**Sierpinski Fractal with Arnold**

Arnold Algorithm

The code below is broken down into separate parts that when combined together generate the Sierpinski fractal though a .ass path. The first section is the main sierpinski.py code that generates the fractal. Next is the #sierpinski\_mel.py that combines all the code and writes out the .ass file that will be used in Maya to constract the fractal. The final two codes are the Arnold\_Particles.ass and the #ass\_header.py. One I created these four

separate python(.py) files I ran the #sierpinski\_mel.py script and then loaded the resulting .ass into Maya. I did so by going to the Arnold tab at the top of an opened Maya file, scrolling to the Stand In section and pressing create. Once there I simply loaded the .ass file into the section labeled Path. Once the Sierpinski fractal was created I then applied an aistandard shader to the standin box and experimented with the lighting and shader.

import random

# Procedure halfstep returns a midpoint between two

# user-defined points

def halfstep(p1, p2):

 x = float(p1[0] + p2[0])/2

 y = float(p1[1] + p2[1])/2

 z = float(p1[2] + p2[2])/2

 return [x, y, z]

# Selects a value randomly from the input list

def pickpnt(pnts):

 return random.choice(pnts)

#------------------------------------------------

def sierpinski(verts, seed\_pnt, num):

 data = []

 for n in range(num):

 vert = pickpnt(verts)

 seed\_pnt = halfstep(vert, seed\_pnt)

 data.append(seed\_pnt)

 return data

if \_\_name\_\_ == '\_\_main\_\_':

 print "Debug"

 verts = [ [0,0,1], [1,0,-1], [-1,0,-1], [0,1.5,-0.2] ]

 seed\_pnt = [0,0.5,0]

 pnts = sierpinski(verts, seed\_pnt, 200)

 print pnts

#-----------------------------------------------------

# sierpinski\_mel.py

from sierpinski import sierpinski

from ass\_header import header

from random import uniform

def getBounds(verts):

 minx = 99999

 miny = 99999

 minz = 99999

 maxx = -99999

 maxy = -99999

 maxz = -99999

 for vert in verts:

 minx = min(minx, vert[0])

 miny = min(miny, vert[1])

 minz = min(minz, vert[2])

 maxx = max(maxx, vert[0])

 maxy = max(maxy, vert[1])

 maxz = max(maxz, vert[2])

 return [minx,miny,minz,maxx,maxy,maxz]

def sierpinski\_arnold(data\_path, num):

 verts = [ [0,0,1], [1,0,-1], [-1,0,-1], [0,1.5,-0.2] ]

 seed\_pnt = [0,0.5,0]

 data = sierpinski(verts, seed\_pnt, num)

 bounds = getBounds(verts)

 # Open a data file (mel)

 f = open(data\_path, 'w')

 f.write('### bounds: {} {} {} {} {} {} \n'.format(\*bounds))

 f.write('%s\n' % header)

 # Now insert the positional data

 f.write('points %d 1 VECTOR\n' % len(data))

 for pnt in data:

 f.write(' %f %f %f\n' % (pnt[0],pnt[1],pnt[2]))

 # Next the radii

 f.write('radius %d 1 FLOAT\n' % len(data))

 for pnt in data:

 f.write(' %f \n' % 0.003)

 # Next the rgb values

 f.write('rgbPP %d 1 RGB\n' % len(data))

 for pnt in data:

 r = uniform(0.0, 1.0)

 g = uniform(0.0, 1.0)

 b = uniform(0.0, 1.0)

 f.write(' %f %f %f \n' % (r,g,b))

 f.write('} \n')

 f.close()

if \_\_name\_\_ == '\_\_main\_\_':

 sierpinski\_arnold('/home/njones26/mount/stuhome/tech312/arnold/3d\_sierpinski.ass', 1200000)

#---------------------------------------------------------------------

Arnold\_Particles.ass

### exported: Thu Jan 11 14:08:20 2018

### user: njones26

### bounds: -0.354899 -1.148319 -0.339228 0.236235 0.00799609 0.315343

### fps: 24.000000

### render\_layer: defaultRenderLayer

### frame: 1.000000

points

{

 name nParticleShape1

 visibility 255

 receive\_shadows on

 self\_shadows on

 matrix

 1 0 0 0

 0 1 0 0

 0 0 1 0

 0 0 0 1

 opaque on

 matte off

 mode "disk"

 min\_pixel\_width 0

 step\_size 0

 declare rgbPP uniform RGB

 # UNIQUE DATA BEGIN\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 id 2669626447

 points 2 1 VECTOR

 -0.00253447657 -0.0159148201 -0.00246792519

 0.00543983886 -0.0164807029 -0.00867026765

 radius 2 1 FLOAT

 0.0199999996

 0.0199999996

 rgbPP 2 1 RGB

 1 1 1

 1 1 1

 # UNIQUE DATA END\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

#--------------------------------------------------------------

# ass\_header.py

header = """

### fps: 24.000000

### render\_layer: defaultRenderLayer

### frame: 1.000000

points

{

 name nParticleShape1

 visibility 255

 receive\_shadows on

 self\_shadows on

 matrix

 1 0 0 0

 0 1 0 0

 0 0 1 0

 0 0 0 1

 opaque on

 matte off

 mode "disk"

 min\_pixel\_width 0

 step\_size 0

 declare rgbPP uniform RGB

 id 2669626447

"""