CSE3000 – Research Project – Final Report (2023)
Delft University of Technology
M. Skrodzki(Responsible Professor), R. Bidarra(Supervisor) and G. Smaragdakis(Examiner)

Short Paper

Exploring the Impact of a Procedurally Generated Environment on Immersion in Virtual Hyperbolic Space

J. A. Rijsdijk

[1]TU Delft University of Technology, The Netherlands

Abstract

Virtual Reality allows for the ultimate immersion in environments not naturally encountered. Still, hyperbolic environments are extremely difficult to get used to. This paper explores whether immersion in virtual hyperbolic environments can be enhanced by introducing a procedurally generated world. We propose a Wave Function Collapse (WFC) algorithm that can divide the hyperbolic plane into distinct areas and populate the space with dynamically generated objects. This can assist players in recognising areas visited prior, which can create an intuitive feel for how hyperbolic space operates. Although 5-order square tiling is used in this paper, the proposed algorithm can be extended to other tiling approaches. Evidence suggests players in a populated hyperbolic environment need on average fewer steps to reach their objective compared to a control group with an empty environment, although the time taken for both groups is roughly the same. From this and a qualitative analysis we infer that players feel more immersed in the environment when it is procedurally generated instead of mostly empty.

1. Introduction

Hyperbolic space is a type of geometry that is different from the Euclidean geometry we experience in the real world. Developing an intuition for how it works can be quite challenging. Since Virtually Reality (VR) allows people to be fully immersed in a task [PPW97], the Holonomy [YBS*22] project was started as a software project in 2022 to help people develop an intuition for hyperbolic space. During this project, a simple VR game was developed, in which players are tasked to navigate to a destination flag through hyperbolic space. However, the explored world is mostly empty, featuring only the objective representing a goalpost the player is supposed to reach. It is therefore difficult to become fully immersed in the world.

As the entire world looks the same, navigating through this game is rather challenging without additional tools. Several mechanisms to aid the player like a minimap and hot-cold colouring had already been introduced to make navigation possible. However, navigation purely based on the environment is thus far impossible. As the environment is not used for navigation, and players rely too much on a minimap to move around, it is hard to fully immerse oneself in the hyperbolic space.

However, it is nearly impossible to design a level beforehand, as hyperbolic space grows exponentially with respect to its radius, and it is impossible to manually create an infinitely big environment. Therefore, we propose an algorithm to procedurally generate a world that inhabits this hyperbolic space.

Algorithms that dynamically generate an environment around a player that explores it have been long in development. Today, many video games use procedural content generation (PCG) to generate lively worlds for players to immerse themselves in. However, this is not yet applied to hyperbolic spaces in Virtual Reality, where immersion is key. While players exploring hyperbolic space have been studied [PHHK19], they have not yet focused on the effect procedural virtual 3D environments have on immersion in hyperbolic worlds.

Our research aims to address this gap by exploring how a world in the hyperbolic plane can be procedurally generated. The impact on navigational performance and experience of players is evaluated, by comparing players in our populated world to an empty world.

In particular, we aim to answer the following research question: Does procedurally generating an environment immerse players more compared to an empty environment in a virtual hyperbolic world? To answer the research question, several sub-questions must be answered:

- What algorithm can populate the hyperbolic space with distinct objects or landmarks?
- What objects or landmarks can be used to fill the tiles to improve immersion?
- How do players respond to a populated world?

To explore these questions, the paper is divided into the following sections. First, related work is analysed in section 2. After which, a generic Wave Function Collapse algorithm is discussed in section 3. The implementation specifically made to answer the

© 2023 The Author(s)

research question is discussed in section 4. The player evaluation conducted to verify the implementation and results are discussed in section 5. Responsible research is related back to this paper in section 6. Finally, the conclusion follows in section 7.

2. Related Work

There has been a lot of research into Immersion in Virtual Reality (VR), Procedural Content Generation (PCG) and hyperbolic space. However, the impact of a procedurally generated environment on immersion and navigation performance in hyperbolic space has not been studied. This section provides an overview of existing research to serve as necessary background to answer the research question posed in section 1.

2.1. Holonomy

This research builds mostly upon a software project called Holonomy [YBS*22] from 2022, which aimed to explore hyperbolic geometry by means of traversing a Non-Euclidean world in VR. During this project, a game was developed in which players were tasked to navigate to an objective. Moreover, players are restricted to a 3x3 Euclidean grid matching a 3x3 meter real-life perimeter, depicted in VR as walls. To explore past these walls, players have to traverse the space in such a way that different parts of the environment are rotated in.

2.2. Hyperbolic Space

Hyperbolic space is an alternative way to represent geometry. Just like in Euclidean space, it can be tiled by repeating patterns. For instance, a square grid can perfectly tile an infinite plane in Euclidean space. Holonomy uses a grid that tiles the hyperbolic plane using 5-order square tiling, which results in each vertex, or corner of a square, being connected to five squares total, as opposed to four squares that one would expect in Euclidean geometry. We can use a Poincaré map to render this hyperbolic geometry, as discussed in [KMP11] and shown in figure 1. In short, the Poincaré map is able to render an infinite plane onto a finite Euclidean circle. Due to this, cells that are even at short distances away from the cell currently rendered in the middle of the map are rendered much smaller.

Square grids in Euclidean space are easy to handle, as they can be indexed based on their position vector relative to the origin. For instance, a cell that takes 1 horizontal step and 2 vertical steps to get to from $cell[0,0]$, can be identified as $cell[1,2]$. Unfortunately, this idea does not translate directly into hyperbolic tiling as one identifier could map to multiple cells. Holonomy [YBS*22] used a graph structure to solve this, depicted in figure 1. Each cell is identified by the shortest path from the origin to it. Firstly, a cardinal direction, North N, East E, South S or West W, is assigned to a path to orientate it relative to the origin. From there on, for each step the path is appended by forward f, left l or right r. No shortest path requires the use of backwards steps. If a player walks the following path from the origin facing north: `fowards, fowards, right, left`, the player will end up in cell $Nfrl$

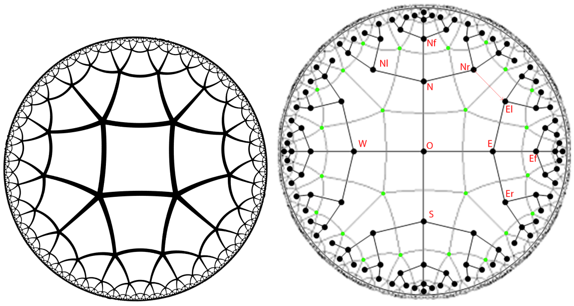

Figure 1: *On the left, an example Poincaré map depicting hyperbolic 5-order square tiling is shown. On the right, the underlying graph used in Holonomy to identify cells is displayed. Cells can be uniquely identified by the shortest path from the origin to it.*

2.3. Immersion

Immersive media allows someone to fully experience a different environment, and lose track of space and time in the real world [Rei17]. To evaluate how much one is immersed, S. Argawal et al. propose we could make use of questionnaires or physiological and behavioural measures [ASB*20]. While questionnaires held after the fact are an effective and relatively easy way to assess immersion, they could suffer from the recency effect and participants could inaccurately recall their experiences. Instead, Cox et al. have recommended eye tracking to study immersion [CCBJ16].

Several studies have been conducted on immersion in virtual spaces. One such study by R. Pausch et al. concluded that virtual reality can increase search task performance, by virtue of immersion [PPW97]. Navigation in hyperbolic space has also been studied in virtual reality. A study by V. A. Pisani et al. concluded that it is likely people are able to navigate hyperbolic spaces just as efficiently as Euclidean spaces when sufficiently immersed [PHHK19].

By making the environments more interesting, players can feel a better connection to the world they are exploring. According to a particular study by Dr R. Darken, landmarks are an invaluable tool in orientation and are thus important to increase immersion [DP01].

2.4. Procedural Content Generation

However, in most prior research mostly empty spaces, or spaces not quite related to real environments are used. This is in part due to the fact that it is hard to construct environments when the geometry is not Euclidean. In the virtual domain, Procedural Content Generation (PCG) is used to construct large or infinite worlds. Elaborate techniques have been developed to distribute landmarks such as trees in digital worlds [JNP21].

Furthermore, noise maps are used to generate natural geographical landmarks such as mountains and rivers [STBB14]. A value noise map can be generated using techniques such as Perlin noise. This noise map can then be sampled, and mapped to height values of a large subdivided plane mesh for use in an environment [Par14]. Despite its strengths, PCG using noise maps is hard to translate into Non-Euclidean space, as it relies on Euclidean noise.

© 2023 The Author(s)

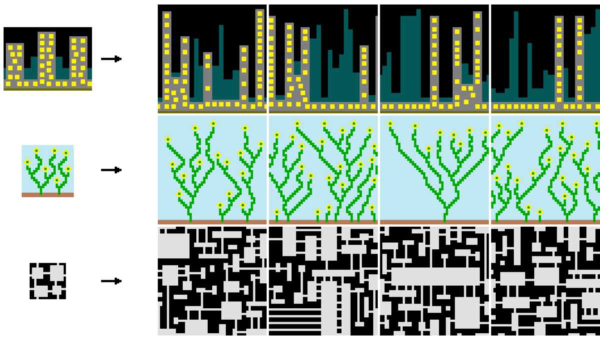

Figure 2: *Wave function Collapse, from Maxim Gumin's Github [Gum16]. Here, output images that are locally similar to the input were generated using WFC. The algorithm infers what pixels can fit together based on the input, and outputs new images that behave according to those rules.*

2.5. Wave Function Collapse

One technique that does translate rather well is Wave Function Collapse (WFC) [CHF20]. In particular, Hierarchical WFC [AB23] could be used as a basis to divide the world into distinct areas. WFC is a constraint solver algorithm, that tries to create an output based on a set of input rules, similar to how the game Sudoku works. When a cell has not collapsed into a single state yet, it is in superposition. Here it is assumed that any of its possible states could be valid. Once collapsed into a single state, the cell can impose constraints on neighbouring cells in superposition, which limits the possible states these cells can collapse into. By propagating such constraints, the WFC algorithm is able to generate complex worlds. WFC only needs information about neighbouring cells, which is easily obtainable in hyperbolic tiling, and therefore directly applies to our setting.

Constraints can be designed in several ways. For instance, M. Gumin used local similarity to an input to generate output bitmaps based on an input bitmap [Gum16]. The input was divided up into images of size $N \times N$ and the output was constructed using only those images with probabilities of appearing similar to the density of that image in the input bitmap. This leads to results as found in Figure 2. It is not always easy or even possible to make such inputs. Constraints for the WFC algorithm can be specified on a tile basis in that case, e.g. Tile A cannot be adjacent to Tile B.

3. General Approach

As described in section 2, a Wave Function Collapse (WFC) implementation was chosen to populate the hyperbolic plane. In section 3.1, we aim to illustrate a general implementation of a WFC algorithm. Section 3.2 highlights how introducing Entropy from information theory can increase the effectiveness of a WFC implementation.

3.1. Wave Function Collapse

The world that WFC generates, is made up of small sections called tiles. These tiles are created beforehand and can depict real-life objects, such as a patch of grass or a tree, see section 4.3. They are placed in a grid of cells, where each cell c has a set of possible tiles T, each having its own list of possible orientations. When a cell is being collapsed, it randomly selects any one of its possible tiles, in a possible rotation. The cell now propagates constraints to its neighbours based on the tile it chose, as discussed in section 2.5. Algorithm 1 describes how Wave Function Collapse works on a high level.

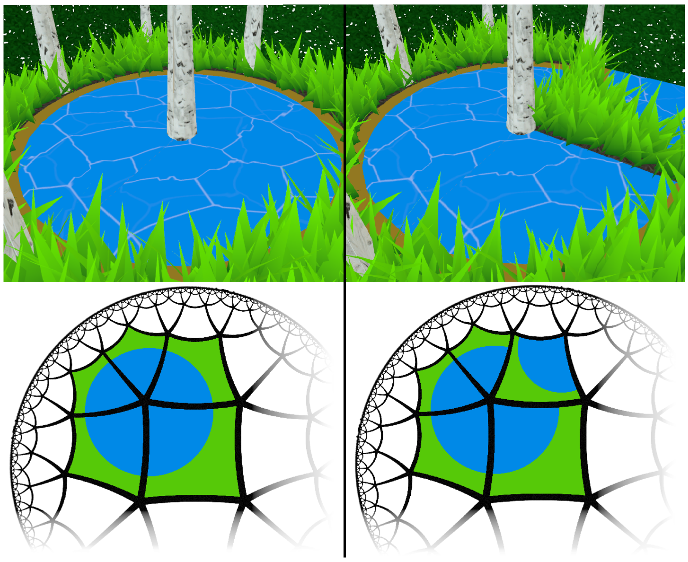

Figure 3: *On the left, an example of how orientational constraints can be used to connect tiles with distinct edges is shown. By distinctly identifying the left and right pond edges, wave function collapse can impose constraints such that left edges only connect to right edges, and vice versa. On the right is shown why it is important that tiles are connected in the correct orientation. If not done correctly, the illusion of objects spanning multiple tiles is broken.*

Algorithm 1 Wave Function Collapse

while an uncollapsed cell exists **do**
  Cell $C \leftarrow$ GET any cell with lowest entropy
  Tile $T \leftarrow$ Collapse C into any of its possible tiles
  for all neighbours of C **do**
    Remove possible tiles that are not compatible with T

Constraints can be further expanded on by introducing orientational constraints. Similar to a jigsaw puzzle, tiles can only connect to each other when orientated correctly, i.e. when the proper edges align, see Figure 3. Moreover, hierarchy can be introduced. The hierarchy will first determine what group of tiles a cell belongs to, after which it can collapse into any of those tiles. This can help designers with artistic intent [AB23], but later in section 4.4 will be explained that it is used to group similar objects in our implementation.

© 2023 The Author(s)

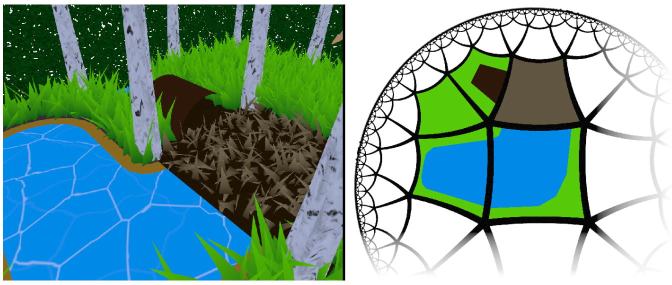

Figure 4: *Here, a contradiction arises, as we did not make a tile that connects to both a log and a river tile. Our WFC implementation opts to place a "dry grass" tile here, as this looks reasonably natural and is thus not completely immersion-breaking.*

3.2. Entropy

A major issue that holds WFC back is contradictions, which arise when a cell has no tiles it could collapse into. In this case, too many constraints have been applied to a single cell in such a way that not a single tile could possibly satisfy all constraints, as displayed in figure 4. One way to combat these is by introducing a concept called entropy [DVJ04]. Entropy in information theory aims to measure the uncertainty of a random variable. In our context, we can define the entropy of a cell H_c, where each cell c has a set of tiles T and each tile t has a probability p_t^c, as:

$$H_c = -\sum_{t \in T} p_t^c \log(p_t^c)$$

Low entropy essentially means high certainty of collapsing into one tile. By calculating the entropy for each cell, we can choose the cell with the least entropy to collapse first. This lowers the probability of contradictions arising, as imposing additional constraints on low entropy cells can result in no possible tile.

4. Implementation

In this section we explore the implementation necessary to answer the research question. Section 4.1 will cover the changes made specifically to enhance immersion by creating a believable aesthetic. Section 4.2 covers the main WFC algorithm that is used to generate the environment. 4.3 covers what tiles were used to decorate the environment, 4.4 discusses the adaptation of Hierarchical Wave Function Collapse [AB23] to create biomes or distinct areas in the environment, and finally section 4.5 discusses procedurally generated landmarks used to populate the world.

4.1. Immersion

During prior tests of holonomy [YBS*22] players reported that the experienced virtual world does not properly convey that it is possible to explore an infinite hyperbolic plane. To solve this, another theme was considered. At first, Holonomy used a "haunted house" aesthetic, as illustrated on the left of figure 9, but the solid walls gave the world a very closed-off look. Instead, we propose a "magic forest" theme as it might better convey the feeling of an endlessly explorable world. Instead of dark walls, a bright skybox and vegetation are used.

To achieve this look, several design choices were made. Firstly, walls had a blue gradient applied to them, to convey the feeling of a sky. Secondly, a canopy made of leaves and branches was created to be used in every tile, and every pole was made to look like a tree trunk. Finally, tiles are decorated on the ground with grass and other natural objects such as small tree logs or ponds, as displayed on the right of figure 9.

4.2. Wave Function Collapse

The specific algorithm we implement is very similar to the classic WFC algorithm as discussed in 3.1. However, changes had to be made to allow the research question to be answered. Firstly, the algorithm had to be adapted to work in the hyperbolic plane, and to only generate around the player. Oftentimes, WFC is used to generate a world in a finite Euclidean grid, which allows one to regenerate the world should any contradictions appear. Our particular instance demands a solution that works in infinite hyperbolic space. To achieve this, all grid positions or cells close to the player's current position can be gathered. If a tile already exists here, this tile can be instantiated and shown to the player. Otherwise, run the WFC algorithm until every single cell around the player is collapsed.

While hyperbolic 5-order square tiling is quite different from Euclidean square tiling, they both have one property in common. Each square tile is directly connected to 4 distinct neighbours, one at each edge, see figure 3. WFC constraints can thus be propagated to just these neighbours, which allows the WFC algorithm to directly apply to the hyperbolic grid. As discussed in section 2.5, constraints can be based on $N \cdot N$ patterns in sample inputs. In our case, this is challenging as a hyperbolic level editor is out of the scope of our research and thus creating inputs to generate constraints will not apply here. Instead, constraints can be manually determined on a per-tile basis.

4.3. Tiles

Tiles are the backbone of the generation algorithm, as they make up the resulting world a player experiences. In our implementation, they have the following data associated with them:

- A name, acting as a unique identifier
- A colour, which will show up on the mini-map when hot-cold colouring is disabled.
- A 3D object, which will be instantiated at run time and shown to the player. This represents the actual room or tile in 3d space.
- A weight, used in the weighted random tile selected when collapsing a cell. To calculate the entropy of a cell, as discussed in section 3.2, the weights of each tile in its tileset are normalized.
- A set of constraints.

To make sure tiles can be connected in proper orientations, each tile edge can have a unique ID string and a list of IDs it connects to. As this is done per edge, the WFC algorithm can make sure that neighbouring cells are constrained to orient themselves in such a way, that their list of IDs contains the edge ID of this tile. Tiles additionally store a list of tiles that cannot be neighbours. This constraint can be propagated by simply preventing neighbouring cells from collapsing into any tile in this list.

© 2023 The Author(s)

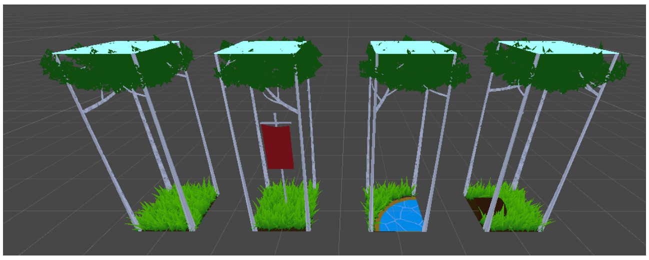

Figure 5: *Here from left to right, four tiles depicting grass, the objective, a water pond, and a log are shown. The leftmost ones have no orientational constraints, while the rightmost ones do. The log connects to other logs on the leftmost edge, and the pond connects to other ponds in such a way that only circles can be formed, as shown in figure 3.*

We can use tiles that represent objects one would expect to find in a forest, like grass, water and logs. Some tiles such as logs use orientational constraints. This allows these objects to seemingly span multiple cells, while in actuality they are made up of multiple tiles. Other tiles such as flower and grass tiles do not feature any orientational constraints, and they can connect to any other edges that have no such constraints set. In figure 5, four of our tiles are displayed, and all tiles can be found in appendix A.

4.4. Hierarchy

In nature, objects are not uniformly scattered but are often grouped together. One way to mimic this behaviour in WFC is by introducing hierarchy. We chose an approach similar to that discussed by S. Alaka and R. Bidarra [AB23], where each cell contains a tileset and each tileset contains a list of distinct weighted tiles or other tilesets. We chose to limit the hierarchy to one layer for simplicity. By first determining what tileset a certain cell should sample from before collapsing into a single tile, it is easier to design distinct areas. This way, an artist could for instance easily group tree tiles to make a forest biome, or group river and grass tiles to make a plains biome. Biomes might improve immersion, as a player could get the sense of moving through different areas in the world, instead moving through a collection of arbitrarily combined objects.

We propose a simple propagation technique to achieve these biomes, the result of which is displayed in images 1 and 2 in figure 6. If a cell has no biome, a random one is chosen. Each biome has a certain radius integer assigned to it, which is then assigned to the cell as a propagation depth. When the cell collapses into a tile, it propagates its biome to all neighbouring uncollapsed cells, setting their propagation depth equal to its own minus one. Once a cell reaches a depth of zero, it will no longer propagate its biome. This allows neighbouring tiles to randomly be assigned a new biome, starting the propagation sequence over again.

There are cases where forcing a biome is undesired. When a tile with an orientational constraint is generated next to a cell whose biome has collapsed, a contradiction arises when the biome does not feature a tile to align with the edge of the neighbouring tile imposing the constraint, as discussed in section 3.2. This can be solved by merely marking a biome as desirable, resorting to complying with the orientational constraint and only collapsing into the desired biome when no contradictions arise from it.

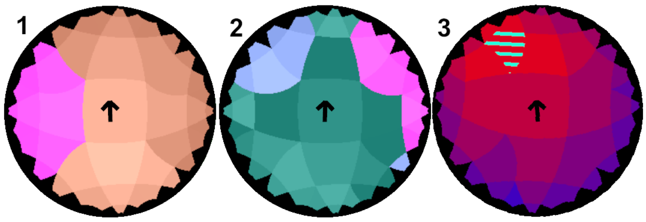

Figure 6: *Image 1 and 2 represent two instances of the on-screen minimap, where each distinct hue corresponds to a unique biome. A slight colour variance is introduced to separate the tiles on the minimap better. Image 3 represents the same minimap but rendered with hot-cold colouring instead, where red means close to and blue means far from the objective. The objective itself is highlighted using cyan stripes. The arrow in the centre points in all cases in the direction the player is facing.*

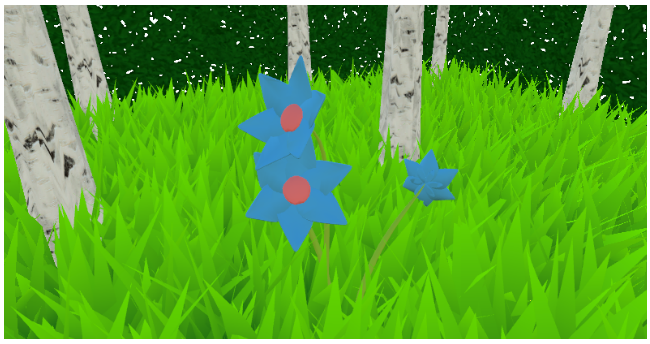

Figure 7: *A dynamically generated flower. We deterministically generate the colours, transform and flower count by seeding a random number generator with its corresponding grid location. This ensures that each flower tile has a unique flower that is the same no matter when the player decides to visit it.*

4.5. Landmark Generation

To complement the WFC algorithm, each tile can deterministically generate geometry to distinguish itself from its neighbours. For instance, when the WFC algorithm dictates that a certain cell is a flower tile, the flower can generate its own random primary and secondary colours using a random number generator seeded with the hash of the position of the tile in the hyperbolic graph, as demonstrated in figure 7. We believe that these unique-looking tiles could make a player able to better orient themselves, by recognising distinct objects they encountered prior.

5. Evaluation

This section will dive deeper into how the impact of our procedurally generated environment on immersion was evaluated. Section 5.1 will cover the setup of the player experiments. Section 5.2

© 2023 The Author(s)

covers the results of player performance and their responses to the evaluation form quantitatively, and section 5.3 explores the more nuanced player feedback received during the player experiments. The analysis is combined and discussed in section 5.4.

5.1. Experimental Setup

To evaluate the implementation discussed in section 4, a player experiment was conducted. Players were randomly grouped into two groups, A and B. Group A had a total of 15 players walk through an empty world and had a minimap with hot-cold colouring, as displayed in image 3 in figure 6. Group B had 8 players and they had the same minimap as group A. However, the world they walked through was populated using the earlier discussed WFC algorithm, as seen in figure 9.

The evaluation was made up of three steps. Firstly, players had to fill in a pre-test questionnaire where they answer how much experience they have with Video Games and VR, and how much they know about hyperbolic geometry. Secondly, the participants played through a short tutorial level to get used to the game, during which we gave them verbal instructions. Afterwards, they played Holonomy [YBS*22] on their own. Players were tasked to find the objective in three different levels, all at a comparable distance from the origin. The objectives were located at positions $Nffrf$, $Elffr$ and $Srfrl$, which represent cell positions as discussed in section 2.2, and they visually appear to the player as the second tile from the left in figure 5 or appendix A. The performance of each player was measured in time and steps taken to reach the objective. In the last step, participants had to fill in an evaluation questionnaire where they reflect on their experience, and yet again judge how well they understand hyperbolic geometry.

The tutorial was designed to make the experiment as streamlined as possible. Before the tutorial, participants put on the VR headset. The headset was only taken off after the final level was done to keep the entire experience seamless. First, players were told how the minimap works, and they are given a short explanation of hot-cold colouring. They are then instructed to take a short path to an objective as seen in Figure 8, which is placed at grid position Nlf. The grid position *Nl* is easily accessible by taking two steps, and by moving there from the origin they soon realize the objective is right in front of them obstructed only by a wall. They are then instructed to walk a counter-clockwise circle around the pillar they just moved past, which causes the virtual hyperbolic world to rotate 90 degrees. The objective is now visible within the play area, and the player is instructed to grab it.

After the experiment, players were asked to fill in a survey to describe their experiences. These questions were answered on a 6-point Likert scale, with 1 being `Strongly Disagree` and 6 `Strongly Agree`. The full survey can be found in table 1 and appendix B

5.2. Quantitative Analysis

We ran the Wilcoxon rank sum test on the responses to the evaluation survey from groups A and B to check for significant differences, and the results are displayed in table 1. Not a single difference was significant, but one that came close was the question "My navigation towards the objective went well". For this question, the median score was lower in the populated environment group compared to the empty environment group with a p-value of 0.102. This could result from the fact that a populated environment distracts from the task at hand.

However, when we look at the box plots in figure 10, we see that there is no significant difference in the average time taken. Moreover, The average steps taken appear to be lower for the group with the populated environment compared to the control group. This directly contradicts the fact that group B felt like they navigated to the objective poorly. We believe that despite the environment distracting a player, it could help build an intuitive feel for how the game works faster, thus leading to fewer steps required to reach the objective.

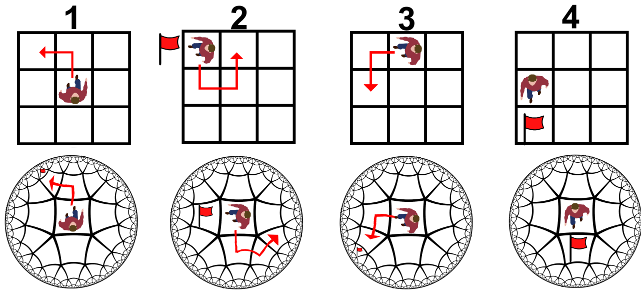

Figure 8: *Here the path walked during the tutorial is visualized. The top square grids show the player's position in the real environment, with arrows displaying the steps taken. The bottom Poincaré maps correspond to the player's position relative to the objective in 5-order square tiled hyperbolic space, with the arrows showing how steps in Euclidean space in real life map to steps in the virtual world. The flag depicts the objective the player is trying to reach.*

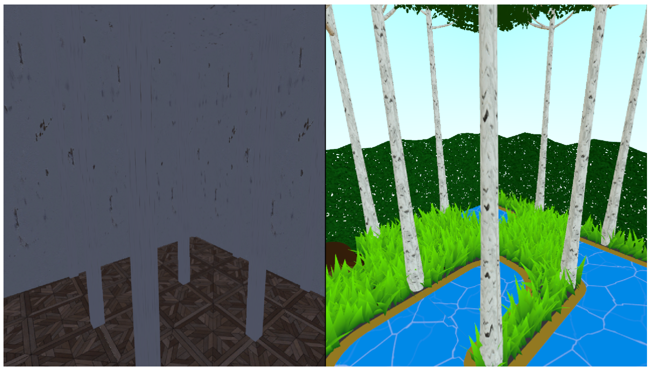

Figure 9: *On the left, an example of the VR space of case A, which still uses the old haunted house theme. On the right, an example of case B is given. Case A has no objects other than a flag objective and poles to separate the rooms. Case B is decorated using the WFC algorithm explained in section 4*

© 2023 The Author(s)

Survey Question	Group A Med	Group B Med	p-value
My navigation towards the objective went well.	4	2.5	0.102
The minimap was very helpful with finding the objective.	6	6	0.969
The minimap was easy to read and understand.	5	4	0.404
I have a good understanding of hyperbolic geometry.	3	3	0.895
I felt comfortable in the environment.	4	3.5	0.767
The environment helped me orientate.	4	2.5	0.322
The environment helped me navigate towards the objective.	2	2	0.557

Table 1: *The median responses from the player evaluation survey from group A with 15 responses and group B with 8, corresponding to the empty and populated environment respectively. The p-values were calculated using the Wilcoxon rank sum test with continuity correction. The full survey can be found in appendix B*

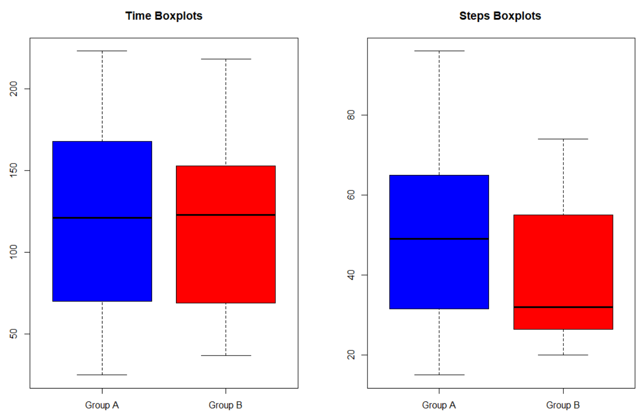

Figure 10: *Here, the box plots generated from the average time and steps taken over all three levels are shown. Blue represents group A with an empty environment, and red represents group B with the populated environment.*

5.3. Qualitative Analysis

Besides filling in a quantitative evaluation survey, players were asked to give additional feedback. Some players from group B noted that they tried to follow the rivers found in the environment. However, these rivers are completely unrelated to the objective and only serve as objects to populate the space. Additionally, some players hesitated to walk onto a tile populated with a river or a log, as they interpreted those tiles as obstacles they had to walk around. Moreover, some players highlighted that they enjoyed finding "pretty flowers", similar to the one in figure 7.

many players from group A explicitly mentioned that they completely ignored the environment, whereas no such claims were made for group B. The environment in case A was likely interpreted wrongly too. A player from group A expressed that the environment was confusing to them as it appeared to follow Euclidean geometry, while the map indicated otherwise. Additionally, a large majority of players from both groups noted looking mostly at the minimap during the experiment which makes sense as the environment itself gives no navigational clues in either case A or B.

5.4. Discussion

In our analysis, we found that players from group B, who experienced the populated environment, interacted with it more naturally and made more remarks about it than players from group A, who experienced the empty environment. Our quantitative analysis showed that players from group B required fewer steps but the same time on average compared to group A to complete the navigational task. However, group B on average reported that they felt like their navigation did not go well, compared to group A who reported that it did go well. We suggest this could have been caused by the environment slightly distracting group B.

Additionally, our qualitative analysis showed that some players from group B tried to follow rivers in the environment, even though they were unrelated to the objective. Many players from group A ignored the environment, with one expressing confusion about its geometry. Furthermore, both groups reported mostly looking at the minimap, as it was necessary to complete the given task. These findings suggest that a populated environment can affect how players interact with and are thus immersed in a virtual hyperbolic world.

6. Responsible Research

To ensure this research was responsibly conducted, several measures were in place. Section 6.1 will explore how responsible research was applied during implementation and section 6.2 will explain so for the user evaluation.

6.1. Implementation

The source code is on the TU-Delft EWI Gitlab repository. While the correctness of the WFC code was not the main concern of this paper, it is imperative that it is operational for use during user evaluation. For this reason, core implementations were tested and user verified by using example worlds.

6.2. User Evaluation

Participants were all selected on a voluntary basis. They were all given the opportunity to grasp the core principle of the evaluation, by informing them that they would explore hyperbolic geometry using a VR headset. During the evaluation, every single user was treated equally. Anyone who participated in user evaluations is uniquely identified using an ID number, which is not linked to any private information whatsoever. When a participant first arrived, a new ID was picked, and the user was required to enter this ID in both surveys. The only other private information stored is age, gender and major, as this could provide useful during analysis. During the evaluation, safety is ensured by clearing the entire testing

© 2023 The Author(s)

perimeter and clearly informing the user about the workings of the environment.

7. Conclusions and Future Work

We can conclude that a procedurally generated environment immerses players more compared to an empty environment in a virtual hyperbolic world. While none of the results from our quantitative analysis discussed in section 5.2 is statistically significant, combined with our qualitative analysis found in section 5.3, we suggest that the populated environment helps to better immerse the player in the game. Not only did players directly react to the environment when faced with natural objects such as rivers, but they also needed fewer steps on average to navigate to the goal. We infer that this could have been caused by being able to build a better intuition for the mechanics of Holonomy [YBS*22] thanks to a populated environment.

In future research, it would be beneficial to conduct more user tests to increase the reliability of our findings. Additionally, another between-subject experiment could be conducted, where players do not use a minimap. Instead, they will be fully reliant on the environment to navigate to their objective. The environments from cases A and B as discussed in section 5.1 will both have to be adapted to support this, for instance by displaying hot-cold colouring on the ground as opposed to exclusively on the minimap.

Another way to evaluate immersion could be by utilizing eye-tracking, as highlighted in section 2.3. In a between-subject experiment, players could be given 5 minutes in either a populated world or an empty world to see how they respond to those cases. Arbitrary landmarks could be sporadically placed throughout a level and be marked on the minimap, giving the players an incentive to walk around trying to find them.

Acknowledgement

We would like to thank all who have participated in the player experiments, and S. Jochems, who helped with the evaluation. His research fully dived into the effects different ways of displaying minimaps have on navigation performance. Group A in our research, as discussed in section 5 was fully covered by him. Jochems also provided the right part of figure 1.

References

[AB23] ALAKA S., BIDARRA R.: Hierarchical semantic wave function collapse. In *Proceedings of the 18th International Conference on the Foundations of Digital Games, FDG 2023* (2023), Lopes P., Luz F., Liapis A., Engstrom H., (Eds.), ACM International Conference Proceeding Series, Association for Computing Machinery (ACM), p. 10. doi:10.1145/3582437.3587209. 3, 4, 5

[ASB*20] AGREWAL S., SIMON A., BECH S., BÆRENTSEN K., FORCHAMMER S.: Defining immersion:: Literature review and implications for research on audiovisual experiences. *Journal of the Audio Engineering Society 68*, 6 (2020), 404–417. doi:10.17743/jaes.2020.0039. 2

[CCBJ16] COX A. L., CAIRNS P., BERTHOUZE N., JENNETT C.: The use of eyetracking for measuring immersion. *CogSci 2006 Workshop: What Have Eye Movements Told Us So Far, and What Is Next* (2016). 2

[CHF20] CHENG D., HAN H., FEI G.: Automatic generation of game levels based on controllable wave function collapse algorithm. In *Entertainment Computing – ICEC 2020* (Cham, 2020), Nunes N. J., Ma L., Wang M., Correia N., Pan Z., (Eds.), Springer International Publishing, pp. 37–50. 3

[DP01] DARKEN R., PETERSON B.: Spatial orientation, wayfinding, and representation. *Handbk Virtual Environ 2002* (11 2001). doi:10.1201/b17360-24. 2

[DVJ04] DALEY D. J., VERE-JONES D.: Scoring probability forecasts for point processes: the entropy score and information gain. *Journal of Applied Probability 41*, A (2004), 297–312. doi:10.1239/jap/1082552206. 4

[Gum16] GUMIN M.: Wave Function Collapse Algorithm, Sept. 2016. URL: https://github.com/mxgmn/WaveFunctionCollapse. 3

[JNP21] JÆGER P., NILSSON N., PALAMAS G.: Can't see the forest for the trees: Perceiving realism of procedural generated trees in first-person games. *EAI Endorsed Transactions on Creative Technologies 8*, 26 (2021), e4. 2

[KMP11] KINSEY L. C., MOORE T. E., PRASSIDIS E.: *Geometry & Symmetry*. John Wiley & Sons, Inc, 111 River Street, Hoboken, 2011. 2

[Par14] PARBERRY I.: Designer worlds: Procedural generation of infinite terrain from real-world elevation data. *Journal of Computer Graphics Techniques 3*, 1 (2014). 2

[PHHK19] PISANI V. A., HURD O., HAWTHORNE N., KURNIAWAN S.: Navigation by walking in hyperbolic space using virtual reality. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts* (2019), pp. 611–618. 1, 2

[PPW97] PAUSCH R., PROFFITT D., WILLIAMS G.: Quantifying immersion in virtual reality. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), pp. 13–18. 1, 2

[Rei17] REICHENBACH D.: *Losing Time and Space: Experiencing Immersion*. 12 2017, pp. 503–512. doi:10.1007/978-3-319-44418-5_39. 2

[STBB14] SMELIK R. M., TUTENEL T., BIDARRA R., BENES B.: A survey on procedural modelling for virtual worlds. *Computer Graphics Forum 33*, 6 (2014), 31–50. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12276, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12276, doi:https://doi.org/10.1111/cgf.12276. 2

[YBS*22] YARAR B., BAKKER B., SNELLENBERG R., SLOTBOOM R., LI W.: *"Holonomy": a non-Euclidean labyrinth game in virtual reality*. Tech. rep., TU Delft, 2022. URL: http://resolver.tudelft.nl/uuid:60d473f0-f327-411e-a402-95d44e27f088. 1, 2, 4, 6, 8

© 2023 The Author(s)

Appendix A: Tiles

Objective Tile

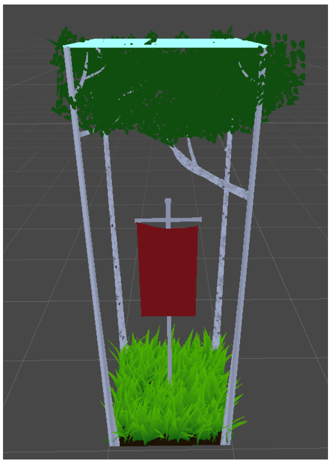

Figure 11: *This tile is the objective tile a player is supposed to reach.*

River Tiles

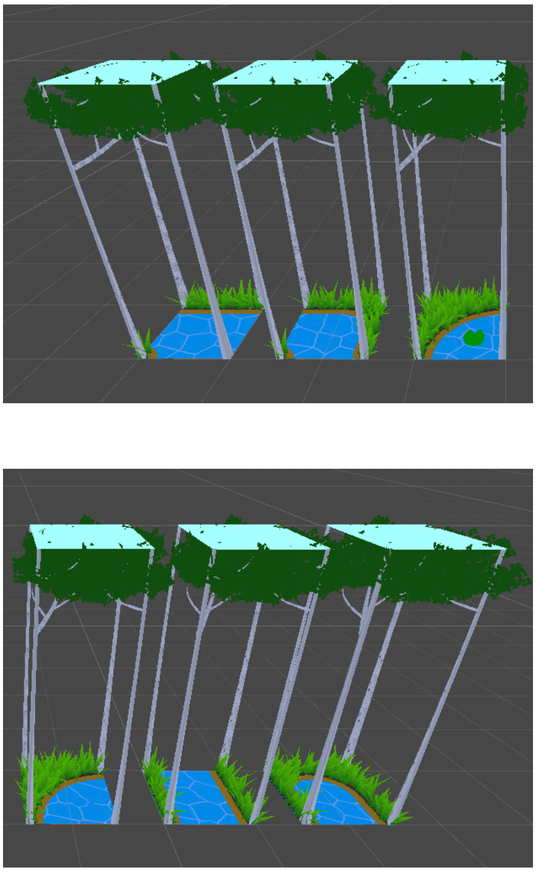

Figure 12: *These tiles are used in a river biomes.*

Forest Tiles

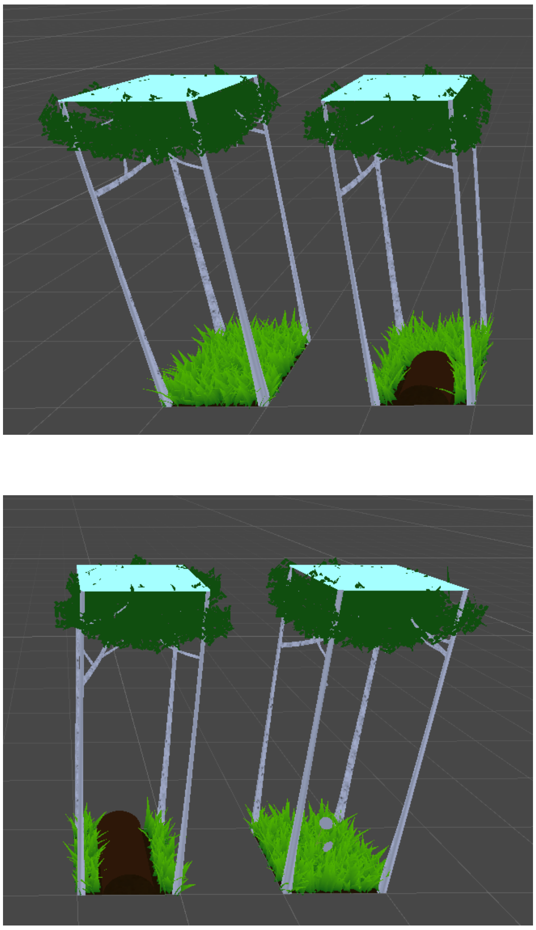

Figure 13: *These tiles are used in forest and flower field biomes.*

Contradiction Tiles

Figure 14: *If a contradiction arises, one of these tiles will appear instead.*

© 2023 The Author(s)

Appendix B: Evaluation Survey

What is your participant number *

Your answer

My navigation towards the objective went well. *

	1	2	3	4	5	6	
disagree	○	○	○	○	○	○	agree

The minimap was very helpful with finding the objective. *

	1	2	3	4	5	6	
disagree	○	○	○	○	○	○	agree

The minimap was easy to read and understand. *

	1	2	3	4	5	6	
disagree	○	○	○	○	○	○	agree

I have a good understanding of hyperbolic geometry *

	1	2	3	4	5	6	
disagree	○	○	○	○	○	○	agree

I felt comfortable in the environment *

	1	2	3	4	5	6	
Strongly disagree	○	○	○	○	○	○	Strongly agree

The environment helped me orientate *

	1	2	3	4	5	6	
Strongly disagree	○	○	○	○	○	○	Strongly agree

The environment helped me navigate towards the objective *

	1	2	3	4	5	6	
Strongly disagree	○	○	○	○	○	○	Strongly agree

Is there anything you would like to add about your experience?

Your answer

© 2023 The Author(s)