Project 2: Procedural Animation
Bradley Sakaguchi
VSFX 350

Ascension

I recreated a kinetic sculpture called “Ascension” by Bob Potts. I initially started looking for industrial Machines but found them to be somewhat lackluster. I then looked to some kinetic sculptures to find some pieces with more interesting movement and interconnected parts. While it was hard to find a sculpture that fit these criteria I eventually found Bob Potts. I was interested in “Ascension” because of the smooth movements that were being derived from its complex constraints.

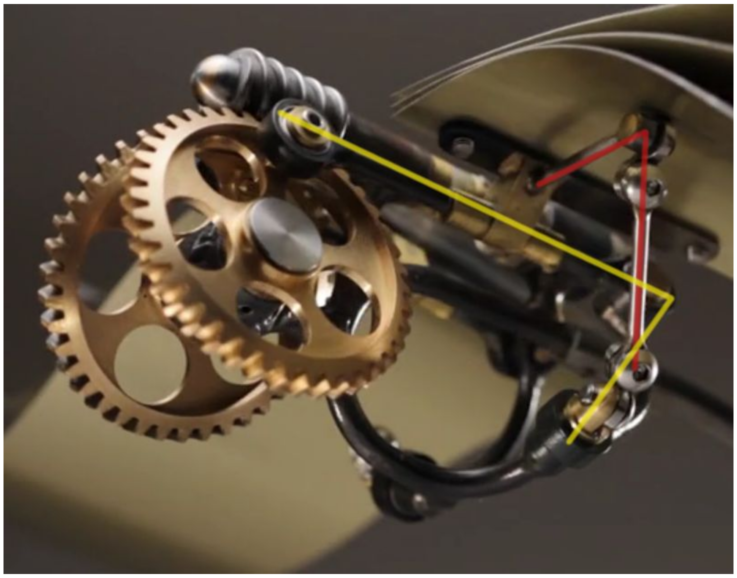
Process

gear

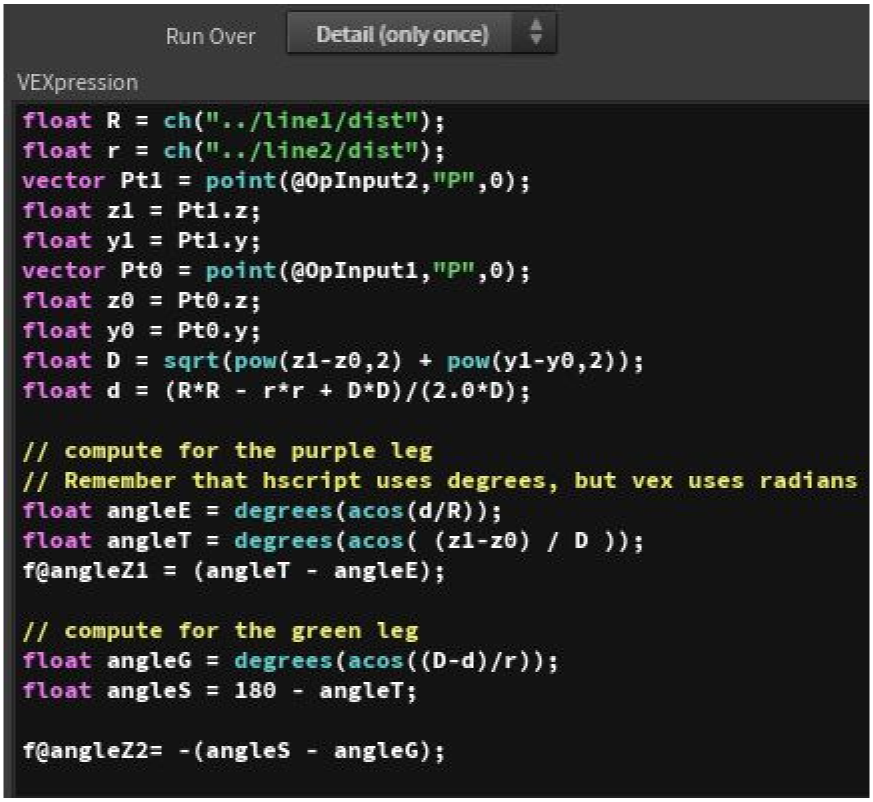
The First thing I went about creating was the “worm and wheel” gear. I stared from a cylinder mod 2 in a group node to select every other face and extruded that group. I found the gear ratio for the worm and wheel was 1 to the number of teeth on the wheel. So my wheel was rotating $F*2 and the wheel had 40 teeth the worm rotated 2*40 = $F*80.

constraints

I then went about creating the 2 joints. (yellow and red)



I was very interested in the Vex 2 point Constraints and they were used in two of the main joints that generated the motion.



Run Over: Detail (only once)

VEXpression

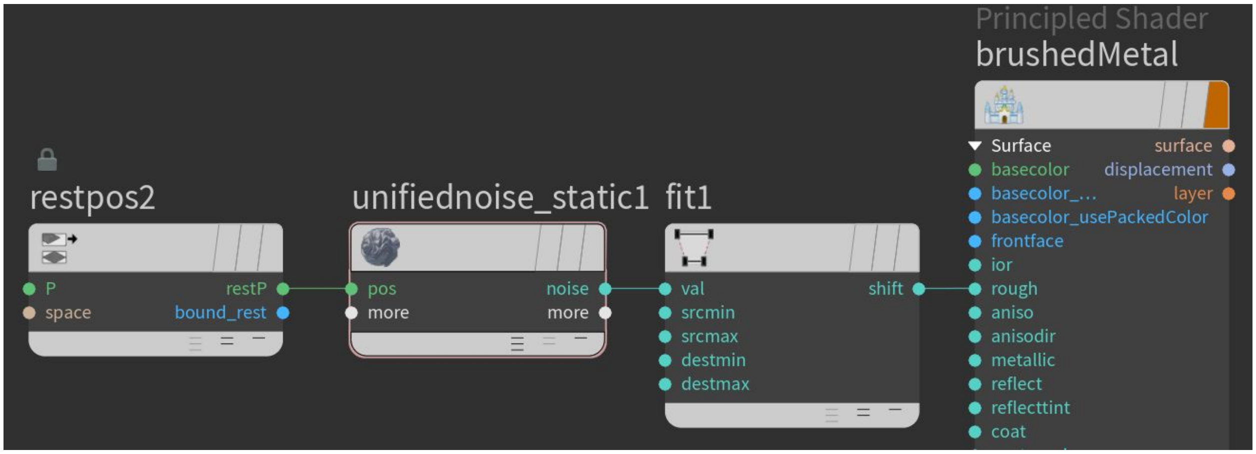
```
float R = ch("../line1/dist");
float r = ch("../line2/dist");
vector Pt1 = point(@OpInput2,"P",0);
float z1 = Pt1.z;
float y1 = Pt1.y;
vector Pt0 = point(@OpInput1,"P",0);
float z0 = Pt0.z;
float y0 = Pt0.y;
float D = sqrt(pow(z1-z0,2) + pow(y1-y0,2));
float d = (R*R - r*r + D*D)/(2.0*D);

// compute for the purple leg
// Remember that hscript uses degrees, but vex uses radians
float angleE = degrees(acos(d/R));
float angleT = degrees(acos( (z1-z0) / D ));
f@angleZ1 = (angleT - angleE);

// compute for the green leg
float angleG = degrees(acos((D-d)/r));
float angleS = 180 - angleT;

f@angleZ2= -(angleS - angleG);
```

For the yellow joint I just modified the vex code given to us to suit the axis that I needed.

The complication came from the red joint. While the yellow joint was along one axis the red Joint was along 2. However the top most pivot point was constrained to only rotate on one axis while the bottom pivot could rotate on 2. This complicated things severely and I had to come up with another solution. I decided to do a normal 2 point constraint along one axis and have that drive the rotation of the top pivot. I then used an add node to stretch the gap between that and the bottom pivot point. This results in a stretching rod instead of a static one son I took some creative liberties and made it into a piston instead.

shaders

The brushed metal shaders were created using distorted unified noises driving the roughness



This was not too hard to achieve the trick was to find rest geometry and make sure that the 3D noise would remain static on the mesh. If I had a peice that did not have a static position I used a time shift node and set it to a static frame to get the rest position.

Improvements

- Realistic red joint Constraint (IK system?)
- Shaders, implement a procedural carbon fiber texture