Re: Calculating new coordinates in 3D after moving an object

Source: http://sci.tech−archive.net/Archive/sci.math/2007−10/msg03820.html

- *From*: matt271829−news@xxxxxxxxxxx
- *Date*: Sat, 20 Oct 2007 16:46:58 −0700

On Oct 20, 5:05 pm, andreas <anpa...@xxxxxxxxx> wrote:

> Good evening.
>
> I have a question on how to calculate an objects coordinated after altering its position.
>
> Example:
>
> A triangle with coordinates;
>
> P1 : −4, −4, 4 (x, y, z)
> P2 : −3, −4, 4
> P3 : −3, −4, 5
>
> Let's say I want to rotate the line P2−P3 about 14 degrees around the axis P1−P2.
>
> How can I find out the new coordinates of P3?
>
> It would be a simple task using trigonometry if it it was only a 2D space, but 3D is puzzling me a little.
> Someone hinted me that "Quaternion" would be the solution, but I tried to understand but didn't. Embarrassingly that area is a little over my understanding of mathematics.
>
> Another question is if the object would be more complex, such as a Dodecahedron.
>
> I assume I first must find the a central point around which the rotation should be done, but since all points are moving...
> How can I find out the new coordinates of each and every one?
>
> Any pointers to online resources or direct help are much appreciated.
>
> Thank you.

You don't need to find any "central point": 3D rotations are about an axis (a straight line), and the axis is all you need to know. Assuming a points−and−straight−lines object, the complexity of the object makes no difference to the complexity of the problem, except in the very obvious sense that you have to calculate the rotation for every point.

The usual method would be to translate (shift) the figure so that the axis of rotation passes through the origin, perform the rotation, and then perform the reverse translation. The translations are trivial, so what you really need to know is how to rotate a point (x,y,z) through angle theta about a vector (i,j,k) (i.e. about a line between (0,0,0) and (i,j,k)). To make the things marginally easier, assume also that the vector has been normalised to have length 1 (i.e. i^2 + j^2 + k^2 = 1).

The formulas below, which are just a matrix multiplication done explicitly, achieve this. The rotated point is returned in (x',y',z'). You also need to consider the clockwise/anticlockwise issue. Let's assume the handedness of the coordinate system is such that when looking towards origin from a point in (+,+,+) octant, the positive x, y, z axes are anticlockwise in that order. In these formulas positive theta then corresponds to an anticlockwise rotation when viewed looking al