FL Tiled Display App

June 2013

Installation

A tarball of all of the source code is here:
http://www.alcf.anl.gov/research/fl/flinternal/MISC/TILED_DISPLAY_SOFTWARE.tar.gz
It contains

- flTile
- flapp

which are required, and

- flWii
- airstream

which are optional. This should run fine without these last two, but I left them in in case you want to poke around at them.

Dependencies

These modules are available through the package manager on many Linux distributions, though they may not be for raspbian. These modules certainly have their own dependencies, too, which the package manager will fetch for you. Hopefully they are satisfied on the Pi!

pygame (http://www.pygame.org)
pyffmpeg (https://code.google.com/p/pyffmpeg)
    Requires: libavcodec, libavformat, libavutil
pyOpenGL (http://pyopengl.sourceforge.net)
Python Imaging Library (PIL) (https://pypi.python.org/pypi/PIL)
pyMPI (http://sourceforge.net/projects/pympi)
Twisted (http://twistedmatrix.com) (may be optional)
MPI

You might see references in the code or error messages to an “airstream” module or a “bluetooth” module, but the app should succeed without them.

Configuration

The layout of hosts and displays is specified in a configuration file that uses Python syntax. A sample configuration file can be found here. The configuration describes the structure and relationship between parts: a configuration contains machines, and a machine contains tiles and windows.

Execution

There is a tarball that contains a movie, and chopped up movie here:
http://www.alcf.anl.gov/research/fl/flinternal/MISC/TILED_DISPLAY_MOVIE.tar.gz (2.2GB)

In the flTile directory there is a script:
am8screens.sh

You'll need to edit this script to point to your machinefile, and your installation of pyMPI.
This script also points to a config file like the one referenced above. These live in the **configs** directory under flTile. (The config file should have a .py extension, but that is omitted when naming it in this script.)

To play a movie, you run:
./am8screens.sh /path/to/<media> [--scale=1.3x1.3] [--pos=300x250]

Where <media> can be either an individual movie (movie.mp4, etc.) or a chopped movie (movie.tdmovie, which is a directory which contains movie files of the individual chunks, and a description.txt file which gives the layout info for the chunks. See the example in the above TILED_DISPLAY_MOVIE tarball.) The optional scale and pos options take xy values for scaling and positioning the movie on the display. For position, the positive y axis is directed upward.

I believe that <media> can also be an individual image, or the path to a directory with multiple images. If a directory of images, they will get stacked, overlapping one another. This is where the flWii and airstream stuff would come into play, which would allow you to pick images up and move them around, etc.

Somewhere in all of this I believe is the code for running in streaming mode as well, where it catches movie streams pumped out by vl3 and stitches them back together. You probably know more about that than we do.

AM Movie Creation

How to Create a Tiled Display Movie for the AM.

******* This documentation is likely out of date, so use it more as a guide than actual step-by-step instructions. There was some code added to the scripts to enable them to be run in parallel, but those updates are not documented here *******

These directions are for the python tiled display application.

If the movie is not large and does not span the display, you can trying to load movie as you would any normal image file:

```
sh mpiTileArg.sh mymovie.mp4
```

For large movies, having the tiles only decoding and rendering part of the movie improves performance.

The creation scripts and an example are in the flTile/scripts directory.

`( /home/eolson/am-macs/src/flPy/flTile/scripts )`

The example_create_movie.txt contains these directions:

The first step is not necessary if individual frames are already available.

1. movieToImages.py mymovie.mp4 mymovieframes
2. chopImagesToImageTiles.py mymovieframes mymovietiles 128x128
3. imageTilesToTiledMovie.py mymovietiles mymovie.tdmovie

The 128x128 is the size of the tiles to create.

If you don't run the first step, the frame filenames for the second step are expected in this format: anyname_00000000.png

---

Here are a couple more notes on how to do this on the am-macs:

I generally do the creation in the sandbox on am-mac2 (about 18GB currently free)

1. copy my video file (or video frames) to /sandbox/eolson
2. Run the lines above to created a .tdmovie
3. copy the resuling .tdmovie to either all sandboxes or the shared home directory (I assume it loads faster when it's in /sandbox)

Here's how I copy a movie to all sandboxes: python ~eolson/am-macs/scripts/run.py "scp -r am-mac2:/sandbox/eolson/mymovie.tdmovie /sandbox/eolson/movies" It just runs scp on all machines. Note the slightly different directories so it's not trying to overwrite itself on am-mac2.

- Run as normal: (on am-mac1)

```
/home/eolson/am-macs/scripts/gotodemo
 sh mpiTileArg.sh /sandbox/eolson/movies/enzo.tdmovie
```