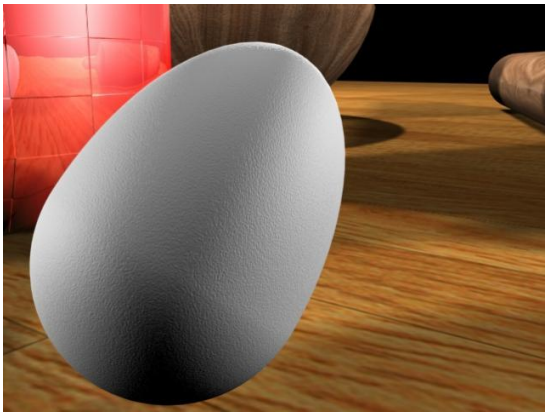
# A solution for 3ds max

## How to do:



# Making a 3d Egg, translating and rotating it in MaxScript.

**Made by: Ricardo Sánchez Encinas**

**Ciudad Obregón, Sonora, México**

**23/05/2010**

Ulsa Noroeste

*A project made for my animation class.*

## Introduction:

This is the explication of the process of how the egg was made. This is not an explanation of the code. You can see the code in http://www.scriptspot.com/3ds-max/scripts/egg-maker-a-rigged-egg-with-scripting, download the installer, install it and open the code generated in the rotation script, position script and the macro file.

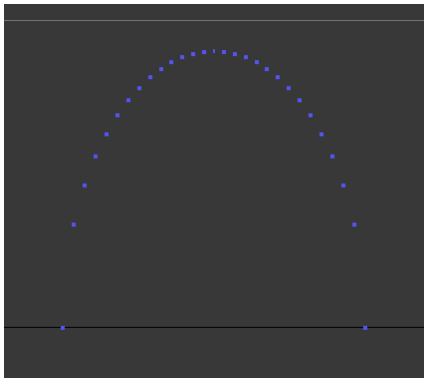
The basic logic of how to make the shape and the rig is in this file. It's recommended to know about Analytic geometry and quaternion before read this how-to.

Have the code open at the same time when you read this document will be useful.

## The beginning (Making the poly):

I had to create the shape of the egg, but it won't be simple as create a sphere and modify this shape for making an egg. It's possible to create eggs in this way but it is not what I need for this case.

It's necessary to make an egg with a formula for make totally controlled egg. There are a lot of formulas which can be used for describing the form of an egg like parabolas, circles, ellipses, hyperbolas (conic sections). For this case, the different halves of ellipses will be used. Because they look like an egg and they can be connected easily and parabolas can', the same problem for hyperbolas and circles can't describe an egg.



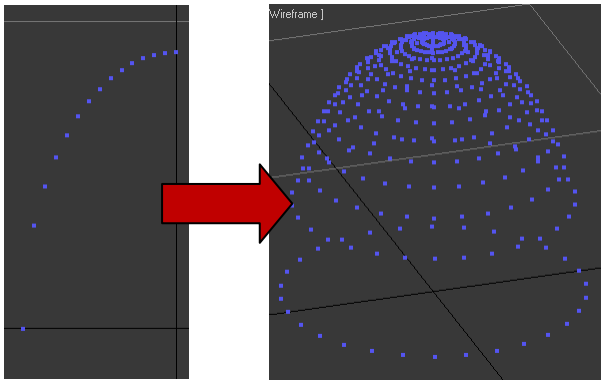
After decide which formula will be used. It's time to make an empty editable poly with maxscript (It's not possible, but it's possible to create an editable_mesh and convert it to Editable_Poly).

When it is created, the vertexes have to be added. The vertexes' position will be calculated with the formula of the ellipse.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

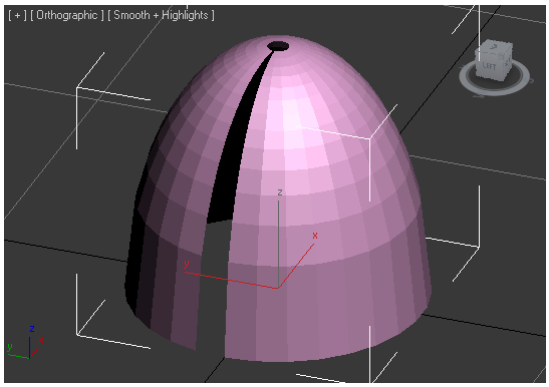
Two different halves are necessary for the egg. A half for positive values in Z and one for negative values have to be added. The formula has to be in a canonical form (y = …) and y in this case will be z. The two halves have to have the same 'a', but different 'b', the upper half will have a bigger 'a' than the bottom half. And then the shape of the egg is ready, with 2 different half of ellipses.

But do not forget that a 3d poly is making, not just a 2d shape. Then it's necessary to use a loop for generating a cloud of vertexes that will be united later. A quarter of the ellipse is better for do that than the half.



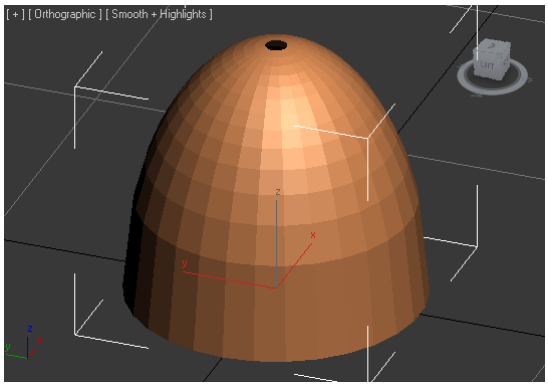

And now the vertexes of the upper half are ready for making the polygons. For connecting the vertexes, they have to be in a bidirectional array with loops and rings. With a loop in a loop and this array, it's possible to transform the cloud of vertexes in the upper part of an egg. The main

loop is for acceding to the horizontal circles and the loop inside is for the vertexes in it.



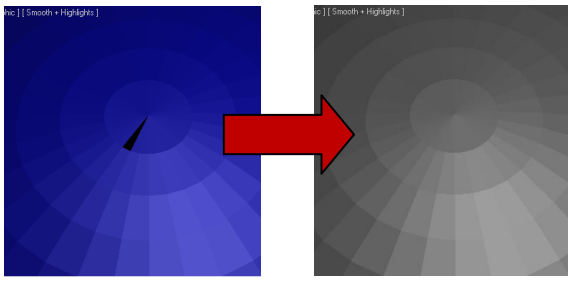

See the code of the macro script (lines 40-48)

Almost all of the polygons can be created with this loops, but not every polygon. After those ones, another loop is necessary to close the egg (line 47). This loop can be the same than the loop for acceding to the circles of vertexes as it is in the code.

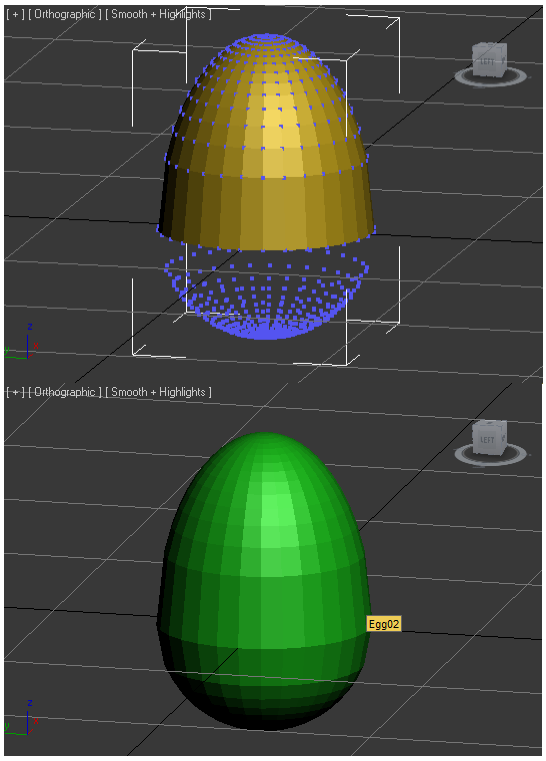



A circle is empty in the upper side of the egg. Close it with another loop is necessary; the procedure will be different, because it's not made of quadrangles like the rest of the egg. This will be made of triangles. (Line 50-51)

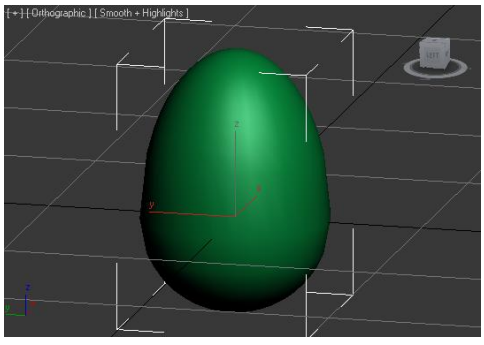
The same problem will occurred with the last triangle. But a loop will not be necessary because it's just one triangle to make.



And the same procedure will follow for the bottom part of the egg. Just the 'b' will change and it won't be positive, it will be negative.



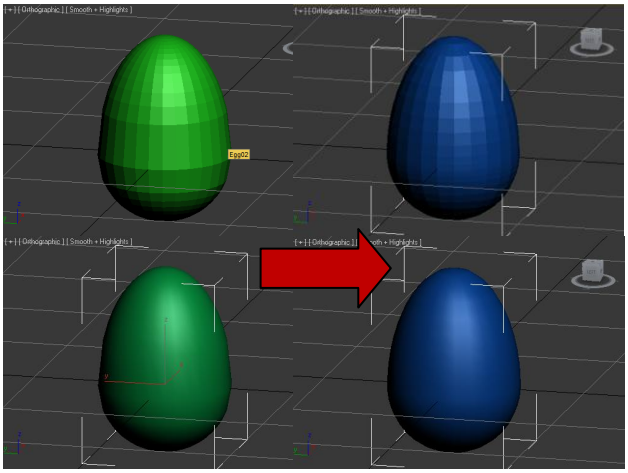

And now, all the polygons have to have the same smoothing group, because eggs don't looks like the last image.

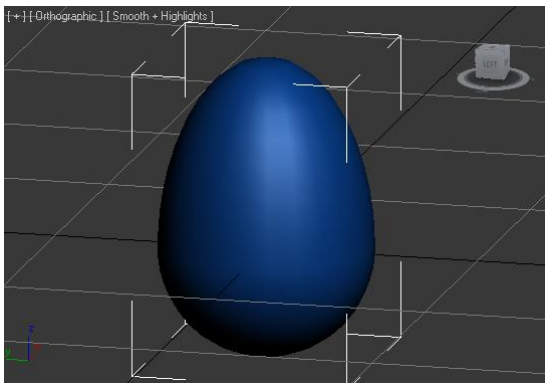


And the egg is ready. But it has much resolution in the extremes but not in the in the half. That is because we have used a linear equation for the x coordinate of the vertexes. The solution is to use a not linear expression. In this case a quadratic equation is used. (Line 24)

$$bad\ result\ x\ =\ t$$

$$better\ result\ x\ =\ t^2$$





## 2D Rotating and translating:

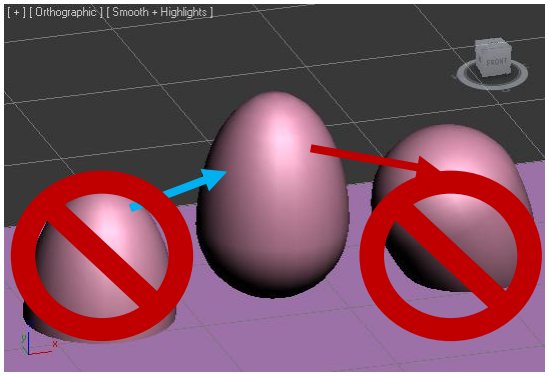
It's not the main topic of the project, it's just for understand wha will happen in 3D rotation and translation.

It is explained for X axis and Z axis.

The easiest part of that is rotating, it's not a big trouble. It is just taking the X coordinate and divide it by the radius of the the egg ( 'a' value when we create the egg). And it's enough. It won't be exactly because b ≠ a. But it is not bad, it will be a nice effect in the final motion.

Now the egg is ready for rotating when it is moving in X axis.

The real problem is that eggs can't be inside the table.




If we rotate it manully it will take many hours to avoid this problem. That is why this script exists.

The egg was created with math formulas because we want to know this shape everytime. That is what let us fix this problem.

With a simple  Sine of the angule of rotation we can know the the a valid X to introduce in the formula of the ellipse, and with it, it's posible to get the X coordinate.

After getting those coordinate, the height of the egg in one point could be calculated.

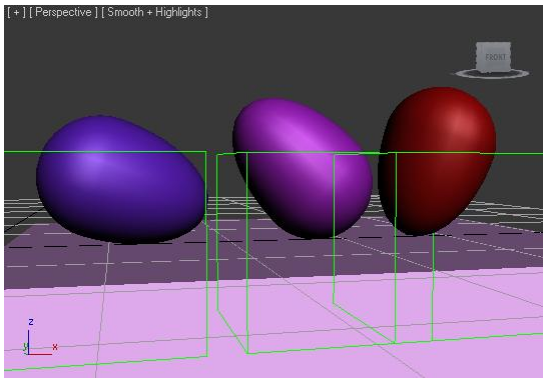
$$x = sin(\theta) * a$$

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

$$x^2 + y^2 = height^2$$

And now the problem of rotating the egg in 2D is solved.

Just remember that we have 2 different b in the shape. We have to validate if the egg have the upper part down or the bottom part. (Example: *if (ang >90 and ang <270) then Calculate with b of upper part else …*)




**3D Rotating and translating:**

It's working in a 3d software, then the 2d solution is not enough.

The problem here will be much greater than in 2d because Euler rotation was good for 2d. But Euler do not work properly in 3d. Then quaternions have to be used. And they are not a simple number, they are hypercomplex with one real dimension and 3 imaginary.

The egg have a dummy as a parent, because it will be useful for controlling it, and the dummy can be controlled by a ray and the egg could be used in not linear surfaces.

For rotating the egg in every direction. Euler will be useless. The very short code for rotating in 2d will become in a long code. For read it, generate a rigged egg and then open the rotation script in it. The coide is commented with the explanation.

But in simple words. The solution is:

First we have to get the actual position of the egg, and compare it with the position in the last frame. Calculate the distance between these two points, calculate a normal vector of this distance. And now, with the calculated distance, the angle to rotate can be calculated in the same way than 2d (distance/radius). Create a new quaternion structure with these values (*quat ang normal_vector*) and the difference of rotation for the new frame is calculated. Adding the difference to the last frame rotation will be enough for knowing the actual rotation.

But remember that avoiding the collition with the surface is necessary.

The point of touching of the egg and the surface will be the pivot or center of the dummy.

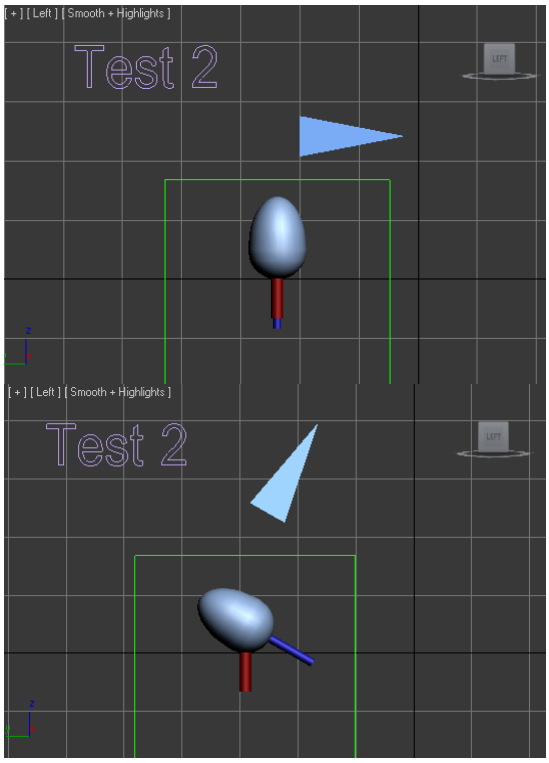
See the code in the position script of the egg. (Just the important lines are commented)

As in 2d, distinguish the part which is down is important.

Remember that the rotation is in quaternions. And it is in 3d, that means that the angle in quaterion is not useful for the equation.

And it's necessary to calculate a vector that rotate with the egg, it can rotate many times tought the surface of the egg in different curves with big angle or an unusual vectors, but at the end it will have a final value that will be useful for calculating the rotation.

In this image the vector for calculating the final angle is the blue, and the red is a static vector. The pyramid is a meter that indicate the difference between the angle of the red and the blue vector.

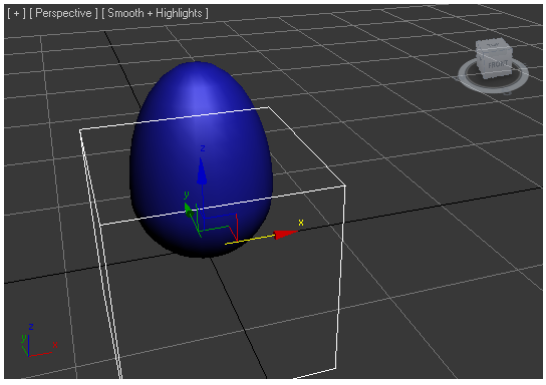


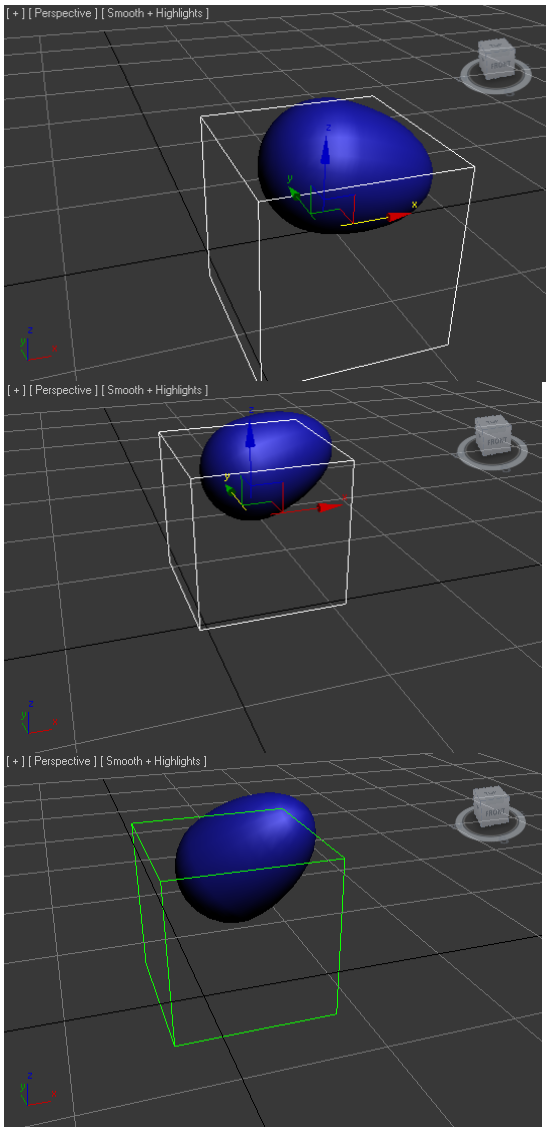

The value of the meter grow when the blue vector is far of the red vector.

This angle that indicate the meter is calculated with a dot product of the red and the blue vector.

This new angle is the angle which will help us to know the final 2d rotation of the egg. This angle is ready for be introduced in the same formula of 2d position.

And now the egg can rotate in any direction(X and Y) and its rotation can be calculated and the position in Z too.




As in the images.

*Thanks for reading.*