第六章 心理学工具箱

Psychotoolbox/PTB

6.1 什么是心理学工具箱

- Psychotoolbox（PTB）集合了众多适合心理学实验的MATLAB函数。
  - PTB是由David Brainard, Denis Pelli, Mario Kleiner 和 Allen Ingling 等心理学家完成的。
  - PTB是目前使用范围最广的开源心理学实验工具箱（Lin et al., 2022）。

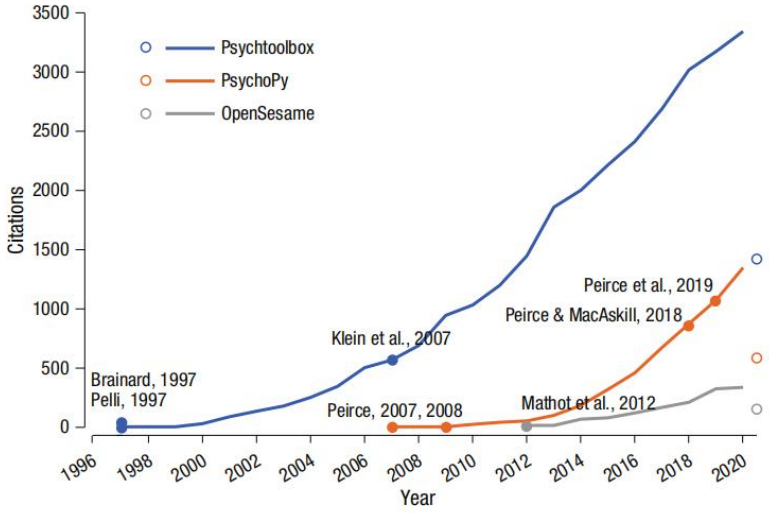
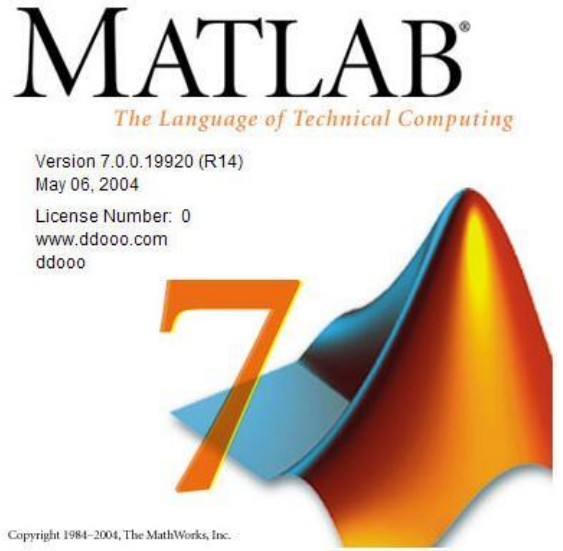


Fig. 1. Year-to-year citations of three major open-source software packages for stimuli presentation and response collection: Psychtoolbox (based on the total citations to Brainard, 1997; Klein et al., 2007; and Pelli, 1997), PsychoPy (based on the total citations to Peirce, 2007, 2009; Peirce et al., 2019; and Peirce & MacAskill, 2018), and OpenSesame (based on the total citations to Mathot et al., 2012). The year 2021 is partial, up to May 15 (data are represented by the open circles). Each dot represents a source article, aligned to the year of publication and the corresponding software package. Citations are based on Google Scholar recorded on May 15, 2021.

Lin, Z., Yang, Z., Feng, C., & Zhang, Y. (2022). PsyBuilder: an open-source, cross-platform graphical experiment builder for Psychtoolbox with built-in performance optimization. Advances in Methods and Practices in Psychological Science, 5(1), 25152459211070573.

【MATLAB+PTB优势】

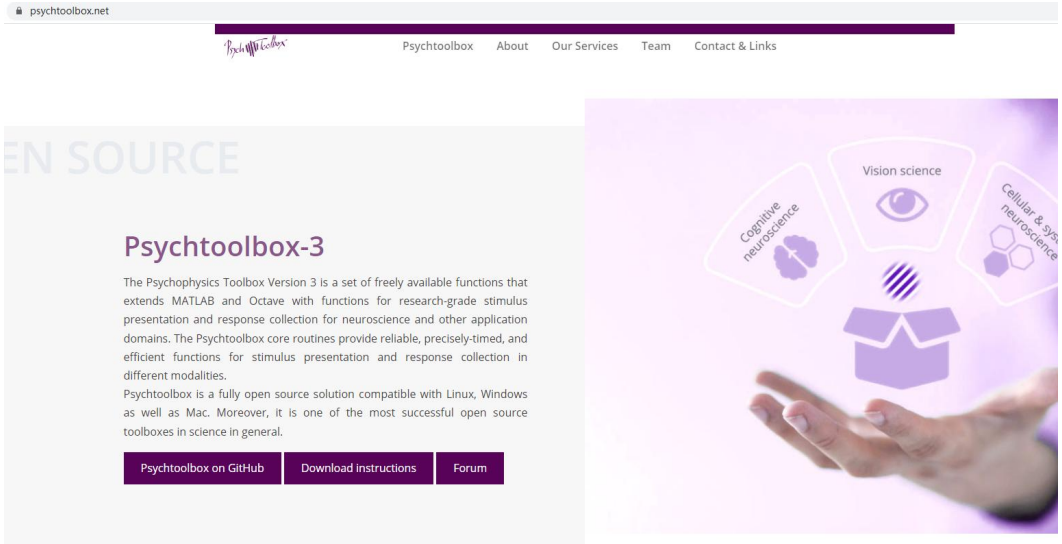


- 入门简单
- 中间过程是“黑箱”
- 拓展性弱

- 入门稍难
- 方便交互
- 拓展性强（“外挂”）
- 迁移性强（编程思维）

6.2 如何获得心理学工具箱

- 官网www.psychtoolbox.net/
  - PTB 3.0版本
- 中文教材附带资料

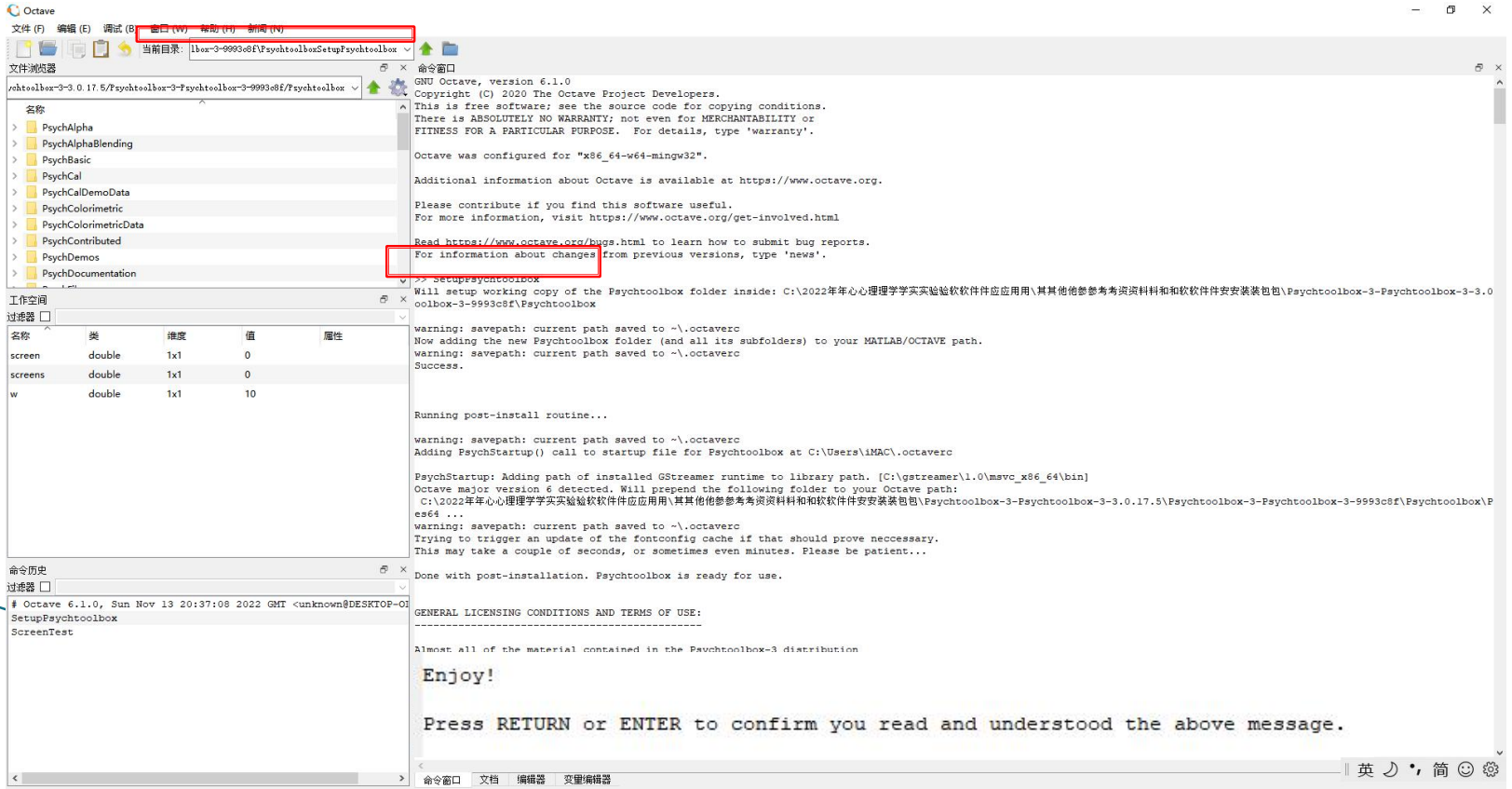


6.3 启动Psychotoolbox

- 在爱课本章学习资料列表中下载
- 安装gstreamer和Silk
- 将Psychotoolbox解压后的文件夹拷贝到LearningMatlab文件夹中
- 找到\Psychtoolbox-3-Psychtoolbox-3-9993c8f\Psychtoolbox子文件夹

6.3 启动Psychotoolbox

- 把当前路径拷贝到MATLAB或者Octave的‘当前目录’栏
- 在命令窗口>>输入SetupPsychotoolbox
- 出现Enjoy后，再按回车ENTER键即可



- 在命令窗口输入ScreenTest，开始检测是否成功安装。
- 如果没有下图所示，说明有问题。需要根据错误提示进行对应的操作。
- 如果使用Octave，需要删除opengl32.dll这个文件
  - 例如:C:\Program Files\GNU Octave\Octave-6.1.0\mingw64\bin\opengl32.dll

```
>> ScreenTest

***** ScreenTest: Testing Screen 0 *****


PTB-INFO: This is Psychtoolbox-3 for Microsoft Windows, under GNU/Octave 64-Bit (Version 3.0.17 - Build date: Dec  3 2020).
PTB-INFO: OS support status: Windows 10 (Version 10.0) supported and tested to some limited degree.
PTB-INFO: Type 'PsychtoolboxVersion' for more detailed version information.
PTB-INFO: Most parts of the Psychtoolbox distribution are licensed to you under terms of the MIT License, with
PTB-INFO: some restrictions. See file 'License.txt' in the Psychtoolbox root folder for the exact licensing conditions.

PTB-INFO: For information about paid priority support, community membership and commercial services, please type
PTB-INFO: 'PsychPaidSupportAndServices'.



PTB-INFO: Seems like Psychtoolbox is running inside a Virtual Machine? This doesn't provide sufficient
PTB-INFO: visual stimulus timing precision for research grade visual stimulation. I will disable most
PTB-INFO: tests and calibrations so you can at least get your scripts running for demo purposes. Other
PTB-INFO: presentation modalities and various Psychtoolbox functions will only work with limited functionality
PTB-INFO: and precision. Only use this for demos and simple tests, not for real experiment sessions!

PTB-INFO: Another reason for lack of hardware OpenGL could be that GNU/Octave's own opengl32.dll library
PTB-INFO: has not been deleted or renamed by you, so it is enforcing software rendering.
PTB-INFO: You have to delete or rename that file and restart Octave for this to work.
PTB-INFO: E.g., on Octave-6.1 at its standard installation location, the file to delete or rename would be likely this:
PTB-INFO: C:\Program Files\GNU Octave\Octave-6.1.0\mingw64\bin\opengl32.dll



PTB-INFO: OpenGL-Renderer is VMware, Inc. :: llvmpipe (LLVM 7.1.0, 256 bits) :: 3.1 Mesa 20.1.10
PTB-INFO: VBL startline = 1080 , VBL Endline = -1
PTB-INFO: Beamposition queries unsupported or defective on this system. Using basic timestamping as fallback.
PTB-INFO: Timestamps returned by Screen('Flip') will be therefore less robust and accurate.
PTB-INFO: Measured monitor refresh interval from VBLsync = 0.000000 ms [inf Hz]. (0 valid samples taken, stddev=0.000000 ms.)
PTB-INFO: Reported monitor refresh interval from operating system = 16.949153 ms [59.000000 Hz].
PTB-INFO: Small deviations between reported values are normal and no reason to worry.
PTB-INFO: ================================================================================================================================
PTB-INFO: WINDOWS DWM DESKTOP COMPOSITOR IS ACTIVE. On this Windows-10 or later system, Psychtoolbox can no longer reliably detect if
PTB-INFO: this will cause trouble for timing and integrity of visual stimuli or not. You might be just fine, or you could be in trouble.
PTB-INFO: Use external measurement equipment and independent procedures to verify reliability of timing if you care about proper timing.
PTB-INFO: ================================================================================================================================
PTB-INFO: All display tests and calibrations disabled. Assuming a refresh interval of 59.000000 Hz. Timing will be inaccurate!


WARNING: This session of your experiment was run by you with the setting Screen('Preference', 'SkipSyncTests', 2).
WARNING: This means that some internal self-tests and calibrations were skipped. Your stimulus presentation timing
WARNING: may have been wrong. This is fine for development and debugging of your experiment, but for running the real
WARNING: study, please make sure to set Screen('Preference', 'SkipSyncTests', 0) for maximum accuracy and reliability.

***** ScreenTest: Done With Screen 0 *****
```

6.4 使用Psychotoolbox的函数

- 新建一个m文件，命名为FlipScreen
  - 先写清楚标题header信息。

```
% FlipScreen.m
%
% This program opens a window using Psychtoolbox 3.0,
% wait for 5 secs, and then makes the window white for 5 secs, and then closes
% the window.
%
% written for Psychtoolbox 3 on the PC by YW and revise in 2022
```

- Screen函数是Psychotoolbox最重要的函数，包括一系列的子函数。
- 在命令窗口中输入Screen或者doc Screen，查看关于Screen函数的介绍。
  - doc Screen
  - Screen

```
clear all

screens=0;
[wPtr,rect]=Screen('OpenWindow',screens, 0, []);
HideCursor;
tic
while toc<5
end
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end
Screen('Close', wPtr);
ShowCursor;
```

```
>> Screen
Usage:

% Activate compatibility mode: Try to behave like the old MacOS-9 Psychtoolbox:
oldEnableFlag=Screen('Preference', 'EmulateOldPTB', [enableFlag]);

% Open or close a window or texture:
[windowPtr,rect]=Screen('OpenWindow',windowPtrOrScreenNumber [,color] [,rect] [,
entRect][,fbOverrideRect][,vrrParams=[]]);
[windowPtr,rect]=Screen('OpenOffscreenWindow',windowPtrOrScreenNumber [,color] [
textureIndex=Screen('MakeTexture', WindowIndex, imageMatrix [, optimizeForDrawAn
oldParams = Screen('PanelFitter', windowPtr [, newParams]);
Screen('Close', [windowOrTextureIndex or list of textureIndices/offscreenWindowI
Screen('CloseAll');

%  Draw lines and solids like QuickDraw and DirectX (OS 9 and Windows):
currentbuffer = Screen('SelectStereoDrawBuffer', windowPtr [, bufferid] [, param
Screen('DrawLine', windowPtr [,color], fromH, fromV, toH, toV [,penWidth]);
Screen('DrawArc',windowPtr,[color],[rect],startAngle,arcAngle)
Screen('FrameArc',windowPtr,[color],[rect],startAngle,arcAngle[,penWidth] [,penH
Screen('FillArc',windowPtr,[color],[rect],startAngle,arcAngle)
Screen('FillRect', windowPtr [,color] [,rect] );
Screen('FrameRect', windowPtr [,color] [,rect] [,penWidth]);
Screen('FillOval', windowPtr [,color] [,rect] [,perfectUpToMaxDiameter]);
Screen('FrameOval', windowPtr [,color] [,rect] [,penWidth] [,penHeight] [,penMod
Screen('FramePoly', windowPtr [,color], pointList [,penWidth]);
Screen('FillPoly', windowPtr [,color], pointList [, isConvex]);

% New OpenGL functions for OS X:
```

```
clear all

screens=0;
[wPtr,rect]=Screen('OpenWindow',screens, 0, []);
HideCursor;
tic
while toc<5
end
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end
Screen('Close', wPtr);
ShowCursor;
```

试一试：>>Screen 'OpenWindow'?

- 函数输入参数：
  - 'OpenWindow'：打开主页面/主窗口
  - screens＝0：在主显示器上打开
  - 0：窗口为黑色
  - ［］：使用默认值，窗口占据整个屏幕
- 函数输出参数：
  - wPtr（window pointer）：窗口的句柄（handle）
  - rect：窗口的大小（例如，左上角坐标[0 0]和右下角坐标[1600 900]）

```
rect =

     0     0  1600   900
```

- [windowPtr,rect]=Screen('OpenWindow',windowPtrOrScreenNumber [,color] [,rect][,pixelSize][,numberOfBuffers][,stereomode][,multisample][,imaging mode]);

  - 颜色color
    - 默认为白色
    - 自定义颜色：红绿蓝三色数值，[255 255 255]为白色，[255 0 0]红色，以此类推
  - 大小rect
    - 默认为屏幕大小
    - 自定义大小：[0 0 250 100]，屏幕左上角坐标为（0，0），右下角坐标为（1024，768）在此坐标系下定义
  - 剩余输入参数一般不需额外设置，使用默认值即可。

```
clear all

screens=0;
[wPtr,rect]=Screen('OpenWindow',screens, 0, []);
HideCursor;
tic
while toc<5
end
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end
Screen('Close', wPtr);
ShowCursor;
```

While end是另一种循环语句

- HideCursor隐藏鼠标
- ShowCursor显示鼠标
- Tic开始计时
- Toc持续读取时间
  - 例如，持续读取时间直到5秒，然后结束

```
clear all

screens=0;
[wPtr,rect]=Screen('OpenWindow',screens, 0, []);
HideCursor;
tic
while toc<5
end
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end
Screen('Close', wPtr);
ShowCursor;
```

Help BlackIndex

- 获得当前电脑黑色和白色对应的数值。
- 根据该数值的范围可设置其他颜色。
  - [127 127 127]为灰色

```
black =

     0


white =

   255
```

```
clear all

screens=0;
[wPtr,rect]=Screen('OpenWindow',screens, 0, []);
HideCursor;
tic
while toc<5
end
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end
Screen('Close', wPtr);
ShowCursor;
```

Screen 'FillRect'?

- FillRect在窗口画实心矩形
  - Screen('FillRect',wPtr,black);将矩形填充为黑色
  - Screen('FillRect',wPtr,white);将矩形填充为白色
  - Screen('FillRect', wPtr , [255 0 0]);将矩形填充为红色
  - Screen('FillRect', wPtr, [0 255 0], [0 0 50 50])将指定区域填充为绿色

```
clear all

screens=0;
[wPtr,rect]=Screen('OpenWindow',screens, 0, []);
HideCursor;
tic
while toc<5
end
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end
Screen('Close', wPtr);
ShowCursor;
```

Screen 'Flip'?

- Flip页面/窗口切换
  - 把后台窗口切换到前台来，使后台窗口的内容快速呈现
  - 使要呈现的后台窗口从显示器刷新的最新一帧(frame)开始，使呈现时间更精确

```
clear all

screens=0;
[wPtr,rect]=Screen('OpenWindow',screens, 0, []);
HideCursor;
tic
while toc<5
end
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end
Screen('Close', wPtr);
ShowCursor;
```

Screen 'Close'?

- 关闭指定窗口
- Screen('CloseAll')关闭所有窗口

【练习】

- 在FlipScreen中尝试做一些改动，看看效果有什么不同之处？

```
clear all

screens=0;
% [wPtr,rect]=Screen('OpenWindow',screens,0, []);
[wPtr,rect]=Screen('OpenWindow',screens, [0 0 255]);% open a winodw with blue screen
HideCursor;
tic
while toc<3    % waiting for 3 secs
end
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
% Screen('FillRect',wPtr,white);
Screen('FillRect',wPtr,[0 255 0],[500 500 1000 1200]); % fill in a green rectangle in [500 500 1000 2000]
Screen(wPtr, 'Flip');
tic
while toc<1 % waiting for 1 secs
end
Screen('Close', wPtr);
ShowCursor;
```

【PTB的Screen函数核心功能】

- 呈现刺激
  - 灵活呈现丰富的视觉、听觉刺激
  - 时间精度高（ms）
- 记录行为反应（下一章内容）
  - 准确灵敏
  - 时间精度高（ms）

【PTB使用流程】

- 屏幕/页面（screen）=窗口（window）

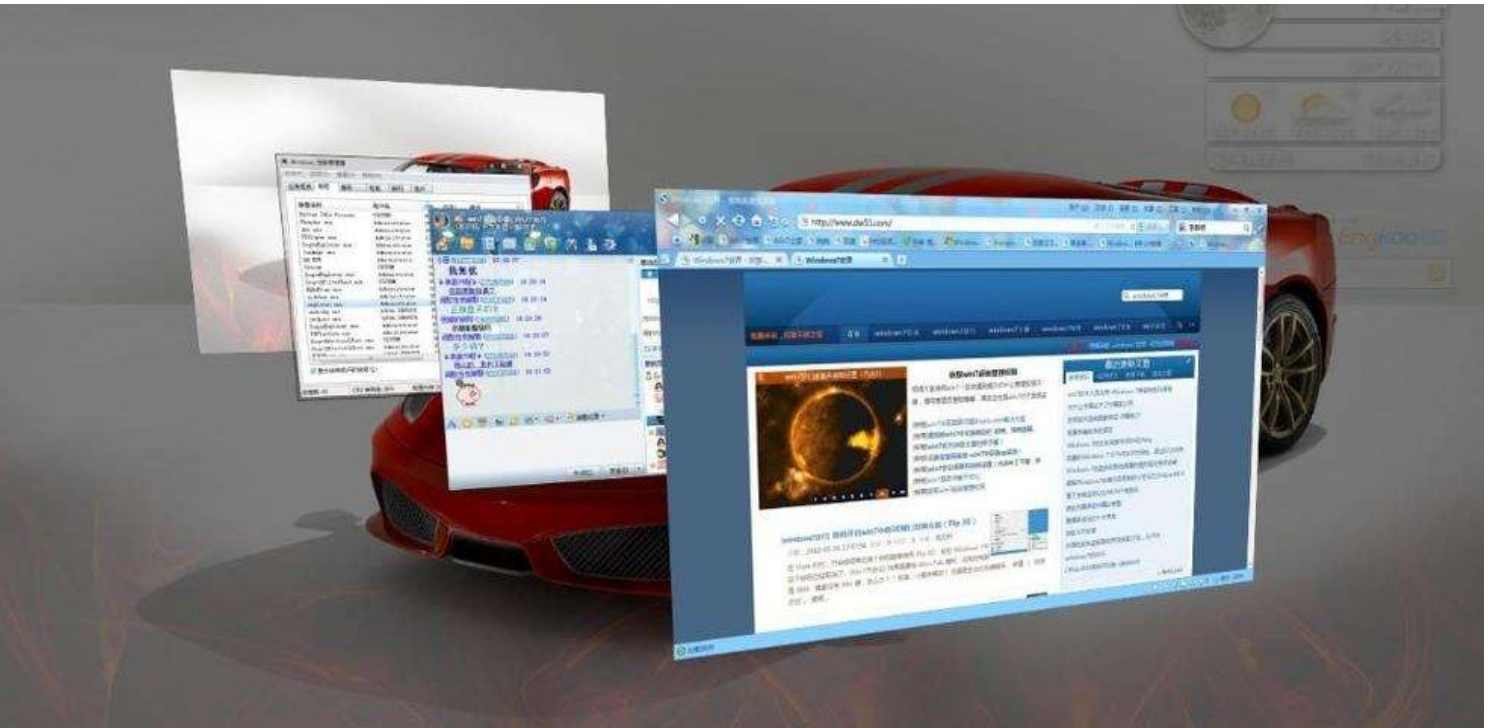
打开窗口	• OpenWindow
窗口内容管理	• DrawLine、DrawText、DrawTextures等 • FillRect、FillOval等
窗口调用&反应记录	• Flip • KbCheck、KbName
关闭窗口	• Close • CloseAll

【PTB的窗口内容管理】

- 通过Screen子函数的输入参数，对窗口内容的属性进行精细控制，如视觉刺激的：
  - 形状
  - 颜色
  - 位置
  - 大小
  - 其他
- [newX,newY]=Screen('DrawText', windowPtr, text [,x] [,y] [,color] [,backgroundColor] [,yPositionIsBaseline] [,swapTextDirection]);

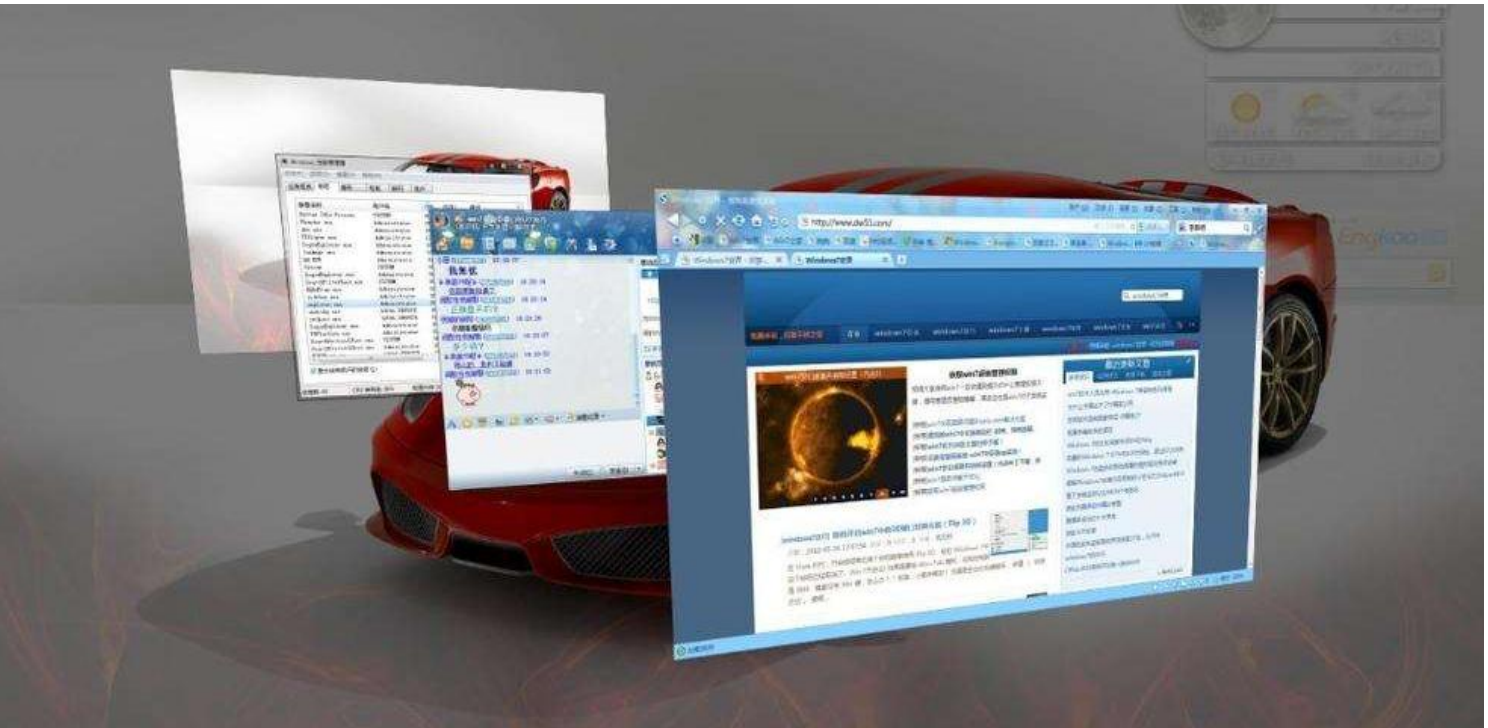
【PTB的窗口切换及时间控制】

- 窗口内容管理先在后台缓冲页面（offscreen）进行。
- 需要显示时，通过Screen的子函数Flip及其参数，控制后台缓冲页面切换到前台主页面（onscreen）。
- 此时后台缓冲页面就空了，等待新的窗口内容管理。



【PTB的窗口切换及时间控制】

- 可实现快速呈现，获得快速流畅的画面。
- 可控制刺激呈现与显示器的帧（frame）刷新同步。
- 可控制在指定时间呈现刺激，以及刺激的总呈现时间。



PTB卡住时的处理方法

- Screen函数在运行过程中会隐藏MATLAB命令窗口。
- 如果Screen函数运行中出现问题屏幕会卡住：
  - Ctrl+C, Alt+Tab把MATLAB命令窗口调出来，输入
    - clear Screen
    - clear mex
    - Screen('CloseAll')
  - Ctrl+Alt+Delete从任务管理器关闭Screen程序

6.4 使用Psychotoolbox的函数

- 新建1个m文件，命名为FunkyScreen。
  - 先写清楚标题header信息。

```
1  %  FunkyScreen.m
2  %
3  %  opens a window using psychtoolbox 3,
4  %  makes the window to show shapes and texts in a funky way
5  %
6  %  Written by YW and revised in 2022
```

```
screenNum=0;
[wPtr,rect]=Screen('OpenWindow',screenNum);
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
% blank the Screen and wait a second
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
HideCursor;
tic
while toc<5
end
```

- 打开一个窗口
- 填充为白色
- 切换到前台来显示
- 暂停5秒钟

```
% make a rectangle in the middle of the screen flip colors and size
Screen('FillRect',wPtr,black);
vbl=Screen(wPtr, 'Flip'); % collect the time for the first flip with vbl
monitorFlipInterval=Screen('GetFlipInterval', wPtr);
```

Screen 'GetFlipInterval'?

- 将窗口填充为黑色，切换到前台来显示
- vbl为切换/呈现窗口开始的时刻点
- GetFlipInterval获得每刷新一帧所用的时间
  - 如果刷新频率是60HZ，monitorFlipInterval是1000/60=17毫秒

提醒：忽略变量名称的长短，关键是了解变量的含义

```
flipSpd=13;
```

提醒：flipSpd可以设置为任意数值

```
for i=1:10
    Screen('FillRect',wPtr,[0 0 255], [100 150 200 250]);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
    % flip 13 frames after vbl
    Screen('FillRect',wPtr,[255 0 0], [100 150 400 450]);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
end
```

- 将窗口的指定方形区域[100 150 200 250]填充为蓝色
- 把后台窗口切换到前台显示，距离上一次切换的时间间隔是flipSpd（13）帧后：vbl+flipSpd*monitorFlipInterval（将帧数转换为秒）
- 再将窗口指定方形区域填充为红色
- 把后台窗口切换到前台显示，距离上一次切换的时间间隔是flipSpd（13）帧后：vbl+flipSpd*monitorFlipInterval（将帧数转换为秒）
- 连续运行10次：for i=1:10...end

算一算：蓝色方形的呈现时间是多少秒？红色呢？

```
% blank the screen and wait a while
Screen('FillRect',wPtr,black);
Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
tic
while toc<1
end
```

- 将窗口填充为黑色
- 将黑色的后台窗口切换到前台显示，与循环最后一次切换后台窗口时间间隔为13帧：vbl+flipSpd*monitorFlipInterval（将帧数转换为秒）
- 等待1秒钟

```
% make circles flip colors& size
Screen('FillRect',wPtr,black);
vbl=Screen(wPtr, 'Flip');
for i=1:10
    Screen('FillOval',wPtr,[0 180 255], [ 500 500 600 600]);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
    Screen('FillOval',wPtr,[0 255 0], [ 400 400 900 700]);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
end
```

Screen 'FillOval'?

- 将窗口填充为黑色
- 切换到前台显示
- 将指定区域填充为指定颜色的圆形/椭圆形
- 间隔13帧切换到前台显示
- 换一种颜色填充指定区域
- 间隔13帧切换到前台显示
- 运行10次

```
% blank the screen and wait a while
Screen('FillRect',wPtr,black);
Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
tic
while toc<1
end
```

- 将窗口填充为黑色
- 间隔13帧后切换到前台
- 等待1秒钟

```
% make lines that flip colors size  & position
Screen('FillRect',wPtr,black);
vbl=Screen(wPtr, 'Flip');
for i=1:10    Screen 'DrawLine'?
    Screen('DrawLine',wPtr,[0 255 255], 500, 200, 700 ,600, 5);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
    Screen('DrawLine',wPtr,[255 255 0], 100, 600, 600 ,100, 5);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
end
```

- 在窗口画指定颜色的直线
- 提供直线的起点和终点坐标及粗细

```
% blank the screen and wait a while
Screen('FillRect',wPtr,black);
Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
tic
while toc<1
end
```

- 将窗口填充为黑色
- 间隔13帧后调用到前台
- 等待1秒钟

- 画各种线条和形状的函数
- % Draw lines and solids like QuickDraw and DirectX (OS 9 and Windows):

- currentbuffer = Screen('SelectStereoDrawBuffer', windowPtr [, bufferid] [, param1]);
- Screen('DrawLine', windowPtr [,color], fromH, fromV, toH, toV [,penWidth]);

- Screen('DrawArc',windowPtr,[color],[rect],startAngle,arcAngle)
- Screen('FillArc',windowPtr,[color],[rect],startAngle,arcAngle)

- Screen('FillRect', windowPtr [,color] [,rect]);
- Screen('FrameRect', windowPtr [,color] [,rect] [,penWidth]);

- Screen('FillOval', windowPtr [,color] [,rect]);
- Screen('FrameOval', windowPtr [,color] [,rect] [,penWidth] [,penHeight] [,penMode]);

```
% combine the stimuli
Screen('FillRect',wPtr,black);
vbl=Screen(wPtr, 'Flip');
for i=1:10
    Screen('FillRect',wPtr,[0 0 255], [100 150 200 250]);
    Screen('DrawLine',wPtr,[0 255 255], 500, 200, 700 ,600, 5);
    Screen('FillOval',wPtr,[0 180 255], [ 500 500 600 600]);
    Screen('TextSize', wPtr , 150);
    Screen('DrawText', wPtr, 'FUNKY!!', 200, 20, [255 50 255]);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));

    Screen('FillRect',wPtr,[255 0 0], [100 150 400 450]);
    Screen('FillOval',wPtr,[0 255 0], [ 400 400 900 700]);
    Screen('DrawLine',wPtr,[255 255 0], 100, 600, 600 ,100, 5);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
end
```

Screen 'DrawText'? Screen 'TextSize'?

- 将长方形、圆形/椭圆形及直线在窗口同时呈现
- 各种图形的先后顺序有讲究，后画的图会叠加在先画的图上面
- 窗口同时呈现文本，提供字号（150）、起点坐标（200，20）等

- 在窗口写文本的函数
- % Draw Text in windows

- textModes = Screen('TextModes');
- oldCopyMode=Screen('TextMode', windowPtr [,textMode]);
- oldTextSize=Screen('TextSize', windowPtr [,textSize]);
- oldStyle=Screen('TextStyle', windowPtr [,style]);
- [oldFontName,oldFontNumber]=Screen(windowPtr,'TextFont' [,fontNameOrNumber]);
- [normBoundsRect, offsetBoundsRect]=Screen('TextBounds', windowPtr, text);
- [newX,newY]=Screen('DrawText', windowPtr, text [,x] [,y] [,color] [,backgroundColor] [,yPositionIsBaseline]);
- oldTextColor=Screen('TextColor', windowPtr [,colorVector]);
- oldTextBackgroundColor=Screen('TextBackgroundColor', windowPtr [,colorVector]);

```
% blank the screen and wait a second
Screen('FillRect',wPtr,black);
Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
tic
while toc<1
end

Screen('CloseAll');
ShowCursor
```

- 关闭所有窗口

【练习】

- 在FunkyScreen中尝试做一些改动，看看效果有什么不同之处？

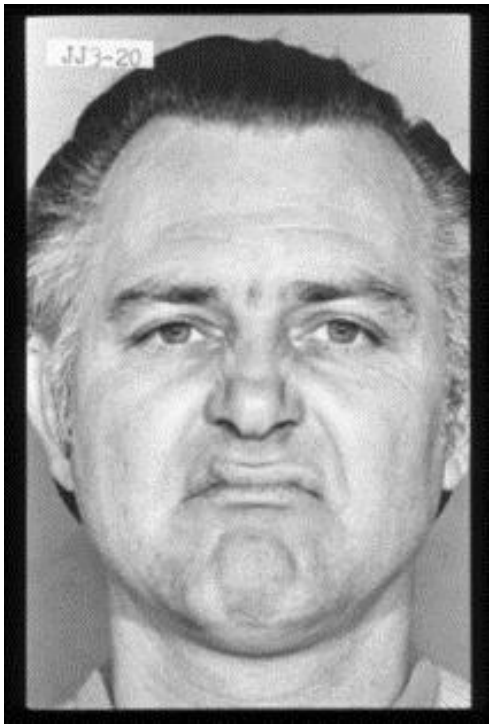
```
% flipSpd=13;
flipSpd=20;

for i=1:10
    Screen('FillRect',wPtr,[0 0 255], [100 150 200 250]);
    % vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
    vbl=Screen(wPtr, 'Flip', vbl+((flipSpd+10)*monitorFlipInterval));
    % flip 13 frames after vbl
    Screen('FillRect',wPtr,[255 0 0], [100 150 400 450]);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
end
```

试一试：flipSpd， vbl的变量名称都可以修改

```
Screen('FillRect',wPtr,black);
vbl=Screen(wPtr, 'Flip');
for i=1:10
    % Screen('FillOval',wPtr,[0 180 255], [ 500 500 600 600]);
    Screen('FillOval',wPtr,[255 180 255], [ 200 200 700 700]);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
    Screen('FillOval',wPtr,[0 255 0], [ 400 400 900 700]);
    vbl=Screen(wPtr, 'Flip', vbl+(flipSpd*monitorFlipInterval));
end
```

【加载已有图片/图像texture】

- 如果想呈现的图像是已有的图片（例如，情绪图片等）该如何操作？



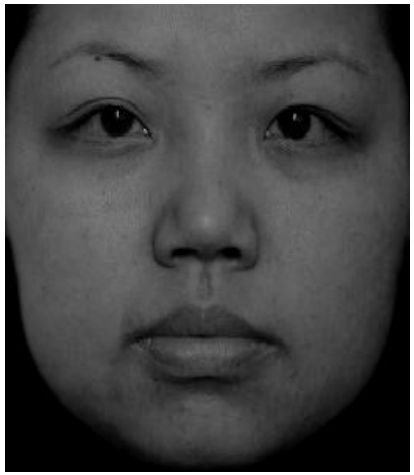
- PutUpImread.m文件
  - imread可读取bmp, jpg, gif, png, tif等图片类型

```
screen=0;
[wPtr,rect]=Screen('OpenWindow',screen);
HideCursor;
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end
```

Help imread

```
disgustface = imread('040.tif');
textureIndex=Screen('MakeTexture', wPtr, disgustface);
Screen('DrawTexture', wPtr, textureIndex );
Screen(wPtr, 'Flip');
tic
while toc<6
end

Screen('Close', wPtr);
ShowCursor;
```

- 如果想呈现多张图片该如何操作？






```
screen=0;
[wPtr,rect]=Screen('OpenWindow',screen);
HideCursor;
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end

path=pwd;
picfolder='\FACES';
pathpic=[pwd picfolder];
file=dir(pathpic);
for i=3:length(file)
    face{i-2} = imread([pathpic '\' file(i).name],'bmp');
    textureIndex=Screen('MakeTexture', wPtr, face{i-2});
    ret=Screen('PreloadTextures', wPtr, textureIndex);
    Screen('DrawTexture', wPtr, textureIndex );
    Screen(wPtr, 'Flip');
    tic
    while toc<2
    end
end
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end

Screen('Close', wPtr);
ShowCursor;
```

- PutUpImreadMultiple.m文件
  - pathpic设定图片文件夹所在路径，或者直接输入路径；

    ```
    pathpic='F:\2022年心理学实验软件应用\第六章的m文件\FACES';
    ```

  - 用dir(pathpic)获得该路径下所有文件和路径的属性

Q: 如何获得该路径下所有bmp图片文件的属性？

A:filebmp=dir([pathpic '*.bmp']);

```
screen=0;
[wPtr,rect]=Screen('OpenWindow',screen);
HideCursor;
black=BlackIndex(wPtr);
white=WhiteIndex(wPtr);
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end

path=pwd;
picfolder='\FACES';
pathpic=[pwd picfolder];
file=dir(pathpic);
for i=3:length(file)
    face{i-2} = imread([pathpic '\' file(i).name],'bmp');
    textureIndex=Screen('MakeTexture', wPtr, face{i-2});
    ret=Screen('PreloadTextures', wPtr, textureIndex);
    Screen('DrawTexture', wPtr, textureIndex );
    Screen(wPtr, 'Flip');
    tic
    while toc<2
    end
end
Screen('FillRect',wPtr,white);
Screen(wPtr, 'Flip');
tic
while toc<5
end

Screen('Close', wPtr);
ShowCursor;
```

- 利用循环语句呈现多张图片

Q:图片的呈现时间还可以用什么语句控制？

A:flip

【PTB使用流程】

- 屏幕/页面（screen）=窗口（window）

打开窗口	• OpenWindow
窗口内容管理	• DrawLine、DrawText、DrawTextures等 • FillRect、FillOval等
窗口调用&反应记录	• Flip • KbCheck、KbName
关闭窗口	• Close • CloseAll

【PTB的Screen函数核心功能】

- 呈现刺激
  - 灵活呈现丰富的视觉、听觉刺激
  - 时间精度高（ms）

- 记录行为反应（下一章内容）
  - 准确灵敏
  - 时间精度高（ms）

Q: 如何利用PTB呈现经典Stroop实验中的刺激？

受电脑具体配置的影响，正式实验需检测确保时间精度

```
WARNING: This session of your experiment was run by you with the setting Screen('Preference', 'SkipSyncTests', 2).
WARNING: This means that some internal self-tests and calibrations were skipped. Your stimulus presentation timing
WARNING: may have been wrong. This is fine for development and debugging of your experiment, but for running the real
WARNING: study, please make sure to set Screen('Preference', 'SkipSyncTests', 0) for maximum accuracy and reliability.
```

6.5 使用Psychotoolbox呈现实验刺激

- 新建一个m文件，命名为StroopStim
  - Octave6.1对中文文本的兼容性不佳，中文刺激需要采用图片格式。

StroopStim.m

```
1  % StroopStim.m
2  %
3  % This program presents stroop like stimuli.
4  % Written by YW, 2022
```

提醒：忽略变量名称的长短，关键是了解变量的含义

```
try
  clear all
  close all

  % predefine parameters
  BlockNum=1;% how may block in this experiment
  StimDuration=2000;% 2000ms
  ISImin=1000;% the minmum interstimulus interval (ISI)
  ISImax=1500;% the maxmum interstimulus interval (ISI)
  TxtSize=100;% the size of the texts
  TxtFont='Arial'; % the font of the texts

  Words={'Red', 'Green', 'Blue'};
  StimColor={[255 0 0];[0 255 0];[0 0 255]};
  count=0;
  for i=1:length(Words)
    for j=1:length(StimColor)
      count=count+1;
      Stim(count).word=Words{i}; % stimulus word
      Stim(count).color=StimColor{j};% stimulus color
    end
  end
```

- 实验组数、刺激时间大小等信息存入变量，方便灵活修改
- 利用结构变量Stim组织刺激序列

```
  % open window
  screenNum=0;
  [wPtr,rect]=Screen('OpenWindow',screenNum);
  HideCursor;
  black=BlackIndex(wPtr);
  white=WhiteIndex(wPtr);
  monitorFlipInterval=Screen('GetFlipInterval', wPtr);
  hz=FrameRate(wPtr);
  StimDuration=round((StimDuration/1000)*hz); % the number of frames for stimulusduration
  ISIrange=round(((ISImax-ISImin)/1000)*hz);
  ISImin=round((ISImin/1000)*hz);
  ISI=round(rand(BlockNum,length(Stim))*ISIrange+ISImin);% the number of frames for neach trial's ISI in a range
```

- 利用PTB获取刷新频率、每帧刷新时间等信息，将刺激呈现时间和两个刺激之间的随机间隔时间（ISI）换算成具体的帧数

```
41    % blank the Screen
42    starttime=Screen('Flip',wPtr); % collect the time for the first flip with starttime
43    for ii=1:BlockNum
44
45      Screen('TextSize', wPtr , TxtSize);
46      Screen( 'TextFont', wPtr ,TxtFont);
47      StimRand=Stim(randperm(length(Stim))); % randomize all stimuli
48
49      for jj=1:length(Stim)
50        TxtRect= Screen('TextBounds',wPtr,StimRand(jj).word); % get the size of each stimulus text
51        TxtLoc=[round(rect(3)/2-TxtRect(3)/2),round(rect(4)/2-TxtRect(4)/2)]; % compute the upperleft
52        Screen('DrawText', wPtr, StimRand(jj).word, TxtLoc(1), TxtLoc(2),StimRand(jj).color);
53        starttime=Screen(wPtr, 'Flip', starttime+(ISI(ii,jj)*monitorFlipInterval)); % time in secs
54        starttime=Screen('Flip', wPtr,  starttime+(StimDuration*monitorFlipInterval));% time in secs
55      end
56    end
57    ISIlast=round(rand(1)*ISIrange+ISImin); % the last trial's ISI
58    % blank the screen and wait a second
59    Screen('FillRect',wPtr,white);
60    Screen(wPtr, 'Flip', starttime+(ISIlast*monitorFlipInterval));
61
62    Screen('CloseAll');
63    ShowCursor
64  catch
65    Screen('CloseAll')
66    rethrow(lasterror)
67  end
```

- 获取实验开始的系统时间starttime
- 用randperm对刺激序列进行随机化
- 用TextBounds获取刺激文本大小
- 计算当文本在屏幕中央呈现时左上角和右下角的坐标TxtLoc

测试过程中输出错误信息的方法

- try catch end 命令

```
1  try
2     % write here the code written
3  catch
4     Screen('CloseAll')
5     rethrow(lasterror)
6  end
```

- 命令语句写在此处

开始PRACTICE啦！