Visualizing Geometric Surfaces

Clarissa Benitez, Daniel Heilman, Ananth Jayadev

Mason Experimental Geometry Lab

George Mason University

April 27, 2021

What is a Manifold?

We think of a *manifold* as a topological space which locally resembles the Euclidean plane near each point. On these spaces, you can also talk about geometrical structures. As an example, the sphere is a two dimensional manifold with positive curvature, but locally it seems flat. When we as humans walk on Earth, we don't feel that we are walking on a curved surface, even though we are.

Purpose of Project

The point of this project was to learn about mathematical structures such as the cylinder and the sphere. But far too often we tend to look at these structures in an abstract way and analyze them via formulas or theorems. While this sort of analysis has its place, it is sometimes useful to visualize these structures using software like the C# programming language and the Unity Game Engine. This allows us to better understand these mathematical structures.

Translation Surface

A *translation surface* is a surface obtained from identifying the sides of a polygon in the Euclidean plane by translations.

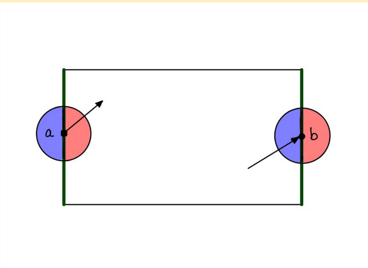


Figure: Here is an example of how the point a and b are equated and how the area around the point translates the arrow

Hyperbolic Space

Hyperbolic space is a kind of space which has constant negative curvature. We can contrast this with the sphere, which has positive curvature. A common image used to depict hyperbolic space is MC Escher's painting of Angels and Devils. In this painting we see a circular surface covered with Angels and Devils interlocking with each other. The figures are larger towards the center of the surface and smaller near the boundaries of the surface, though in hyperbolic space they all have the same size.

Figure: This is the rather famous artwork of MC Escher titles 'Angels and Devils.' This is an example of hyperbolic space.

Work done in brief

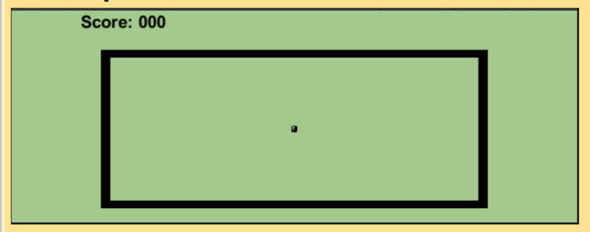
Snakes in the Plane

We inherited a large code base with many different C files and Unity Scenes. The issue was that nothing really worked properly. So the first thing we did was simply try to get everything working again. After that we made some superficial changes to the games by making a score counter, fixing the sizes of the snake heads which move in the game, and adding GIF's which depict the topological structure the snake is moving on in the game. More recently, we actually made whole new scenes on different geometric surfaces like the cylinder and the dodecagon.

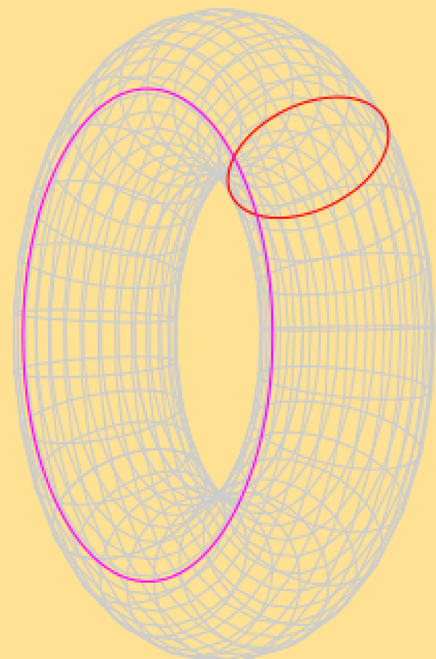
Graphics

Snakes in the Plane

- This is a screenshot of the original mode of the Snake game. This works exactly like the normal snake game; the snake can't run into the boundaries and can't run into itself. This simulates the Euclidean plane.

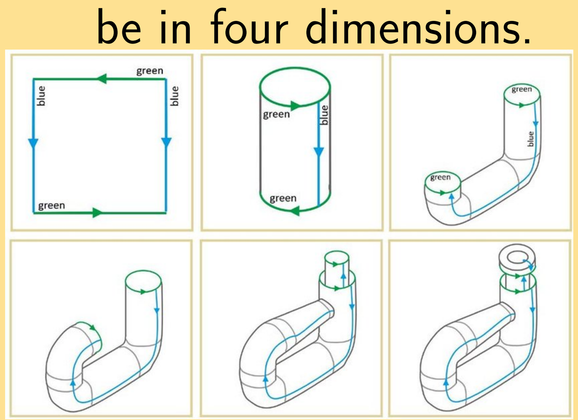


- We can simulate a torus by taking the image above and make a rule such that when the snake crosses one side it lands on the opposite side.



Surfaces out of Paper

- In the image below we see that a flat Euclidean plane, which appears to us as a square, can be transformed into a Klein Bottle. We start by attaching the two sides on the left and right. This makes a cylinder.
- Then, because the other two arrows face the opposite direction, we have to "twist" the structure to make the same sides meet. This cannot actually be done in three dimensional space; we need an extra dimension to do this. As a result, the Klein Bottle appears to be a self-intersecting surface when in fact it would not be in four dimensions.



Hyperbolic Geometry Intro.

To preface, *Euclidean geometry* is your frame of reference. It is what we are taught in school and is generally how we think of movement. Different geometries can each be described by a set of *axioms*, or statements assumed to be true as a premise for other arguments. Five axioms are used to define *Euclidean space*. *Hyperbolic geometry* shares all but but one of these.

Definition (Hyperbolic Parallel Postulate)

For any given line R and point P not on R, in the plane containing both line R and point P there are at least two distinct lines through P that do not intersect R.

Hyperbolic Soccer

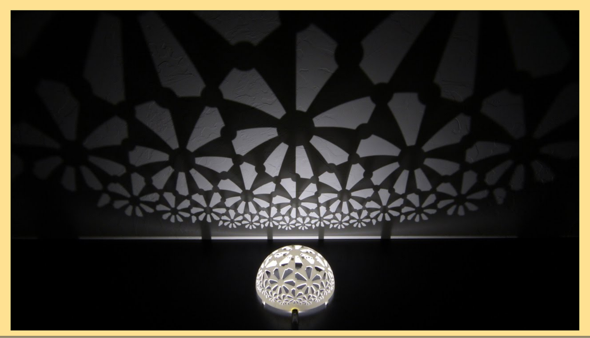
Hyperbolic Soccer, created in Unity, is a game with the goal of having a ball reach a goal after sending it in some initial direction on a Hyperbolic plane. The movement code uses the Poincaré half-plane model of hyperbolic geometry. Geodesics are vertical lines and semi-circles, which did the real axis (the lower boundary) at right angles. Lengths are distorted, but angles are correct.

We inherited an early version of a previous MEGL project. We began by deciphering the existing code and understanding ourselves what was necessary to work to improve the visualization aspect. Bugs with the camera and the pointer (shows the direction the player is aiming) were resolved. A ball trail was added. From there, our goal was to, firstly, make the math concept behind the movement more obvious to a player. The soccer field boundaries, which began as two straight lines and a goal, became a projection of hyperbolic geodesics. Aesthetic changes were our next priority. Tweaks such as camera angle, etc. were fixed, but the main idea was to make the field resemble a soccer field.

Visual References

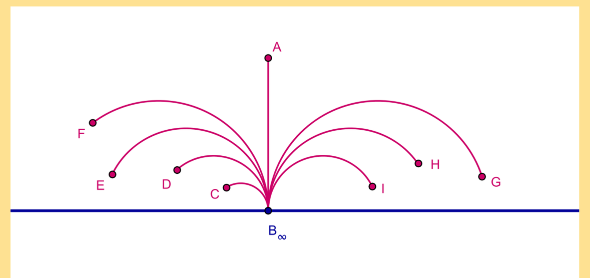
Poincaré Half-Plane Model

- Below is a visualization of the half-plane model; a stereographic projection of a half-sphere.



Hyperbolic Parallels

- Below are parallel rays in the Poincaré half-plane model of hyperbolic geometry.



Conclusion and Future Work

- Throughout the course of the project we have learned quite a bit about both the math which inspired the creation of the project and some aspects of programming as well. Before we started doing this project we had a vague understanding of topology and certainly didn't know too much about hyperbolic geometry. Doing this project has allowed us to improve our understanding of these areas.
- While we have made quite a bit of improvements to our project, there are still things which can be improved. Most of these improvements have to do with the programming side of the project though. In particular, the graphics of the Snake game can be improved further.

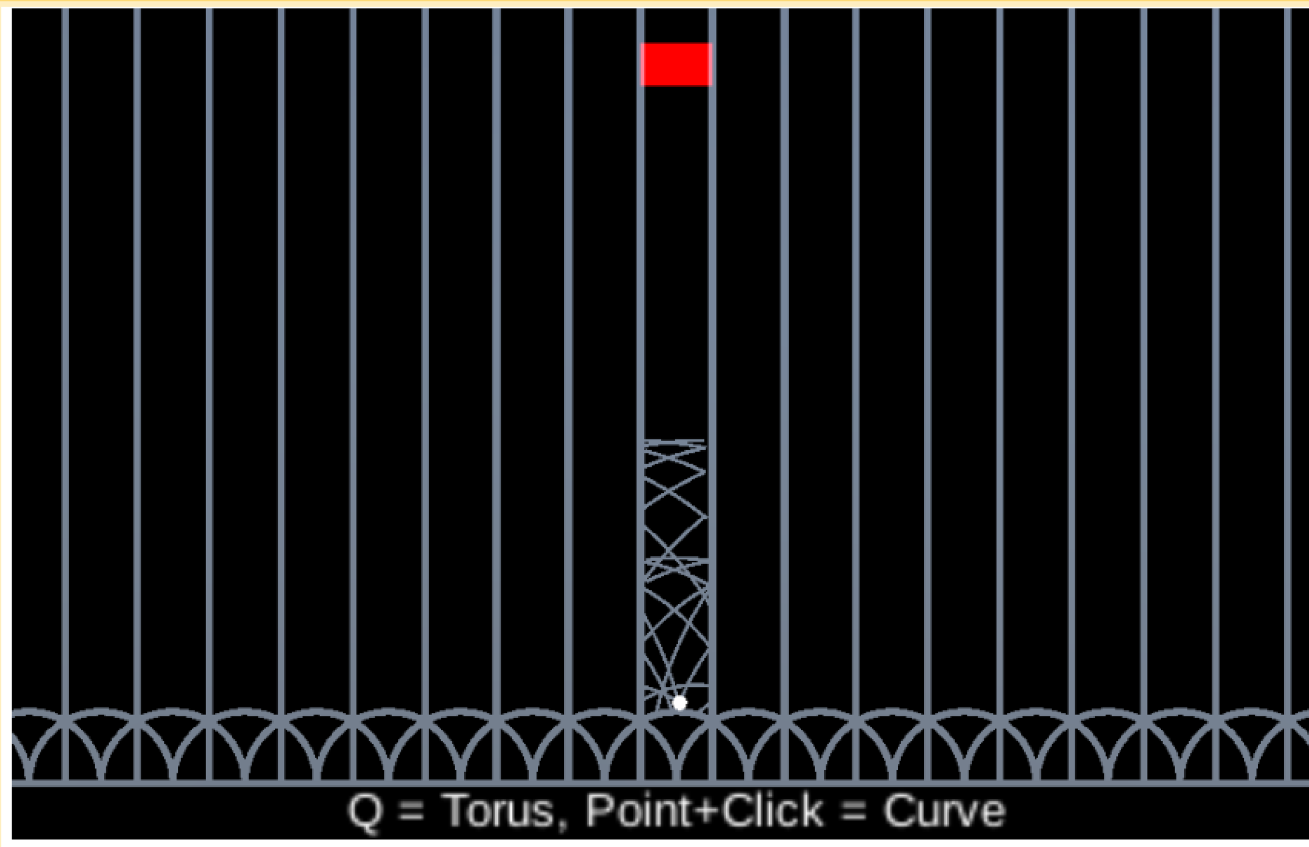
Acknowledgments

All of us would like to thank Dr. Anton Lukyanenko, Quincy Frias, and Don Brusaferro for their advice and support with respect to the project.

References

- https://docs.unity3d.com/Manual/index.html
- https://docs.microsoft.com/en-us/dotnet/csharp/
- https://math.stackexchange.com/
- https://mathoverflow.net/

Hyperbolic Soccer Game



This is an image of the Hyperbolic Soccer game, where the ball is trying to get to the goal. When you click on the game, it ends up hitting the boundaries repeatedly.