Homework 9 Due Today

Homework 10 Due November 14 (TWO weeks)

Readings for Today
None

Readings for Next Week
Attaway Chapter 9
Simple Animations

Simple animations are easy in Psychtoolbox. You just need to draw to the monitor and control the timing.

```
for x=1:xwidth/xstep

    sz = 50;
    px1 = x*xstep; py1 = 500;
    px2 = x*xstep + sz; py2 = 500 + sz;
    nangs = 20;
    ang = abs(mod(x,nangs)-mod(x,nangs)/2)*4;
    Screen('FillArc',wPtr,[255 255 0],
        [px1 py1 px2 py2],90+ang,360-ang*2);

    % display backbuffer after ITI
    [VBLT SOTime] = Screen(wPtr, 'Flip', 0);
end
```
KbCheck

t1 = GetSecs;
while true
  
  [touch, tpress, keyCode] = KbCheck;

  if touch
    fprintf('Key being pressed\n');
    break;
  end

  WaitSecs(0.0005);
end

KbCheck only checks keyboard when it's called
KbCheck

t1 = GetSecs;
while true
	n[touch, tpress, keyCode] = KbCheck;

if touch
    fprintf('Key being pressed\n');
    break;
end

WaitSecs(0.0005);
end

so if a key is pressed and released during this interval, it will be missed by KbCheck
KbCheck

t1 = GetSecs;
while true
  [touch, tpress, keyCode] = KbCheck;
  if touch
    fprintf('Key being pressed\n');
    break;
  end

  WaitSecs(0.0005);
end

not a problem for keyboard keypresses and if the loop only monitors for keypresses
KbCheck

t1 = GetSecs;
while true
    [touch, tpress, keyCode] = KbCheck;

    % COMPLEX IMAGE PROCESSING HERE

    WaitSecs(0.0005);
end

it could be a problem if there is a lot of time-consuming processing in this loop
button boxes

some response boxes (e.g., for fMRI or MEG) translate a physical button press into a signal that's read as a keyboard keypress for only a few milliseconds
KbCheck versus KbQueue

**KbCheck**
checks the current state of the keyboard

**KbQueue**
keys are added to a queue in the background (probably via interrupts)

KbQueue routines create and act on the keyboard queue, not on the keyboard state directly

the queue can only hold 30 events

but only the first and last keypress press and release times are recorded
KbQueue commands

KbQueueCreate();  
KbQueueStart();  
while true  
  [pressed, firstPress, firstRelease, ...  
    lastPress, lastRelease] = KbQueueCheck();  
  if pressed  
    fprintf('You pressed key %i which is %s\n', ...  
      find(firstPress), KbName(firstPress));  
  end  
end  
KbQueueStop();  
KbQueueRelease();

it takes a bit of time to create the queue
KbQueue commands

KbQueueCreate();

KbQueueStart();

while true
  [pressed, firstPress, firstRelease, ...]
  lastPress, lastRelease] = KbQueueCheck();
  if pressed
    fprintf('You pressed key %i which is %s\n', ...)
    find(firstPress), KbName(firstPress));
  end
end
KbQueueStop();

KbQueueRelease();
KbQueue commands

KbQueueCreate();
:
:
KbQueueStart();
:
:
while true

  [pressed, firstPress, firstRelease, ... 
    lastPress, lastRelease] = KbQueueCheck();
  if pressed
    fprintf('You pressed key %i which is %s\n', ...
          find(firstPress), KbName(firstPress));
  end
end

KbQueueStop(); stop delivering keys to the queue
:
KbQueueRelease();
KbQueue commands

KbQueueCreate();

KbQueueStart();

while true

  [pressed, firstPress, firstRelease, ...]
  lastPress, lastRelease] = KbQueueCheck();

  if pressed
    fprintf('You pressed key %i which is %s\n', ...
              find(firstPress), KbName(firstPress));
  end

end

KbQueueStop();

KbQueueRelease(); delete the queue
KbQueue commands

KbQueueCreate();
:
KbQueueStart();
:
while true
  [pressed, firstPress, firstRelease, ...
    lastPress, lastRelease] = KbQueueCheck();
  if pressed
    fprintf('You pressed key %i which is %s\n', ...
      find(firstPress), KbName(firstPress));
  end
end
KbQueueStop();
:
KbQueueRelease();

like KbCheck but returns first and last key pressed
KbQueue commands

KbQueueCreate();
::
KbQueueStart();
::
while true
    [pressed, firstPress, firstRelease, ...  
        lastPress, lastRelease] = KbQueueCheck();
    if pressed
        fprintf('You pressed key %i which is %s\n', ...  
            find(firstPress), KbName(firstPress));
    end
end
KbQueueStop();
::
KbQueueRelease();

now, the firstPress (and lastPress) arrays have a time at the index location of when a key was pressed
Simple Sounds for feedback
Simple Sounds for feedback

For simple auditory feedback, the `Snd` command is easy to use. If sound is used as stimuli (auditory / audiovisual) the `PsychPortAudio` commands should be used.

```matlab
% create error beep sound
% ret = MakeBeep(freq,dur);
errorbeep = MakeBeep(350,0.1);

% play that sound
Snd('Play', errorbeep);
```

>> SimpleVoiceTriggerDemo  
check SimpleVoiceTriggerDemo.m
Sounds and recording with PsychPortAudio

>> SimpleVoiceTriggerDemo

check SimpleVoiceTriggerDemo.m

You can play audio files synced with images or movies, play audio files at a particular point in time, record subjects, create voice triggers.

A lot of the programming principles for displaying precisely timed visual stimuli apply to precisely timed audio stimuli.
Using the mouse
Using the mouse

We've already seen:
HideCursor
ShowCursor
Using the mouse

We've already seen:
HideCursor
ShowCursor

More commands:
% set mouse to x, y
SetMouse(cX,cY,wPtr)

% get mouse x, y position and button states
[x,y,buttons] = GetMouse(wPtr);
Using the mouse

>> MouseTraceDemo2

You can review the code for this and other demos.
Integrating Stimulus Timing and Measuring Responses and Response Times
Examples

Imagine the following experiment (e.g., for fMRI):
- each trial must begin every 2500ms
- on each trial a stimulus is shown
- subjects make a response
- choice and response time is recorded (if on time)
- between each trial there must be at least 500ms blank ITI

let's think about this in pseudocode first
Examples

Imagine the following experiment (e.g., for fMRI):
- each trial must begin every 2500ms
- on each trial a stimulus is shown
- subjects make a response
- choice and response time is recorded (if on time)
- between each trial there must be at least 500ms blank ITI
- what if we want to allow people to respond during ITI?
  but you might worry about keypresses carrying over

let's think about this in pseudocode first
Examples

Imagine the following experiment:
- at the start of each trial, a fixation cross is shown for an exponentially distributed foreperiod (minimum of 250ms, maximum of 2250ms)
- during the fixation cross period, the subject must keep their finger on the space bar the whole time
- after the fixation cross appears, an image is shown
- subject makes a 1-9 preference rating
- choice and response time is recorded

Let's think about this in pseudocode first
Examples

Imagine the following experiment:

- each stimulus is shown for 2, 4, or 8 screen refreshes and then cleared
- subject makes a two-alternative forced choice decision any time after stimulus onset
- choice and response time is recorded from stimulus onset
- after response, the stimulus is cleared
- 500ms ITI

let's think about this in pseudocode first