Puffer Locomotive Procedural Animation Breakdown

Diana Castellvi
March 12th 2019
Houdini Version: 17.0.352

Important statistics:

Average render time: 10.6 min (720HD, 4x4 samples, 8/17 rays)
Number of lights in scene: 1 (HDRI)
Complexity of geometry (approximate):

- Locomotive model geometry: 75,000 polys
- Ground and tracks geometry: 230 polys

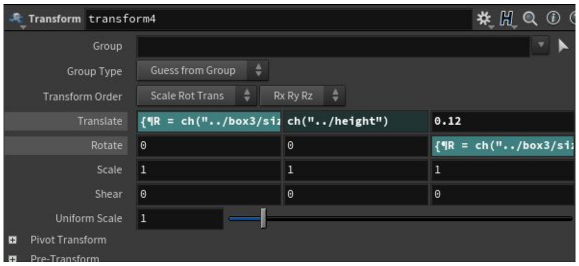
About the project:
The goal of this project was to create a procedural animation in Houdini. I chose to recreate the movement of a puffer locomotive with gears and a two point constraint mechanism. The reference for the animation and model can be found in the following link: https://www.youtube.com/watch?v=UVW494V_AUg

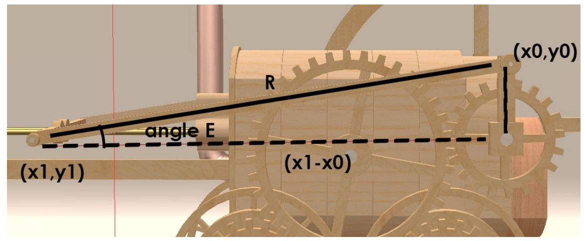
Result vs. Reference Comparison

Procedural animation:

- Two point constraint:
    - There is one two-point constraint on each side of the locomotive, in which the rotation point is one of the gears that moves a horizontal bar in one axis. I used the Pythagorean Theorem to find out in which position the horizontal bar is at each frame and to make the connecting rod between both to follow the rotation of the gear. There

were two expressions used in a transform node connected to the rod, one for the rotation in z and one for the translate in x.





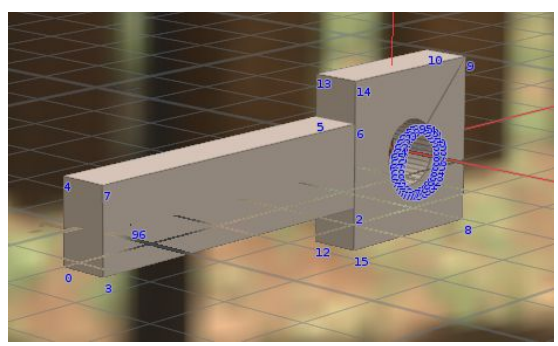
Translate x :

```
{
R = ch("../box3/sizex")-.12;
y0 = point("../transform5",96,"P",1);
# y1 = point("../xform16",40,"P",1);
# to avoid infinite recursion - use the height of the rod
y1 = ch("../height");
changeY = y1 - y0;
changeX = sqrt(R * R - changeY * changeY);
x1 = (point("../transform5",96,"P",0) - changeX);
return x1;
}
```

Rotate z:

```
{
R = ch("../box3/sizex");
y1 = ch("../height");
y0 =
point("../transform5",96,"P",1);
angleE = acos((y1-y0)/R);
return -(270 - angleE);
}
```

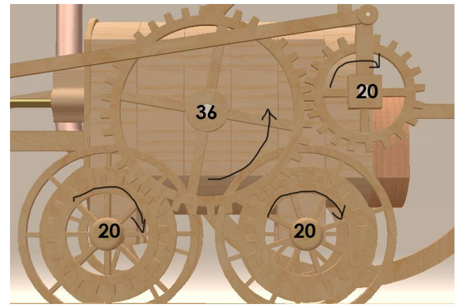
Point y0 is a point created with the add node and the point which the rod connects to.



- Rotation of the gears:
  - Because the whole model is moving in x, the gears/wheels use the formula for finding how long a wheel travels per resolution, which is:

    (car tx at a given time step) / (2 * PI * r) * 360

- ○ To make the gears rotate correctly we have to give rotation to one and then use that rotation and the teeth of the gears to find the correct rotation of all the gears.



To find out the rotation of the big gear of 36 teeth, the formula I used was:

-(ch("../transform1/rz")*(20/36))

I'm referencing the rotation of the gears/wheels that move the locomotive and I'm multiplying it by the number of teeth of the reference gear divided by the number of teeth of the gear we want to find the rotation for. The negative sign is to make it rotate in the correct direction, which is the opposite.

Since all the other gears are rotating in the same direction and have the same number of teeth, all of them will have the same rotation.

Problems encountered and solutions:

- Modeling: The only problem I had was that the model has a lot of small pieces and a couple of complex shapes, which was causing me to spend a lot of time modeling. That is why I decided to use the trace node to create some of the geometry from images I created in Photoshop. This approach really made it easier for me to model everything more efficiently.