Spiral

```
#include "math.h"
#include "voplib.h"

float easeOutCirc ( float t )
{
    return sqrt ( 1 - ( pow ( ( 1 - t ), 2 ) ) );
}

float index = @ptnum;
float numpts = @numpt;
float startAngle = radians ( ch("angle") );
float dir = 2 * ch("dir") - 1;
float steps = ( numpts - 1 ) / ch("turns");
float stepAngle = ( 2 * PI / steps ) * dir;

float inc = index / ( numpts - 1 );
int mirror = chi("spherical");
float linear = ( 1 + mirror ) * inc;
if ( mirror && index + 1 > numpts / 2 )
    linear = ( 1 + mirror ) * ( 1 - inc );

float circ = easeOutCirc ( linear );
float interp = linear + ( circ - linear ) * ch("roundness");
float r = ( ch("rx") + interp * ( ch("ry") - ch("rx") ) );

//  Apply power to radius at the end (after curvature)
inc = ( ( numpts - 1 ) - index ) / ( numpts - 1 );
float theta = 2 * PI * inc;
if ( mirror && index + 1 > numpts / 2 )
    theta = 2 * PI * ( 1 - inc );
r *= pow ( ch("falloff"), theta );

float angle = index * stepAngle + startAngle;
float x = sin ( angle ) * r;
float z = cos ( angle ) * r;
float h = index / ( numpts - 1 );
float y = vop_bias ( h, 0.5 * ch("bias") + 0.5 );
y = vop_gain ( y, 0.5 * ch("gain") + 0.5 ) * ch("height");

matrix3 xform = dihedral ( { 0, 1, 0 }, { 0, 0, -1 } ) * lookat ( 0, normalize ( chv("n") ) );
@P = ch("scale") * set ( x, y, z ) * xform + chv("t");
```

Revision #1
Created 24 September 2024 04:45:55 by Anthony
Updated 24 September 2024 04:46:15 by Anthony