COGS 119/219
MATLAB for Experimental Research

Fall 2014 – Week 7

Cont. Psychtoolbox

Displaying Images

Screen (cont'd)

See examples on the site for Screen

```
ScreenIntro.m
ScreenBlackWhite.m
FunScreen.m
```

Displaying images in PTB: PutImage

```
% Read in image just like usual
>> myimg = imread('sungod','jpg');

% Use PutImage function

Try  >> Screen PutImage?

>> Screen('PutImage', w, myimg);
>> Screen('Flip',w);
```

Displaying images in PTB: Texture

Preferred method

The texture method is much faster and flexible compared to PutImage

```
>> Screen MakeTexture?
```

```
TextureIndex = Screen('MakeTexture',
WindowIndex, imageMatrix, optional
arguments…)
```

```
>> Screen DrawTexture?
```

Download **playwithtextures.m** from the class website.

Textures

```
% Read in image just like usual
myimg = imread(’sungod','jpg');

% make texture, nothing happens on screen.
% this just prepares the image as texture on
% offscreen window

mytex = Screen('MakeTexture', w, myimg);

Screen('DrawTexture', w, mytex);
Screen('Flip',w);
```

Note where the image displays

Textures, Screen, Window

What's the difference between window, screen and texture?

http://psychtoolbox.org/wikka.php?wakka=FaqTextureWindow

For more PTB notes, info, etc see file on the site called Week6PTB_AdditionalNotes.pdf

Images at different locations on screen

- Let's display an image on different locations on Screen.
- Download **playwithtextures2.m** and sungod.jpg from the class website and run the m file.
- Image is displayed first on the left and then on the right of the screen. And then both.
- Do you understand how we can specify where can display an image?
- Why do we see text and the images together only in the last part?

Images at different locations on screen

- `PutImage` and `DrawTexture` both require a location for the image as an input argument.
- We need to calculate the coordinates of where we want to put your image (use your geometry knowledge).
- Start by calculating the coordinates of the center of the screen.

```
>> help RectCenter
```

- Then, calculate the coordinates of the image location relative to the center based on the image size
- Do you understand what is happening in lines 36-40 of **playwithtextures2.m** ?

Moving stimuli

- Movies can be made by changing the position of the stimulus each frame
- Exercise: Draw a white circle that moves randomly on screen without crossing the top, bottom, left and right borders of the screen.
- How can you do that? Any ideas?

randompath.m

```
[w,rect]=screen('OpenWindow',0,[0 0 0], [0 0 800 600]);
r = 50; % radius of circle (pixels)
v = 20; % velocity (pixels per frame)
x = rect(3)/2;
y = rect(4)/2;
keypress = 0; %initialize

% KbCheck within the while loop below checks keyboard status
% its output reports keyboard state (and more see help KbCheck

while(~keypress) % draw circle in new position unless key is pressed

    Screen('FillOval',w,[255 255 255],[x-r,y-r,x+r,y+r]);
    Screen('Flip',w);
    % compute new position
    x = x+v*(2*rand-1); y = y+v*(2*rand-1);
    % check borders to make sure circle won't go beyond screen
    x = max(x,r); % left border
    x = min(x,rect(3)-r); % right border
    y = max(y,r); % top border
    y = min(y,rect(4)-r); % bottom border
    % check whether any keys are pressed
    keypress = KbCheck;
end

Screen('Close',w);
```

From here to end PTB tips

See also Week7PTB_AdditionalNotes.pdf and Week7_PTB_SchneiderNotes.pdf

Some helpful tips about PTB

See "`help PsychDemos`" for many demos which demonstrate Screen's capabilities.

Some helpful tips about PTB

http://psychtoolbox.org/PsychtoolboxFaq

Psychtoolbox-3 Frequently Asked Questions (FAQ)

- How do I get PTB-3?
- How do I close a screen and return to the command line (Mac or Win)?
- Is PTB-3 backwards compatible with PTB-2 (Mac or Win)?
- Does the PTB-3 run on the new Intel-based Macintosh computers?
- How can I skip the checks that Screen performs when it starts up?
- How do I make the initial screen black instead of white?
- Why does it take 40 s to open my first window?
- When opening an onscreen window, Psychtoolbox shows a blue or white screen and then nothing happens?!?
- What's the difference between a texture, a window, and a screen?
- How do I duplicate an offscreen window?
- How to perform keyboard check for multiple keyboards (Mac OS/X Only)?
- Can I set TextSize (and other parameters) for all windows/screens?
- Can offscreen windows (created with 'OpenOffScreenWindow') have multiple buffers?
- Windows: Why does my virus checker complain about the PTB-3 distribution?
- What do the timestamps returned by Screen('Flip') mean?
- Is it possible to get 10-bit DAC resolution with PTB-3 under WinXP or OS/X?
- How to display images with transparent backgrounds?
- How can I try to improve timing and performance of PTB-3 code?

Some helpful tips about PTB

SYNC Troubles:

If you see some timing errors, complaints about sync or VBL, see this: http://docs.psychtoolbox.org/SyncTrouble.

Some helpful tips about PTB

If your computer only has one screen (the typical scenario) and your program produces a Matlab error while your full-screen window is open, you'll hear the beep, but you won't be able to see the Matlab Command Window.

Follow the instructions below for bringing forward the command window, then type "`clear Screen`" to flush just the Screen MEX file, or "`clear mex`" to flush all the MEX files. When flushed, as part of its exit sequence, Screen closes all its windows, restores the screen's normal color table, and shows the cursor.

Or you can get just those effects, without flushing, by calling `Screen('CloseAll')` or `sca` - which is an abbreviation for `Screen('CloseAll')`.

Some helpful tips about PTB

Recovering errors after a crash

You can use Matlab's EVAL command to do this for you automatically. E.g. if your program is called "foo.m", run your program by calling EVAL:

```
eval('foo','clear screen;error("error in foo")')
```

If an error occurs in FOO, Matlab, instead of halting, will execute the second argument to EVAL, which restores your screen and reports the error.

Some helpful tips about PTB

Command-zero brings the Matlab Command window forward. (Type a zero "0" while holding the apple-cloverleaf "command" key down.)

Ctrl-C halts any program. (Type a "c" while holding down the "Ctrl" key). Sometimes, Ctrl-C fails to halt progams executing in a Matlab process run with the "-nojvm" option. To halt a runaway Psychtoolbox script in Psychtoolbox you might resort to the Windows Task Manager to kill the Matlab process. (Use Ctrl-Alt-Delete to open a window from which you can start the Task Manager.)

Some helpful tips about PTB

Windows:: Ctrl-Alt-Delete allows you to launch the Windows task manager, which reduces the Psychtoolbox onscreen windows when it opens. (Simultaneosly press the "Ctrl", "Alt", and "Delete" keys.) There are also simpler ways of reducing the Psychtoolbox window which are specific to particular versions of Windows.

Windows 2000: Alt-Tab will bring another application to the foreground, minimizing the Matlab Psychtoolbox window.

OS-X: Apple-Command-Escape executes "Force Quit" on Matlab, closing Matlab and all of its windows.

Some helpful tips about PTB

OPENGL:

OpenGL is a library for computer graphics. It allows us to create applications which render high-quality color images with 3D objects. You are not required to go into details of OpenGL at the moment. http://www.opengl.org/about/overview

Some helpful tips about PTB

WARNINGS:

You can disable PTB's warnings about graphics and timing for the time being by including the following in your code:

```
Screen('Preference', 'Verbosity', 0);
Screen('Preference', 'SkipSyncTests',1);
Screen('Preference', 'VisualDebugLevel',0);
```

Type `Screen('Preference?')` in the command line to get more info.