Steamshovel

Advanced Tricks

Steve Jackson

IceCube Collaboration Meeting
11 October 2013

Steamshovel

A scriptable graphical tool for visualizing and exploring IceCube event data.

- Replaces glshovel
- Recently added to offline-software trunk
- Goal: Any (non-programmer) IceCube person can use it

Steamshovel

A scriptable graphical tool for visualizing and exploring IceCube event data.

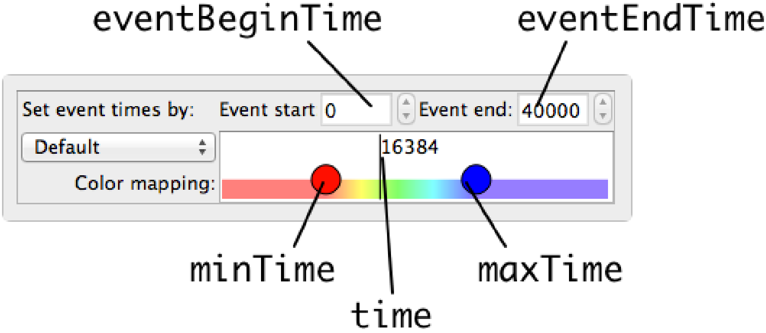
- Replaces glshovel
- Recently added to offline-software trunk
- Goal: Any (non-programmer) IceCube person can use it
- Further goal: Advanced icetray users can use it *cleverly*.

In this talk: a few ways to be clever.

In-program Scripting

Most things controllable with the steamshovel GUI have scripting equivalents.

For example, the following can be read and written as properties of `window.timeline`:



eventBeginTime

eventEndTime

Set event times by: Event start 0 Event end: 40000

Default

16384

Color mapping:

minTime

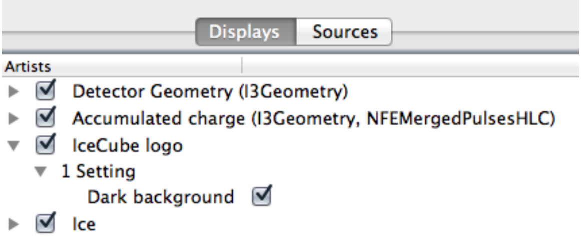
maxTime

time

These are documented in doc file `scripting.rst`.

In-program Scripting

Artists can be added, removed, and configured via scripting as well.



Artists

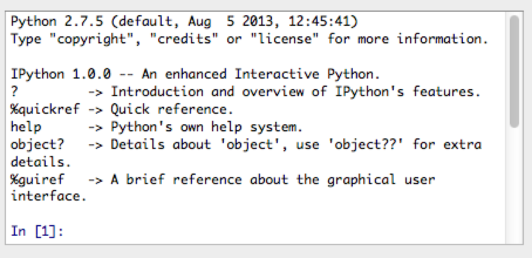
- ▶ ☑ Detector Geometry (I3Geometry)
- ▶ ☑ Accumulated charge (I3Geometry, NFEMergedPulsesHLC)
- ▼ ☑ IceCube logo
  - ▼ 1 Setting
    - Dark background ☑
- ▶ ☑ Ice

To see your current settings as a script, run:

```
from icecube.steamshovel.utils
             import scenariotools
print scenariotools.exectableScenarioDump()
```

Python interaction

Where available, the embedded IPython console is part of the Steamshovel GUI.



```
Python 2.7.5 (default, Aug  5 2013, 12:45:41)
Type "copyright", "credits" or "license" for more information.

IPython 1.0.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra
details.
%guiref   -> A brief reference about the graphical user
interface.

In [1]:
```

This console is synchronous with the Qt event loop, so if you call `time.sleep(5)`, then the program will just hang for 5 seconds.

However...

Python interaction

```
> steamshovel --help
```

```
Command-line options:
--console      Use the shell console instead
               of the interactive GUI console
```

...the shell-based console is asynchronous with Qt events. Calling `sleep()` will not halt the GUI.

This can be used to write display scripts that change their behavior over time, e.g. show one set of reconstructions, wait ten seconds, then change to a different set.

Useful for non-interactive event displays?

Network Data

As of last month, Steamshovel can read i3 files over a socket:

```
   > steamshovel socket://localhost
```

nathanw's file server: `cat myfile.i3 | nc -l 1313`

dshultz's file server: coming soon

Networked files can be loaded alongside local files (e.g. use a local GCD file). The network file must be the **last** one loaded.

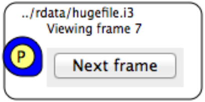
Mimicking Network Data

```
> steamshovel --help
```

```
Command-line options:
-i [ --incremental ]  Treat the last specified
                      file as an incremental
                      source, like a socket.
```

This debugging option pretends the last file on the command line is a socket, and loads only one frame at a time.

This allows much faster loading of large compressed files.



../rdata/hugefile.i3
Viewing frame 7

Automation Options

```
  > steamshovel --help
```

```
  Command-line options:
    -s [ --script ] arg   Script to execute
    --batch               Batch execution: do not
                          display a window
```

Combine `--batch` and one or more `-s` scripts for fully automatic, command-line screenshots.

Steamshovel requires X forwarding to run over ssh, even with `--batch`.

Writing Custom Artist Types

To make your new artist available to the program:

```
import my_artist_module
window.gl.scenario.registerArtist(
    my_artist_module.MyArtist )
```

Steamshovel will remember this new type between sessions.

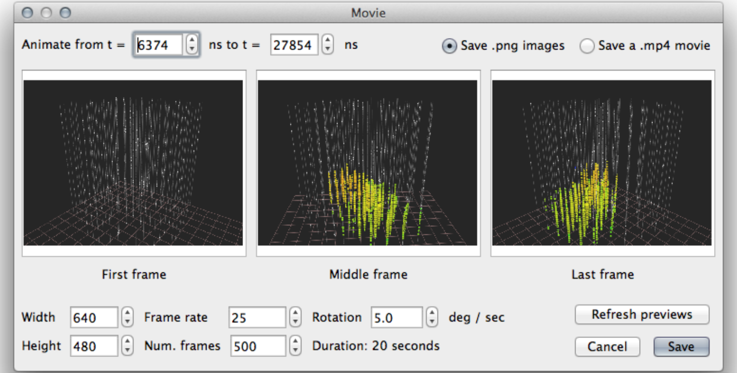
If you edit your artist while Steamshovel is running, then:

```
reload( my_artist_module )
```

Newly created MyArtist objects will be up to date. Much faster than restarting Steamshovel for every change you make!

Making Movies

New GUI dialog for creating movie files:



Movie

Animate from t = 6374 ns to t = 27854 ns ● Save .png images ○ Save a .mp4 movie

First frame

Middle frame

Last frame

Width 640 Frame rate 25 Rotation 5.0 deg / sec

Height 480 Num. frames 500 Duration: 20 seconds

The movie generation utility is scriptable, too! Again, see `scripting.rst`.

Making Awesome Movies

Movie generation also provides a python hook function:

```
   window.movieEngine.setPreframeHook( f )
```

The python object `f` will be called before each frame of the movie is rendered. Use this for:

- Complex camera motion
- Programmable animation speed
- Changes to artist settings during movies

You could also make a single movie that spans multiple events this way.

That is all!

Please send questions and feedback to

`sjackso@icecube.wisc.edu`

Thanks to Steamshovel's many testers and contributors.

Thanks to Don for presenting.