

HyperRogue: Playing with Hyperbolic Geometry

Eryk Kopczyński
University of Warsaw, Poland
erykk@mimuw.edu.pl

Dorota Celińska
University of Warsaw, Poland
dcelinska@wne.uw.edu.pl

Marek Čtrnáct
marek14@seznam.cz

Abstract

HyperRogue is a computer game whose action takes place in the hyperbolic plane. We discuss how HyperRogue is relevant for mathematicians, artists, teachers, and game designers interested in hyperbolic geometry.

Introduction

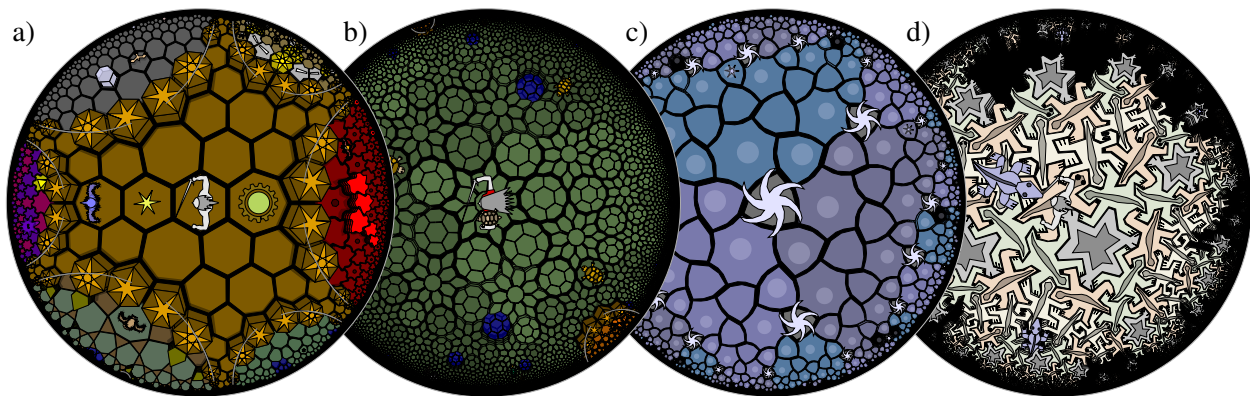


Figure 1 : Example lands in HyperRogue: (a) *Crossroads II*, (b) *Galápagos*, (c) *Windy Plains*, (d) *Reptiles*.

Hyperbolic geometry and tessellations of the hyperbolic plane have been of great interest to mathematical artists [16], most notably M.C. Escher and his *Circle Limit* series [4]. Yet, there were almost no attempts to bring more complexity and life to the hyperbolic plane. This is what our game, HyperRogue, sets out to do. We believe that it is relevant to mathematical artists for several reasons.

First, fans of Escher's art, or the famous Hofstadter's book *Gödel, Escher, Bach: an Eternal Golden Braid* [10], will find many things appealing to them. The graphical style of HyperRogue is directly inspired by Escher's tessellations, some of the music is based on Shepherd tones and crab canons if the gameplay in the given area is based on similar principles. As for Gödel, there is also one game mechanic reminding of a logical paradox: magical Orbs normally lose their power each turn, but *Orb of Time* prevents this, as long as the given Orb had no effect. Does *Orb of Time* lose its power if it is the only Orb you have?

Second, we believe that exploring the world of HyperRogue is one of the best ways to understand hyperbolic geometry. In typical courses one would learn that, for example, the sum of angles of a hyperbolic triangle is less than 180° , but it is not clear how this could be applicable, or what would be the effects for people living in the hyperbolic world; even in other existing hyperbolic games and VRs [8, 13, 17] the world is periodic and sterile. In HyperRogue, players are led to gain intuitions about hyperbolic geometry, and to pose questions about how things known from our world would work there. For example, the design of the *Hive*, one of the lands in HyperRogue, incorporates studying war strategies, like army formations, that are meaningful in wars taking place in hyperbolic plane.

Third, we believe that game design, or creating interesting maps and rule systems for games, is also a kind of art. The first version of HyperRogue had very simple rules: take Chess, but replace the “chessboard” with a tiling of the hyperbolic plane, and give just one piece to the player (player character, or PC); in your turn you can wait, move to an adjacent tile if it is not blocked by a wall, or attack an enemy on an adjacent tile (destroying it); after your move, each enemy moves (with similar choices). Enemies do not act very intelligently—they just try to get to the PC using the shortest possible route and do not avoid being attacked—but they can appear in large numbers. That would not be very interesting in a Euclidean tiling, but hyperbolic geometry makes it fun—not only because it feels very different, but also because it is possible to use the hyperbolic properties tactically, to deal with large groups of enemies. On the other hand, many of the traditional game design techniques would not make sense in a virtually infinite world. *No Mans Sky* is praised for having 10^{19} stars in its universe—our game has more space in 200 tiles around the starting point because of using hyperbolic geometry, and one would usually go further during the game. HyperRogue explores how hyperbolic geometry could be used for interesting gameplay, and how to deal with infinity. The basic game is grid-based and turn-based, but a *shmup mode* exists to explore gameplay with continuous hyperbolic space and time.

HyperRogue takes much from roguelikes, a genre of computer games that is relatively unknown to the mainstream gamers [5]. Roguelikes are often viewed as a subgenre of computer role-playing games (CRPGs), however, they have evolved separately from the mainstream; with their focus on gameplay requiring player skill and positioning instead of following a pre-written story, we view them as sharing more with board games such as Chess than with (typically very easy) CRPGs.

HyperRogue follows the roguelike tradition of permanent failure: when one loses the game, they have to start a new one—from the beginning, except from the fact that they have learnt from their failure, and this knowledge stays with them. Some players might feel that their game ends when too many enemies are generated at the same time, which is a random event not under the control. While it is definitely possible to fail due to bad luck, the best players complete the difficult *Hyperstone Quest*, which involves getting 10 treasures in every land, with high probability, in roughly two hours. It may take you many tries to win the game, but then that is because you enjoy the gameplay (otherwise you would not be playing), but you have to learn, and learning is fun. This could be seen as a metaphor of how we learn mathematics—as shown by the famous quotes of Euclid himself (*There is no royal road to geometry*) and Gauss (*The enchanting charms of this sublime science reveal themselves in all their beauty only to those who have the courage to go deeply into it.*), we have to learn the basic stuff first, then proceed to more complicated topics.¹

Geometry of HyperRogue

HyperRogue uses the order-7 truncated triangular tiling of the hyperbolic plane, also known as the *hyperbolic soccerball* tiling. One advantage of this tiling is that it is similar in spirit to the hex tiling commonly used in tactical games. This advantage is shared by the more regular order-3 heptagonal tiling; however, the heptagons in that tiling are very big, and less appropriate for a game. HyperRogue allows experimenting with other curvatures—switching to the heptagonal tiling effectively increases the curvature, while replacing the heptagons with hexagons leads to the normal Euclidean hex grid, and if we go further by replacing them with pentagons, we get the positive curvature of the truncated icosahedron (i.e., the usual spherical soccerball) or dodecahedron.

The distinction between hexagons and heptagons is important in gameplay—in some sense, the hyperbolic effects are concentrated on the heptagons. Suppose that the player character (PC) and two enemies are standing on the vertices of a triangle in the hex grid, as in Figure 2d. We cannot attack any of the enemies,

¹ The in-game tutorial serves as an introduction to hyperbolic geometry, with the challenge removed.

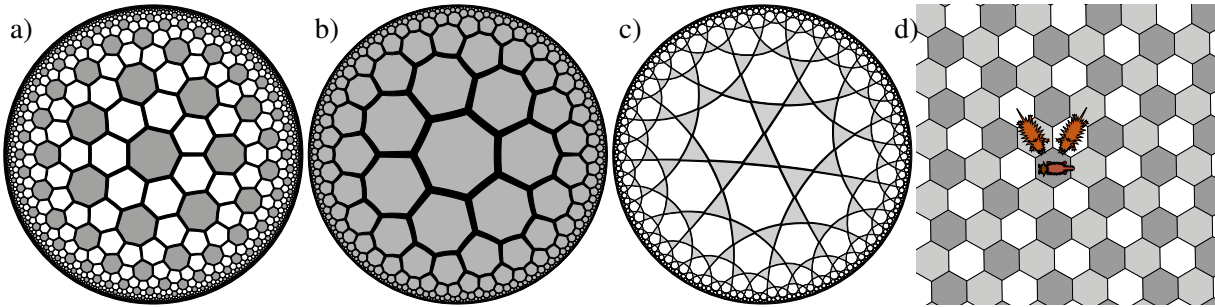


Figure 2: Tilings. (a) truncated triangular tiling, (b) heptagonal tiling, (c) triheptagonal tiling used in the Warped Coast, (d) Euclidean hex grid.

because the other one would attack us then. We could escape, but then the enemies move parallel to us, thus recreating the same configuration. However, if we were actually standing on a heptagon in the hyperbolic grid, one of the possible moves will make the enemies unable to recreate the configuration, and allow us to attack them one by one. (For completeness: on a dodecahedron the enemies will capture us quickly, and on the soccerball, the only way to avoid capture is to consistently avoid the pentagons; thus, the smaller the curvature, the easier our situation is.) This is not caused by using our particular tiling, but a general fact of hyperbolic geometry—there is one shortest path between two points, and if one is forced to avoid that path, they must use another one, which will be an *equidistant curve* rather than a straight line, and thus it will be longer by a linear factor, rather than an additive constant, as would happen in Euclidean geometry. This is especially visible in the *Land of Eternal Motion*, where the tiles fall after someone moves on them, so even a single pursuer has to take another path, and thus is completely unable to pursue the PC, even though moving with the same speed.

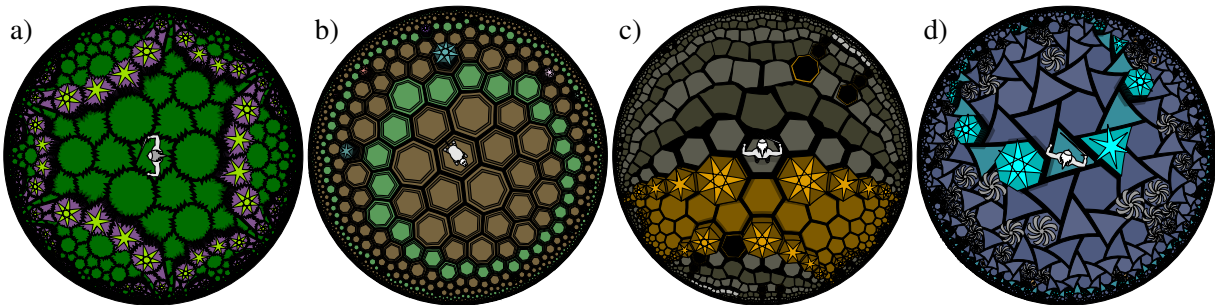


Figure 3: Curves: (a) straight lines in Vineyard, (b) circles in Hive, (c) equidistants in Ivory Tower, (d) horocycles in Temple of Cthulhu.

The well known hyperbolic curves have their discrete analogs in our tiling. The *distance* between two tiles is the length of the shortest path between them. A *straight line*, also called *geodesic*, is a path p such that the distance between tiles p_a and p_b is always exactly $|a - b|$. A *circle* is a cycle consisting of all tiles in distance (radius) r from the given point, called the center of the cycle. An *equidistant* is a curve consisting of tiles in distance r from the given straight line. A *horocycle* is a limit circle of infinite radius—a curve such that every segment of it is a segment of a circle, with larger segments corresponding to greater radii (the discrete approximation allows us to give a slightly simpler definition of the horocycle than the usual one). Two curves p and q are *concentric* if the distance from q is the same for each point in p .

Let a_d be the numbers of tiles in distance d from some heptagon in our tiling i.e., the perimeter of a circle of radius d . We believe this sequence has not been previously studied. The initial values of this sequence are 1, 7, 14, 28, 49, 84, 147, 252, 434, 749... and for $d \geq 5$, the following recursive formula holds:

$a_{d+4} = a_{d+3} + a_{d+2} + a_{d+1} - a_d$, and thus $a_d = \Theta(\gamma^d)$, where $\gamma = \frac{1+\sqrt{13}+\sqrt{2\sqrt{13}-2}}{4} \approx 1.72208 \approx \sqrt{3}$. This γ is the fundamental constant of our tiling, as the same recursive formula eventually holds when we start from a hexagon, or include the whole area of the circle, instead of just the boundary, and thus all these sequences are asymptotically $\Theta(\gamma^d)$; γ^d is also the ratio of lengths of corresponding segments of concentric horocycles in distance d , and the limit of the similar ratio for circles and equidistants, as $r \rightarrow \infty$. In the heptagonal tiling the respective value of γ is $\phi + 1 = \phi^2 = 2.618\dots$, i.e., the sequence grows exactly twice as fast as the Fibonacci sequence.

What Can We Learn, or How Does Hyperbolic Geometry Affect the Gameplay

The world of HyperRogue consists currently of about 50 different *lands*. Each land has a specific theme (treasure to collect, enemies to fight), and specific mechanics displaying different aspects of hyperbolic geometry. Regular straight lines are usually employed as borders between the lands (see Figure 1a), but curves of large radius (circles, equidistants, and horocycles of infinite radius) are used in some cases. For example, *Camelot* is a large circle without internal markings; players are required to devise an algorithm to reach the center. It is practically impossible to reach the center solely on Euclidean intuitions – by looking at a small part of a large radius curve it is impossible to tell the exact way to the center, what is the radius, nor which kind of curve it is (their curvatures are almost equal). In a large radius curve, moving into a direction which appears promising takes one back to the edge very quickly. In other cases the consecutive concentric horocycles are marked, so that finding the correct direction is easy; for example, in the *Temple of Cthulhu* every sixth horocycle is marked. Since all horocycles have roughly the same shape, moving towards the center of the Temple gives the players an impression of endless descent, a bit like in Escher’s famous *Ascending and Descending*.

A very important aspect of HyperRogue is that the number of tiles in radius r is exponential. Players quickly learn that it is practically impossible to reach a location they have been before, unless some method of marking the way has been used. Exponential space is not the only reason why this is difficult – the other reason is curvature: if you move from A to B , from B to C , and from C to A , the map will usually be rotated; this happens because the sum of external angles of the triangle ABC is not 360° . Some time later, they might be surprised about one situation where they do return – going back towards the center of a large radius curve. This commonly happens in the *Ivory Tower*, where artificial gravity pushes creatures towards the straight line bordering the Tower; after ascending in a random way, and descending back to the borderline, one typically lands very close to their starting point. Each level of the *Ivory Tower* (levels are marked with alternating colors in Figure 3c) is an equidistant curve, giving insight to how a three-dimensional hyperbolic space with gravity might operate – if one floor is two tiles high, and we have two points A and B on some level in distance d along the equidistant, the distance between the corresponding points A' and B' on the next floor along their equidistant will be $\gamma^2 \approx 3$ times larger. Then, they might be surprised yet again when they find a randomly generated large obstacle on their way to the center of a large radius curve (as happens e.g. in the *Princess Quest*) – even though it seems that circumventing such obstacles is impossible (trying other routes pushes you back into the obstacle) this can be done with persistence.

The *Yendor Quest* requires one to return to a location 100 steps away, which is practically impossible without devising some method (γ^{100} possible directions). *Yendor Quest* and *Camelot* are examples of quests which would be trivial in Euclidean. On the other hand, in the *Galápagos* one receives a quest which is easy in hyperbolic plane, but hopeless in Euclidean. The player encounters a Baby Tortoise, and has to find an adult of the same species. However, each tile in the *Galápagos* has 21 binary environmental factors, which change smoothly (most of the factors are equal for tiles which are close); and for each of 2^{21} combinations of the factors, there is a different species. In the Euclidean world, one would have to travel $\Omega(\sqrt{2^{21}})$ tiles to find the adult tortoise matching a random baby – and that is under (definitely false) assumptions that we

know where to go, that each tile contains a tortoise, and environmental factors change quickly. However, in the hyperbolic world, the task can be completed quite easily, simply by moving towards areas where more factors match.

Exponential space is also at the core of game design of HyperRogue. The play in roguelikes and similar games progresses from easy to difficult levels; on the easy levels, the player might find some powers or resources (equipment, consumables, XP, etc.), which will be useful on later levels. In this design, unlimited amount of space on each level leads to *grinding*, i.e., staying on easier levels for long times in order to collect large amounts of resources. Allowing grinding is considered a bad design, since it leads to boring gameplay, and modern well designed roguelikes avoid this, usually by restricting the size of levels. However, HyperRogue has to take a different approach. The player's progress and score is measured by the number of *treasures* they have collected; the more treasures they have collected in a given land, the more enemies appear in this land, thus making it harder and harder to collect more treasures. If permanent equipment or consumables did appear on the map, it would be possible to grind; for this reason, there are no permanent or consumable resources to be found, other than *Dead Orbs* which have very limited uses. One possible solution which combines permanent and consumable powers with infinity is to give powers only as extra prizes for collecting treasure – since the ability to collect treasures is limited, so are the prizes. The current versions of HyperRogue do not provide consumables in this way, but a permanent effect of collecting many treasures in a given land is access to magical orbs (which appear randomly and provide temporary power-ups of various kinds), and also access to more advanced lands.

Even though negative curvature is probably the thing that students remember best about the hyperbolic plane, its effect on the gameplay, or living in the hyperbolic world, appears quite limited, compared to exponential space and its consequences. One land where curvature is featured prominently is the *Burial Grounds*, where treasures have to be excavated using a magical energy sword. The problem here is that the sword has to be aimed in the specific direction (where the wall is soft), and the PC is unable to rotate their sword, which just floats in the air next to the PC, maintaining its relative angle to the PC's movement. In a Euclidean world, the sword would always face the same direction, but here, we can use the curvature to rotate the sword. Understanding that the sword will rotate when we go around a loop, and that the angle of its rotation is proportional to the area inside the loop, is essential for excavating treasures efficiently; the players will have to learn this to be successful in the *Burial Grounds*.

Procedural Generation in the Hyperbolic Plane

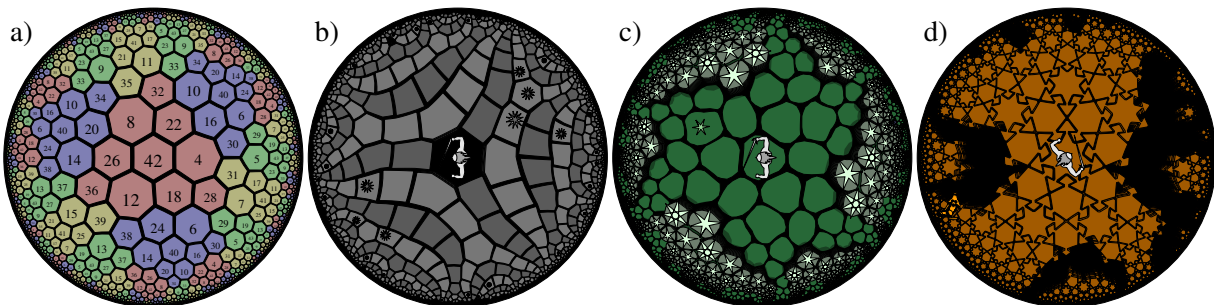


Figure 4: (a) *Zebra pattern*, (b) *Zebra*, (c) *Emerald Mine*, (d) *Dragon Chasms*.

The infinite world of HyperRogue is procedurally generated on the fly. In this section we describe the methods used to place terrain features such as walls, trees, or chasms.

Random patterns. The easiest method is to place terrain features randomly: each tile independently becomes a wall with probability, say, 50%, thus creating a maze. This method yields much more satisfactory results than in Euclidean geometry (dead ends are rare) due to the tree-like nature of the hyperbolic plane. This method can be further differentiated by altering the probability, refining the results using a cellular automaton (the *Living Cave*), or by placing large obstacles with smaller probabilities.

Periodic patterns. Figure 4a depicts a periodic encoding, which we call the *Zebra pattern*. The hyperbolic plane has been tiled in a regular way with a “triangle” consisting of 7 hexagons and 3 heptagons. Triangles have been colored into four colors, and for each colored triangle, ten codes have been given to its tiles, a different set of ten codes for each color. The coding is periodic, i.e., there is a subgroup H_Z of the group G of isometries of our tiling such that the set of tiles with the same code are exactly its orbits. The code for each cell can be computed algorithmically.

A single periodic encoding can be used to create many different periodic patterns. Even though the *Zebra* (Figure 4b) and the *Windy Plains* (Figure 1b) look completely different at the first glance, they are both invariant under H_Z (i.e. isometries which map the encoding into itself also map these patterns into itself) and thus both can be obtained simply by assigning colors to codes from the encoding above. The *random pattern mode* can be used to try random invariant patterns. The *Palace* is based on another regular encoding derived from tiling a hyperbolic plane with circles of radius 2 (50 · 4 codes), and several other lands are based on the *Emerald pattern* consisting of 40 · 4 codes.

The most basic “unit” that can tile a plane would be a triangle connecting centers of three cells at a single vertex; such a triangle is formed by two 1/6s of a hexagon and one 1/7 of a heptagon, so the ratio of hexagons to heptagons is 7:3. Any pattern that tiles the whole plane must preserve the same ratio, and therefore the number of cells in the pattern must be a multiple of 10.

Hyperbolic periodic patterns typically create beautiful tree-like structures. Effects of using a similar process in Euclidean geometry are much more trivial – we can only get a kind of checkered or striped pattern.

Fractal landscapes. The diamond-square algorithm [7] can be used to generate realistic looking Euclidean landscapes; they could be viewed as a two-dimensional generalization of random walk, where we smoothly assign a random value to each point in the grid $\{0, n\}^2$. Ideally, the fractal landscape should be a random function $f : \Omega \times T \rightarrow \mathbb{R}$, where Ω is the probabilistic space and T is the set of all tiles, such that for each isometry i of the hyperbolic plane, f_i given by $f_i(\omega, t) = f(\omega, i(t)) - f(\omega, i^{-1}(t_0))$ has the same distribution as f . As far as we know, the diamond-square algorithm cannot be easily extended to the whole plane \mathbb{Z}^2 ; however, we have found an efficient algorithm working on a hyperbolic tiling [19]. Fractal landscapes are used to generate chasms in the *Dragon chasms*, and environmental factors in the *Galápagos* (Figure 1c) are also generated based on a slightly modified version of this algorithm.

Combinations. All these three methods can be combined – in *Reptiles* (see Figure 1d) a fractal landscape is used to generate the chasms, Zebra pattern is used as a base for the tessellation, and it is randomly decided which of the reptile-shaped floor tiles are actually sleeping reptiles.

Models of Hyperbolic Geometry

The game displays its world in the Poincaré model by default. However, it is also possible to experiment with many other models.

The Minkowski hyperboloid is the set of points (x, y, z) such that $z = \sqrt{1 + x^2 + y^2}$; equipped with the Minkowski metric, it is the model of hyperbolic geometry used internally by HyperRogue [19]. Poincaré, Beltrami-Klein and Gans models can all be obtained from the Minkowski hyperboloid model by viewing the Minkowski hyperboloid from $(0, 0, -p)$, where $p = 0$ for Klein-Beltrami, $p = 1$ for Poincaré, and $p \rightarrow \infty$

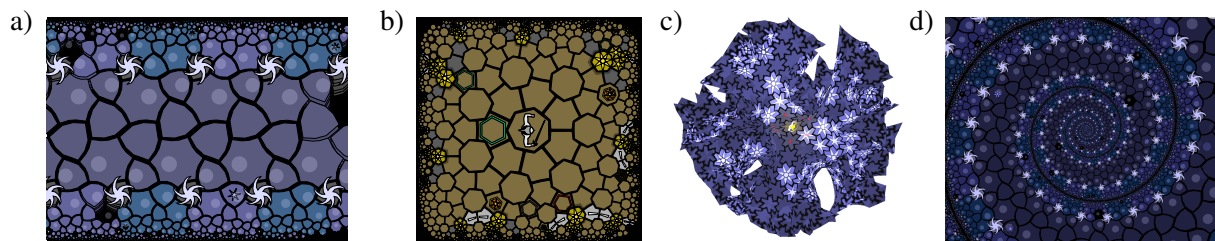


Figure 5: (a) *Band model*, (b) *conformal square* [6], (c) *the Hypersian Rug model*, (d) *conformal spiral*.

for Gans. If we assume that the HyperRogue world is actually a three-dimensional hyperbolic space, the ground level is an equidistant surface c units under some plane, and we are viewing it from a point g units above the plane, this corresponds to $p = \frac{\tanh g}{\tanh c}$. Thus, HyperRogue’s standard 3D display, using the Poincaré model for the ground level and intermediate values of p for tops of walls and other levels, accurately shows how the situation would look from the bird’s eye, assuming $g = c = 1$.

One counterintuitive property of hyperbolic geometry is that, even though it may seem that you change your movement direction a lot during the game, your path is actually very close to a straight line, except the parts where you were backtracking to a previous location (as happens in the *Ivory Tower* or during the *Yendor Quest*, for example). The band model [2] is a lesser-known conformal model, corresponding to the Mercator projection of a sphere. In this model, a fixed straight line L is mapped isometrically to a Euclidean straight line. By picking the line L to be the line passing through the starting and ending locations in the given game, the player can view their whole path as a single straight band. Another effect can be achieved with the Poincaré half-plane model – if L is mapped to a vertical line in the half-plane model, the whole game can be seen by zooming the model in or out.

The *Hypersian Rug* model can be used to play the game on an approximate isometric embedding of hyperbolic plane in \mathbb{R}^3 , similar to what you get from hyperbolic crocheting [9]. This gives a good insight into the curved nature of HyperRogue’s world, and how the points close to the edge of the Poincaré disk are actually much further away from each other than it appears. Such embedding is shown on Figure 5c, although a static image does not do it justice – it is much better to rotate the model in HyperRogue, and even better to touch a real crocheting or paper model.

Applications, or More than a Game

We believe that HyperRogue is one of the best ways to teach or learn about hyperbolic geometry. Even though it does not teach you any mathematical formulas, a very good intuitive grasp of hyperbolic geometry will be necessary to achieve success in the game, as we have argued in the previous sections. Other than the game itself, the built-in map and vector graphics editor could be used by mathematicians and artists, e.g., to find new periodic patterns or new tessellations; this is however currently limited only to tessellations and patterns derived from the order-3 heptagonal tiling in some way.

It is worth to mention that HyperRogue has potential applications outside of education and art. Hyperbolic geometry has gained interest in the data visualization and network science communities. Exponential amount of space allows to easily fit very big trees [12, 14, 11]. Hyperbolic geometry is also intrinsic in scale-free networks [15, 1]. The publicly available source code of HyperRogue contains a very rich set of powerful tools for representing, visualizing, navigating, and performing computations in the hyperbolic plane, both continuous and based on the order-3 heptagonal tiling. These tools are very useful not only for games, but also could be used as a basis of hyperbolic graph visualization software [3] or efficient implementations of algorithms based on hyperbolic geometry.

Acknowledgments

HyperRogue could not achieve its current state if not for the help of its players, constantly reporting bugs and coming up with great ideas for new lands or other features. We would like to especially thank Shawn Parrotte for making the music, tricosahedron and wonderfullizardofoz on Steam for many great discussions and ideas, and Tristan Parker (Sprite Guard) for making a long series of videos about HyperRogue. The full list of significant contributors can be found in the in-game credits and on the HyperRogue website [18]. This work is partially supported by the National Science Centre, Poland, grant DEC-2016/21/N/HS4/02100.

References

- [1] Thomas Bläsius, Tobias Friedrich, Anton Krohmer, and Sören Laue. Efficient embedding of scale-free graphs in the hyperbolic plane. In *European Symposium on Algorithms (ESA)*, pages 16:1–16:18, 2016.
- [2] Vladimir Bulatov. Conformal models of the hyperbolic geometry, 2010. Available online at <http://bulatov.org/math/1001/index.html> (as of Jan 20, 2017).
- [3] Dorota Celińska and Eryk Kopczyński. Programming languages in GitHub: a visualization in hyperbolic plane. In *Proceedings of ICWSM 2017, Montreal, Canada, May 16-18, 2017*. To appear.
- [4] H. S. M. Coxeter. The non-Euclidean symmetry of Escher’s picture Circle Limit III. *Leonardo*, 12:19–25, 1979.
- [5] David L. Craddock. *Dungeon Hacks: How NetHack, Angband, and Other Roguelikes Changed the Course of Video Games*. Press Start Press, 1st edition, 2015.
- [6] Chamberlain Fong. The conformal hyperbolic square and its ilk. In *Proceedings of Bridges 2016: Mathematics, Music, Art, Architecture, Education, Culture*, pages 179–186. Tessellations Publishing.
- [7] Alain Fournier, Don Fussell, and Loren Carpenter. Computer rendering of stochastic models. *Commun. ACM*, 25(6):371–384, June 1982.
- [8] Vi Hart, Andrea Hawksley, Elisabetta A. Matsumoto, and Henry Segerman. Non-euclidean virtual reality I: explorations of \mathbb{H}^3 . In *Proceedings of Bridges 2017: Mathematics, Music, Art, Architecture, Culture*. Tessellations Publishing, 2017.
- [9] D. W. Henderson and Daina Tamina. Crocheting the hyperbolic plane. *Mathematical Intelligencer*, 23(2):17–28, 2001.
- [10] Douglas R. Hofstadter. *Godel, Escher, Bach: An Eternal Golden Braid*. Basic Books, Inc., 1979.
- [11] T. Hughes, Y. Hyun, and D. Liberles. Visualising very large phylogenetic trees in three dimensional hyperbolic space. *BMC Bioinformatics*, 5:48, Apr 2004. <http://www.caida.org/tools/visualization/walrus/gallery1/>.
- [12] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’95, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [13] David Madore, 2013. <http://www.madore.org/~david/math/hyperbolic-maze.html> (Jan 27, 2017).
- [14] Tamara Munzner. Exploring large graphs in 3d hyperbolic space. *IEEE Computer Graphics and Applications*, 18(4):18–23, 1998.
- [15] Fragkiskos Papadopoulos, Maksim Kitsak, M. Angeles Serrano, Marian Boguñá, and Dmitri Krioukov. Popularity versus Similarity in Growing Networks. *Nature*, 489:537–540, Sep 2012.
- [16] Radmila Sazdanovic. Fisheye view of tessellations. In *Proceedings of Bridges 2012: Mathematics, Music, Art, Architecture, Culture*, pages 361–364, Phoenix, Arizona, USA, 2012. Tessellations Publishing.
- [17] Jeff Weeks, 2009. <http://www.geometrygames.org/HyperbolicGames/> (as of Jan 27, 2017).
- [18] HyperRogue website. <http://www.roguetemple.com/z/hyper/> (as of Jan 27, 2017).
- [19] HyperRogue: programming. <http://www.roguetemple.com/z/hyper/dev.php> (as of Jan 27, 2017).