CREATING 3D ANIMATIONS IN FLASH USING MATHEMATICAL MODELING

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

Melanie Anderson
Department of Business
University of Pittsburgh-Titusville
Titusville, PA 16354
moanders@pitt.edu

Introduction:

In this paper mathematical modeling will be used to create 3-dimesional objects and then move them in 3-space using 6 degrees of freedom-translations along the three coordinate's axes and rotations about each of these axes. The mathematics used in the projects requires: coordinate systems in R^2 and R^3, rotation about object and world axes in R^3, Euler angles and quarternions, shearing and scaling matrices, translations and rotations via 4x4 matrices, projections onto a 2-D screen, parametric curves in R^3, back-face culling, normal vectors, z-averaging, and graph theory. Mathematical modeling for computer applications is so diverse that its study serves as an excellent source of practical examples and projects in many undergraduate (and graduate) courses.

The programs in this presentation were written in the ActionScript language of *Flash*. The programs were saved in the .swf format and embedded in web pages. Students were not required to learn how to program in ActionScript-just to be able to alter the code and re-run the programs. This worked well after walking through a few examples.

Projects:

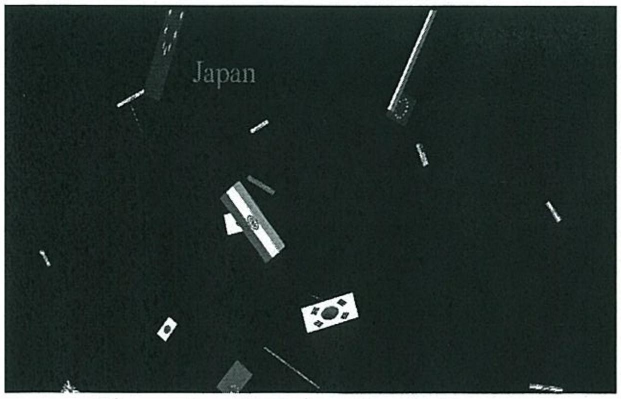

Figure 1a-3D Educational Game

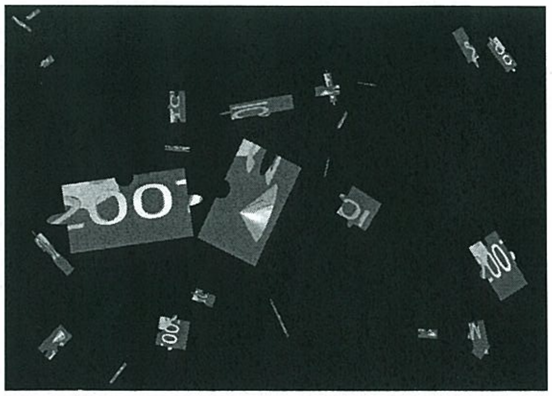
Figure 1b-3D Jigsaw Puzzle

The first project (illustrated in figure 1a) was to create a 3-dimensional game for children to help them learn to identify the flags of various countries. This game was used as part of a SIFE (Students in Free Enterprise) project to help schools in the local community. The flags rotate about their local axes as well as about an origin in R^3 as they move towards the screen. As the names of various countries appear, the students have to move the mouse cursor over a corresponding flag-which then indicates that they are correct with various special effects.

The second project (illustrated in figure 1b) was to create a jigsaw puzzle in 3-D as part of the SIFE team's annual presentation-this year entitled "Coming Together". The pieces begin as a completed puzzle showing the front side of the puzzle. The pieces then move apart, rotate in 3-space, and then reform showing the reverse side of the puzzle.

Coordinate Systems:

In computer graphics, as well as many engineering applications, the coordinates systems used in R^2 and R^3 are not the ones that are taught in most math classes-the origin (0,0,0) at the center, positive x towards the viewer, positive y to the right, and positive z upwards.

In R^2

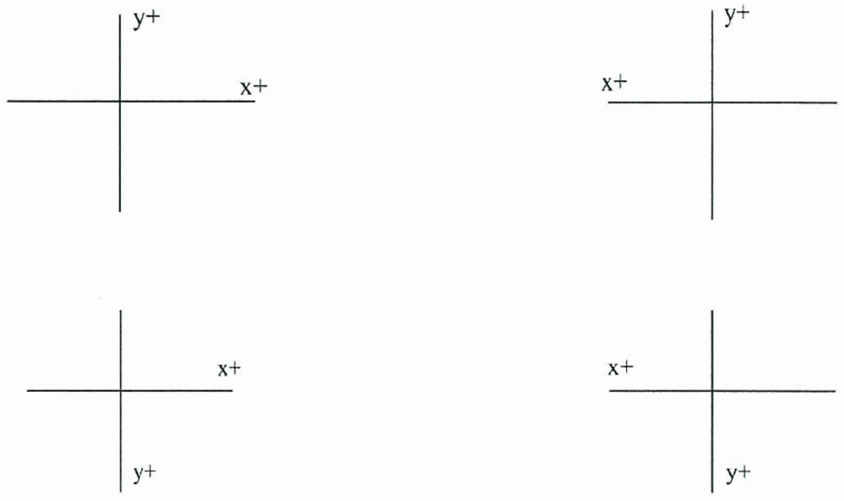

Figure 2-Coordinate Systems

Reversing the x and y labels gives 8 possibilities.

In languages such as Java, ActionScript, and JavaScript, the coordinate system used is the one where positive x is to the left and positive y is down. The origin is at the upper left corner.

In R^3

For x, y, and z there are 8 possibilities. Given 6 permutations for labeling the axes there are 48 possible coordinate systems.

Rotations in R^3

Rotations about the object's axes in R^3 can be accomplished using Euler angles. In general, using a right-handed system, the following rotation matrix creates a counter-clockwise rotation about a unit normal vector $n = < n_x, n_y, n_z >$ [1].

$$R_n(\theta) = \begin{bmatrix} c+(1-c)n_x^2 & (1-c)n_x n_y - sn_z & (1-c)n_x n_z + sn_y \\ (1-c)n_x n_y + sn_z & c+(1-c)n_y^2 & (1-c)n_y n_z - sn_x \\ (1-c)n_x n_z - sn_y & (1-c)n_y n_z + sn_x & c+(1-c)n_z^2 \end{bmatrix}$$

where s=sin(θ) and c=cos(θ), θ being the angle of rotation. Using Euler angles for rotations can cause several interpolation problems: They often do not take the shortest path and do not have constant angular velocity.

To overcome these problems SLERP (Spherical Linear Interpolation) using quaternions may be used.

Quaternions:

Define a quarternion-which consists of a real part and 3 complex parts by

$$q=q_0+iq_1+jq_2+kq_3=q_0+\mathbf{q}$$

where

$$i^2=j^2=k^2=ijk=-1, \; ij=k=-ji, \; jk=i=-kj, \text{ and } ki=j=-ik$$

Given two quaternions p and q

$$p=p_0+ip_1+jp_2+kp_3=p_0+\mathbf{p}$$
$$q=q_0+iq_1+jq_2+kq_3=q_0+\mathbf{q},$$

quaternion multiplication is defined as follows [2]:

$$pq = p_0 q_0 - \mathbf{q} \bullet \mathbf{p} + p_0 \mathbf{q} + q_0 \mathbf{p} + \mathbf{p} \times \mathbf{q}$$

Given pure quaternions (0 real part):

$$q = iq_1 + jq_2 + kq_3 = \mathbf{q}$$
$$p = ip_1 + jp_2 + kp_3 = \mathbf{p}$$

their product is given by:

$$pq = -\mathbf{p} \bullet \mathbf{q} + \mathbf{p} \times \mathbf{q}$$

Note that the product involves the dot product and the cross product of the vector parts of p and q. The cross product, being nonsymmetrical, implies that quaternion products are nonsymmetric.

To construct rotations in R^3 we define the unit vector

$$q = \cos(\frac{\theta}{2}) + u \sin(\frac{\theta}{2}) = q_0 + q_1 i + q_2 j + q_3 k$$

where *u* in a unit vector in R^3.

Defining the reciprocal of q as follows:

$$q^{-1} = q_0 - iq_1 - jq_2 - kq_3 = q_0 - \mathbf{q},$$

where $v \in R^3$ is represented as a pure quaternion ($0+v$).

then

$$qvq^{-1}$$

is a right-handed rotation of *v* by an angle of θ about the vector q [2].

Quaternions are an interesting topic for discussion in history of mathematics courses. The "Quaternion Debate" of the 1890's, which pitted vector analysis against quaternions, has been reignited in the last decade via computer programming for game design.

A project related to the two given above is that of creating and rotating 3-D objects. A simple cube is created below (the equivalent of "Hello World" in programming).

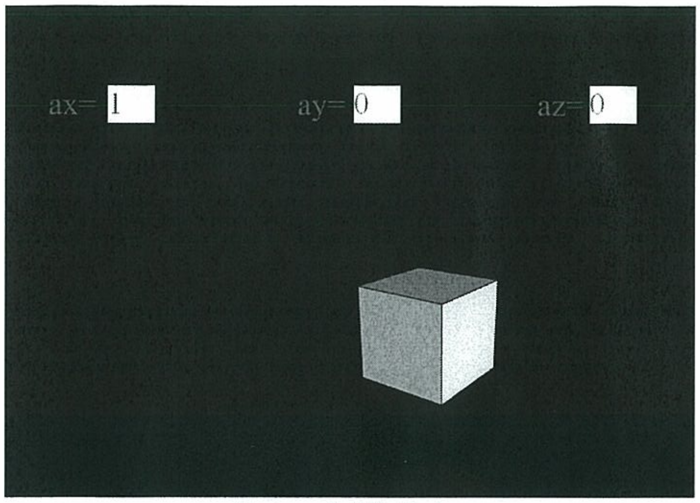

Figure 3-3D Cube

Problems involving graph theory (such as-can an object be created by a series of straight lines without retracing a given line) may also be studied. Performing back-face culling (which surface appears in front of which other surfaces) can be studied using z-averaging or normal vectors (amongst other methods) is also an interesting mathematical exercise.

Conclusion

Computer programming and computer graphics allows many examples and projects in mathematical modeling to be created and studied in undergraduate and graduate courses. The advantages of such projects are that students get to see immediately whether their ideas and equations actually work. Students also can see the connections between their various courses. A single project can involve concepts from trigonometry, calculus, linear algebra, numerical analysis, mathematical modeling, physics, etc.... Students are also motivated to learn more mathematics in order to create better effects and more efficient programs.

References:

[1] 3D Math Primer for Graphics and Game Development, Wordworth Publishing, Inc., 2002, by Dunn and Parberry.

[2] Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality, Princeton University Press, 1999, by Kuipers.