Exploring Visualizations

An Overview of a Seminar in 3D Modeling and Printing

Nicholas J. Owad

University Of Nebraska – Lincoln

Summer 2015

nowad2@math.unl.edu

nick.owad.org

3D Modeling Workshop

During the 2014-15 Year, I ran a workshop meant to introduce faculty, grads, and undergrads to 3D modeling and 3D printing.

3D Modeling Workshop

During the 2014-15 Year, I ran a workshop meant to introduce faculty, grads, and undergrads to 3D modeling and 3D printing.

- The Goal of the Workshop: To Make the participants able to design their own ideas and print them.

3D Modeling Workshop

During the 2014-15 Year, I ran a workshop meant to introduce faculty, grads, and undergrads to 3D modeling and 3D printing.

- The Goal of the Workshop: To Make the participants able to design their own ideas and print them.
- One session, 1-2 hours a week

3D Modeling Workshop

During the 2014-15 Year, I ran a workshop meant to introduce faculty, grads, and undergrads to 3D modeling and 3D printing.

- The Goal of the Workshop: To Make the participants able to design their own ideas and print them.
- One session, 1-2 hours a week
- Ran in a workshop manner - Not lecture

3D Modeling Workshop

During the 2014-15 Year, I ran a workshop meant to introduce faculty, grads, and undergrads to 3D modeling and 3D printing.

- The Goal of the Workshop: To Make the participants able to design their own ideas and print them.
- One session, 1-2 hours a week
- Ran in a workshop manner - Not lecture
- Projects (ideally) take a single workshop, but many went for longer

3D Modeling Workshop

During the 2014-15 Year, I ran a workshop meant to introduce faculty, grads, and undergrads to 3D modeling and 3D printing.

- The Goal of the Workshop: To Make the participants able to design their own ideas and print them.
- One session, 1-2 hours a week
- Ran in a workshop manner - Not lecture
- Projects (ideally) take a single workshop, but many went for longer
- Used Rhino 5 for Windows (30 license school lab $975)

3D Modeling Workshop

During the 2014-15 Year, I ran a workshop meant to introduce faculty, grads, and undergrads to 3D modeling and 3D printing.

- The Goal of the Workshop: To Make the participants able to design their own ideas and print them.
- One session, 1-2 hours a week
- Ran in a workshop manner - Not lecture
- Projects (ideally) take a single workshop, but many went for longer
- Used Rhino 5 for Windows (30 license school lab $975)
- Local company, owned by an alumnus, let us use his Makerbot Replicator 2 at cost of material

This Talk's Goals

The Goal of this talk is to give you an outline for a workshop you want to run.

This Talk's Goals

The Goal of this talk is to give you an outline for a workshop you want to run.

Or to let you get some ideas of things you want to make yourself.

This Talk's Goals

The Goal of this talk is to give you an outline for a workshop you want to run.

Or to let you get some ideas of things you want to make yourself.

Let everyone realize how useful customizable 3D models can be for a mathematician

This Talk's Goals

The Goal of this talk is to give you an outline for a workshop you want to run.

Or to let you get some ideas of things you want to make yourself.

Let everyone realize how useful customizable 3D models can be for a mathematician in research or teaching.

Basics
Point and Click design

The beginning of the workshop focused on users becoming familiar with the GUI and basic creation tools available to them.

Basics
Point and Click design

The beginning of the workshop focused on users becoming familiar with the GUI and basic creation tools available to them.

- Understanding how the viewports work and rotating/panning them

Basics
Point and Click design

The beginning of the workshop focused on users becoming familiar with the GUI and basic creation tools available to them.

- Understanding how the viewports work and rotating/panning them
- Placing objects: Points, lines, etc

Basics
Point and Click design

The beginning of the workshop focused on users becoming familiar with the GUI and basic creation tools available to them.

- Understanding how the viewports work and rotating/panning them
- Placing objects: Points, lines, etc
- Working in a digital 3D environment

Basics
Point and Click design

The beginning of the workshop focused on users becoming familiar with the GUI and basic creation tools available to them.

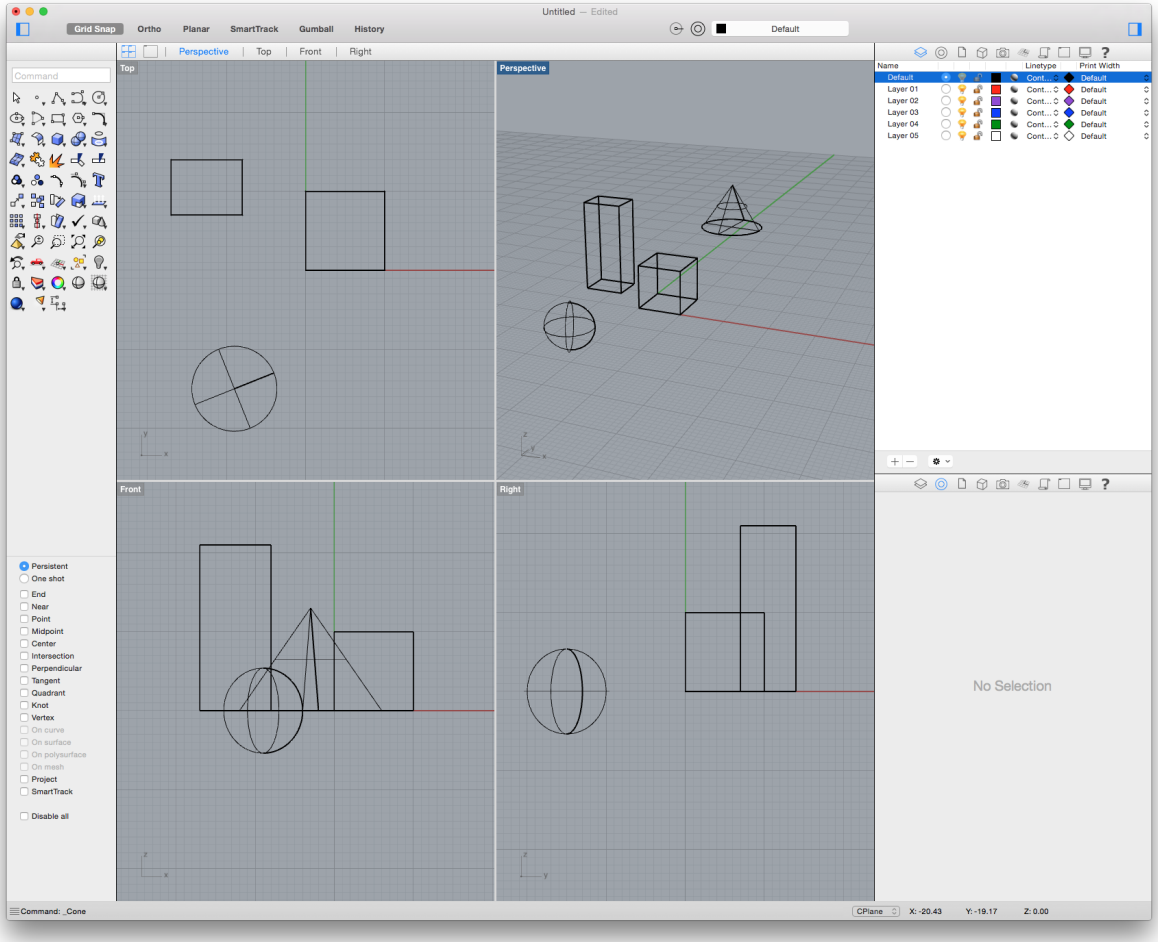
- Understanding how the viewports work and rotating/panning them
- Placing objects: Points, lines, etc
- Working in a digital 3D environment
- This is a long process - About half a semester

Basics
Point and Click design

The beginning of the workshop focused on users becoming familiar with the GUI and basic creation tools available to them.

- Understanding how the viewports work and rotating/panning them
- Placing objects: Points, lines, etc
- Working in a digital 3D environment
- This is a long process - About half a semester
- The following projects were designed to make the user comfortable in this new world

GUI



Castle - Day 1

The first project was not meant to be printed. Commands they learn:

Castle - Day 1

The first project was not meant to be printed. Commands they learn:

- Create cubes, spheres, cones, cylinders, etc

Castle - Day 1

The first project was not meant to be printed. Commands they learn:

- Create cubes, spheres, cones, cylinders, etc
- Copy and Paste

Castle - Day 1

The first project was not meant to be printed. Commands they learn:

- Create cubes, spheres, cones, cylinders, etc
- Copy and Paste
- Move

Castle - Day 1

The first project was not meant to be printed. Commands they learn:

- Create cubes, spheres, cones, cylinders, etc
- Copy and Paste
- Move

Directions: Build a *SWEET* Castle with the shapes you can now create.

Castle - Day 1

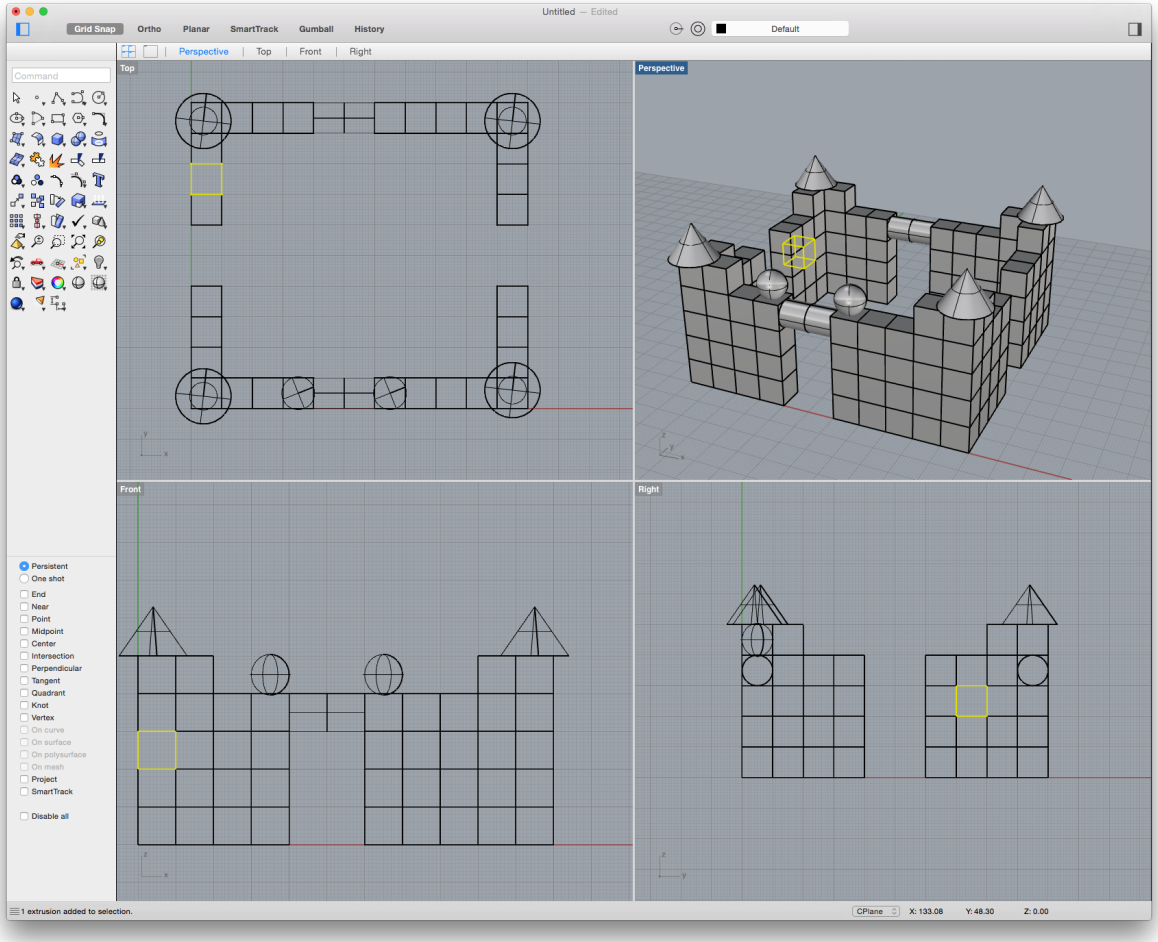
The first project was not meant to be printed. Commands they learn:

- Create cubes, spheres, cones, cylinders, etc
- Copy and Paste
- Move

Directions: Build a *SWEET* Castle with the shapes you can now create.

Main skill they acquire: Intuition about 3D space they are working in

Castle - Day 1



Square Circle Triangle Game - Day 2

Using all 3 dimensions

Commands they learn:

Square Circle Triangle Game - Day 2

Using all 3 dimensions

Commands they learn:

- Boolean operations: Difference, Intersection, Union

Square Circle Triangle Game - Day 2

Using all 3 dimensions

Commands they learn:

- Boolean operations: Difference, Intersection, Union
- Rotate

Square Circle Triangle Game - Day 2

Using all 3 dimensions

Commands they learn:

- Boolean operations: Difference, Intersection, Union
- Rotate

Direction 1: Build a rectangluar prism that has a square hole, circle hole, and triangle hole (all the "same" size).

Square Circle Triangle Game - Day 2

Using all 3 dimensions

Commands they learn:

- Boolean operations: Difference, Intersection, Union
- Rotate

Direction 1: Build a rectangluar prism that has a square hole, circle hole, and triangle hole (all the "same" size).

Direction 2: Build a single object that can pass through each hole and fill it completely. (Hint: This is possible.)

Square Circle Triangle Game - Day 2

Using all 3 dimensions

Part two of this project:

Square Circle Triangle Game - Day 2

Using all 3 dimensions

Part two of this project:

Directions: Pick 3 words of the same length, n, and "do the same thing." That is, create n blocks which have 3 letters on each block so that from the 3 directions they are all visible.

Square Circle Triangle Game - Day 2

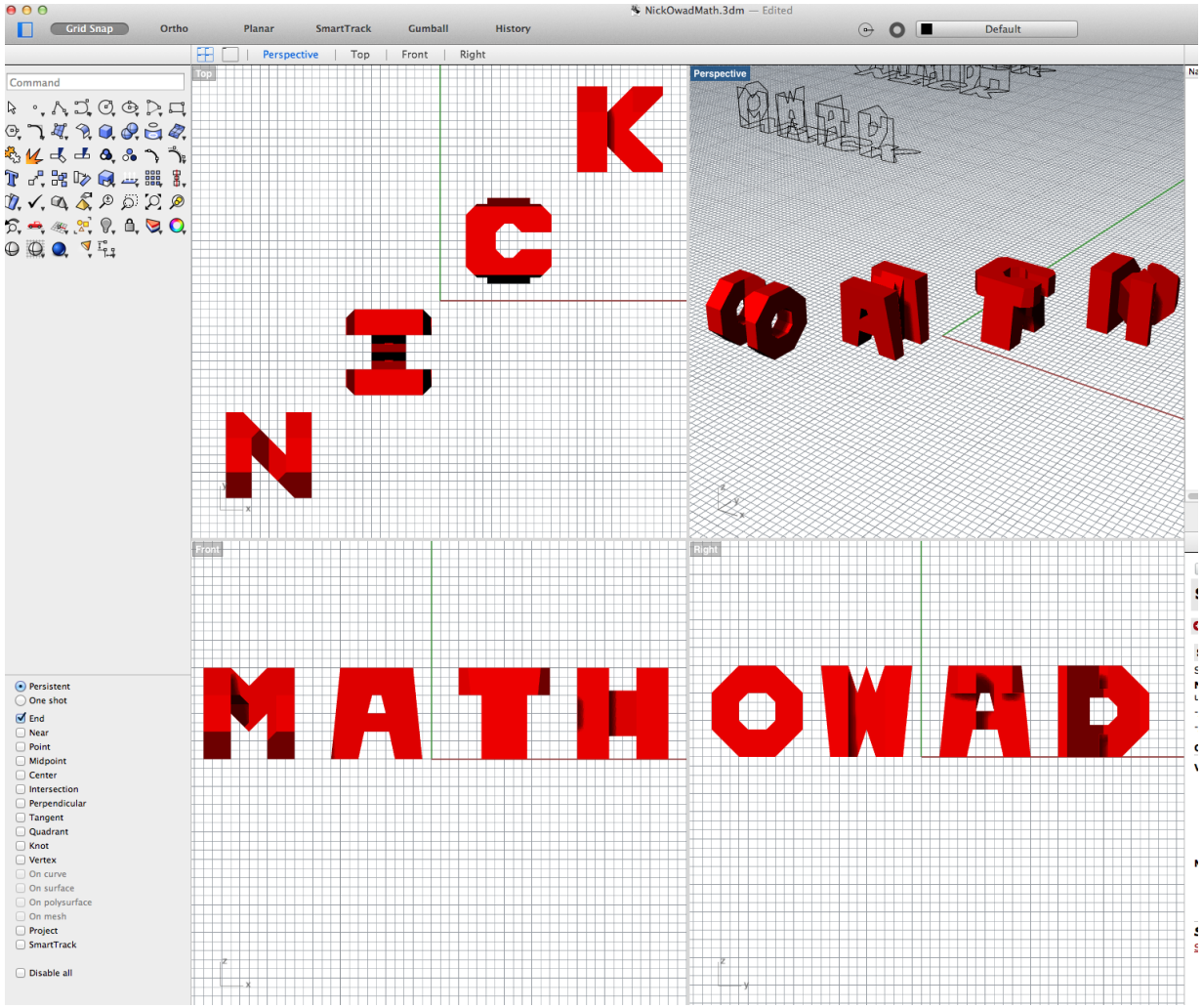
Using all 3 dimensions

Part two of this project:

Directions: Pick 3 words of the same length, n, and "do the same thing." That is, create n blocks which have 3 letters on each block so that from the 3 directions they are all visible.

Failure - Much to complicated for the second project.

Square Circle Triangle Game - Day 2



Extrusions - Replacement Day 2

What to do instead:

Extrusions - Replacement Day 2

What to do instead:

Extrustions!

Extrusions - Replacement Day 2

What to do instead:

Extrustions!

Commands they learn:

- Placing bitmaps

Extrusions - Replacement Day 2

What to do instead:

Extrustions!

Commands they learn:

- Placing bitmaps
- Extrude curve

Extrusions - Replacement Day 2

What to do instead:

Extrustions!

Commands they learn:

- Placing bitmaps
- Extrude curve
- Trim

Extrusions - Replacement Day 2

What to do instead:

Extrustions!

Commands they learn:

- Placing bitmaps
- Extrude curve
- Trim
- Join

Extrusions - Replacement Day 2

What to do instead:

Extrustions!

Commands they learn:

- Placing bitmaps
- Extrude curve
- Trim
- Join

Directions: Come to class with a picture (Symbol, Emblem, etc.) Draw the outline with interpolated curves.

Extrusions - Replacement Day 2

What to do instead:

Extrustions!

Commands they learn:

- Placing bitmaps
- Extrude curve
- Trim
- Join

Directions: Come to class with a picture (Symbol, Emblem, etc.) Draw the outline with interpolated curves.

MUST BE A SINGLE CLOSED CURVE!

Extrusions - Replacement Day 2

What to do instead:

Extrustions!

Commands they learn:

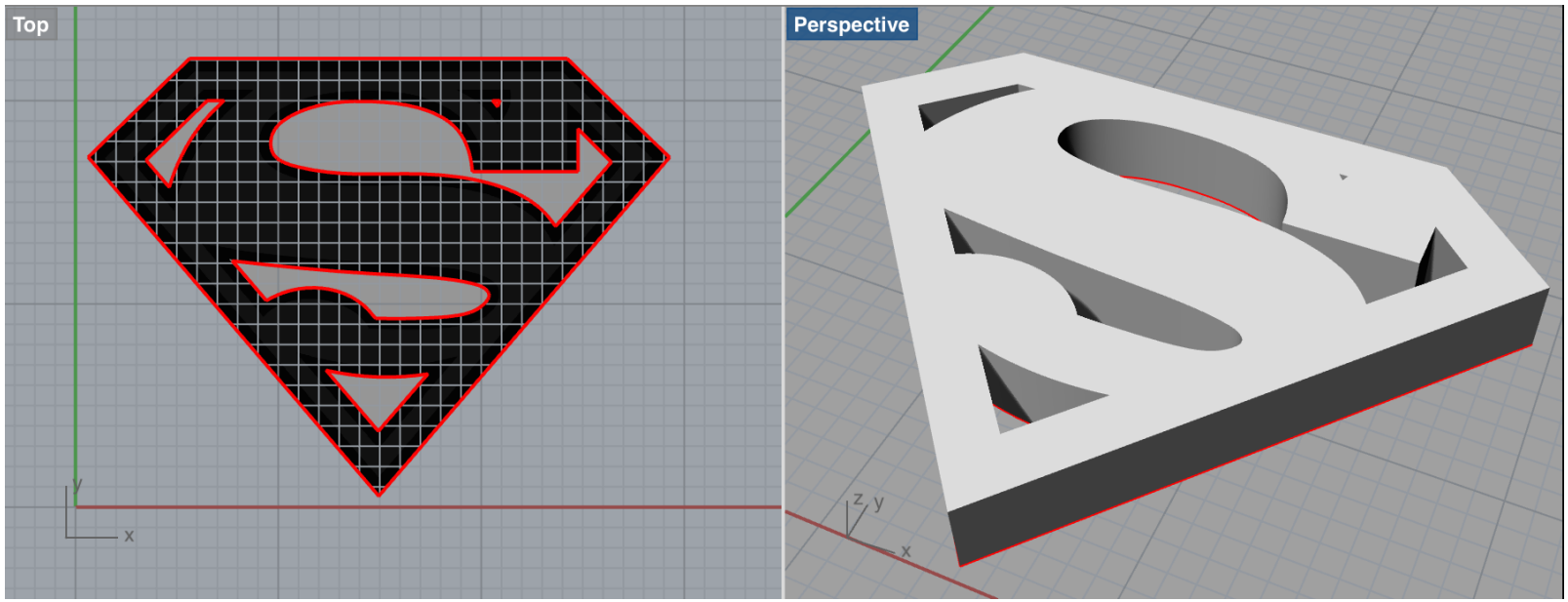
- Placing bitmaps
- Extrude curve
- Trim
- Join

Directions: Come to class with a picture (Symbol, Emblem, etc.) Draw the outline with interpolated curves.

MUST BE A SINGLE CLOSED CURVE!

Extrude it.

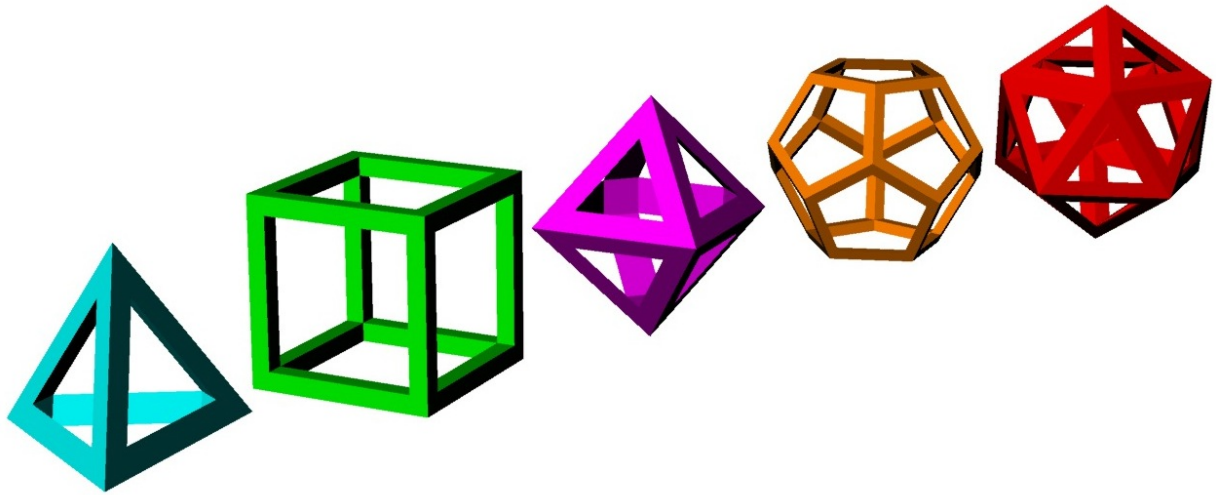
Extrusions - Replacement Day 2



Platonic Solids

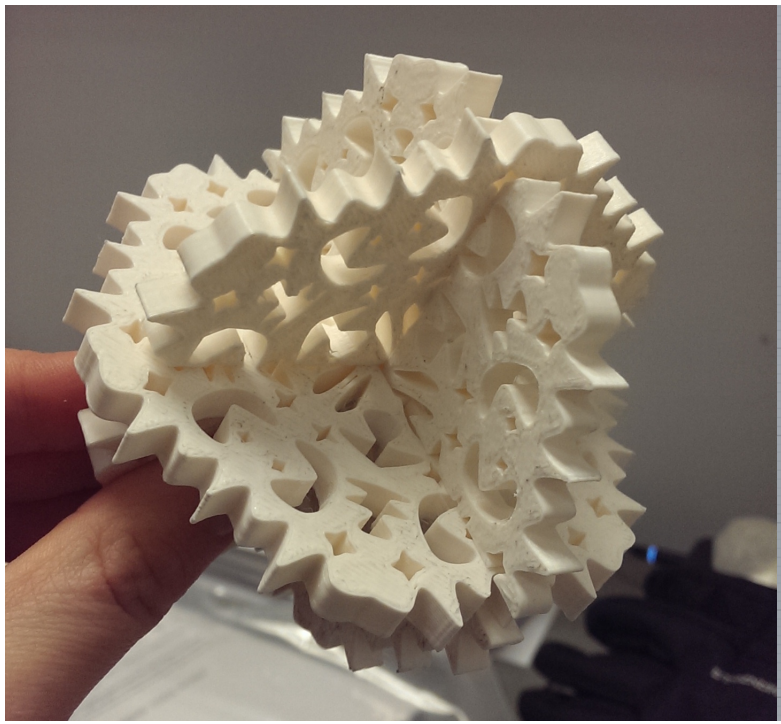
Platonic Solids

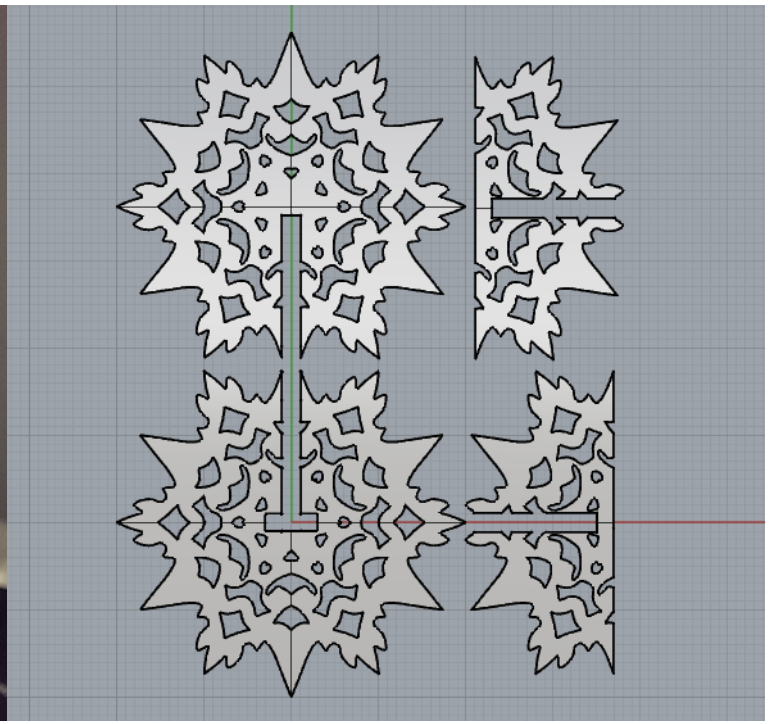
- Commands: Array (polar), 3D rotate, Osnap
- Lots of ways to actually build the models: from faces, vertices, etc
- Spent a month just on these 5 shapes



Seasonal

Pumpkins and Snowflakes





Modeling with Python

Disclaimer: I am a novice programmer.

Modeling with Python

Disclaimer: I am a novice programmer. But I managed to get some nice results with a little help:

Modeling with Python

Disclaimer: I am a novice programmer. But I managed to get some nice results with a little help:

- Henry Segerman

Modeling with Python

Disclaimer: I am a novice programmer. But I managed to get some nice results with a little help:

- Henry Segerman
- Google: Python Rhino tutorials

Modeling with Python

Disclaimer: I am a novice programmer. But I managed to get some nice results with a little help:

- Henry Segerman
- Google: Python Rhino tutorials
  One I used: ▸ vimeo.com/28619851

Modeling with Python

Disclaimer: I am a novice programmer. But I managed to get some nice results with a little help:

- Henry Segerman
- Google: Python Rhino tutorials
  One I used: ▸ vimeo.com/28619851
- Rhino.Python Programmer's Reference

Modeling with Python

Disclaimer: I am a novice programmer. But I managed to get some nice results with a little help:

- Henry Segerman
- Google: Python Rhino tutorials
  One I used: ▸ vimeo.com/28619851
- Rhino.Python Programmer's Reference
  ▸ 4.rhino3d.com/5/ironpython/index.html

Modeling with Python

Disclaimer: I am a novice programmer. But I managed to get some nice results with a little help:

- Henry Segerman
- Google: Python Rhino tutorials
  One I used: ▸ vimeo.com/28619851
- Rhino.Python Programmer's Reference
  ▸ 4.rhino3d.com/5/ironpython/index.html

This talk will be much more Rhino specific now.

Graphs

We will be building the graphs the same way most graphing applications do:

Graphs

We will be building the graphs the same way most graphing applications do:

Plot a bunch of points and connect them.

Graphs

We will be building the graphs the same way most graphing applications do:

Plot a bunch of points and connect them.

Rhino uses nurbs

Graphs

We will be building the graphs the same way most graphing applications do:

Plot a bunch of points and connect them.

Rhino uses nurbs (Non-Uniform Rational Basis Splines).

Graphs

We will be building the graphs the same way most graphing applications do:

Plot a bunch of points and connect them.

Rhino uses nurbs (Non-Uniform Rational Basis Splines).

Fancy way to say: connect the points with polynomial curves so they are smooth.

Graphs

We will be building the graphs the same way most graphing applications do:

Plot a bunch of points and connect them.

Rhino uses nurbs (Non-Uniform Rational Basis Splines).

Fancy way to say: connect the points with polynomial curves so they are smooth.

Rhino command: AddInterpCurve

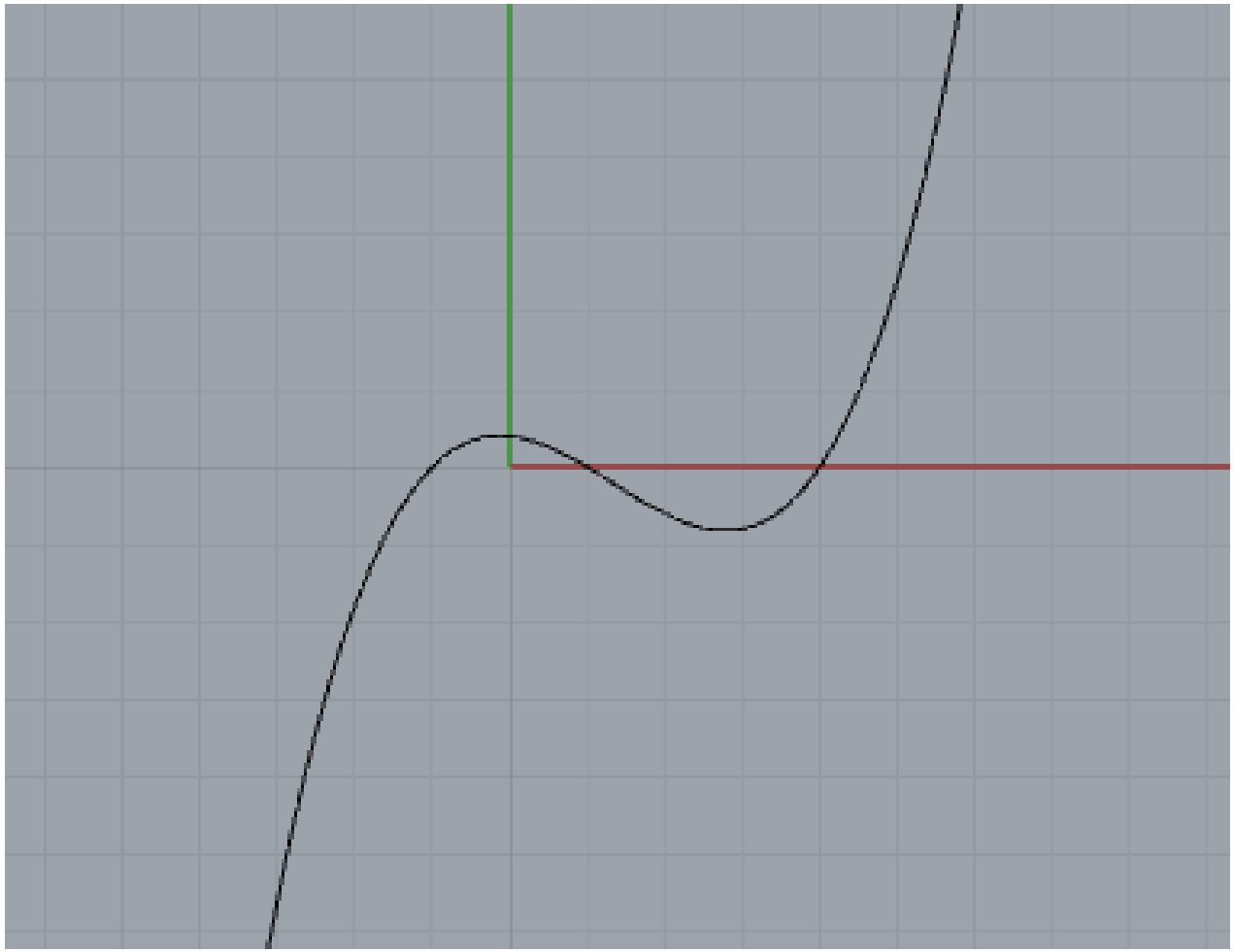
Graphs

The code from Henry that we use:

```
import sys, os
import rhinoscriptsyntax as rhino
from math import *

def draw_parametric_curve(function, param_range, num_points = 64):
  curve_pts = []
  for i in range(num_points):
    x = param_range[0] + (param_range[1] - param_range[0]) * float(i)/float(num_
    point = function(x)
    if point != None:
      curve_pts.append( rhino.AddPoint(point) )
  out = rhino.AddInterpCurve(curve_pts)
  rhino.DeleteObjects(curve_pts)
  return out


def cubic(x):
        return [x,.1*(x+1)*(x-1)*(x-4),0]


draw_parametric_curve(cubic, (-5,10))
```

Graphs

The result:



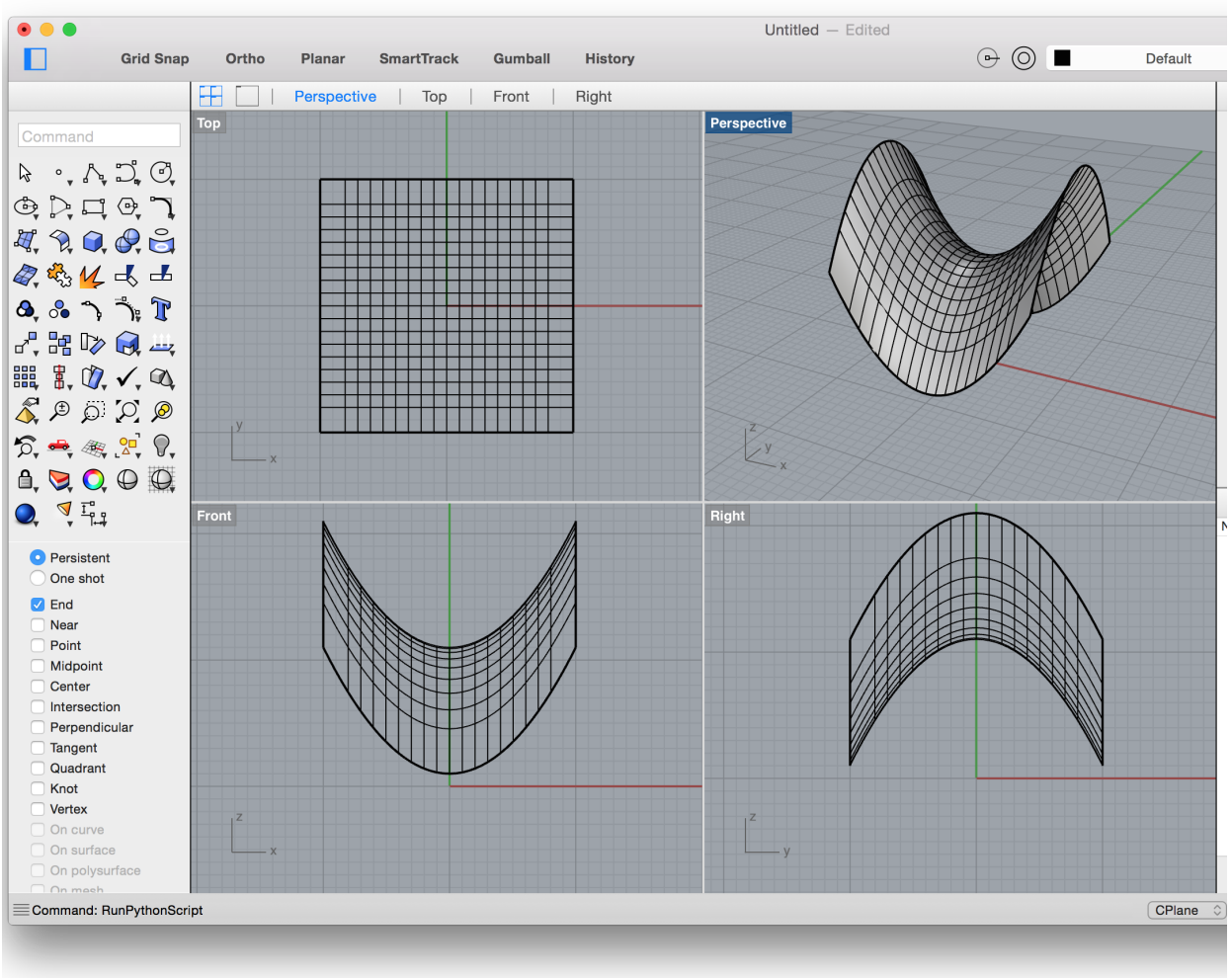
Graphs

For surfaces, graph:

$$z = \frac{1}{10}\left(x^2 - y^2\right)$$

```
import rhinoscriptsyntax as rs

count = 21,21

def s(x,y):
        return (.1*((x-10)**2-(y-10)**2)+11)


points = []

for i in range(count[0]):

    for j in range(count[1]):

        pt = i-10, j-10, s(i,j)

        points.append(pt)

rs.AddSrfPtGrid(count, points)
```

Surface result:



Integration

Lets talk about approximating

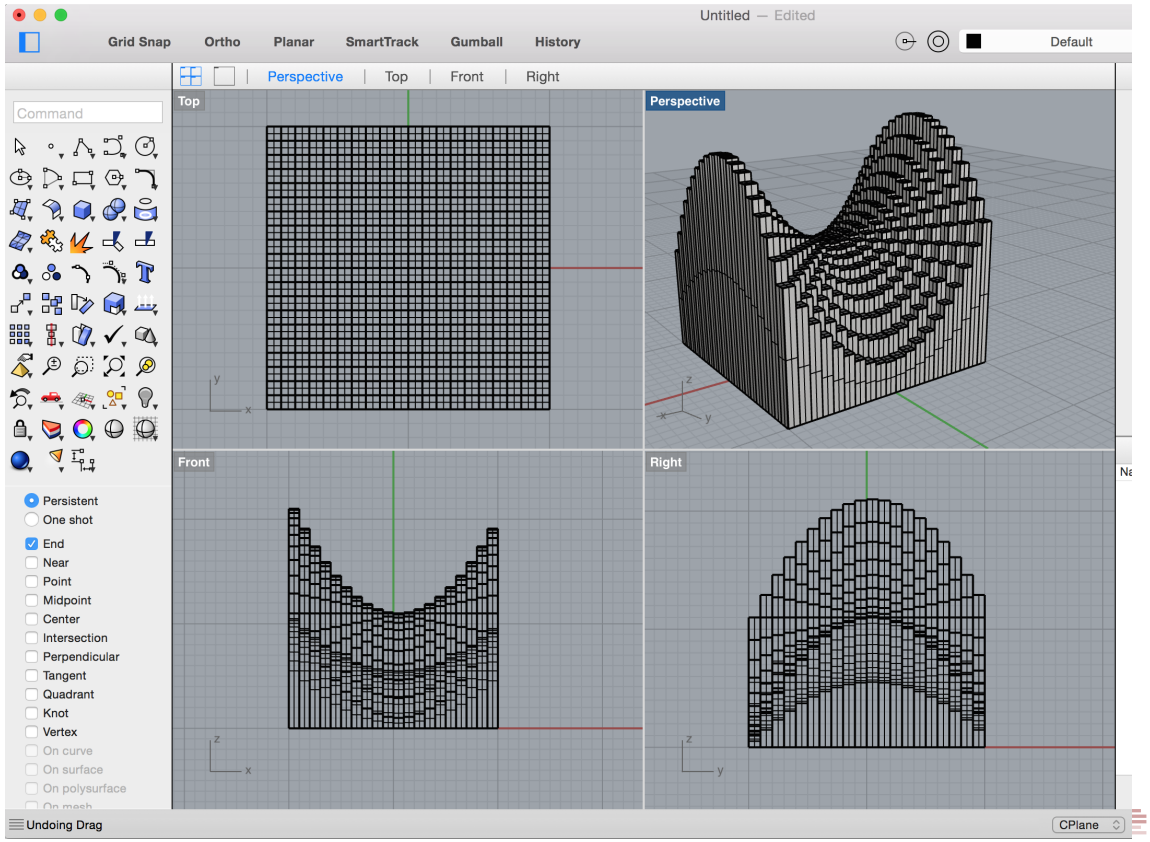
$$\int_{x=-10}^{10}\int_{y=-10}^{10}\frac{1}{10}\left(x^2-y^2\right)\mathrm{d}y\mathrm{d}x$$

```
import rhinoscriptsyntax as rs
from math import*

def s(x,y):
        return (.1*((x-10)**2-(y-10)**2)+11)


count = 20,20

for i in range(count[0]):
        for j in range(count[1]):
                rs.AddBox([ [i-10,j-10,0], [i-9,j-10,0],
                        [i-9,j-9,0], [i-10,j-9,0],
                        [i-10,j-10,s(i,j)], [i-9,j-10,s(i,j)],
                        [i-9,j-9,s(i,j)], [i-10,j-9,s(i,j)] ] )
```

Integration

Lets talk about approximating

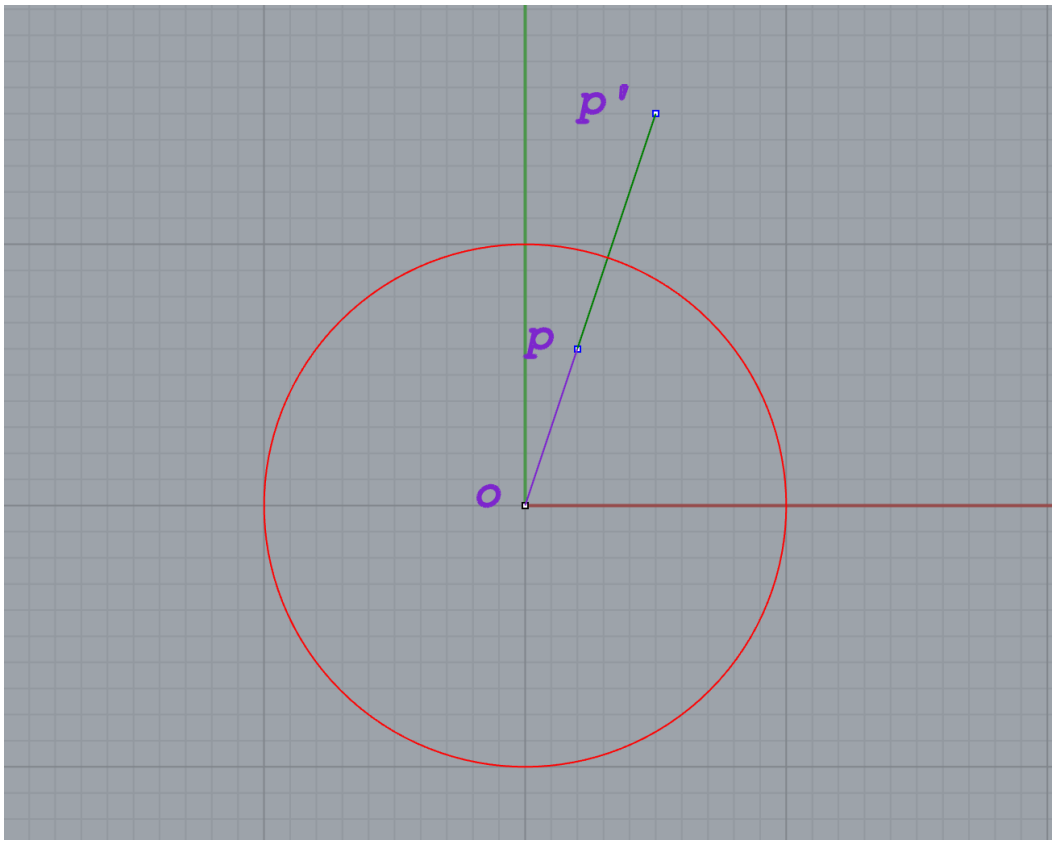
$$\int_{x=-10}^{10} \int_{y=-10}^{10} \frac{1}{10} \left(x^2 - y^2\right) \mathrm{d}y\mathrm{d}x$$



Hyperbolic Geometry

Hyperbolic Geometry

Given a point P, we can invert it about a circle of radius r and obtain a new point P' by the simple relation $OP \times OP' = r^2$.



Inverting a point about a circle

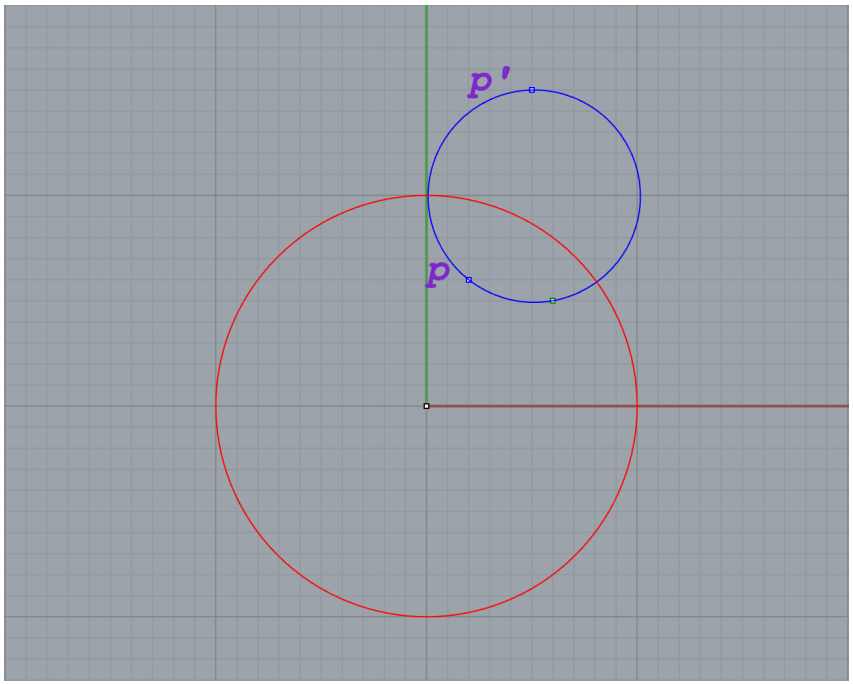

```
import rhinoscriptsyntax as rs
from math import*

basecircle = rs.GetObject("Select circle to invert about"

if rs.IsCircle(basecircle):

    radius = rs.CircleRadius(basecircle)

    center = rs.CircleCenterPoint(basecircle)

point = rs.GetObject("Select point to invert")

if rs.IsPoint(point):

        dist = rs.Distance( point, center)

a = (radius / dist ) ** 2


rs.ScaleObject( point, center, (a,a,a), True )
```

Inverting a point about a circle

Lines in hyperbolic geometry are circles that intersect our red circle perpendicularly. For every two points there is a unique line that passes through them.

Inverting a point about a circle

Lines in hyperbolic geometry are circles that intersect our red circle perpendicularly. For every two points there is a unique line that passes through them. To draw the line, we just invert one of the two given points, and draw the unique circle formed by those three points.



Inverting a point about a circle

Thank you!