TEACHING 3D MATHEMATICAL MODELING AND INTERPOLATION

Paul Bouthellier
Department of Mathematics and Computer Science
University of Pittsburgh-Titusville
Titusville, PA 16354
pbouthe@pitt.edu

Computer graphics can be used to illustrate mathematical concepts from junior high school through graduate school. In this paper we shall look at modeling 3D objects such as snow globe, a pyramid of oranges, and a screen saver.

Some of the concepts that computer graphics can illustrate are: translations (2D and 3D), skewing, rotations and projections in 3D via rotation matrices and quaternions, Boolean add and subtracts, rotation of cameras (in space and about their axes), morphing, refraction, reflections, mirrors, flipping about points, lines, and planes, Bezier curves and surfaces, deformations, scaling, collision detection (planes, spheres, cylinders, boxes, wireframe meshes, byricentric coordinates), creating spheres, cones, tori, Euler angles, quaternions, and Gimbal lock, extrusions into 3D, interpolation, fractal design, coordinate systems (local, global, camera, clipping, object-and mapping from one to another), numerical accuracy and round-off errors, Inverse Kinematics, spherical and cylindrical coordinate systems, polar and rectangular systems for mappings, forces: directional, point, damping, flow, normals and dot-products, NURBS, motions via quaternion interpolation, comparing the efficiency of rotation matrices and quaternions, spherical trigonometry, matrix transformations and how areas and volumes are altered/preserved, projecting images onto surfaces such as cubes and spheres, histograms for color distributions, eigenvectors and eigenvalues, and covariance matrices.

The packages used in this paper were: Studio 3D Max, Maya, Cinema 4D, Poser, Swift3D, Carrara, Paint Shop Pro, Photoshop, Swish Max, and Flash.

One thing I would like to note: Even though it takes years to become proficient (or even half-way decent) at creating 3D graphics and the required programming, it only takes a few days (or even hours) to learn enough basics to illustrate the mathematical concepts we may need in class.

Example 1- Snow Globe 2011

Figure 1-Create with Poser, Carrara, KRT Filters, and Python

In our mathematics classes we encourage our students to break hard problems into simpler pieces; this can be illustrated by trying to create any 3D object. Almost any manufactured object consists of many small pieces which are then put together by some process.

For the above snow globe:

- Create a snowflake using Bezier curves then extrude it into R^3
- Use spherical coordinates and transformations to create a uniform distribution in a spherical container
- Create the base using a Bezier curve and rotate it about the vertical axis.
- Project the text "ICTCM 2011" onto the base

These are illustrated below:

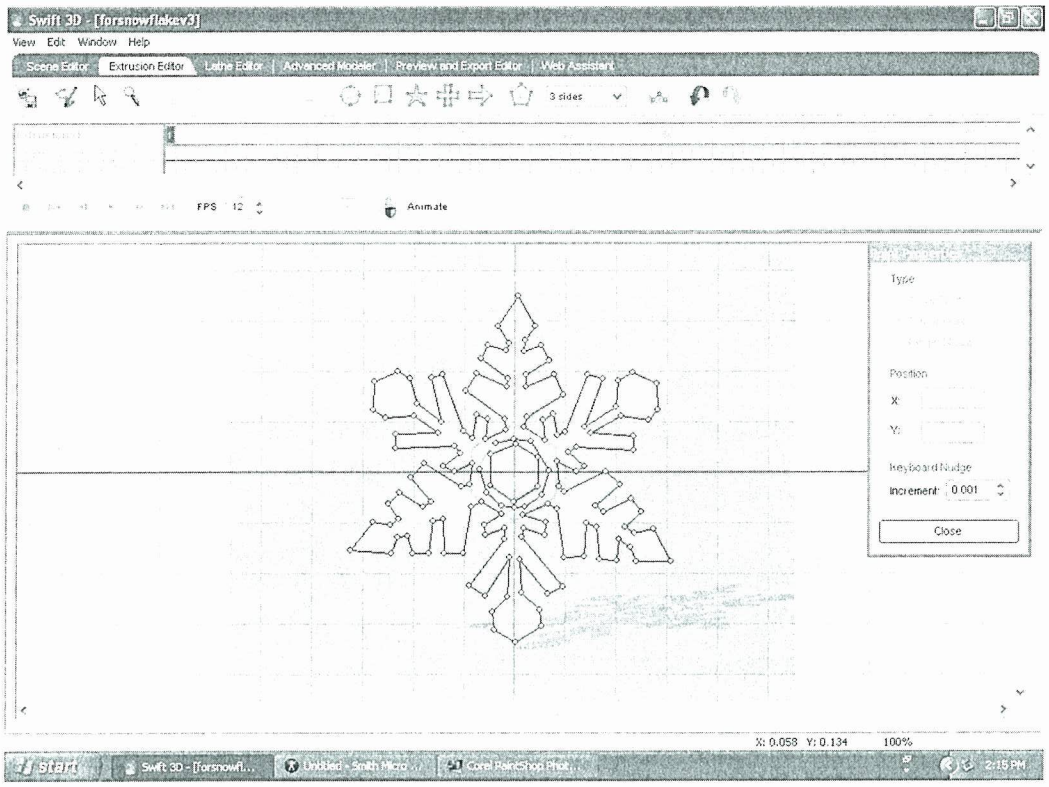

Figure 2-Creating the Flakes in Swift 3D using Bezier Curves

snowglobe2.py - Notepad

File Edit Format View Help

```
for a in range(numflakes):
    phi.append(math.acos(2*random.random()-1))
    theta.append(2*math.pi*random.random())
    p.append(r*math.pow((.9+.1*random.random()),.33333333333))


currentframe=scene.Frame()
d=currentframe


numframes=scene.NumFrames()

while d < numframes:

    scene.SetFrame(d)

    for e in range(numflakes):


        phi[e]=math.acos(2*random.random()-1)
        theta[e]=2*math.pi*random.random()
        p[e]=r*math.pow((.9+.1*random.random()),.33333333333)
        sqs[e].ParameterByCode(poser.kParmCodeXROT).SetValue(360*random.random())
        sqs[e].ParameterByCode(poser.kParmCodeYROT).SetValue(360*random.random())
        sqs[e].ParameterByCode(poser.kParmCodeZROT).SetValue(360*random.random())
        sqs[e].ParameterByCode(poser.kParmCodeXTRAN).SetValue(p[e]*math.sin(phi[e])*math.sin(theta[e]-2*math.pi*d/numframes))
        sqs[e].ParameterByCode(poser.kParmCodeYTRAN).SetValue(p[e]*math.cos(phi[e]))
        sqs[e].ParameterByCode(poser.kParmCodeZTRAN).SetValue(p[e]*math.sin(phi[e])*math.cos(theta[e]-2*math.pi*d/numframes))
        #sqs[e].ParameterByCode(poser.kParmCodeZTRAN).SetValue(2-4*random.random())
    d+=1
```

Figure 3-The Python code creating a uniform snowfall in R^3

Example II-A Pyramid of Oranges

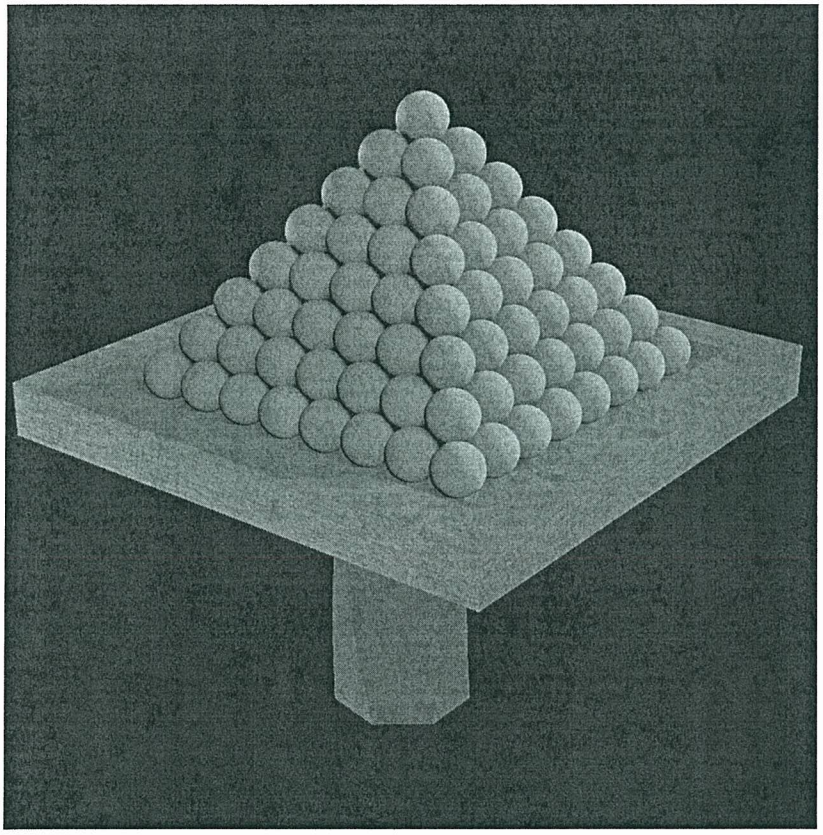

Figure 4- A Pyramid of Oranges (Python and Poser)

Here is a project whose mathematics can be done in a first level algebra course. All it involves is simple linear functions, one application of the Pythagorean Theorem, and one induction step.

The code is presented below:

In the final block of code is (almost) all the mathematics.

- Create each level of oranges by simple linear equations.
- The height of each level is done by the Pythagorean Theorem

To compute the total number of oranges needed, in line 10 the variable total uses the formula for the sum of squares.

```
File  Edit  Format  View  Help
import random
import poser
import math

scene = poser.Scene()
actor = scene.CurrentActor()
sqs=[]
levels=8
#c is diameter of sphere and internal/external conversion ratio
c=.84/8.6
total=levels*(levels+1)*(2*levels+1)/6
num=0
forxtran=[]
forytran=[]
forztran=[]

for i in range(total):
    newprop=scene.CreatePropFromGeom(actor.Geometry(),"hereitis")
    sqs.append(newprop)
    paramxrot=sqs[i].ParameterByCode(poser.kParmCodeXROT)
    paramyrot=sqs[i].ParameterByCode(poser.kParmCodeYROT)
    paramzrot=sqs[i].ParameterByCode(poser.kParmCodeZROT)
    forscale=sqs[i].ParameterByCode(poser.kParmCodeASCALE)
    forscale.SetValue(1)
    paramxrot.SetValue(0)
    paramyrot.SetValue(0)
    paramzrot.SetValue(0)
    propmaterials=sqs[i].Materials()
    for mtl in propmaterials:
        mtl.SetDiffuseColor(.925,.4549,0)
       #mtl.SetSpecularColor(0,0,0)
       #mtl.SetAmbientColor(0,0,0)
    forxtran.append(sqs[i].ParameterByCode(poser.kParmCodeXTRAN))
    forytran.append(sqs[i].ParameterByCode(poser.kParmCodeYTRAN))
    forztran.append(sqs[i].ParameterByCode(poser.kParmCodeZTRAN))


for i in range(levels):
    for j in range (levels-i):
        for k in range(levels-i):
            forxtran[num].SetValue(i*c/2+j*c)
            forytran[num].SetValue(i*.7071*c)
            forztran[num].SetValue(i*c/2+k*c)
            num+=1
```

Figure 5-The Python code to create the Pyramid of Oranges

Overall, a very simple application of mathematics that leads to a nice result. This problem can be extended to that of sphere packing problems in containers of irregular shapes.

Example III- A Screen Saver

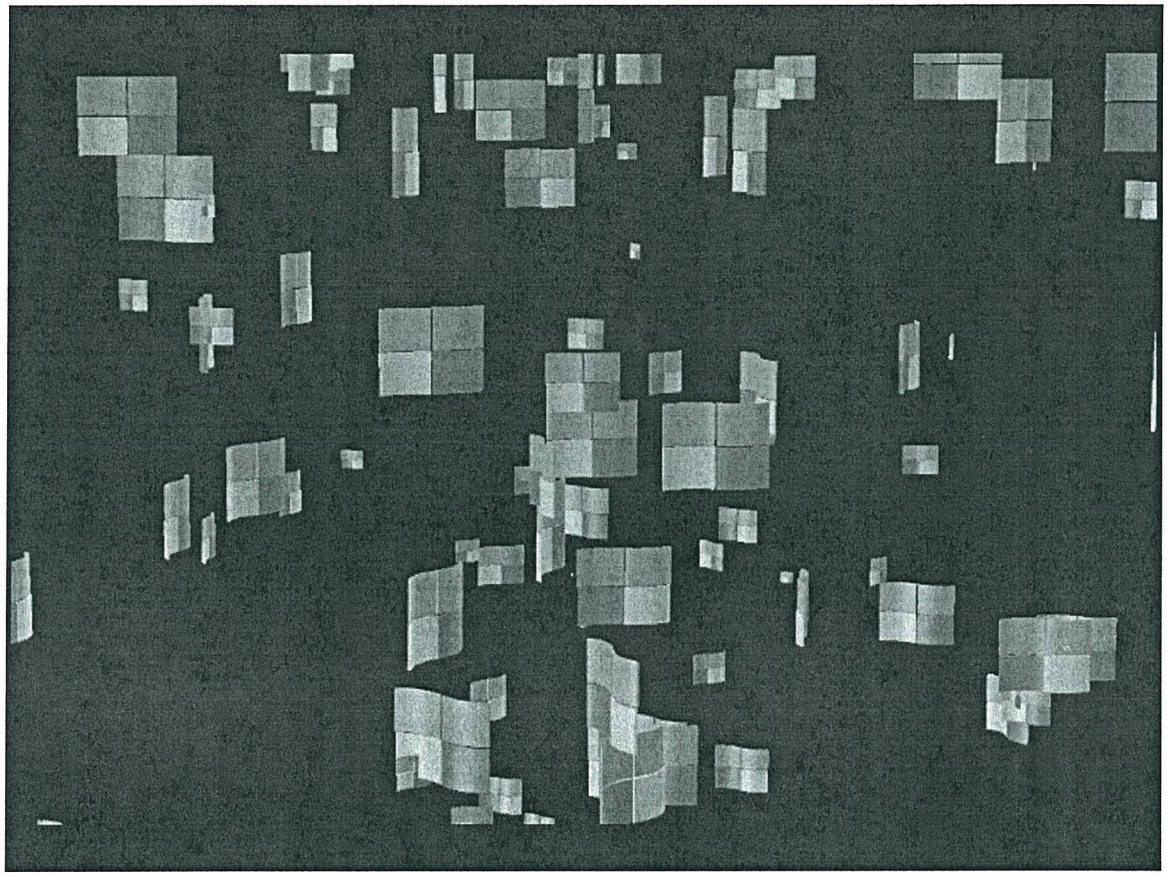

Figure 6-A Still from a 3D Screen Saver (Python and Poser)

The first object that had to be created was a single copy of the logo. A close-up is provided below:

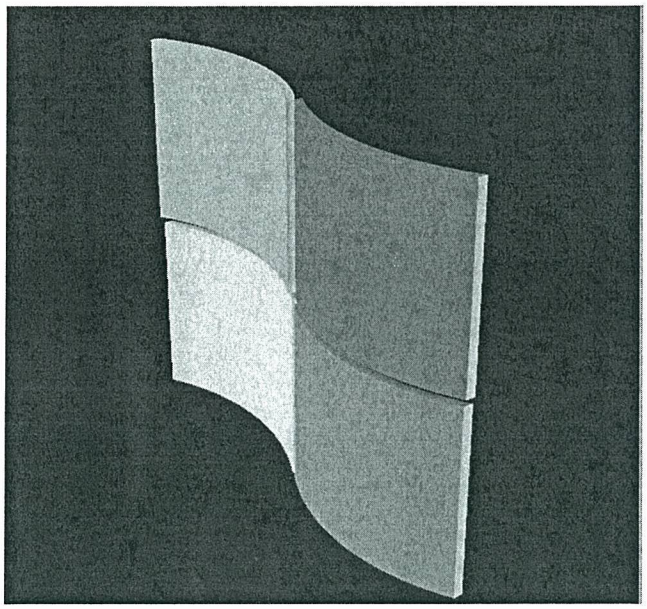

Figure 7-A 3D Logo (Using Studio 3D Max and Carrara)

To create this object the following steps were needed:

1. Create a Bezier curve and a connected translated copy to form the base of one (of the four) sections.
2. Extrude the base along the normal to create one section.
3. Make three copies and rotate, reflect, and translate as necessary.

Once we have one copy, the remaining 149 copies were easy.

To create the animated screen saver the following steps were necessary:

1. Randomly assign each of the logos a z-value.
2. Use basic trigonometry to rotate the windows about the center of the scene (with a phase shift so they start and stay separated)
3. Use the same trigonometry to have the windows rotate about their own y-axis.

The code to create the screen saver is given below:

Figure 8-Simple Trigonometry Creating a Screen Saver