Some Math Behind M.C. Escher's *Circle Limit* Patterns

Douglas Dunham
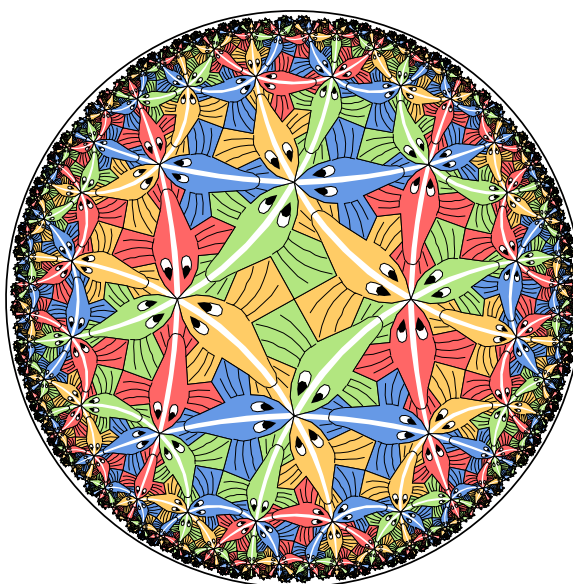
Department of Computer Science

University of Minnesota, Duluth

Duluth, MN 55812-3036, USA

E-mail: ddunham@d.umn.edu

Web Site: http://www.d.umn.edu/~ddunham/

Outline

- **Some history of *Circle Limit* patterns**

- **Some mathematics and symmetries of *Circle Limit* patterns**

- **How to make *Circle Limit* patterns**

- **Conclusions**

In Figure 7 we see another such group, with the important difference that now the angle-sum of each triangle is less than two right angles and the number of triangles is infinite. The group is again generated by inversions in three circles, but the figure is no longer a picture of something in space. We do not find it as easy as before to imagine that the smaller peripheral triangles are the same size as those in the middle. But in so far as we succeed in stretching our imagination to this extent, we are visualizing the non-Euclidean plane of Gauss, Bolyai and Lobatschewsky.
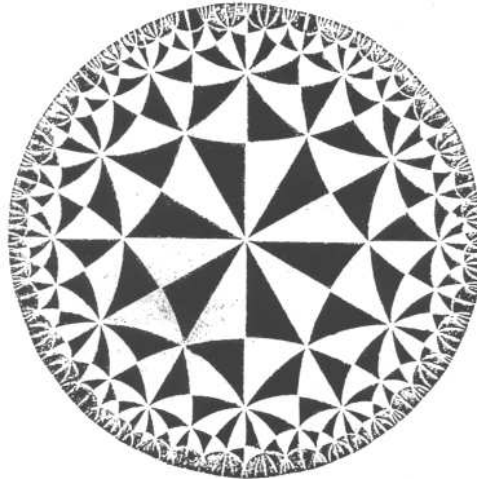

FIGURE 7

This is one way to generalize the idea of symmetry. Another is to increase the number of dimensions. Plate IV shows a wire model made by Mr. P. S. Donchian of Hartford, Conn.[3] This represents an orthogonal projection of a four-dimensional hyper-solid bounded by 120 regular dodecahedra. The model has the same 120 symmetry-operations as a single dodecahedron, but the four-dimensional polytope itself has a symmetry group of order $20^2 = 14400$
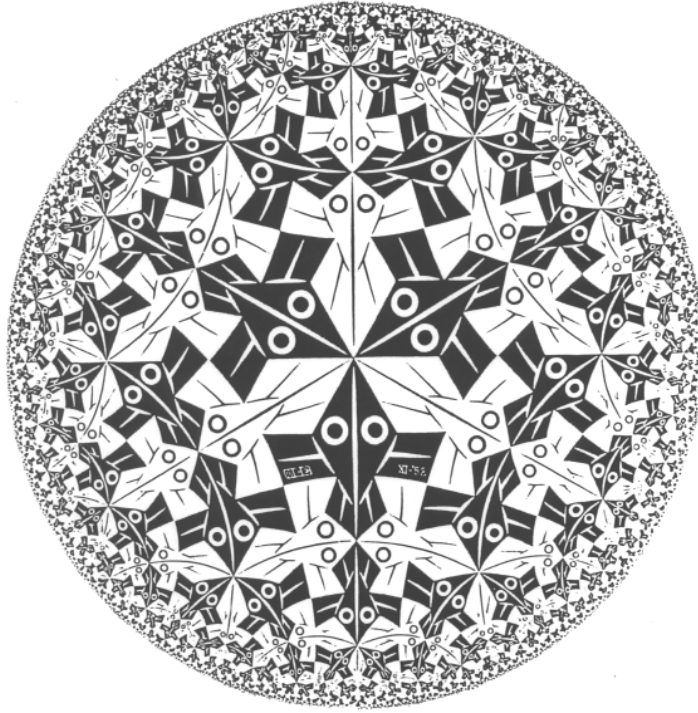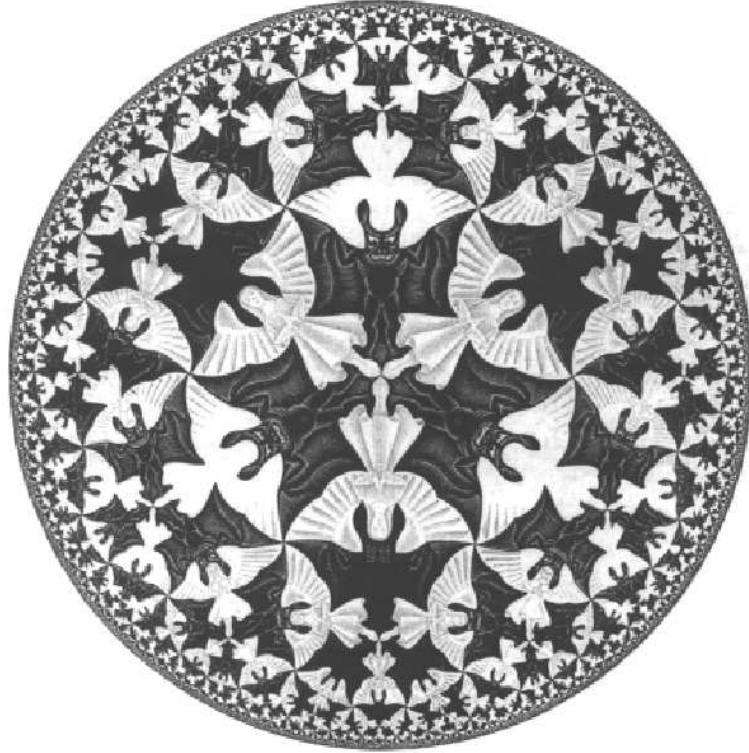
Circle Limit I

Circle Limit II

Circle Limit III

Circle Limit IV

Two Math Questions:

- **What are the correct orientations for the *Circle Limit* patterns?**

- **What is the symmetry groups of the *Circle Limit* patterns?**

What are the correct orientations of *Circle Limit I*, *Circle Limit II*, and *Circle Limit III*?

Answer: **They are usually correctly oriented, except that *Circle Limit II* is sometimes rotated 45 degrees so that cross arms are horizontal.**

What is the correct orientation for *Circle Limit IV*? Answer:

A Sub-question

Why aren't there signatures in all three of the blank devils' faces next to the central angels?
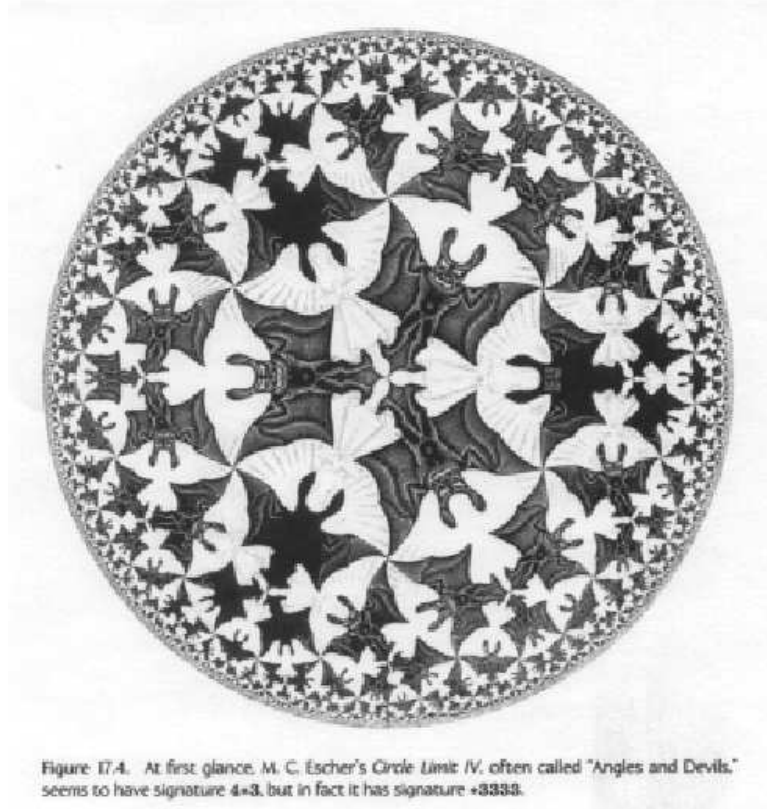
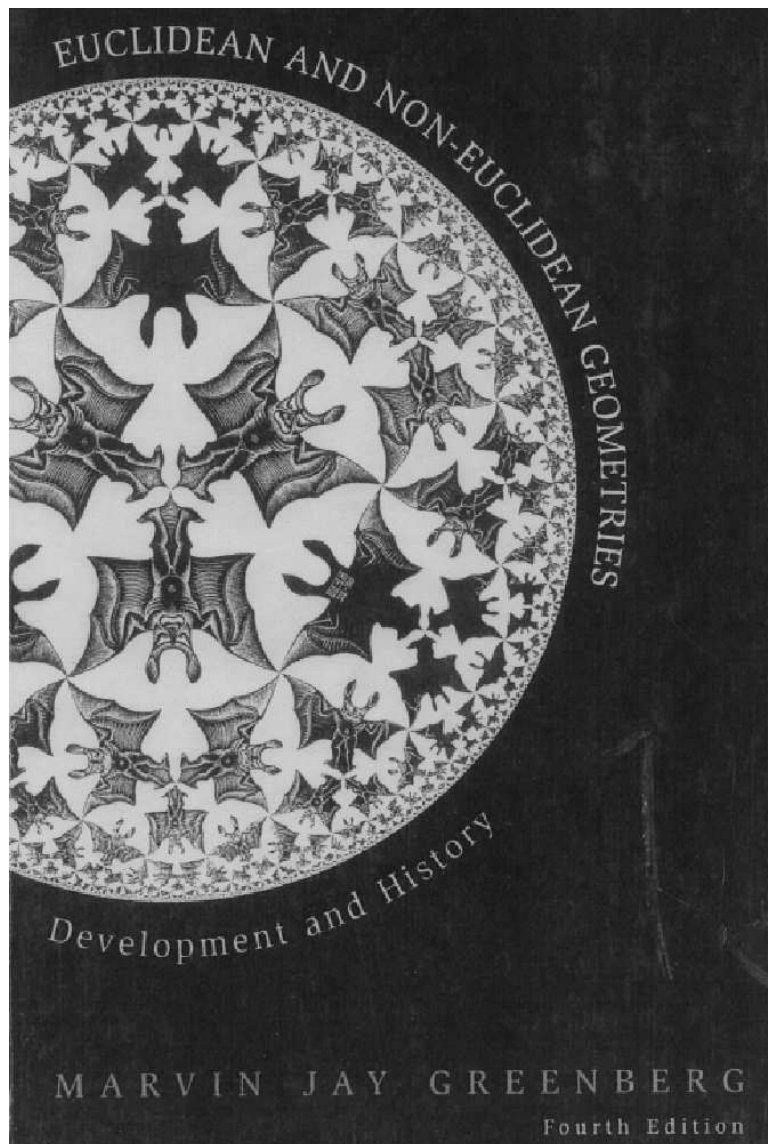# Escher's Signature (graphic)

Escher's Signature (text)

```
MCE
VII-'60
```

Why Examine Orientation Now?

- *The Symmetries of Things*
  **Conway, Burgiel, Goodman‑Strass
  A.K. Peters, 2008, Figure 17.4 page 224.**

- *Euclidean and Non-Euclidean Geometries: Development and History* **4**th **Ed.
  Marvin Greenberg
  W.H. Freeman, 2008, cover.**

- *Analysis, Geometry, and Modeling in Finance: Advanced Methods in Option Pricing,* **P. Henry-Labordère, Chapman & Hall/CRC Press, 2008, cover.**

- *Math and Art: an Introduction to Visual Mathematics,* **S. Kalajdzievski, Chapman & Hall/CRC Press, 2008, page 166.**
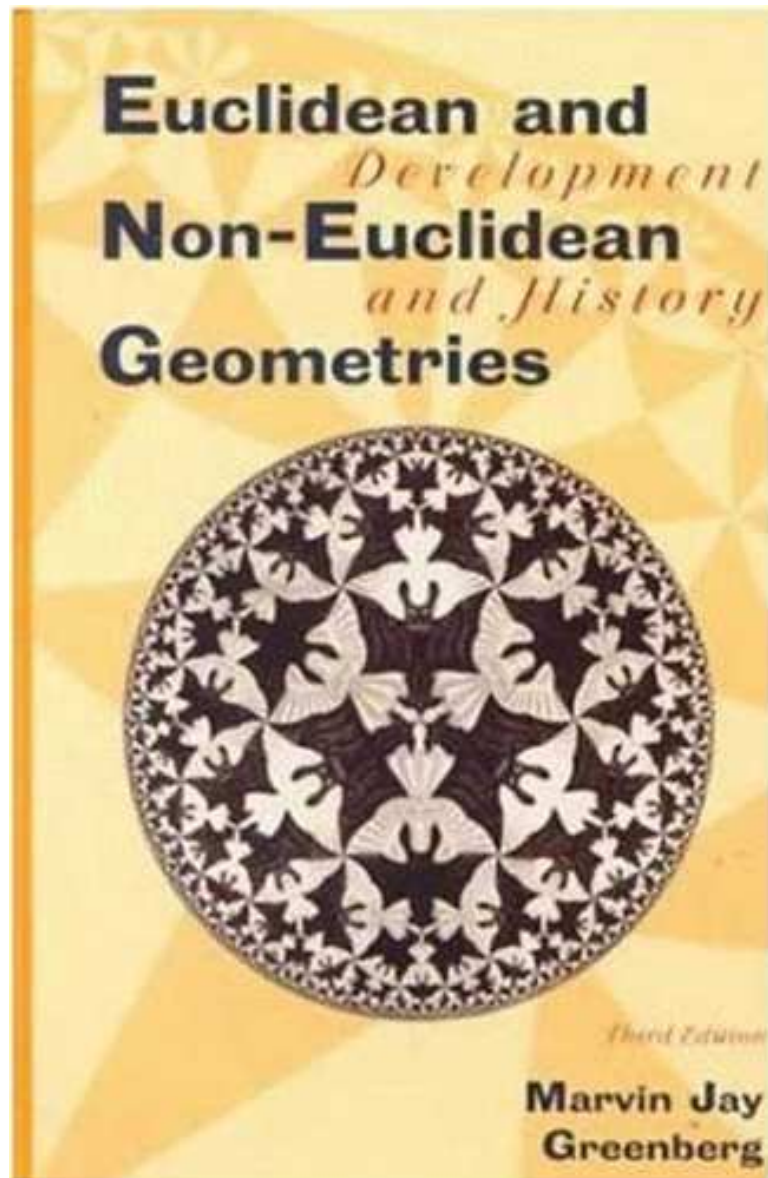
Figure 17.4. At first glance, M. C. Escher's *Circle Limit IV*, often called "Angles and Devils," seems to have signature 4∗3, but in fact it has signature ∗3333.

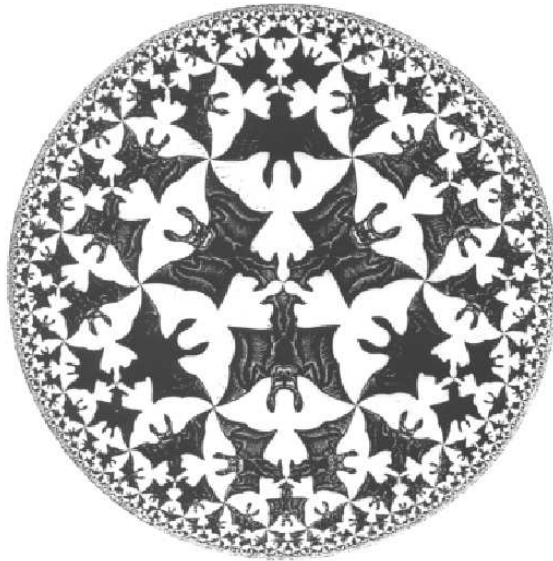# *Euclidean and Non-Euclidean Geometries,* 4th **Ed.**

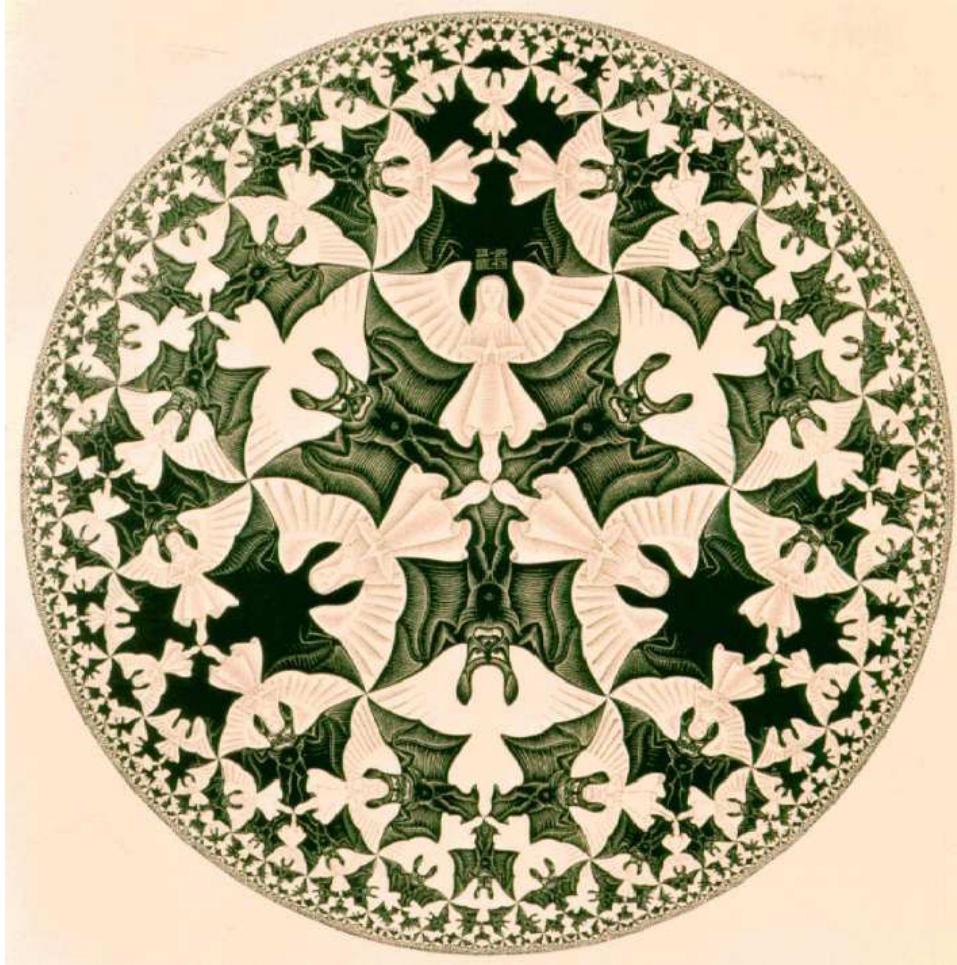# *Euclidean and Non-Euclidean Geometries,* 3rd **Ed.**

The World of M.C. Escher His Life and Complete Graphic Work
(Harry N. Abrams, 1981)

- **Page 98 (large image): correctly oriented**

- **Page 322 (small image — Catalogue number 436):**

M.C. Escher web site

The Signature

# Second Question: What is the symmetry group of *Circle Limit I* ?

It is $cmm_{3,2}$ using a notation suggested by Coxeter.

In Coxeter's notation $cmm_{p/2,q/2}$ is the symmetry group generated by reflections in the sides of a rhombus with vertex angles $\pi/2p$ and $\pi/2q$ (p and q must be even). This generalizes the Euclidean group $cmm = cmm_{2,2}$.

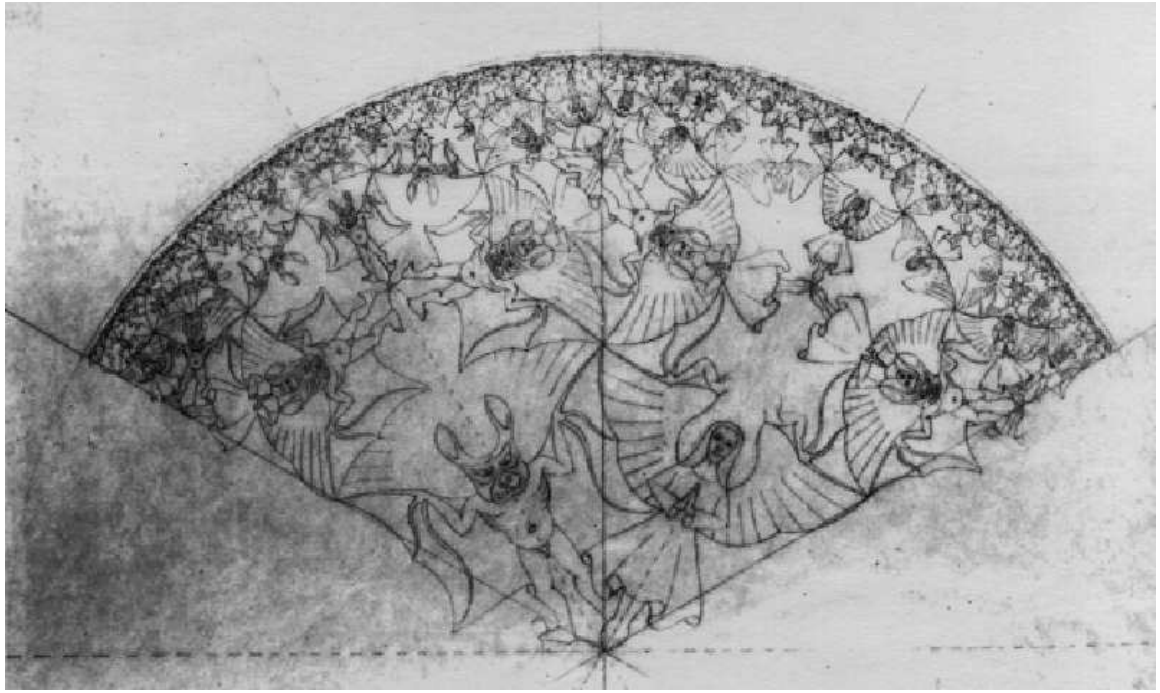# What are the symmetry groups of *Circle Limit II* and *Circle Limit IV* ?

Answer: they are $[3^+, 8]$ and $[4^+, 6]$ (again, Coxeter's notation) — but the last is true *only* if we assume that the interior details of all angels and devils have been filled in.

A *Circle Limit IV* Pattern with Symmetry Group [4$^+$, 6]
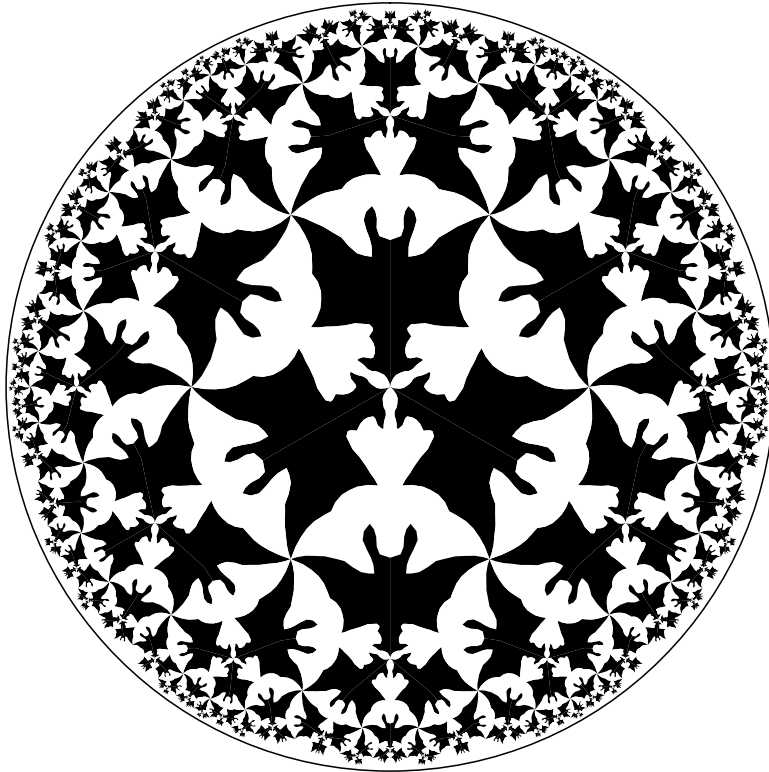
What is the actual symmetry group of *Circle Limit IV* ? C$_3$?

It seems to be D_3

Escher (in *M.C. Escher The Graphic Work* Barnes & Noble/TASCHEN 2007 ISBN 0-7607-9669-6, page 10):

Here too, we have the components diminishing in size as they move outwards. The six largest (three white angels and three black devils) are arranged about the centre and radiate from it. The disc is divided into six sections in which, turn and turn about, the angels on a black background and then the devils on a white one gain the upper hand. In this way, heaven and hell change place six times. In the intermediate, "earthly" stages, they are equivalent.

What is the symmetry group of *Circle Limit III* ?

Answer: it is (4,3,3) in a common notation, generated by 4-fold rotations about the right fin tips of the fish, and 3-fold rotations about the noses and left fin tips.

Creating Repeating Hyperbolic Patterns
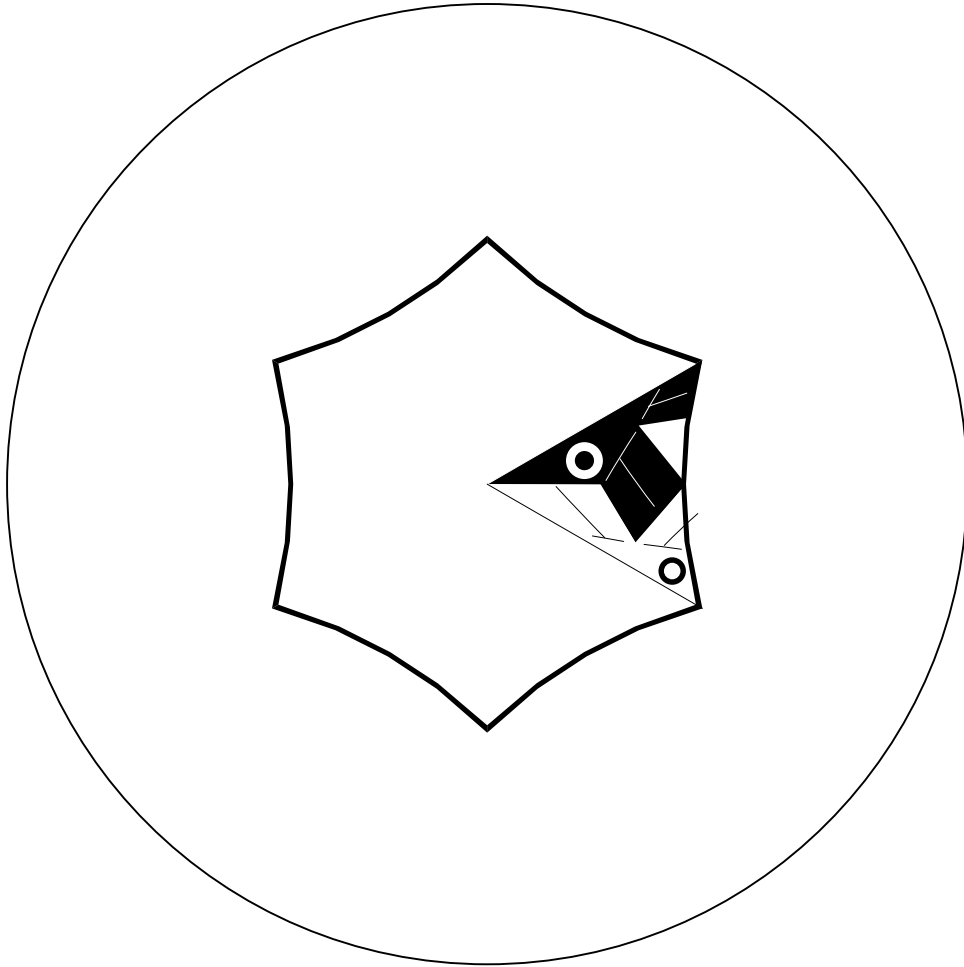
A two-step process:

1. Design the basic subpattern or *motif* (discussed in Math & Design 2004)

2. Transform copies of the motif about the hyperbolic plane: *replication*

Poincaré Disk Model of Hyperbolic Geometry
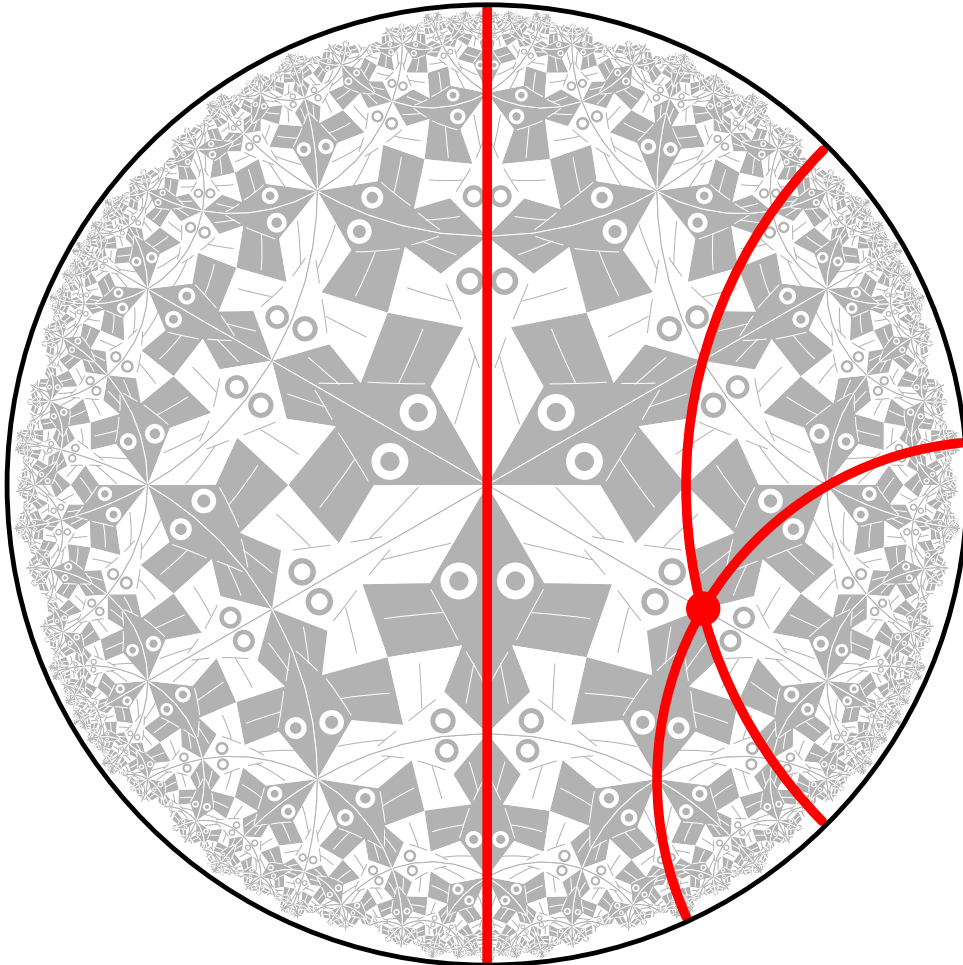
Repeating Patterns

A *repeating pattern* is composed of congruent copies of the motif.

The Regular Tessellations $\{p, q\}$

- **The *regular tessellation* $\{p, q\}$ is a tiling composed of regular p-sided polygons, or *p-gons* meeting q at each vertex.**

- **It is necessary that $(p-2)(q-2) > 4$ for the tessellation to be hyperbolic.**

- **If $(p-2)(q-2) = 4$ or $(p-2)(q-2) < 4$ the tessellation is Euclidean or spherical respectively.**

The Regular Tessellation $\{6, 4\}$
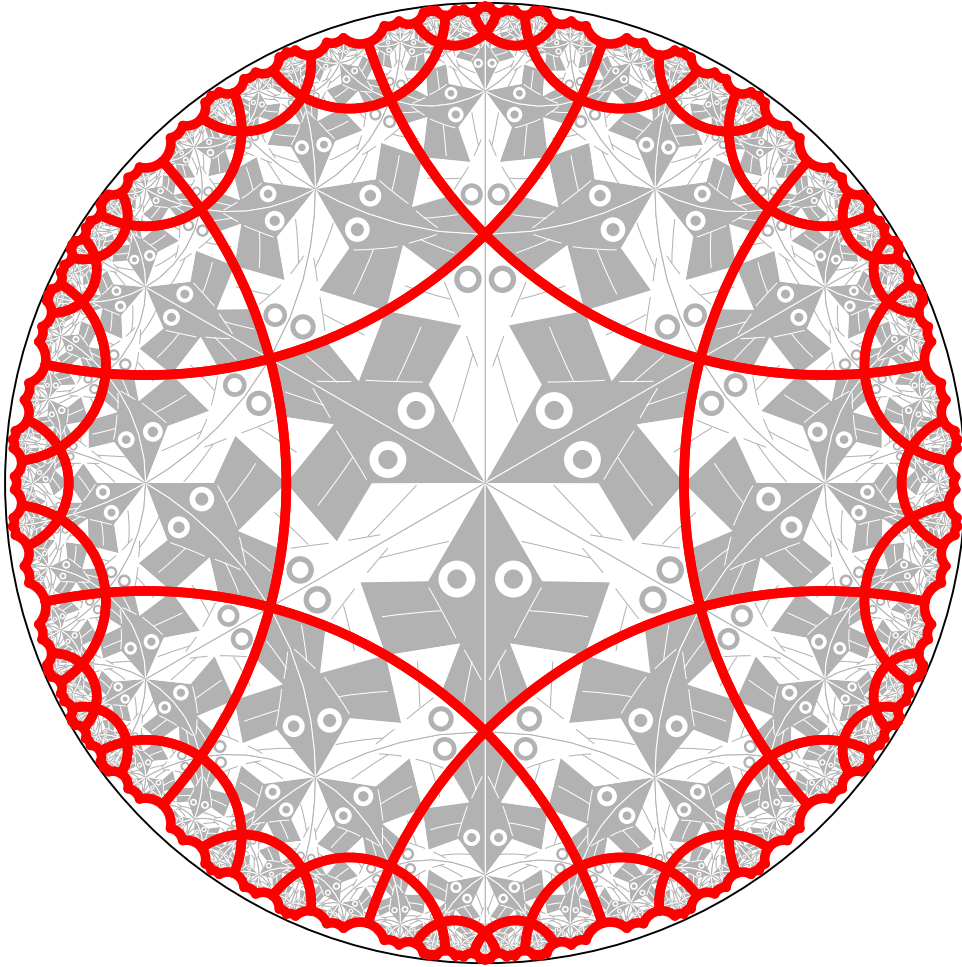
The Replication Algorithm

To reduce the number of transformations and to simplify the replication process, we form the *p-gon pattern* from all the copies of the motif touching the center of the bounding circle.

- Thus in order to replicate the pattern, we apply transformations to the p-gon pattern rather than to each individual motif.

- Some parts of the p-gon pattern may protrude from the enclosing p-gon, as long as there are corresponding indentations, so that the final pattern will fit together like a jigsaw puzzle.

- The p-gon pattern is often called the *translation unit* in repeating Euclidean patterns.

The p-gon pattern for *Circle Limit I*
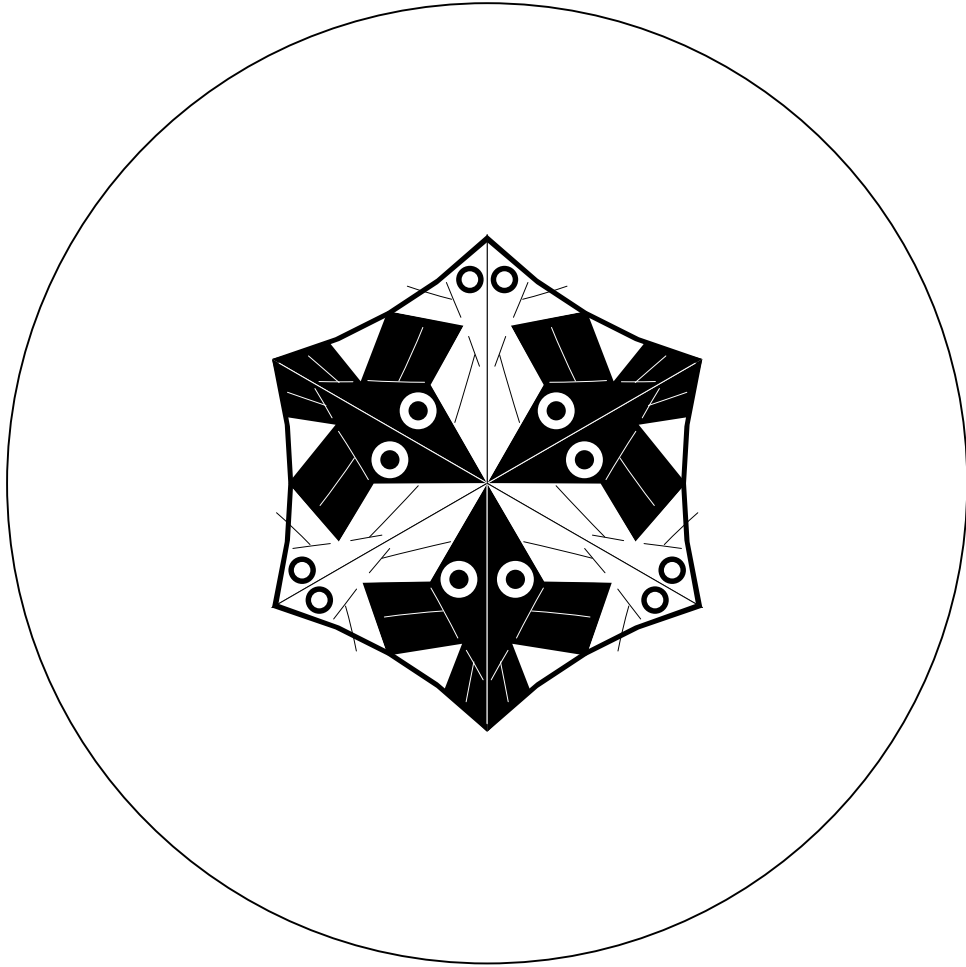
Layers of p-gons

We note that the p-gons of a $\{p, q\}$ tessellation are arranged in *layers* as follows:

- The first layer is just the central p-gon.

- The $k + 1^{st}$ layer consists of all p-gons sharing an edge or a vertex with a p-gon in the kth layer (and no previous layers).

- Theoretically a repeating hyperbolic pattern has an infinite number of layers, however if we only replicate a small number of layers, this is usually enough to appear to fill the bounding circle to our Euclidean eyes.

The Replication Algorithm

The replication algorithm consists of two parts:

1. **A top-level "driver" routine** `replicate()` **that draws the first layer, and calls a second routine,** `recursiveRep()`**, to draw the rest of the layers.**

2. **A routine** `recursiveRep()` **that recursively draws the rest of the desired number of layers.**

A tiling pattern is determined by how the p-gon pattern is transformed across p-gon edges. These transformations are in the array `edgeTran[]`

The Top-level Routine `replicate()`

```
Replicate ( motif ) {

  drawPgon ( motif, IDENT ) ;  // Draw central p-go

  for ( i = 1 to p ) {     // Iterate over each vert

    qTran = edgeTran[i-1]

    for ( j = 1 to q-2 ) {  // Iterate about a vert

      exposure = (j == 1) ? MIN_EXP : MAX_EXP ;
      recursiveRep ( motif, qTran, 2, exposure ) ;
      qTran = addToTran ( qTran, -1 ) ;

    }
  }
}
```

The Routine `recursiveRep()`

```
recursiveRep ( motif, initialTran, layer, exposure ) {

   drawPgon ( motif, initialTran ) ;   // Draw p-gon pattern

   if ( layer < maxLayer ) {            // If any more layers
      pShift = ( exposure == MIN_EXP ) ? 1 : 0 ;
      verticesToDo = ( exposure == MIN_EXP ) ? p-3 : p-2 ;

      for ( i = 1 to verticesToDo ) {  // Iterate over vertic
         pTran = computeTran ( initialTran, pShift ) ;
         qSkip = ( i == 1 ) ? -1 : 0 ;
         qTran = addToTran ( pTran, qSkip ) ;
         pgonsToDo = ( i == 1 ) ? q-3 : q-2 ;

         for ( j = 1 to pgonsToDo ) { // Iterate about a vert
            newExposure = ( i == 1 ) ? MIN_EXP : MAX_EXP ;
            recursiveRep(motif, qTran, layer+1, newExposure);
            qTran = addToTran ( qTran, -1 ) ;
         }
         pShift = (pShift + 1) % p ;  // Advance to next vert
      }
   }
}
```

Conclusions and Future Work

- **Enhance the program to draw patterns with more complex symmetries.**

- **Automate the generation of color symmetry.**