FlyFly User Manual

## Before starting

There are a number of things that can be done in order to improve the general performance of the system running the stimuli. The most important thing is to make sure the operating system used is as clean as possible with a minimum of programs running. If possible all automatic updates, virus scans and similar should be turned off. It is also beneficial to disconnect the computer from the network and to unplug external devices. Depending on the OS, there might be a number of performance enhancing settings to do, such as turning of unnecessary graphical bling-bling.

## Required components

To run the program you need the 32bit version of MatLab 10 with Psychophysics toolbox 3 (PTB-3) and the folder containing all the FlyFly files. Older MatLab version should work with some changes in the program code, such as replacing the tilde character[[1]](#footnote-1) (~) with a temporary variable. Older versions of PTB have not been tested and might not function.

If you don´t have psychophysics toolbox you can get it at **psychtoolbox**.**org**/. Note that PTB-3 won´t install with the 64bit versions of MatLab.

## Launching the program

Open MatLab and change the *current folder* to the root folder of FlyFly - this folder is named FlyFly 1.7.1 or similar. Launch the program by running the file flyfly.m (you may either right click on the file in the Current folder view and select Run file or just type flyfly in the command window).

## The main window

Running the file flyfly.m will start the *main* application. You should see a window similar to Figure 1 (with possible small deviations due to version history).



Figure 1: main view

To the left you see a list of available experiments. Double click one or several experiments to add them to your project, i.e. the list *Chosen stimuli*. You can add several experiments of the same kind, using different parameter settings. For example, with the *target* experiment, the first stimulus could be used to map a receptive field and the second to do a size tuning experiment.

Once added, it is possible to remove or rename the experiments. It is also possible to copy an experiment, which might be useful if you want to make small changes to an existing experiment.

For experienced users, it is possible to import parts from an existing project. In the File menu, the "Import Stimuli" option makes it possible to select an existing saved project and specify which experiments in that project to import.

Clicking the 'Next' button will send you to the first experiment. If you haven´t added one, nothing will happen.

## The screen

To run any experiment you will need to initialize something called a Screen. If you run a dual display setup you will probably use one monitor to control the user interface and one to display the stimuli. To see the numbering of your available monitors either type Screen('Screens') in the command window or press Tools -> Settings from the main view. Under the Screen panel there should be a line saying "Available screen numbers: " followed by the numbers (starting at 0). For a single monitor this will be '0', with two monitors '0, 1' etc.

From the settings panel you can change which screen to use (if you have multiple) and also switch between full screen or a smaller partial screen. The partial screen is by default placed with its left top corner in the left top corner of the monitor with the task bar. To change this position you have to make a change in the matlab file *initFcn.m*.



Figure 2: Main settings

The screen is initialized by clicking on the "Init Screen" button. Once the screen is initialized, it covers everything behind it, which makes it impossible to click there. You can close the screen by clicking the "Kill Screen" button or by typing Screen('CloseAll'); in the matlab command window.

**IMPORTANT**: If you cover both the matlab command window and the FlyFly GUI, it is impossible to close the screen in a neat way. In OSX you can force quit matlab by pressing *cmd+q*. In windows it is usually possible to get the taskbar in front of the screen using alt+tab, which allows the screen to be closed (right click -> close window) .

## Settings

Pressing Tools -> Settings opens up the settings panel.

### Trigger

From the settings panel it is possible to change the size and position of the photodiode trigger. The coordinate system used for the position starts in the top left corner with positive x to the right and positive y downwards.

The trigger color is by default white (on) and black (off). To change it, find the variables triggerRGBon and triggerRGBoff in the file *initFcn* and change the values accordingly. Note that you have to restart the program to apply these changes, because the file runs at start-up.

### Screen

It is possible to change the gamma of the screen used.

The button called Init Test . Pressing this button will initialize a new window and then automatically close it after 2s. If you´re unsure which monitor corresponds to which number, use this function in order to test, so you don´t cover the user interface by accident.

## The tableGui

This Gui lets you enter parameters from a table view, which is suitable for many types of experiments.

 

1

6

5

4

3

2

**1:** Name of experiment (*Rect* *Targets*) and current layer (*Layer 1: Rect Target*). You can change the names by pressing the *Edit* buttons on the left. You can add more layers in the layer manager.

**2:** Info panel. Gives the resolution and update frequency of the screen, if it has been initialized. *Skipped frames* gives the number of skipped frames for the last experiment and *Skipped total* gives the total number of frames skipped. This value can be reset with the *Reset* button to the left.

**3:** The parameter table. Here it is possible to set all the parameters to be used in the current layer. If a cell is marked in the table you have to click once outside the table to be able to make a new action, such as clicking on a button.

**4:** Sequencer panel. Change *Number of trials* to run by typing in a new number and pressing *OK*. Marking *Trial Subset* will play the trial(s) between the chosen numbers.

**5:** Interpolate panel. The *Linear* and *Log* buttons interpolate between the end values (first and last) on a single row in the parameters table. The row is chosen by clicking on the row in the table. The shuffle button shuffles the order of values in the row randomly.

The *Use value* button takes the currently selected value and applies it to all the *n*th cells in the same row, where *n* is the number in the box to the right of the button (not shown here) . I.e. if the number in the box is 2, then every second value will be the same as the marked one.

Checking the 'All' box means the same action will be applied to all rows at the same time.

**6:** Before running an experiment the screen needs to be initialized by pressing *Init Screen*. At initializing, a number of tests is run by PTB-3 to measure the screen update frequency and synchronize with the vertical retrace. Sometimes, notably on computers with older graphic cards, this fails and an error message will be thrown. In that case just press the button again.

To close down the screen after an experiment press the button *Kill Screen*. The screen will also automatically be closed when the program itself is shut down.

You can draw a grid to the screen by pressing the *grid* button. This is useful to get a quick idea on which coordinates to use in your experiment. You can clear the screen again by pressing the *clear* button.

## The layer manager



All stimuli are implemented as layers. This means that all stimuli can be combined together if needed. From the layer manager it is possible to add new layers from the list of available stimuli. Stimuli added will draw "over" the previous one, meaning that if you want for example a small target moving over a sine grating, the target should be added last or moved to the bottom of the list to be visible.

You need at least one layer to draw a stimuli.

## The 'target' stimulus

//Parameters

'Height' - Target height in pixels

'Width' - Target width in pixels

'Xpos' - Target starting position in pixels

'Ypos' - target starting position in pixels (top is 0)

'Velocity' - Target speed in pixels/s

'Direction' - Target motion direction

'Target Brightness' - Target brightness (0-255, 0=black, 255=white)

'Background' - BG brightness (0-255, 0=black, 255=white)

'Time' - How many seconds the target will be rendered

'PauseTime' - Length of pause between trials. During pause the trigger is turned off.

'PreStimTime' - Length of pre stim

'PostStimTime' - Length of post stim

The *target* stimulus draws a black square to the screen and moves it at a uniform speed for a specified amount of time.

## The 'sine' stimulus

//Parameters
'Wavelength' - Wavelength in pixels

'Temporal Freq' - Temporal frequency (cycles/s)

'Direction' - Direction of motion

'PatchHeight' - Height of grating patch in pixels

'patchWidth' - Width of grating patch in pixels

'Patch Xpos' - Patch position, left

'Patch Ypos' - Patch position, top

'Contrast' - Patch contrast (0-1: 0 = gray image, 1 = normal)

'Time' - How many seconds the grating will be rendered

'PauseTime' - Length of pause between trials. During pause the trigger is turned off.

'PreStimTime' - Length of pre stim

'PostStimTime' - Length of post stim

The *sine* stimulus draws a patch filled with a sinusoidal grating.

## The 'rolling image' stimulus

## //Parameters'Speed' - Speed (pixels/s)

'Direction' - Angle relative to monitor top

'Xpos' - Image offset on x-axis

'Ypos' - Image offset on y-axis

'Height' - Image height (pixels)

'Width' - Image width (pixels)

'Starting position' - Starting position of image

'Contrast' - Contrast between 0 and 1

'Time' - How many seconds the image will be rendered

'PauseTime' - Length of pause between trials. During pause the trigger is turned off.

'PreStimTime' - Length of pre stim

'PostStimTime' - Length of post stim

Sets a background image and moves it across the screen. The image will have its top left corner in the corner of the screen. Height and width is size of visible patch of image. Making the patch larger than the image will not stretch it. *Xpos* and *Ypos* move the patch relative to its starting position.

## The 'Starfield' stimulus

//Parameters
'Brightness' - Brightness of dots
'Dot size' - Size of dots when right in front of the viewer
'number of dots' - Number of dots used in calculations
'Sideslip' - Sideslip (cm/s)
'Lift' - Lift (cm/s)
'Thrust' - Thrust (cm/s)
'Pitch' - Pitch (deg/s)
'Yaw' - Yaw (deg/s)
'Roll' - Roll (deg/s)
'Time' - How many seconds the image will be rendered

'PauseTime' - Length of pause between trials. During pause the trigger is turned off.

'PreStimTime' - Length of pre stim

'PostStimTime' - Length of post stim

Draws a cloud of dots and translates and or rotates it. Number of dots is the number of dots in the cloud, which is not the same as number of dots on screen. Dots on screen depend on the total number of dots, distance to monitor and monitor width (adjusted in settings).

## The '.Mat sequence'

//Parameters
'Fps' - Frames per s
'Xpos' - Placement on screen (center)
'Ypos' - Placement on screen (center)
'Time' - How many seconds the image will be rendered

'PauseTime' - Length of pause between trials. During pause the trigger is turned off.

'PreStimTime' - Length of pre stim

'PostStimTime' - Length of post stim

Loads a 3D matrix and displays each "layer" (z-dimension) of it as a frame. Note that this takes up quite a lot of memory, meaning large matrices and/or long sequences might be impossible to play.

## Common problems

*How can I abort a sequence of trials currently running?*

-Press ctrl-c in the matlab command window. Note that quiting like this means that matlab never reaches the end of the function running, meaning **no parameters will be saved**. Also, since the experiment is not finished properly it is not possible to start a new one (see below).

*The "Run button" is grey, even though I initialized the screen.*

-Try pressing the "Clear" button (will update screen status and refresh the GUI).

*The screen won´t initialize!*

-Try switching between "partial" and "fullscreen" mode. Make sure a valid screen number is entered.

1. used to ignore the output from a function, first available in MatLab 10 [↑](#footnote-ref-1)