checkergen user manual

published January 23, 2011

Contents

1 Introduction

Checkergen is a simple program written in Python which generates flashing checkerboard patterns for use in psychophysics experiments. A command line interface (CLI) is provided for the setting of parameters, and output either to screen or to file is provided.

In checkergen, you can create checkerboard patterns and vary the size, position, colors, amount of distortion along the x and y axes, as well as the frequency and phase of flashing.

You can then display these patterns on screen, and at the same time send trigger signals through a serial or parallel port to a data acquisition machine. Alternatively, you can export the patterns as a series of image files which can then be displayed by another stimulus-presenting program.

2 Requirements

- Python 2.7 or Python 2.6 + argparse module
- pyglet (version >= 1.1.4 recommended)
- pySerial if you want to send trigger signals via a serial port
- pyParallel if you want to send trigger signals via a parallel port
  - giveio driver required on Windows
  - see pyParallel homepage for other requirements
- pywin32 if you want to increase process priority on Windows
- OpenGL installation with the *GL_EXT_framebuffer_object* extension for export and scale-to-fullscreen functionality

3 Running *checkergen*

Open a command prompt or terminal and make sure you are in the same directory as the file `checkergen.py`. Also make sure that Python is on your system path. On a UNIX-like OS, enter:

```
./checkergen.py
```

On Windows, enter:

```
checkergen.py
```

Once you run checkergen, the checkergen prompt will appear:

```
Enter 'help' for a list of commands.
Enter 'quit' or Ctrl-D to exit.
(ckg)
```

Checkergen can be supplied with the path to a project file as an argument, such that that project will be loaded upon startup. For other options, run checkergen with `--help` as an option.

4 Projects

Stimuli you create in checkergen are stored as *projects*. Checkergen projects contain all your checkerboards, as well as various other settings like framerate in frames per second (fps) and background color. Projects are saved as XML files with `.ckg` extension. The name of the project is always the same as the name of the file it is saved as, excluding the extension.

4.1 Creating a new project

To create a new project, enter:

```
(ckg) new [projectname]
```

`projectname` is optional and defaults to 'untitled'. All project settings are initialized to reasonable defaults.

4.2 Saving a project

To save the project you've just created, enter:

```
(ckg) save [path/to/file.ckg]
```

If the path is not specified, your project will be saved to the current working directory as `projectname.ckg`. The path can either be absolute or relative, but must always refer to a file with a `.ckg` extension. The name of the saved file becomes the project name.

4.3 Opening an existing project

To open an existing project, enter:

```
(ckg) open path/to/file.ckg
```

As with `save`, the specified file must have the `.ckg` extension. The current working directory will be changed to the directory containing the specified file.

4.4 Closing the project

To close the project, enter:

```
(ckg) close
```

If unsaved changes exist in the project, you will be prompted to save the file first.

4.5 Project settings

Project settings can be edited using the `set` command. The usage is as follows:

```
set [--name NAME] [--fps FPS] [--res WIDTH,HEIGHT]
    [--bg COLOR] [--pre SECONDS] [--post SECONDS]
    [--cross_cols COLOR1,COLOR2] [--cross_times TIME1,TIME2]
```

A full description can be seen by entering `help set` into the checkergen prompt.

An important point to note is to always remember to `set` the fps of your project to the framerate of the monitor you are using, as there is no auto-detect functionality. Otherwise, the pattern animation will not be displayed at the correct speed. For example, if your monitor has a 120 Hz refresh rate, do:

```
(ckg) set --fps 120
```

4.6 Listing project information

To get an overview of the current project, enter the `ls` command. This will list all project settings, as well as all display groups, checkerboards, and their properties. The following is some sample output:

```
(ckg) ls
**************************PROJECT SETTINGS**************************
            name      fps      resolution            bg color
         2groups      120         800,600         127,127,127
                pre-display                post-display
                          0                           0
              cross colors                 cross times
             0;0;0,255;0;0                  Infinity,1

*******************************GROUP 0********************************
          pre-display                display          post-display
                    5                        5                     0
   shape id        dims       init_unit        end_unit        position
           0         5,5           40,40           50,50         380,320
           1         5,5           40,40           50,50         420,320

   shape id                      colors        anchor    freq    phase
           0           0;0;0,255;255;255   bottomright       2        0
           1           0;0;0,255;255;255    bottomleft       1       90

*******************************GROUP 1********************************
          pre-display                display          post-display
                    5                        5                     0
   shape id        dims       init_unit        end_unit        position
           0         5,5           40,40           50,50         380,320
           1         5,5           40,40           50,50         420,320

   shape id                      colors        anchor    freq    phase
           0           0;0;0,255;255;255   bottomright       1      180
           1           0;0;0,255;255;255    bottomleft       2      270
```

5 Display groups

In order to display different sets of checkerboards at different points in time, checkerboards are organized into *display groups*. All checkerboards in a display group are dis-

played together for a certain duration, after which the next display group takes over.

In addition to specifying the duration of a display group, you can also specify the duration a blank screen is shown before and after the display group is actually displayed.

5.1 Creating a display group

Display groups are created using `mkgrp`:

```
(ckg) mkgrp [pre] [disp] [post]
```

where

- `pre` is the time in seconds a blank screen is shown before the checkerboards in a display group are displayed,
- `disp` is the time in seconds the checkerboards are actually displayed on-screen,
- `post` is the time in seconds a blank screen is shown after the checkerboards in a display group are displayed.

`pre` and `post` default to `0`, while `disp` defaults to `Infinity`. Creating a display group automatically makes it the current active context for creation and editing of checkerboards.

5.2 Changing the active group

All commands that manipulate checkerboards are directed to the display group that is currently active for editing. To change the current active display group, do:

```
(ckg) chgrp [group_id]
```

where `group_id` is the ID number of the group you want to switch to. The first group you create will have ID number 0, the next will have ID number 1, and so on. If `group_id` is not specified, then checkergen prints a message telling you which is the current active display group.

5.3 Editing display groups

`edgrp` can be used to edit display groups that have already been created. It is used as follows:

```
edgrp [--pre SECONDS] [--disp SECONDS] [--post SECONDS]
      list_of_group_ids
```

The options you can specify are the same as in `mkgrp`. Edits will be performed on all the groups specified by the ID numbers in `list_of_group_ids`. At least one group id must be specified.

5.4 Removing groups

To remove display groups, simply do:

```
(ckg) rmgrp list_of_group_ids
```

If the active display group is removed, then group 0 becomes the new active display group (unless there are no more groups in the project).

6 Checkerboards

Checkerboards are managed using similar commands to those used for display groups. These commands only affect the current active display group, so `chgrp` must be used before checkerboards in another group can be managed.

As mentioned in the introduction, checkerboard patterns have a variety of attributes. A full list of attributes follows:

- dimensions (in number of unit cells)
- size of the initial unit cell (in pixels)
- size of the final unit cell (in pixels)
- position (of the anchor, in pixels from bottom-right corner of window)
- anchor (i.e. where the initial cell is positioned within the board e.g. top-left)
- colors (which alternate both in space and time)
- frequency of flashing / color alternation (in Hz)
- initial phase of color alternation (in degrees)

6.1 Creating a checkerboard

To create a checkerboard in the current display group, use `mk`. The usage is as follows:

```
mk dims init_unit end_unit position anchor cols freq [phase]
```

Except for the phase, which defaults to `0`, you have to specify all the attributes mentioned above. For more detailed information, enter `help mk`.

6.2 Editing checkerboards

Similar to `edgrp`, you can edit checkerboards in the current group by using `ed` and specifying a list of checkerboard ID numbers. The usage is as follows:

```
ed [--dims WIDTH,HEIGHT] [--init_unit WIDTH,HEIGHT]
   [--end_unit WIDTH,HEIGHT] [--position X,Y]
   [--anchor LOCATION] [--cols COLOR1,COLOR2]
   [--freq FREQ] [--phase PHASE]
   list_of_checkerboard_ids
```

For more detailed information, enter `help ed`.

6.3 Removing checkerboards

Much like `rmgrp`, to remove checkerboards from the current group, simply use `rm`:

```
(ckg) rm list_of_checkerboard_ids
```

7 Display

To display the project on-screen, use the `display` command. To stop displaying project animation midway, press ESC. The usage is as follows:

```
display [-f] [-p LEVEL] [-r N] [-pt] [-lt] [-ld] [-ss] [-sp]
        [list_of_group_ids]
```

With no options specified, the project will be displayed in a window, going through each display group in order. If you want display groups to be displayed in a specific order, just supply a list of group ids as arguments. If you want to repeat displaying groups in that order several times, then give the `-r/--repeat` option and specify how many repeats you want. For example, entering:

```
display -r 20 2 0 1 1
```

will result in group 2 being displayed first, then group 0, then group 1, then group 1 again, with the sequence of groups being shown 20 times in total.

The following subsections describe the other options in greater detail.

7.1 Fullscreen display

The `-f/--fullscreen` flag, when specified, causes the project to be displayed in fullscreen. If the project's resolution is not the same as the screen's resolution, then checkergen will try to stretch the animation to fit the screen. This requires the use of framebuffer objects, so your OpenGL installation must support the *GL_EXT_framebuffer_object* extension.

7.2 Setting the process priority

To reduce the number of monitor refreshes that are missed, checkergen can increase the priority of the Python process it is running in, but thus far only on Windows. This functionality requires the pywin32 module, and is invoked by giving the `-p/--priority` flag and specifying the priority level.

There are 4 priority levels, low, normal, high and realtime, which can be also specified by integers in the range 0 to 3 (i.e. 0 means low, 3 means realtime). When conducting an experiment, use `display -p realtime` for the best performance. However, using realtime priority might cause the computer to become less responsive to user input in some cases.

7.3 Logging time information

Specifying the `-lt/--logtime` flag will enable logging of each frame's timestamp, while specifying the `-ld/--logdur` flag will enable logging of each frame's duration. This information will be output to a log file in the same directory as the project file once display is done, with the filename as `projectname.log`. If either kind of logging is enabled, and if signalling is also enabled (`-sp` or `-ss` flags), then the log file will also record the trigger signals sent at each frame, if there were any.

7.4 Sending triggers

To send trigger signals via the parallel port to the data acquisition machine, specify the `-sp/--sigpar` flag. To send signals via the serial port, specify the `-ss/--sigser` flag. Serial port functionality has not, as of this date, been tested.

Unique triggers are sent upon several events:

- When the checkerboards in a display group appear on-screen (42 sent)

- When the checkerboards in a display group disappear from the screen (17 sent)
- Whenever a checkerboard undergoes a color reversal (64 + board id sent)

Should two or more checkerboards happen to undergo a color reversal at the same time, the trigger sent will be for the checkerboard with the largest ID number.

Triggers are sent immediately after the screen flip operation returns, ensuring as far as possible that the stimulus onset is synchronized with the sending of the trigger. For a discussion of issues regarding synchronization of the screen with the signal-sending port, see section 7.5.

7.5 Photodiode testing

To facilitate testing of stimulus onset time and refresh rates using a photodiode, the `-pt/--phototest` flag can be specified. When specified, a small white test rectangle will be displayed (for the duration of one frame) in the top-left corner of the window/screen whenever the checkerboards in a display group appear on-screen. Hence, this should coincide with a trigger sent by the parallel or serial port.

However tests have shown that on certain set-ups, the trigger is sent by the port one frame earlier than when the photodiode signal is detected, suggesting that the trigger is sent before the screen flip operation has completed entirely. If this happens to you, it may be due to vertical sync (vsync) not being properly enabled. To play safe, it is advisable to configure OpenGL or your graphics driver to force vsync on for all applications. After doing so, the port signal should be synchronized with the photodiode signal.

8 Export

The `export` command is similar to the `display` command, allowing you to specify the list of group ids to exported, as well as allowing you to specify the length of the project animation that should be exported (in seconds). Only PNG is supported as an export format. The usage is as follows:

```
usage: export [-n] [-r N] [duration] [dir] [list_of_group_ids]
```

For more detailed information, enter `help export` into the checkergen prompt.