An Interactive Java Program to Generate Hyperbolic Repeating Patterns Based on Regular Tessellations Including Hyperbolic Lines and Equidistant Curves

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

Sakethram Karumuri

IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE
Dr. Douglas Dunham
July, 2015

Acknowledgements

I would like to gratefully acknowledge my advisor, Dr. Douglas Dunham whose constant guidance and support helped me in completing this thesis. Without his persistent help, this work would not have been completed.

I would like to thank my committee members, Dr. Peter Willemsen and Dr. Steven Trogdon for their support and encouragement. I would like to thank my exceptional department faculty, especially Dr. Ted Pedersen, Dr. Huayang Wang and Dr. Hudson Turner for teaching me the nitty gritty of Computer Science concepts. I would like to thank Lori Lucia and Clare Ford for providing support in times of need.

I would also like to thank all my friends and fellow graduate students especially Shiva Kumar Chittamuru and Mounika Alla for sharing their knowledge and experience which helped me to complete my thesis.

Finally, I would like to thank my parents and my sister for their unconditional trust and timely encouragement. It was their love that helped me rise up during the times of adversity.

Dedicated to

my mom,

Mrs. Savitri Karumuri,

my dad,

Mr. Syam Babu Karumuri,

and my sister,

Mrs. Prathyusha Gunishetty

Abstract

Repeating patterns have been utilized as art by various cultures all through the history. Amidst the 20th century, the prominent Dutch artist M.C. Escher was the first to make repeating hyperbolic patterns that were creative in nature. These patterns were very hard to draw using a hand without the help of a PC. Previously, some work was done in C, C++ and Java that automates the process of drawing these repeated patterns.

This research focuses on enhancing the Java program by enabling user interaction and by providing support for special lines and curves such as hyperbolic lines and equidistant curves. This program also allows the user to create a new data file or load an existing data file that contains information about the repeating hyperbolic pattern. It also allows the user to make changes in the pattern by using a graphical user interface.

Table of Contents

List of Figures

Chapter 1

Introduction

The practice of generating repeating patterns dates back to the Bronze Age. In the Mesopotamian Valley, Sumerians embellished their structures with mosaics in geometric patterns made of hardened clay. A lot of societies like Chinese, Roman, Japanese, Greek, Egyptian, and Arabic have been using the geometric tessellations to decorate their houses. These examples were Euclidean in nature. The Greek mathematician Euclid is the founder of the well known Euclidean Geometry. Euclid's *Elements* is a famous book on Euclidean Geometry that served as a textbook. Euclidean geometry is for the most part the most helpful approach to portray the physical world around us as it manages two dimensional and three dimensional articles.

M. C. Escher, a famous Dutch graphic artist is well known for his woodcuts, lithographs and impossible constructions that are based on Euclidean tilings. Most of his early work was based on tessellations which were formed by employing repeated tilings in the Euclidean plane [4]. After meeting a Canadian mathematician H. S. M. Coxeter, he was inspired and developed interest in drawing hyperbolic tessellations which are regular tilings in the hyperbolic plane. The most popular work on hyperbolic tessellations were his Circle Limit I - IV patterns which were accurate to millimeters.

Even though Escher did not have a proficiency in Mathematics, he drew those patterns using hand by visualizing and understanding the intricacies of hyperbolic geometry and art [5]. As Escher lacked any assistance from a computer, he spent days and months to produce a complex pattern with many details.

Many mathematicians were interested in his work and they tried to generate his patterns using a computer program. One such algorithm was developed by Dr. Dunham [1][3][6], which was earlier implemented in various languages like C, C++ and Java. All the older programs supported a limited number of shapes and had many shortcomings. My current thesis work focuses on extending one such existing Java application which was originally written by Vejendla [8] by addressing those shortcomings and by adding other special lines and curves such as Equidistant Curves and Hyperbolic lines. The current program also has many interaction capabilities which make it more convenient to the user to generate the patterns. Adding support to these new shapes makes the program compatible with input data files that include these special lines and curves. A sample of program output is shown in Figure 1.1.

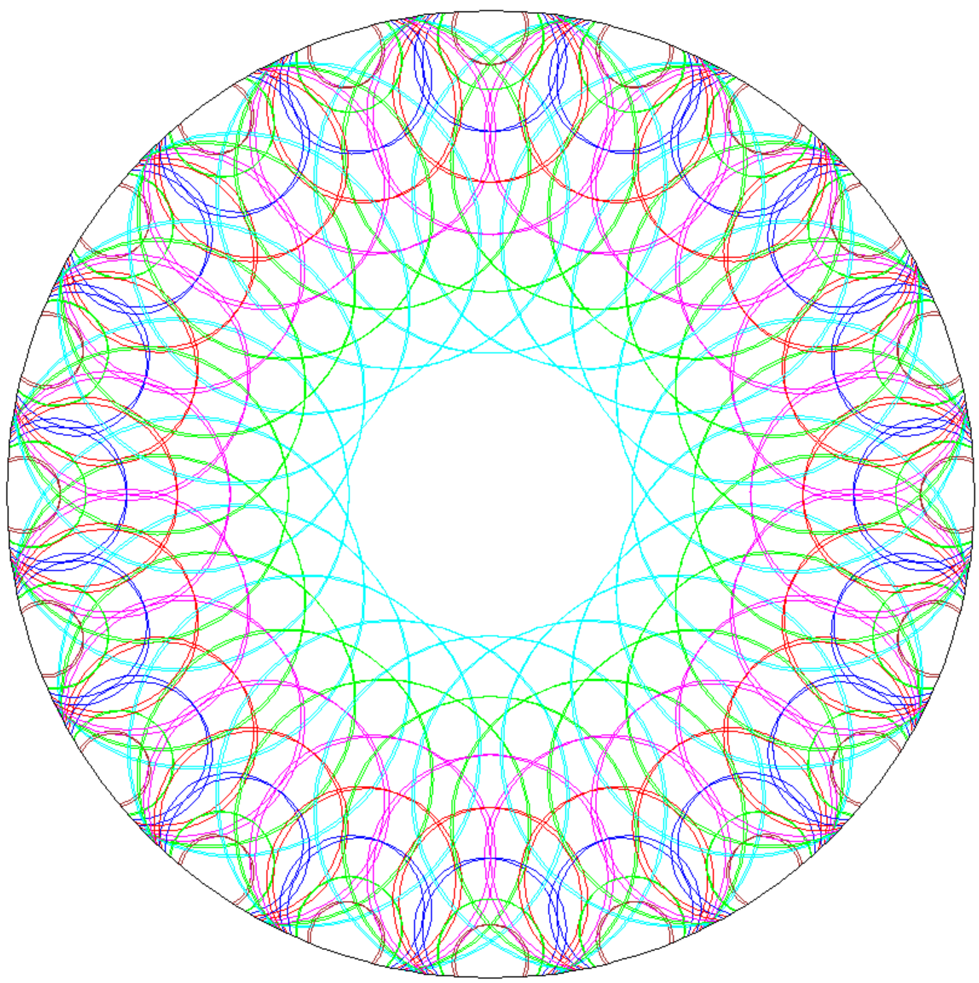

Figure 1.1: A Sample of Program Output

Chapter 2

Geometry

"Geometry" is initially from the Greek word "geometrein" which signifies "to quantify earth." It focuses on the study of shapes, sizes and spatial properties. Ancient geometry was essentially a set of principles and methods that could be utilized to help with daily applications. In spite of the fact that the Egyptians are credited just like the first individuals to think about geometry by the Greek history specialist Herodotus, numerous civilizations thought about geometric standards. Today, geometry is a branch of math that studies shapes in spaces of different measurements and sorts. The most well-known sorts of geometries are: Euclidean geometry, Hyperbolic geometry, Spherical geometry, Symplectic geometry, etc [13].

2.1 Euclidean Geometry

All the hypotheses in the Euclidean Geometry are derived from a fixed number of axioms. Hence it is called an axiomatic system.

There are five postulates (axioms) stated by Euclid that forms basis for the Euclidean Geometry.

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

For more than two thousand years, no other geometry was known except Euclidean geometry and hence the word “Euclidean” was not used. Mathematicians thought about the parallel postulate as an extraordinary postulate, varying from the initial four postulates. They didn't question that it was genuine, yet they suspected that it was a theorem instead of an axiom. In the event that it truly were a theorem, then it could be demonstrated and not simply assumed. In the late 18th century, this issue of demonstrating the parallel postulate had been tried by numerous mathematicians. This issue made a few mathematicians accept that demonstrating the parallel postulate was impossible. This did not at all imply that it couldn't be demonstrated, yet this thought prompted the disclosure of non-Euclidean geometries. Since neutral geometry, comprising of Euclid's initial four axioms, did not in itself infer the parallel postulate, mathematicians suspected that there must be an alternate geometry that was based on the initial four axioms and the negation of the parallel postulate.

2.2 Non - Euclidean Geometry

At the point when numerous individuals are chipping away at the same issue with almost no correspondence between them, various independent discoveries are made at almost the same time. This has been apparent various times all through the historical backdrop of Mathematics and Science.

While non-Euclidean geometry is actually the investigation of any geometry that is definitely not Euclidean, a standout amongst the most helpful non-Euclidean geometries is Hyperbolic geometry and Spherical geometry. Hyperbolic geometry is the geometry found by Bolyai, Gauss, Lobachevsky, and Schweikart and is the geometry of hyperbolic space. Spherical geometry is the geometry of two dimensional surface of a sphere.

2.2.1 Hyperbolic Geometry

Hyperbolic geometry is the geometry of saddle shaped surfaces that have a constant negative Gaussian curvature as shown in Figure 2.1.

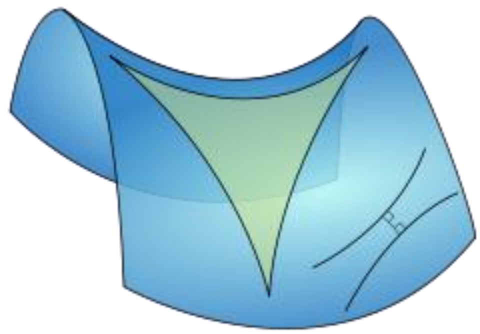

Figure 2.1: Hyperbolic plane

All the first four Euclidean postulates hold true for hyperbolic geometry but not the fifth one. It is a non- Euclidean geometry based on the fact that the parallel postulate does not hold on a hyperbolic plane. Rather, hyperbolic geometry is based on the converse of parallel postulate also called as the hyperbolic axiom which 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 2.2.

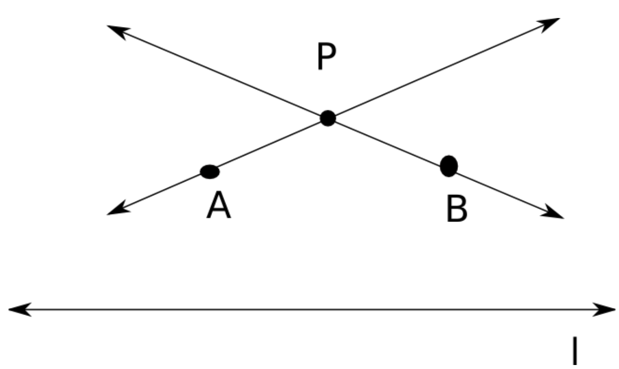

Figure 2.2: Hyperbolic Axiom

Many other useful properties of hyperbolic geometry are proved using this hyperbolic axiom. Some of the important properties of hyperbolic geometry are:

1. The sum of the angles of a triangle is less than 180° [9].
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.

Hyperbolic geometry is interpreted using various models such as Poincaré Disk, Beltrami Klein and Weierstrass models which are discussed in detail in the next chapter.

2.2.2 Spherical Geometry

Spherical geometry is another useful non-Euclidean geometry that has numerous applications in navigation and astronomy. It is the study of geometry on the surface of a sphere. However, the smallest partitions on a spherical surface can be approximated by a two dimensional plane. The lines in spherical geometry are defined as the great circles which are the biggest circles that can be drawn on a sphere. A segment in a great circle is the shortest curve that connects two points on a sphere. Great circles have their centers located at the center of the sphere. Examples for the great circles are the longitudinal lines and equator on the surface of Earth. It is also important to note that latitudinal lines are not great circles and hence not considered as lines. Figure 2.3 demonstrates the lines / great circles in spherical geometry.

Only second and fourth of the Euclid's five postulates are obeyed in the Spherical geometry. First, third and fifth postulates are violated due to various obvious facts. Unlike Euclidean and Hyperbolic geometries, Spherical geometry does not have any concept of parallel lines i.e., there is no point through which a line can be drawn that never intersects a given line.

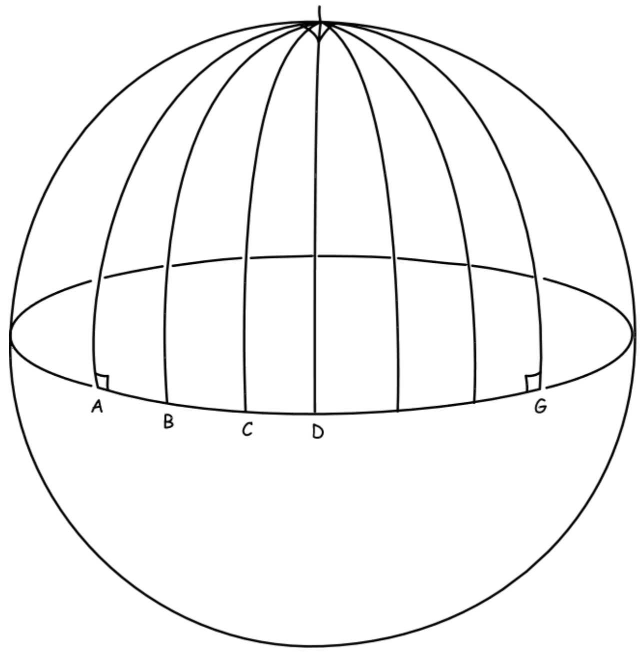

Figure 2.3: Lines in Spherical geometry

A few interesting properties of Spherical geometry are listed below:

1. Sum of interior angles in a triangle is always greater than 180° and less than 540° [10].
2. Any two lines intersect only at two diametrically opposite points called the anti nodal points.
3. Two triangles with the same angle sum are equal in area.

2.3 Comparison of Parallel Postulate

Figure 2.4 compares the Euclid's fifth postulate or the parallel postulate in three geometries.

1. Euclidean - Consistent (only one parallel line exist)
2. Spherical - Inconsistent (no parallel lines exist)
3. Hyperbolic - Inconsistent (at least two parallel lines exist)

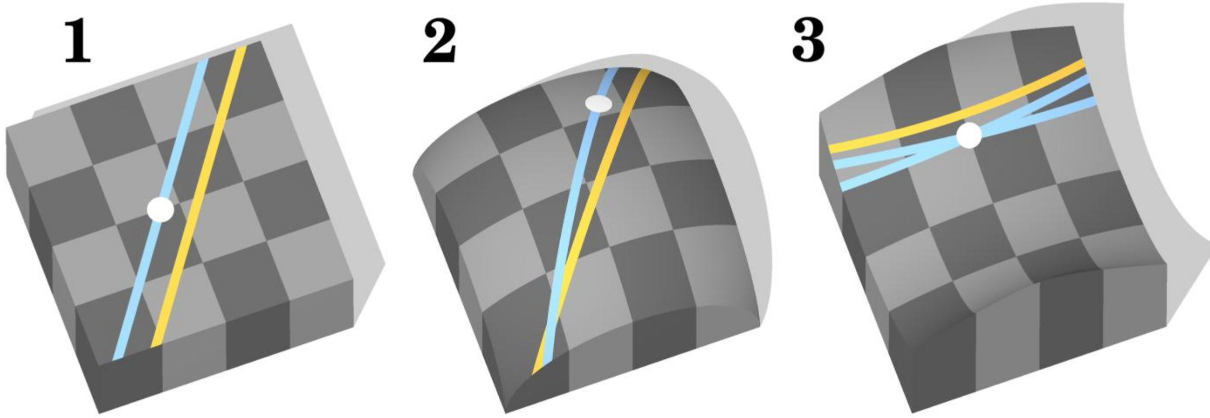

Figure 2.4: Comparison of parallel postulate

Chapter 3

Hyperbolic Geometry Models

Hyperbolic geometry models are used to define hyperbolic space in order to satisfy the axioms of hyperbolic geometry. There are many models to represent the hyperbolic space out of which very few are mostly used. Models can be categorized into finite and infinite models based on their nature. For example, the Poincaré Disk and Klein Models are finite models as they have a boundary, whereas the Weierstrass Model is an infinite model and is embedded in three dimensional Euclidean space. Objects from one model can be projected onto another model through a process called as isomorphism. Isomorphism between Poincaré Disk Model, Klein Model and Weierstrass Models are discussed in the later sections.

3.1 The Poincaré Disk Model

Poincaré Disk Model was created by a French mathematician named Henri Poincaré. It is considered to be the easiest of all models to represent the hyperbolic space. It is sometimes called as the Poincaré Ball Model for representing n - dimensional hyperbolic geometry. This model is conformal due to the fact that the angles are represented accurately and hence it is also known as Conformal Disk Model. However, distances are distorted in this model.

A *point* in this model have a similar meaning as in Euclidean geometry and it is any point inside the Euclidean unit circle. Any point P(x, y) satisfies the condition, $x^2 + y^2 < 1$.

A *line* in this model can be represented in two ways. It can either be represented as the diameter (an open chord passing through the center) of the unit circle or as an open arc of a circle orthogonal to the boundary of the unit circle [9].

The terms *lies on* and *between* have a similar meaning as in Euclidean geometry. The angle between two intersecting lines can be measured as the angle between the tangents drawn to the arcs at the point of intersection.

The lines in Poincaré Disk Model are shown in Figure 3.1.

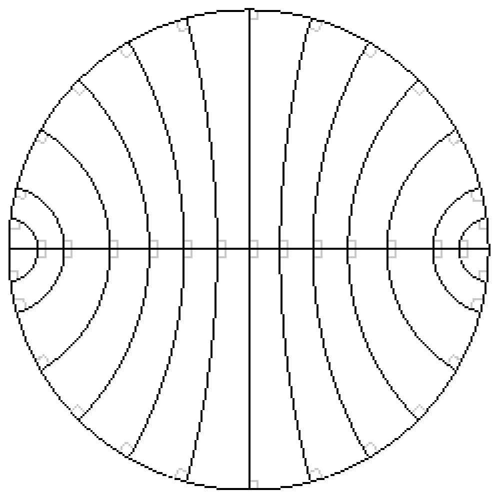

Figure 3.1: Lines in Poincaré Disk Model

3.2 The Beltrami - Klein Model

This Model was created by a German mathematician Felix Klein. It is also known as the Klein Model. This model is similar to the Poincaré Disk Model in many ways.

A *point* in this model is represented as any point inside the Euclidean unit circle centered at the origin. Any point P(x, y) satisfies the condition, $x^2 + y^2 < 1$.

A *line* in this model is an open chord on the unit circle [9]. An open chord is a closed chord minus its endpoints. The terms *lies on* and *between* have a similar meaning as in Euclidean geometry.

This model is not conformal as the angles between lines are not represented accurately by this model.

The lines in the Klein Model are shown in Figure 3.2.

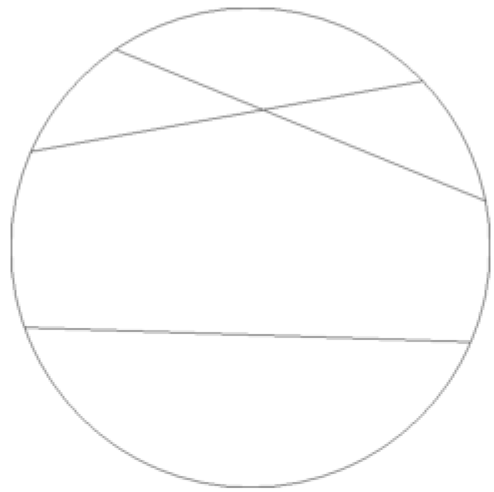

Figure 3.2: Lines in Klein Model

3.3 The Weierstrass Model

The Weierstrass Model is an infinite model of hyperbolic geometry as the hyperbolic plane is defined on the surface of a hyperboloid which expands in three dimensions infinitely. Even though this model uses a three dimensional space, it has many advantages over the other two models. It has a few properties in common with the sphere for elliptic geometry. Also, the above two models can be obtained by projections of this model.

The parametric equation of a hyperboloid is given by:

$$\langle X, X\rangle = x^2 + y^2 - z^2 = - K^2$$

where, X(x, y, z) is a point on the hyperboloid

The above equation produces two sheets of hyperboloids about the origin (one on either side of the XY plane). As the points on the lower sheet are just the reflections of the points on the upper sheet, it is discarded. Hence the above is equation is reduced to represent just the upper sheet [11] which is given by:

$$\langle X, X\rangle = -K^2 \text{ and } z > 0$$

A *point* in this model is represented using the above parametric equation. Any point which satisfies the equation lies on the hyperboloid.

A *line* in this model can be obtained by intersecting a plane with the hyperboloid through the origin. If a point satisfies the below equation, then line L *lies on* the plane.

$$\langle X, L\rangle = 0 \text{ and } z > 0$$

The brown hyperbola on the hyperboloid in Figure 3.3 represents a line in the Weierstrass Model.

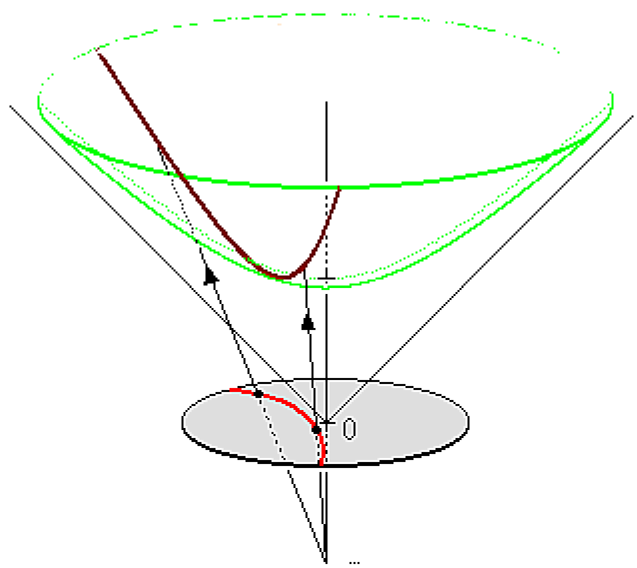

Figure 3.3: Lines in Weierstrass Model

3.4 Isomorphism

In the current algorithm, the Weierstrass Model was used to perform computations needed for transformations and the Poincaré Disk Model was used to display the results due to its simplicity. But for the algorithm to work using two different models, we need a way to map the objects from one model to the other and isomorphism is a way to do it. Isomorphism is the process of projecting objects from one model to the other preserving their structures and properties. The Weierstrass Model can be projected onto the other two models and hence it is isomorphic to both the Poincaré and Klein models [9].

3.4.1 Between Weierstrass and Poincaré Models

In Figure 3.3, the line on the hyperboloid denotes a line in the Weierstrass Model and the line on the unit circle denotes the line in the Poincaré Disk Model. These mapping can be obtained by stereographic projection of the Weierstrass Model onto the XY plane towards the point (0, 0, -1).

To map a point W [x, y, z] on the Weierstrass Model onto the Poincaré Disk Model, we use the below equation:

$$[x, y, z] \rightarrow \frac{1}{z+1}[x, y, 0]$$

To map a point P [x, y, 0] on the Poincaré Disk Model onto the Weierstrass Model, we use the below equation:

$$[x, y, 0] \rightarrow \frac{1}{(1 - x^2 - y^2)}[2x, 2y, 1 + x^2 + y^2]$$

3.4.2 Between Weierstrass and Klein Models

The objects in the Klein Model are obtained by stereographic projection of the Weierstrass Model onto the Z = 1 plane towards the point (0, 0, 0) using the equation:

For a point W [x, y, z],

$$[x, y, z] \rightarrow [\frac{x}{z}, \frac{y}{z}, 1]$$

The Klein to Weierstrass inverse projection is given by the equation:

For a point P [x, y, 0],

$$[x, y, 1] \rightarrow \frac{1}{(1 - x^2 - y^2)} [x, y, 1]$$

3.4.3 Between Poincaré and Klein Models

To map a vector *p* representing a point in the Poincaré Disk Model to a point in the Klein Model, we use the below equation:

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

For a vector *k* representing a point in the Klein Model, the inverse projection on the Poincaré Disk Model is given by the below equation:

$$p = \frac{k}{1 + \sqrt{1 - k . k}} = \frac{(1 - \sqrt{1 - k . k})\, k}{k . k}$$

Chapter – 4

Special Lines and Curves

This chapter focuses on the properties of special lines and curves that are implemented in the current program. It also includes figures for few properties to ensure better understandability.

4.1 Hyperbolic Lines

There are two types of lines in the hyperbolic plane.

1. Lines that go through the center of the bounding circle are called diameter lines which look like straight lines.
2. Open circular arcs that are orthogonal to the bounding circle.

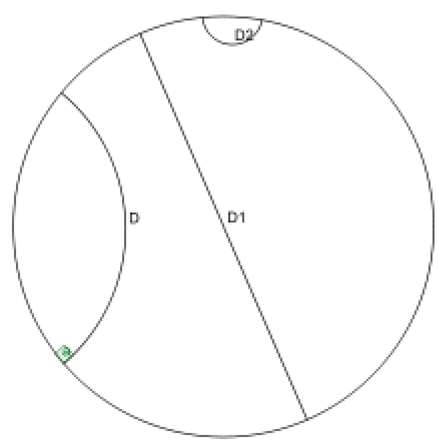

Figure 4.1: Types of hyperbolic lines

As shown in Figure 4.1, D1 is the diameter line that passes through the center of the circle and D2 and D3 are the second type of lines which are open arcs that are orthogonal to the bounding circle.

A line in the Poincaré disk becomes straight in the Euclidean sense when it passes through the center of the disk as shown in Figure 4.2.

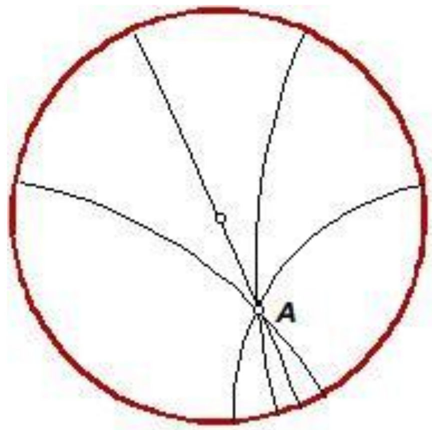

Figure 4.2: Nature of hyperbolic lines

The angle between two lines is the measure of the Euclidean angle between the tangents drawn to the lines at their points of intersection as shown in Figure 4.3.

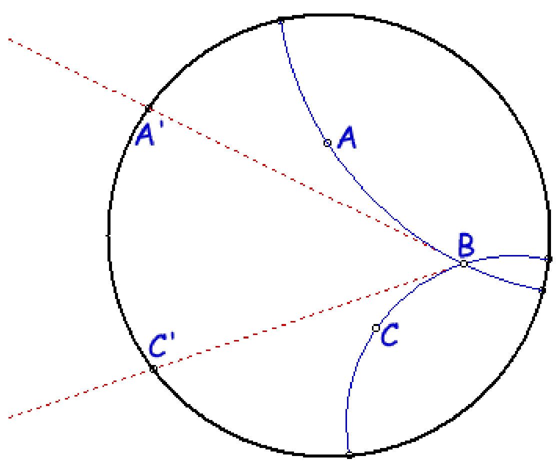

Figure 4.3: Angle between two intersecting lines in Poincaré Disk Model

Two lines on the hyperbolic plane can intersect in at most one point as shown in Figure 4.4.

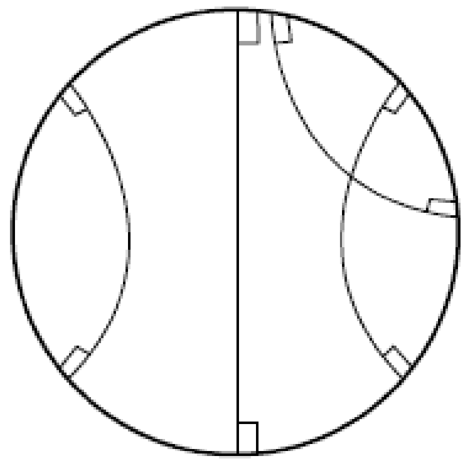

Figure 4.4: Intersection of hyperbolic lines

Given the two intersecting lines on a Euclidean plane, it can be observed that the adjacent angles are supplementary and the opposite angles are congruent. Similarly, on a hyperbolic plane, the adjacent angles on a hyperbolic line are supplementary and that the opposite angles are congruent.

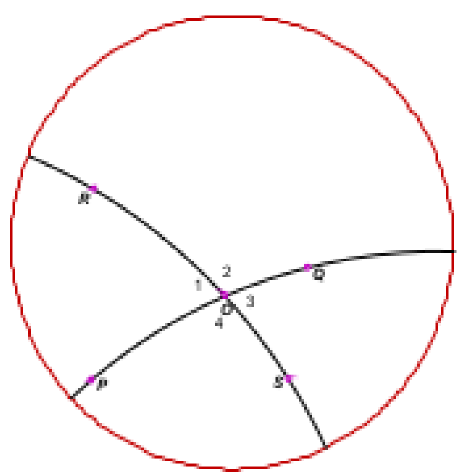

Figure 4.5: Adjacent and opposite angles

In Figure 4.5**,** angles 1, 2 (for example) are supplementary and angles 1, 3 (for example) are congruent.

If there is a line *m* on the hyperbolic plane and a point *P* not on *m*, then more than one line parallel to *m* can be drawn through *P* as shown in Figure 4.6. In the hyperbolic plane, lines are parallel if they do not have common points. Furthermore, parallel lines are not equidistant from each other as they are in Euclidean plane.

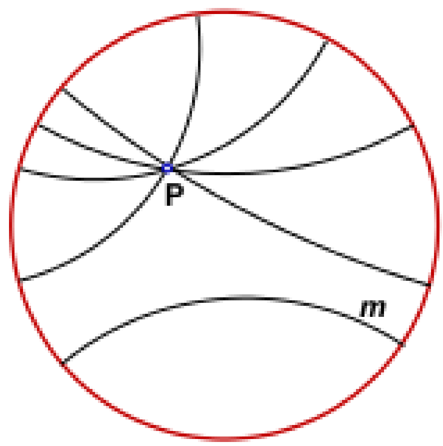

Figure 4.6: Parallel lines in hyperbolic plane

If there is a line *l* on the hyperbolic plane and a point *P* not on *l*, only one line perpendicular to *l* can be drawn from *P* to the line (similar to Euclidean plane) as shown in Figure 4.7.

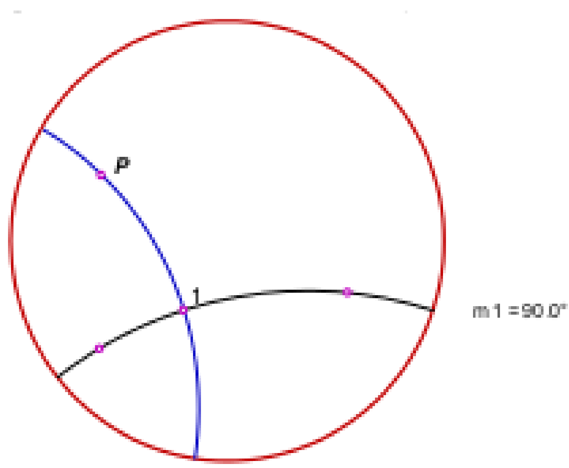

Figure 4.7: Perpendicular lines in hyperbolic plane

The lines in the hyperbolic plane are not finite in length even though they appear to be finite. Distances are distorted in the Poincaré Model, and the boundary of the disk is considered to be at infinity. As we walk towards the boundary of the disk representing infinity, we can notice that steps get progressively shorter.

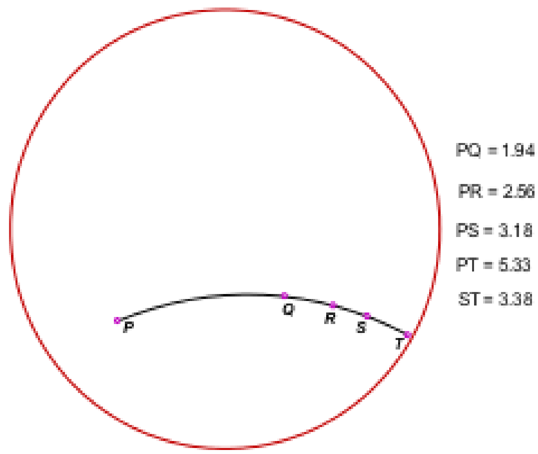

Figure 4.8: Distance Distortion in Poincaré Disk Model

As shown in Figure 4.8, the distance from P to Q is 1.94 units while the distance from S to T is 3.38 units, yet the distance from S to T, and also between other two consecutive points, appear to be much less than that from P to Q.

If a pair of parallel lines and a transversal are drawn on the hyperbolic plane, then the corresponding angles and the alternate angles are not congruent and the consecutive interior angles on the hyperbolic plane are not supplementary. This is because the parallel lines are not equidistant from each other in the hyperbolic plane. This is demonstrated in Figure 4.9.

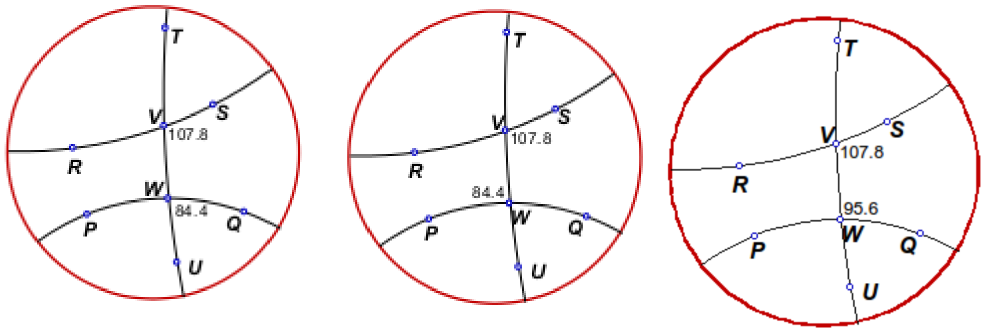

Figure 4.9: Corresponding, alternate and interior angles

If there are two parallel lines l and m on the hyperbolic plane, and at a point on m a perpendicular traversal is drawn, then the transversal for one of a pair of parallel lines does not necessarily intersect the second line as it does in the Euclidean plane [15]. When the transversal does intersect, it is not perpendicular to the second line. These two cases are shown in Figure 4.10.

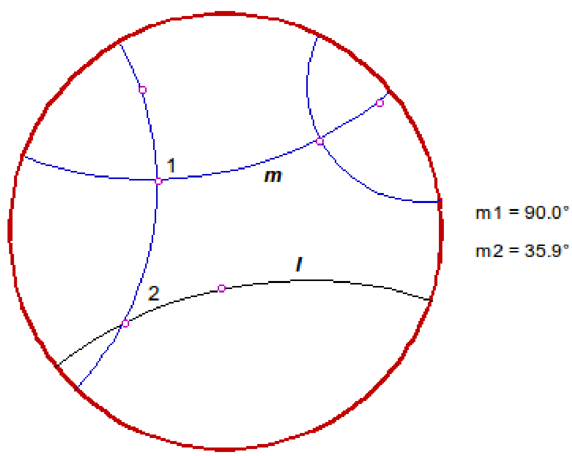

Figure 4.10: Multiple perpendicular lines

On the Euclidean plane, if two lines are parallel to the same line, then they are always parallel to each other. However, on the hyperbolic plane, it is possible for two lines parallel to a third line to be parallel and also to be non-parallel. This is shown in Figure 4.11.

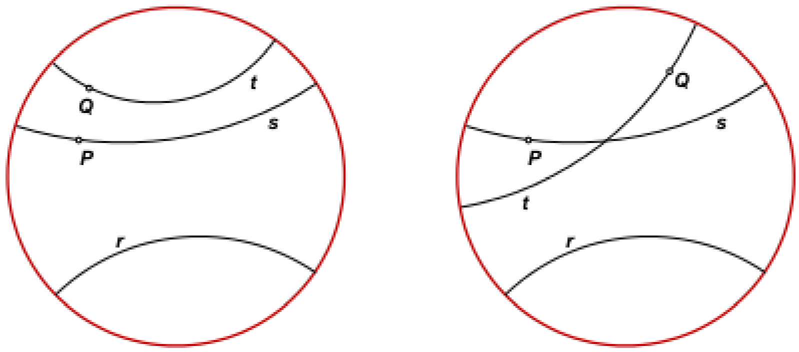

Figure 4.11: Multiple parallel lines

If two lines are perpendicular to the same line on the Euclidean plane, then the two perpendicular lines are parallel. This holds true for the hyperbolic plane too as shown in Figure 4.12.

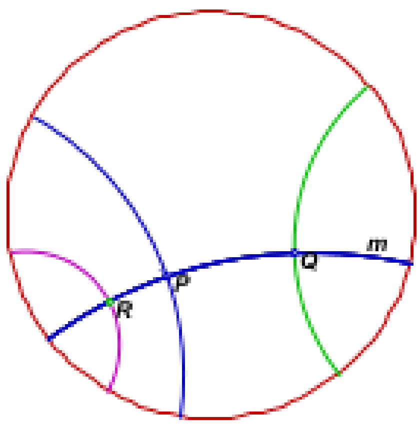

Figure 4.12: Multiple perpendiculars to a line are parallel

4.2 Equidistant Curves

In hyperbolic geometry, an equidistant curve is a curve whose points have the same orthogonal distance from a given hyperbolic line. It is also called a hypercycle or a hypercircle [14]. For example, an equidistant curve of a circle is a circle. In Lobachevskian geometry, an equidistant curve of a straight line is the locus of points at a given distance from the line with that line as a base. In Euclidean geometry, an equidistant curve of a straight line is a straight line.

Given a straight line L and a point P not on L, we can construct an equidistant curve by considering all points Q on the same side of L as P, so that the perpendicular distance from Q to L is same as that of the perpendicular distance from P to L.

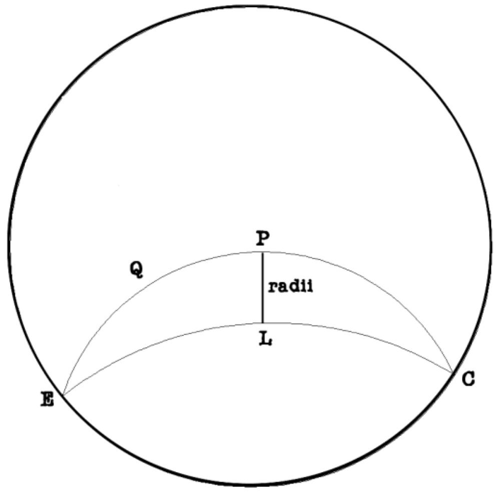

Figure 4.13: Equidistant Curve

EQPC is called the hypercycle or the equidistant curve. The line L is called the axis, or base line, and the common length of the perpendicular or orthogonal segments is called the distance. The perpendicular segments defining the equidistant curve are called its radii. We can also observe that in a plane, given a line and a point not on it, there is only one equidistant curve passing through the point of that given base line.

The following are the properties of equidistant curves that are analogous to the properties of regular Euclidean circles:

1. Circles with equal radii are congruent, those with unequal radii are not. Similarly, equidistant curves with equal distances are congruent, those with unequal distances are not.
2. A line cannot cut an equidistant curve in more than two points.
3. If a line cuts an equidistant curve in one point, it will cut the curve in a second point unless it is tangent to the curve or parallel to its base line.

4. A tangent line to an equidistant curve at a point P is defined as the line perpendicular to the radius at that point. Since the tangent line and the base line have a common perpendicular, they must be hyperparallel. This orthogonal segment is the shortest distance between the two lines. Thus, each point on the tangent line must be at a greater perpendicular distance from the axis than the corresponding point on the equidistant curve. Thus, the tangent line can intersect the equidistant curve in only one point.
5. A line perpendicular to a chord of an equidistant curve at its midpoint is a radius and it bisects the arc subtended by the chord.
6. Two circles intersect in at most two points. Similarly, two equidistant curves intersect in at most two points.
7. No three points of an equidistant curve are collinear.
8. The axis and distance of an equidistant curve are uniquely determined.

In the Poincaré Disk Model, equidistant curves are represented by lines and circular arcs that intersect the bounding circle in non-right angles i.e. acute or obtuse angles, while the axis intersects the boundary circle in the same points, but at right angles. Points on such arcs are at an equal hyperbolic distance from the hyperbolic line, which is the base line, with the same endpoints on the bounding circle. For any acute angle and hyperbolic line, there are two equidistant curves, one on each side of the line, making that angle with the bounding circle. Equidistant curves are the hyperbolic analogy of small circles in spherical geometry. For example, every point on a small circle of latitude is an equal distance from the equatorial great circle and there is another small circle in the opposite hemisphere the same distance from the equator.

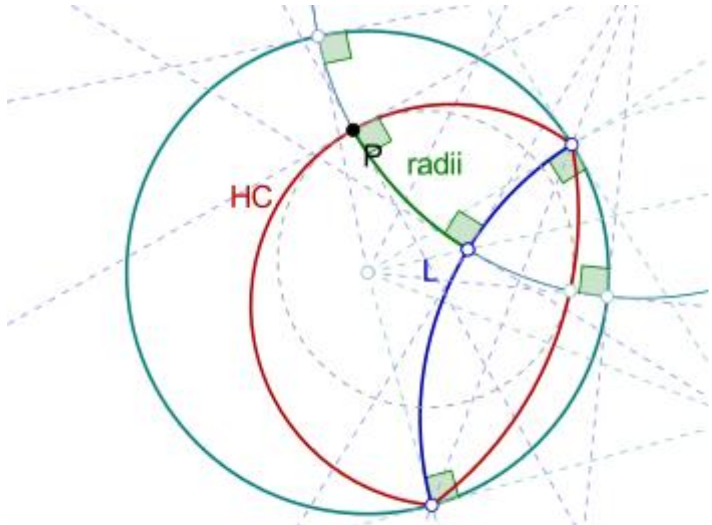

Figure 4.14: A Poincaré disk to demonstrate the equidistant curve

In Figure 4.14, HC is the hypercycle or the equidistant curve. The line L is the axis or the base line. The common length of the orthogonal segments is the distance. The orthogonal segments defining the equidistant curve are its radii.

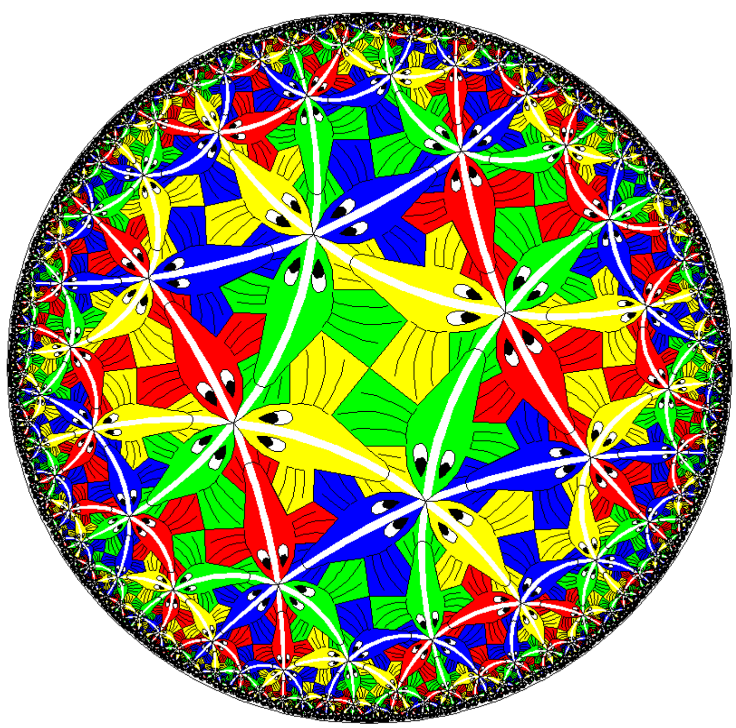

Figure 4.15: Escher's Circle Limit III Pattern

In Escher's Circle Limit III pattern as shown in Figure 4.15, the white backbones of each stream of fish make prominent arcs on the print and are usually falsely assumed that these arcs are hyperbolic lines i.e. circular arcs perpendicular to the bounding circle. However, careful measurements of Circle Limit III show that all the white arcs make angles of approximately 80 degrees with the bounding circle. This is correct, since the backbone arcs are not hyperbolic lines, but equidistant curves, each point of which is an equal hyperbolic distance from a hyperbolic base line. Each of the backbone arcs in *Circle Limit III* makes the same angle ***A*** with the bounding circle. Coxeter used hyperbolic trigonometry to show that ***A*** is given by the following expression:

$$\cos(A) = \sqrt{\frac{3\sqrt{2}-4}{8}}$$

The value of ***A*** is about 79.97 degrees, which Escher accurately constructed to high precision.

Chapter – 5

Hyperbolic Patterns

In the previous chapters, we have discussed about the hyperbolic geometry, models representing hyperbolic geometry and various shapes that can be drawn using those models. We have also seen how different shapes look in the Poincaré Disk Model. This chapter introduces us to a few terms such as *Tessellations* and *Hyperbolic Patterns.* The following sections discuss these terms in detail. We later discuss how these patterns are generated using a replication algorithm that was implemented in this program.

5.1 Tessellations

A *tessellation* is defined as a repeating pattern that can be formed by translating and transforming congruent copies of a basic sub-pattern. A sub-pattern is usually composed of one or more geometric shapes and is often called as tile. Thus, a tessellation is a repeated tiling of a surface (Euclidean or hyperbolic) with no overlaps and gaps. There are two main types of tessellations which are listed below:

1. Regular Tessellations: Regular tessellations are formed by repeating a congruent regular polygon in all directions along the plane. In a regular polygon, all the sides and angles are equivalent. On a Euclidean plane, only three kinds of regular tessellations are possible (composed using a triangle, square or a hexagon). Figure 5.1 shows these regular tessellations on a Euclidean plane.

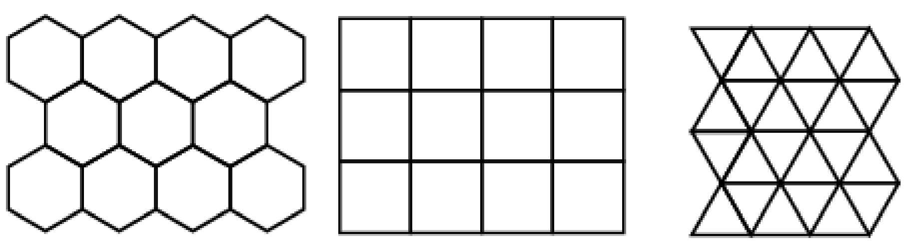

Figure 5.1**:** Regular tessellations on a Euclidean plane

2. Semi - Regular Tessellations: Semi - Regular tessellations are formed by repeating more than one regular polygon. Interestingly, in these tessellations, the arrangement of polygons at every vertex point is identical. Figure 5.2 shows these tessellations on a Euclidean plane.

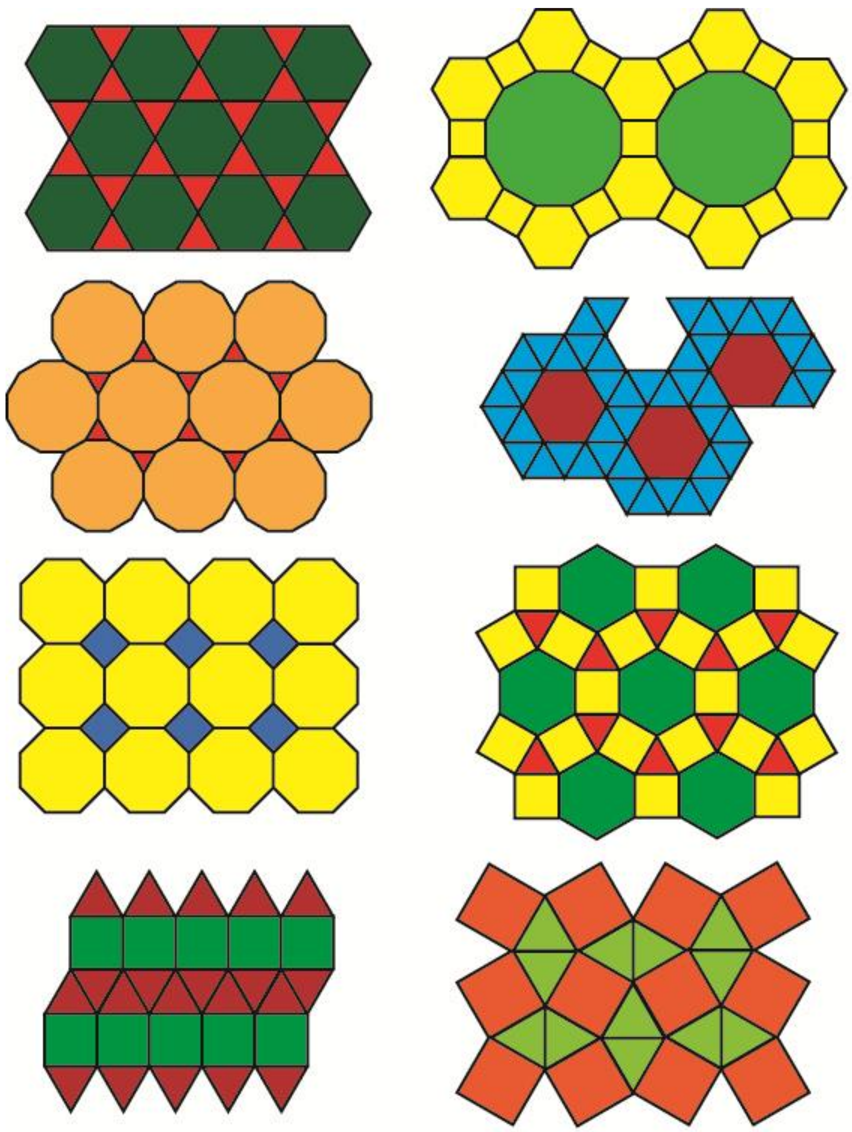

Figure 5.2: Semi - Regular tessellations on a Euclidean plane

5.2 Repeating Hyperbolic Patterns

A repeating pattern as discussed in the above sections is formed by replicating a sub pattern which is called as *motif*. These repeating patterns can also be generated on a Hyperbolic plane by replicating the hyperbolically congruent copies of motif. The current thesis is focussed on generating repeating hyperbolic patterns based on regular tessellations. From this point, the terms *repeating hyperbolic patterns* and *repeating patterns* will be used interchangeably. Figure 5.3 shows a sample regular hyperbolic pattern.

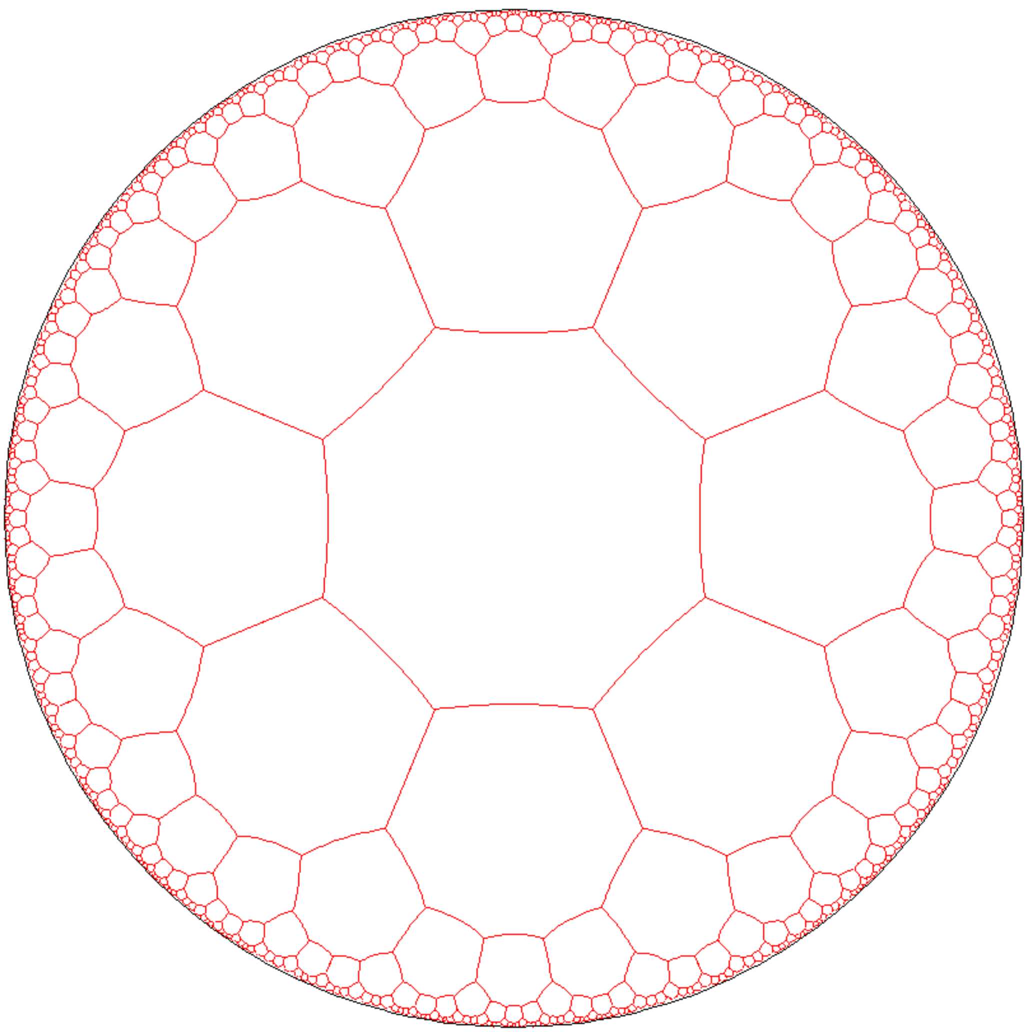

Figure 5.3**:** The regular tessellation {8, 3} on a hyperbolic plane

Regular tessellations are denoted using the notation {p, q}, where p denotes a regular p - sided polygon which will meet q congruent copies at each vertex. The Figure 5.3 denotes an {8, 3} regular tessellation.

For a tessellation {p, q} to be in the hyperbolic plane, the condition $(p - 2)(q - 2) > 4$ must be satisfied. When $(p - 2)(q - 2) = 4$, it denotes a Euclidean tessellation and when $(p - 2)(q - 2) < 4$, it is a spherical tessellation. Figure 5.4 denotes a sample repeating hyperbolic pattern.

Figure 5.4: A computer generated version of Escher's Circle Limit II pattern based on the {8, 3} tessellation

5.3 Symmetry Groups

A *symmetry operation* also called *symmetry* is as an isometry that transforms a pattern onto itself. An isometry is a hyperbolic distance preserving transformation which can either be a reflection, a translation, a glide reflection or a rotation. The *symmetry group* of a pattern is the set of all symmetries of that pattern.

The fixed lines of reflection inside a repeating pattern are called *mirrors* or *lines of symmetry* [2]. These lines of symmetry of a {p, q} tessellation divide each p-gon into 2p right angled triangles with the other two acute angles being π / p and π / q in every triangle. It is also important to note that the sum of interior angles in these 2p triangles is less than 2π because of the properties of hyperbolic geometry. The symmetry group for a tessellation {p, q} can be generated by the reflections across the sides of each triangle. The symmetry group is denoted using the notation [p, q]. Figure 5.5 shows the symmetry group [6, 4] for the tessellation {6, 4}.

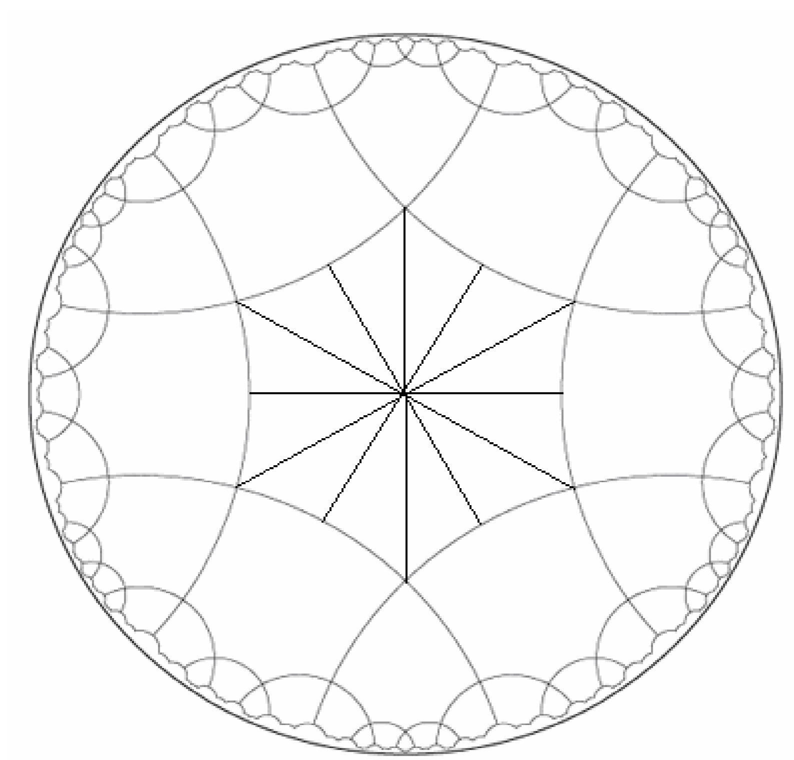

Figure 5.5: A {6, 4} tessellation with symmetry group [6, 4]

There are two subgroups of index 2, for the symmetry group [p, q]. First subgroup of index 2 is $[p, q]^+$ [12], which means that there are *twice as many* symmetries in the group as $[p, q]^+$. This subgroup can be generated in two ways. The first method is to include all symmetries from [p, q] which are generated by applying an even number of reflections from [p, q]. The second method is to apply any two of the three rotations, π, 2π / p, 2π / q, about the corresponding vertices of the right angled triangles formed by the lines of symmetry. Figure 5.6 shows a sample pattern generated using the subgroup $[5, 4]^+$.

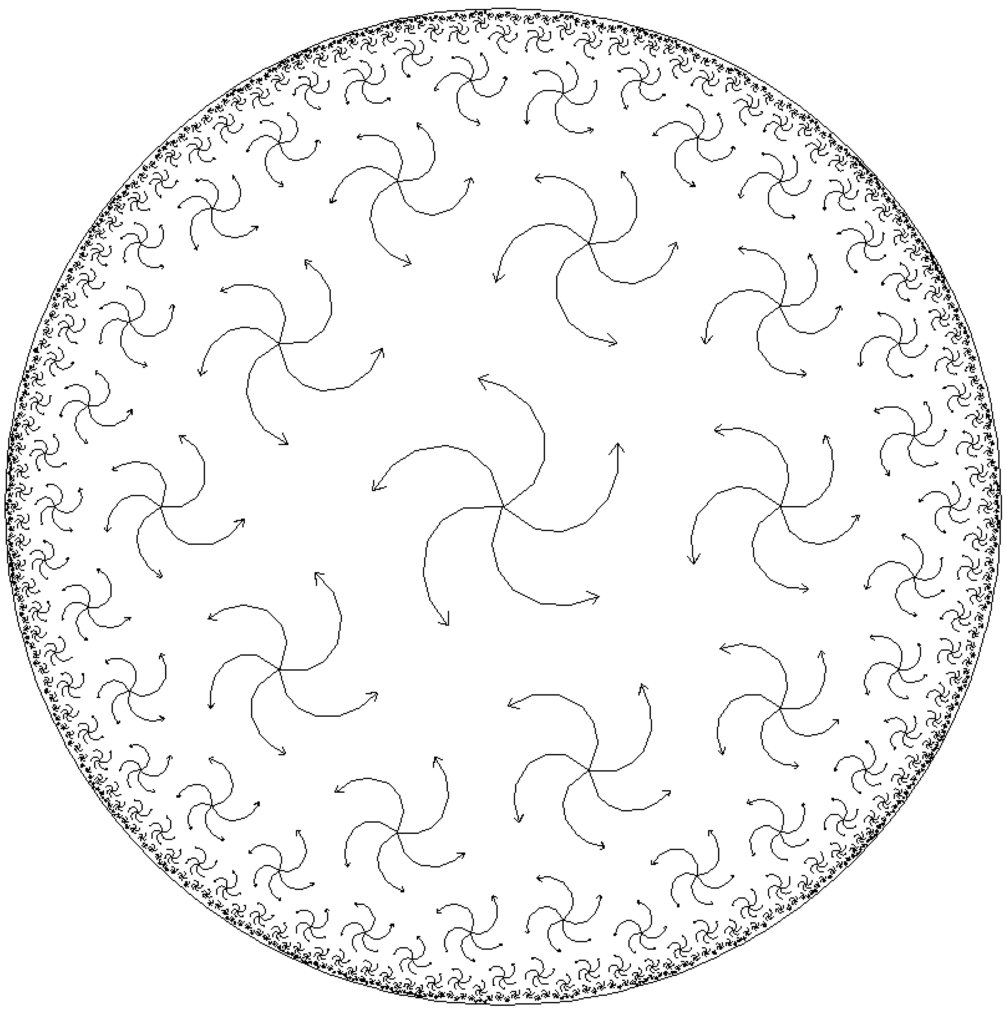

Figure 5.6: A sample pattern of symmetry subgroup $[5, 4]^+$

A second subgroup of the symmetry group [p, q] is $[p^+, q]$. This is also a subgroup of index 2 [12]. This subgroup can be generated by rotating π / p degrees about the center of p-gon and a reflection in one side of that p-gon of tessellation {p, q}. Figure 5.7 shows a sample pattern generated using the subgroup $[5^+, 4]$. We can observe that the arrows in the pattern (Figure 5.7) have both clockwise and anti-clockwise directions unlike the pattern (Figure 5.6) which has all the arrows in the anti-clockwise direction.

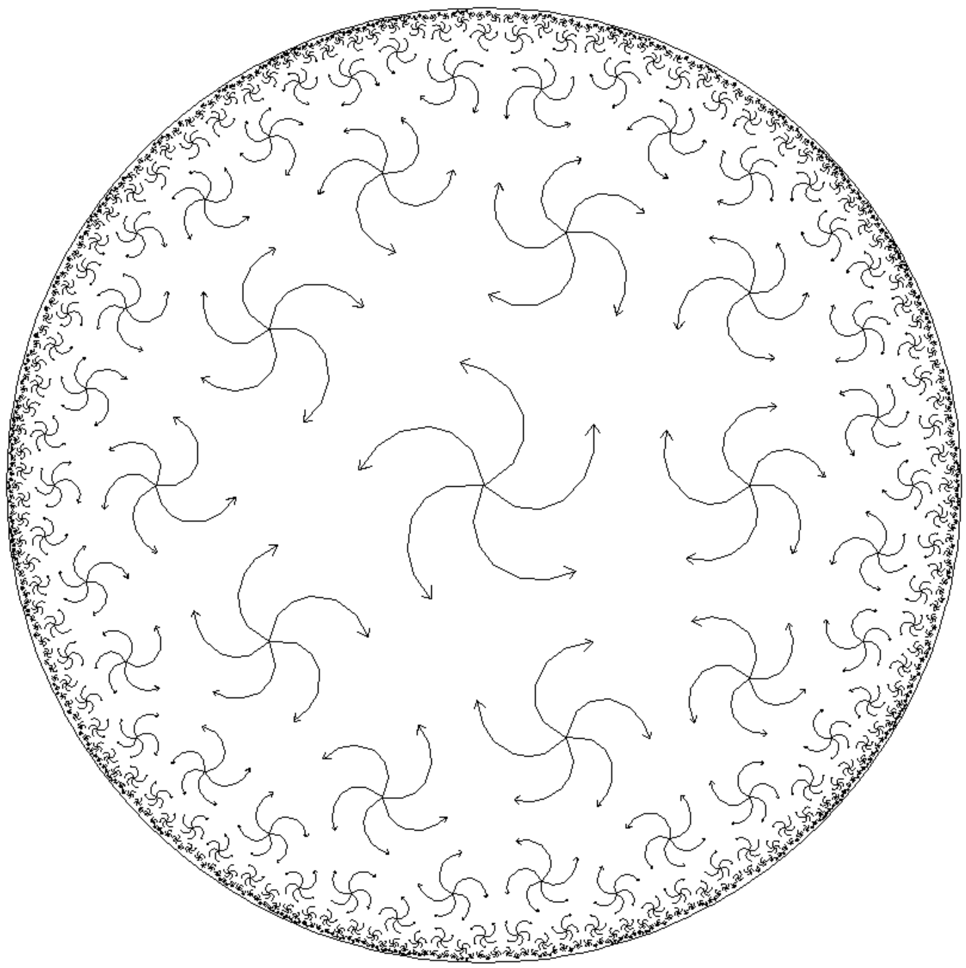

Figure 5.7: A sample pattern of symmetry subgroup $[5^+, 4]$

5.4 Motif and Fundamental Region

Motif and *Fundamental Region* are the two most important terms in this chapter. A *Motif* is the basic sub-pattern which is used to generate the repeating pattern. As defined by Dunham [12], if the hyperbolic plane is covered by the transformed copies of a connected set under elements of a symmetry group without any overlapping, that set is called a *fundamental region* for the symmetry group. The resulting repeating hyperbolic pattern will be interlocking, if the motif covers the entire fundamental region. Figure 5.8 shows a sample pattern that is interlocking.

Figure 5.8: A sample interlocking pattern with the fundamental region shown in dark boundaries

5.5 Repeating Pattern Generation Algorithm

This section provides a brief overview of the repeating hyperbolic pattern generation algorithm proposed by Dunham [3] as a part of his research work. This algorithm was previously implemented in Java language (using Graphics2D framework) by Vejendla [8]. Our current program is an extension to the program written by Vejendla [8].

The process of replicating the motif to generate the entire repeating pattern involves two steps. The first step in the process is to replicate the fundamental region and create the *central p-gon.* This central p-gon is filled by rotating the motif around the p-gon center and/or reflecting it across the diameters and perpendicular bisectors of the edges until it is entirely filled. This central p-gon forms the first layer of the repeating pattern. Figures 5.9 (before filling) and 5.10 **(**after filling**)** shows the creation of central p-gon using the motif.

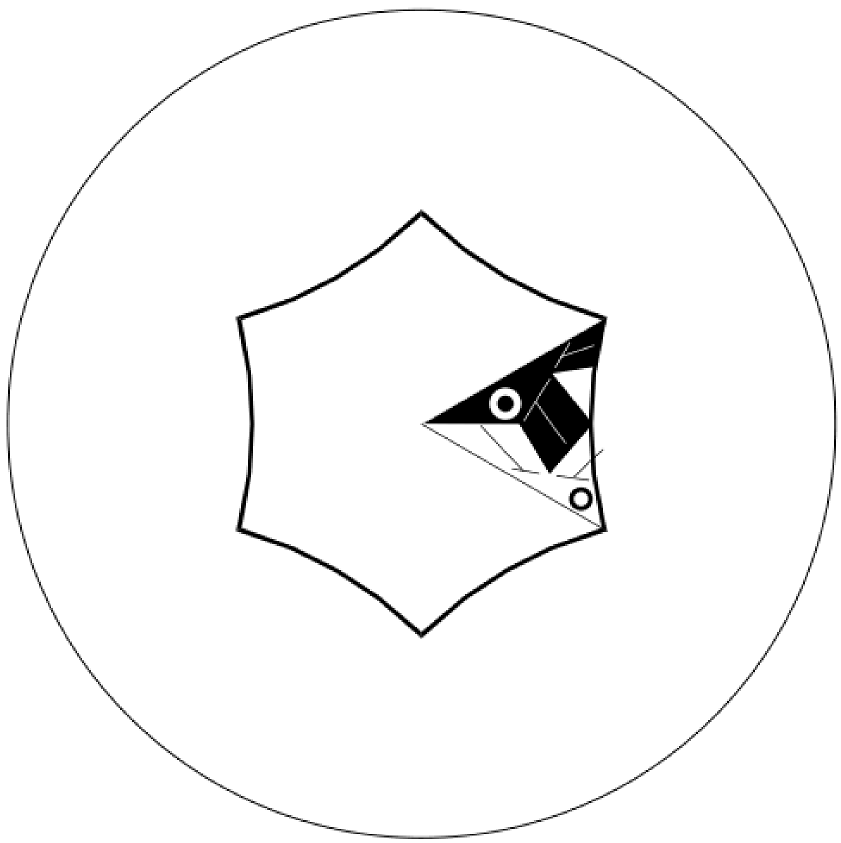

Figure 5.9: Pattern with the central p-gon before replicating motif

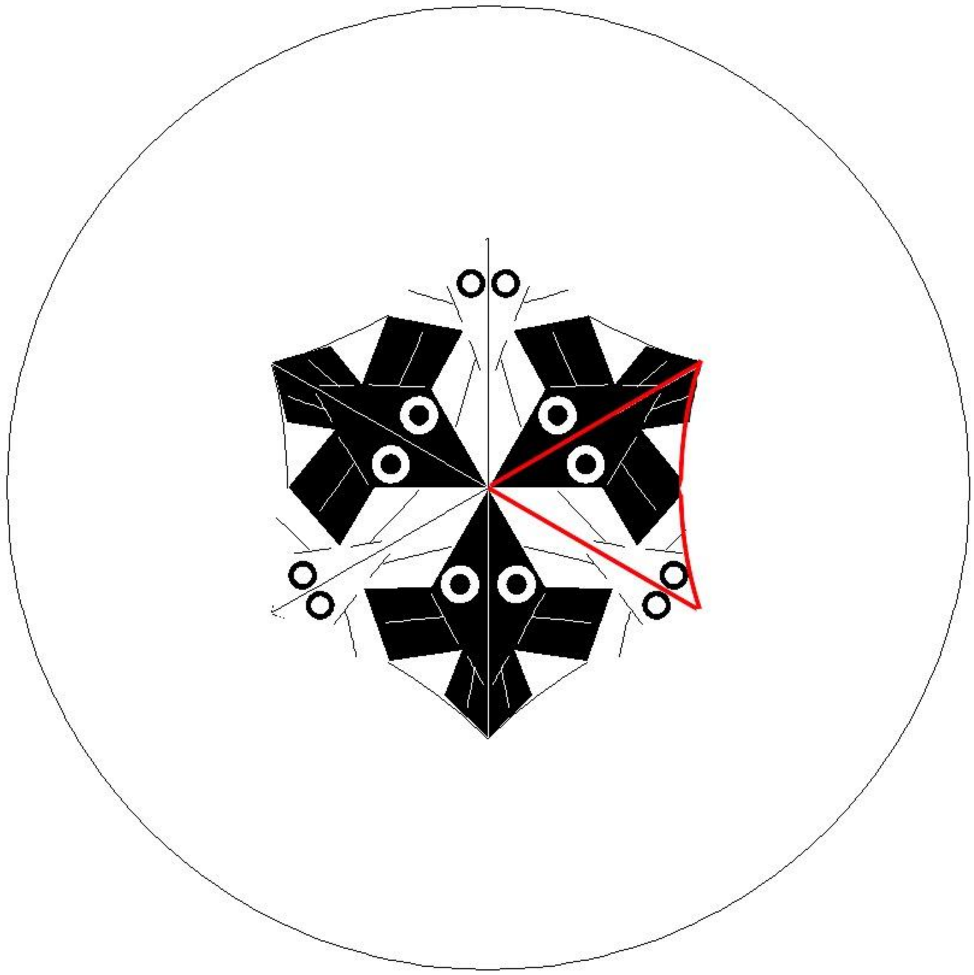

Figure 5.10: Pattern with the central p-gon after replicating the motif (motif is shown in red boundaries)

The second step is to replicate the p-gon pattern to generate the entire repeating pattern in layers. It is more efficient to replicate the entire p-gon than replicating the motif because the number of transformations can be greatly reduced and the transformations are less prone to the roundoff errors. The layers are generated recursively. The first layer in the pattern is generated using the first step of the algorithm and the *k + 1* layer consists of all the p-gons sharing an edge or vertex with the *k*-layer. Figure 5.11 shows the layer generation up to 4 layers. The p-gon 1 is rotated about vertex A to draw p-gon 2, which is rotated about vertex B to draw p-gons 3, which is again rotated about vertex C to draw p-gons 4.

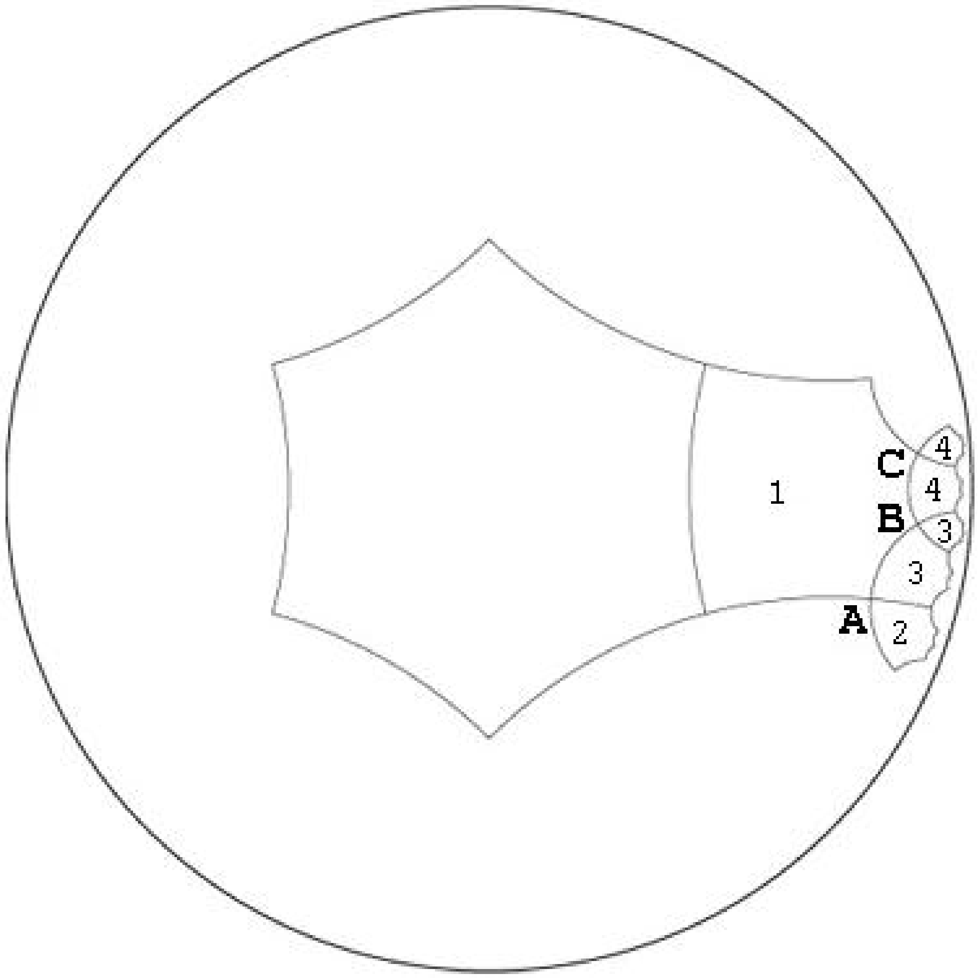

Figure 5.11: Recursive generation of pattern from layer 2 to layer 3

All the points of the fundamental region undergo symmetry operations to transform the p-gon from one layer to the next layer. Every point is projected onto the Weierstrass Model by using the inverse projection formula (in section 3.4.1). By taking the product of the vector representing the coordinates with the Lorenz Matrix representing the symmetry operation, the point is then transformed to a new location. The new point is then projected back to the Poincaré Model.

5.6 Implementation of Replication Algorithm

This section describes the second step of creating the repeating patterns i.e., the recursive replication algorithm in more detail. To generate the $(k + 1)^{st}$ layer from the k^{th} *layer*, the algorithm iterates over each vertex of the p-gon in the k^{th} layer which it shares with $(k + 1)^{st}$ layer. For each vertex, the algorithm calculates the number of polygons needed to draw the next layer from that vertex. Then the algorithm calls itself recursively for the vertices of the p-gons in the newly formed layer. As shown in Figure 5.11, after p-gon 1 is drawn, the replication algorithm is called for each of its vertices A, B and C. The process continues by calling the algorithm for all the exposed vertices of the new p-gons.

5.7 Implementation of Hyperbolic Lines and Equidistant Curves

As mentioned earlier, a hyperbolic line can either be a diameter or an open circular arc in the Poincaré Disk Model which intersects the bounding circle orthogonally. It passes through two points clicked by the user and has the end points on the bounding circle. A hyperbolic line segment is a segment of hyperbolic line inside the bounding circle with its endpoints being the points clicked by the user. It shares the same Euclidean center with the hyperbolic line.

A hyperbolic line or a line segment can be constructed using two points which the user clicks in the bounding circle. In either case, the center of the circle corresponding to the hyperbolic line is calculated first. Then, for a line segment (or arc), the angle subtended by it at the center is calculated and drawn. But, for the hyperbolic line, the two points of intersection of the bounding circle with the circle corresponding to the hyperbolic line are calculated and the angle subtended by the line is calculated to draw the actual line.

An equidistant curve is also a circular arc in the Poincaré Disk Model. However, it may not intersect the bounding circle orthogonally. This property makes the equidistant curve a special curve. It is important to note that there exists at most two equidistant curves for a hyperbolic line, one on either side of the line for a given perpendicular distance (or radius) *d.* An equidistant curve can be constructed using three points which the user clicks in the bounding circle. The first two points define a hyperbolic line and the third point defines the equidistant curve to the hyperbolic line passing through this point. Initially, the hyperbolic line is constructed using the method that is mentioned above. Then, the center of the circle corresponding to the equidistant curve is calculated using three points (two points of intersection and the third point from the user). The angle subtended by the equidistant curve at the equidistant curve's center is calculated to draw the actual curve.

Chapter – 6

Graphical User Interface

This chapter is focussed on the graphical user interface of the program which has multiple functionalities such as creating new patterns, opening existing patterns, modifying them and saving them as a data file onto the disk. That format of a data file is fixed and is discussed in Appendix.

Figure 6.1 shows the user interface during the startup of the program. The main components of the user interface includes a few menu items, a drawing canvas for displaying the patterns, a notes panel for assisting the user, a color selection combo box for choosing the current color, a zoom panel for enabling a closer look at the generated pattern and a points panel for listing the points required to generate the pattern. Each component is explained in detail in the later sections in this chapter.

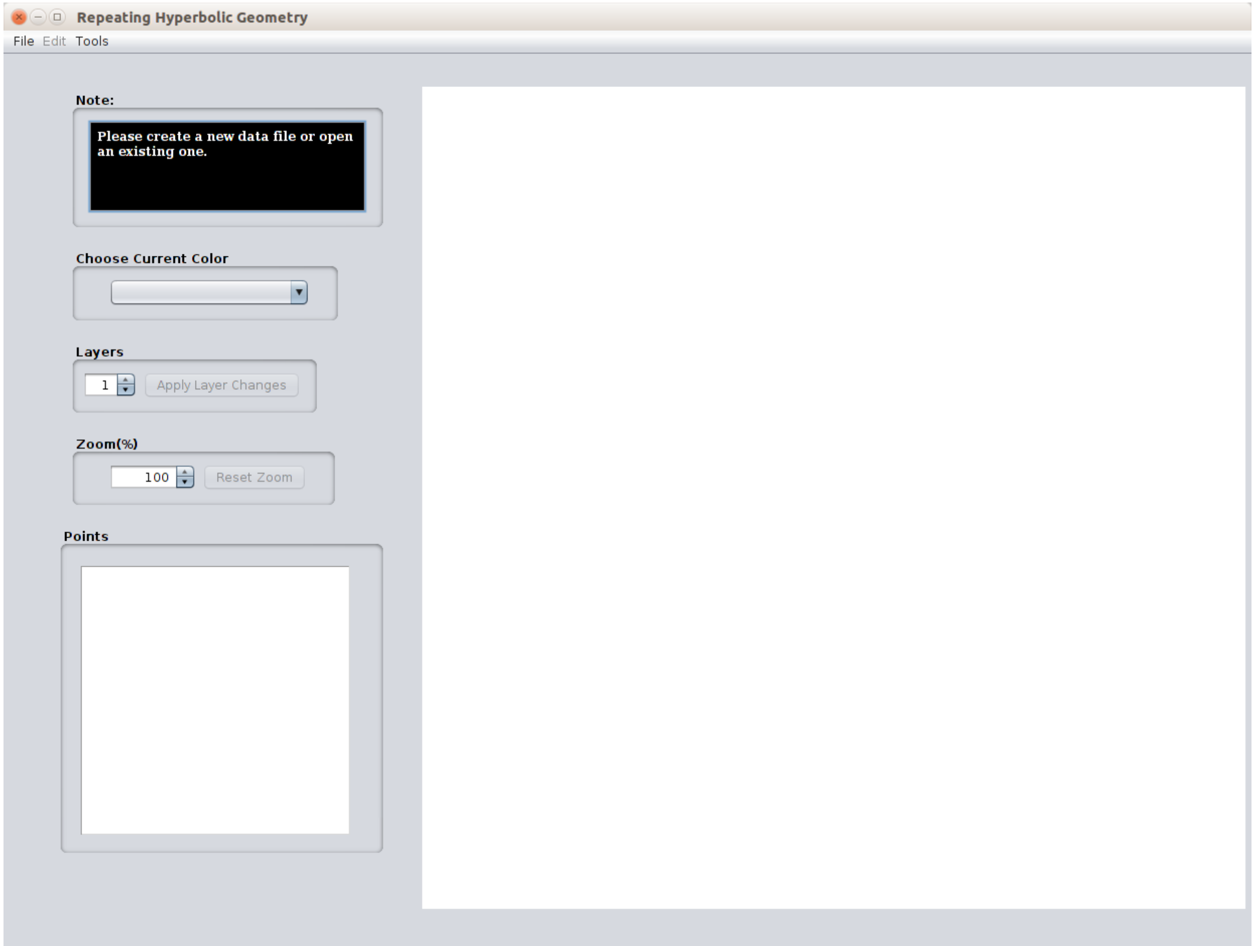

Figure 6.1: Interface during startup

6.1 File Menu

Figure 6.2 shows the user interface with the File menu opened. It has four menu items, New, Open, Save and Exit.

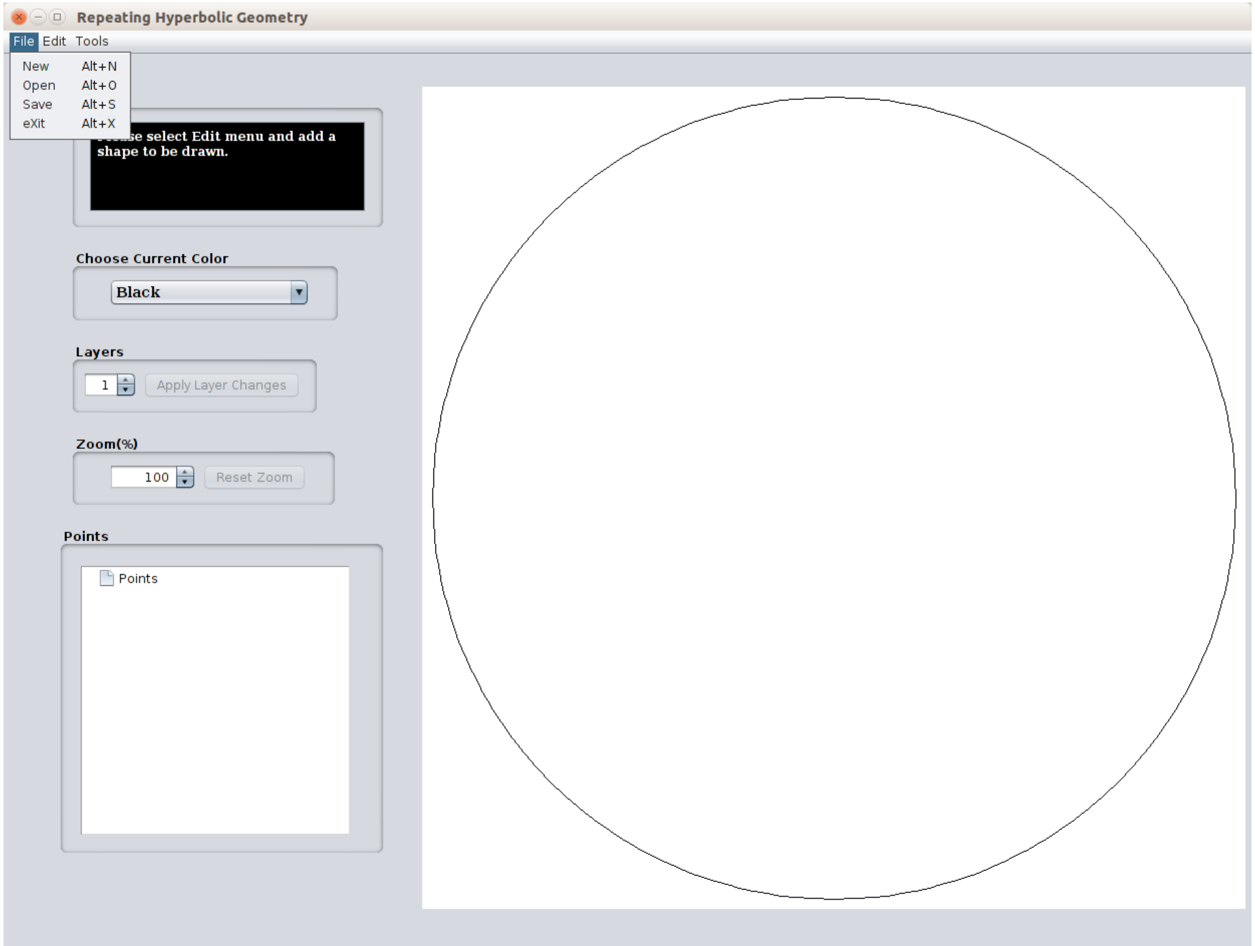

Figure 6.2: Interface when File menu is clicked

1. File -> New menu is used to create new patterns with the user adjustable settings. A pattern has various parameters which the user has to choose before he starts to generate any pattern. The parameters include:
    a. p - the number of sides the the central polygon
    b. q - the number of polygons meeting at each vertex
    c. the maximum number of colors used to generate the pattern
    d. the kind of reflection symmetry
    e. the transformation data

    Figure 6.3 shows the user interface when the New menu item is clicked. It has all the parameters mentioned above along with some sample input data. The settings in the figure generate a {6, 4} tessellation using the shapes that will be added to it.

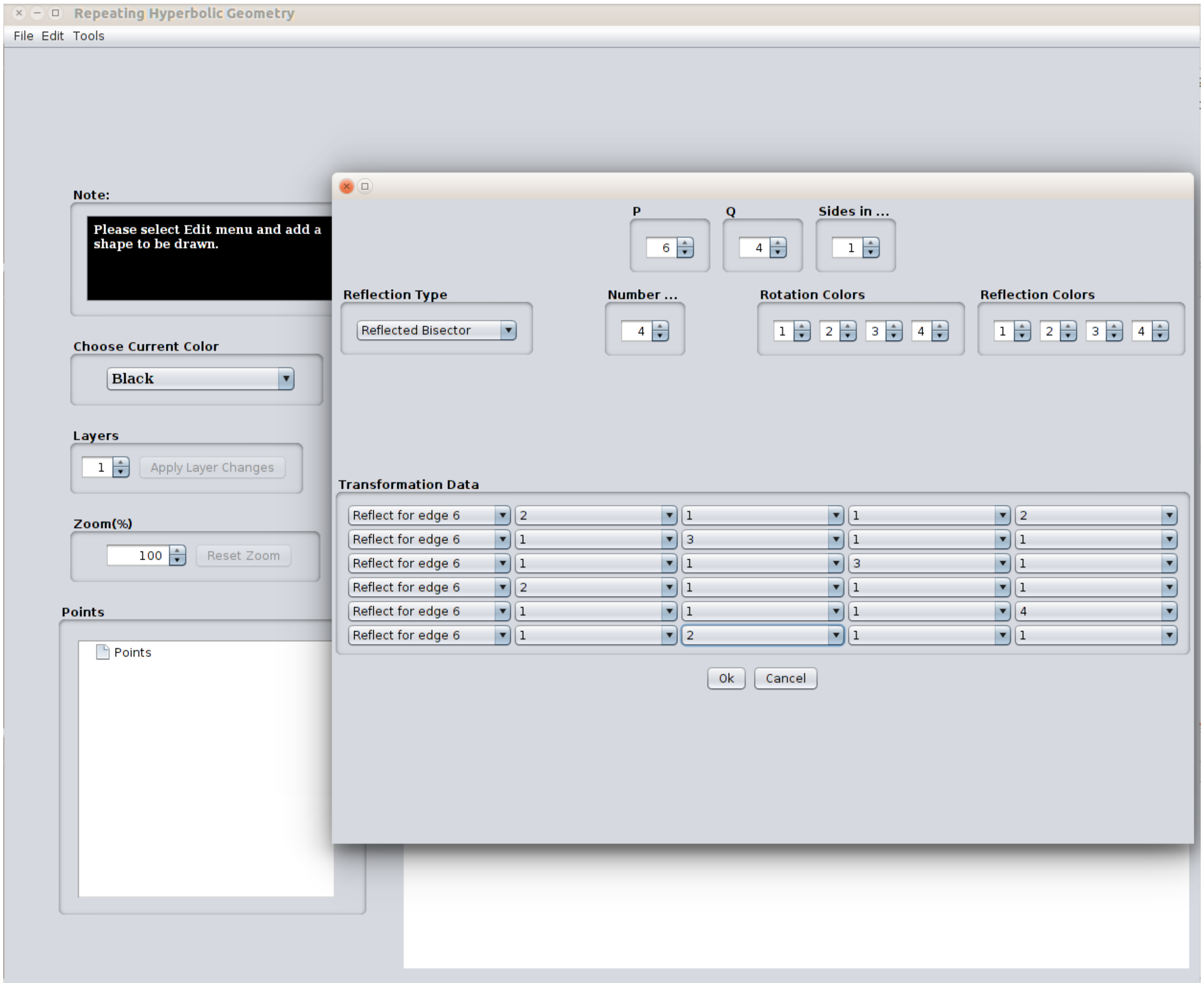

Figure 6.3: New dialogue box with user editable settings

2. Open menu is used to open existing data files on the file system. As the data file is opened, all the settings in that data file are loaded into the program automatically. An opened data file can also be modified and saved for further use. Adding shapes is to a new or existing pattern is discussed in the later sections.

   Figure 6.4 shows the user interface when the Open menu is clicked. It shows the existing data files in the file system that are ready to be opened.

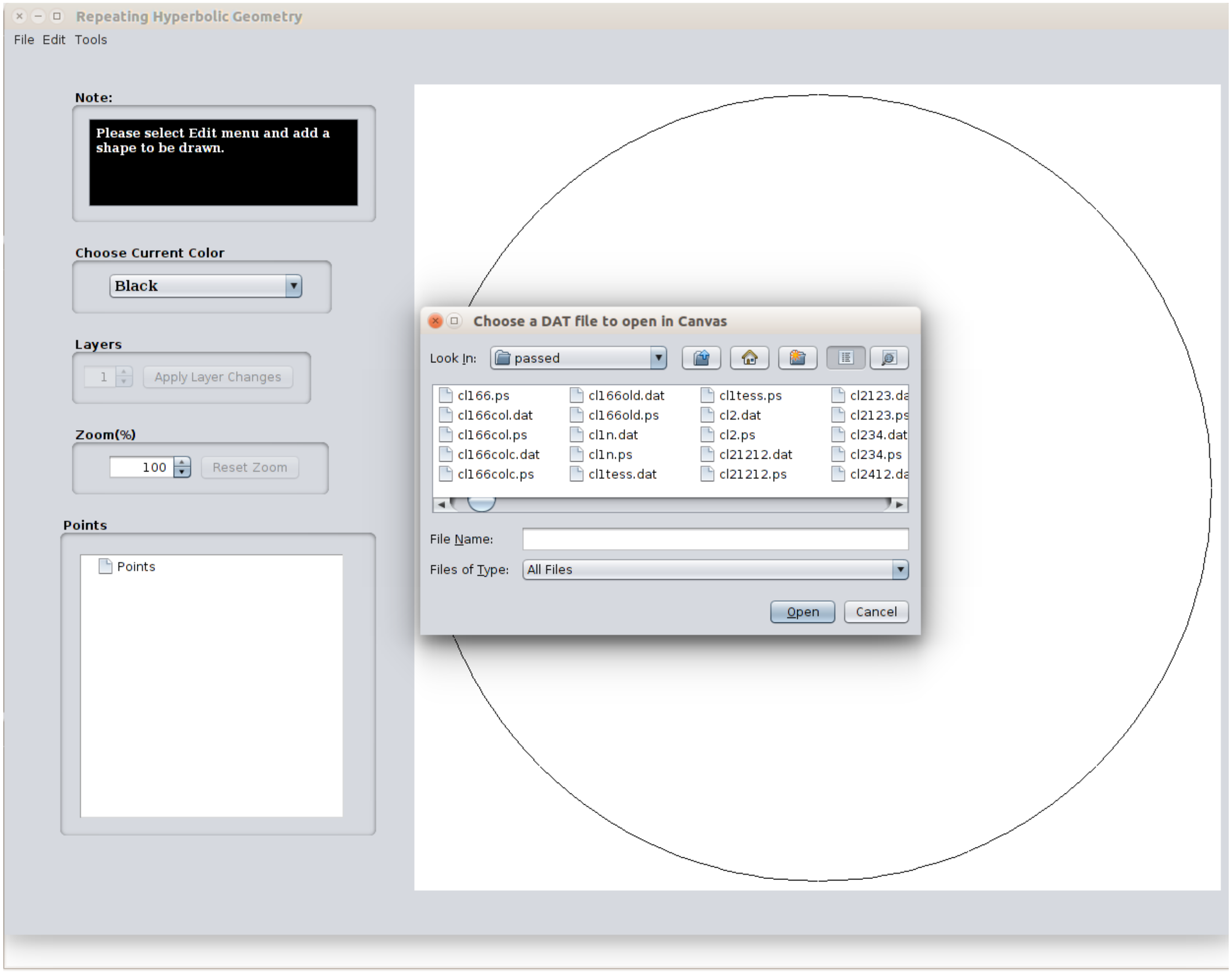

Figure 6.4: Dialogue box to open existing data files

3. Save menu is used to save a pattern on the file system. When the user chooses to save the pattern, all the data points, settings of the current pattern are stored in a data file. It checks if the same data file exists and if it does, warns the user accordingly. The user interface looks in the same way as in the Figure 6.4.
4. Exit menu is used to exit the program. Alternatively, it can also be closed by clicking the X on the window. When the user clicks the exit menu, it prompts the user to save the generated pattern.

6.2 Edit Menu

The main purpose of the edit menu is to edit a given pattern by adding new shapes or removing existing shapes from it. The figure supports a wide variety of shapes that are listed below:

1. Circle / Filled Circle
2. Filled Polygon
3. Filled Pgon
4. Polyline
5. Hyperbolic Line Segment
6. Hyperbolic Line
7. Equidistant Curve
8. Horocycle
9. Horocycle with Lines

Figure 6.5 shows the user interface when the edit menu is clicked. It has all the shapes listed above.

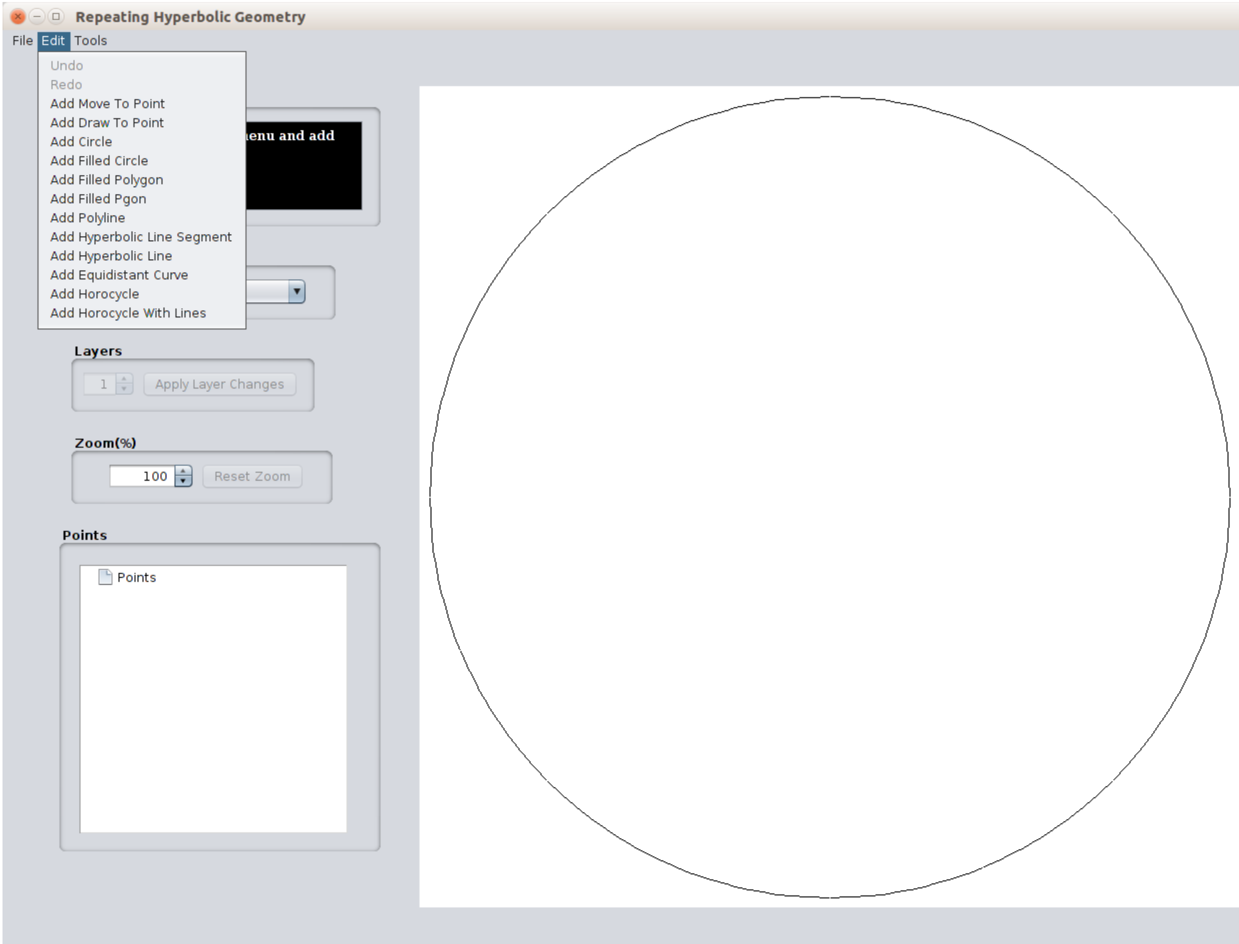

Figure 6.5: User interface when the Edit menu is clicked

In addition to these shapes, there is also an option to undo/redo a shape that helps the user to draw patterns more efficiently by eliminating the necessity to create a new file every time he wants to modify a pattern.

Few validations are done on the data files that are loaded into the program. All the points in the data file are verified to check if they properly define a particular shape. The program throws an error message when an invalid data file is loaded. The program also makes sure if the user is clicking on the valid portion of the drawing canvas. An information dialogue box is shown when the user clicks outside of the bounding circle. This scenario can be visualized in Figure 6.6.

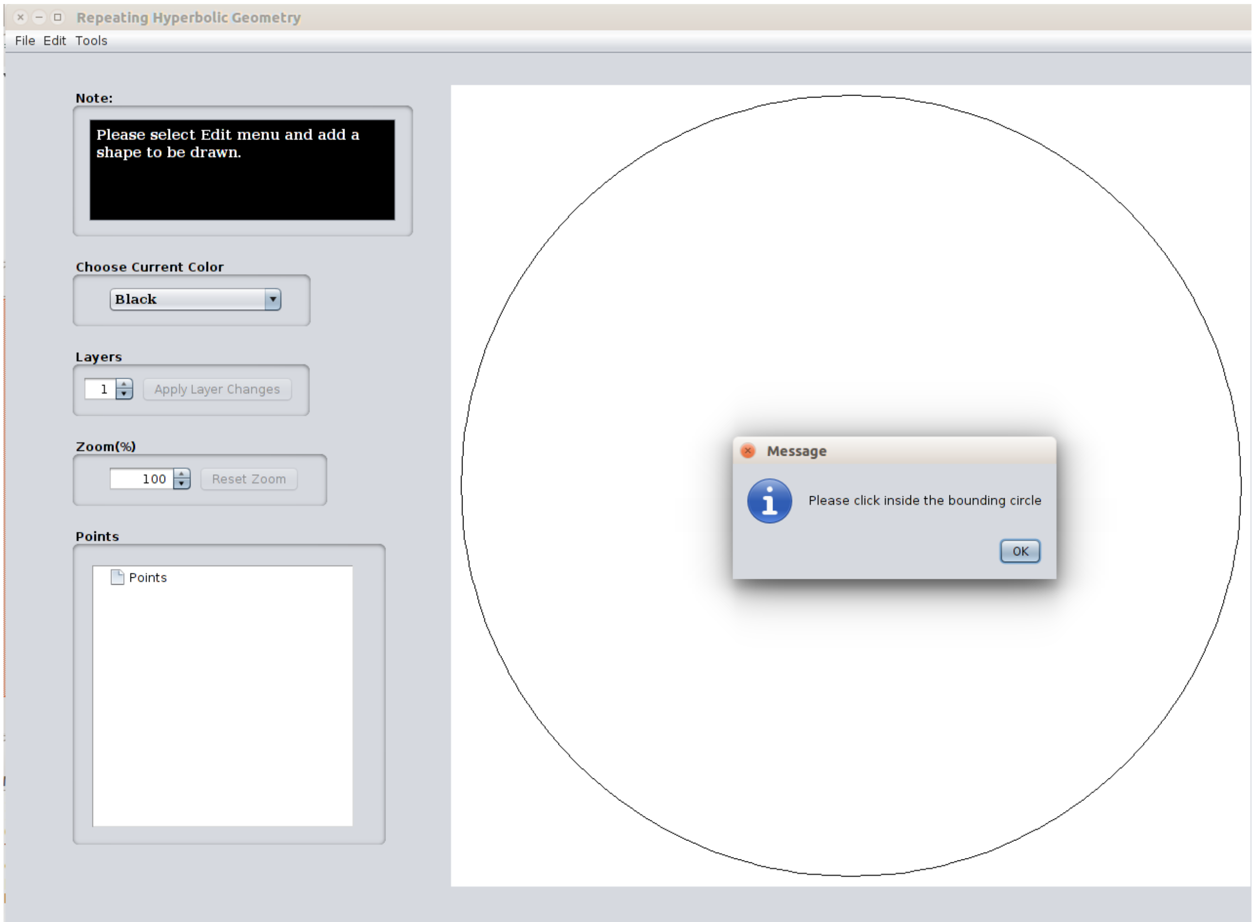

Figure 6.6: Interface showing an information message when the user clicks outside the circle

Figure 6.7 shows the process of generating a {6, 4} tessellation with Hyperbolic line segments when 1 layer is applied.

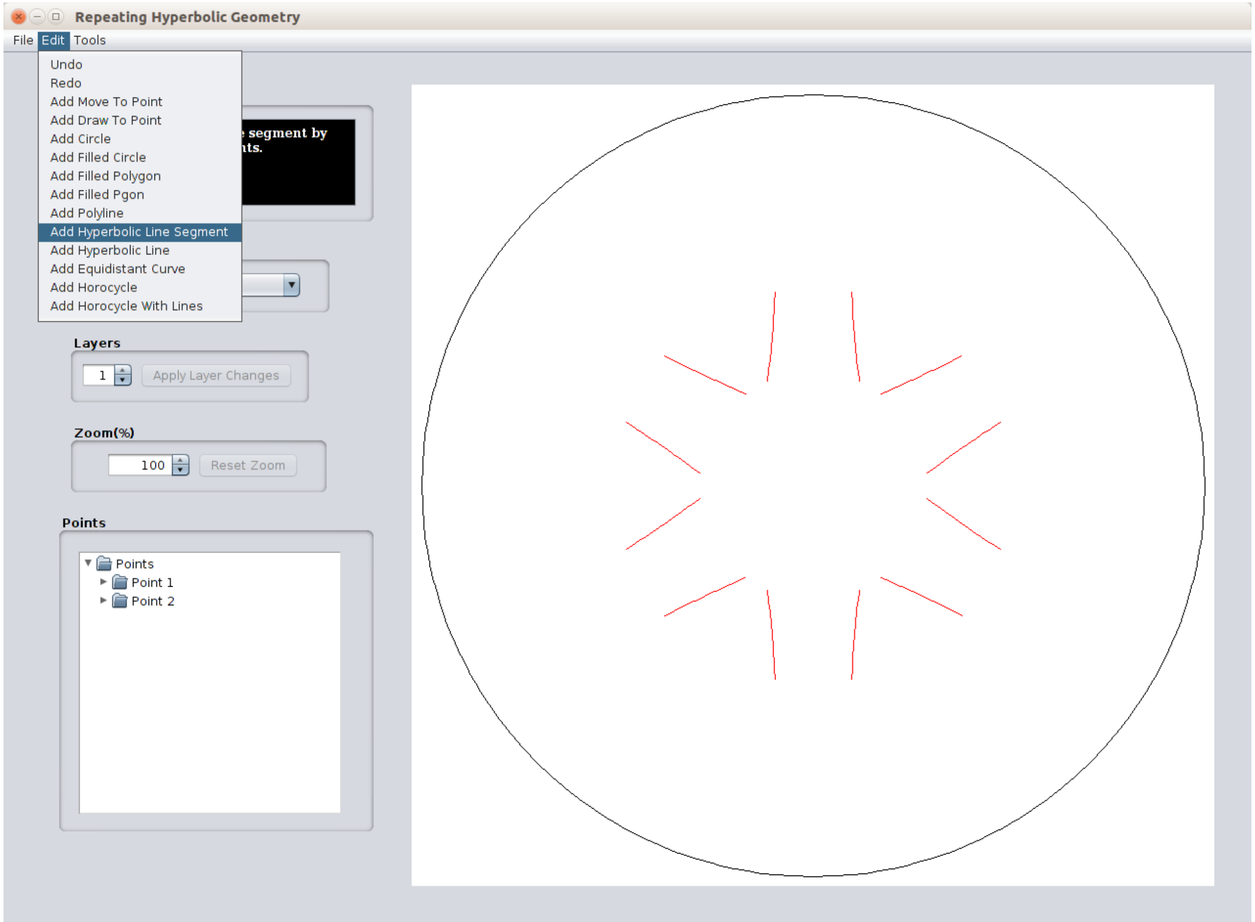

Figure 6.7: A regular {6, 4} tessellation of Hyperbolic line segments with 1 layer

Figure 6.8 shows the process of generating a {6, 4} tessellation with Hyperbolic line when 1 layer is applied.

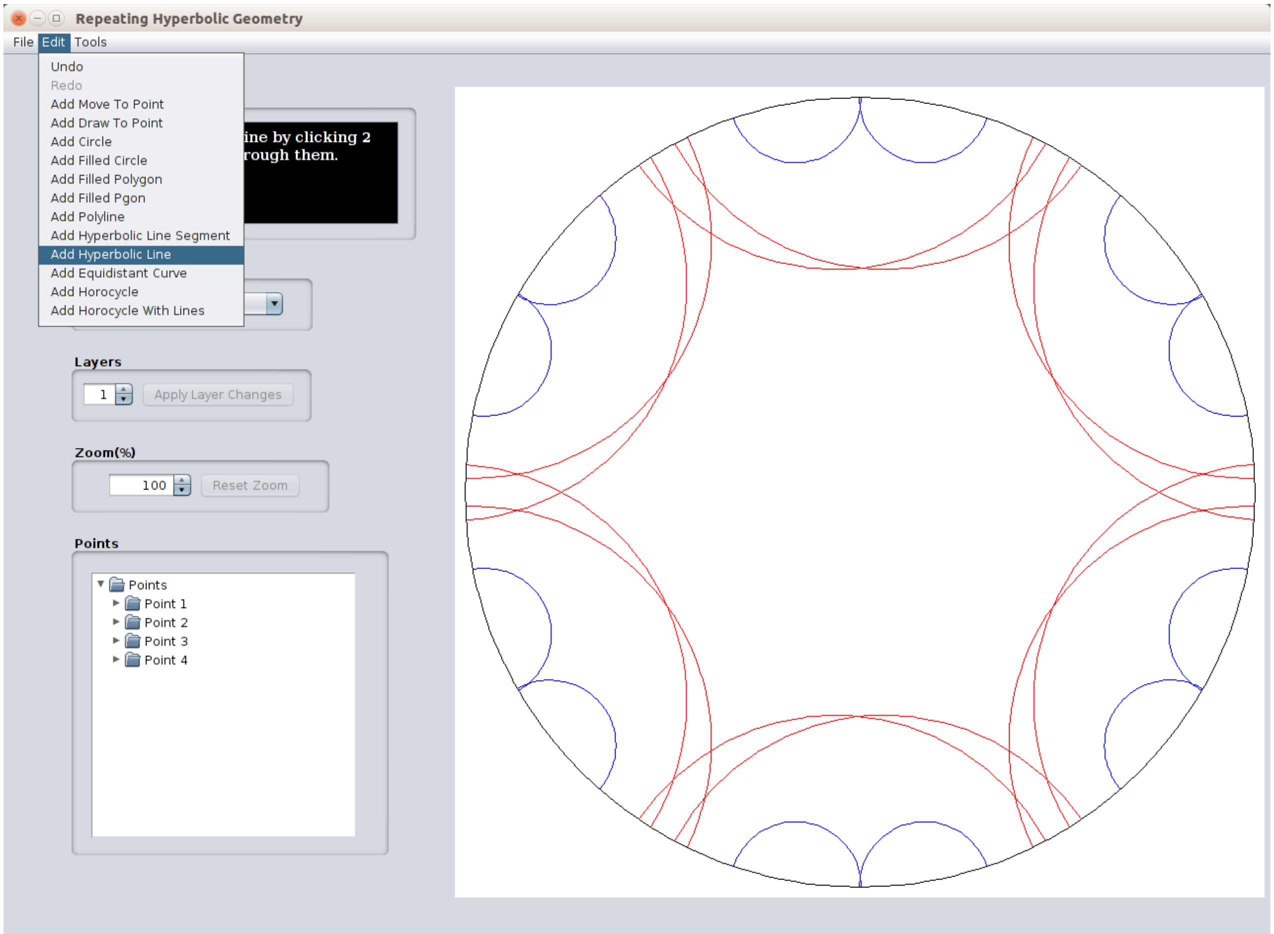

Figure 6.8: A regular {6, 4} tessellation of Hyperbolic lines with 1 layer

Figure 6.9 shows the process of generating a {6, 4} tessellation with Equidistant Curves when 1 layer is applied.

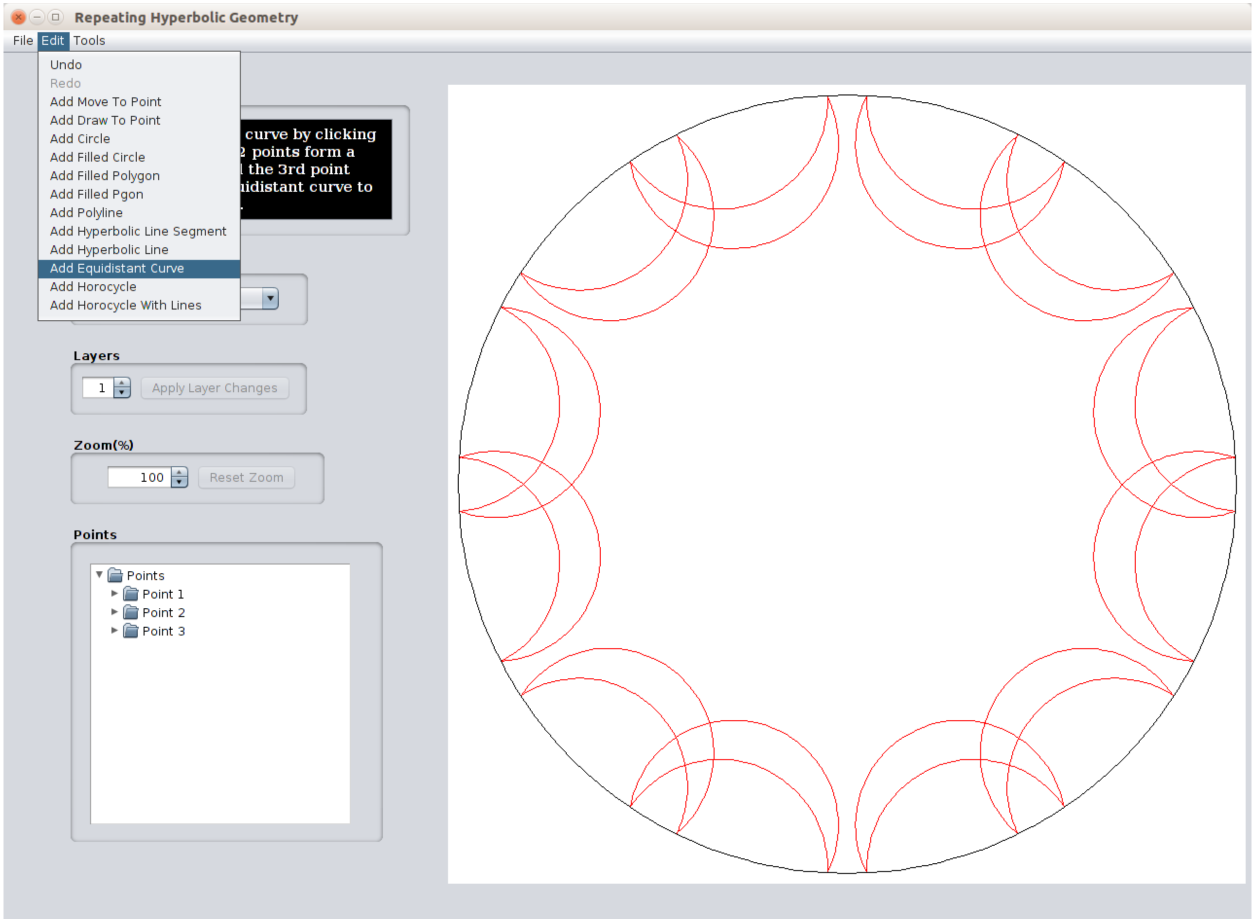

Figure 6.9: A regular {6, 4} tessellation of Equidistant Curves with 1 layer

6.3 Toolbar

The toolbar on the left side of the user interface in Figure 6.1 has a few other options that are discussed below.

1. Notes Panel - Useful information is displayed in this text area, that assists the user when creating a new pattern or when adding shapes to an existing pattern. Information such as the number of points the user has to choose to draw a particular shape is displayed in this text area. A sample {6, 4} tessellation of equidistant curves along with its notes are shown in Figures 6.10 and 6.11.

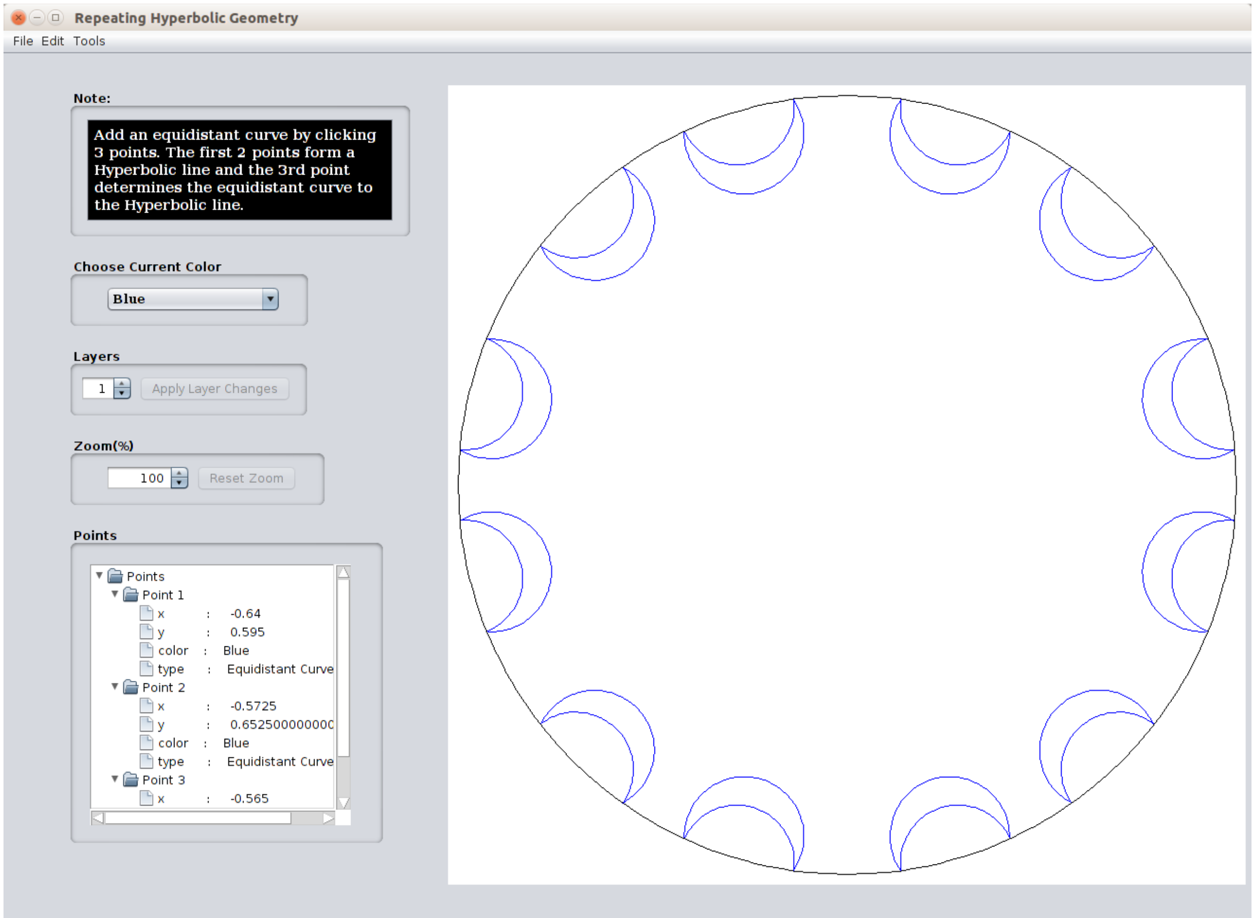

Figure 6.10: A regular {6, 4} tessellation of equidistant curves with 1 layer

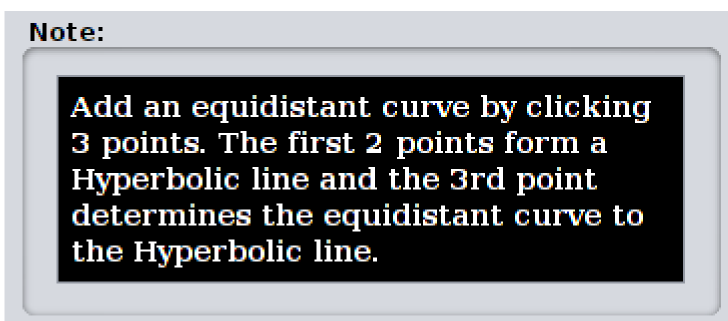

Figure 6.11: A sample note for assisting the user to draw an equidistant curve

2. Choose Current Color Panel - The combo box in this panel has a list of all colors that are supported for a particular data file. The number of colors displayed

depends on the maximum number of colors parameter supplied by the user while creating a new pattern. The combo box can be seen in Figure 6.1.

3. Layers Panel - The value of spinner in this panel denotes the number of layers, the central p-gon has to be replicated for generating the pattern. More layers produce a much thicker pattern towards the edge of the bounding circle. Layer changes can be applied by clicking on the Apply Layer Changes button. The effect of the number of layers can be seen in the Figures 6.12 (2 layers, less thick) and 6.13 (5 layers, much thicker)

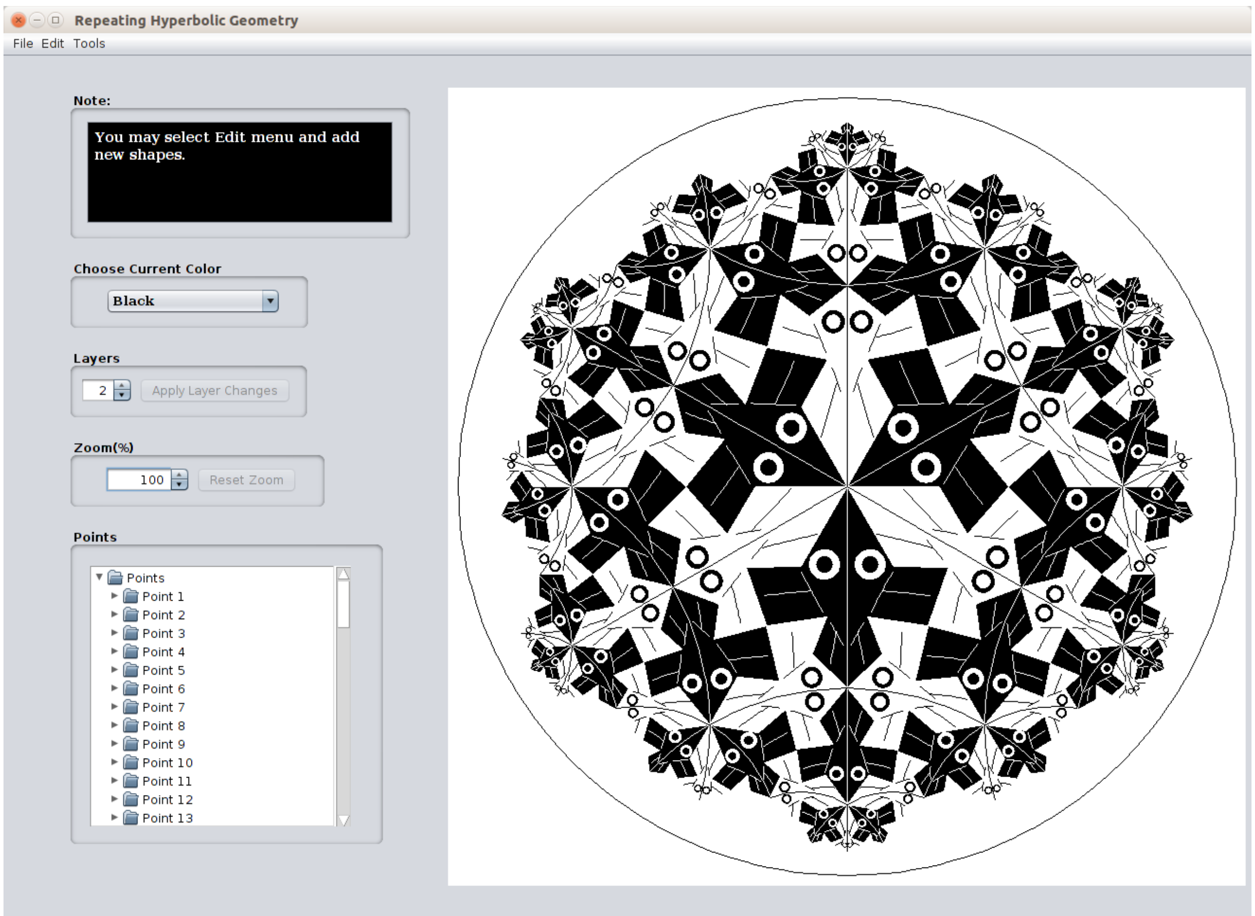

Figure 6.12: Escher’s Circle Limit I Pattern with 2 layers

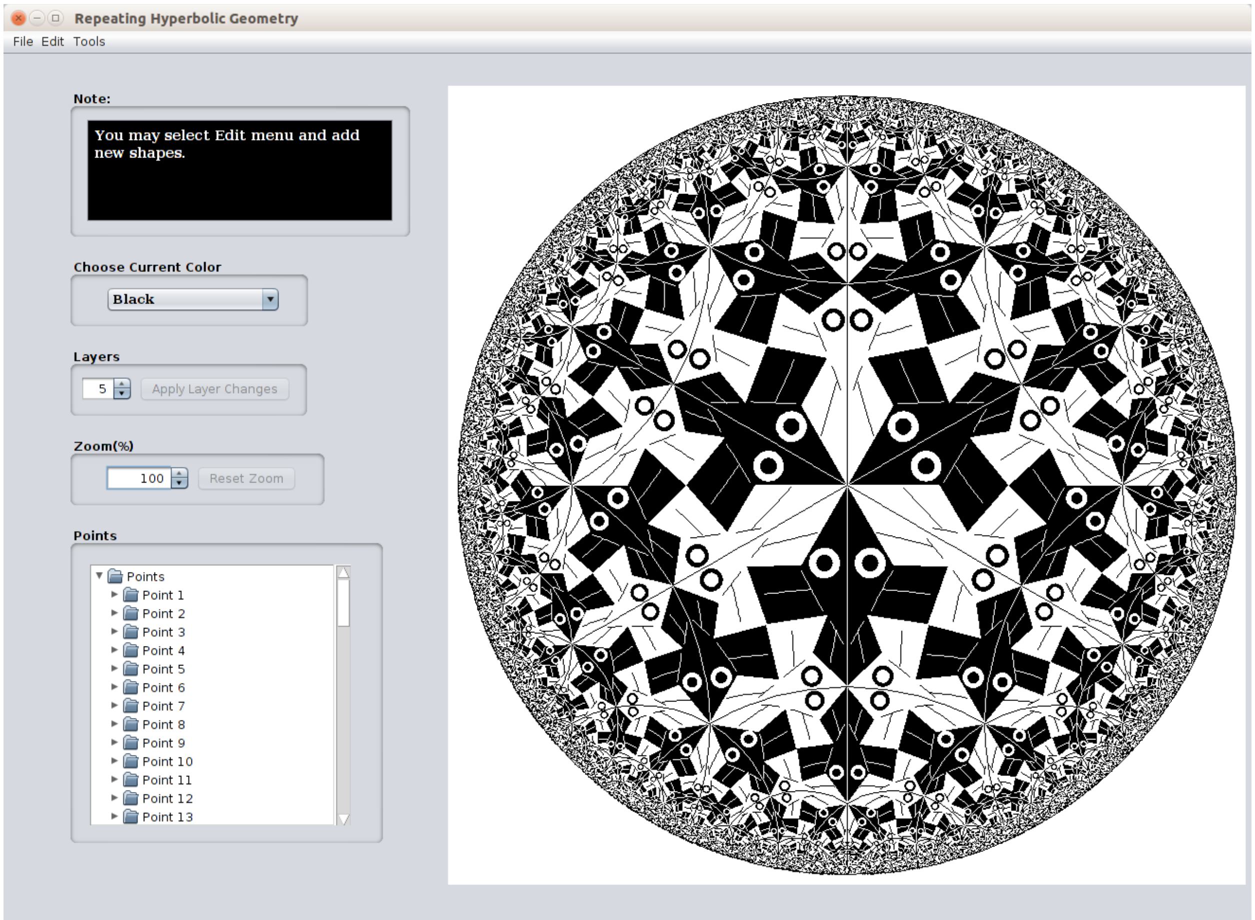

Figure 6.13: Escher's Circle Limit I Pattern with 5 layers

4. Zoom Panel - The value of spinner in this panel denotes the percentage of zoom that can be set for having a more closer look at the pattern. Figure 6.14 shows a part of Escher's Circle Limit I zoomed to 150%.

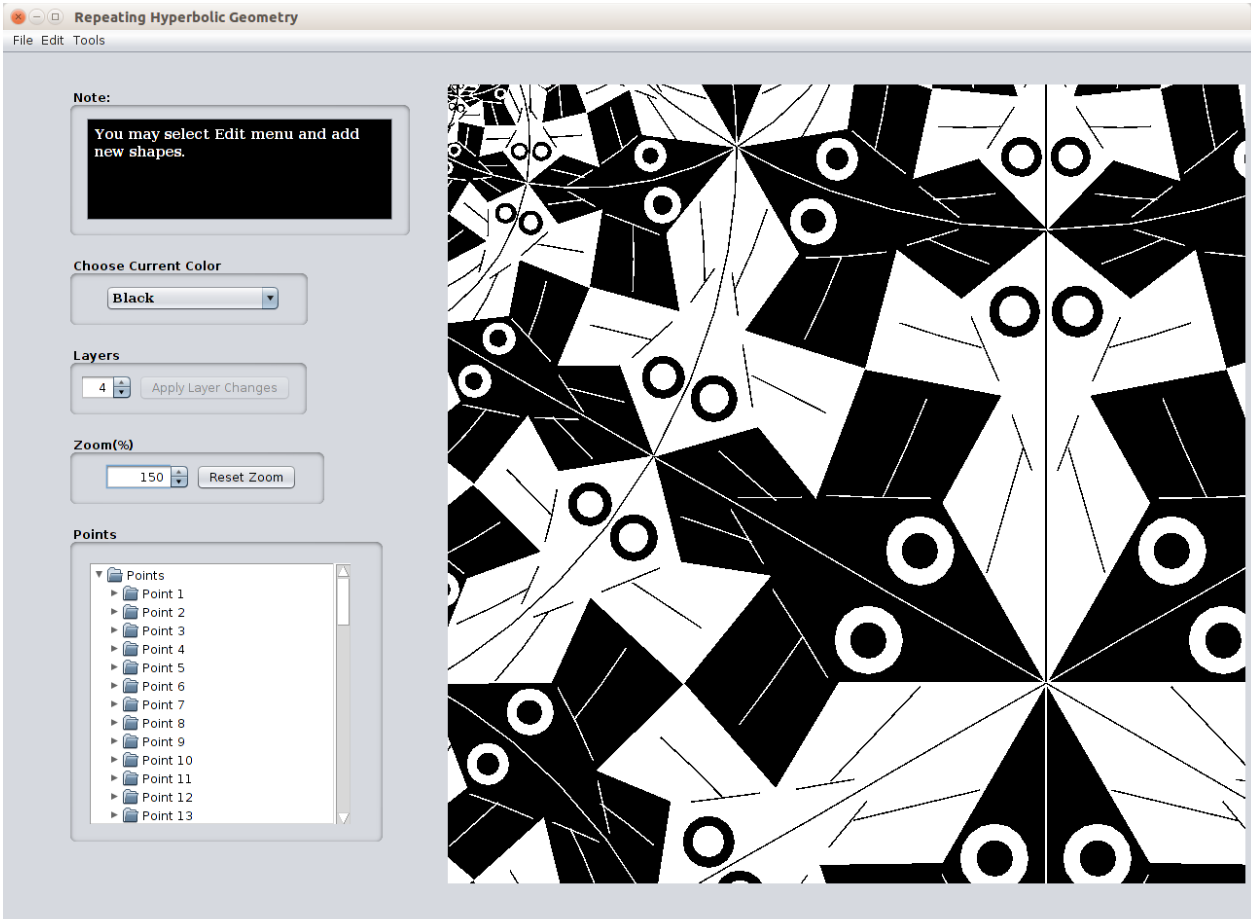

Figure 6.14: Escher's Circle Limit I zoomed to 150%

5. Points Box Panel - All the points that were used to draw various shapes will be displayed in this panel. Only points that are required to generate the motif are included in the list. Each point can further be expanded to show its properties such as its coordinates, color and type of point. Figure 6.15 shows the section of toolbar that has the points box. All points in the points box were expanded to expose their properties.

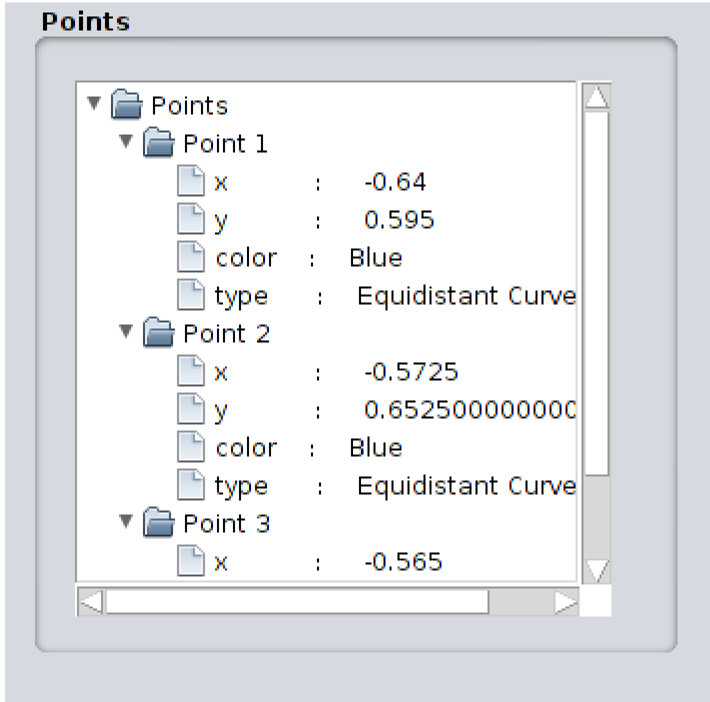

Figure 6.15: Points box with points expanded for the equidistant curve in Figure 6.10

The above sections provide a general overview of the functionalities supported by the user interface for this program.

Chapter – 7

Results

This chapter shows the results that are generated using the current program. As discussed earlier, this program was originally written in C language using Motif Framework. It was also written in C++ using Qt framework for the user interface by Becker [7]. But since these older programs had many deficiencies in terms of error checking, space and time complexities, it was later implemented in Java by Vejendla [2] making it more efficient and portable. The current program is an extension to the Java program to overcome interaction difficulties and limited support for shapes. The next few pages contain some screenshots of the current programs output which include special lines and curves. Some of the existing patterns are also included to demonstrate the backward compatibility of the current program.

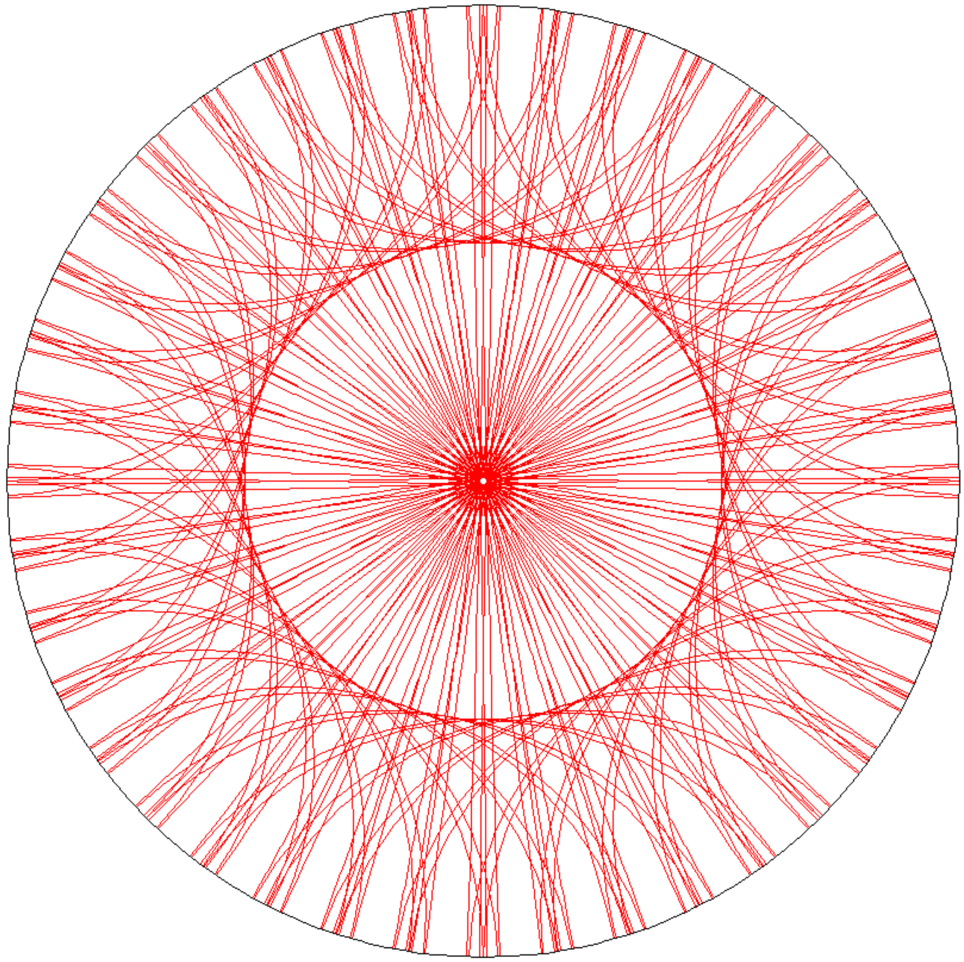

Figure 7.1: A pattern based on {40, 20} regular tessellation drawn using hyperbolic lines (with 1 layer)

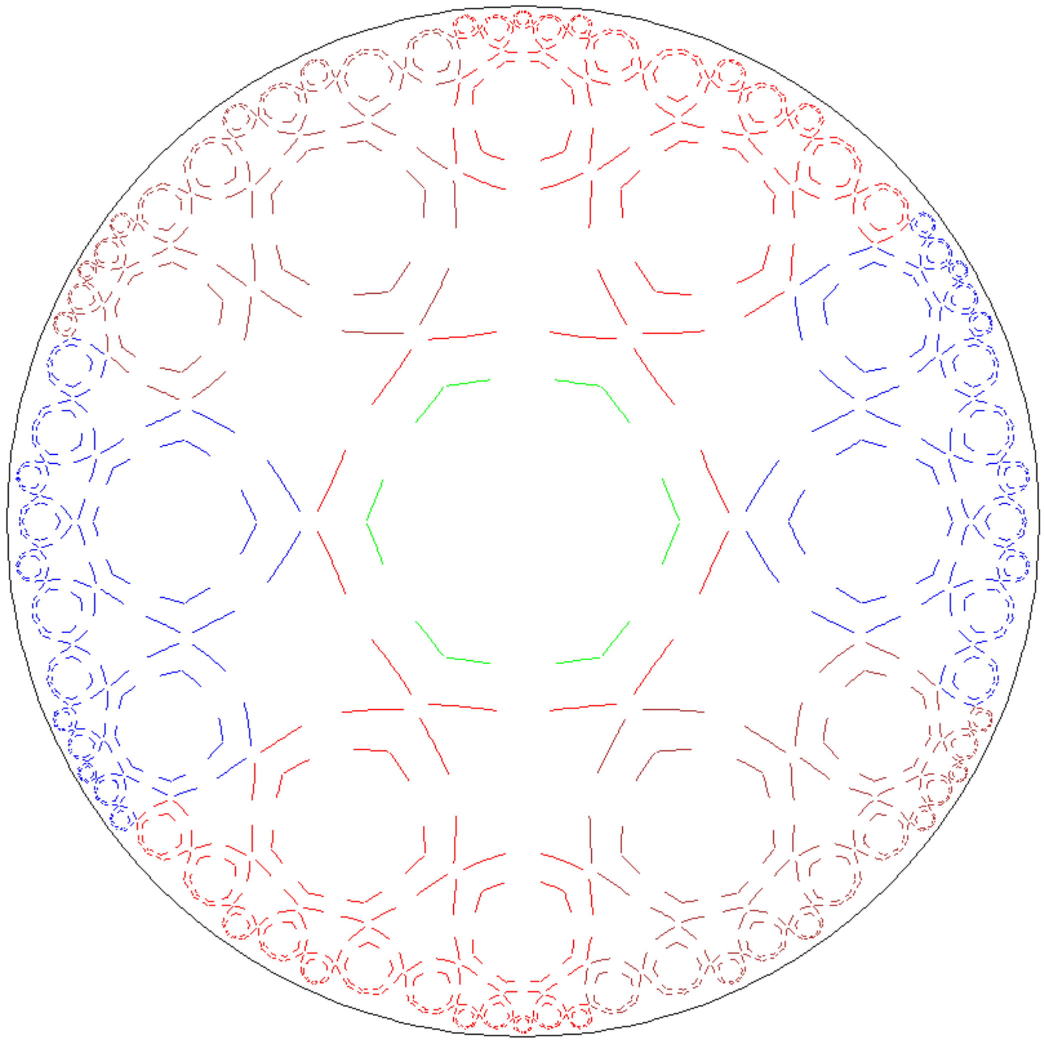

Figure 7.2: A pattern based on {6, 4} regular tessellation drawn using hyperbolic line segments (with 3 layers)

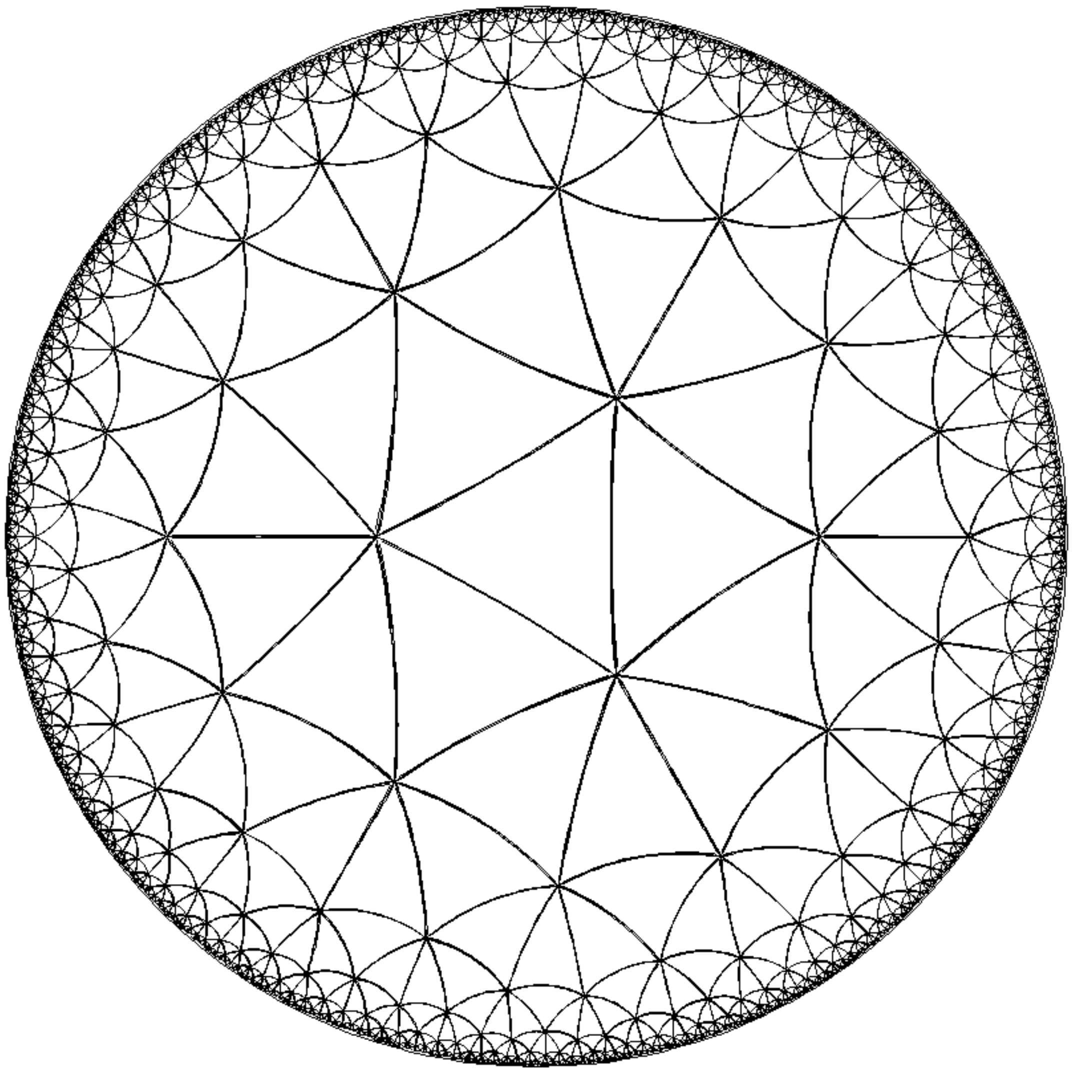

Figure 7.3: The {3, 7} regular tessellation drawn using hyperbolic line segments (with 7 layers)

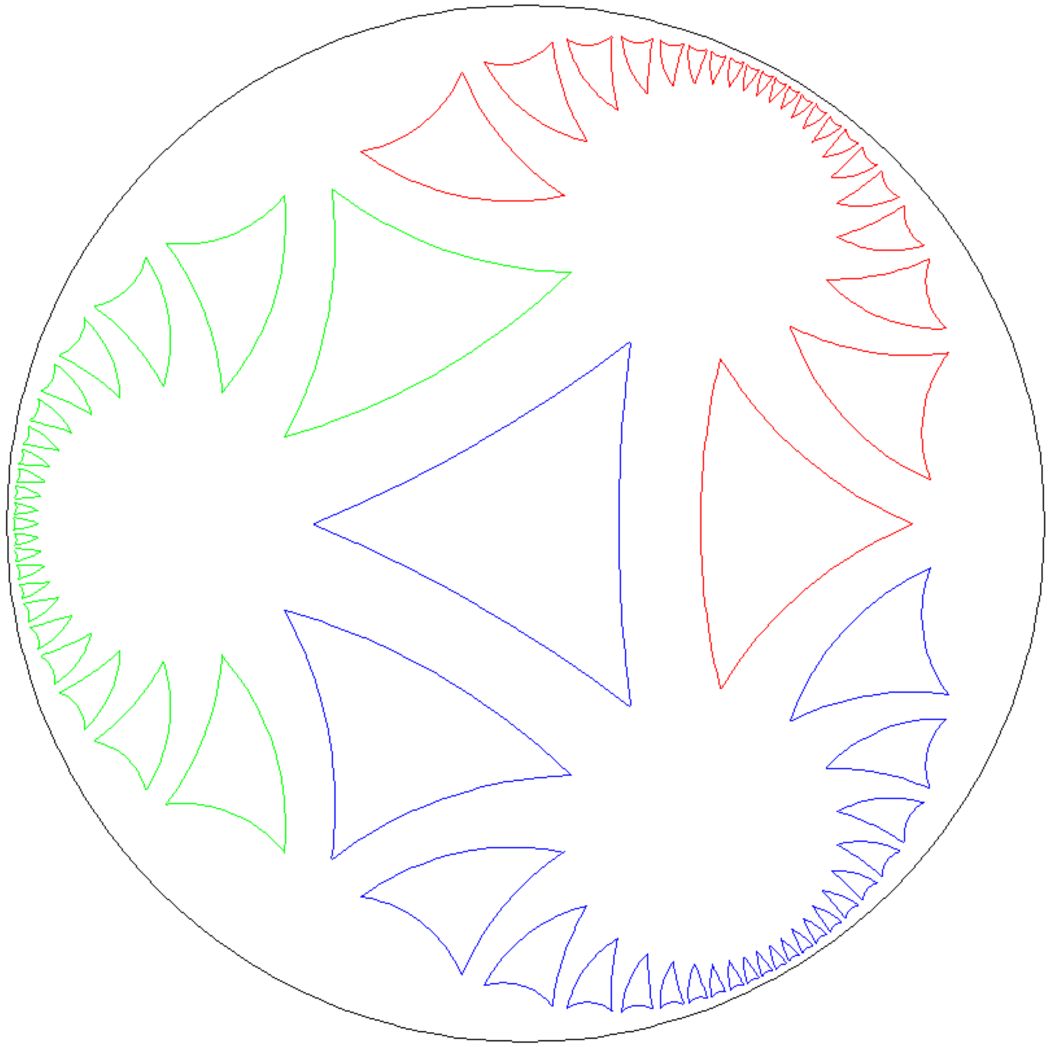

Figure 7.4: A pattern based on the {3, 24} regular tessellation drawn using hyperbolic line segments (with 2 layers)

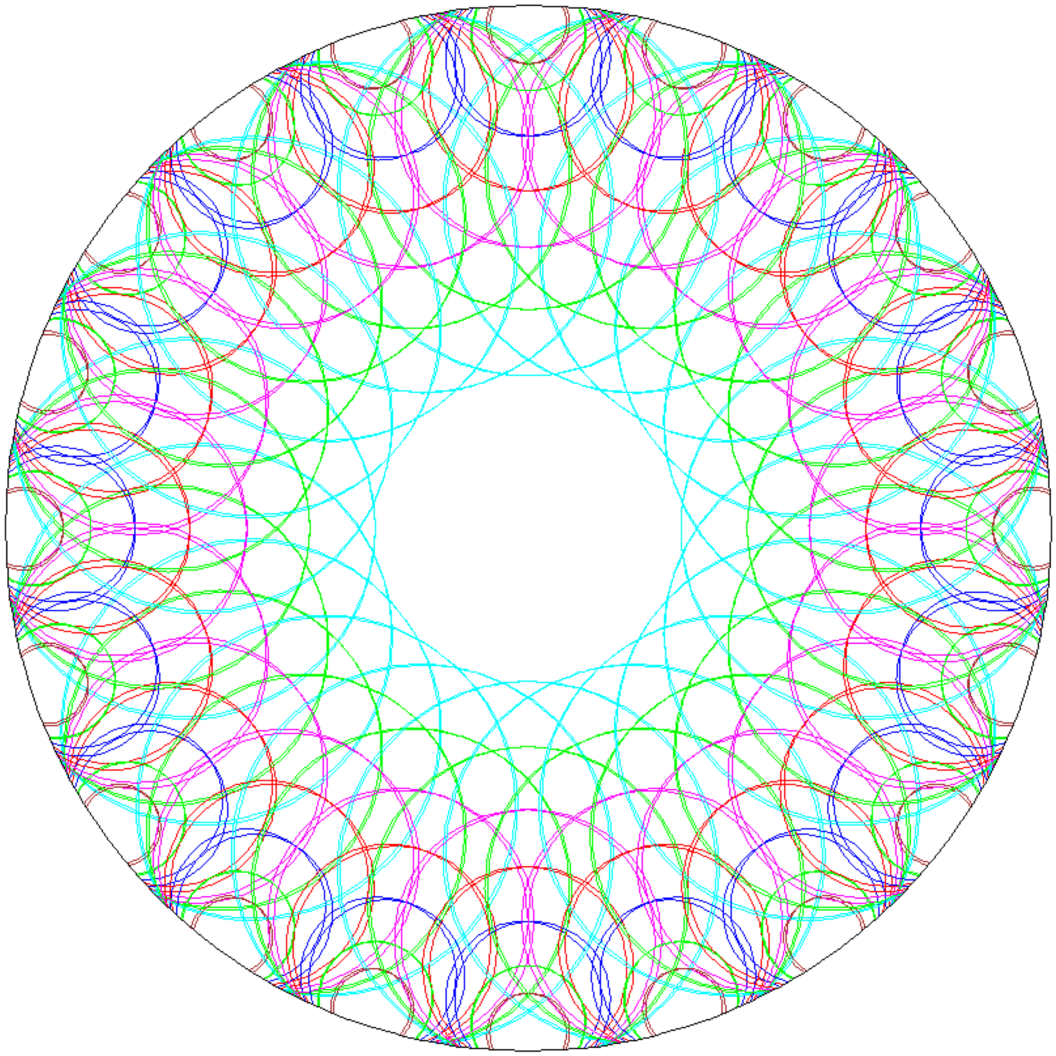

Figure 7.5: A pattern based on the {20, 10} regular tessellation drawn using equidistant curves (with 1 layer)

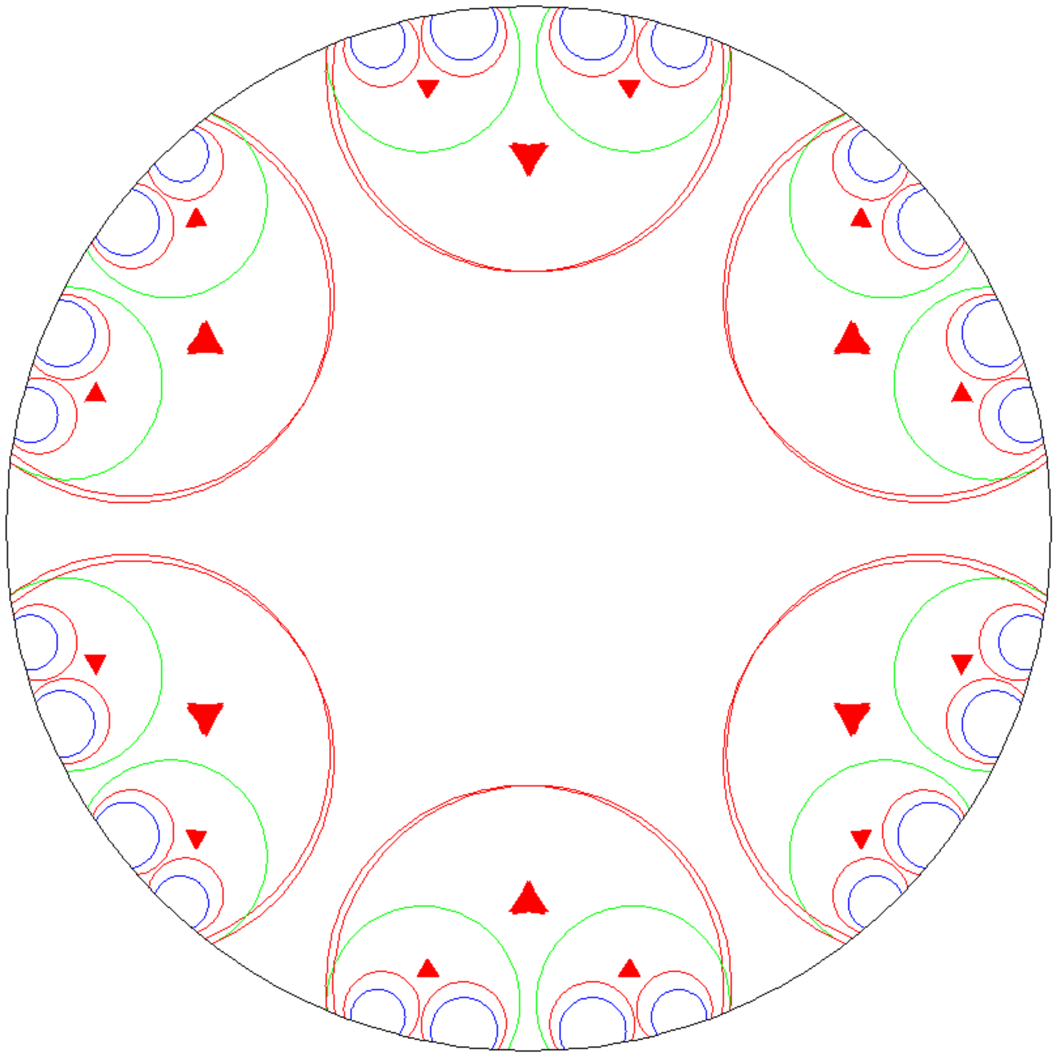

Figure 7.6: An owl-within-owl pattern based on the {6, 5} regular tessellation drawn using equidistant curves and filled polygons (with 1 layer)

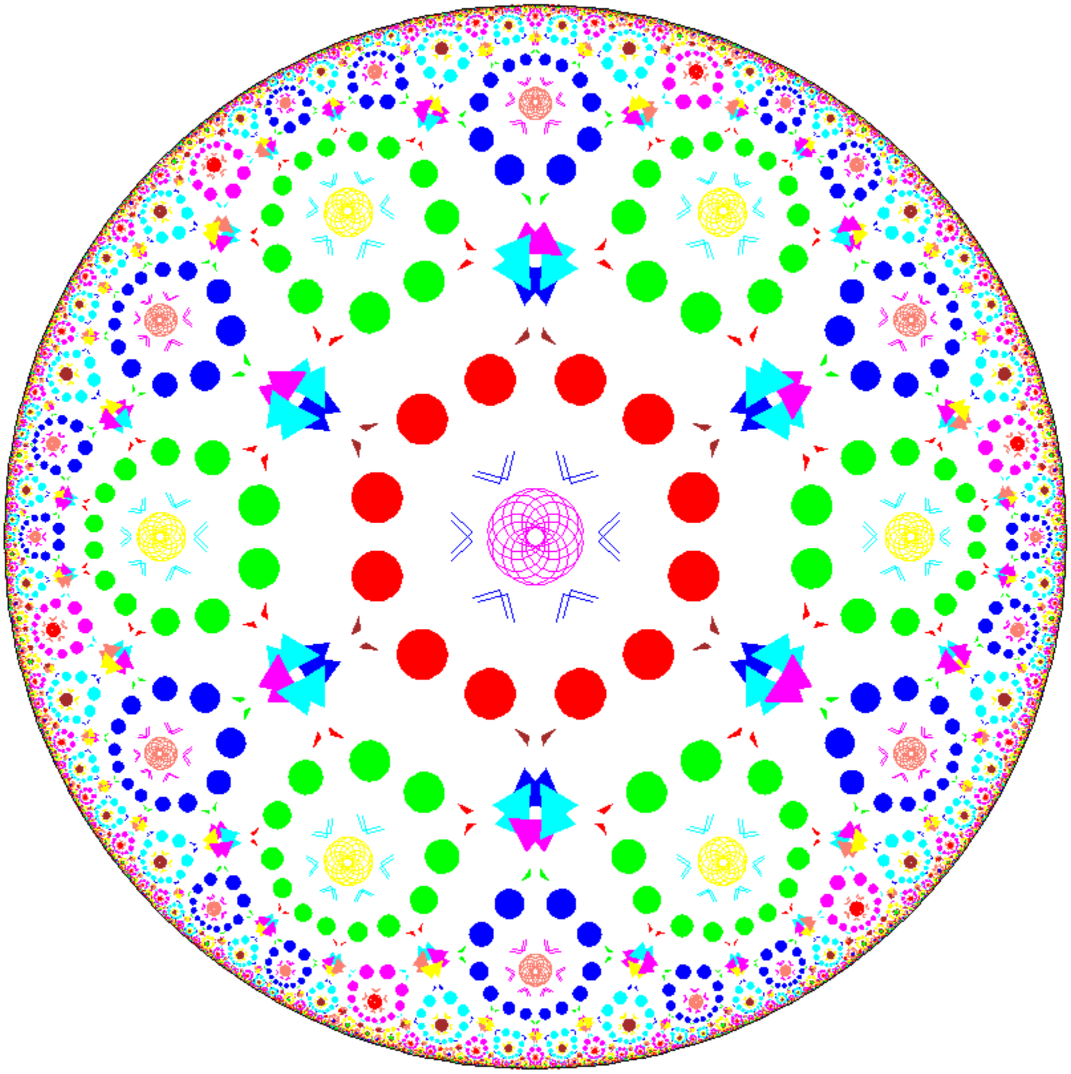

Figure 7.7: A pattern based on the {6, 4} regular tessellation drawn using multiple shapes (with 4 layers)

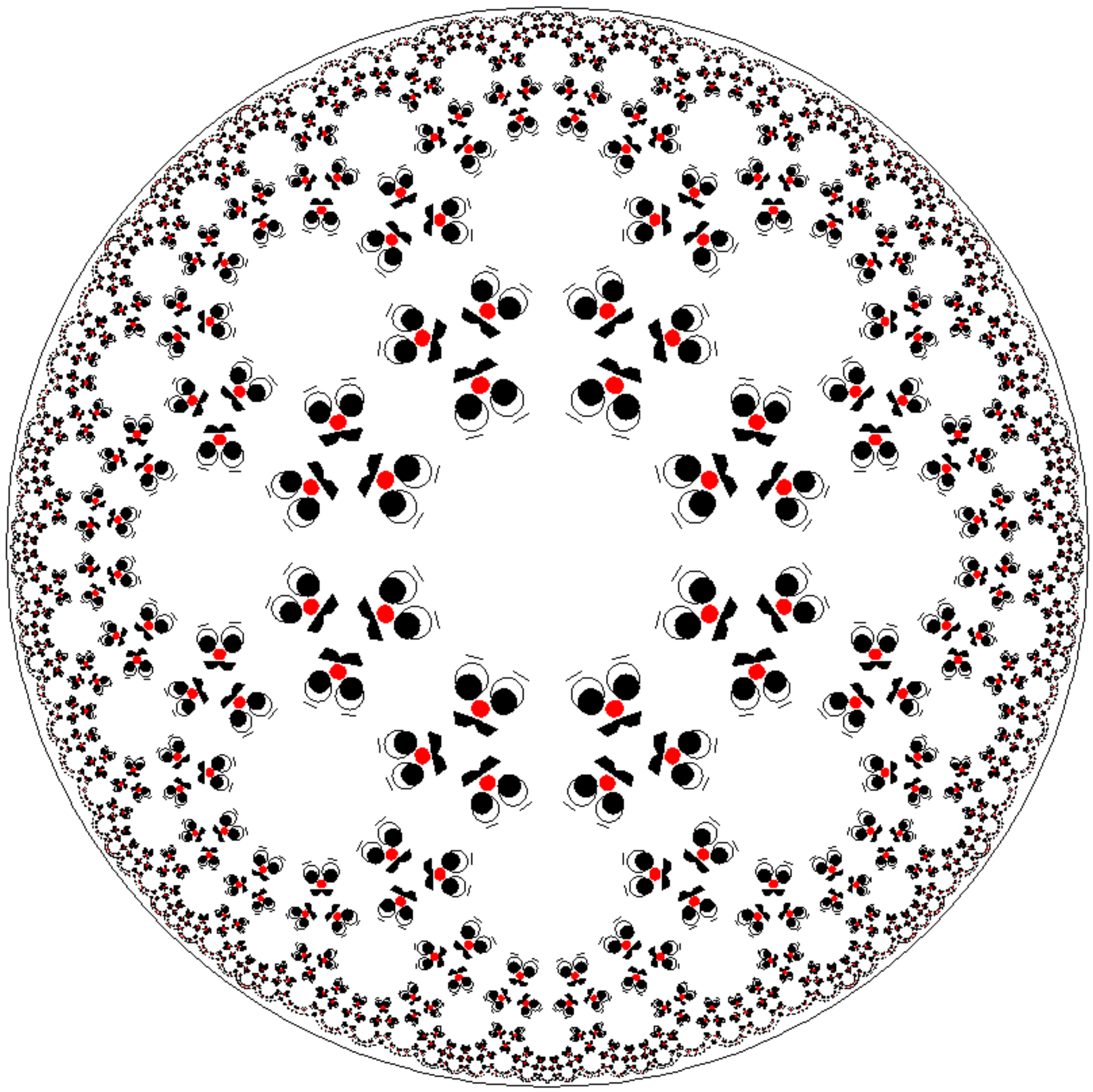

Figure 7.8: A jokers pattern based on the {8, 3} regular tessellation drawn using multiple shapes (with 4 layers)

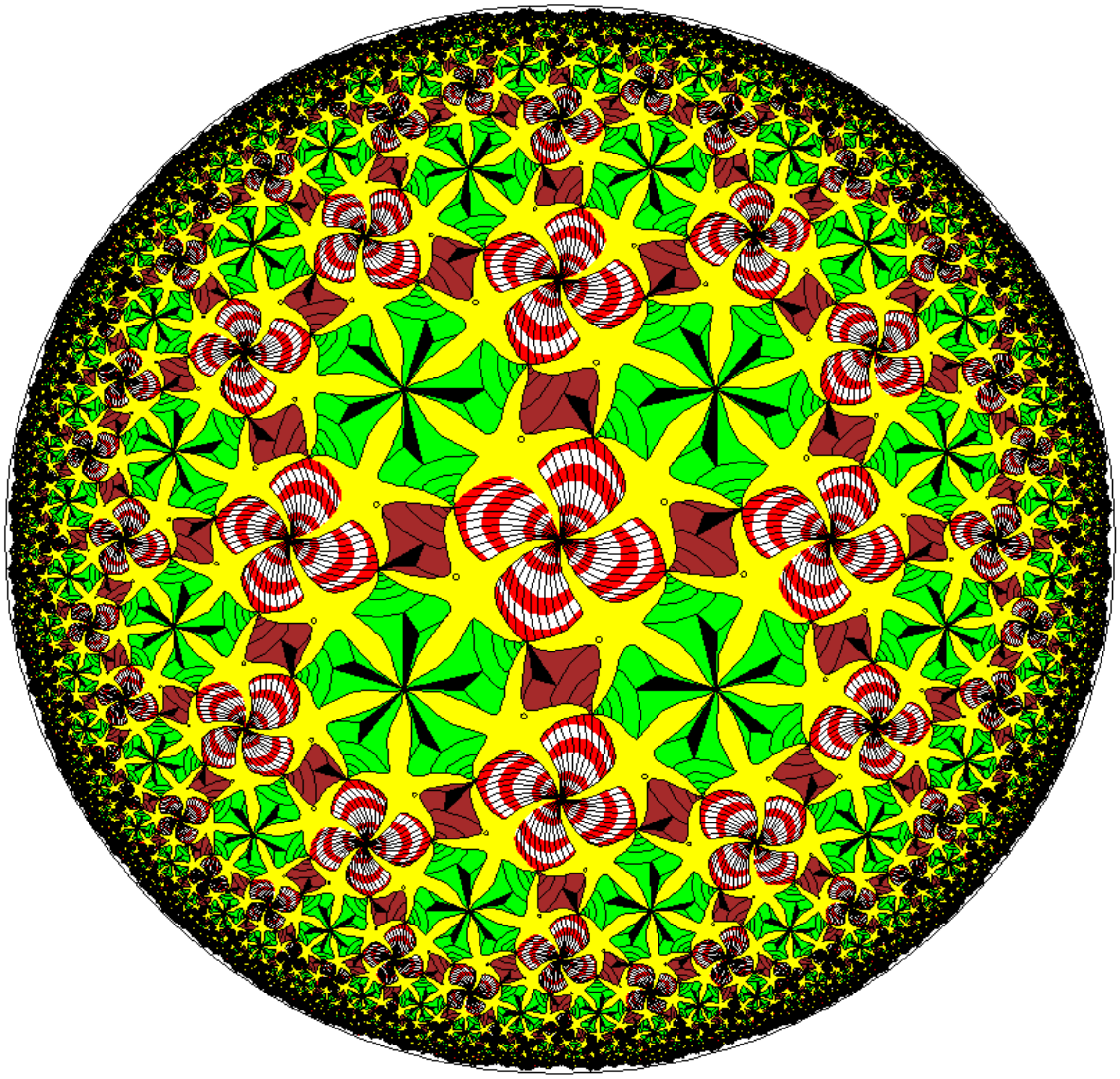

Figure 7.9: A pattern based on the {4, 5} regular tessellation drawn using multiple shapes (with 5 layers)

Chapter 8

Conclusion

The current research work is an attempt to enhance an existing Java program that generates repeating hyperbolic patterns based on regular tessellations {p, q}. This modified and improved version of the program fixes many issues with the older program, provides additional capabilities by supporting more complex shapes such as hyperbolic line segments, equidistant curves, and also provides a rich user interface. Also, the program is portable and compatible across platforms. It uses the Weierstrass Model for all computations that involves transformations. It also uses the Poincaré Disk Model for displaying the resulting patterns. It was rigorously tested on many data files and the results were as expected. As a result of this work, it provides support to more data files that involve special lines and curves.

Chapter 9

Future Work

This section covers the possible enhancements that can be made to the current program in future. This program could allow either p or q to be infinite such that all the edges or all the vertices are on the bounding circle. It could have more interaction capabilities to select and drag a drawn object.

The replication algorithm could further be extended to support semi-regular tessellations in which a pattern is drawn using two or more regular polygons. It could also be extended to generate repeating hyperbolic patterns on a three dimensional space.

The current algorithm redraws shapes every time when layers are changed. However, it could be implemented using dynamic programming to avoid repetitions thereby making it more time efficient. Another research direction could be to integrate artificial intelligence to reduce the number of points that are used to transform a central p-gon when the number of layers is increased.

References

[1] Dunham, Douglas, John Lindgren, and David Witte. "Creating repeating hyperbolic patterns." ACM SIGGRAPH Computer Graphics. Vol. 15. No. 3. ACM, 1981.

[2] Dunham, Douglas. "Hyperbolic symmetry." Computers & Mathematics with Applications 12.1 (1986): 139-153.

[3] Dunham, Douglas. "An algorithm to generate repeating hyperbolic patterns." the Proceedings of ISAMA (2007): 111-118.

[4] Abhijit Parsekar. A Unified data representation and Visualization of Patterns based on regular tessellations in the three classical geometries. Master's Thesis, University of Minnesota Duluth 2002.

[5] D. Dunham. Artistic patterns in hyperbolic geometry. In Bridges 1999 Conference Proceedings, pages 239 - 249, 1999.

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

[7] Christopher D. Becker. Creating Repeating Hyperbolic Patterns Based on Regular Tessellations. Master's Thesis, University of Minnesota Duluth 2012.

[8] Vejendla, Maneesha. *Generating Repeating Hyperbolic Patterns based on Regular Tessellations using an Applet*. Diss. UNIVERSITY OF MINNESOTA, 2013.

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

[10] Joel Castellanos. What is Non - Euclidean Geometry. http://www.cs.unm.edu/~joel/NonEuclid/noneuclidean.html Accessed 06 – 22 - 2015.

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

[12] Dunham, D. 1986, Hyperbolic symmetry, Computers and Mathematics with Applications, Volume 12B, pages 139 - 153.

[13] Luis Ollrtegui. Different kinds of Geometry. http://www.ehow.com/info_8774739_different-kinds-geometry.html Accessed 06 - 22 - 2015.

[14] Davis C. Royster. Hypercycles and Horocycles,
http://www.msc.uky.edu/droyster/courses/spring04/classnotes/Chapter%2010.pdf
Accessed 06 – 22 – 2015

[15] Robert Bob Gardener. Hyperbolic Geometry.
https://faculty.etsu.edu/gardnerr/noneuclidean/hyperbolic.pdf Accessed 06 - 22 - 2015

[16] Hyperbolic geometry. (2015, July 5). In Wikipedia, The Free Encyclopedia.
Retrieved 15:00, July 9, 2015, from
https://en.wikipedia.org/w/index.php?title=Hyperbolic_geometry&oldid=670052722

[17] Poincaré disk model. (2015, May 25). In Wikipedia, The Free Encyclopedia.
Retrieved 15:37, July 9, 2015, from
https://en.wikipedia.org/w/index.php?title=Poincar%C3%A9_disk_model&oldid=663955784

[18] Hypercycle (geometry). (2015, June 17). In Wikipedia, The Free Encyclopedia.
Retrieved 15:39, July 9, 2015, from
https://en.wikipedia.org/w/index.php?title=Hypercycle_(geometry)&oldid=667420297

Appendix

Data File Format

This section explains the format of the input data files for this program. Here is a sample data file, “*leaf64.dat”* that creates a leaf pattern based on a {6, 4} tessellation.

```
6 4 1 0 8 0
1 2 4 5 3 6 7 8
1 2 3 4 5 6 7 8
1 1 2 4 3 5 6 7 8
2 1 2 3 5 4 6 7 8
3 1 2 5 4 3 6 7 8
1 1 2 4 3 5 6 7 8
2 1 2 3 5 4 6 7 8
3 1 2 5 4 3 6 7 8
154
0.000000e+00 0.000000e+00 3 4 3
8.718379e-02 -1.867884e-03 3 5 3
1.242561e-01 3.397056e-02 3 5 3
1.903965e-01 -2.223319e-02 3 5 3
2.304449e-01 5.087258e-02 3 5 3
2.905698e-01 -1.361819e-02 3 5 3
3.247583e-01 2.470912e-02 3 5 3
4.142136e-01 0.000000e+00 3 5 3
4.182139e-01 8.033770e-02 3 5 3
3.999545e-01 1.231611e-01 3 5 3
4.294814e-01 1.620274e-01 3 5 3
4.482877e-01 2.588190e-01 3 5 3
4.482877e-01 2.588190e-01 3 5 3
```

3.536291e-01 2.871311e-01 3 5 3

3.234755e-01 3.243104e-01 3 5 3

2.780247e-01 3.206312e-01 3 5 3

2.071068e-01 3.587195e-01 3 5 3

1.409804e-01 2.936035e-01 3 5 3

1.570786e-01 2.448318e-01 3 5 3

7.116552e-02 2.250074e-01 3 5 3

1.144528e-01 1.537716e-01 3 5 3

3.270868e-02 1.245942e-01 3 5 3

4.520953e-02 7.456943e-02 3 5 3

0.000000e+00 0.000000e+00 3 6 3

4.482877e-01 2.588190e-01 6 9 3

3.838847e-01 2.204066e-01 6 10 3

3.536562e-01 1.973443e-01 6 10 3

3.344118e-01 1.857517e-01 6 10 3

2.974471e-01 1.645867e-01 6 10 3

2.123696e-01 1.126656e-01 6 10 3

1.246093e-01 7.241720e-02 6 10 3

9.021925e-02 5.323166e-02 6 10 3

7.601667e-02 4.425873e-02 6 10 3

6.303594e-02 3.107443e-02 6 10 3

4.924763e-02 2.494073e-02 6 11 3

3.838847e-01 2.204066e-01 6 9 3

3.664977e-01 1.855823e-01 6 10 3

3.612812e-01 1.580484e-01 6 10 3

3.582736e-01 9.265356e-02 6 10 3

3.642796e-01 5.122957e-02 6 11 3

3.612812e-01 1.580484e-01 6 9 3

3.870233e-01 1.397191e-01 6 11 3

3.582736e-01 9.265356e-02 6 9 3

3.813706e-01 6.749890e-02 6 11 3

3.612812e-01 1.580484e-01 6 9 3

3.388365e-01 9.335190e-02 6 11 3

2.974471e-01 1.645867e-01 6 9 3

2.843337e-01 1.407237e-01 6 10 3

2.811153e-01 9.931874e-02 6 10 3

2.955800e-01 2.987712e-02 6 11 3

2.974471e-01 1.645867e-01 6 9 3

2.893049e-01 1.406951e-01 6 10 3

2.960149e-01 1.195620e-01 6 10 3

3.142723e-01 9.656515e-02 6 11 3

2.811153e-01 9.931874e-02 6 9 3

3.039480e-01 5.042158e-02 6 11 3

2.811153e-01 9.931874e-02 6 9 3

2.726883e-01 8.099469e-02 6 10 3

2.586071e-01 6.773849e-02 6 11 3

2.123696e-01 1.126656e-01 6 9 3

1.967700e-01 8.785376e-02 6 10 3

1.902625e-01 7.263855e-02 6 10 3

1.700250e-01 1.530262e-02 6 11 3

1.967700e-01 8.785376e-02 6 9 3

1.971127e-01 7.237390e-02 6 10 3

2.044995e-01 6.109038e-02 6 11 3

1.902625e-01 7.263855e-02 6 9 3

1.889700e-01 5.003681e-02 6 10 3

1.924385e-01 2.775615e-02 6 11 3

1.902625e-01 7.263855e-02 6 9 3

1.762342e-01 5.091897e-02 6 10 3

1.620725e-01 3.722021e-02 6 11 3

1.246093e-01 7.241720e-02 6 9 3

1.117956e-01 6.194406e-02 6 10 3

1.032573e-01 4.766670e-02 6 11 3

9.021925e-02 5.323166e-02 6 9 3

8.005203e-02 5.472995e-02 6 10 3

7.086049e-02 5.433310e-02 6 11 3

2.123696e-01 1.126656e-01 6 9 3

1.739407e-01 1.108010e-01 6 10 3

1.342473e-01 1.083938e-01 6 10 3

1.153302e-01 1.021968e-01 6 10 3

8.727415e-02 1.029964e-01 6 10 3

6.177725e-02 1.100493e-01 6 11 3

1.739407e-01 1.108010e-01 6 9 3

1.541641e-01 9.448453e-02 6 10 3

1.445702e-01 9.196874e-02 6 11 3

1.153302e-01 1.021968e-01 6 9 3

7.396523e-02 8.776484e-02 6 11 3

1.342473e-01 1.083938e-01 6 9 3

1.281707e-01 1.171384e-01 6 10 3

1.158224e-01 1.267150e-01 6 11 3

3.344118e-01 1.857517e-01 6 9 3

2.703299e-01 1.661452e-01 6 10 3

2.243780e-01 1.665960e-01 6 10 3

2.007620e-01 1.702683e-01 6 10 3

1.537569e-01 1.875770e-01 6 10 3

1.254126e-01 2.051917e-01 6 11 3

2.243780e-01 1.665960e-01 6 9 3

1.896600e-01 1.594824e-01 6 10 3

1.590107e-01 1.477371e-01 6 11 3

1.537569e-01 1.875770e-01 6 9 3

1.488072e-01 1.833674e-01 6 10 3

1.374977e-01 1.778609e-01 6 11 3

2.007620e-01 1.702683e-01 6 9 3

1.882355e-01 1.797716e-01 6 10 3

1.703520e-01 2.074055e-01 6 11 3

2.703299e-01 1.661452e-01 6 9 3

2.678915e-01 1.785477e-01 6 10 3

2.547833e-01 1.907303e-01 6 10 3

2.374570e-01 1.970701e-01 6 11 3

3.838847e-01 2.204066e-01 6 9 3

3.149163e-01 2.165769e-01 6 10 3

2.880510e-01 2.249948e-01 6 10 3

2.592256e-01 2.388251e-01 6 10 3

2.248457e-01 2.623780e-01 6 10 3

1.984553e-01 2.931059e-01 6 11 3

2.880510e-01 2.249948e-01 6 9 3

2.632461e-01 2.232413e-01 6 10 3

2.488376e-01 2.160667e-01 6 11 3

2.592256e-01 2.388251e-01 6 9 3

2.324054e-01 2.424553e-01 6 10 3

2.043148e-01 2.383755e-01 6 11 3

2.880510e-01 2.249948e-01 6 9 3

2.682065e-01 2.395910e-01 6 10 3

2.584993e-01 2.596991e-01 6 10 3

2.570599e-01 2.841959e-01 6 11 3

3.149163e-01 2.165769e-01 6 9 3

3.077455e-01 2.311557e-01 6 10 3

3.085148e-01 2.584319e-01 6 11 3

0.000000e+00 0.000000e+00 1 9 3

8.718379e-02 -1.867884e-03 1 10 3

1.242561e-01 3.397056e-02 1 10 3

1.903965e-01 -2.223319e-02 1 10 3

2.304449e-01 5.087258e-02 1 10 3

2.905698e-01 -1.361819e-02 1 10 3

3.247583e-01 2.470912e-02 1 10 3

4.142136e-01 0.000000e+00 1 10 3

4.182139e-01 8.033770e-02 1 10 3

3.999545e-01 1.231611e-01 1 10 3

4.294814e-01 1.620274e-01 1 10 3

4.482877e-01 2.588190e-01 1 10 3

4.482877e-01 2.588190e-01 1 10 3

3.536291e-01 2.871311e-01 1 10 3

3.234755e-01 3.243104e-01 1 10 3

2.780247e-01 3.206312e-01 1 10 3

2.071068e-01 3.587195e-01 1 10 3

1.409804e-01 2.936035e-01 1 10 3

1.570786e-01 2.448318e-01 1 10 3

7.116552e-02 2.250074e-01 1 10 3

1.144528e-01 1.537716e-01 1 10 3

3.270868e-02 1.245942e-01 1 10 3

4.520953e-02 7.456943e-02 1 10 3

0.000000e+00 0.000000e+00 1 11 3

The pattern generated using the above data file is shown in Figure A.1:

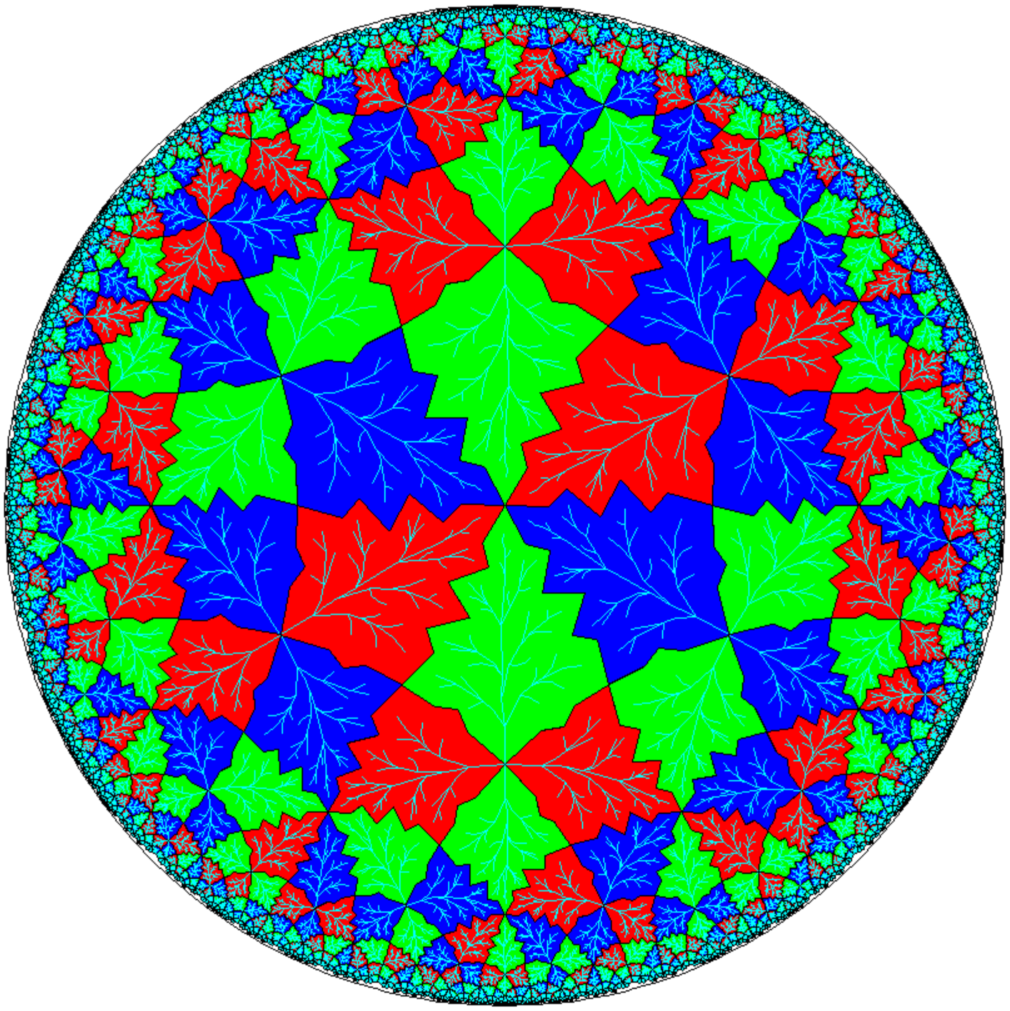

Figure A.1: A pattern based on the {6, 4} tessellation

In the first line, 6 4 1 0 8 0:

• The first number is the value of p, i.e. p = 6 in this case. The central polygon in Figure A1 is an 6-gon.

• The second number is the value of q, i.e. q = 4 in this case (this pattern is based on the tessellation {6, 4}). The 6-gon in Figure A.1 meets four other 6-gons at each vertex.

• The third number, 1 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 ∗

π/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 4 5 3 6 7 8, 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 1 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 3 5 6 7 8 (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 2 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 3 5 4 6 7 8. This pattern continues for four more lines. The next line consists of a single number, the number of points that make up the motif. It is 154 in this case. Following that line are 154 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-coordinate 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”
- – 20 “Equidistant Curve” (there must be three of these in succession)
- – 21 “Horocycle”
- – 22 “Horocycle With Lines”
- – 23 “Hyperbolic Line Segment”
- – 24 “Hyperbolic Line”

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