CIS 462/CIS562 Computer Animation (Fall 2018)
Homework Assignment 8

(Particle Systems)

Part III – Houdini-based Fireworks Simulation

Step 0:

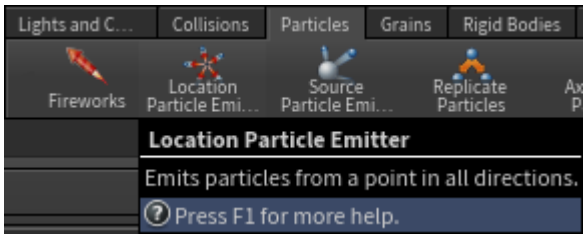
Go to http://www.sidefx.com/ and download the free Apprentice version that is available. The Apprentice version will insert a small watermark in your renders and limit the render size, but that is fine for this assignment. Houdini Apprentice is available for Windows, Mac, and Linux. This tutorial was written for version 16.0.600.

Step 1:

Familiarize yourself with the UI. Houdini has several quick start tutorials on their website here. You should learn to navigate the scene, use the parameter and network views, and create nodes.

Step 2:

Let's create a particle emitter for the firework projectiles. Open the Particles shelf and click Location Particle Emitter. Hover your mouse over the viewport and press Enter to accept the default placement at the origin.



Step 3:

Press play! By default, Houdini will play as fast as it can. To change playback to realtime, click on the clock at the bottom right:

Notice that the timeline will turn blue to indicate that the frames have been cached. If it's orange, it's because you changed the simulation partway through an existing simulation, so you should restart your playback from the first frame.

Step 4:

We should now be inside the AutoDopNetwork in the network view. Take a look at the nodes. Middle-mouse-click on any of the nodes to learn more.

- The popobject1 node represents the object we see in the viewport.
- The merge nodes simply merge multiple streams of particles.
- The popsolver1 node solves the forces on the particles and updates the particles' attributes.
- The gravity1 node applies gravity to all our objects.
- The location node generates the initial values of all the particles.

Step 5:

First, let's create the shell particles using the location node. These will represent the packaged fireworks *before* they explode into colorful sparks. Inside the parameter view, we'll need to change some parameters for the particles to be more like the shells.

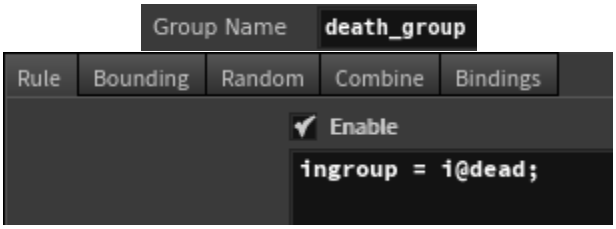
- Hover over the location node's attributes In Houdini to learn more.
- Birth >> Constant Birth Rate – Lower this so that we only have several fireworks at once.
- Birth >> Life Expectancy – This represents the time until the particle dies. We'll use it so that the fireworks explode when this happens. Adjust accordingly.
- Birth >> Life Variance – Add some randomness to make the simulation more realistic.
- Attributes >> Velocity – This represents the initial velocity of the particles. Change this so that particles are shot up into the air.
- Attributes >> Variance – Change this to add variety to the initial velocities of the particles.

With enough tweaking, you should have a few shell particles shooting up every second and dying when they reach near their apex in the air.

Step 6:

Now we want to have the firework shell explode into many more spark particles when it dies. To do this, we can use particle groups, which allow us to reuse these dead particles elsewhere.

We'll create a group of the dead particles that we can use to spawn additional particles. Add a POP Group node in between the location node and the merge2 node. In the parameter view, give the group a name like "death_group". Then, enable the Rule tab so that we can filter out the particles using VEX, Houdini's own special language for running custom code. Learn more here.

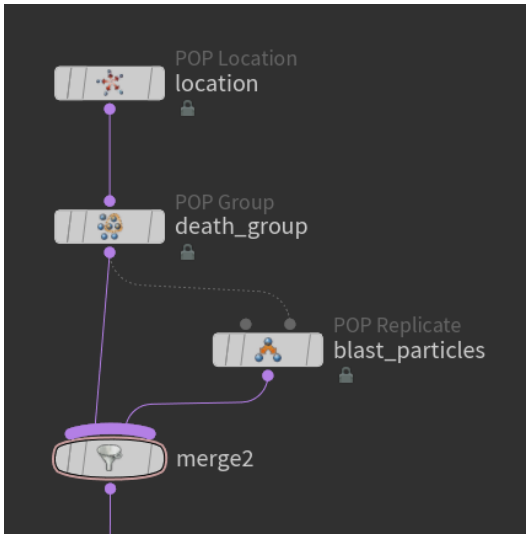


Set the rule to "`ingroup = i@dead;`" Notice the semicolon. This is roughly equivalent to "`bool ingroup = isdead;`" The expression is tested on each of the particles from the location node, and if `ingroup` is true, the particle is added to the "death group". Rename the POP Group node to "death_group" so that it's easy to remember the group name.

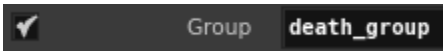
Step 7:

Now we want to duplicate each dead firework shell into many, many spark particles. Create a POP Replicate node. We'll rename it to blast_particles. Create a connection from the output of death_group

to the Reference Stream input of blast_particles. This essentially creates a separate particle stream that we can use for blast particles only. Then connect the output of blast_particles to merge2. Remember you can hover over an input to see what it's for.



We only want to duplicate death_group particles here, which represent firework shells that are ready to explode. So, check the Group checkbox in blast_particles's parameter view, then set Group to "death_group". If you play now, you'll notice that the explosion is pretty dull.



Step 8:

Before we change the settings of blast_particles, we want to apply an air drag force to the particles for realism, which can affect our parameters. Add a POP Drag node in between blast_particles and the merge2 node.

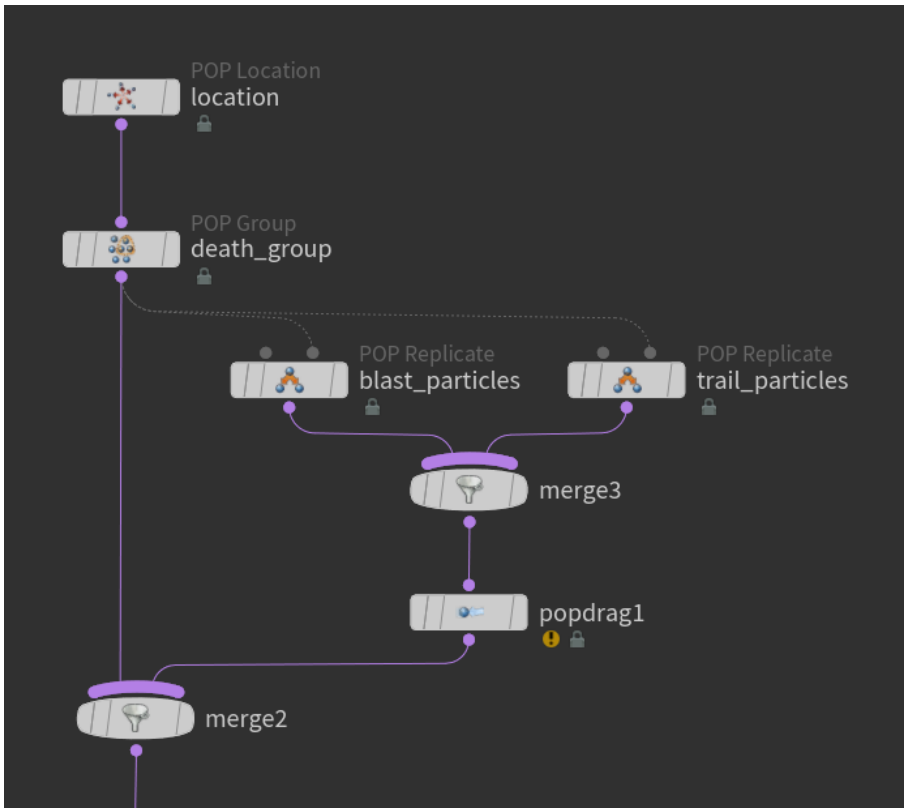
Now we can change some parameters in blast_particles:

- Birth >> Constant Birth Rate – Set this to 0. We'll use impulse activation here since particles only have one frame to spawn.
- Birth >> Impulse Count – Adjust to your liking. This is the number of particles spawned each time the replicate node is run on a particle.
- Birth >> Life Expectancy – Set the lifetime of the sparks to something realistic.
- Birth >> Life Variance – Add variance to the lifetime for more realism.
- Shape >> Shape – Change to "Point" so that the sparks spawn from the original point's location.
- Attributes >> Initial Velocity – Change to "Add to inherited velocity" to mimic an additive explosive force since sparks inherit their shells' velocities.
- Attributes >> Variance – This represents the scale of the explosion, the randomness of the direction and magnitude of each spark particle. Adjust to your liking.
- Feel free to adjust the amount of air resistance in the drag node.

Step 9:

Let's add a trail of smoke (or spark) particles to the firework shell. Add another POP Replicate node named "trail_particles" and connect the output of death_group to the Reference Stream input of trail_particles. Don't change the group settings for this node though! We always want the shell particles to emit a trail, not just when they die.

We also want to apply our drag force node to both blast_particles and trail_particles, but popdrag1 will only accept one particle stream at a time. To work around this, create a new Merge node and connect blast_particles and trail_particles to this node. Now, we can connect the merge node to the drag node.



We could have connected trail_particles to the location node instead of the death_group node and gotten the same results. The only difference would be that trail_particles wouldn't know about the death_group.

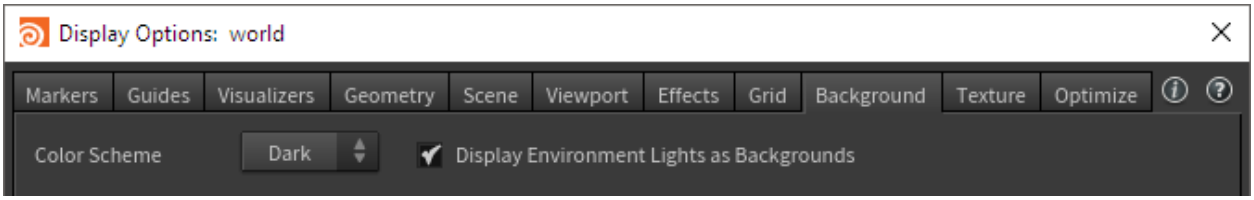
You should now be able to adjust the familiar settings in trail_particles to achieve the trail effect that you want. Lower the value of Attributes >> Inherit Velocity to mimic expulsion forces from the firework. This is because the expulsion forces on the trail particles negate the inherited velocity from the shell particles.

Step 10:

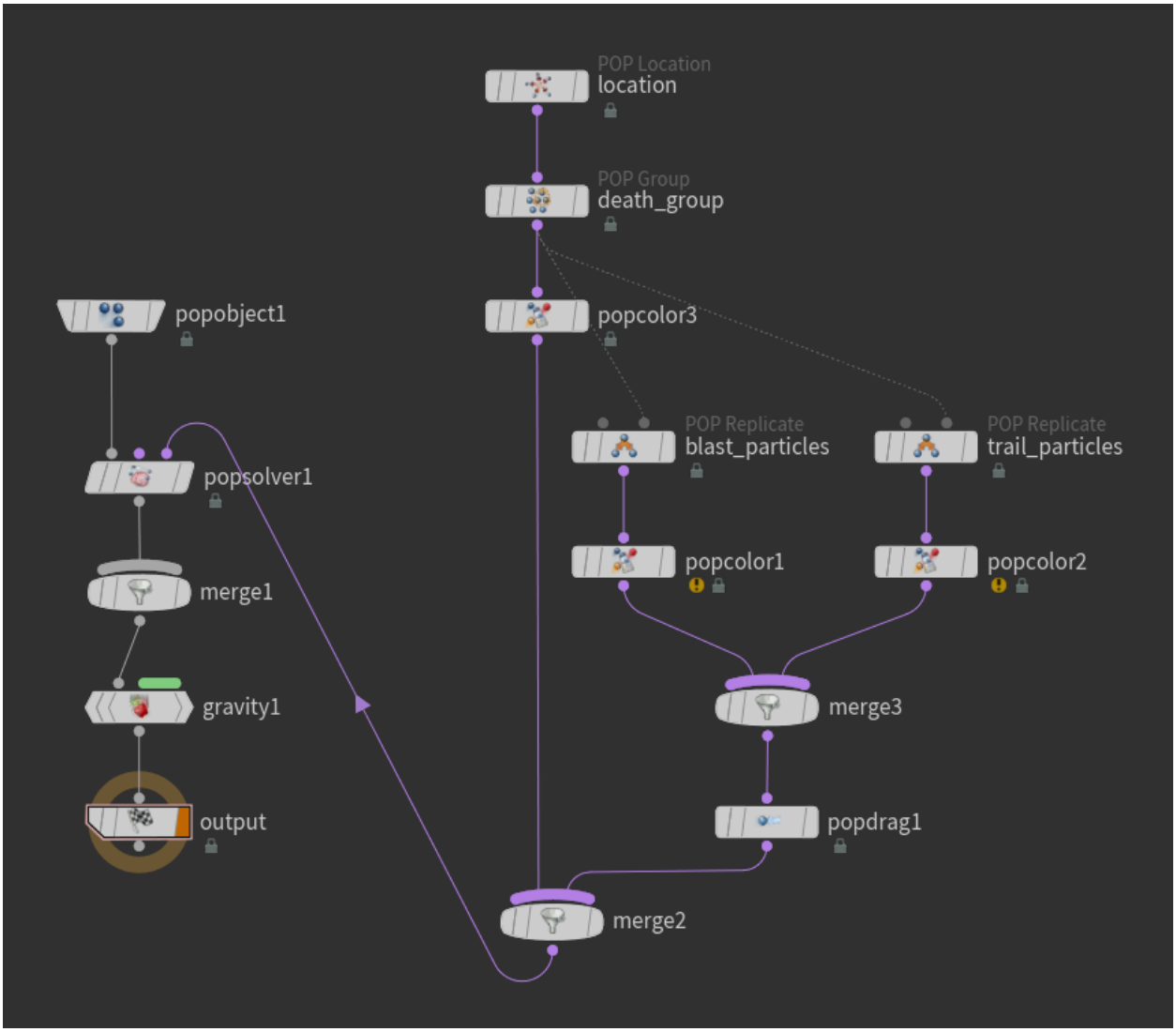
Your fireworks should look much better now! Add a POP Color node to each of your three particle streams to give the particles some vibrant color. You can set Color >> Color Type to Ramp to change the color of the particle over its lifetime. Notice that the VEXpression is set to "`ramp = @nage;`" to

achieve this. `@nage` is the normalized age from 0 to 1. Play around with other color types and VEXpressions!

The default blue background makes it difficult to judge the colors. You can change the viewport color by hovering your mouse over the viewport and pressing D. This should open the Display Options. Change Background >> Color Scheme to Dark.



You're finished! You should now have a realistic firework simulation. The resulting AutoDopNetwork should look something like this:



Step 11:

The final task is to render the simulation. Follow these steps:

- Right-click on the flipbook on the bottom left of the viewport: 
- Click “Flipbook with New Settings…”
- Change the size of the render to be less than or equal to 1280x720, which is the maximum allowed render resolution for Houdini Apprentice.
- Click “Accept”. This will render what you see in the viewport into Houdini’s MPlay player, much like Maya’s Playblast feature.
- If you are on Mac:
    - File >> Export >> QuickTime Movie Exporter
    - Save and rename the resulting file to HW8_LastnameFirstname.mov
- If you are on Windows or Linux:
    - Exporting to video doesn’t seem to work with Houdini Apprentice.
    - Click “File >> Save Sequence As...”

- Change the export filename to something like imageName$F.tga. $F is the VEX variable for the frame number. Targa files work well with most video encoders.
- Click “Save”. This will save the rendered image of each frame of your simulation.
- Use your favorite video editor to convert the image sequence. If you don’t have one, download and install FFmpeg, an open-source video encoder.
- Run the command: `ffmpeg -r 24 -f image2 -i imageName%d.tga -vcodec libx264 HW8_LastnameFirstname.mp4`

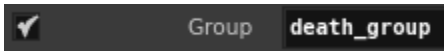
Congratulations on your work with Houdini!

Troubleshooting

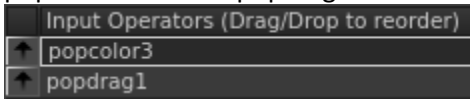
- If Houdini stops responding
  - You may have too many particles. Lower your birth rates.
- If your simulation is missing some parts or is not appearing at all
  - Make sure that the output node has the output flag set

- If on Mac, your flipbook isn’t rendering
  - You may need to allow Houdini through your Mac’s firewall so that Houdini can send itself its renderings.
  - In System Preferences >> Security & Privacy >> Firewall >> Firewall Options…
  - Then allow incoming connections for Houdini FX and Image Viewer.
- If your POP Color nodes have “Invalid Group” warnings
  - This is usually normal because there aren’t any particles to make groups from. However, if it persists, continue reading.
  - Check for any typos or invisible characters in “death_group”.
    - Make sure this is inside of the POP Group node's "Group Name" attribute.

    - Make sure this is inside blast_particle's "Group" attribute with the checkbox next to it enabled.



  - In the merge node closest to the POP Solver, make sure that the stream coming from the death_group node is first.
    - In our example network, if you click on the merge2 node, you should see that popcolor3 is above popdrag1. Reorder them to get something like this:



    - The rationale is that the primary stream (i.e.: the location stream) should run first so that any particles that die will be kept for the reference streams (i.e.: the trail and the blast particles).