Nemoros* 2 (2020), 73–81.

Martin Skrodzki. 2021. Illustrations of non-Euclidean geometry in virtual reality. In *Virtual Images: Trilogy of Synthetic Realities I*, Lars C. Grabbe, Patrick Rupert-Kruse, and Norbert M. Schmitz (Eds.). Büchner, Marburg, Germany, 192–217.

Riley Slotboom. 2023. *Rendering Non-Euclidean Space in Virtual Reality Using Portals.* Technical Report. http://resolver.tudelft.nl/uuid:efc67ab5-16bb-4f01-9478-7cee03968257

Ravi Snellenberg. 2023. *Shortest Path algorithms for the traversal of an Order-5 square tiling from within a confined space.* Technical Report. http://resolver.tudelft.nl/uuid:b27fd9bd-8f8a-49b4-b1f8-5e6774a0f304

Kevin Verbeek and Subhash Suri. 2014. Metric Embedding, Hyperbolic Space, and Social Networks. *Computational Geometry* 59 (2014), 1–12. https://doi.org/10.1016/j.comgeo.2016.08.003

Jialin Wang, Rongkai Shi, Wenxuan Zheng, Weijie Xie, Dominic Kao, and Hai-Ning Liang. 2023. Effect of Frame Rate on User Experience, Performance, and Simulator Sickness in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 29, 5 (2023), 2478–2488.