



KISS Pyro - Simple Workflows

Building Simulations in SOPs and building workflows.

AGENDA

KISS PYRO

AGENDA

Building Simple Pyro Setups inside SOPs

Building Sources, colliders and pumps using VDB Volumes

File Cache SOPs as explicit cache points

Building simple HDA to encapsulate simulations for use anywhere

ROP Network to automate results

Building Simpler Simulations

The current Shelf workflow for setting up simulations is effective in getting results quickly. It provides a convenient way of providing preset setups using various constraints anticipating a great deal of different input conditions.

But if we reduce the number of variables for input conditions, we can greatly simplify the simulation setups.

If you provide all source geometry as cached alembic or bgeo sequences with animation baked out for all sources, colliders, sinks, pumps and more, you can build the entire simulation system inside a single Object node. All workflows would be with SOPs.

KISS or Keeping It Simple Stupid is in play here in many ways.

Building Simpler Simulations | Shelf Tools

So what about the Simulation shelf tools?

The shelf itself is a fantastic UI tool. The key purpose of a shelf tool is to offer the artist a simple way to execute a script. With clever design, the tool builder can create an entire workflow.

This is the case with the current state of the various simulation shelf tools. Focusing in on Pyro, the various preset tools indeed take source geometry and construct various network setups with preset values to achieve a nice result quickly.

They are maintained from release to release.

The Simulation shelf tools offer us quick access to values that generate good results.

So what's the issue?

Building Simpler Simulations | Shelf Tools

How to improve current Simulation Shelf tools?

I would argue, nothing. They work. Very well. But like everything in Houdini, they are there to inform designers what works and more importantly, a blank canvas for us to build in our own workflows.

Some areas that we can address to KISS?

- Reduce number of nodes placed all over Houdini.
- Reduce number of shelf tool steps to set up simulations.
- Clarify cache steps and improve Simulation automation
- Make simulations that work in Houdini Core

KISS Simulations | Reduce Nodes

We can reduce the number of nodes when building up Simulations.

The current shelf tools presume that the simulation inputs could be derived from animated Houdini objects and thusly create Object Merge SOPs inside new objects to:

- Decouple the Simulation from any objects in the scene.
- Handle any object level transforms by using Object Merge SOP to crack transforms and remove any scene dependent constraints.
- A way to maintain the integrity of Objects in the scene by leaving them relatively untouched.

That last point is not quite true. We pollute existing objects with additional SOPs to pre-process the geometry. For example when using shatter setups for Bullet RBD simulations, we pollute the existing objects with the bullet fracture setup.

KISS Simulations | Reduce Nodes

To reduce the number of nodes, one strategy would be to assume that the artist provides geometry caches to the simulation. The disk caches provide

- First class sources to the Simulation that are baked and devoid of scene constraints.
- Efficiency in that no scene evaluation needs to take place.
- No more object in the scene to represent the geometry.

The last point where there are no representative objects in the scene is discomfoting as you need to solve the issue of providing everything that is required for that geometry to render: Materials, Object Properties, etc.

But in the real world, this is done in layout where the various disk caches are loaded in to a final scene. By KISS the simulation setup and execution, the overhead is reduced to just that.

KISS Simulations | Reduce Steps

We can reduce number of shelf tool steps to set up simulations.

Shelf setups for simulations are designed to support ad hoc creation of simulations. It would be challenging to package up simulations authored from the existing shelf tools to support tasks such as:

- Simplifying the setup so that the creation of effects shots is more accessible to a broader audience.
- Shield artists from working explicitly with DOP networks and the various Micro-Solver DOP nodes.
- Sequence Shot production where you apply the same setup to varying input constraints.

Working in SOPs

KISS Simulations | Simplify Caching

With the default simulation shelf setups, the various locations for caching geometry are located throughout the hierarchy. Depending on the various sources, caches can be located in several objects, inside the DOP network itself and the additional objects generated to grab the simulation data to cache and render.

KISS Simulations | In Houdini Core

By building simplified wrappers around the various DOP networks and setups that are available in SOPs Geometry context, you simplify the setup.

By working up the simulation in SOPs Geometry, you now can build up simulations in Houdini Core. By exposing the various construction bits on the simulation HDA and by providing other tools to create first class imports for sourcing, collisions and more, you can build up a very effective workflow to support sequence type effects and to ensure consistency between shots.

qLib TOOLS

A PROCEDURAL ASSET LIBRARY FOR SIDEFX HOUDINI

[HTTP://QLAB.GITHUB.IO/QLIB/](http://qlab.github.io/qlib/)

QLIB

qLib is a library of very useful set of production proven Houdini tools. Tools to help visualizing data, complete fast deformer suite, very fast variants of key SOPs and workflow related tools.

The Shot tools offer us a very quick way to implement production workflows with little effort. We will be looking at:

Shot qL Object node

Shot Builder Object node

Environment qL Object node

qLib TOOLS

A PROCEDURAL ASSET LIBRARY FOR SIDEFX HOUDINI

[HTTP://QLAB.GITHUB.IO/QLIB/](http://qlab.github.io/qlib/)

QLIB INSTALLATION

Armed with a proper text editor, and yes I use vim (vi improved!) but you can get by with any plain text editor such as Windows Notepad, Mac textEdit as long as you save as ascii plain text, and on Linux use any suitable raw text editor.

Go to the qLib website to see the install instructions. Starting out with houdini environments, edit the houdini.env file found in your \$HOME houdini directory. Just copy and paste the lines from the qLib install notes in to your houdini.env file and re-launch Houdini. Don't forget to set the proper install directory path in houdini.env where indicated.

On Windows the location is:

`$HOME\Documents\houdini16.5\houdini.env`

On Mac the location is:

`$HOME/Library/Preferences/houdini/16.5/houdini.env`

On Windows the location is:

`$HOME/houdini16.5/houdini.env`