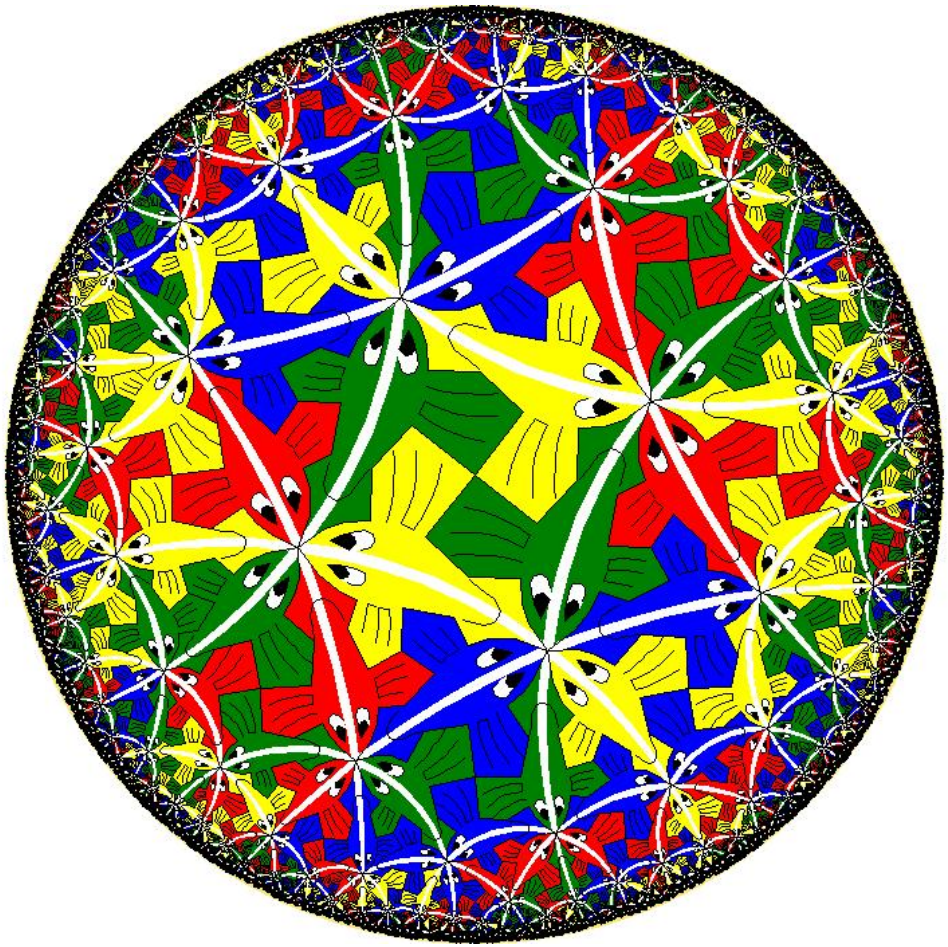
HyperArt – Generating Hyperbolic patterns for Regular and Non-Regular p-gons

Ajit Datar

Department of Computer Science
University of Minnesota, Duluth

Graduate Thesis Colloquium, 2005

A Hyper*Art* rendition of Circle Limit III



Outline

1. Motivation
   - Hyperbolic Patterns eh?
2. Theory
   - Lets Learn Geometry
   - Definitions Galore
3. Algorithms
   - Concepts
   - Regular tessellations
   - Non-regular tessellations
4. HyperArt
   - Features
   - Design

Not *Just* Pretty Pictures

- Tessellation is a covering of the plane with (symmetrical) patterns.
- Types of tessellations :
  - Euclidean – Penrose tiling.
  - Spherical – Temari balls
  - Hyperbolic – Escher's Circle Limit Patterns.
- Challenging and aesthetically rewarding!

From Desk to Desktop

- First hyperbolic patterns - M. C. Escher – *by hand!*.
- H. S. M. Coxeter – well known geometer – non–Euclidean geometry
- Dr. Douglas Dunham – computer algorithms and programs.

Goals

- Implement and refine *Regular* and *Non-regular* hyperbolic tessellation algorithms.
- To provide a unifying extensible programming framework for these algorithms.

Absolute Geometry

- A straight line segment can be drawn joining any two points.
- Any straight line segment can be extended indefinitely in a straight line.
- Given any straight line segment, a circle can be drawn having the segment as radius and one endpoint as center.
- All right angle are congruent.

Absolute Geometry

- A straight line segment can be drawn joining any two points.
- Any straight line segment can be extended indefinitely in a straight line.
- Given any straight line segment, a circle can be drawn having the segment as radius and one endpoint as center.
- All right angle are congruent.

Absolute Geometry

- A straight line segment can be drawn joining any two points.
- Any straight line segment can be extended indefinitely in a straight line.
- Given any straight line segment, a circle can be drawn having the segment as radius and one endpoint as center.
- All right angle are congruent.

Absolute Geometry

- A straight line segment can be drawn joining any two points.
- Any straight line segment can be extended indefinitely in a straight line.
- Given any straight line segment, a circle can be drawn having the segment as radius and one endpoint as center.
- All right angle are congruent.

Absolute Geometry

- A straight line segment can be drawn joining any two points.
- Any straight line segment can be extended indefinitely in a straight line.
- Given any straight line segment, a circle can be drawn having the segment as radius and one endpoint as center.
- All right angle are congruent.

The parallel postulate

Euclidean Parallel postulate

If two lines are drawn which intersect a third 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 far enough.

Types of geometries

Reinterpret the Euclidean parallel postulate as . . .

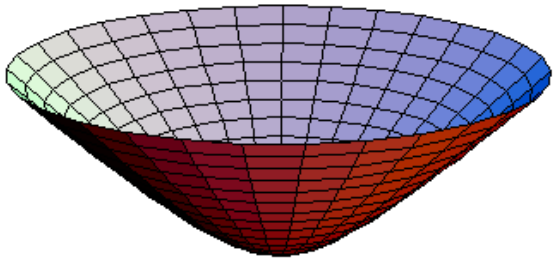
Elliptical/Spherical Parallel postulate

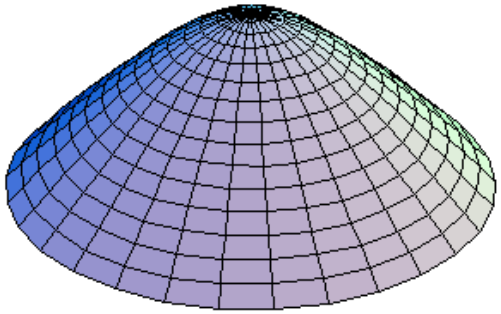
Through any point in the plane, there exists no lines parallel to the given line.

Hyperbolic parallel postulate

For any infinite straight line L and a point P not on it, there are many infinitely extending straight lines that pass through P and do not intersect L.

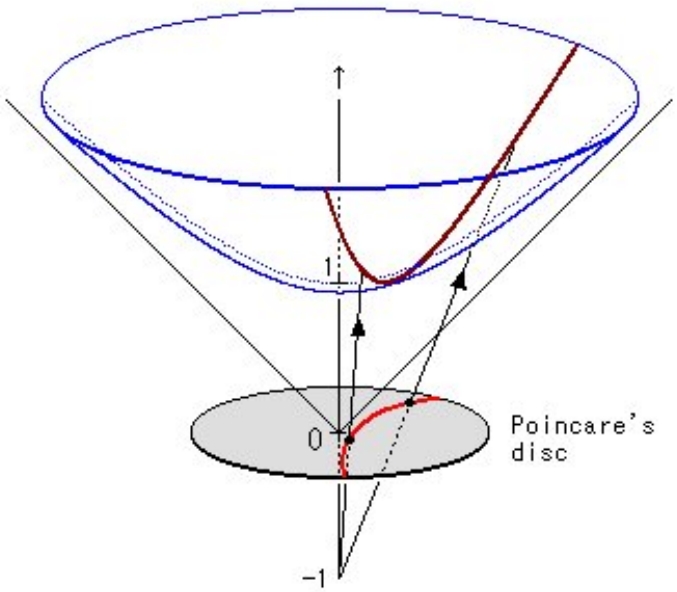
Weierstrass model





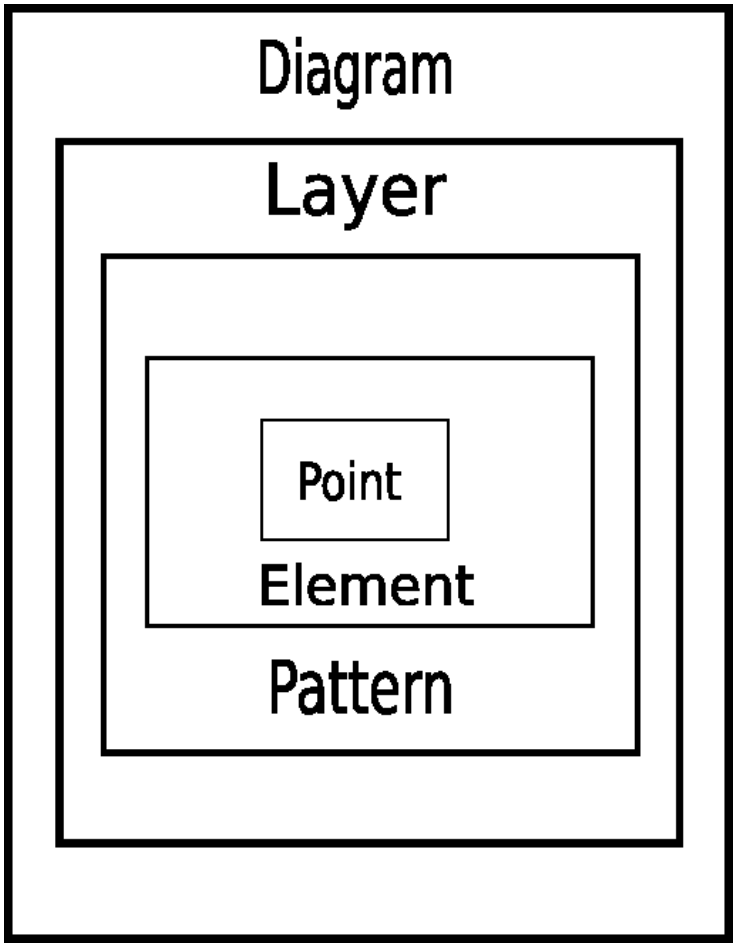
- Upper sheet of the hyperboloid of revolution $z = \sqrt{1 + x^2 + y^2}$.
- Hyperbolic lines are the arcs lying in this plane

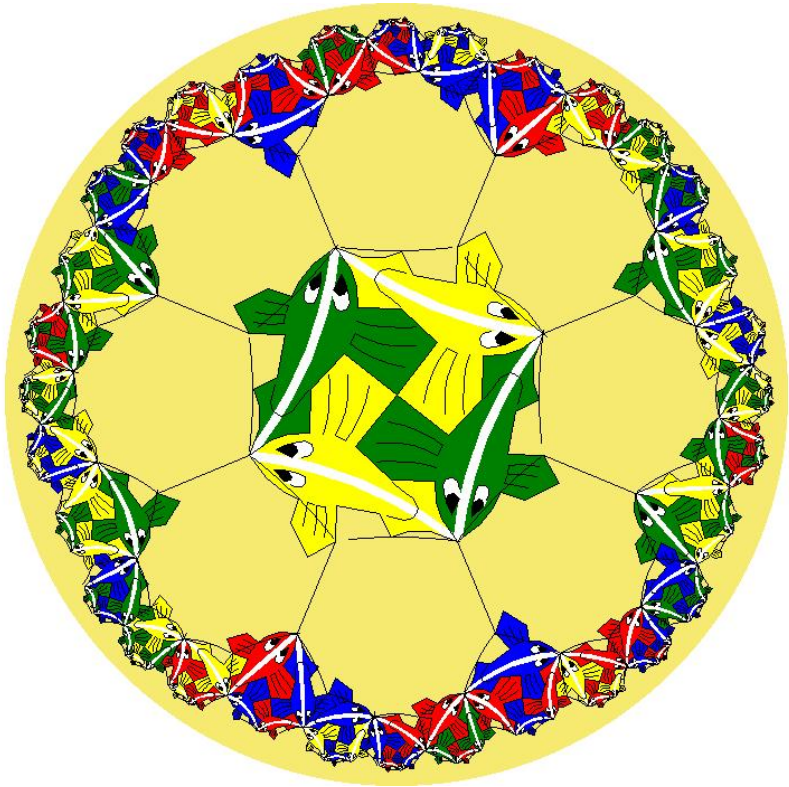
Poincaré model



- Projection of Weierstrass on a unit circle at origin towards $(0, -1, 0)$
- Hyperbolic lines are
  - arcs orthogonal to boundary
  - diameters of the disk
- "boundaryless" model

Parts of a Diagram





Central P-gon and Fundamental region

Definition

Central p-gon pattern is a pattern which remains invariant under certain transformations of the hyperbolic plane.

Definition

The fundamental region is a region in the hyperbolic plane which when transformed by all the transformations in the symmetry group, will cover the hyperbolic plane. *A Fundamental Pattern* is a pattern in the Fundamental region.

Central P-gon and Fundamental region

Definition

Central p-gon pattern is a pattern which remains invariant under certain transformations of the hyperbolic plane.

Definition

The fundamental region is a region in the hyperbolic plane which when transformed by all the transformations in the symmetry group, will cover the hyperbolic plane. *A Fundamental Pattern* is a pattern in the Fundamental region.

Symmetry groups

p is the number of sides/vertexes of the p-gon.

q is the number of p-gons meeting at a vertex.

- *Symmetry group* is a group of transformations of the hyperbolic plane that preserve a pattern.
- Infinitely many symmetry groups.
- eg Symmetry groups of Regular tessellations p,q
  - $[p, q]$ – 3 reflections
  - $[p, q]+$ – 3 rotations
  - $[p+, q]$ – p rotations about origin and a reflection across p-gon edge
  - $[p, q+]$ – q rotations about vertex and a reflection across p-gon edge

Symmetry groups

p is the number of sides/vertexes of the p-gon.
q is the number of p-gons meeting at a vertex.

- *Symmetry group* is a group of transformations of the hyperbolic plane that preserve a pattern.
- Infinitely many symmetry groups.
- eg Symmetry groups of Regular tessellations p,q
  - $[p, q]$ – 3 reflections
  - $[p, q]+$ – 3 rotations
  - $[p+, q]$ – p rotations about origin and a reflection across p-gon edge
  - $[p, q+]$ – q rotations about vertex and a reflection across p-gon edge

Symmetry groups

p is the number of sides/vertexes of the p-gon.
q is the number of p-gons meeting at a vertex.

- *Symmetry group* is a group of transformations of the hyperbolic plane that preserve a pattern.
- Infinitely many symmetry groups.
- eg Symmetry groups of Regular tessellations p,q
  - $[p, q]$ – 3 reflections
  - $[p, q]+$ – 3 rotations
  - $[p+, q]$ – p rotations about origin and a reflection across p-gon edge
  - $[p, q+]$ – q rotations about vertex and a reflection across p-gon edge

Classification of Algorithms

- According to p-gon type
  - Regular p-gon algorithms
  - Non-regular p-gon algorithms
- According to replication order
  - Hamiltonian methods.
  - Spanning-tree methods
- According to *central* p-gon
  - P-gon center at the origin
  - P-gon vertex at the origin

Classification of Algorithms

- According to p-gon type
  - Regular p-gon algorithms
  - Non-regular p-gon algorithms
- According to replication order
  - Hamiltonian methods.
  - Spanning-tree methods
- According to *central* p-gon
  - P-gon center at the origin
  - P-gon vertex at the origin

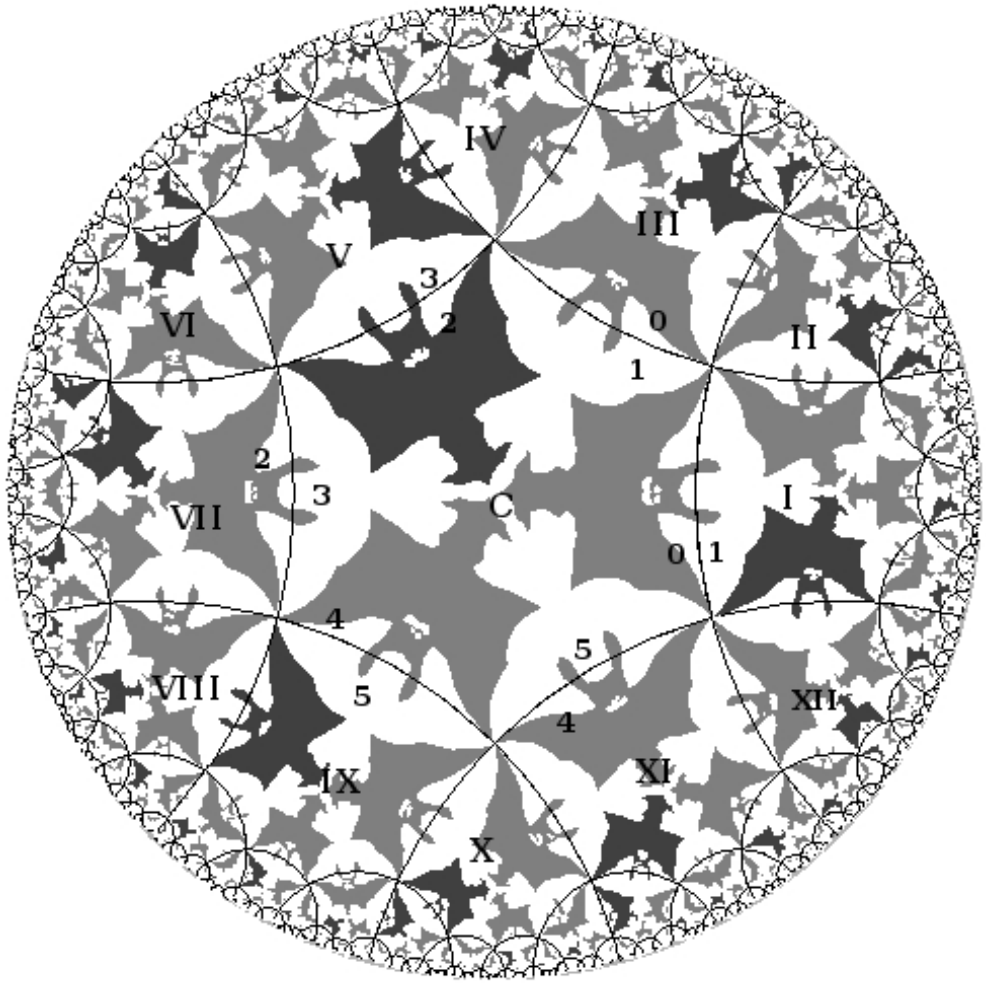
Classification of Algorithms

- According to p-gon type
  - Regular p-gon algorithms
  - Non-regular p-gon algorithms
- According to replication order
  - Hamiltonian methods.
  - Spanning-tree methods
- According to *central* p-gon
  - P-gon center at the origin
  - P-gon vertex at the origin

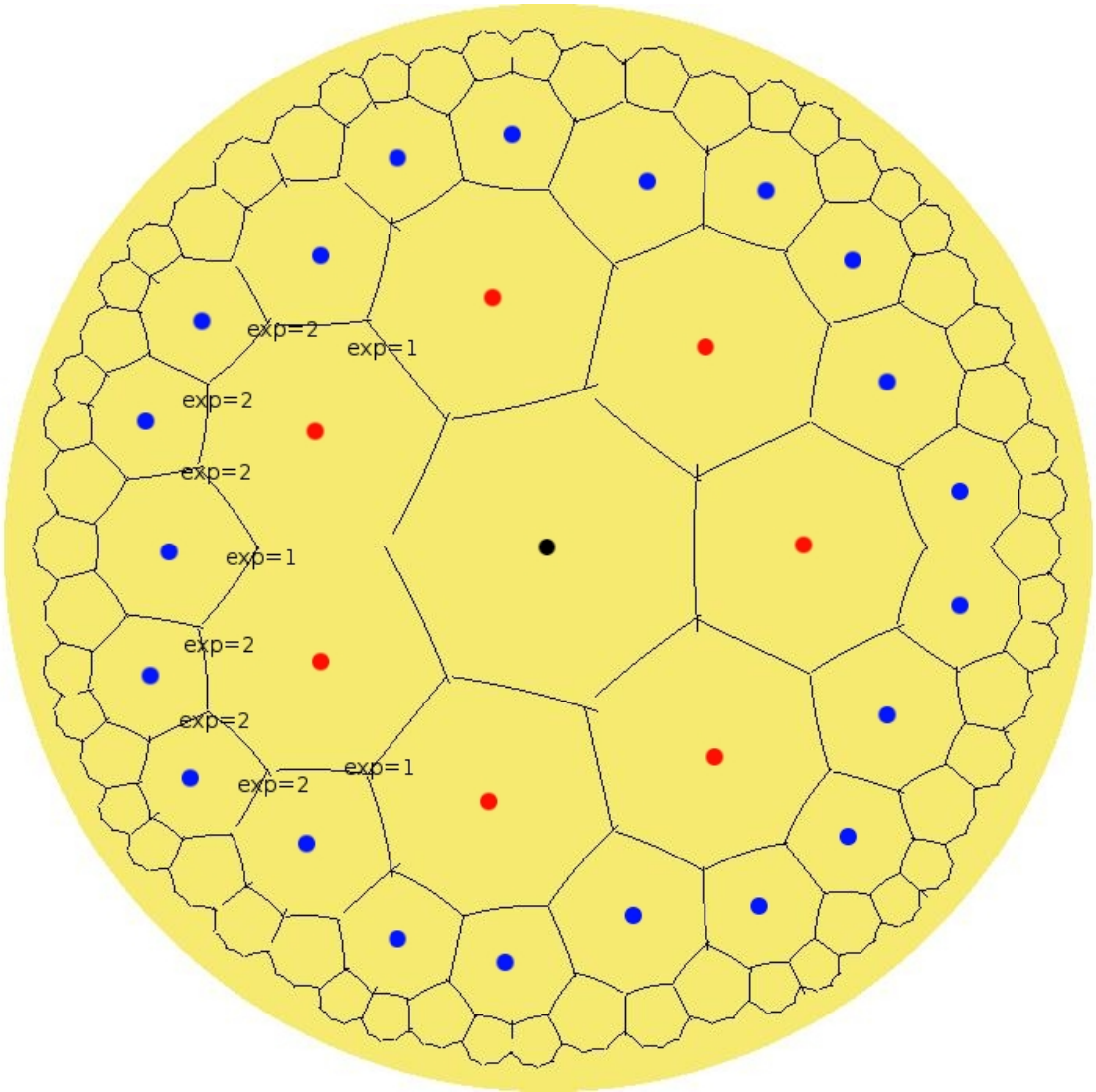
Classification of Algorithms

- According to p-gon type
  - Regular p-gon algorithms
  - Non-regular p-gon algorithms
- According to replication order
  - Hamiltonian methods.
  - Spanning-tree methods
- According to *central* p-gon
  - P-gon center at the origin
  - P-gon vertex at the origin

Edge Adjacency and Orientation



Exposure



Exposure of a vertex in the k^{th} layer : the number of p-gons in the $(k+1)^{st}$ layer sharing that vertex.

Generating the central p-gon

- Needed for Regular p-gon algorithms.
- Fundamental pattern needs to be replicated to fill up central p-gon
- which is then replicated to fill up the disk
- Transformations used :
  - Reflection across p-gon radius
  - Reflection across edge bisector
  - Rotation around p-gon center

Generating the central p-gon

- Needed for Regular p-gon algorithms.
- Fundamental pattern needs to be replicated to fill up central p-gon
- which is then replicated to fill up the disk
- Transformations used :
  - Reflection across p-gon radius
  - Reflection across edge bisector
  - Rotation around p-gon center

Generating the central p-gon

- Needed for Regular p-gon algorithms.
- Fundamental pattern needs to be replicated to fill up central p-gon
- which is then replicated to fill up the disk
- Transformations used :
  - Reflection across p-gon radius
  - Reflection across edge bisector
  - Rotation around p-gon center

Generating the central p-gon

- Needed for Regular p-gon algorithms.
- Fundamental pattern needs to be replicated to fill up central p-gon
- which is then replicated to fill up the disk
- Transformations used :
  - Reflection across p-gon radius
  - Reflection across edge bisector
  - Rotation around p-gon center

Generating the central p-gon

- Needed for Regular p-gon algorithms.
- Fundamental pattern needs to be replicated to fill up central p-gon
- which is then replicated to fill up the disk
- Transformations used :
  - Reflection across p-gon radius
  - Reflection across edge bisector
  - Rotation around p-gon center

Generating the central p-gon

- Needed for Regular p-gon algorithms.
- Fundamental pattern needs to be replicated to fill up central p-gon
- which is then replicated to fill up the disk
- Transformations used :
  - Reflection across p-gon radius
  - Reflection across edge bisector
  - Rotation around p-gon center

Generating the central p-gon

- Needed for Regular p-gon algorithms.
- Fundamental pattern needs to be replicated to fill up central p-gon
- which is then replicated to fill up the disk
- Transformations used :
  - Reflection across p-gon radius
  - Reflection across edge bisector
  - Rotation around p-gon center

Regular P-gon Pattern Replication

- Spanning tree method of replication.
- The idea is to
  - Start with the central p-gon
  - Visit each shared vertex.
  - And recursively process the optimal number of p-gons around it.

Lets look at an example in HyperArt

Regular P-gon Pattern Replication

- Spanning tree method of replication.
- The idea is to
  - Start with the central p-gon
  - Visit each shared vertex.
  - And recursively process the optimal number of p-gons around it.

Lets look at an example in HyperArt

Regular P-gon Pattern Replication

- Spanning tree method of replication.
- The idea is to
  - Start with the central p-gon
  - Visit each shared vertex.
  - And recursively process the optimal number of p-gons around it.

Lets look at an example in HyperArt

Regular P-gon Pattern Replication

- Spanning tree method of replication.
- The idea is to
  - Start with the central p-gon
  - Visit each shared vertex.
  - And recursively process the optimal number of p-gons around it.

Lets look at an example in HyperArt

Regular P-gon Pattern Replication

- Spanning tree method of replication.
- The idea is to
  - Start with the central p-gon
  - Visit each shared vertex.
  - And recursively process the optimal number of p-gons around it.

Lets look at an example in HyperArt

Regular P-gon Pattern Replication

- Spanning tree method of replication.
- The idea is to
  - Start with the central p-gon
  - Visit each shared vertex.
  - And recursively process the optimal number of p-gons around it.

Lets look at an example in HyperArt

Non-Regular P-gon Pattern Replication

- P-gon *is* the Fundamental region.
- But now we have different number of p-gons meeting at each vertex.
- Fundamental region moved to put vertex at center.
- ∴ the concept of layers is slightly changed.
- Except for some implementation differences, replication is then same as the Regular P-gon case.

Lets look at an example in HyperArt

Non-Regular P-gon Pattern Replication

- P-gon *is* the Fundamental region.
- But now we have different number of p-gons meeting at each vertex.
- Fundamental region moved to put vertex at center.
- ∴ the concept of layers is slightly changed.
- Except for some implementation differences, replication is then same as the Regular P-gon case.

Lets look at an example in HyperArt

Non-Regular P-gon Pattern Replication

- P-gon *is* the Fundamental region.
- But now we have different number of p-gons meeting at each vertex.
- Fundamental region moved to put vertex at center.
- ∴ the concept of layers is slightly changed.
- Except for some implementation differences, replication is then same as the Regular P-gon case.

Lets look at an example in HyperArt

Non-Regular P-gon Pattern Replication

- P-gon *is* the Fundamental region.
- But now we have different number of p-gons meeting at each vertex.
- Fundamental region moved to put vertex at center.
- ∴ the concept of layers is slightly changed.
- Except for some implementation differences, replication is then same as the Regular P-gon case.

Lets look at an example in HyperArt

Non-Regular P-gon Pattern Replication

- P-gon *is* the Fundamental region.
- But now we have different number of p-gons meeting at each vertex.
- Fundamental region moved to put vertex at center.
- ∴ the concept of layers is slightly changed.
- Except for some implementation differences, replication is then same as the Regular P-gon case.

Lets look at an example in HyperArt

Non-Regular P-gon Pattern Replication

- P-gon *is* the Fundamental region.
- But now we have different number of p-gons meeting at each vertex.
- Fundamental region moved to put vertex at center.
- ∴ the concept of layers is slightly changed.
- Except for some implementation differences, replication is then same as the Regular P-gon case.

Lets look at an example in HyperArt

Non-Regular P-gon Pattern Replication

- P-gon *is* the Fundamental region.
- But now we have different number of p-gons meeting at each vertex.
- Fundamental region moved to put vertex at center.
- ∴ the concept of layers is slightly changed.
- Except for some implementation differences, replication is then same as the Regular P-gon case.

Lets look at an example in Hyper*Art*

Hyper*Art* Features From User's Perspective

- XML data file format
- Crossplatform – Written in Qt from trolltech
- Single unified interface for different types of algorithms
- Layer and Frame toggling
- Algorithm animation and stepping
- Modern UI with zooming, panning and printing support.
- Diagram export to popular image formats.

Hyper*Art* Features From User's Perspective

- XML data file format
- Crossplatform – Written in Qt from trolltech
- Single unified interface for different types of algorithms
- Layer and Frame toggling
- Algorithm animation and stepping
- Modern UI with zooming, panning and printing support.
- Diagram export to popular image formats.

Hyper*Art* Features From User's Perspective

- XML data file format
- Crossplatform – Written in Qt from trolltech
- Single unified interface for different types of algorithms
- Layer and Frame toggling
- Algorithm animation and stepping
- Modern UI with zooming, panning and printing support.
- Diagram export to popular image formats.

Hyper*Art* Features From User's Perspective

- XML data file format
- Crossplatform – Written in Qt from trolltech
- Single unified interface for different types of algorithms
- Layer and Frame toggling
- Algorithm animation and stepping
- Modern UI with zooming, panning and printing support.
- Diagram export to popular image formats.

Hyper*Art* Features From User's Perspective

- XML data file format
- Crossplatform – Written in Qt from trolltech
- Single unified interface for different types of algorithms
- Layer and Frame toggling
- Algorithm animation and stepping
- Modern UI with zooming, panning and printing support.
- Diagram export to popular image formats.

Hyper*Art* Features From User's Perspective

- XML data file format
- Crossplatform – Written in Qt from trolltech
- Single unified interface for different types of algorithms
- Layer and Frame toggling
- Algorithm animation and stepping
- Modern UI with zooming, panning and printing support.
- Diagram export to popular image formats.

Hyper*Art* Features From User's Perspective

- XML data file format
- Crossplatform – Written in Qt from trolltech
- Single unified interface for different types of algorithms
- Layer and Frame toggling
- Algorithm animation and stepping
- Modern UI with zooming, panning and printing support.
- Diagram export to popular image formats.

Hyper*Art* Features From User's Perspective

- XML data file format
- Crossplatform – Written in Qt from trolltech
- Single unified interface for different types of algorithms
- Layer and Frame toggling
- Algorithm animation and stepping
- Modern UI with zooming, panning and printing support.
- Diagram export to popular image formats.

Hyper*Art* Features From Researcher's Perspective

- Document view architecture.
- Abstraction of Diagram and View classes.
- Importing old dat files to the new XML format.
- Element based patterns instead of pen based.
- Source-code available under GPL license on sourceforge.net.

Hyper*Art* Features From Researcher's Perspective

- Document view architecture.
- Abstraction of Diagram and View classes.
- Importing old dat files to the new XML format.
- Element based patterns instead of pen based.
- Source-code available under GPL license on sourceforge.net.

Hyper*Art* Features From Researcher's Perspective

- Document view architecture.
- Abstraction of Diagram and View classes.
- Importing old dat files to the new XML format.
- Element based patterns instead of pen based.
- Source-code available under GPL license on sourceforge.net.

Hyper*Art* Features From Researcher's Perspective

- Document view architecture.
- Abstraction of Diagram and View classes.
- Importing old dat files to the new XML format.
- Element based patterns instead of pen based.
- Source-code available under GPL license on sourceforge.net.

Hyper*Art* Features From Researcher's Perspective

- Document view architecture.
- Abstraction of Diagram and View classes.
- Importing old dat files to the new XML format.
- Element based patterns instead of pen based.
- Source-code available under GPL license on sourceforge.net.

Design of HyperArt

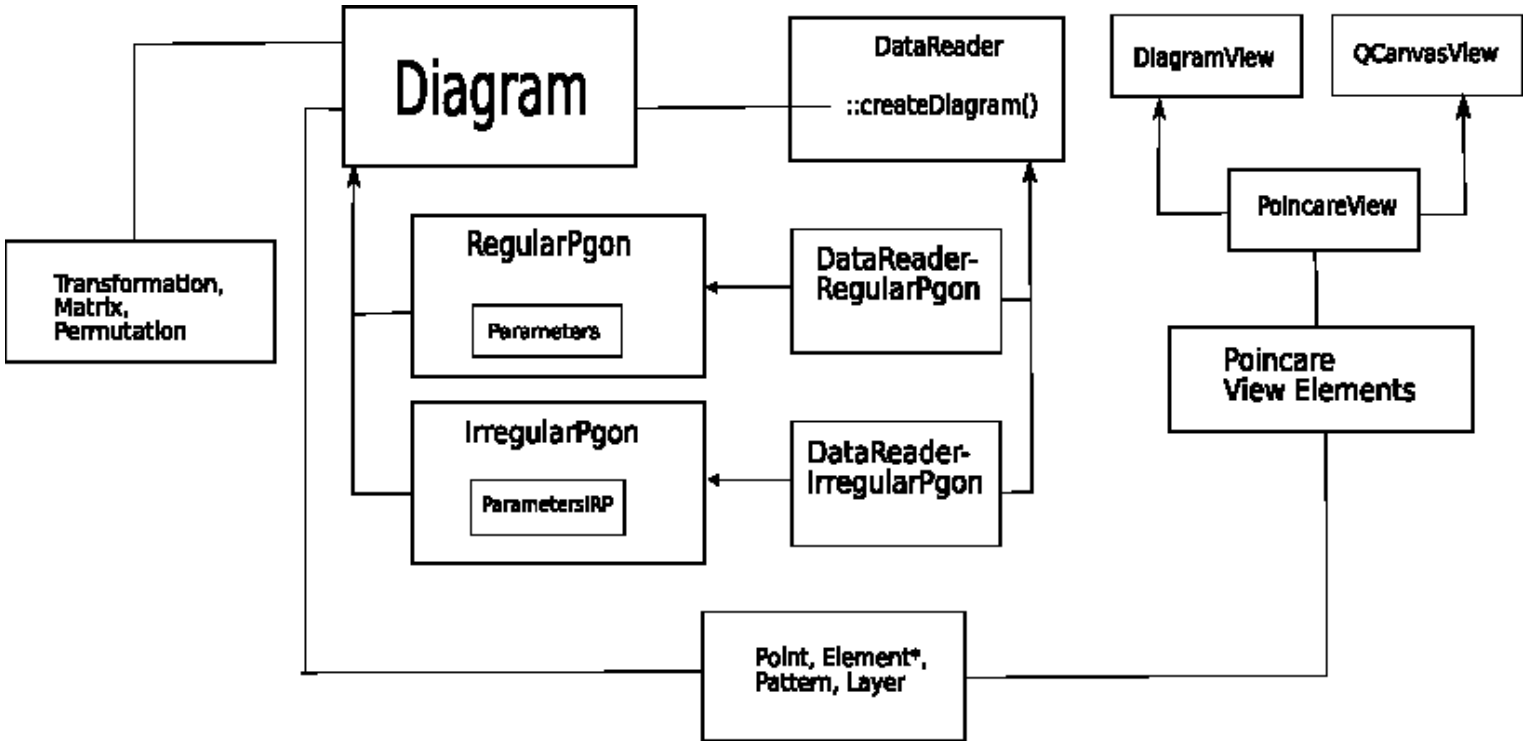


Diagram classes

- Diagram – abstract base class
  - RegularPgon
  - IrregularPgon
- Parameters – For RegularPgon
  - ParametersIRP – For IrregularPgon
- Element
  - Circle
  - EuclidPoly, EuclidPolyLine
  - HyperPoly, HyperPolyLine
- Other utility classes such as Matrix, Permutation, Transformation

Diagram classes

- Diagram – abstract base class
  - RegularPgon
  - IrregularPgon
- Parameters – For RegularPgon
  - ParametersIRP – For IrregularPgon
- Element
  - Circle
  - EuclidPoly, EuclidPolyLine
  - HyperPoly, HyperPolyLine
- Other utility classes such as Matrix, Permutation, Transformation

Diagram classes

- Diagram – abstract base class
  - RegularPgon
  - IrregularPgon
- Parameters – For RegularPgon
  - ParametersIRP – For IrregularPgon
- Element
  - Circle
  - EuclidPoly, EuclidPolyLine
  - HyperPoly, HyperPolyLine
- Other utility classes such as Matrix, Permutation, Transformation

Diagram classes

- Diagram – abstract base class
  - RegularPgon
  - IrregularPgon
- Parameters – For RegularPgon
  - ParametersIRP – For IrregularPgon
- Element
  - Circle
  - EuclidPoly, EuclidPolyLine
  - HyperPoly, HyperPolyLine
- Other utility classes such as Matrix, Permutation, Transformation

Presentation classes

- DiagramView – abstract base view class
  - PoincareView – derives from DiagramView and QCanvasView
- Canvas counterparts of Element classes such as CanvasHyperPoly

Presentation classes

- DiagramView – abstract base view class
  - PoincareView – derives from DiagramView and QCanvasView
- Canvas counterparts of Element classes such as CanvasHyperPoly

Reader classes

- DataReader – abstract base class for readers, also a Diagram Factory
  - DataReaderRegularPgon
  - DataReaderIrregularPgon

Summary

- Provided the unifying and extensible HyperArt framework
- Implemented and refined Regular and Non-Regular P-gon algorithms

- Future Work
  - Diagram Designer for HyperArt
  - Implement other algorithms.

Summary

- Provided the unifying and extensible HyperArt framework
- Implemented and refined Regular and Non-Regular P-gon algorithms

- Future Work
  - Diagram Designer for HyperArt
  - Implement other algorithms.

Lets see some nice designs . . .

For Further Reading I

HSM. Coxeter

Non-Euclidean Geometry.

Mathematical Association of America, 1988, ISBN 0-88385- 522-4.

Goldennumber.net

Penrose Tiling

http://goldennumber.net/penrose.htm, 1997-2005.

Diana Vandervoort

Temari balls exhibition.

http://temari.com, 2003.

For Further Reading II

Wikipedia

Hyperboloid of 2 sheets

http://en.wikipedia.org/wiki/Image:HyperboloidOfTwoSheets

2004.

D. Dunham

Creating Repeating Hyperbolic Patterns

Computer Graphics, 1981, Volume 15, Number 3.

A. Datar

Hyperart webpage.

http://hyperart.sourceforge.net, 2005.