

The S cone psychophysics system makes use of Psychtoolbox in MATLAB. It has a few main components that should hopefully make running experiments and defining stimuli relatively straightforward.

## PsychtoolboxSession:

This manages creation of a window to display Psychtoolbox stimuli and holds parameters relating to the hardware. It also provides methods to clear the display window, change the default window background intensity, as well as close the Psychtoolbox window.

When experiments are run, they will be run within one of these sessions. To do this, a session will be created and then given to a function that runs an experiment.

An example of the usage of a Psychtoolbox session is below (without actually running an experiment yet):

```
% make a PsychtoolboxSession, also include some optional parameters to control
% different aspects of the window (see implementation of the PsychtoolboxSession
% for a description of all of the optional parameters)
session = SConePsychophysics.PsychtoolboxSession('DebugMode', true, ...
    'DebugModeFullscreen', false, ...
    'DebugModeScreenRectangle', [0 0 640 480], ...
    'BackgroundIntensity', [0.5 0.5 0.5]);
```

```
% change the background
session.SetBackground([0.4 0.4 1]);

% clear the screen (this updates to new background)
session.Clear();
```

```
% close the session
session.Close();
```

## Stimulus Parameters:

A stimulus parameters object will hold all of the stimuli for a given experiment. Each different stimulus type will have a unique set of stimulus parameters. These, along with a session, will be given to the functions that run an experiment. A stimulus parameters object, when created is essentially an empty list of parameters. Before using it, each required parameter must be specified. The file in which the parameters object is defined will show which parameters need defining and provide an explanation of what these parameters control. An example of how to create and define stimulus parameters for a Benham's top experiment is below:

```
% create a stimulus parameters object (in this case, the specific one for
% Benham's top)
parameters = SConePsychophysics.StimulusGenerators.BenhamTop.Parameters();

% the parameters object will start without any of the parameters being
% defined, now go through and define them all
parameters.backgroundIntensities = [0.8 0.8 0.8];
parameters.darkIntensities = [0 0 0];
parameters.frequency = 1;
```

```

parameters.offsetStepSize = 0.01;
parameters.maxOffset = 0.3;
parameters.minOffset = -0.3;
parameters.radius = 100;
parameters.numArcGroups = 3;
parameters.numArcsInGroups = [4 3 3];
parameters.arcGroupThetas = ...
    {[0 pi / 3], [(pi / 3) (2 * pi / 3)], [(2 * pi / 3) pi]};
parameters.arcThickness = 0.04;
parameters.arcMargin = 0.04;
parameters.startArcRadius = 0.18;
parameters.colorSpace = SConePsychophysics.Constants.COLOR_SPACE_LMS;

```

## Generator Function:

Now that we have defined the parameters for a Benham's top experiment, it is time to define what function will use those parameters to generate the stimulus. This is done by creating a function handle to the function we want to be used. A function handle allows one to create a variable that simply points to some function. This way, that variable can be provided as an input to another function, and this function will then know where to look to find the function to construct the stimuli. Creating a function handle is done by using the `@` symbol in front of the function to which you want the handle to refer. For the Benham's top experiment, it is done as follows.

```

generatorFxn = @SConePsychophysics.StimulusGenerators.BenhamsTop.Generate;

```

## Running an Experiment:

To run an experiment, we used the Main method within SConePsychophysics. This can be called by using SConePsychophysics.Main(..., ..., ...); As inputs, it takes the things we have constructed above: the stimulus parameters object, a generator function, a Psychtoolbox session, and also the path of a folder in which to save the results. A simple experiment can be initiated as follows:

```

% define the path of a save folder (by default, this is setup to just save the
% results to the desktop in a folder 's_cone_psychophysics_demo' and a subfolder
% based on the current date and time - this is just to make sure the demo doesn't
% overwrite any files potentially saved on the desktop already with the same name)
saveFolder = fullfile(getenv('HOME'), ...
    'Desktop', 's_cone_psychophysics_demo', ...
    ['demo_results_' char(datetime('now', 'Format', 'yyyy_MM_dd_hh_mm_ss'))]);
SConePsychophysics.Utils.MakeDirs(saveFolder);

% create a new session for this experiment
session = SConePsychophysics.PsychtoolboxSession('DebugMode', true, ...
    'DebugModeFullscreen', false, ...
    'DebugModeScreenRectangle', [0 0 640 480], ...
    'BackgroundIntensity', parameters.backgroundIntensityMonitorSpace);

% run the actual experiment
SConePsychophysics.Main(parameters, generatorFxn, session, saveFolder);

% add this so that the session automatically closes once the experiment is
% finished (were this not to be here, we could immediately run another
% experiment (for example, with a new parameter set) while keeping the same
% session active, and only close it at the very end of the set of experiments)
session.Close();

```