PsyBuilder 0.1
A Brief Step by Step Tutorial

Yang Zhang[1] and Zhicheng Lin[2]

1 Department of Psychology, Soochow University, Suzhou, China
2 Applied Psychology Program, The Chinese University of Hong Kong, Shenzhen, China

July 30, 2020

Introduction

For most researchers in experimental psychology and related fields, computer programming is an inevitable part of the job, which often takes a considerable amount of time and energy from the researcher, and is prone to errors. Psychtoolbox (Brainard, 1997; Kleiner, Brainard, & Pelli, 2007; Pelli, 1997) provides a powerful and free toolkit in MATLAB and Octave, but because it requires coding, many people for one reason or another do not have the skill set to take advantage of it, and even skilled users often find themselves having to spend a lot of time just trying to implement a new experiment. With this in mind, we have designed a graphical user interface, named PsyBuilder, which for the first time allows users to compile an experiment within a relatively short amount of time simply by dragging and dropping.

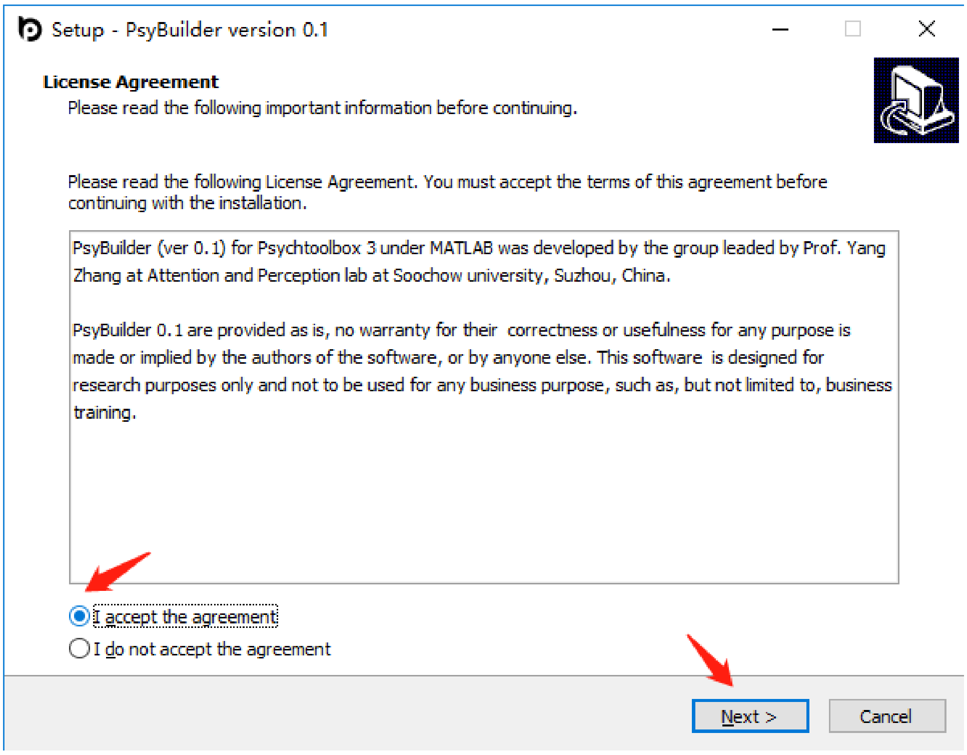
In this brief manual, we will explain how to install PsyBuilder (Chapter 1), its main user interface, the basic settings and operations in it (Chapter 2). Finally, we use the classic Stroop color word interference paradigm as an example to briefly explain how to use PsyBuilder to build an experiment from scratch (Chapter 3).

Chapter 1: Installation Guide

Download PsyBuilder

Visit the following website to download the latest version of PsyBuilder:
http://web.suda.edu.cn/yzhangpsy/projects.html

Installation in Windows



Setup - PsyBuilder version 0.1

License Agreement
Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

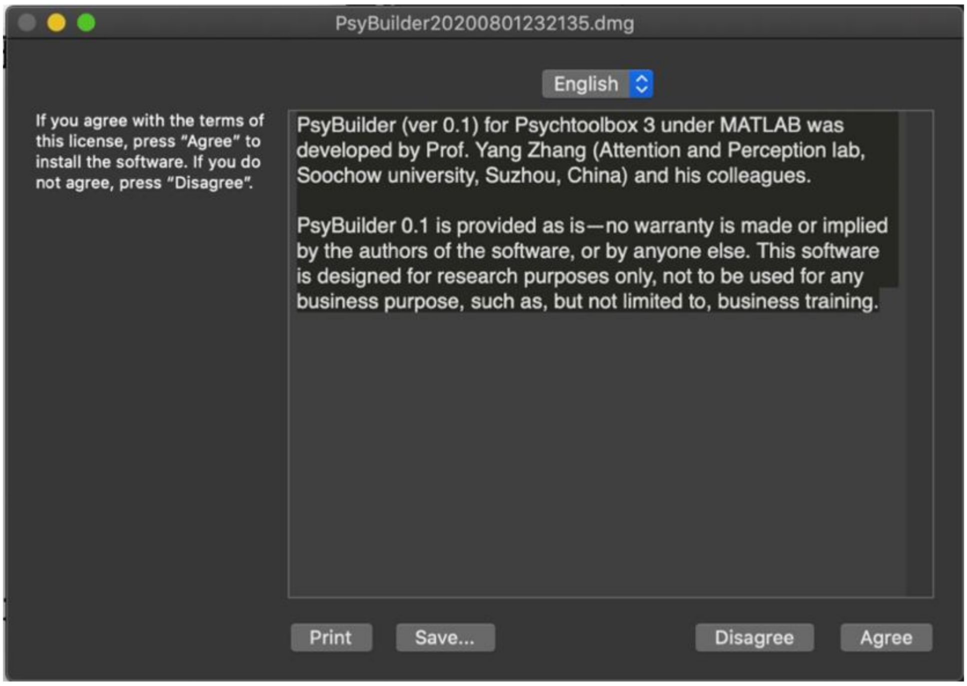
PsyBuilder (ver 0.1) for Psychtoolbox 3 under MATLAB was developed by the group leaded by Prof. Yang Zhang at Attention and Perception lab at Soochow university, Suzhou, China.

PsyBuilder 0.1 are provided as is, no warranty for their correctness or usefulness for any purpose is made or implied by the authors of the software, or by anyone else. This software is designed for research purposes only and not to be used for any business purpose, such as, but not limited to, business training.

● I accept the agreement
○ I do not accept the agreement

Unzip the downloaded installation file (PsyBuilderxxxx.zip, xxxx is a string of numbers, representing the time when the software is generated) and double-click the PsyBuilder.exe installation file. Select “I accept the agreement” on the License Agreement interface (see above), and click “Next” to install.

Installation in Mac OS

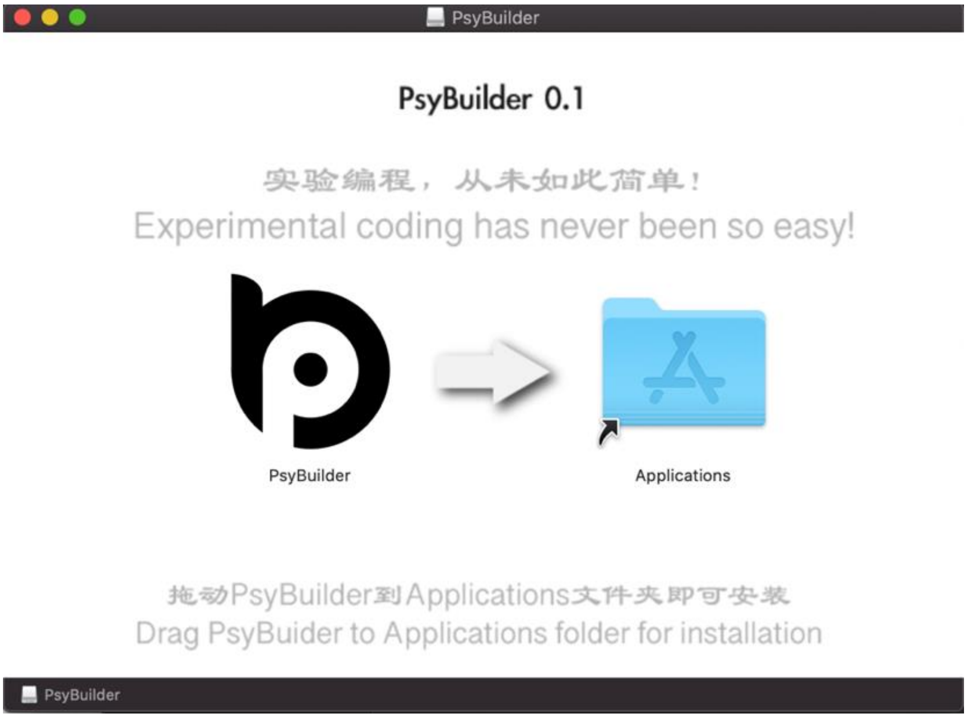


If you agree with the terms of this license, press "Agree" to install the software. If you do not agree, press "Disagree".

PsyBuilder (ver 0.1) for Psychtoolbox 3 under MATLAB was developed by Prof. Yang Zhang (Attention and Perception lab, Soochow university, Suzhou, China) and his colleagues.

PsyBuilder 0.1 is provided as is—no warranty is made or implied by the authors of the software, or by anyone else. This software is designed for research purposes only, not to be used for any business purpose, such as, but not limited to, business training.

Double-click the downloaded PsyBuilderxxxx.dmg file (xxxx is a string of numbers, representing the time point when the software is generated). Click “Agree” on the license instructions interface (see above) to agree with the terms of the license.



PsyBuilder 0.1

实验编程，从未如此简单！

Experimental coding has never been so easy!

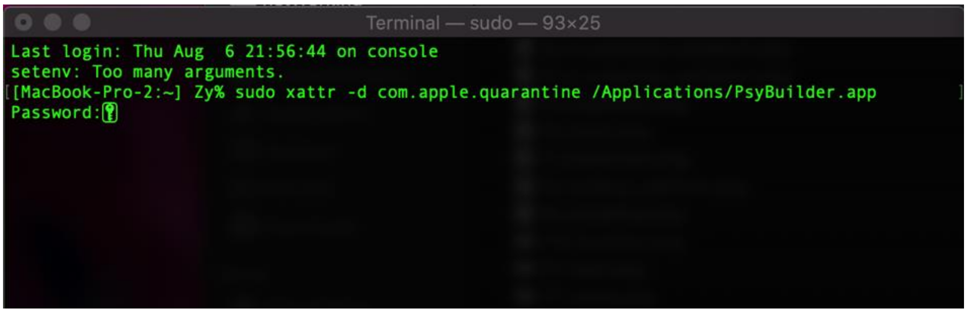
拖动PsyBuilder到Applications文件夹即可安装

Drag PsyBuider to Applications folder for installation

In the pop-up interface (see above), drag the PsyBuilder icon to the Applications icon on the right and release it to complete the installation.

Because of the specifics in the Mac OS system, we also need to give permissions to the installed PsyBuilder before it can be used. First, perform any of the following operations to open the Terminal window:

1. Click the "Launchpad" icon in the Dock, type "Terminal" in the search bar, and then double-click "Terminal".
2. Use the shortcut key “Command” + “Space”, type "Terminal" in the search bar, and then press the “Return” key on the "Terminal" link that appears.
3. In the "Finder", open the "Applications/Utilities" folder, and double-click to start the "Terminal".

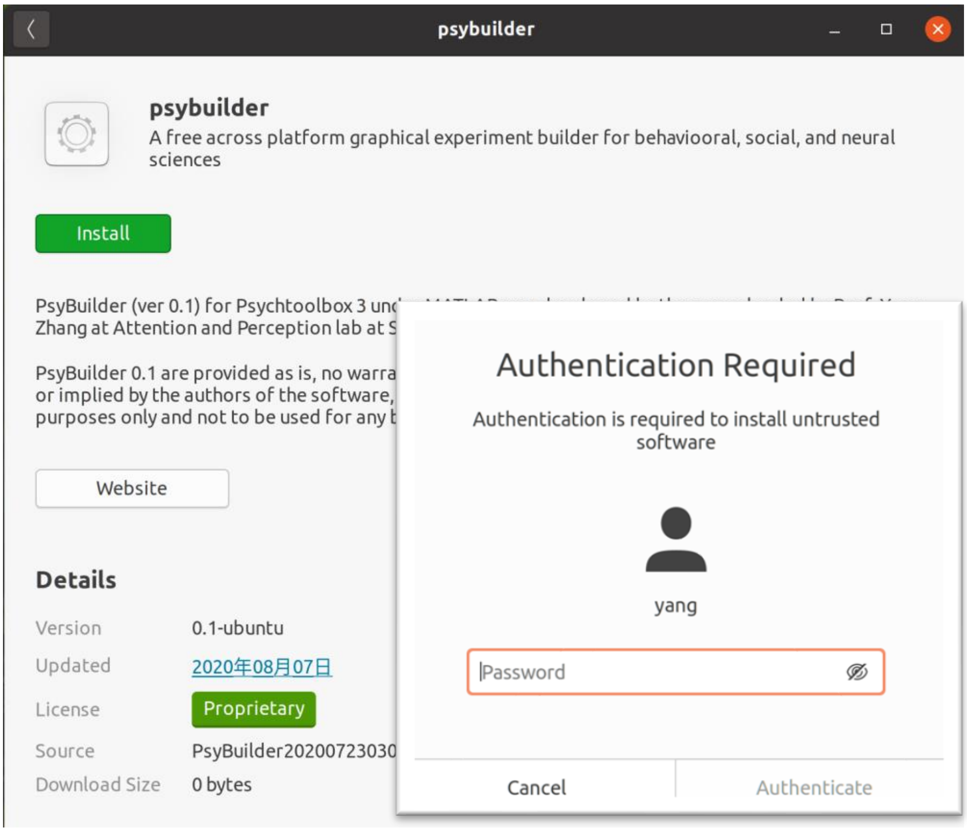


```
Last login: Thu Aug  6 21:56:44 on console
setenv: Too many arguments.
[[MacBook-Pro-2:~] Zy% sudo xattr -d com.apple.quarantine /Applications/PsyBuilder.app       ]
Password:
```

Then, enter the following code in the opened terminal window, press “Return” to run it, and then enter the password of the current computer user:

sudo xattr -d com.apple.quarantine /Applications/PsyBuilder.app

Installation in Linux (Ubuntu or Debian)



psybuilder

A free across platform graphical experiment builder for behaviooral, social, and neural sciences

PsyBuilder (ver 0.1) for Psychtoolbox 3 und
Zhang at Attention and Perception lab at S

PsyBuilder 0.1 are provided as is, no warra
or implied by the authors of the software,
purposes only and not to be used for any t

Details

Version	0.1-ubuntu
Updated	2020年08月07日
License	Proprietary
Source	PsyBuilder20200723030
Download Size	0 bytes

Authentication Required

Authentication is required to install untrusted software

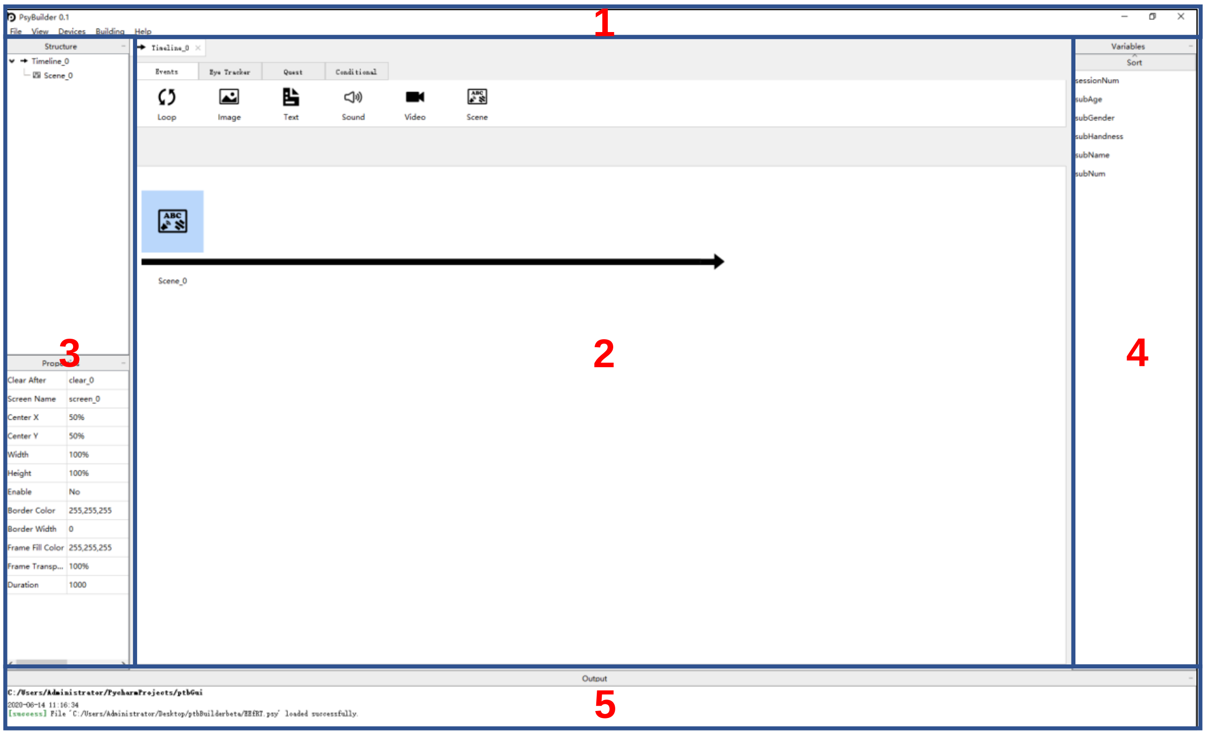
yang

Password

Double-click the downloaded PsyBuilderxxxx.deb file (xxxx is a string of numbers, representing the time when the software is generated), which will open the system's default release graphics software center. Click “Installation” in the interface (see above). In the pop-up "Authentication" dialog box, enter the password of the current computer user and click "Authenticate" to install.

Chapter 2: Brief Introduction to the Interface

Main Interface



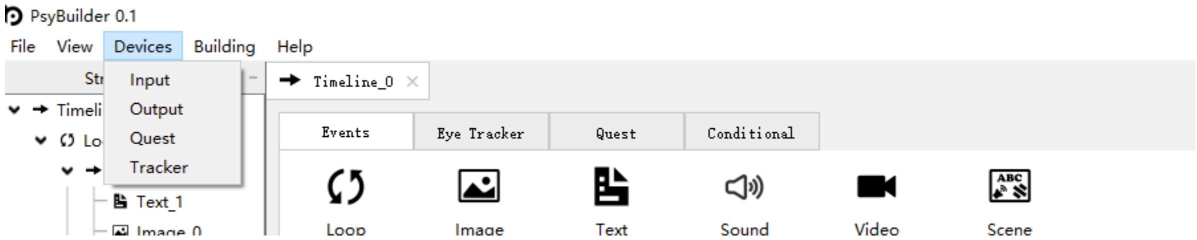
Clear After	clear_0
Screen Name	screen_0
Center X	50%
Center Y	50%
Width	100%
Height	100%
Enable	No
Border Color	255,255,255
Border Width	0
Frame Fill Color	255,255,255
Frame Transp...	100%
Duration	1000

```
C:/Users/Administrator/PycharmProjects/ptbGui
2020-06-14 11:16:34
[success] File 'C:/Users/Administrator/Desktop/ptbBuilderbeta/EEfRT.psy' loaded successfully.
```

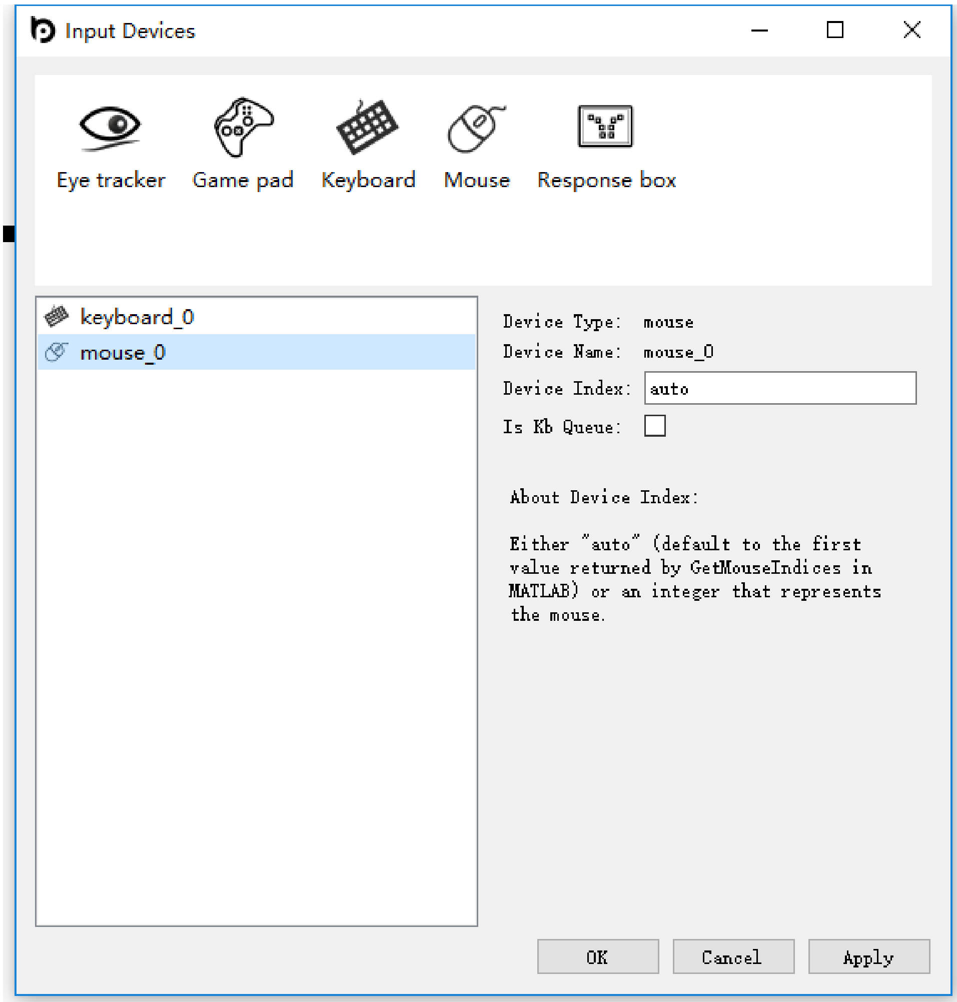
The main interface consists of 5 parts, as represented by numbers 1 to 5 in the figure above:

1. *Menu*: explained below
2. *Timeline*: the main operating window
3. Experiment *Structure* (top) and *Properties* (bottom)
4. *Variables*: show all the referenceable variables of the current event
5. *Output*: show compilation status, error prompts, etc.

Menu Bar



Devices: This menu allows users to specify the input, output, eye tracker, and other equipment required by the current experiment, as well as the initial structure of the QUEST procedure (for quick estimation of perceptual thresholds).

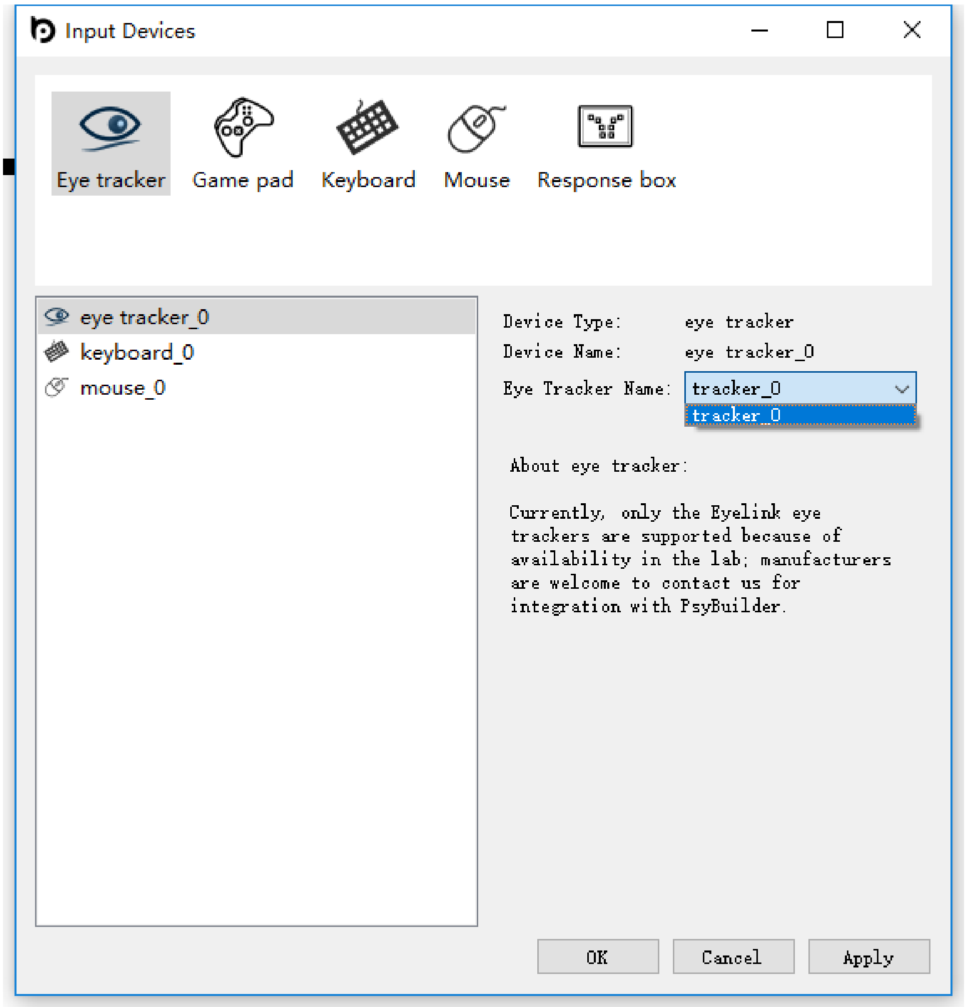


Device Type: mouse
Device Name: mouse_0
Device Index: auto
Is Kb Queue: □

About Device Index:

Either "auto" (default to the first value returned by GetMouseIndices in MATLAB) or an integer that represents the mouse.

Input Devices: Users can define the input devices needed for the experiment, including keyboards, mice, game consoles, response boxes produced by Cedrus (https://cedrus.com), eye trackers, etc. Selected by default are keyboard and mouse. To add other input devices, just drag the corresponding icon from the upper part to the blank space at the lower left. Note: only the input devices defined here can be used for the experiment as the input.



Device Type: eye tracker

Device Name: eye tracker_0

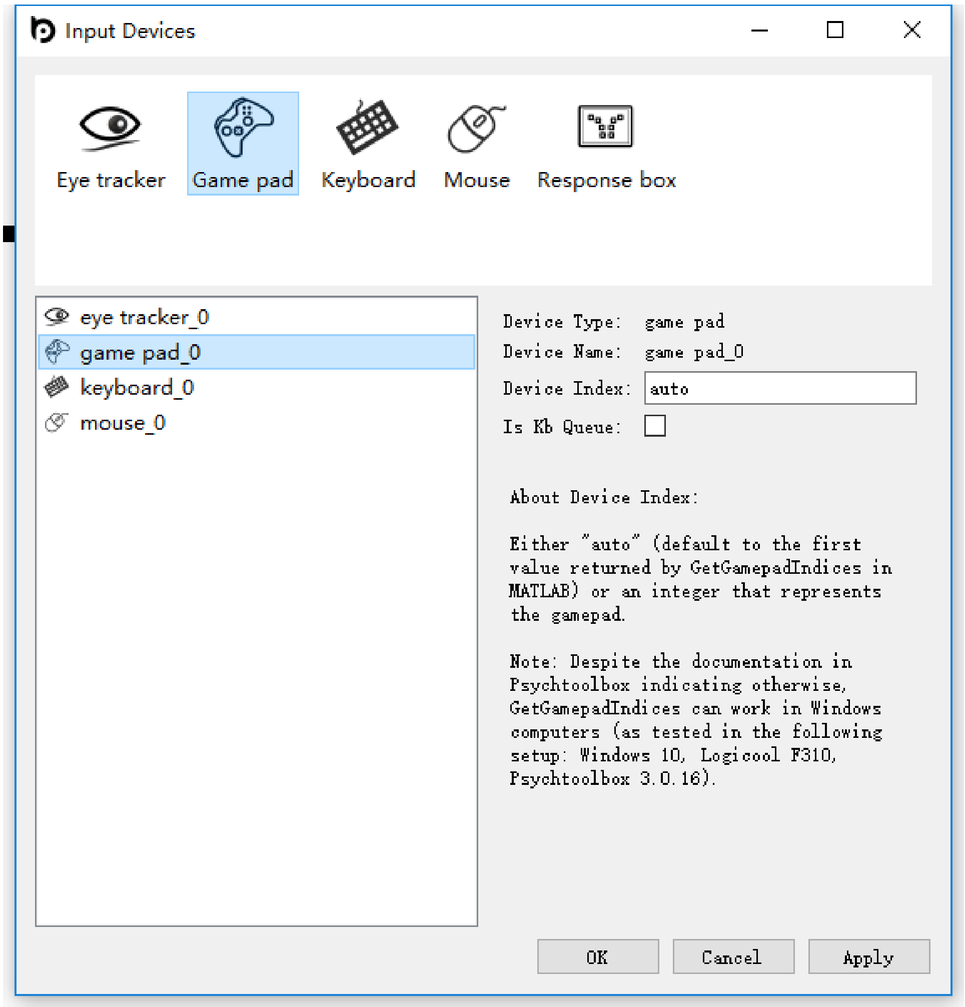
Eye Tracker Name: tracker_0

tracker_0

About eye tracker:

Currently, only the Eyelink eye trackers are supported because of availability in the lab; manufacturers are welcome to contact us for integration with PsyBuilder.

Eye tracker: Define an eye tracking device as an input device available for the current experiment. Currently supported reaction modes include: start blink, end blink, start saccade, end saccade, start fixation, end fixation, and fixation update. The seven reaction types are listed in the reaction code, represented by numbers 3 to 9 respectively.



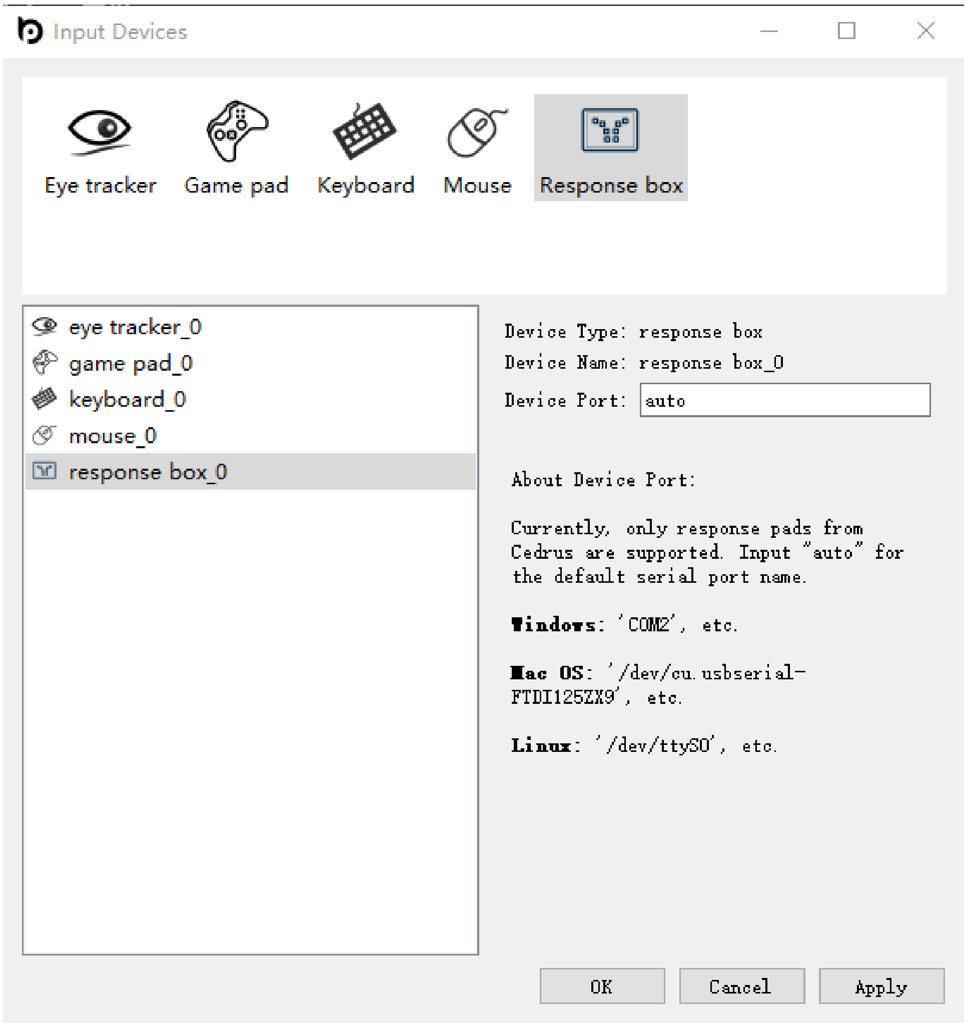
Device Type: game pad
Device Name: game pad_0
Device Index: auto
Is Kb Queue: □

About Device Index:

Either "auto" (default to the first value returned by GetGamepadIndices in MATLAB) or an integer that represents the gamepad.

Note: Despite the documentation in Psychtoolbox indicating otherwise, GetGamepadIndices can work in Windows computers (as tested in the following setup: Windows 10, Logicool F310, Psychtoolbox 3.0.16).

Game pad: Define a game console handle as an input device available for the current experiment. The checkbox for Is Kb Queue determines whether the button presses of the current device are recorded by KbQueue or KbCheck (for a detailed explanation of the KbQueue and related functions, please refer to http://psychtoolbox.org/docs /KbQueueCreate).



Device Type: response box

Device Name: response box_0

Device Port: auto

About Device Port:

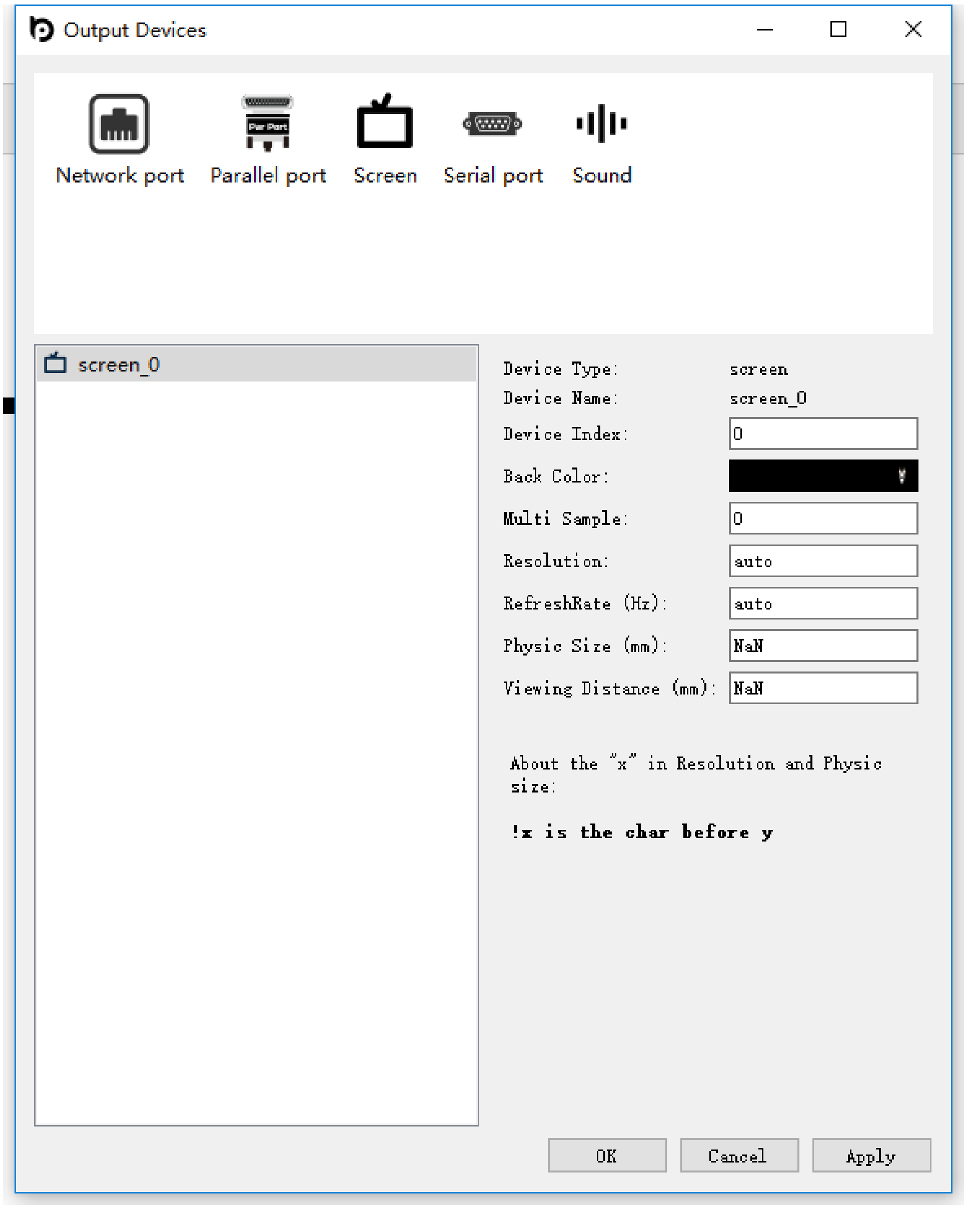
Currently, only response pads from Cedrus are supported. Input "auto" for the default serial port name.

Windows: 'COM2', etc.

Mac OS: '/dev/cu.usbserial-FTDI125ZX9', etc.

Linux: '/dev/ttyS0', etc.

Response Box: Define a response box device as an input device available for the current experiment. Currently only reaction boxes from Cedrus are supported.



Device Type: screen

Device Name: screen_0

Device Index: 0

Back Color:

Multi Sample: 0

Resolution: auto

RefreshRate (Hz): auto

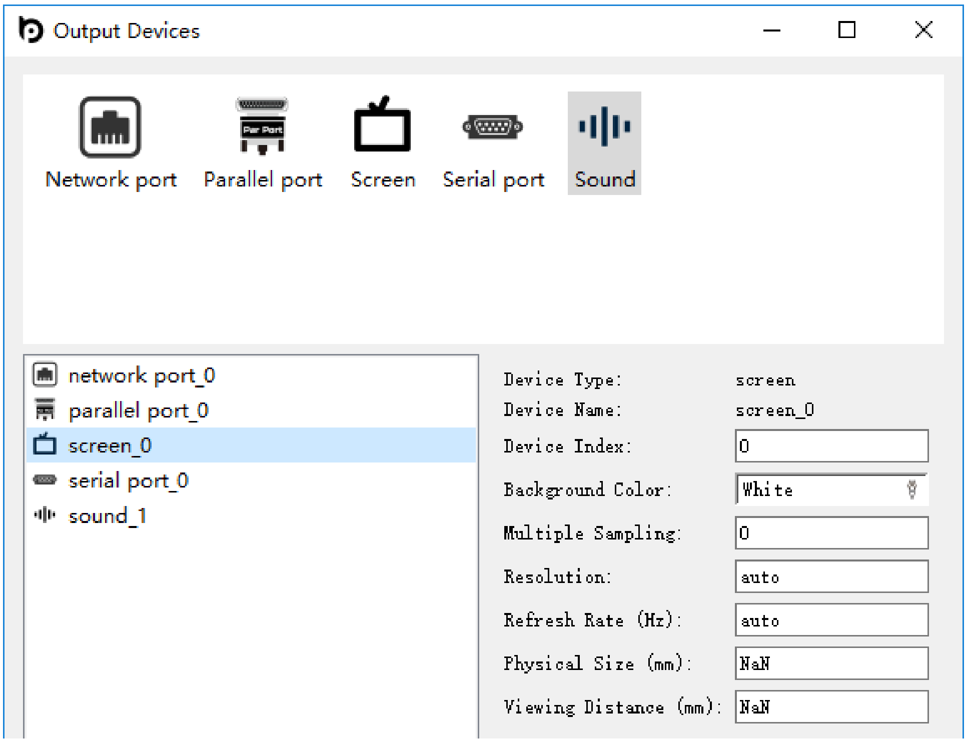
Physic Size (mm): NaN

Viewing Distance (mm): NaN

About the "x" in Resolution and Physic size:

!x is the char before y

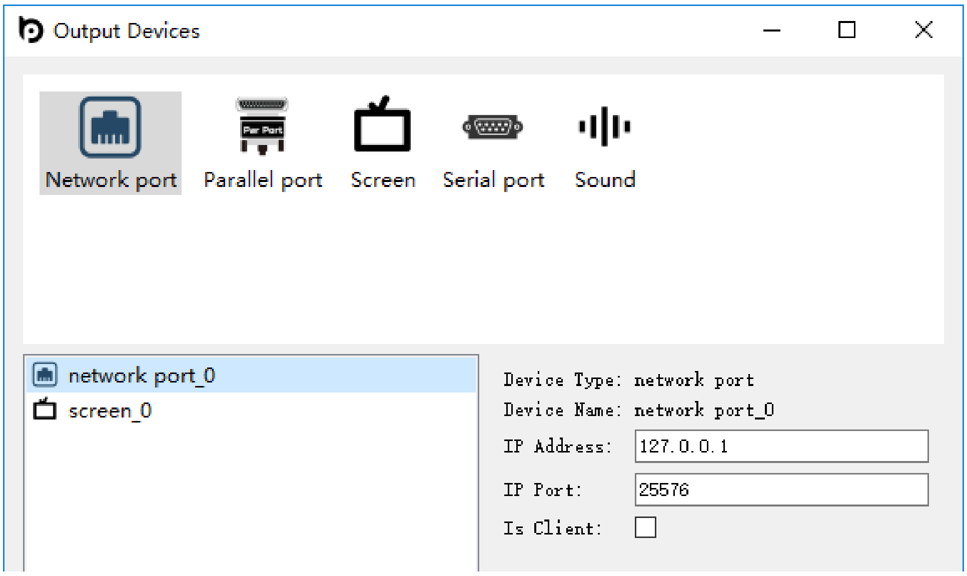
Output Devices: Users can define the output devices needed for the experiment, including display screen, sound device, parallel port, serial port, and network port. The default option includes display devices. To add other output devices, drag the corresponding icon from the top to the blank space at the bottom left. Note: Only the output devices defined here can be used by the current experiment as an output.



network port_0
parallel port_0
screen_0
serial port_0
sound_1

Device Type: screen
Device Name: screen_0
Device Index: 0
Background Color: White
Multiple Sampling: 0
Resolution: auto
Refresh Rate (Hz): auto
Physical Size (mm): NaN
Viewing Distance (mm): NaN

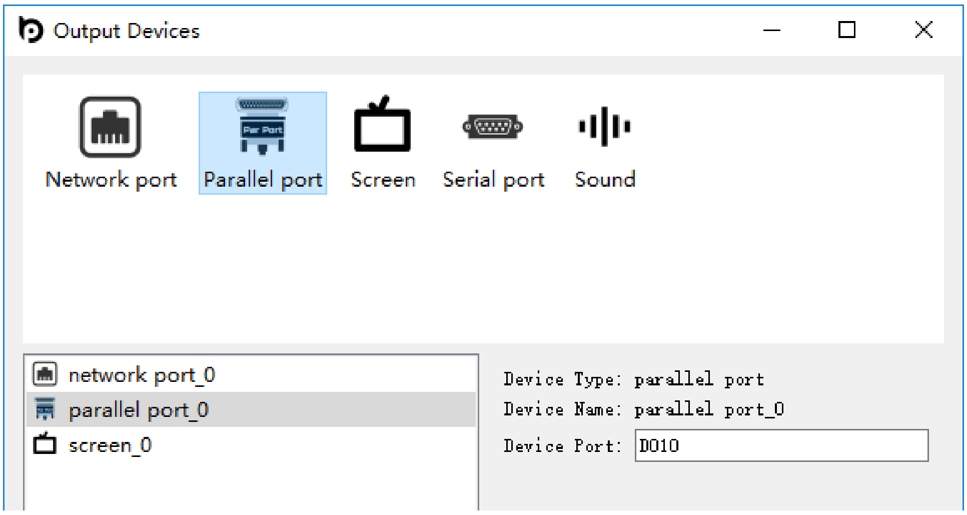
Screen: Define the screen device on which visual stimuli will appear.



network port_0
screen_0

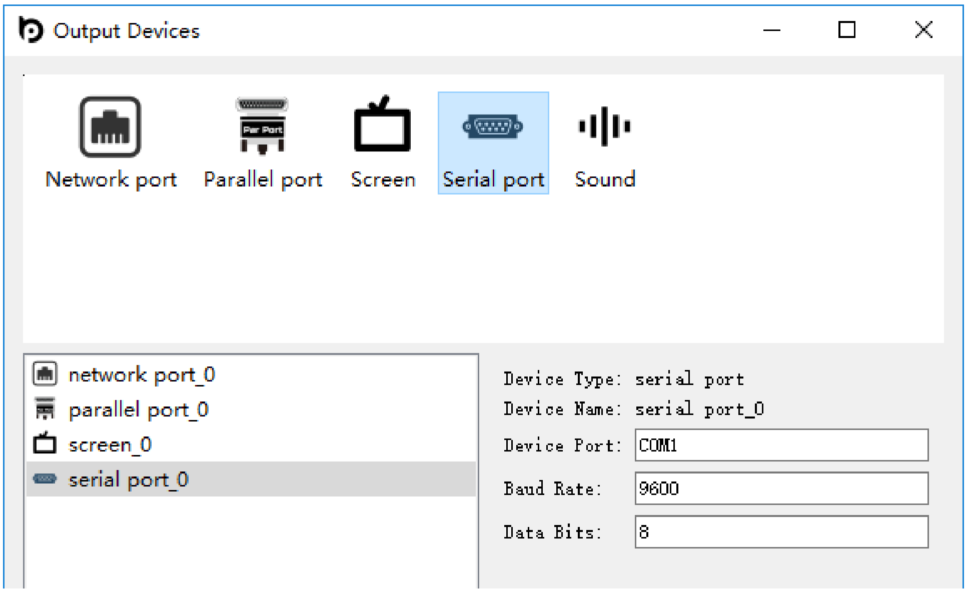
Device Type: network port
Device Name: network port_0
IP Address: 127.0.0.1
IP Port: 25576
Is Client: □

Network port: Define a network port for communication with external devices. The checkbox for Is Client determines whether the current device is used as a client or as a server.



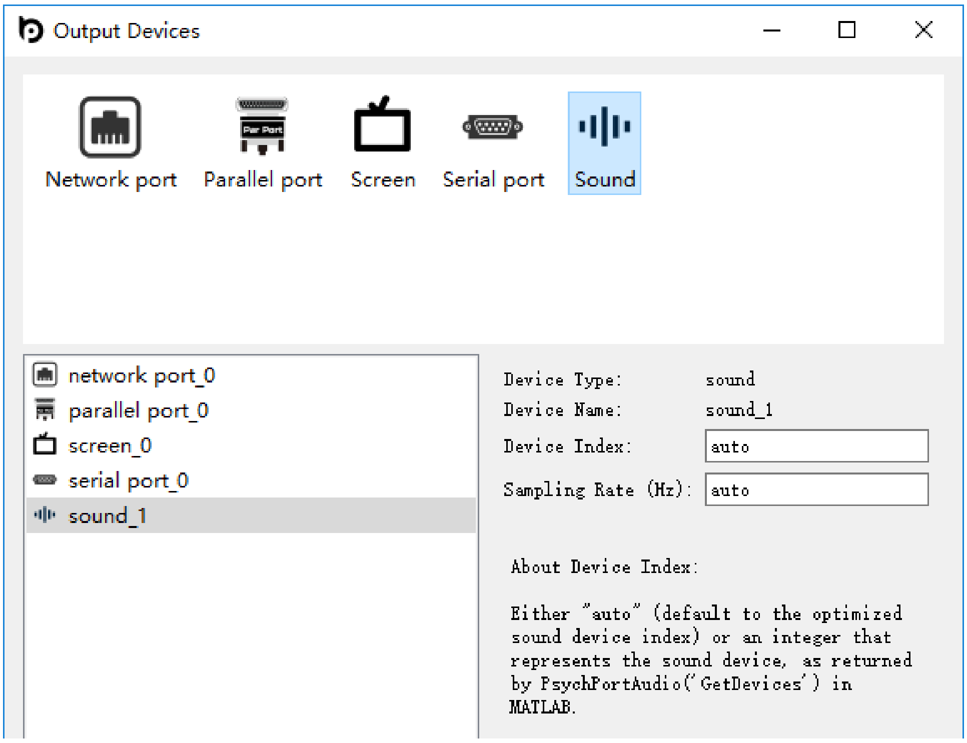
Device Type: parallel port
Device Name: parallel port_0
Device Port: D010

Parallel port: Define a parallel port for communication with external devices. Note that under the Linux operating system, if you are using a PCI-to-parallel device, you can use the command lspci -v to find the hardware address of the parallel device.



Device Type: serial port
Device Name: serial port_0
Device Port: COM1
Baud Rate: 9600
Data Bits: 8

Serial port: Define a serial port for communication with external devices.

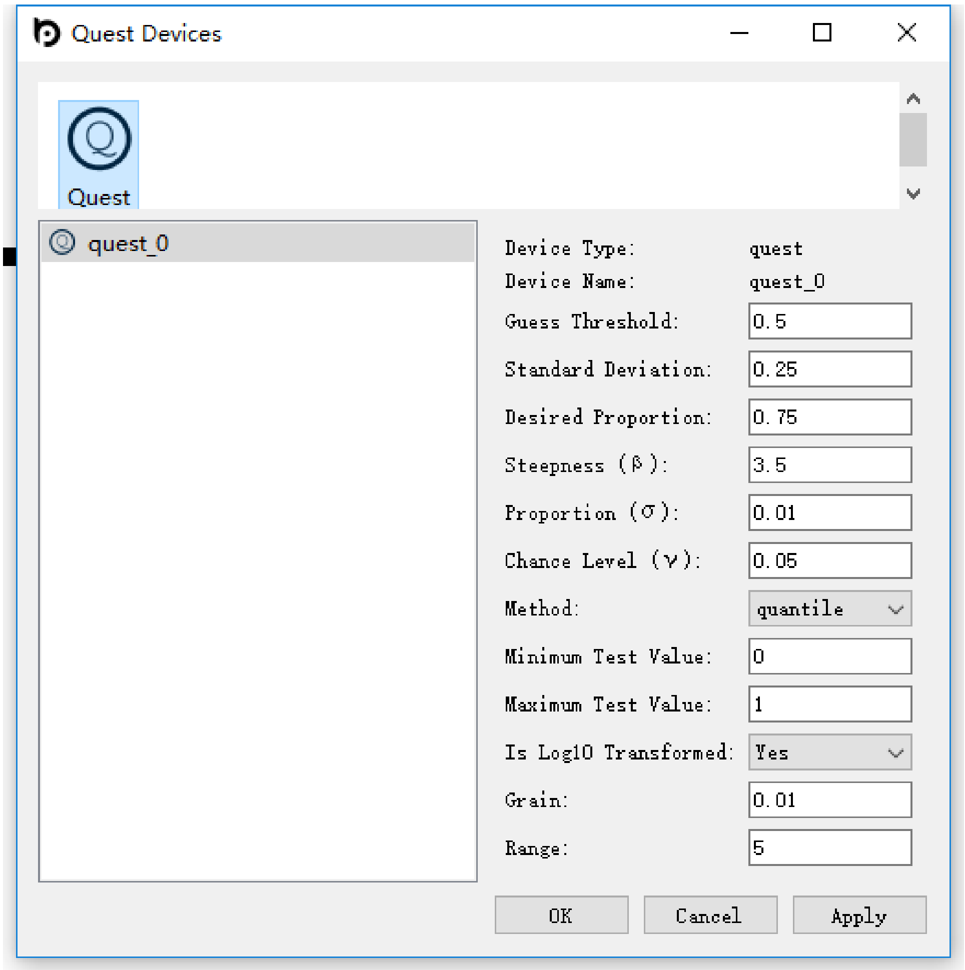


Device Type: sound
Device Name: sound_1
Device Index: auto
Sampling Rate (Hz): auto

About Device Index:

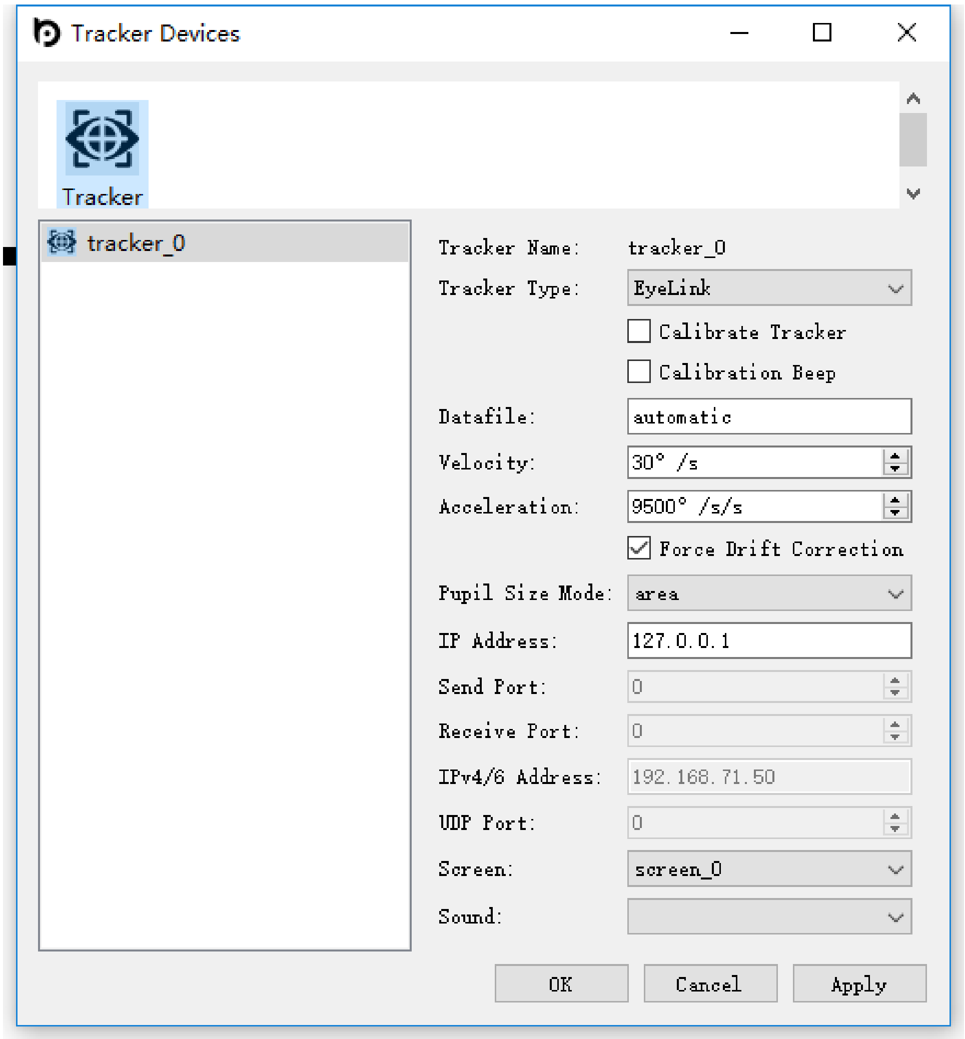
Either "auto" (default to the optimized sound device index) or an integer that represents the sound device, as returned by PsychPortAudio('GetDevices') in MATLAB.

Sound: Define a sound device for playing audio files. You need to fill in the hardware ID of the device you would like to use in the box *Device Index*; you can type in “PsychPortAudio('GetDevices')” in MATLAB to get the sound device information of the current system. By default, the most suitable one will be selected based on the current operating system (see http://psychtoolbox.org/docs/PsychPortAudio-Open for details).



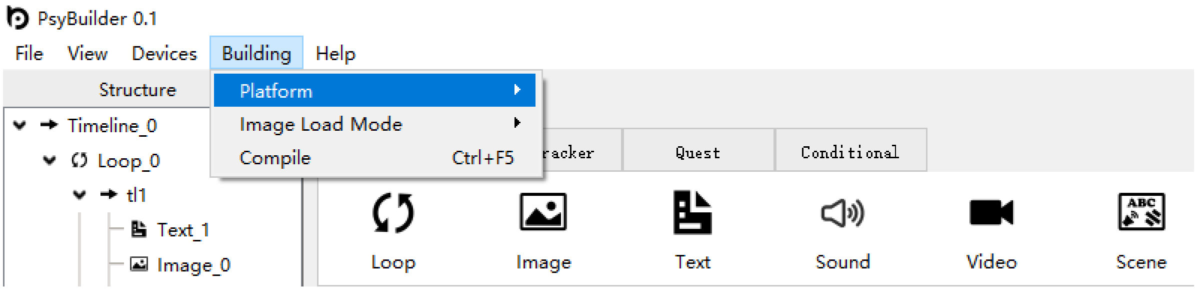
Device Type: quest
Device Name: quest_0
Guess Threshold: 0.5
Standard Deviation: 0.25
Desired Proportion: 0.75
Steepness (β): 3.5
Proportion (σ): 0.01
Chance Level (γ): 0.05
Method: quantile
Minimum Test Value: 0
Maximum Test Value: 1
Is Log10 Transformed: Yes
Grain: 0.01
Range: 5

Quest: Define the initial parameters in QUEST. Quest is an adaptive method for quickly estimating perceptual thresholds using Bayesian inference. For a detailed description of this method, see Watson and Pelli (1983).

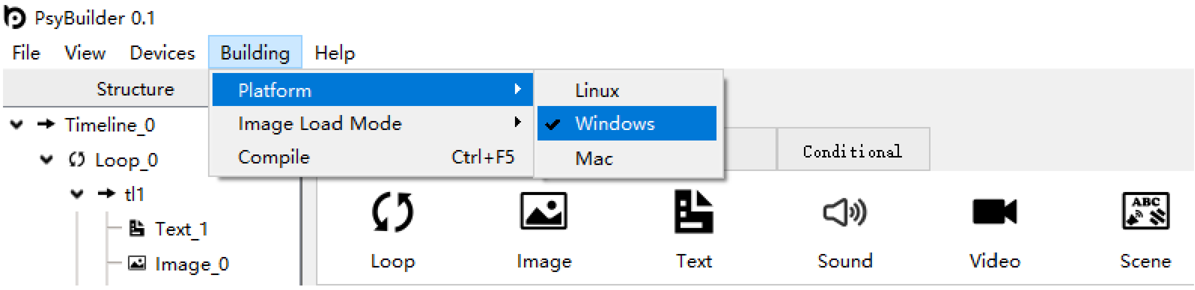


Tracker Name: tracker_0
Tracker Type: EyeLink
□ Calibrate Tracker
□ Calibration Beep
Datafile: automatic
Velocity: 30° /s
Acceleration: 9500° /s/s
☑ Force Drift Correction
Pupil Size Mode: area
IP Address: 127.0.0.1
Send Port: 0
Receive Port: 0
IPv4/6 Address: 192.168.71.50
UDP Port: 0
Screen: screen_0
Sound:

Eye tracker: Define the relevant parameters of the eye tracker for the current experiment (currently only EyeLink devices are supported). Note: After setting up the eye tracker, you also need to include “Eye tracker” as an input device in “Input Devices” before it can be used in the current experiment.



Building: Settings for compiling the current graphic representation of the experiment into a MATLAB m file.



Platform: Define the operating system platform under which the m file will be run. The default is the operating system you are using right now.

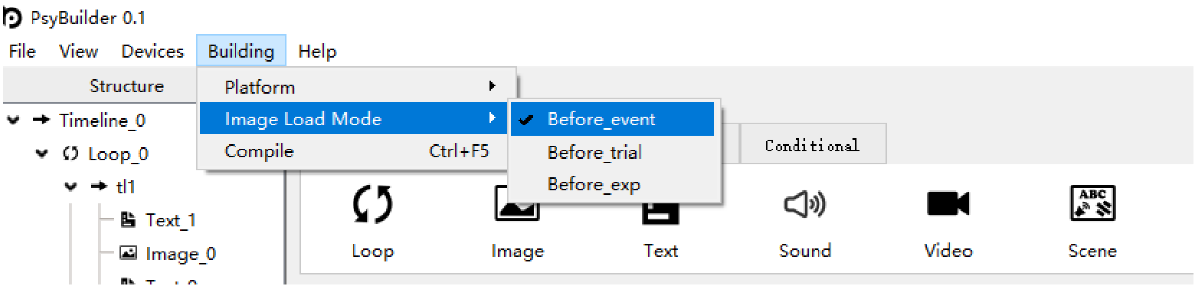


Image Load Mode: Define the mode to load the pictures. There are three modes, Before_event, Before_trial, and Before_exp, that is, to load the pictures from the hard disk to the RAM (random access memory) and then to the graphics card memory before each event, before each trial, and before the start of the experiment, respectively. Currently, only the first option is supported.

Compile: Compile the current graphic representation of the experiment into a MATLAB m file.

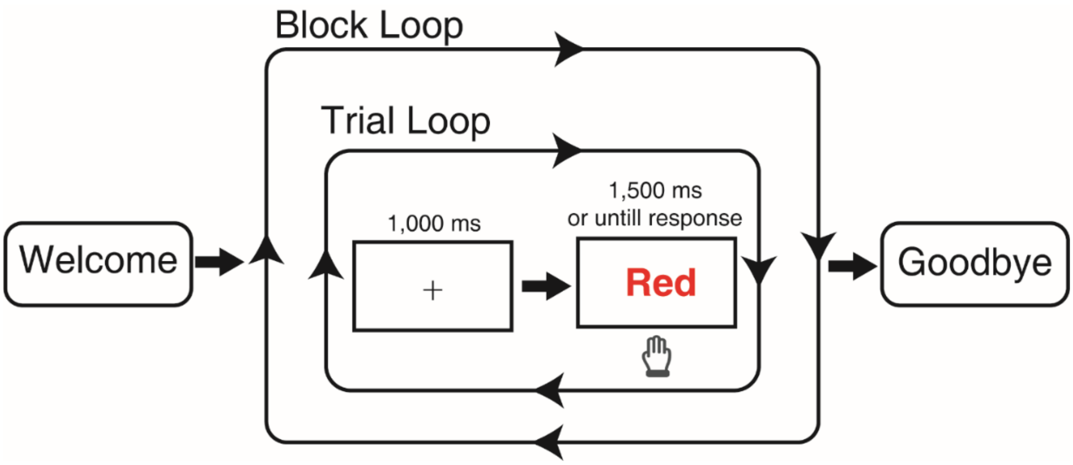
Chapter 3: Building a Stroop Task

The Classic Stroop Task

The color word interference task (Stroop, 1935) is a classic experimental task in psychology. In this task, participants see a different colored word each time and need to respond quickly and accurately to the ink color of the colored word while ignoring its semantic meaning. Here we will provide a step by step tutorial on how to use PsyBuilder to implement this task. The final completed program can also be found in the menu bar, by clicking on *Stroop Task* within *Demos* under the *Help* menu. To implement this task (and other tasks in general), we need to consider the experimental design and then plan out a general implementation strategy, as explained below.

Experimental Design: The task has one factor concerning the congruency between the ink color and the semantic meaning of the word, with three within-subject levels: congruent, incongruent, and neutral. In the congruent condition, the ink color and the semantic color are the same (e.g., Red); in the incongruent condition, the ink color and the semantic color are of different colors (e.g., Red); in the neutral condition, the word itself is unrelated to colors (e.g., Cat).

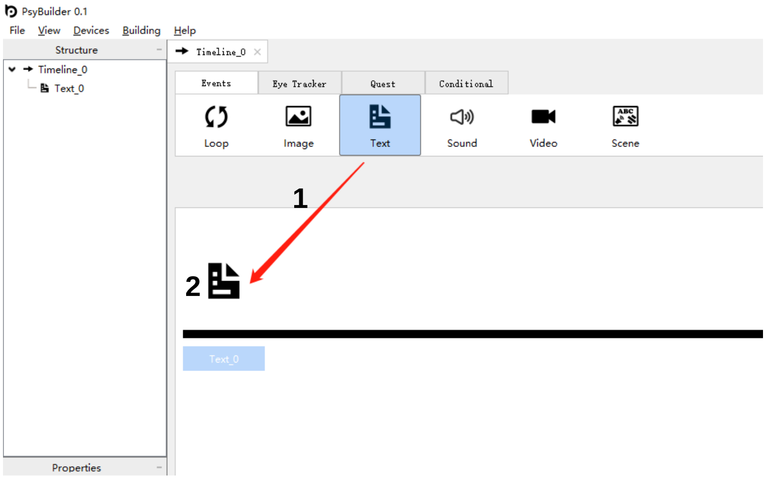
Overall Implementation: The overall flow of the experiment is shown in the figure below. The task contains two blocks, each with 18 trials, six each for the three conditions (congruent, incongruent, and neutral). The ink color of the stimulus is chosen from a set of three colors (red, green and blue), and participants are required to make a judgement regarding the ink color by pressing one of three keys on the keyboard (“a”, “s”, or “d”). In each trial, a fixation point (+) is displayed at the center of the screen for 1,000 milliseconds, and then a colored word appears at the same position until response but up to 1,500 milliseconds. To implement this overall strategy, we illustrate with the following seven major steps.

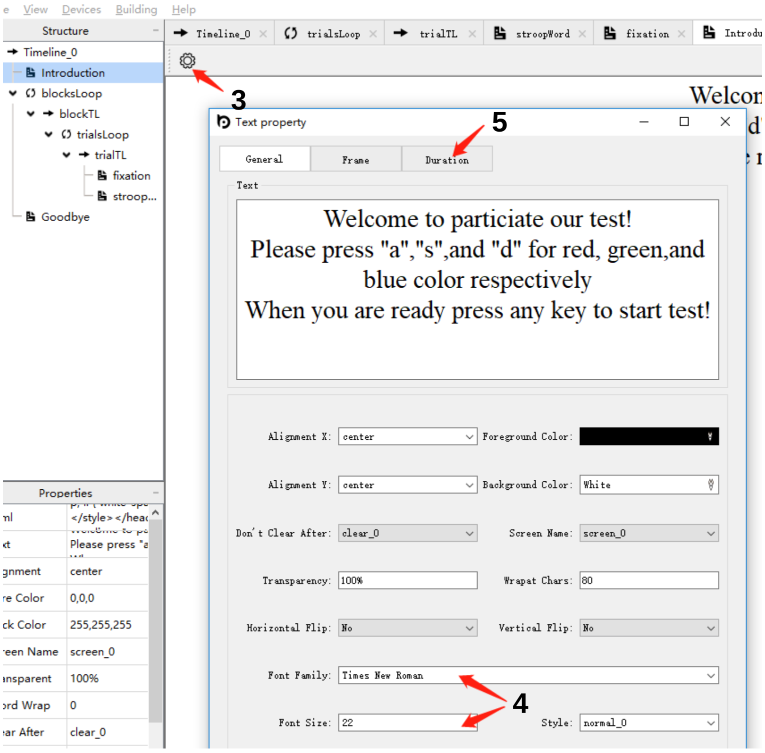


Step 1: Set up Stimulus Presentation and Response Collection Devices

On the main *Menu*, click *Input* under *Devices* to define the response collection device, and click *Output* there to define the stimulus presentation device. Here, since the current task requires only keyboard responses (*Keyboard*) and needs only one stimuli presentation device (*Screen*), and since these are selected by default, no additional operations are needed.

Step 2: Set up Greeting Display


PsyBuilder 0.1
File View Devices Building Help
Structure
Timeline_0
Text_0
Timeline_0
Events Eye Tracker Quest Conditional
Loop Image Text Sound Video Scene
1
2
Text_0
Properties



Welcome to particiate our test!
Please press "a","s",and "d" for red, green,and blue color respectively
When you are ready press any key to start test!

Alignment X: center
Foreground Color:
Alignment Y: center
Background Color: White
Don't Clear After: clear_0
Screen Name: screen_0
Transparency: 100%
Wrapat Chars: 80
Horizontal Flip: No
Vertical Flip: No
Font Family: Times New Roman
Font Size: 22
Style: normal_0

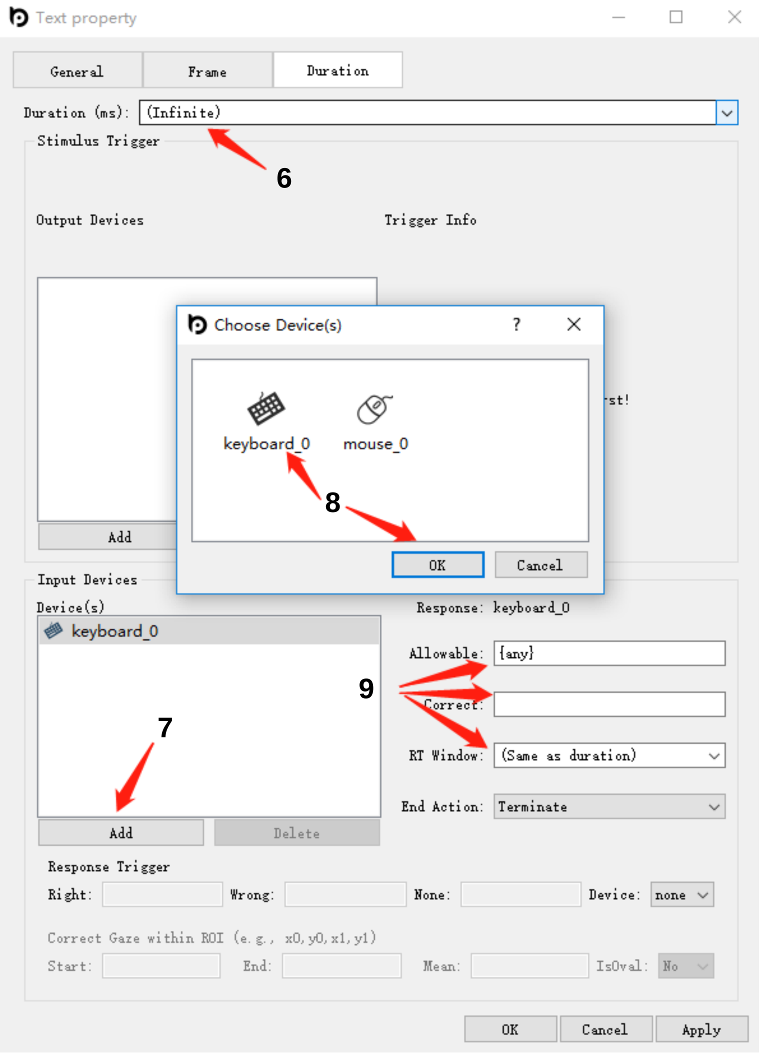
1. In the *Timeline* window, select and drag the *Text* icon in the *Events* tab onto the “Timeline_0” space below. Double-click the name "Text_0" to change it to “Introduction” (or select the name and press “F2”). Note: All event names can only start with a letter and consist of letters, underscores, and numbers.

2. Double-click the *Text* icon in the timeline to open it for editing. Enter the instruction for the experiment in the opened interface.

3. Click the *Setting* icon on the upper left to open the *Text property* box.

4. Under the *General* tab, edit the properties of the text: set the font to “Times New Roman” and the font size to “22”. Then click the *Apply* button below to apply the settings.

5. Click the *Duration* tab to set the presentation time and response collection there as follows.



Duration (ms): (Infinite)

Response: keyboard_0

Allowable: {any}

Correct:

RT Window: (Same as duration)

End Action: Terminate

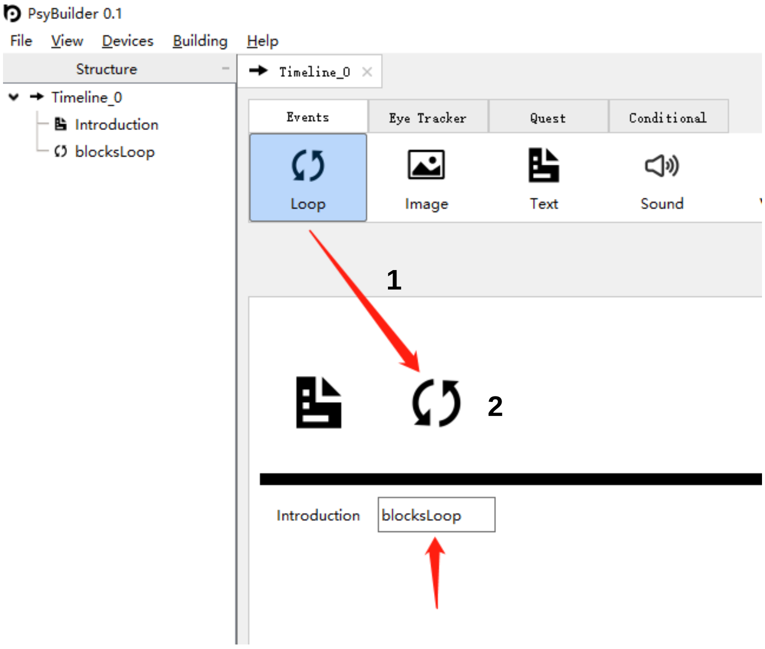
6. In the *Duration (ms)* box, select "(Infinite)" from the drop-down menu to set the presentation time of the current event to be forever until key press.

7. In the *Input Devices* box, click the *Add* button to set the response collection device for the current event.

8. In the pop-up box, select keyboard ("keyboard_0") and then click *OK* to confirm.

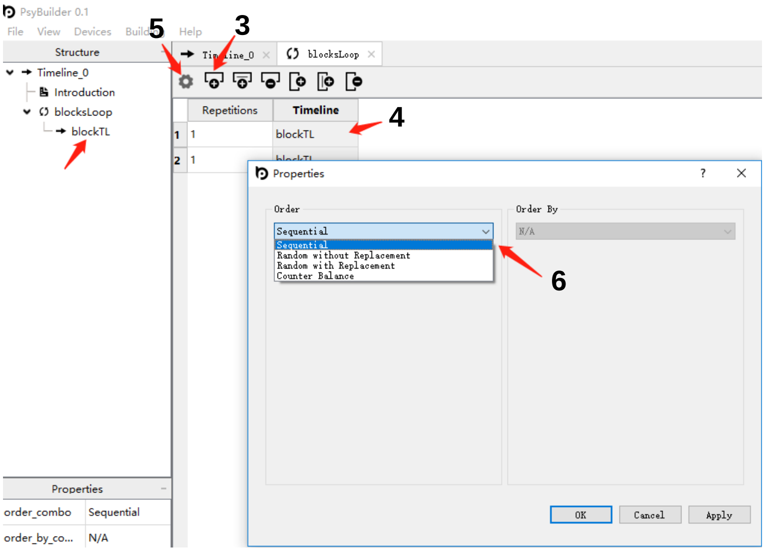
9. Set the response keys in the box *Allowable* ({any} for all keys). Define the correct response button in the box *Correct* (leave it blank if any key is OK). Set the time window for recording the response in the box *RT Window*; "(Same as duration)" means the same as the event duration. Set the action after key press in the box *End Action*; "Terminate", "Terminate Till Release", and "(None)" mean to terminate the event, wait until the key is released to end the current event, and do not respond to the key, respectively.

Step 3: Set up Loops



1. In the *Events* tab, select and drag the *Loop* icon to the timeline and place it behind the "Introduction" event that we just created. Double-click the name to change it to "blocksLoop".

2. Double-click the *Loop* icon in the timeline to set the block loop as follows.



	Repetitions	Timeline
1	1	blockTL
2	1	blockTL

Properties	
order_combo	Sequential
order_by_co...	N/A

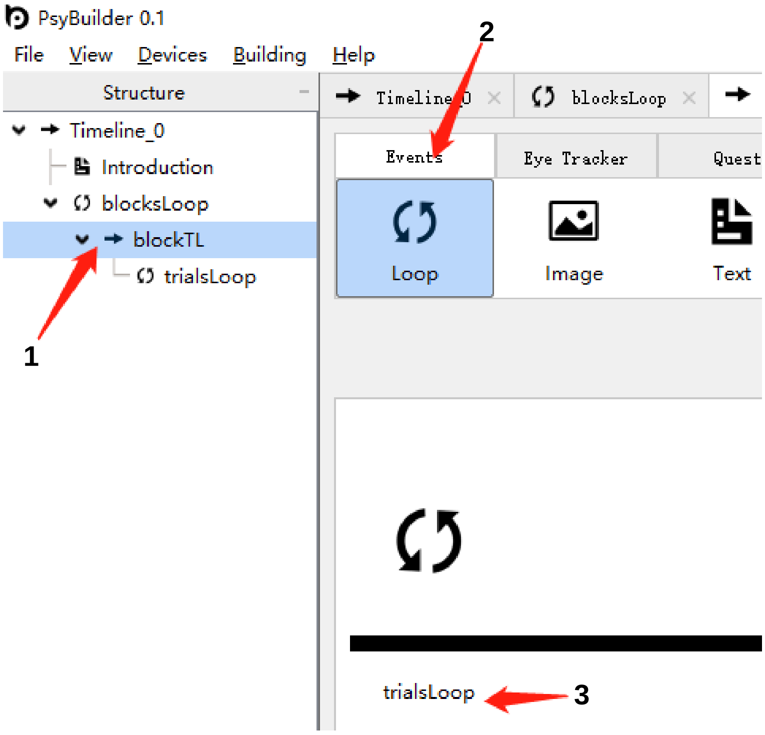
3. Click the *Add-one-row* button to create 2 blocks (or simply change the *Repetitions* column in the first row from 1 to 2 since the two blocks are exactly the same in the current task).

4. In the *Timeline* column, create a new timeline for each block with the name “blockTL”. Notice that the “blockTL” timeline also appears in the *Structure* window on the left.

5. Click the *Setting* icon (top left).

6. Select “Sequential” (running each row one by one) in the drop-down box from *Order* to set the looping mode.

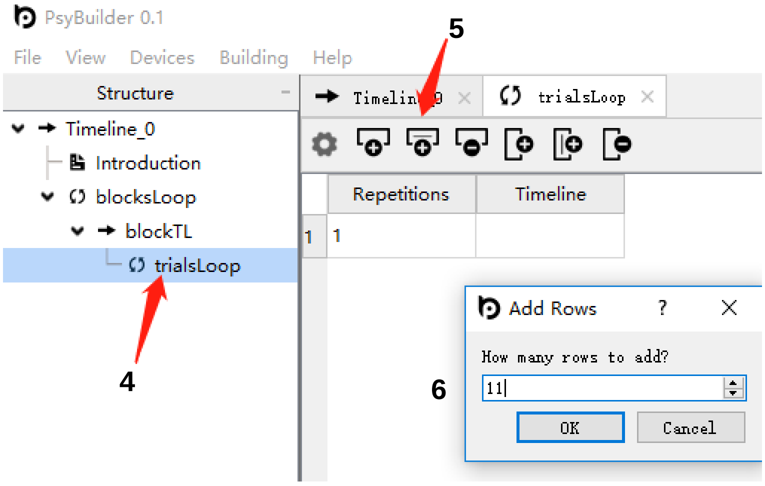
Step 4: Set up Block Timeline



1. In the *Structure* window, double-click “blockTL” to open its timeline.

2. In the *Events* tab under *Timeline*, drag the *Loop* icon to the “blockTL” timeline below.

3. Double-click the name of the loop and change it to “trialsLoop”.



	Repetitions	Timeline
1	1	

How many rows to add?

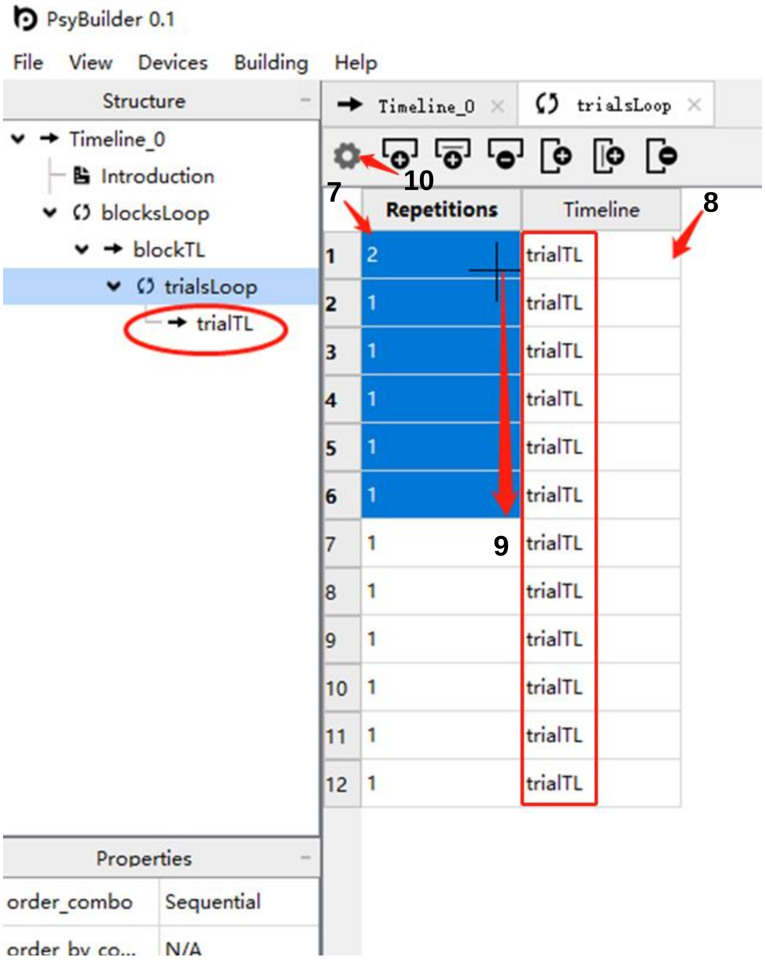
11

OK　Cancel

4. In the *Structure* window, double-click “trialsLoop” to open it and define the looping for the trials.

5. In the *Timeline* window, double-click the *Add-multiple-row* icon.

6. In the pop-up box *Add Rows*, fill in 11 and then click *OK* to add 11 rows.



	Repetitions	Timeline
1	2	trialTL
2	1	trialTL
3	1	trialTL
4	1	trialTL
5	1	trialTL
6	1	trialTL
7	1	trialTL
8	1	trialTL
9	1	trialTL
10	1	trialTL
11	1	trialTL
12	1	trialTL

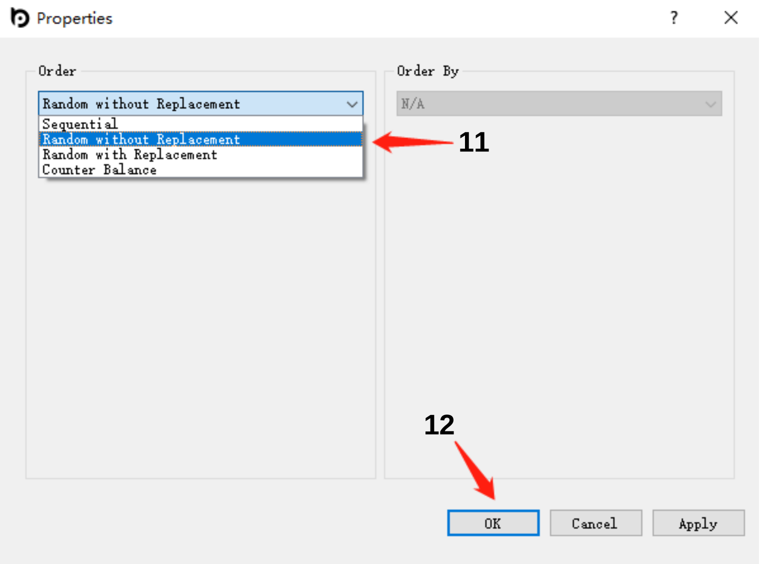
order_combo	Sequential
order by co...	N/A

7. In the first row, change the value for the *Repetitions* column from 1 to 2; do the same for rows 2 to 6 (either type in separately, or drag down the value 2 from the first row to the end of the sixth row while holding down the “Alt”/“Option” key.

8. In the first row, fill in the value for the *Timeline* column as “trialTL” and press the “Enter” key to create a new timeline for the trial. Notice that in the *Structure* window (left) an icon named “trialTL” also appears there.

9. Do the last step for the remaining trials (either type in separately or use the drag down method mentioned above).

10. Click the *Setting* button (top left) to open the *Properties* box for looping as follows.



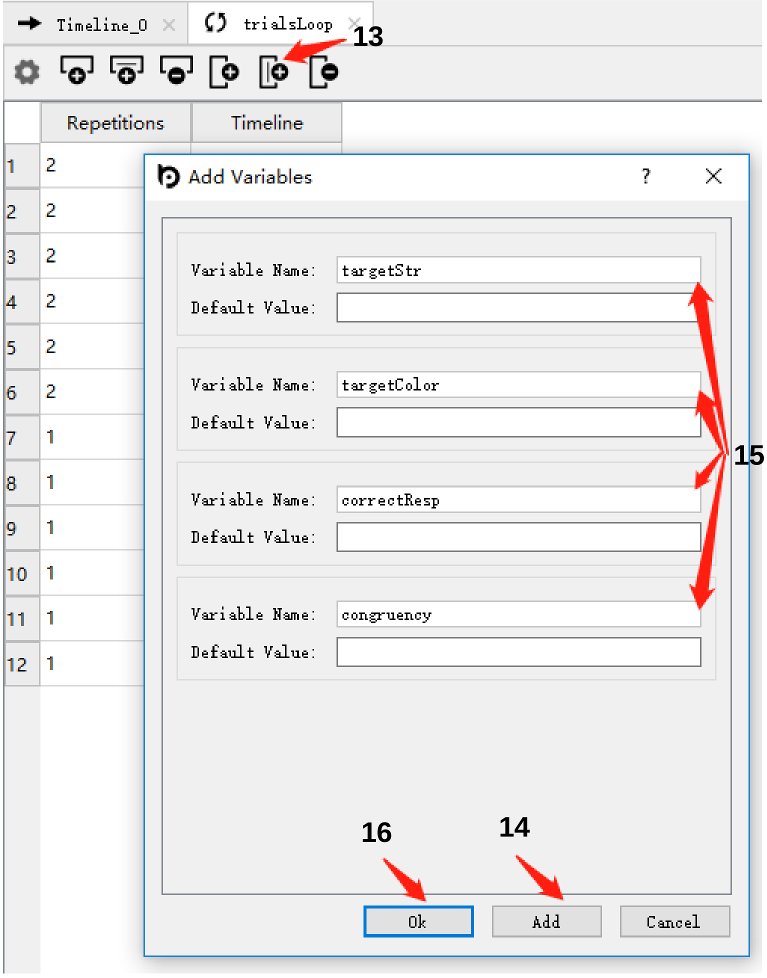
Order: Random without Replacement

- Sequential
- Random without Replacement
- Random with Replacement
- Counter Balance

Order By: N/A

11. In the *Properties* box, select "Random without Replacement" in the drop-down menu for *Order* to randomize the presentation of the trials within "trialsLoop".

12. Click *OK* to confirm the choice.



	Repetitions	Timeline
1	2	
2	2	
3	2	
4	2	
5	2	
6	2	
7	1	
8	1	
9	1	
10	1	
11	1	
12	1	

Variable Name: targetStr
Default Value:

Variable Name: targetColor
Default Value:

Variable Name: correctResp
Default Value:

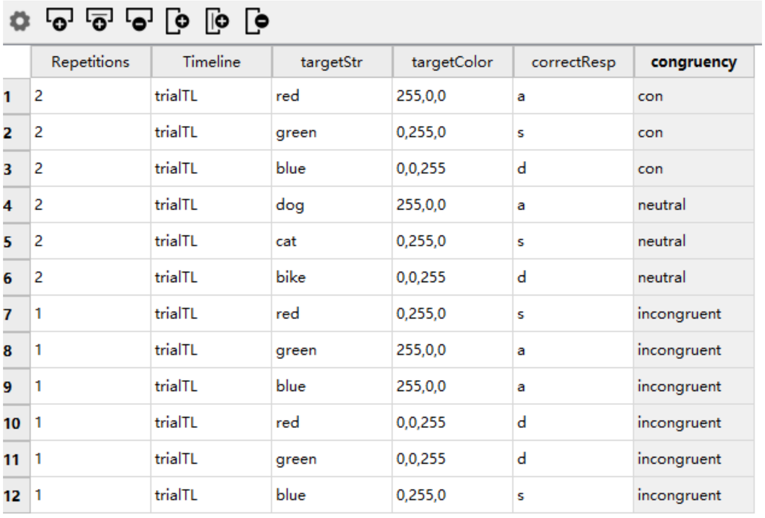
Variable Name: congruency
Default Value:

13. In the *Timeline* menu bar, click the *Add-multiple-variable* icon.

14. Click the *Add* button in the pop-up box *Add Variables* to add a new variable; in total, click four times to add four variables.

15. In *Variable Name*, for each of the four variables, type in "targetStr", "targetColor", "correctResp", and "congruency", which are used to specify the text, color, correct response key, and congruency condition for the trials, respectively.

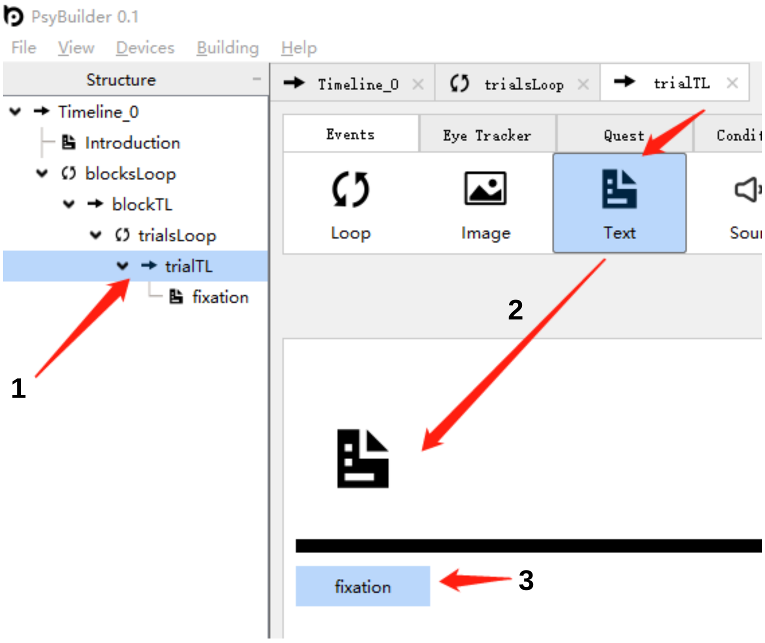
16. Click *OK* to confirm the operation.



	Repetitions	Timeline	targetStr	targetColor	correctResp	congruency
1	2	trialTL	red	255,0,0	a	con
2	2	trialTL	green	0,255,0	s	con
3	2	trialTL	blue	0,0,255	d	con
4	2	trialTL	dog	255,0,0	a	neutral
5	2	trialTL	cat	0,255,0	s	neutral
6	2	trialTL	bike	0,0,255	d	neutral
7	1	trialTL	red	0,255,0	s	incongruent
8	1	trialTL	green	255,0,0	a	incongruent
9	1	trialTL	blue	255,0,0	a	incongruent
10	1	trialTL	red	0,0,255	d	incongruent
11	1	trialTL	green	0,0,255	d	incongruent
12	1	trialTL	blue	0,255,0	s	incongruent

17. Fill in the details for the four new variables: fill in the text under "targetStr", fill in the corresponding ink color under "targetColor", fill in the correct button under "correctResp", and finally fill in the congruency condition in "congruency" (use "con", "neutral", and "incongruent" for the congruent, neutral, and incongruent conditions), as detailed in the left picture.

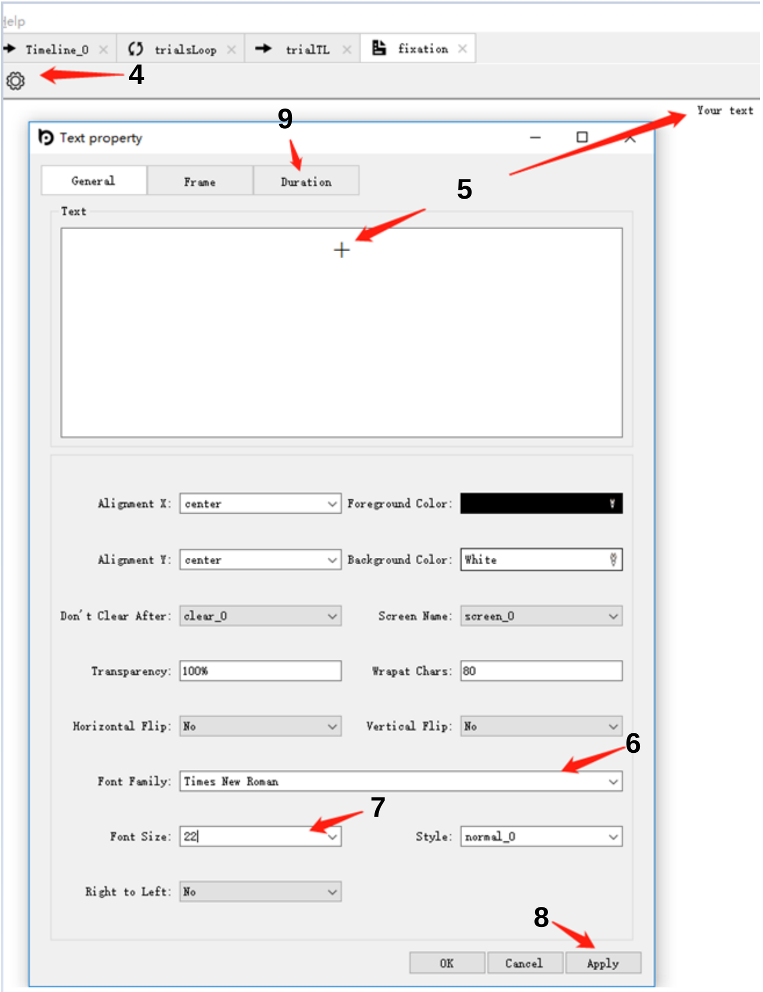
Step 5: Set up Events in Each Trial



1. In the *Structure* window, double-click the "trialTL" icon to open it in the *Timeline* window.

2. In the *Events* tab, select and drag the *Text* icon to the timeline space; double-click the name "Text_1" to change it to "fixation" (alternatively, select the name and press the "F2" key to change the name). Note: All event names can only start with a letter and consist of letters, underscores, and numbers.

3. Double-click the *Text* icon in the timeline to open it for editing as follows.



Text: +

Alignment X: center
Foreground Color:
Alignment Y: center
Background Color: White
Don't Clear After: clear_0
Screen Name: screen_0
Transparency: 100%
Wrapat Chars: 80
Horizontal Flip: No
Vertical Flip: No
Font Family: Times New Roman
Font Size: 22
Style: normal_0
Right to Left: No

4. Double-click the *Setting* button (top left).

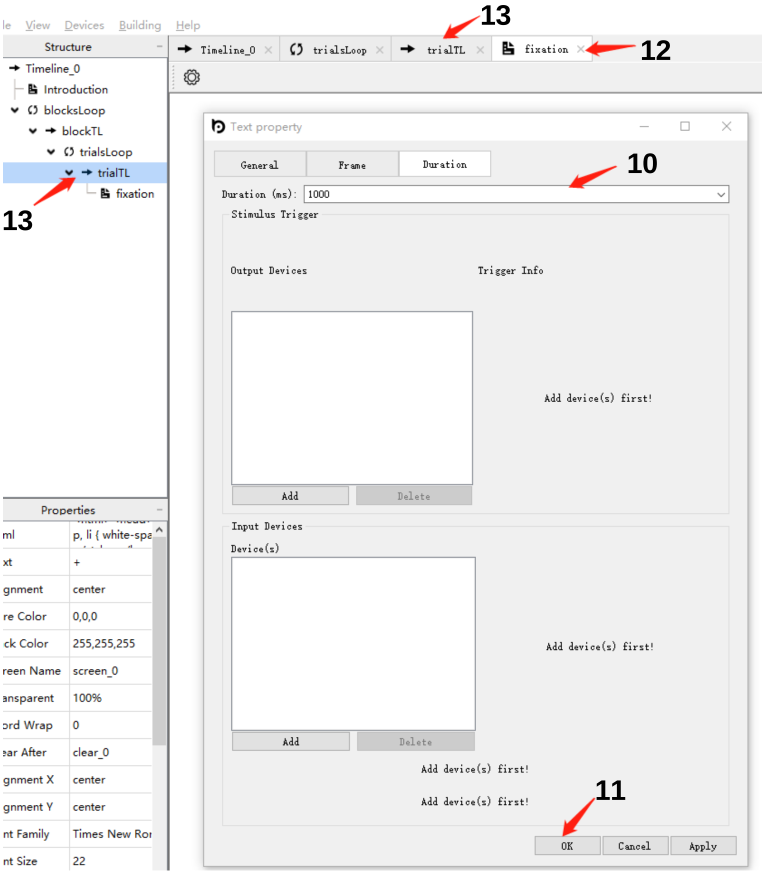
5. In the open box *Text*, modify the text content to "+" (alternatively, edit the text content directly on the *Setting* page).

6. In the *Font Family* drop-down menu, select “Times New Roman” as the text font. Note: At present, the fonts in PsyBuilder do not completely match the fonts in Psychtoolbox; please confirm whether Psychtoolbox actually supports the font selected.

7. In the *Font Size* drop-down menu, select “22” as the text size.

8. Click *Apply* to confirm.

9. Click the *Duration* tab to edit timing information as follows.



Duration (ms): 1000

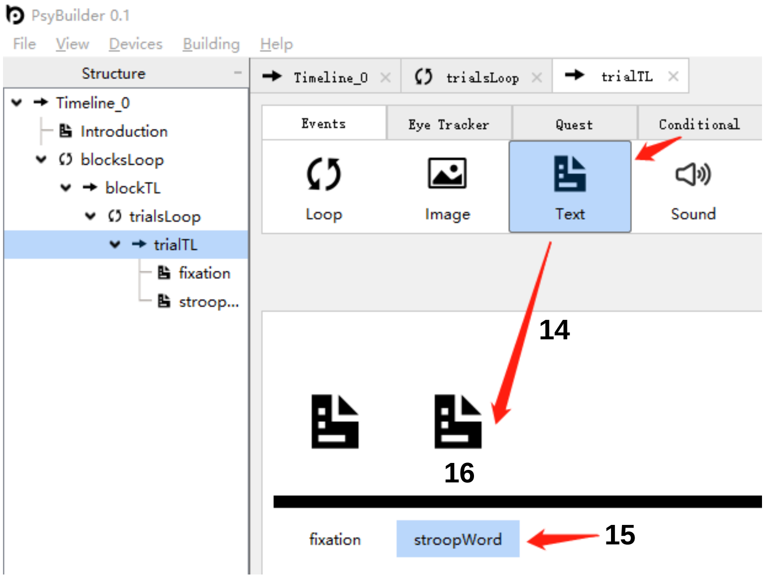
ml	p, li { white-spa
xt	+
gnment	center
re Color	0,0,0
ck Color	255,255,255
reen Name	screen_0
ansparent	100%
ord Wrap	0
ear After	clear_0
gnment X	center
gnment Y	center
nt Family	Times New Ror
nt Size	22

10. In the *Duration (ms)* box, selection “1000” as the duration for the event.

11. Click *OK* to confirm.

12. Click the “x” icon in the “fixation” tab to close it.

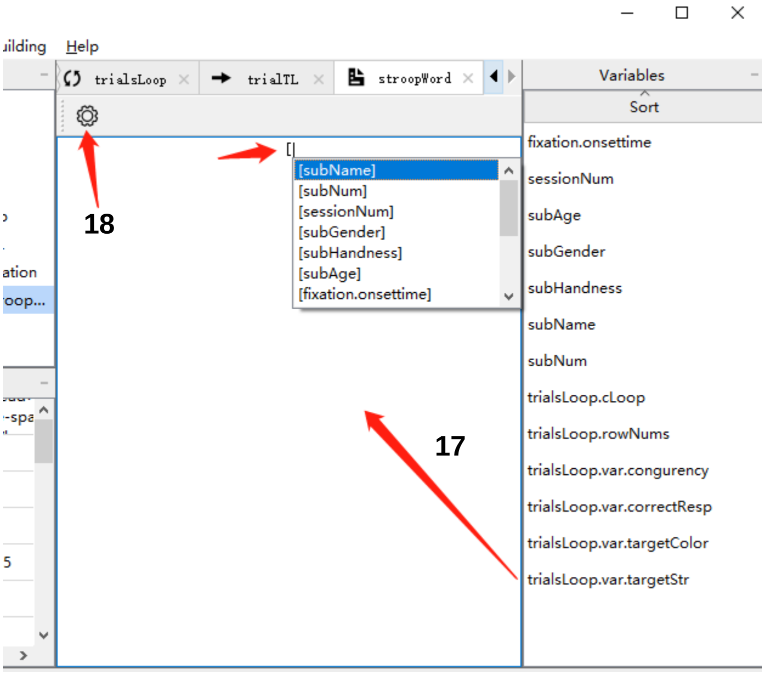
13. In the *Structure* window, double-click the “TrialTL” icon to switch back to the “trialTL” timeline interface (alternatively, click the “trialTL” tab in the *Timeline* window).



14. In the *Events* tab, select and drag the *Text* icon to the timeline below.

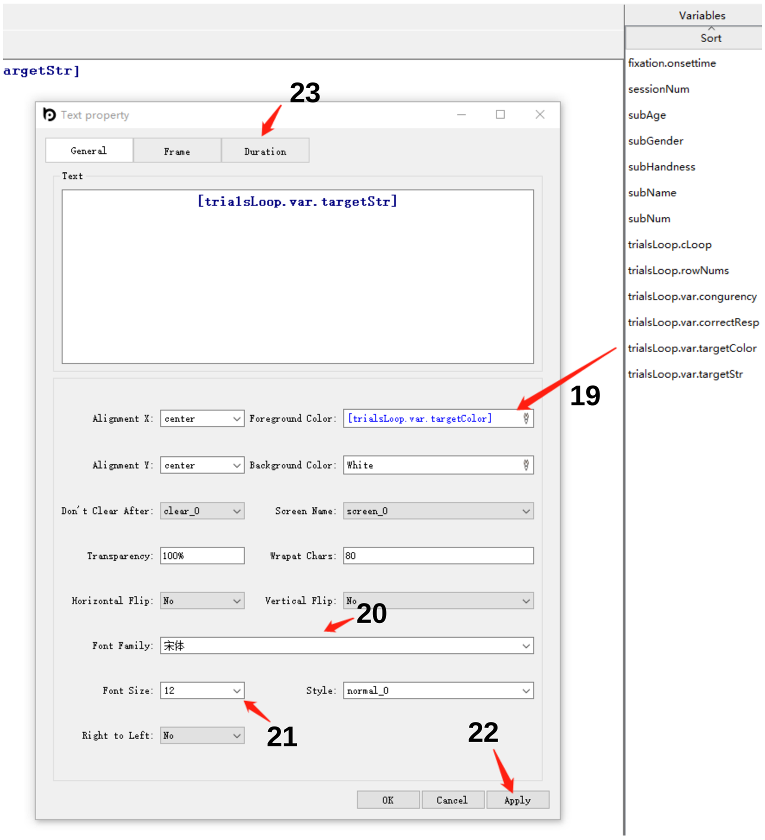
15. Double-click the name "Text_2" to change it to “stroopWord” (alternatively, select the name and press the “F2” key to change the name). Note: All event names can only start with a letter and consist of letter underscores, and numbers.

16. Double-click the “stroopWord” icon in the timeline to open it for editing as follow.



17. Enter "[" in the input interface, then select "trialsLoop.var.targetStr" from the drop-down variables using the "Up" and "Down" arrow keys, and press "Enter" to confirm (alternatively, drag the "trialsLoop.var.targetStr" variable from the *Variables* window to the input interface). Note: The variable here corresponds to the "targetStr" variable in the "trialsLoop", which defines the looping in the trials (i.e., the text of the word here). In PsyBuilder, variables are enclosed within a pair of "[ ]" symbols.

18. Click the *Setting* button (top left) to define the properties as follows.



Text: [trialsLoop.var.targetStr]

Alignment X: center
Foreground Color: [trialsLoop.var.targetColor]
Alignment Y: center
Background Color: White
Don't Clear After: clear_0
Screen Name: screen_0
Transparency: 100%
Wrapat Chars: 80
Horizontal Flip: No
Vertical Flip: No
Font Family: 宋体
Font Size: 12
Style: normal_0
Right to Left: No

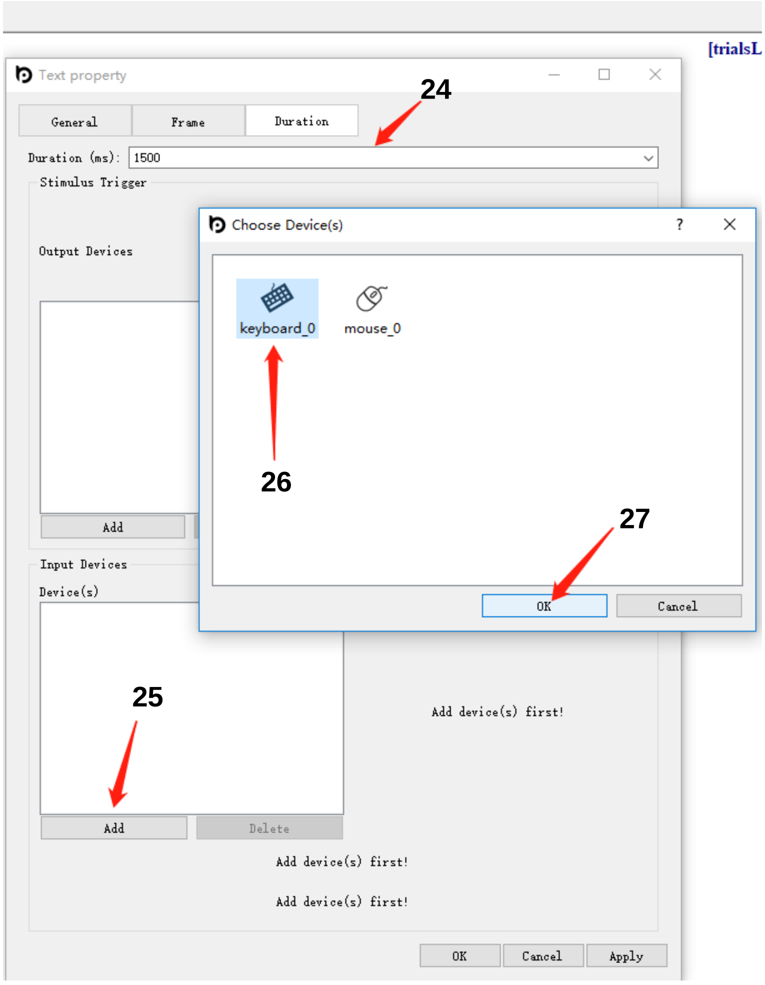
19. Drag and drop the "trialsLoop.var.targetColor" variable from the *Variables* window (right) to the *Foreground Color* box, which defines the ink color of the word (alternatively, type in the "[" symbol and then select the variable from the dropdown menu, or type in the variable directly).

20. In *Font Family*, select "Times New Roman" as the text font.

21. In *Font Size*, select "12" as the font size.

22. Click *Apply* to confirm.

23. Click the *Duration* tab to set timing



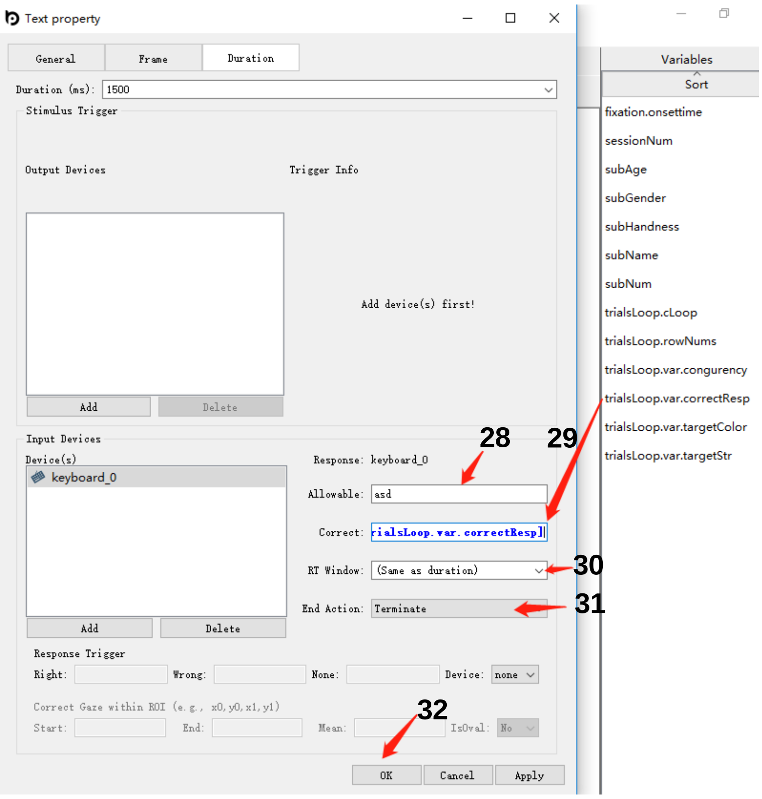
Duration (ms): 1500

24. In *Duration (ms)*, set the presentation duration to “1500” (ms).

25. Click *Add* in the *Input Devices* box to set the response collection device.

26. Select “keyboard_0”.

27. Click *OK* to confirm.



Duration (ms): 1500

Response: keyboard_0

Allowable: asd

Correct: rialsLoop.var.correctResp]

RT Window: (Same as duration)

End Action: Terminate

Device: none

IsOval: No

fixation.onsettime
sessionNum
subAge
subGender
subHandness
subName
subNum
trialsLoop.cLoop
trialsLoop.rowNums
trialsLoop.var.congurency
trialsLoop.var.correctResp
trialsLoop.var.targetColor
trialsLoop.var.targetStr

28. For the selected “keyboard_0” input device, fill in the box *Allowable* with “asd”, which specifies the keys to be recognized (that is, “a”, “s”, and “d”).

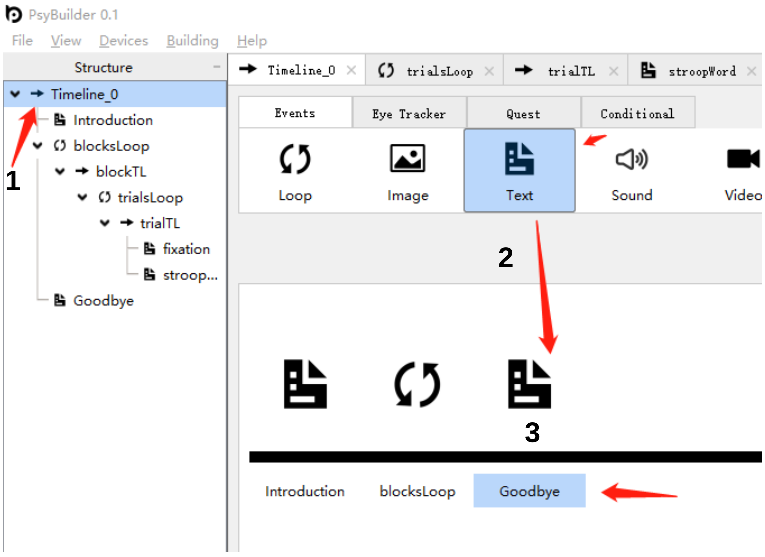
29. From the *Variables* window, drag the variable “trialsLoop.var.correctResp” to the box *Correct*, which defines the correct response key for the current event.

30. In the box *RT Window*, select “(Same as duration)”, which means that only the response during the current event will be recognized.

31. In the box *End Action*, select “Terminate”, which terminates the presentation of the current event when a key press is recognized.

32. Press *OK* to confirm.

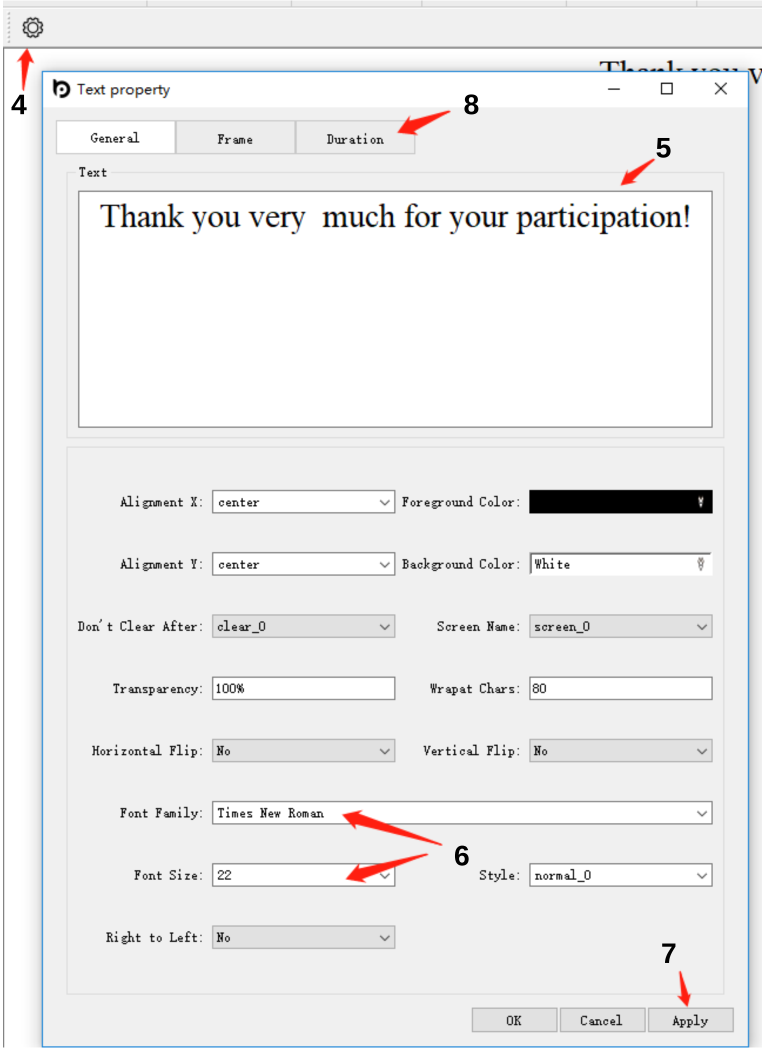
Step 6: Set up Goodbye Display



1. In the *Structure* window (left), double-click the "Timeline_0" icon to open it in the *Timeline* window for editing.

2. In the *Events* tab, drag the *Text* icon to "Timeline_0" and place it behind the "blocksLoop" event.

3. Rename it to "Goodbye", and double-click the icon on the timeline to edit its contents as follows.



Thank you very much for your participation!

Alignment X: center
Foreground Color:
Alignment Y: center
Background Color: White
Don't Clear After: clear_0
Screen Name: screen_0
Transparency: 100%
Wrapat Chars: 80
Horizontal Flip: No
Vertical Flip: No
Font Family: Times New Roman
Font Size: 22
Style: normal_0
Right to Left: No

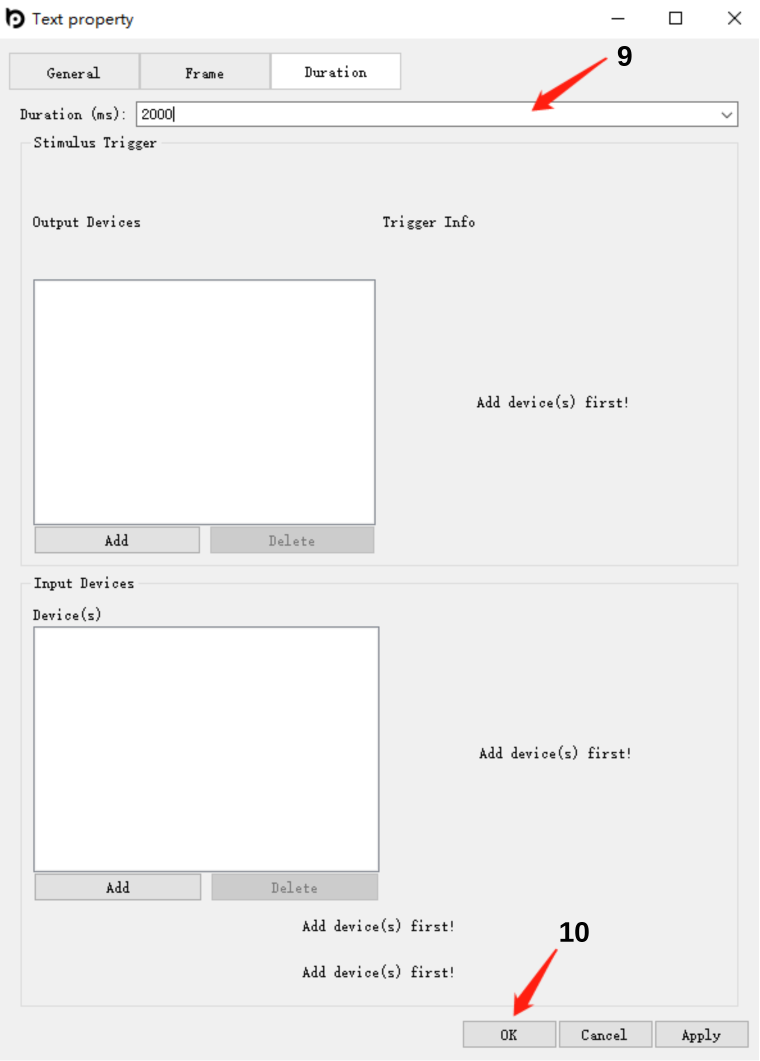
4. Click the *Setting* button (top left).

5. In the *General* tab, modify the content in the *Text* box to: "Thank you very much for your participation!".

6. Set the font to "Times New Roman" and the font size to "22".

7. Click *Apply* to confirm the operation.

8. Click the *Duration* tab to set the timing properties as follows.

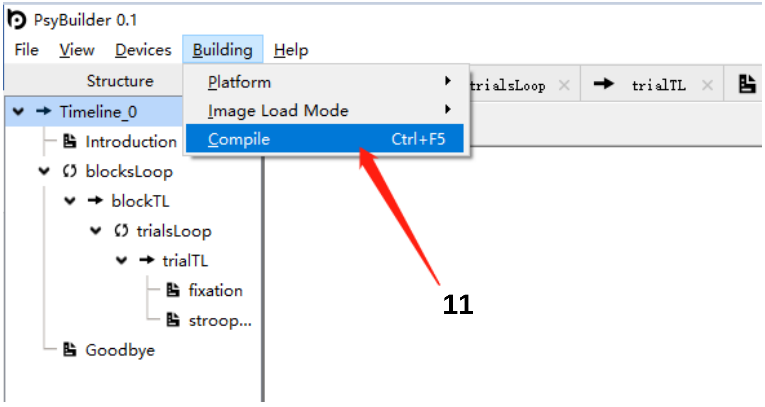


Duration (ms): 2000

9. In the *Duration (ms)* box, set the presentation duration to “2000” (in ms).

10. Click *OK* to confirm the operation.

Step 7: Save the File and Convert it to an M File



11. In the menu bar, click *Building*; then click *Compile* to compile the current project into a MATLAB m file.

Now your m file is ready for running or further editing!

This concludes the tutorial.

References

Brainard, D. H. (1997). The Psychophysics Toolbox. *Spat Vis, 10* (4), 433-436.

Kleiner, M., Brainard, D., & Pelli, D. (2007). What's new in Psychtoolbox-3? *Perception*, 36 ECVP Abstract Supplement.

Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics: transforming numbers into movies. *Spat Vis, 10*(4), 437-442.

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. *Journal of Experimental Psychology, 18*(6), 643–662.

Watson, A. B., & Pelli, D. G. (1983). QUEST: a Bayesian adaptive psychometric method. *Percept Psychophys, 33*(2), 113-120.