Creating Repeating Hyperbolic Patterns Based on Regular Tessellations

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Christopher D. Becker

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Dr. Douglas Dunham

July, 2012

Acknowledgements

I would sincerely like to thank Dr. Douglas Dunham for the support and guidance he provided me during my times of uncertainty. I would also like to thank Dr. Peter Willemsen and Dr. John Greene for agreeing to serve as members of my defense committee.

Finally, I would like to thank my parents for their constant support, patience, and encouragement.

Dedication

Dedicated to
my mom,
Mrs. Tracie Becker,
and my dad,
The Rev. David Becker

Abstract

Repeating patterns have been used in art throughout history. In the middle of the 20th century, the noted Dutch artist M.C. Escher was the first to create repeating hyperbolic patterns that were artistic in nature. These patterns were very tedious to design and draw. Escher did all this work by hand, without the benefit of a computer. This paper discusses how, through the use of a computer program, the creation of repeating hyperbolic patterns is accomplished in a less tedious, more timely manner.

The computer program enables a user to load or create a data file that defines the sub-pattern and other information about the design. The program will take that information and generate the repeating pattern for the user. The user is also able to modify the pattern. The computer program allows the user to precisely and quickly create repeating hyperbolic patterns which will be displayed on the screen. The repeating hyperbolic pattern is also saved as a PostScript file.

Contents

List of Figures

Chapter 1

Introduction

Since ancient times, humans have been generating repeating patterns. In the Mesopotamian Valley, Sumerians decorated their buildings with mosaics in geometric patterns made of hardened clay. Tessellations have been used by many cultures over the years, including Chinese, Roman, Japanese, Greek, Egyptian, and Arabic cultures [1]. These patterns were Euclidean in nature.

Euclidean geometry is a mathematical system that is attributed to the Greek mathematician Euclid of Alexandria. Euclid's *Elements* was the definitive discussion of geometry. It has been one of the most influential books in history and was the standard textbook on geometry for over two thousand years [2]. Euclidean geometry is usually the most convenient way to describe the physical world around us as it deals with two dimensional and three dimensional objects.

The Dutch artist M.C. Escher became fascinated by the Euclidean tilings found at the Alhambra Palace at Grenada, Spain in 1922. After seeing a repeating hyperbolic pattern in an article by H. S. M. Coxeter, he was inspired to create hyperbolic art [3]. His *Circle Limit* patterns are the first, and Escher's only, examples of combining hyperbolic geometry with art [4].

The following chapters will explore a program that was written to allow users to quickly and accurately create repeating hyperbolic patterns based on regular tessellations. In the past, designing and drawing repeating hyperbolic patterns was very tedious and time consuming. When M.C. Escher created his *Circle Limit* patterns, all of the work was done by hand, without the benefit of a computer.

The program is based on an older program. Both of which are based on an algorithm by Dr. Dunham [5]. The new program was designed from scratch to address many shortcomings of the older program, while maintaining backwards compatibility of the data files (discussed in Appendix A). More information on the programs can be found in Chapters 5 and 6. When valid files were tested, the resulting images were as expected. This is covered in more detail in Chapter 6.

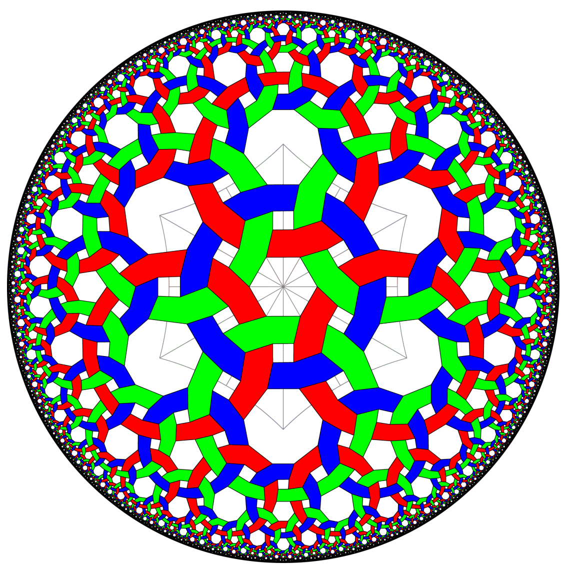

Figure 1.1: An example program output.

Chapter 2

Geometry

The word "geometry" is originally from the Greek word *geometrein* which means "to measure earth." Ancient geometry was basically a set of rules and procedures that could be used to help with everyday applications. Although the Egyptians are attributed as being the first people to know about geometry by the Greek historian Herodotus, many civilizations knew about geometric principles [6].

Today, *geometry* is a branch of mathematics that studies shapes in spaces of various dimensions and types. The most common types of geometries are: planar (which includes shapes such as lines, circles, and squares), solid (which includes shapes such as lines, spheres, and cubes), and spherical geometry (which includes shapes such as spherical lines and spherical polygons).

2.1 Euclidean Geometry

Euclidean geometry is an axiomatic system. This means that all theorems are derived from a finite number of axioms.

Euclid gave five axioms for the basis of his system.

1. Any two points can be joined by a straight line segment.

2. Any straight line segment can be extended indefinitely in a straight line.

3. Given any straight line segment, a circle can be drawn having the line segment as the radius and one endpoint as the center.

4. All right angles are congruent.

5. If two lines intersect a third line in such a way that the sum of the inner angles on one side is less than that of the two right angles, then the two lines inevitably must intersect each other on that side if extended far enough. This axiom is also known as the parallel postulate.

For over two thousand years, the adjective "Euclidean" was unnecessary because no other sort of geometry had been conceived. Mathematicians thought of the parallel postulate as a special postulate, differing from the first four postulates. They did not doubt that it was true, but they thought that it was a theorem rather than an axiom. If it really were a theorem, then it could be proven and not just assumed.

By the last half of the 18th century, this problem of proving the parallel postulate had been attempted by many mathematicians. The difficulty of the problem made some mathematicians believe that proving the parallel postulate was impossible. This did not in any way mean that it could not be proved, but this idea led to the discovery of non-Euclidean geometries. Since *neutral geometry*, consisting of Euclid's first four axioms, did not in itself imply the parallel postulate, mathematicians thought that there must be a different geometry that was based on the first four axioms and the negation of the parallel postulate [7].

2.2 Non-Euclidean Geometry

When many people are working on the same problem with little or no communication between them, multiple independent discoveries are made at nearly the same time. This has been evident numerous times throughout the history of mathematics and science. In this case, it seems that non-Euclidean geometry was discovered at least four times within a twenty-year time span.

Carl Friedrich Gauss seemed to be the first to discover non-Euclidean geometry. He coined the term "non-Euclidean" for the new type of geometry. He worked on this new geometry for many years and discovered many theorems. During that time period, he received a letter from Ferdinand Schweikart that indicated that Schweikart had himself discovered non-Euclidean geometry and had come to similar conclusions and results as Gauss.

Neither Gauss nor Schweikart published any papers related to their discoveries involving non-Euclidean geometry. So, when János Bolyai published his work on non-Euclidean geomtry in the Appendix of his father, Wolfgang's, book, it established the field of non-Euclidean geometry.

It was discovered, however, that Bolyai was not the first person to have a paper published on the topic of non-Euclidean geometry. A Russian mathematics professor, Nikolai Lobachevsky, had already published a paper on the topic. However, since this paper was published in Russian, it was not well known in the European mathematical circles. Once it was translated into French and German, Lobachevsky's work and discoveries in the area of non-Euclidean geometry were reaffirmed [7].

While non-Euclidean geometry is technically the study of any geometry that is not Euclidean, one of the most useful non-Euclidean geometries is hyperbolic geometry. Hyperbolic geometry is the geometry discovered by Bolyai, Gauss, Lobachevsky, and Schweikart [6] and is the geometry of hyperbolic space.

2.2.1 Hyperbolic Geometry

In hyperbolic geometry, all of Euclid's axioms hold with the exception of the parallel postulate. Instead of the parallel postulate, an axiom called the *Hyperbolic axiom* is used. This axiom states that if there exists a line l and a point P not on the line l, there are at least two distinct lines parallel to l passing through P (as shown in Figure 2.1) .

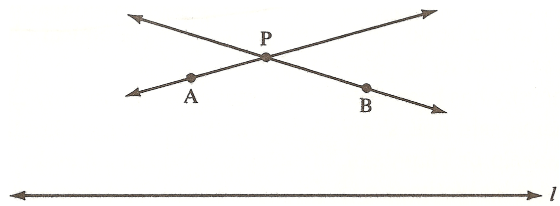

Figure 2.1: Hyperbolic Axiom

This axiom is used to prove and obtain many other userful properties of hyperbolic geometry. Some of the properties of hyperbolic geometry are:

1. The sum of the angles of a triangle is less than 180°.

2. Rectangles do not exist.

3. All convex quadrilaterals have angle sum less than 360°.

4. If two triangles are similar, they are congruent, that is, if the angles of the triangles are equal, so are their sides [6].

A *model* is an interpretation of the primitive terms under which the axioms become true statements. Here, interpretation does not mean "understanding of the meaning," but rather to the more basic sense of "giving a meaning" [7]. Since primitive terms in a mathematical system have no inherent meanings, a model is used to assign meanings that make all the axioms true.

Some models of hyperbolic geometry can represent hyperbolic objects in a finite portion of Euclidean 2-space. Two examples of finite models for hyperbolic geometry are the Beltrami-Klein model and the Poincaré circle model. An example of an infinite model for hyperbolic geometry that is embedded in Euclidean space is the Weierstrass model. The finite models have boundaries which play an important role in the definition of parallel lines in hyperbolic space.

In the next chapter, we discuss the Poincaré Disk, Klein, and Weierstrass models.

Chapter 3

Hyperbolic Geometry Models

3.1 The Poincaré Disk Model

This model was created by the French mathematician, physicist, and philosopher Henri Poincaré. It is considered by some to be the easiest of the models for hyperbolic geometry to comprehend. The Poincaré Disk Model is *conformal.* This means that angles are represented accurately by the model. In this case, however, distances are distorted.

A "point" in this model is any point within a Euclidean unit circle centered at the origin. That is, a point is defined as the 2-tuple (x, y), such that $x^2 + y^2 < 1$. Note that all points on the diameter of the circle are not a part of this model (i.e. $x^2+y^2 \neq 1$).

A "line" in this model is one of two things. A line is either a diameter of the circle, or it is an arc of a circle orthogonal to the boundary of the circle.

The terms "lies on" and "between" maintain their Euclidean meanings.

Since the Poincaré Disk Model is conformal, measuring the angle between the tangents to two intersecting "lines" at the point of intersection will provide the measurement of the angle between two lines [6].

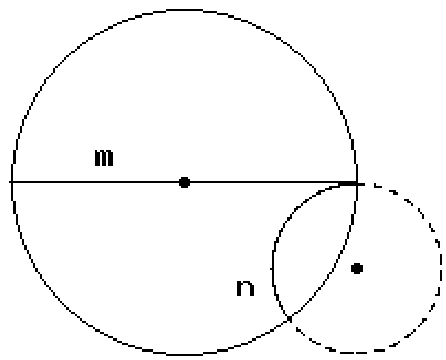

Figure 3.1: Lines m and n on the Poincaré Disk Model

3.2 The Klein Model

This model is also known as the "Beltrami-Klein Model." It is similar to to Poincaré Disk Model in many ways.

A "point" is any point within a Euclidean unit circle centered at the origin. That is, a point is defined as the 2-tuple (x, y), such that $x^2 + y^2 < 1$. Note that all points on the diameter of the circle are not a part of this model (i.e. $x^2 + y^2 \neq 1$).

A "line" in this model is an open chord on the unit circle. An open chord is a chord of the circle with the end points removed.

The terms "lies on" and "between" maintain their Euclidean meanings.

Note that this model is not conformal. That is, the angles are not represented accurately by the model [6].

3.3 Comparison of the Klein and Poincaré Disk Models

As stated previously, the Poincaré Disk Model and the Klein Model have many similarities. They both model hyperbolic space on the unit circle.

3.3.1 Conversion Between the Models

Given vector p that represents a point of the Poincaré Disk Model, the corresponding point of the Klein Model (say vector k) is calculated as:

$$k = \frac{2p}{1 + p \cdot p}. \tag{3.1}$$

Conversely, given vector k that represents a point of the Klein Model, the corresponding point of the Poincaré Disk Model (say vector p again) is calculated as:

$$p = \frac{k}{1 + \sqrt{1 - k \cdot k}} = \frac{(1 - \sqrt{1 - k \cdot k})k}{k \cdot k}. \tag{3.2}$$

Given two ideal points (i.e. points on the boundary of the unit disk), the line for the Klein model is the open chord between them, while the line for the Poincaré Disk model is a circular arc on the two-dimensional subspace generated by the ideal point vectors, orthogonal to the boundary of the circle.

The relationship between the two models is simply a projection from the center of the disk. This projection is a ray from the center of the circle passing through a point of one model passing through the corresponding point of the other model (as seen in Figure 3.2 [1]).

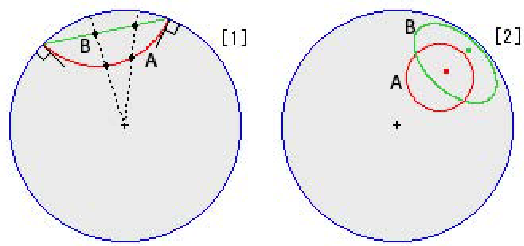

Figure 3.2: Relationship between the Klein Model and the Poincaré Disk Model. (The Poincaré and Klein disks are superimposed on each other.)

Figure 3.2 [1] shows the relationship of a hyperbolic line. The red curve (labeled A) is the circular arc that is perpendicular to the disk edge (circumference at infinity) that is the Poincaré Disk Model representation of the hyperbolic line. The green line (labeled B) is the line that is the Klein Model representation of the hyperbolic line.

Figure 3.2 [2] shows the relationship of a hyperbolic circle with hyperbolic radius of 1 unit. Its center is located at a hyperbolic distance of 1.5 units from the origin. The red circle (labeled A) is the Poincaré Disk Model representation of the hyperbolic circle. The green ellipse (labeled B) is the Klein Model representation of the hyperbolic circle. As shown by Figure 3.2 [2], it is evident that both centers are not at the visual centers of the circle or the ellipse [8].

3.4 The Weierstrass Model

Unlike the Poincaré Disk and Klein models, this model uses a surface in Euclidean 3-space rather than a disk. This model is less well known than the Poincaré Disk Model. However, it is attractive for several reasons. It shares many properties with the sphere as a model for elliptic geometry. For example, in this model, "lines" are intersections of the surface with planes through the origin [2].

Before we actually define the terms "point," "line," "lies on," and "between," some preliminary information is needed. We will be following the approach taken by Faber (see [2]) very closely.

A "point" in real 3-dimensional space ($\mathbb{R}^3$) is respresented by a triplet of real numbers (e.g., $P = (x, y, z)$). This triplet can also be represented as a 3-dimensional vector (e.g., $P = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix}$). Visually, the vector can be thought of as an arrow drawn from the origin ($O = (0, 0, 0)$) to P. From now on the terms "vector" and "point" will be used interchangeably.

Given two vectors, $X = (x_0, x_1, x_2)$ and $Y = (y_0, y_1, y_2)$, their Euclidean *dot product* or inner product, denoted as $X \cdot Y$ is:

$$X \cdot Y = x_0 * y_0 + x_1 * y_1 + x_2 * y_2.$$

On the other hand, the *hyperbolic inner product* of the vectors, denoted as $< X, Y >$ is:

$$< X, Y >= x_0 * y_0 + x_1 * y_1 - x_2 * y_2.$$

Two vectors are called e-orthogonal if $X \cdot Y = 0$ and h-orthogonal if $< X, Y >= 0$.

The equation of a plane in $\mathbb{R}^3$ is a linear equation of the form $A \cdot X = b$ where A is called the coefficient vector and b is a constant. A is normal to the plane. Similarly, if $< A, X >= b$, then A is considered a hyperbolic normal (*h-normal*) of the plane. Also note that any scalar multiple of an h-normal is also an h-normal to the same plane.

We are now ready to define the terms. A "point" in this model is defined to be a point X of $\mathbb{R}^3$ such that $< X, X >= -k^2$ and $x_2 > 0$ (i.e. $x_0^2 + x_1^2 - x_2^2 = -k^2$)where k is a constant. These points are the "upper sheet" of a 2-sheeted hyperboloid. We will call this part of the surface H^2.

As stated previously, a "line" in this model is defined to be the intersection of H^2 with a plane through the origin of $\mathbb{R}^3$. From Euclidean solid geometry, we know that such an intersection is one branch of a hyperbola. This is shown in Figure 3.3.

This plane through the origin has an equation of the following form: $< X, \ell >= 0$ where ℓ is an h-normal of the plane (as seen in Figure 3.3). Since ℓ is an h-normal, we can always assume that $< \ell, \ell >= k^2$. This means ℓ is a point of the single-sheeted hyperboloid with the equation $< X, X >= k^2$. This is shown in Figure 3.4.

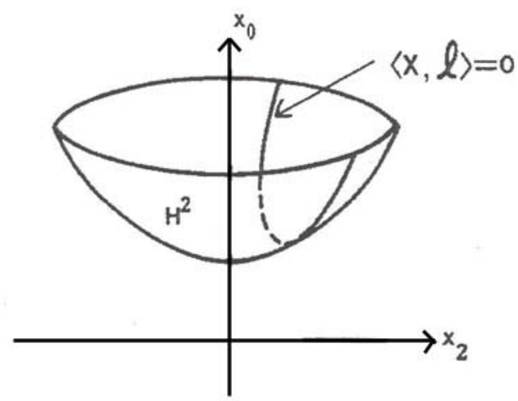

Figure 3.3: A “line” in the Weierstrass Model

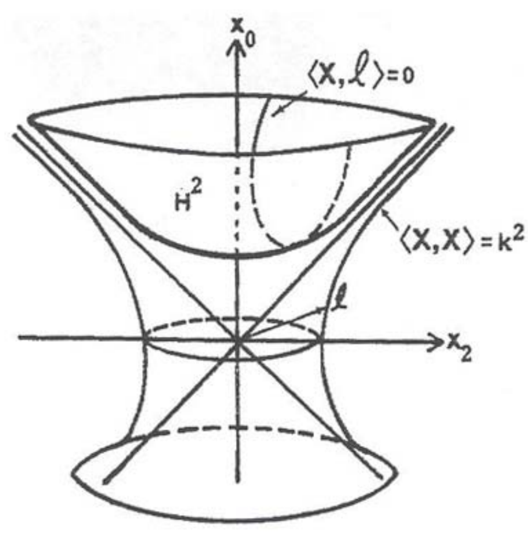

Figure 3.4: $< X, \ell >= 0$ in the Weierstrass Model

In this model, a point X of H^2 “lies on” a line ℓ only if $< X, \ell >= 0$.

Since, as we mentioned before, a line is one branch of a hyperola, given three points A, B, C on a line, C is "between" A and B if ACB or BCA is the order encountered when the branch is traversed in either direction.

3.5 Isomorphism Between the Poincaré Disk and Weierstrass Models

Currently, the Weierstrass Model is used when doing all computations for the transformations and the Poincaré Disk Model is used for displaying the result. Next, various aspects of the isomorphism existing between the Weierstrass and Poincaré Disk Model will be discussed. We will focus on those aspects which are relevant to this thesis.

We consider the upper sheet of the hyperboloid $< X, X >= -k^2$. Without loss of generality, we can assume $k^2 = 1$. Then we have the equation for the hyperboloid as $< X, X >= -1$, which intersects the z-axis at the point $(0, 0, 1)$. If we project a point on the sheet down toward the point $(0, 0, -1)$ stereographically, the projected point lies within the unit circle on the xy-plane with its center at the origin. This is the Poincaré Disk that corresponds to the upper hyperboloid sheet (as seen in Figure 3.5).

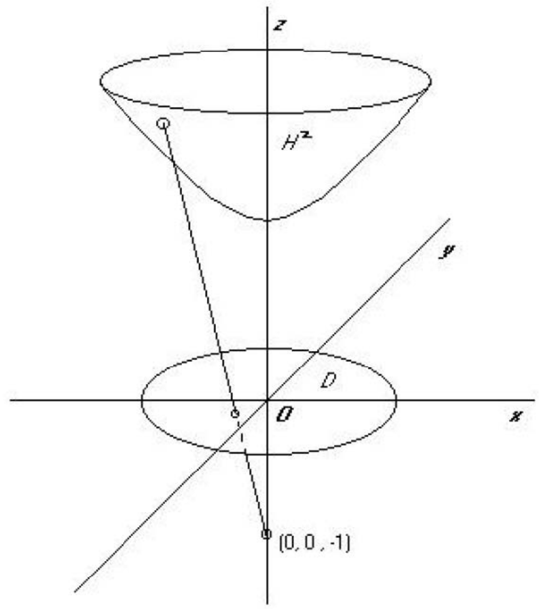

Figure 3.5: Stereographic projection between the Weierstrass model (H^2) and the Poincaré Disk Model (D)

To convert a point from the Weierstrass Model (we'll call it W) to a point for the Poincaré Disk Model, we use the following mapping:

$$W \longrightarrow \frac{1}{1+w_2}\begin{bmatrix} w_0 \\ w_1 \\ 0 \end{bmatrix}, whereW = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}. \tag{3.3}$$

To convert a point from the Poincaré Disk Model (we'll call it P) to a point for the Weierstrass Model, we use the following mapping:

$$P \longrightarrow \frac{1}{1-p_0^2-p_1^2}\begin{bmatrix} 2*p_0 \\ 2*p_1 \\ 1+p_0^2+p_1^2 \end{bmatrix}, whereP = \begin{bmatrix} p_0 \\ p_1 \\ 0 \end{bmatrix}. \tag{3.4}$$

Chapter 4

Patterns

4.1 Tessellations

A *tessellation* is a repeating pattern that covers a (usually 2-dimensional) surface without overlapping or leaving gaps. The pattern is created by replicating and transforming congruent copies of a basic sub-pattern.

Figure 4.1 below is an example of part of a tessellation. The colors were added for illustration. The sub-pattern is an individual polygon.

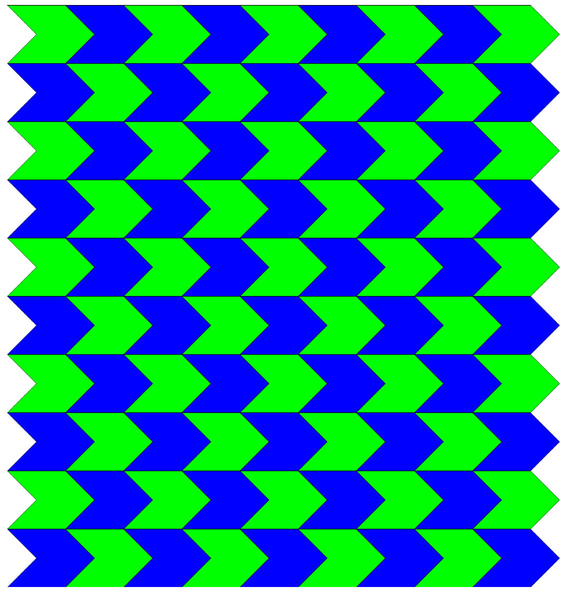

Figure 4.1: A Euclidean tessellation

4.2 Hyperbolic Patterns

As discussed in the previous section, a *repeating pattern* is formed by replicating a sub-pattern. This sub-pattern is called a motif. Copies of the motif do not need to interlock. This thesis will focus on repeating hyperbolic patterns. These patterns are composed of hyperbolically congruent copies of the motif. From this point forward, "repeating pattern" will mean a "repeating hyperbolic pattern."

Regular tessellations {p, q} of the hyperbolic plane are good examples of repeating hyperbolic patterns. The notation {p, q} denotes that the pattern is composed of p-sided polygons which meet q congruent copies at a vertex. While the polygons can be irregular, this paper will only discuss regular polygons. An example of the (regular) tesselation {6, 4} can be seen in Figure 4.2.

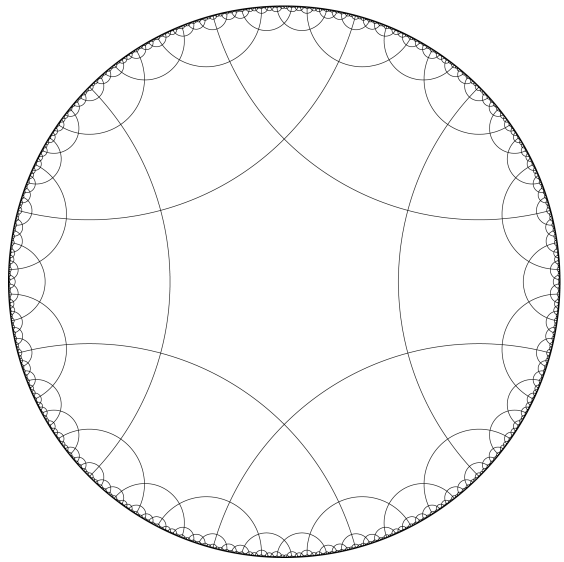

Figure 4.2: The regular tessellation {6, 4}

To obtain a regular tessellation of the hyperbolic plane it is also necessary that $(p-2)(q-2) > 4$. Also, $(p-2)(q-2) = 4$ correlates to Euclidean tessellations (tessellations on a 2-dimensional plane) and $(p-2)(q-2) < 4$ correlates to spherical tessellations. An example of a repeating hyperbolic pattern can be seen in Figure 4.3 below.

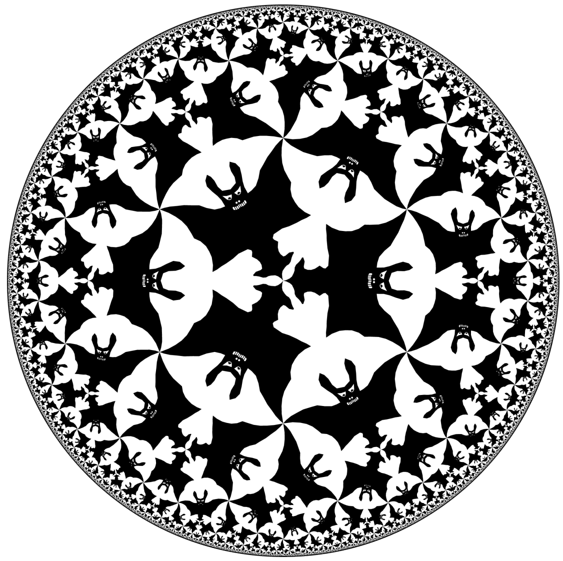

Figure 4.3: A repeating hyperbolic pattern. A computer generated version of M.C. Escher's Circle Limit IV pattern.

4.2.1 Symmetry Groups

A *symmetry operation*, or "symmetry" for short, is an isometry (refer to Appendix B) that transforms a pattern onto itself. The set of all symmetry operations for a figure is called the *symmetry group* for a figure.

The fixed lines of reflection in a repeating pattern are called lines of symmetry or mirrors. The lines of the symmetry of the tessellation {p, q} divide each p-gon into $2p$ right triangles. Each triangle has acute angles of π/p and π/q. Note that the angles do not need to sum to 2π because of hyperbolic geometry. The symmetry group of the regular tessellation {p, q} is denoted by [p, q]. This symmetry group can be generated by reflections across the sides of each triangle. Figure 4.4 is an example of the symmetry group [6, 4] of the regular tessellation {6, 4}. The lines of symmetry for the central p-gon have been drawn in. There are also two symmetry subgroups of [p, q] that will be discussed next.

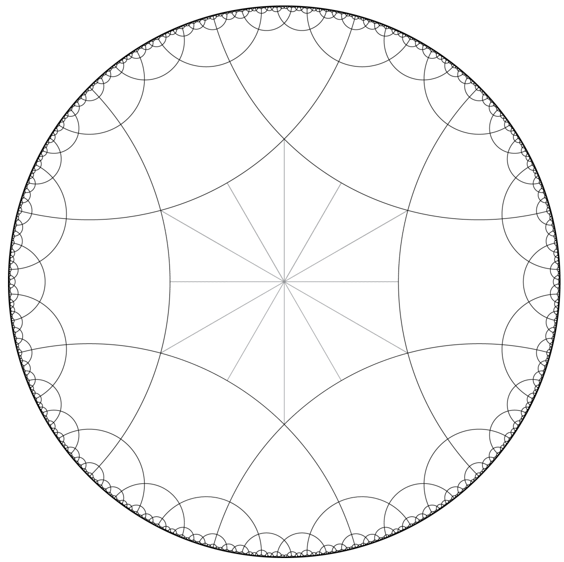

Figure 4.4: A pattern with symmetry group [6, 4]. The lines of symmetry for the central p-gon are shown.

Symmetry Group [p, q]$^+$

The symmetry group [p, q]$^+$ is a symmetry subgroup of [p, q] of index 2 [9]. This means that [p, q] has "twice as many" symmetries in the group as [p, q]$^+$. There are actually two ways to generate this group. The first way is to include all symmetries from [p, q] which are the result of applying an even number of reflections from [p, q]. The second way is to apply any two of the following three rotations: 180, $360/p$, and $360/q$ degrees about the corresponding vertices of the right triangle formed by the mirrors. Figure 4.5 is an example of a pattern generated using the symmetry group [5, 4]$^+$.

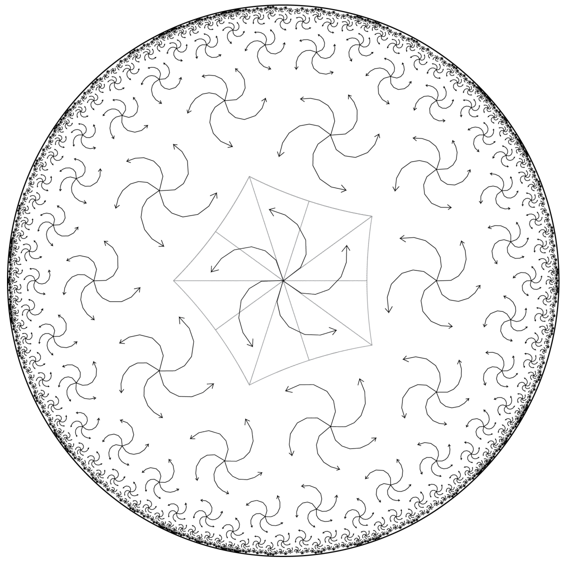

Figure 4.5: A pattern with symmetry group [5, 4]$^+$. The lines of symmetry for the central p-gon are shown.

Symmetry Group [p$^+$, q]

The symmetry group [p$^+$, q] (where q must be even) is also a symmetry subgroup of [p, q] of index 2 [9]. One way to generate this group is to rotate $360/p$ degrees about the center of the p-gon in {p, q} and a reflection in one side of that p-gon. Figure 4.6 is an example of a pattern generated using the symmetry group [5$^+$, 4]. Note the similarities and differences between Figure 4.5 and Figure 4.6. In Figure 4.5, all the arrows have a counterclockwise orientation. However, in contrast to this, Figure 4.6 has arrows with both clockwise and counterclockwise orientations.

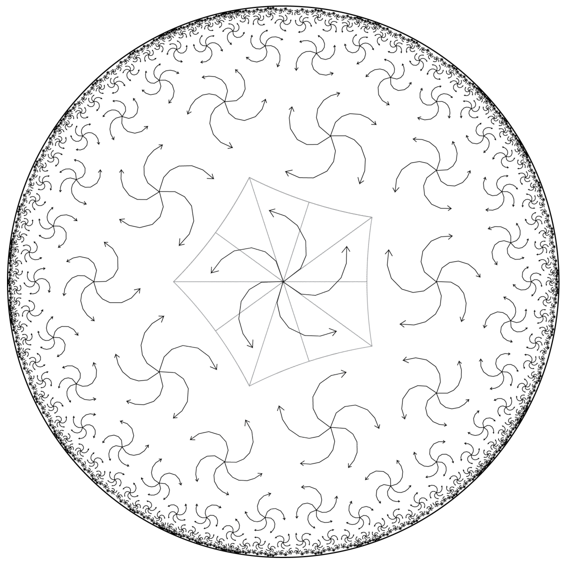

Figure 4.6: A pattern with symmetry group $[5^+, 4]$. The lines of symmetry for the central p-gon are shown.

4.2.2 Motif and Fundamental Region

As defined earlier in this chapter, a *motif* is a basic sub-pattern from which congruent copies are used to generate a repeating pattern. As defined in Dunham [10], if the hyperbolic plane is covered by transformed copies of a connected set under elements of a symmetry group without overlap, that set is called a *fundamental region* for the symmetry group. If the motif covers the entire fundamental region, then the resulting repeating hyperbolic pattern will be interlocking. Figure 4.7 is an example of an interlocking pattern. In contrast, Figure 4.5 and Figure 4.6 are examples of non-interlocking repeating hyperbolic patterns.

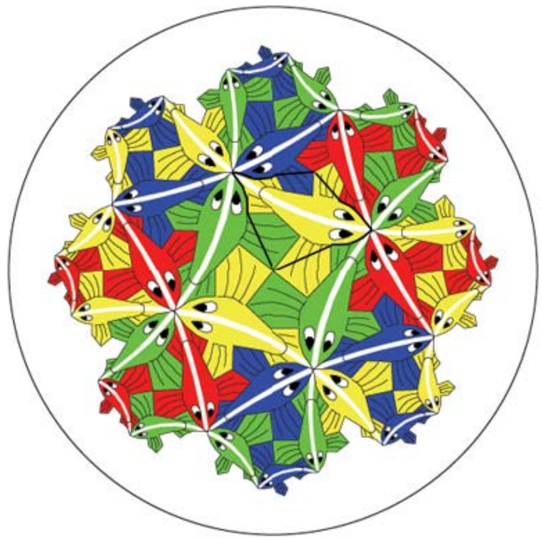

Figure 4.7: An interlocking repeating pattern. The fundamental region is shown with dark boundaries.

4.2.3 Repeating Pattern Generation Algorithm

This section will provide an overview of the repeating hyperbolic pattern generation algorithm that was designed by Dunham [5] as a part of his research on creating repeating hyperbolic patterns. This has been implemented in an older version of the program written in the C programming language and uses the Motif framework for the graphical user interface.

Generating the repeating pattern involves replicating the motif to obtain the complete repeating pattern. The first step of this process is to form what is called the "p-gon pattern." This is done by appropriately rotating the motif around the p-gon center and/or reflecting it across the diameters and perpendicular bisectors of edges until the p-gon is filled with copies of the motif. It is important to note that p-gons of a {p, q} tessellation are arranged in layers. This p-gon pattern is the first layer of the repeating hyperbolic pattern (sometimes called the *central p-gon pattern*). Some of the copies of the motif may protrude from the p-gon as long as there are corresponding indents so the final pattern is interlocking. Only some of the reflections may be used and the rotations may be some, but not all, multiples of $360/p$ degrees. Figure 4.8 is an example of a central p-gon that is not filled by the motif, but when expanded to multiple layers (as shown in Figure 4.7) one can see that it is indeed an interlocking pattern.

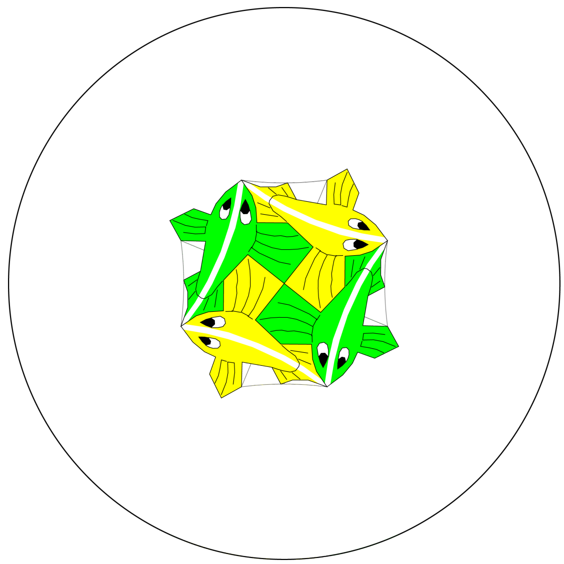

Figure 4.8: The replication of the fundamental region (as was shown in Figure 4.7) to generate the p-gon pattern.

The next step is to replicate this p-gon pattern to complete the final pattern. By replicating the p-gon pattern as a whole, instead of the motif, the algorithm becomes simpler, more efficient, and less susceptible to rounding errors introduced by computer hardware. The layers are generated recursively. As already mentioned, the first layer of the tessellation is the central p-gon. The k+1st layer consists of all the p-gons sharing an edge or vertex with a p-gon in the kth layer. Figure 4.9 shows extending the p-gon pattern from layer 2 to layer 3. The p-gon labeled 1 is first rotated about its vertex A to draw the p-gon labeled 2. Next, it is rotated twice around its vertex B to draw the p-gons labeled 3. Then, it is rotated twice around its vertex C to draw the p-gons labeled 4. This process will be explained in more detail in the next section.

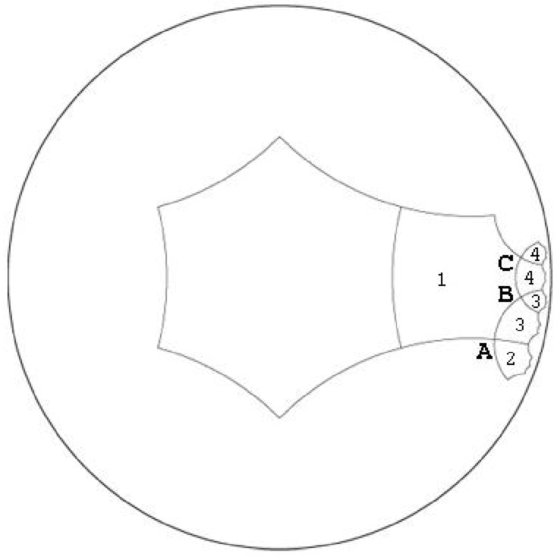

Figure 4.9: Extending the p-gon pattern from layer 2 to layer 3.

The transformation of a p-gon from one layer to the next involves three steps. First, each point of the p-gon pattern is projected to a hyperboloid in the Weierstrass Model (refer to Section 3.4 for more information on this model) using the transformation provided in Equation 3.4 in Section 3.5. Next, each point is transformed to its new location by taking the product of the appropriate Lorenz matrix (refer to Appendix B) for the transformation with the vector representing the coordinates of the point. Lastly, the point is projected back to the Poincaré Disk Model using the transformation provided by Equation 3.3 in Section 3.5. The transformation to and from the Weierstrass Model is done because of the ease of representation of the the transformations using 3x3 Lorenz matrices.

4.2.4 Implementation of the Replication Algorithm

This section will describe in more detail the recursive replication algorithm that is used by the repeating hyperbolic patterns generation algorithm by Dunham [5]. As was discussed in the previous section, to extend the pattern from the k^{th} layer to the $(k+1)^{st}$ layer, the algorithm iterates over each vertex of the p-gon in the k^{th} layer that is exposed to the $(k+1)^{st}$ layer. For each vertex, it calculates the number of polygons that need to be drawn in the $(k+1)^{st}$ layer from that vertex. This number is either $q-2$ or $q-3$, depending on the exposure of the vertex. Then, the algorithm recursively calls itself for the vertices of the p-gons just drawn in the $(k+1)^{st}$ layer. This is shown in Figure 4.9. After the p-gon labeled 1 is drawn, the algorithm is called for each of its vertices (labeled A, B, and C). The process continues by calling the agorithm on the exposed vertices of the p-gons labeled 2 through 4.

Chapter 5

User Interface

5.1 Description

The program provides a graphical user interface that allows the user to create, modify, save a data file, and view a repeating hyperbolic pattern. The program does some input error checking. The motif must be contained in one of the polygons that make up the regular hyperbolic tessellation {p, q} by p-sided polygons with q polygons meeting at a vertex. Note that (p - 2)(q - 2) must be greater than 4 for the tessellation to be hyperbolic.

5.2 Using the Program

When the program starts running, you will be greeted by the main interface as shown in Figure 5.1. From there, the user can either open an existing data file that follows the format discussed in Appendix A or create a new data file using the entries under the "File" menu item. Currently the "Tools" menu has a "Settings" item which at this point in time does nothing, but settings could be added in the future. Whenever the image is redrawn, a PostScript file is also created. When the user opens a file, the PostScript file will be named the same with a ".ps" extension. If a new file is created, the PostScript file is named "tempX.ps" where X is a number. The PostScript file name is changed when the data file is saved and reloaded.

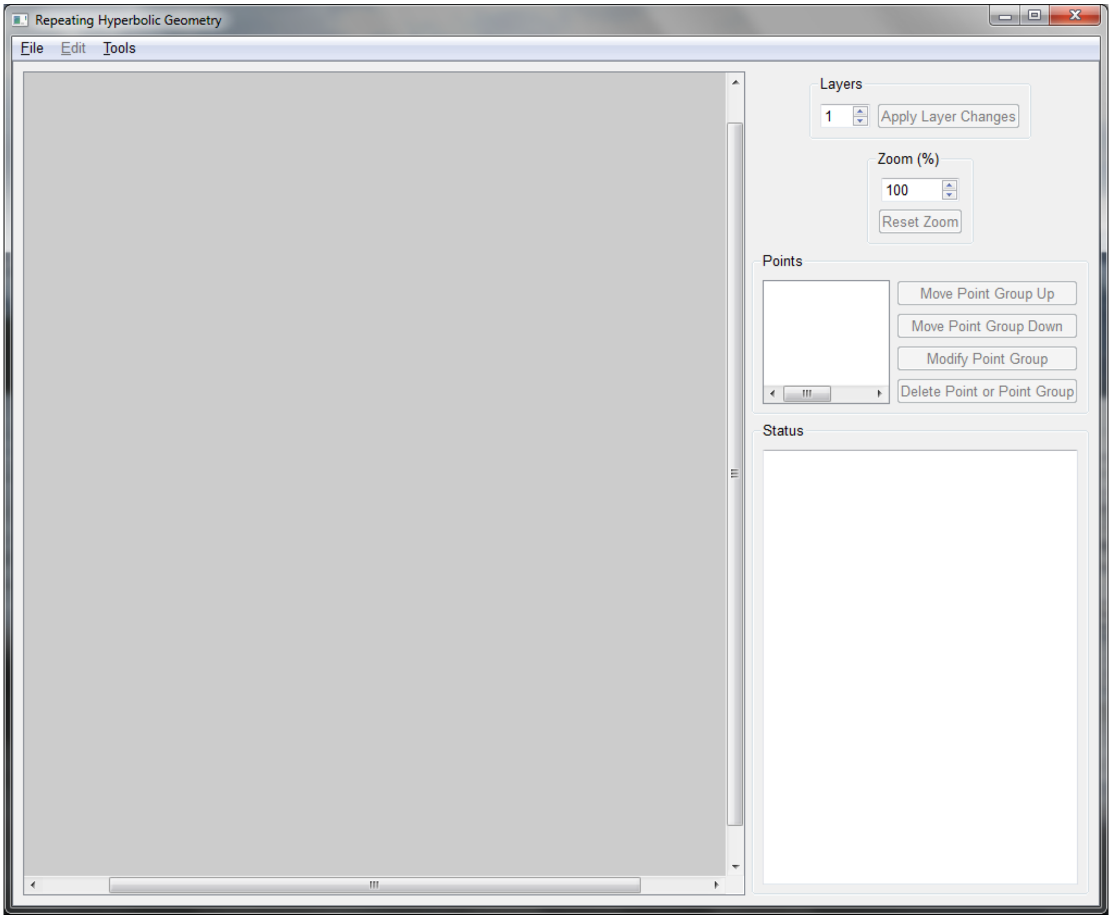

Figure 5.1: Interface when the program is first opened.

5.2.1 Creating a New Data File

When the user selects to create a new file, they are presented with the interface as shown in Figure 5.2. The controls shown on this interface automatically adjust as the user changes the values for p, q, and Number of Colors. The meanings of the values are available in more detail in Appendix A. Error checking is done when the user presses "OK." The form does not close until either the values are valid or the user presses "Cancel."

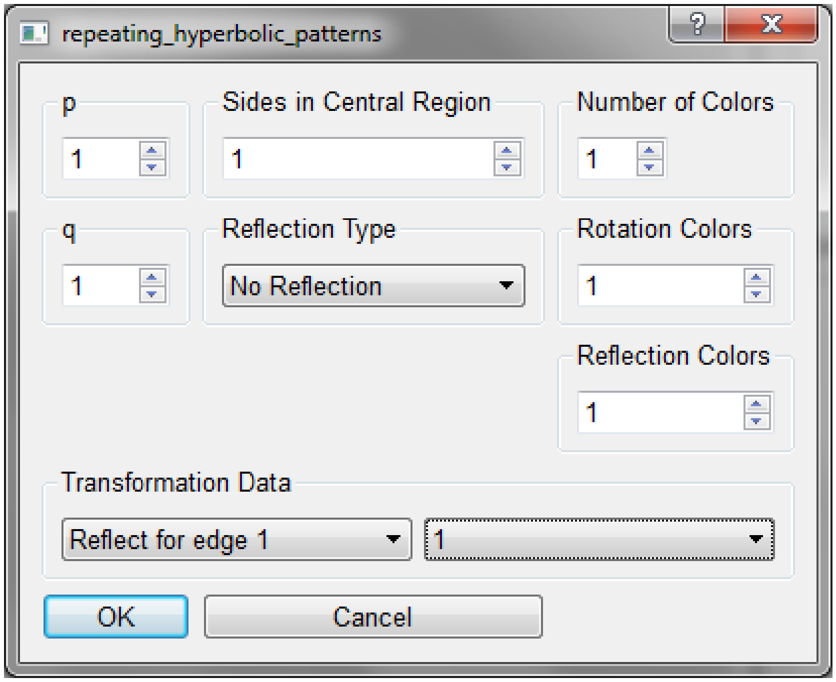

Figure 5.2: Interface for creating a new data file.

Figure 5.3 is the result of creating a new file for a pattern in the group {8, 3}. Note that the symmetry lines for the central p-gon are displayed in a grey color in the center of the circle, which is difficult to see in the example provided here, but easily visible in the PostScript file and on the screen when the program is running.

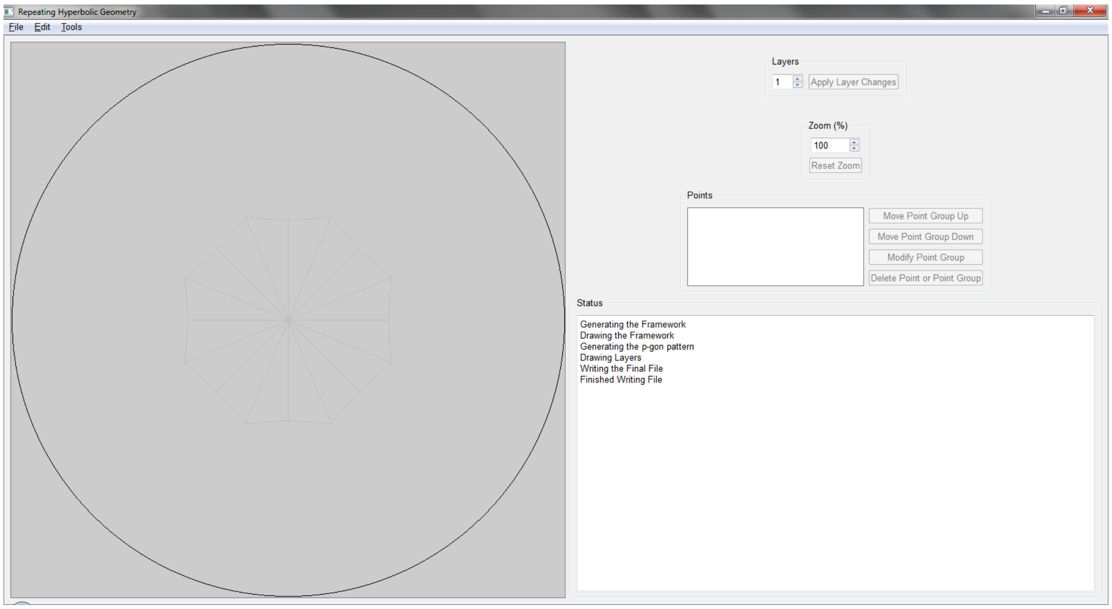

Figure 5.3: Interface after creating new file

5.2.2 Adding Shapes

Shapes (as a sequence of points) may be added by selecting the appropriate shape type under the "Edit" menu. Figure 5.4 is the form that displays when the user selects to add a circle. The number of points (and the possibility to add/subtract points) is dependent on the type of shape selected. This interface is also used if the user edits a point later. However, at that time points cannot be added or subtracted. The interface lets the user specify the color for the shape and the coordinates of the points for the shape. For more information on specifying points, refer to Appendix A. Error checking is performed when the user presses "Add/Edit Object." The form does not close until either the values are valid or the user presses "Cancel."

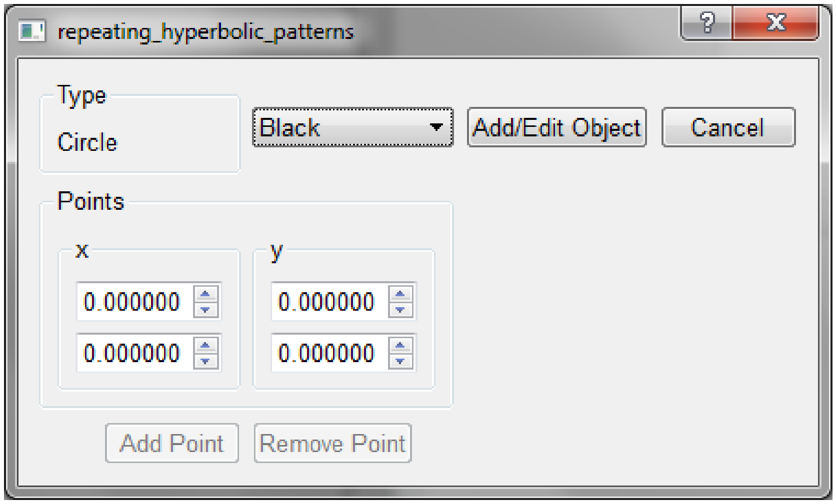

Figure 5.4: Interface for adding/editing a shape

5.2.3 Viewing Patterns

After the user has opened a file or created and added points as described in Section 5.2.1 and Section 5.2.2 above, the pattern will display the first layer of the tessellation as shown in Figure 5.5. The number of layers computed and displayed can be controlled by the “Layers” control box located at the top on the right hand side of the interface. An example of additional layers being drawn is shown in Figure 5.6. The user can also zoom in on the picture by changing the zoom value in the “Zoom” control box located on the right side of the interface under the “Layers” control box. This can be seen in Figure 5.7. Then the user can use the scroll bars to navigate to different areas of the pattern. Valid values for zoom are 100-1000000%.

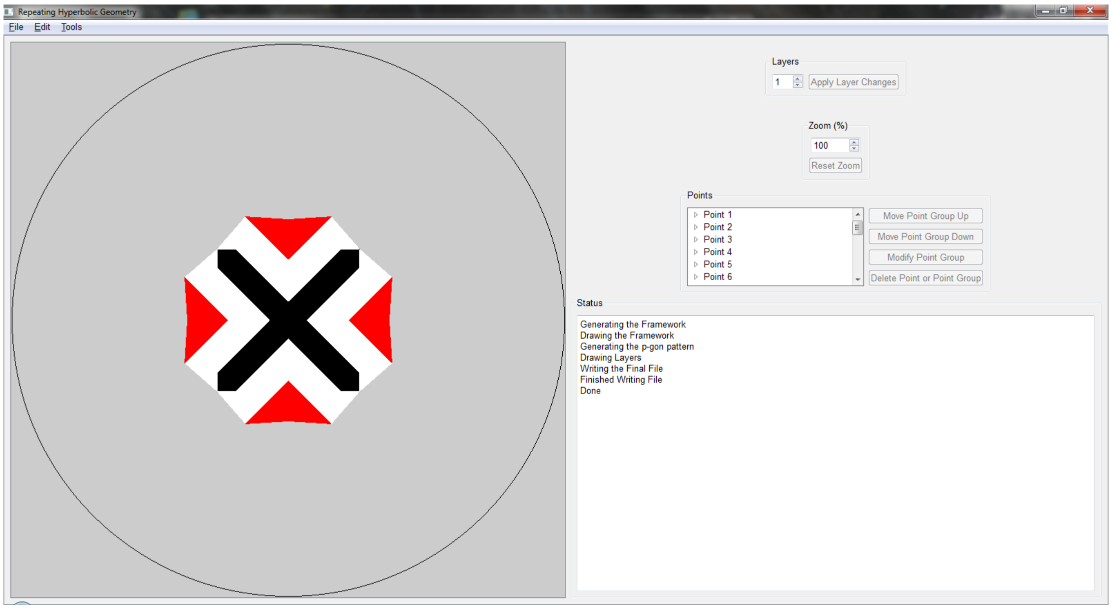

Figure 5.5: Interface after a pattern has been opened or created and points added

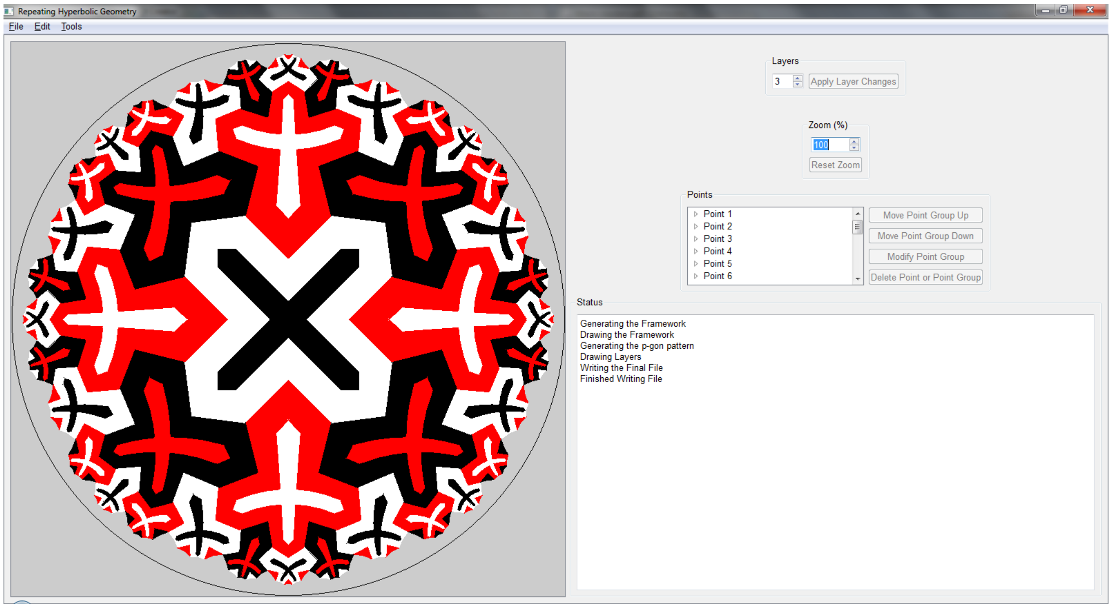

Figure 5.6: Interface with additional layers of the pattern drawn

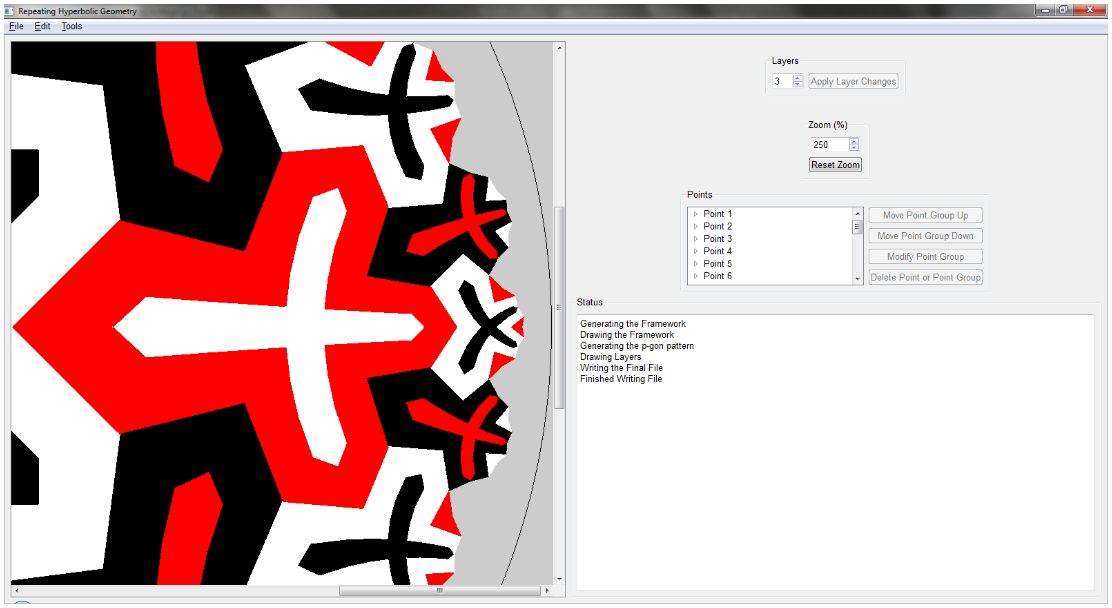

Figure 5.7: Interface after the zoom value has been changed and the user has scrolled to the right edge

5.2.4 Viewing Points and Modifying Shapes

When a pattern has points, information about those points is available by expanding the appropriate point in the "Points" control box on the right side of the interface. The information provided by expanding the point is the x and y coordinates of the point, the color for the point, and the type of point. This is shown in Figure 5.8.

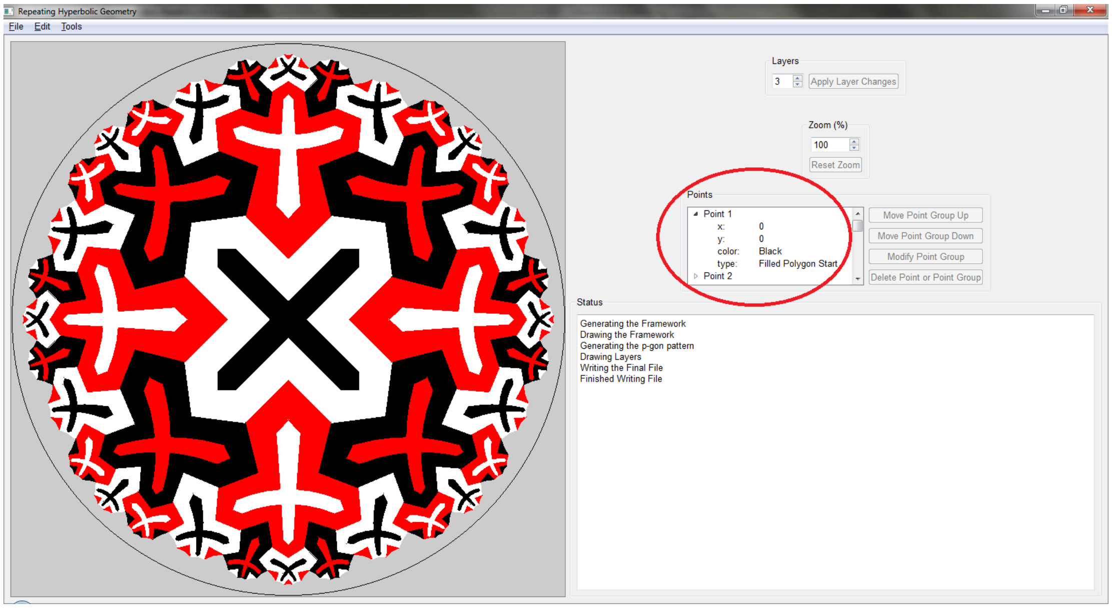

Figure 5.8: Interface after a point has been expanded (in the area circled)

When a point is selected, the buttons to the right of the list of points become enabled. The first two options, “Move Point Group Up” and “Move Point Group Down,” will change the location of the point group within the pattern listing. The third option allows the user to modify the point group using the interface that is described in Section 5.2.2. However, the user will not be able to add or remove points from the group. The last option will delete a point group (or single point if the point is a “Move To” type point).

5.2.5 Saving the Data File

The user can save the data file with a name specified, by selecting the “Save” menu item under the “File” menu.

The above provides an overview of the functionalities provided by the graphical user interface for the program.

Chapter 6

Results

As stated in the last chapter, the earlier program, called the "design" program, was written in C and uses the Motif framework for the Graphical User Interface (GUI). This limited the earlier program by allowing it to only run on the Linux/Unix platform. The new version of the program on which this thesis is based is written in C++ and uses the Qt framework for the GUI. Note that since Qt is available on standard platforms (Windows, Linux/Unix, and Mac), the new program will also compile and run on those platforms. This allows for a wider user base. Vast improvements to the earlier version were necessary in order to create the updated program. Several previous thesis papers which were read alluded to the existence of modifications that were not found in the copy of the code which was referenced. While both programs are based on an algorithm by Dr. Dunham, the new program was designed from scratch using an object-oriented approach to make the code less error-prone and easier to read. This also allows the program to more easily be converted to object-oriented languages such as Java.

While the design program had little or no error checking in place, the updated program has many error checking capabilities in place. In fact, several files that were valid according to the design program were actually invalid when compared to the data format that the program was supposed to follow and is explained in Appendix A. This was discovered when running the files through the new program with the added validation checks.

Another issue with the design program was its lack of documentation. This made the code harder to follow and understand. The new program has more detailed documentation in place.

The design program also had issues with the images generated. For instance, the framework (lines of symmetry for the central p-gon) was not always fully drawn on the screen or in the output PostScript file. The new program fixed these issues.

The design program did not allow the user to create a data file from within the program itself. In order for the program to work, the data file would have to be created manually and then loaded into the program, either as a command line argument or by using the "Load" button within. The new program, on the other hand, allows the user to create a data file from within the program itself.

The following figures show the results of the PostScript output files for various data files.

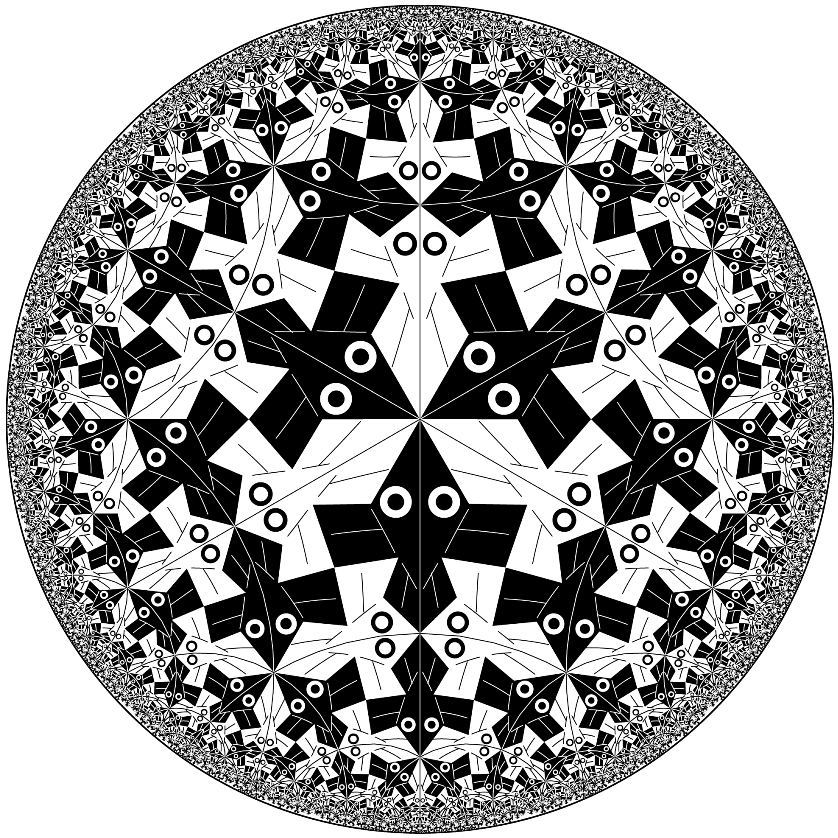

Figure 6.1: Escher's Circle Limit I pattern (with 4 layers drawn).

Figure 6.2: Escher's Circle Limit II pattern (with 4 layers drawn).

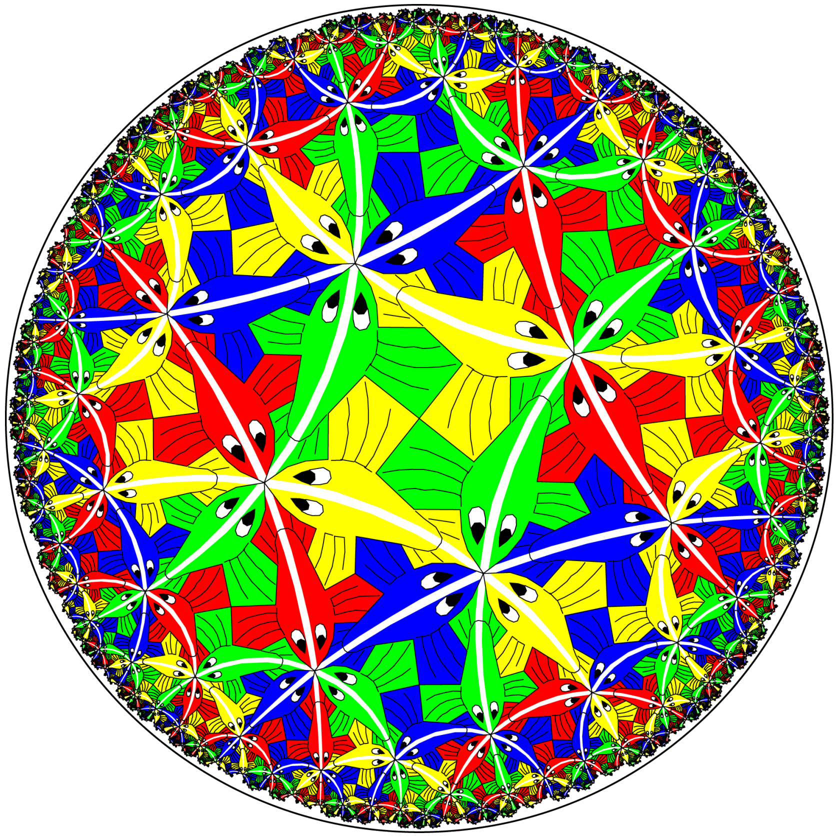

Figure 6.3: Escher's Circle Limit III pattern (with 4 layers drawn).

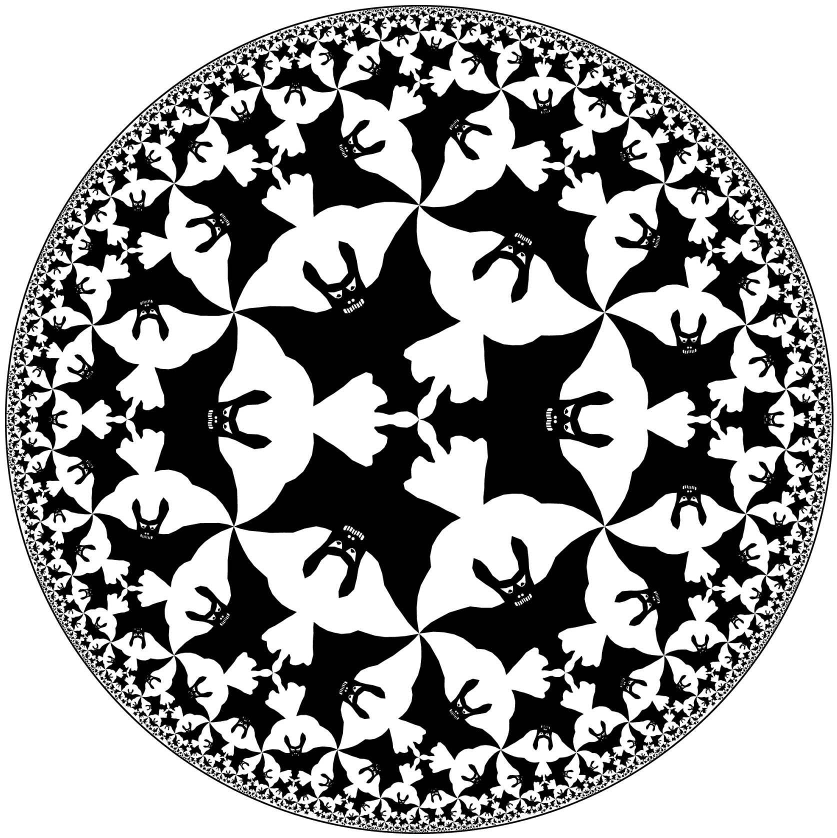

Figure 6.4: Escher's Circle Limit IV pattern (with 4 layers drawn).

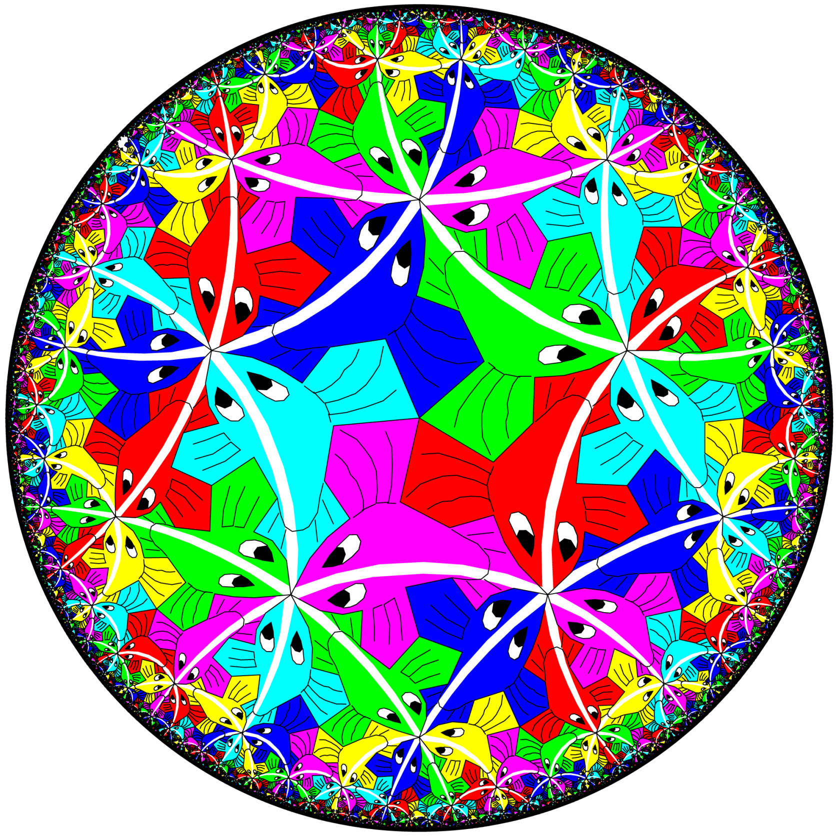

Figure 6.5: A variation on Escher's Circle Limit III pattern (with 5 layers drawn).

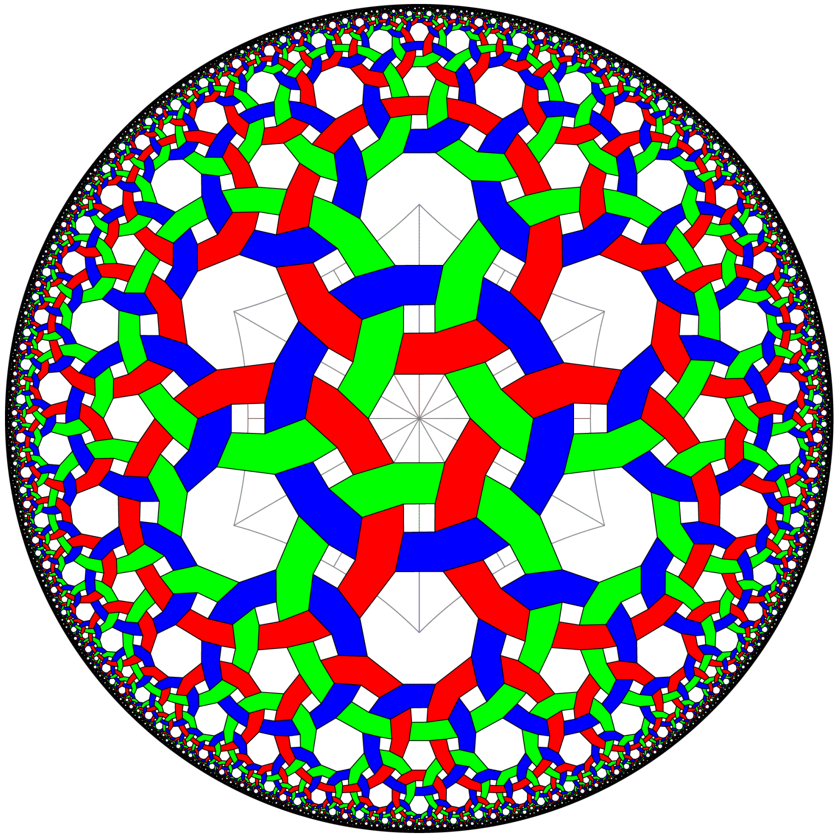

Figure 6.6: A pattern based on {6, 4} (with 5 layers drawn).

Chapter 7

Conclusion

This research involved the design and implemention of an updated version of a program that generates repeating hyperbolic patterns based on $\{p, q\}$. This updated program expanded the capabilities of the design program, fixed several issues from the design program, and made the program cross-platform. The Weierstrass Model of hyperbolic geometry was used for all calculations and the Poincaré Disk Model was used for representing the patterns.

The new program was tested using a variety of test files. The results of running valid data files were as expected. This research is a contribution to Dr. Douglas Dunham's research on creating repeating hyperbolic patterns based on regular tessellations.

Chapter 8

Future Work

This chapter discusses enhancements which can be made to the program that generates repeating hyperbolic patterns.

In general, future work could involve enhancing the user interface to be more user-friendly. This work could include "point and click" point editing and point creation, as well as specifying the zoom level by a bounding box. Future work could also focus on increasing the memory efficiency of the program. Currently, a trade off exists between the complexity of the motif and the number of layers that can be drawn without running out of memory. This varies depending on wide variety of factors, including the operating system used and the manner in which the code was compiled.

Another direction the research could take would be to allow different models, such as the Klein Model or Weierstrass Model, to be used for the graphical representation. The graphical representation of models such as the Weierstrass Model could also be done in 3-D using OpenGL.

Future work could also include converting the program to a program written in Java. This would increase cross-platform compatibility, as well as allow the program to be web-based.

References

[1] Megan Hoopes-Myers. Tour of tessellations. `http://edtech2.boisestate.edu/meganhoopesmyers/502/virtualtour/history.html`, 2010. Accessed 6-5-2012.

[2] Richard L. Faber. *Foundations of Euclidean and Non-Euclidean Geometry*. Marcel Dekker, Inc., New York, 1983.

[3] JOC/EFR. Escher biography. `http://www-gap.dcs.st-and.ac.uk/~history/Biographies/Escher.html`, May 2000. Accessed 6-5-2012.

[4] D. Dunham. Artistic patterns in hyperbolic geometry. In *Bridges 1999 Conference Proceedings*, pages 239–249, 1999.

[5] D. Dunham. Creating regular repeating hyperbolic patterns. In *Fifth Mathematics and Design Conference Proceedings*, pages 15–20, 2007.

[6] Marvin Jay Greenberg. *Euclidean and Non-Euclidean Geometries Second Edition*. W.H. Freeman and Company, New York, 1980.

[7] Richard J. Trudeau. *The Non-Euclidean Revolution*. Birkhäuser, Boston, 1987.

[8] Ajit Maranthe. Incorporating points at infinity in a hyperbolic drawing program. Master's thesis, Universiy of Minnesota Duluth, 2007.

[9] D. Dunham. Creating hyperbolic escher patterns. In *M. C. Escher: Art and Science, H. S. M. Coxeter et al., eds.*, pages 241–248, Amsterdam: North-Holland.

[10] D. Dunham. Hyperbolic symmetry. In *Computers & Mathematics with Applications, (1/2), 12B*, pages 139–153.

Appendix A

Data File Format

This section explains the format of the data file that is used by the program. Checks are used to enforce the file format.

Here is a sample data file, "cl3.dat" that creates Escher's Circle Limit III pattern:

```
8 3 2 0 8 0
1 2 5 8 3 6 7 4
1 2 3 4 5 6 7 8
2 1 2 4 8 5 6 7 3
1 1 2 8 3 5 6 7 4
2 1 2 8 4 3 6 7 5
1 1 2 4 5 3 6 7 8
2 1 2 4 8 5 6 7 3
1 1 2 8 3 5 6 7 4
2 1 2 8 4 3 6 7 5
1 1 2 4 5 3 6 7 8
172
0.000000e+00 -0.000000e+00 4 4 3
-1.048625e-01 -1.322021e-01 4 5 3
-1.233043e-02 -2.196813e-01 4 5 3
9.907347e-02 -2.205755e-01 4 5 3
1.028408e-01 -2.747673e-01 4 5 3
```

1.170408e-01 -3.188163e-01 4 5 3

1.552227e-01 -3.747407e-01 4 5 3

2.124825e-01 -3.316812e-01 4 5 3

2.475278e-01 -2.892804e-01 4 5 3

2.664085e-01 -2.490270e-01 4 5 3

3.281691e-01 -2.710616e-01 4 5 3

4.152348e-01 -2.281971e-01 4 5 3

3.747407e-01 -1.552227e-01 4 5 3

2.850998e-01 -1.527025e-01 4 5 3

3.039132e-01 -4.466229e-02 4 5 3

3.647928e-01 -1.228722e-02 4 5 3

3.501220e-01 3.026425e-02 4 5 3

3.502589e-01 8.677537e-02 4 5 3

3.747407e-01 1.552227e-01 4 5 3

3.188163e-01 1.170408e-01 4 5 3

2.747673e-01 1.028408e-01 4 5 3

2.205755e-01 9.907347e-02 4 5 3

2.196813e-01 -1.233043e-02 4 5 3

1.322021e-01 -1.048625e-01 4 5 3

0.000000e+00 -0.000000e+00 4 6 3

3.747407e-01 1.552227e-01 2 4 3

3.507592e-01 1.387102e-01 2 5 3

3.011020e-01 7.063673e-02 2 5 3

2.612816e-01 2.057143e-03 2 5 3

2.170367e-01 -9.306939e-02 2 5 3

1.772163e-01 -2.014694e-01 2 5 3

1.595184e-01 -2.788980e-01 2 5 3

1.470082e-01 -3.625306e-01 2 5 3

1.552227e-01 -3.747407e-01 2 5 3

1.671755e-01 -3.655061e-01 2 5 3

1.816408e-01 -2.811102e-01 2 5 3

2.015510e-01 -2.036816e-01 2 5 3

2.303102e-01 -1.151918e-01 2 5 3

2.789796e-01 -9.004082e-03 2 5 3

3.232245e-01 6.842449e-02 2 5 3

3.647184e-01 1.275429e-01 2 5 3

3.747407e-01 1.552227e-01 2 6 3

3.167653e-01 -1.326766e-02 1 1 3

3.479433e-01 8.951438e-03 1 2 3

-2.259060e-02 -6.958546e-02 1 9 3

1.847494e-02 -7.775483e-02 1 10 3

5.518467e-02 -9.403069e-02 1 10 3

9.105802e-02 -1.106827e-01 1 10 3

1.177891e-01 -1.293974e-01 1 10 3

1.038273e-01 -1.561682e-01 1 10 3

8.104628e-02 -1.438736e-01 1 10 3

4.427858e-02 -1.296729e-01 1 10 3

1.101375e-02 -1.231180e-01 1 10 3

-1.830104e-02 -1.229831e-01 1 10 3

-4.889938e-02 -1.244500e-01 1 10 3

-8.830872e-02 -1.309044e-01 1 11 3

3.228850e-01 2.841844e-02 1 1 3

3.468254e-01 5.762810e-02 1 2 3

-5.029059e-02 -1.742552e-01 1 9 3

-2.994414e-02 -1.702271e-01 1 10 3

-9.266454e-03 -1.691238e-01 1 10 3

1.291010e-02 -1.734937e-01 1 10 3

3.526055e-02 -1.716380e-01 1 10 3

6.156105e-02 -1.762024e-01 1 10 3

8.373761e-02 -1.805723e-01 1 10 3

1.008959e-01 -1.871993e-01 1 10 3

1.014258e-01 -2.125752e-01 1 11 3

1.046058e-01 -1.578673e-01 1 1 3

9.916513e-02 -1.900268e-01 1 2 3

1.293411e-01 -1.149806e-01 1 1 3
1.181204e-01 -1.323221e-01 1 2 3
3.652341e-01 -1.851126e-01 1 9 3
3.224378e-01 -1.797708e-01 1 10 3
2.877155e-01 -1.810434e-01 1 10 3
2.845687e-01 -2.049993e-01 1 10 3
3.234150e-01 -2.039211e-01 1 10 3
3.587418e-01 -2.105732e-01 1 10 3
4.032522e-01 -2.284635e-01 1 11 3
2.739773e-01 5.150307e-02 1 1 3
2.925848e-01 9.675649e-02 1 2 3
2.426254e-01 2.305833e-02 1 1 3
2.522231e-01 8.577557e-02 1 2 3
2.823908e-01 -2.090525e-01 1 9 3
2.769088e-01 -2.279112e-01 1 10 3
3.085588e-01 -2.331433e-01 1 10 3
3.344122e-01 -2.389333e-01 1 10 3
3.613172e-01 -2.514223e-01 1 11 3
1.602245e-01 -2.640816e-01 1 4 3
1.481020e-01 -2.510612e-01 1 5 3
1.332857e-01 -2.517347e-01 1 5 3
1.240816e-01 -2.649796e-01 1 5 3
1.294694e-01 -2.847347e-01 1 5 3
1.409184e-01 -3.062857e-01 1 5 3
1.530408e-01 -2.840612e-01 1 6 3
1.602245e-01 -2.640816e-01 2 4 3
1.481020e-01 -2.510612e-01 2 5 3
1.332857e-01 -2.517347e-01 2 5 3
1.240816e-01 -2.649796e-01 2 5 3
1.186939e-01 -2.319796e-01 2 5 3
1.276735e-01 -2.167143e-01 2 5 3
1.442857e-01 -2.131224e-01 2 5 3

1.561837e-01 -2.223265e-01 2 5 3

1.600000e-01 -2.402857e-01 2 5 3

1.602245e-01 -2.640816e-01 2 6 3

1.240816e-01 -2.649796e-01 1 9 3

1.186939e-01 -2.319796e-01 1 10 3

1.276735e-01 -2.167143e-01 1 10 3

1.442857e-01 -2.131224e-01 1 10 3

1.561837e-01 -2.223265e-01 1 10 3

1.600000e-01 -2.402857e-01 1 10 3

1.602245e-01 -2.640816e-01 1 11 3

2.243184e-01 -2.845224e-01 1 4 3

2.205224e-01 -2.678204e-01 1 5 3

2.088816e-01 -2.652898e-01 1 5 3

1.957224e-01 -2.744000e-01 1 5 3

1.929388e-01 -2.900898e-01 1 5 3

1.926857e-01 -3.118531e-01 1 5 3

2.114122e-01 -2.981878e-01 1 6 3

2.243184e-01 -2.845224e-01 2 4 3

2.205224e-01 -2.678204e-01 2 5 3

2.088816e-01 -2.652898e-01 2 5 3

1.957224e-01 -2.744000e-01 2 5 3

2.007837e-01 -2.485878e-01 2 5 3

2.106531e-01 -2.339102e-01 2 5 3

2.258367e-01 -2.321388e-01 2 5 3

2.344408e-01 -2.409959e-01 2 5 3

2.349469e-01 -2.582041e-01 2 5 3

2.314041e-01 -2.708571e-01 2 5 3

2.243184e-01 -2.845224e-01 2 6 3

1.957224e-01 -2.744000e-01 1 9 3

2.007837e-01 -2.485878e-01 1 10 3

2.106531e-01 -2.339102e-01 1 10 3

2.258367e-01 -2.321388e-01 1 10 3

2.344408e-01 -2.409959e-01 1 10 3

2.349469e-01 -2.582041e-01 1 10 3

2.314041e-01 -2.708571e-01 1 10 3

2.243184e-01 -2.845224e-01 1 11 3

2.239766e-01 -8.894270e-03 1 9 3

2.431720e-01 1.305842e-02 1 10 3

2.629305e-01 4.038723e-02 1 10 3

2.893634e-01 5.534937e-02 1 10 3

3.004517e-01 5.316439e-02 1 10 3

3.162105e-01 4.078513e-02 1 10 3

3.210383e-01 2.144046e-02 1 10 3

3.161608e-01 -5.342965e-03 1 10 3

3.106788e-01 -2.420170e-02 1 10 3

3.039132e-01 -4.466229e-02 1 11 3

0.000000e+00 -0.000000e+00 1 9 3

-1.048625e-01 -1.322021e-01 1 10 3

-1.233043e-02 -2.196813e-01 1 10 3

9.907347e-02 -2.205755e-01 1 10 3

1.028408e-01 -2.747673e-01 1 10 3

1.170408e-01 -3.188163e-01 1 10 3

1.552227e-01 -3.747407e-01 1 10 3

2.124825e-01 -3.316812e-01 1 10 3

2.475278e-01 -2.892804e-01 1 10 3

2.664085e-01 -2.490270e-01 1 10 3

3.281691e-01 -2.710616e-01 1 10 3

4.152348e-01 -2.281971e-01 1 10 3

3.747407e-01 -1.552227e-01 1 10 3

2.850998e-01 -1.527025e-01 1 10 3

3.039132e-01 -4.466229e-02 1 10 3

3.647928e-01 -1.228722e-02 1 10 3

3.501220e-01 3.026425e-02 1 10 3

3.502589e-01 8.677537e-02 1 10 3

3.747407e-01 1.552227e-01 1 10 3

3.188163e-01 1.170408e-01 1 10 3

2.747673e-01 1.028408e-01 1 10 3

2.205755e-01 9.907347e-02 1 10 3

2.196813e-01 -1.233043e-02 1 10 3

1.322021e-01 -1.048625e-01 1 10 3

0.000000e+00 -0.000000e+00 1 11 3

The pattern corresponding to the above data file is shown below:

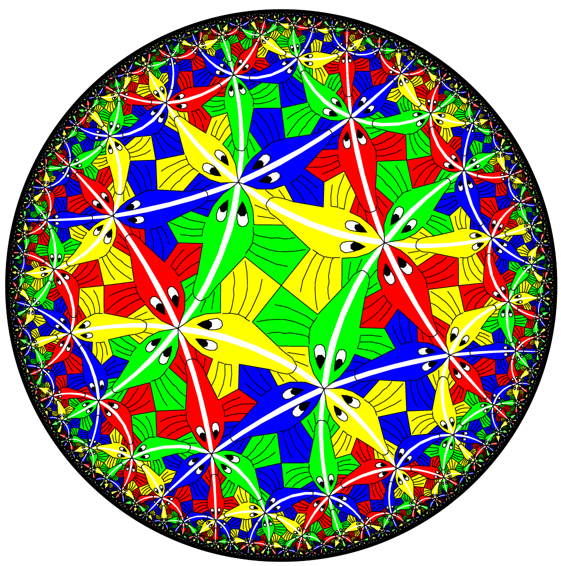

Figure A.1: Escher's Circle Limit III pattern.

In the first line, 8 3 2 0 8 0:

- The first number is the value of p, i.e. p = 8 in this case. The central polygon in Figure A1 is an 8-gon
- The second number is the value of q, i.e. q = 3 in this case (this pattern is based on the tessellation {8, 3}). The 8-gon in Figure A.1 meets three other 8-gons at each vertex.
- The third number, 2 in this case, is the number of "different" sides of the central

p-gon that are used to form the fundamental region (the other sides of the fundamental region are two radii from the center to two vertices of the central p-gon separated by $2*(2*\pi/p)$). This number must divide p, and p divided by this number is the number of copies of the motif that appears in the central p-gon.

- The fourth number is not used and is there merely to maintain compatibility with older versions of the program.
- The fifth number, 8 in this case, must be the highest "color" number of the colors used. The color numbers are:
  - 1 Black
  - 2 White
  - 3 Red
  - 4 Green
  - 5 Blue
  - 6 Cyan
  - 7 Magenta
  - 8 Yellow
  - 9 Salmon
  - 10 Brown
- The sixth number, 0 in this case, indicates the kind of reflection symmetry the pattern has within the central p-sided polygon:
  - 0 indicates that there is no reflection symmetry (only rotation symmetry).
  - 1 indicates that there is reflection symmetry across the perpendicular bisector of one of the edges of the p-gon.
  - 2 indicates that there is reflection symmetry across a radius (from the center to a vertex of the p-sided polygon).

The second line, 1 2 5 8 3 6 7 4, is the color permutation induced by rotating by $2*(2*\pi/p)$ (i.e., the third number of line 1 times $2*(2*\pi/p)$. Note that this is the "array" representation of permutations (not the "mathematical" one using cycles): the values listed are the values of perm[1], perm[2], etc.

The third line, 1 2 3 4 5 6 7 8, is the color permutation induced by the reflection, if the sixth number of line 1 is 1 or 2 (it is just the identity, if the sixth number is 0).

The next p lines consist of a first number followed by a color permutation. The first number of the first of these lines indicates which edge (edge 2 in this case) of the transformed p-gon should lie next to edge 1 of the central p-gon. In general, if this first number is positive, the transformed p-gon is rotated into position; if the number is negative, a reflection is used to move the transformed p-gon into position. Note that the edges are numbered from 1 to p, not from 0 to p-1, so that the edges can be assigned an unambiguous sign (i.e. 0 is not used as +0 = -0). The next eight numbers, 1 2 4 8 5 6 7 3 (perm[1]=1, perm[2]=2, perm[3]=4, etc.), define the color permutation that will be induced when we go across this edge. The initial color permutation is always assumed to be the identity permutation.

The first number of the second of these lines indicates which edge (edge 1 in this case) of the transformed p-gon should lie next to edge 2 of the central p-gon. In this case, the color permutation is 1 2 8 3 5 6 7 4. This pattern continues for six more lines.

The next line consists of a single number, the number of points that make up the motif. It is 172 in this case. Following that line are 172 lines of five numbers each; each line specifies one point. Each line has the following format: x-coordinate y-coordinate color point-type number-of-layers

- The x-coodinate and y-coordinate are within the central p-gon (and hence the unit circle).
- The color is one of the color numbers discussed previously.
- The point-type is one of:
  - 1 "Move To"

  - 2 “Draw To”
  - 3 “Circle” (there must be two of these in succession)
  - 4 Start a (Euclidean) “Filled Polygon”
  - 5 Continue a (Euclidean) “Filled Polygon”
  - 6 End a (Euclidean) “Filled Polygon”
  - 7 “Hyperline” (there must be two of these in succession)
  - 8 “Filled Circle” (there must be two of these in succession)
  - 9 Start a (Euclidean) “Polyline”
  - 10 Continue a (Euclidean) “Polyline”
  - 11 End a (Euclidean) “Polyline”
  - 12 Start a (hyperbolic) “Filled p-gon”
  - 13 Continue a (hyperbolic) “Filled p-gon”
  - 14 End a (hyperbolic) “Filled p-gon”

- The number-of-layers is not used and is there merely to maintain compatibility with older versions of the program.

Appendix B

Definitions

- **Isometry** – Also known as a congruence transformation. A distance-preserving bijective map between two metric spaces (i.e., $d(f(x), f(y)) = d(x, y)$ where f is the map and $d(x, y)$ is the distance function). If two figures can be transformed into each other using an isometry, they are said to be congruent.

- **Lorenz Matrix** – A matrix that maintains the following property: $x^2+y^2-z^2 = c$ where c is some constant after following matrix multiplication:

$$\begin{bmatrix} Lorenz \\ Matrix \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

  That is, $x^2 + y^2 - z^2 = x'^2 + y'^2 - z'^2 = c$.