UNIVERSITY: UNIVERSIDAD LA SALLE NOROESTE
SUBJECT: ANIMACIÓN IV
INSTRUCTOR: ENRIQUE ROSALES
STUDENTS: OMAR R. HERNÁNDEZ
          MARTÍN SOTO
          JENNIFER RÍOS
          MARLON RÁBAGO
          EDILBERTO MÉNDEZ
DATE: MAY THE 24th 2010

RIGGING TO MOVE AND ROTATE AN EGG IN 3DS MAX USING MAXSCRIPT

I INTRODUCTION:

Defining the Problem

When trying to animate a ball, is somehow easy because of the balance and the shape, but what about the shape of an egg? It's obviously not the same and it will not rotate equally, much less in an uneven terrain. Making the rig is not that difficult to do, the hardest part is to get the right idea. The goal of this rigging is to make the egg rotate while keeping it touching the ground automatically, so the animator only has to worry about the speed of the egg and the distance it travels.

Initially the egg rotates as a simple ball, but due to the peculiar shape of the egg, we need to place it on the ground by moving the Z position using MaxScript.

II SOLVING THE PROBLEM

The Solution

The egg has an unique shape, it does not have a geometric standard shape, it seems like an oval but it is not. It is wider on the bottom and more narrow on the top.

This solution is a bit rudimentary, but at the same time it allows you to optimize it in the future in a comfortable way. This script evaluates the lowest point of the egg in the Z axis and defines it as the one touching the ground, so later it pushes it up so the egg is over the ground.

First we need to have a shape of an egg, then you cover it with helper points, (the more points you have will be better the movement of the egg. If you have few points the egg will have rude movements and it will surpass the surface.)
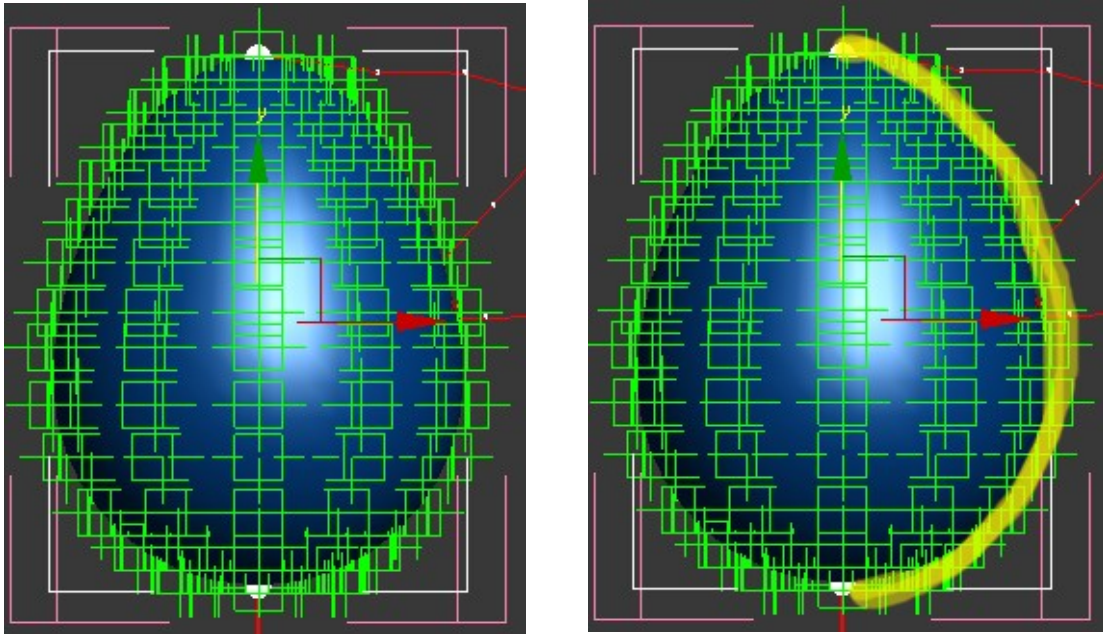
Image 1

In this step we put points in a side of the egg. 14 points are used to cover one side (as shown in image 1), and a set of 12 points every 30 degrees, making a total of 146 points around the egg. In the image perspective, the egg has a perfect shape, so you will have your egg perfectly covered of helper points.

To make sure the egg rotates correctly in all directions, it is necessary to use this rotation script, found in the *cuneytozdas's rolling ball tutorial,* as show in image 2. Also, and very important, the egg is re-parented to a box, this is done so the egg can be moved in the "x" and "y" axis position.
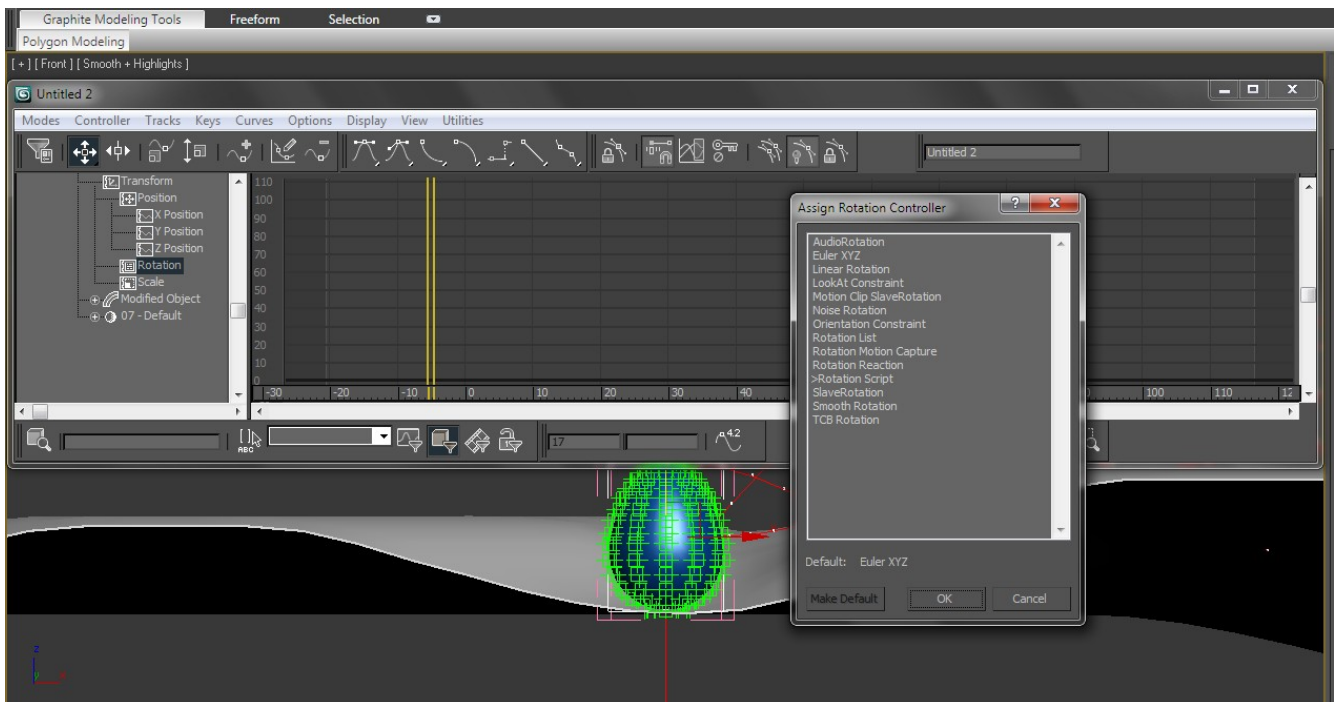

Image2

```
1   rollout elRodaHuevos "El Roda Huevos" width:210 height:210
2   (
3       label lblTitulo "El Roda Huevos" pos:[66,10] width:90 height:19
4       label lblDuracion "Duración de la Animación" pos:[40,37] width:139 height:19
5       label lblFrameInicial "Frame Final" pos:[40,99] width:62 height:14
6       label lbl5 "Frame Inicial" pos:[35,74] width:62 height:15
7       editText edtFrameInicial "" pos:[100,69] width:60 height:17
8       editText edtFramefinal "" pos:[100,96] width:60 height:17
9       button btnAplicar "Aplicar Animación" pos:[55,180] width:98 height:22
10      progressBar Progreso ""  width:180 height:14 color:[187,31,38] pos:[15,150]
11      label lblProgreso "Progreso" pos:[80,130] width:62 height:15
12
13      global puntos = #()
14      global FrameInicial = 0
15      global FrameFinal = 0
16      global Rango = 0.0
17      global ProgresadoPaso = 0.0
18      global ProgresadoTotal = 0.0
19
20          on btnAplicar pressed  do
21      (
22
23
24
25          FrameInicial = edtFrameInicial.text as integer
26          FrameFinal = edtFrameFinal.text as integer
27          Rango = FrameFinal - FrameInicial
28          ProgresadoPaso = (100.0/Rango)
29
30          function seleccionaPuntos i =    2
31          (
32              for obj in  helpers do
33              (
34                  puntos[i] = obj
35                  i = i + 1
36              )
37          )
38
39          function rolamElHuevo t=
40          (
41              obj = $EggOut                        3
42              zmen = puntos[1].pos.z
43              puntoBajo = puntos[1]
44
45              for c in 1 to puntos.count do        4
46              (
47                  z = puntos[c].pos.z
48                  if z < zmen then
49                  (
50                    zmen= z
51                  puntoBajo=puntos[c]
52                  )
53              )
54
55              rayo = ray [obj.pos.x,obj.pos.y,(obj.pos.z +500)] [0,0,-1]   5
56              intersec = intersectRay $Box01  rayo
57
58              obj.pos.z = obj.pos.z - zmen + intersec.pos.z    6
59          )
60
61          animate on                              1
62          (
63
64              seleccionaPuntos 1       2
65
66              for c in FrameInicial to FrameFinal do
67              (
68                  at time c rolamElHuevo 1        3,4,5,6
69                  ProgresadoTotal+=ProgresadoPaso
70                  Pro = (ProgresadoTotal-ProgresadoPaso)
71                  progreso.value= Pro
72              )
73          )
74      )
75
76  )createdialog elRodaHuevos
```

Image 3

Image 3. Shows the code explained in blocks of colors. Next is the explanation of each block.

Block number one is the first being called. It is in charge of creating a keyframe for the egg, then the function in block number 2 is called, this one is in charge of searching the points around the egg and store them in an array.

After that it goes back to block one. This cycle creates the keyframes in the range position that the animator specifies.

Now the function in block 3 is called. This time the variables that are going to be used are declared and assigned.

Then in block 4, the function searches for the point that has the lower z position.

Next is block 5, this one trows a virtual ray to determine the point where the egg touches the ground and then it obtains an object with the coordinate in the z axis needed.

The last block, number 6, subtracts the distance that there is in the z position of the egg from the point in the lower z position, and then it adds the position of the plane intersection to place the egg above the plane.

The remaining code is to create a visual interface of the creation of the animation so the animator can use the script.

This Solution's Weak Points

This are the problems that we know this solution has:

1.- One of the problems that will occur is that, as we use helpers, if there is another one in the scene that is not a child of the egg, the script will still consider it in the logic. So this solution is not that accurate because of its dependency in the points and not in the curvature of the egg.

2.- If the ground is too inclined, the egg will pass trough it by its side. But anyway, in real life an egg rotating in a really uneven plane will crash.

3.- It was necessary to use a points envelope around the egg. This can be also be done from the vertex of a similar or equal egg model, but with low polygons. The use of less points to envelop the egg can cause a less realistic rotation and a bumping effect on the rotation, but an use of many points can cause a higher load to the CPU.

4.- The problem with the egg rotation, is that the rotation script uses history dependent calculations and the function used to change the axis and to rotate the egg is recursive. So, in order to obtain the rotation next of the next frame, the code has to recalculate the previous rotations until the beginning once again, in every new frame, because the recursion ends when the time or frame equals zero or when there is no rotation. This can bring a lot of problems when the animation lasts for a lot of frames and the system runs out of memory.

III CONCLUSIONS

We found a solution to the problem of rotating an egg the more realistic way as is could be in a simple way. Even if the solution has weak points, that can be easily optimized. Nevertheless, its a really good tool to make the animation of an egg.

Maybe the recursive nature of the function actually represent a huge load to the computer, and we have to always think in the optimal solutions.

IV REFERENCES AND LINKS

http://www.cuneytozdas.com/tutorials/maxscript/
3ds Max's Help.
C. Woods, A. Bicalho, C. Murray, 3DS Max 4, Anaya Multimedia.