General Description of `xscript`
Revision: 1.10

Christian Queinnec
Université Paris 6 — Pierre et Marie Curie
LIP6, 4 place Jussieu, 75252 Paris Cedex – France

`xscript` is a script-based video capturer of windows.

`xscript` is based on XVidCap 1.0.9[1] by Rasca Gmelch, Berlin 1997-1999, who published XVidCap under the GNU GPL. `xscript` is, of course, published under GNU GPL [2] (see the COPYING file).

The very small modifications brought to XVidCap were written by Christian Queinnec[3] and by Choun Tong Lieu [4].

Remember `script` ? That utility used to record a terminal session. `xscript` is similar except that it captures a window receiving everything emitted by a script file. This tool is used to enrich an educational CDROM with animations showing how to use `gcc`, `gdb` and other tools. `xscript` is part of the VideoC project[5].

1 `xscript` options

`xscript` is a strict superset of `xvidcap` and supports some new options on the command line.

- **–script *file*** tells where is the script file. I used *.xs* as suffix for `xscript` files.
- **–result *file*** tells where to produce the information describing the images that are captured. By default, this the name of the script file with a *.t* suffix.
- **–window *windowid*** tells which window to grab (use `wininfo` or `xlsclients` to know it). You may also select that window with the mouse as in xvidcap.
- **–delay *number*** starts to execute the script after *number* seconds. This leaves time to set up things if needed. A script should have been previously specified. By default, the script is executed when you press the 'start' button of `xvidcap`. The delay may also be obtained with the `@sleep` command from the script file.
- **–offset_frame** tells `xscript` where to position the control panel above the grabbed area. This is because I noticed on some window manager that the control panel often overlaps the grabbed area.

`xscript` may be run without user intervention (with the previous options) or with full user interactive support (the user selects the window to grab then press the 'start' button). The following options of `xvidcap` are still useful.

- **–fps *number*** is the time lag between any two keys sent to the window to grab. It is as if you typed those keys with that speed. A value of 1 or 0.5 is appropriate. You may change this value dynamically with the `@fps` command.

[1] `http://home.pages.de/~rasca/xvidcap`
[2] `http://www.gnu.org/copyleft/gpl.html`
[3] `mailto:Christian.Queinnec@lip6.fr`
[4] `mailto:Choun-Tong.Lieu@liafa.jussieu.fr`
[5] `http://youpou.lip6.fr/queinnec/VideoC/VideoC.html`

- –**verbose** be verbose of course
- –**file** ***pattern*** to specify the pattern of the produced files. By default, `xvidcap` uses `frm-%04d.xwd`.

2 `xscript` script files

The script is a simple text file whose characters are sent one after another to the grabbed window. Just before a character is sent, the window to be grabbed is captured and an image file is produced (you may see its number increment in the control window of `xvidcap`). There is no sensible way to send a character, waits for its effect and then automatically grabs the window. Therefore the window is grabbed before sending the character rather than some time after. If you know how long to wait, you may use the use the `@sleep` command (see below).

There are two special characters. No capture is done before processing special characters.

Backslash (\) means"send the next character whatever it is". It is therefore possible to send a backslash or an arrobas.

Arrobas (`@`) prefix a command that ends with the next newline (lines are limited to 1Kbyte though). A command often starts with a word possibly followed by some other arguments. Extra stuff after the last argument is completely ignored. There are a number of commands.

- `@` immediately followed by a newline is a 'no-op' command, it does nothing.
- `@#` *`words`* this is a comment. This comment is ignored.
- `@\`*`characterSpecification`* this command sends the character that follows with the appropriate combination of control, meta, shift modifiers. Three modifiers are recognized `\M-` for meta- (or alt-), `\C-` for control- and `\S-` for shift-. The order of modifiers is meaningless. One example is `@\M-\C-a` to send alt-control-a.
- `@capture` requires explicitly the capture of the window to be grabbed. Normally no capture is done before a command is processed and no capture is done while in the midst of a series of commands. Captures are made only before a character is sent to the target window.
- `@nocapture` stops capturing windows until a subsequent `@capture` command. This command allows to send batches of characters and capture windows on demande only.
- `@sleep` *`numberOfMilliSeconds`* tells `xscript` to sleep for that number of milliseconds.
- `@sh` *`shellCommand>`* runs the shell command as a side-effect.
- `@echo` *`words`* outputs these words as they are into the result file (more on this file later).
- `@subtitle` *`words`* outputs these words in the result file as a subtitle i.e., as a line starting with the word `subtitle`. This command is similar to `@echo subtitle` *`words`*.
- `@sync` *`numberOfMilliSeconds`* tells `xscript` to wait until that number of milliseconds has passed. This time is counted from the beginning of the script. This is used to synchronize with some external event.
- `@fps` *`numberOfFramesPerSecond`* tells `xvidcap` to change the delay between any two captures. This is as if you press the next keys with that new speed.
- `@quit` tells `xscript` to stop the capture and exit.

A small example is in order. This script shows simple use of shell commands. Script files often have the *.xs* extension. This script file appears in the distribution as *cdls.xs*.

```
@# $Revision: 1.10 $     to identify the script
@fps 1                           # 1 frame/second
```

```
cd /tmp
ls
@sleep 5000                       # ie 5 seconds
cat > /tmp/coucou.tmp
coucou
@\C-d                             # ie a ^D
@echo sound "commentLS.au"
@subtitle See how many files there are in that directory!
@# When playing this result file, you will hear the sound
@# file and see a running subtitle under the image.
@capture                          # the last image and
@quit
```

3 `xscript` result files

`xscript` produces a result file that contains some information on the generated files as well as other information produced by `@echo` directives from the script. All lines are dated (in milliseconds from the start of the script).

For instance, the produced file for the previous script will contain lines such as:

```
image "cdls0000.xwd" at 4
image "cdls0001.xwd" at 1016
image "cdls0002.xwd" at 2026
image "cdls0003.xwd" at 3036
image "cdls0004.xwd" at 4046
image "cdls0005.xwd" at 5056
image "cdls0006.xwd" at 6066
image "cdls0007.xwd" at 7076
image "cdls0008.xwd" at 8086
image "cdls0009.xwd" at 9096
image "cdls0010.xwd" at 10106
sound "commentLS.au" at 16126
subtitle "See how many files there are in that directory!" at 16126
image "cdls0011.xwd" at 16126
```

You may now combine these images into an animated gif but without the sound files. Currently, I record the sound in separate files and I may work on a player to run these result files. I will probably write some day a converter to the SMIL language. Since I did not find a tool to produce a *.avi* file, I wrote a perl script to produce an html page with animated gifs; the subtitles are turned into text put besides the animated gif and followed by a link to the sound file. This is a poor man's animation but better than nothing.

Here is the resulting animated gif (6516 bytes only):

I will develop this section in some undeterminate future and comment upon the various tools to manage those images. For instance the .xwd format is the fastest (on my machine) for `xvidcap` *to generate. Each glimpse is about 300 Kbytes for a 24x80 window (with a color depth of 16bits). A glimpse converted to gif may give a file of about 30 kbytes but combining some gifs allows to use a differential compression scheme that further shorten an image to a few bytes only.*

4 `xscript` installation

Same as `xvidcap`: run `./configure` then `make xscript` then `make install`. You may test the `xscript` concept with `make cdls.gif` or look at other tests in the first part of the *qnc.mk* makefile.

You may alternatively download an i686 *.rpm* file from the my site[6].

More to come! That's all, enjoy!

[6] `http://youpou.lip6.fr/queinnec/Programs/xscript-latest.i686.rpm`