

Transforms and Junk

1. transforms to attribute matrix:

```
p@orient = quaternion(3@transform);  
v@scale = cracktransform(0,0,2,set(0.0.0). 3@transform);
```

2. rotate packed fracture based on point + distance:

[Screenshot from 2023-06-21 11-48-58.png](#)

```
vector p1= set(@P.x, @P.y, @P.z);  
  
vector crack1 = point(1, "P", 0);  
vector crack2 = point(2, "P", 0);  
vector p2 = crack1-p1;  
vector p3 = crack2-p1;  
float n = fit ( length ( p2 ), 0, ch("maxdist"), ch('mult'), 0 );  
float n2 = fit ( length ( p3 ), 0, ch("maxdist2"), ch('mult2'), 0 );  
  
vector4 q0 = quaternion ( 0 );  
vector4 q1 = sample_orientation_uniform ( rand ( @ptnum ) );  
vector4 q2 = slerp ( q0, q1, n+n2 );  
matrix3 xform = qconvert ( q2 );  
  
setprimintrinsic ( 0, "transform", @ptnum, xform );
```

3. Blending spiral (end beg):

[Screenshot from 2023-06-21 15-48-58.png](#)

```
vector target = point(1, "P", @ptnum);  
float blend = chramp("blendAlongSpiral", @curveu)*chf("multiplier");  
  
@P = lerp(@P, target, blend);
```

4. Position copy via uv:

[Screenshot from 2023-06-21 15-51-53.png](#)

```
v@P = uvsample(1, "P", "uv", @P);
```

5. move near points together:

```
int near = nearpoint(1, @P);  
vector target = point(1, "P", near);  
@P = target;
```

6. Affect the scale of packed prims:

```
//vector scale = fit01(vector(rand(@primnum)), 0,1.46) *@growth;  
vector scale = ch("scale");  
  
matrix3 trn = primintrinsic(0, "transform", @primnum);  
matrix scalem = maketransform(0, 0, {0,0,0}, {0,0,0}, scale, @P);  
trn *= matrix3(scalem);  
setprimintrinsic(0, "transform", @primnum, trn);
```

Revision #8

Created 24 May 2023 15:10:26 by Anthony

Updated 6 March 2025 15:45:55 by Anthony