**Readme for the Cross-Cultural Numeric Identification Task**

Python Code by Paul M. Garrett. Updated 10/08/2018

**Table of Contents**

[Python Installation 2](#_Toc521671815)

[Python Package Installation 2](#_Toc521671816)

[Running the Experiment - Within-Subjects Design 3](#_Toc521671817)

[Running the Experiment - Between-Subjects Design 4](#_Toc521671818)

[Experimental Design 5](#_Toc521671819)

[Experimental Procedure 5](#_Toc521671820)

[University of Newcastle Testing Facilities 6](#_Toc521671821)

[Output File – Headings, Description and Values 8](#_Toc521671822)

[Sizes and Dimensions (NEWCASTLE) 9](#_Toc521671823)

## Python Installation

Experiment runs in Python 2.7.12. 32bit for Windows, please download by clicking this link:

<https://www.python.org/ftp/python/2.7.15/python-2.7.15.msi>

Note: Use 32bit version as a required package *pygame* does not have 64bit support.

When installing, ensure the install allows for ‘pip’ and ‘Add python .exe to Path’ (see Figure 1).

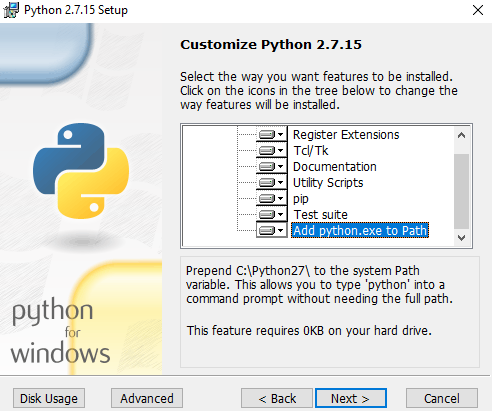


Figure 1. Python 2.7 installation. Ensure pip and add python.exe to Path are selected.

## Python Package Installation

The easiest way to install packages in python is through the pip install module. Open Command Prompt (cmd) in Windows and enter the following lines in order:

python -m pip install --upgrade pip

pip install pygame numpy pillow matplotlib scipy

These two lines should update the pip library to the latest version and install the required packages. Pygame is the backbone package running the experiment; numpy, matplotlib and scipy handle data and plotting; and pillow (aka PIL) is an image processing library.

## Running the Experiment - Within-Subjects Design

Running a within-subjects experiment has been made easy, with the experiment now containing a built-in subject allocation system. The below text will take you through how to use this feature.

1. In the Current Scripts directory, open the RUN.py file in the python IDLE.
2. Once open, press F5 to run, *OR* in the dropdown menu, select Run>>Run Module.
3. A GUI text box will appear, enter a user identifier.
   1. A Within-Subjects User Identifier is made of 3 Numbers and 1 Letter. E.g., 001A
   2. The number is the participant number, and the letter is the session number, *i.e.*, A=1, B=2, C=3, D=4.
   3. Press Enter and the experiment will begin. All instructions are presented within the experiment, however are written in English. If this is an issue, we can rewrite the instructions in Chinese or remove them to allow a verbal walk through of the task.
   4. At the end of each block, and the end of each experimental session, a .csv data file is saved off in the ‘Data’ directory. This will contain a session Language Type prefix, the session number, session letter and date stamp  
      e.g., ENG001A\_ 2017\_08\_28\_15.56.csv.

For the Within-Subject task, a Latin Square has been constructed in the code that allocates subjects to their uniquely ordered task-set e.g., session A English, session B Chinese, session C Thai and session D Dots. The Latin Square is randomly indexed based upon the subject Number (see Figure 2).

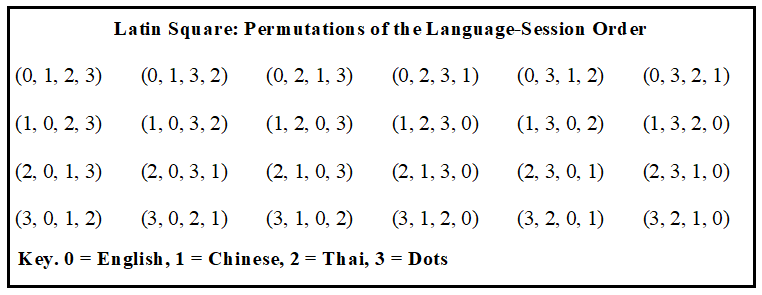


Figure 2. All permutations of task order presented as a Latin Square. For each subject, one permutation is selected using the subject number to select the permutation from a fixed-order randomly generated index e.g., [24, 10, 20, 1, 18…]. For example, using the example random index, Subject 001A would use the 24th Latin Square permutation (3,2,1,0) and in the first session, they would complete the Dots (3) experiment.

## Running the Experiment - Between-Subjects Design

If you wish to run the experiment where a subject only completes a single session, and you wish to select which language type the subject will respond to, you can input an alternative User Identifier.

1. In the Current Scripts directory, open the RUN.py file in the python IDLE.
2. Once open, press F5 to run, *OR* in the dropdown menu, select Run>>Run Module.
3. A GUI text box will appear, enter a user identifier.
   1. A between-subjects user identifier is comprised of seven letters and begins with a Language Prefix, specifying which language type will be displayed to the participant. Prefix include ENG, CHN, THI and DOT.

For example, the User Identifier ‘ENG001A’ would present English stimuli for subject number 1, session A (note, the session letter is still required). Similarly, CHN001A would be Chinese stimuli, THI001A would be Thai and DOT001A would be dots.

## Experimental Design

During each session, a participant will complete one practice block of 135 trials (with feedback), followed by 13 experimental blocks of 90 trials (without feedback). A staircase procedure will occur during the practice block. During the staircase, stimuli are presented within an independently sampled, identically distributed noisy field (Mu=0, Sigma=.25; Mu of 0 indicates a grey RBG value of 128). The stimuli begin at RGB 153, and step by 1 RGB value. If the previous two trials were correct, the staircase steps down by 1 RGB value, and if the previous trial was incorrect, the staircase steps up by 1 RGB value. This modified 2up:1down procedure allows for rapid staircasing down towards a critical contrast value. At the end of the practice block (135 trials) the mean contrast level for the last 30 trials is calculated.

During the experiment, five contrast levels are presented to each subject. One level is selected from the mean of the last 30 contrast presentations during the practice block. Three levels are selected from the three RGB steps below the mean contrast value (harder to view) and one level is selected from 1 RGB step above the mean contrast value. We term this the 3-x-1 design. The design is bias towards harder stimuli as participants tend to improve in their performance over the course of the experiment.

At the end of each block, participants are required to take a 30s break and may view their average accuracy across the task (plotted as a figure). This is to engage the subject in the task and keep them motivated.

## Experimental Procedure

Participants will be seated in a fully illuminated room, 60cm from the monitor. Experimental instructions will be presented on the monitor (in English). Each participant will be required to make accurate responses towards each of the 9 stimulus presentations and their position on the stimulus-wheel (incorrect responses are repeated) before commencing the practice round. Responses will be made using a standard Dell mouse set to a sensitivity of 10 on a Windows 7 (or Windows 10 if you do not have Windows 7) operating system. A response is generated when the mouse pointer passes over the inner-ring of the stimulus-wheel. Practice round and experimental blocks proceed as described in the Experimental Design --- that is, everything is done for you by the code.

## University of Newcastle Testing Facilities

**Monitor.** 58.42cm LED display (60Hz) with a16:9 aspect ratio. 28.64cm (height) x 50.92cm (width). Display resolution 1920 x 1080 (see Figure 3 for example).

**Wheel Size.** Outer wheel diameter 27.5cm (visual angle 25.81 degrees), inner wheel diameter 21.1cm (19.95 degrees visual angle; see Figure 5)

**Stimulus Size.** Stimuli presented in noisy box, 5.8cm x 5.8cm (5.53 degrees visual angle)

**Mask Size.** 12.4cm x 12.4cm (11.8 degrees visual angle)

**Mouse.** Dell 9RRC7 black optical USB wired mouse (see Figure 3). Mouse sensitivity (speed) set to Windows Default 10. Mouse sensitivity can be modified through the ‘Mouse Properties’, search ‘mouse speed’ in the windows search bar. Windows provides a slide-bar, with the central tick equal to 10. This value can be checked in the Registry Editor (search regedit, and look under HKEY\_CURRENT\_USER, Control Panel, Mouse, Mouse Sensitivity; see Figure 4).



Figure 3. Image of Monitor, mouse and testing station at Newcastle University.

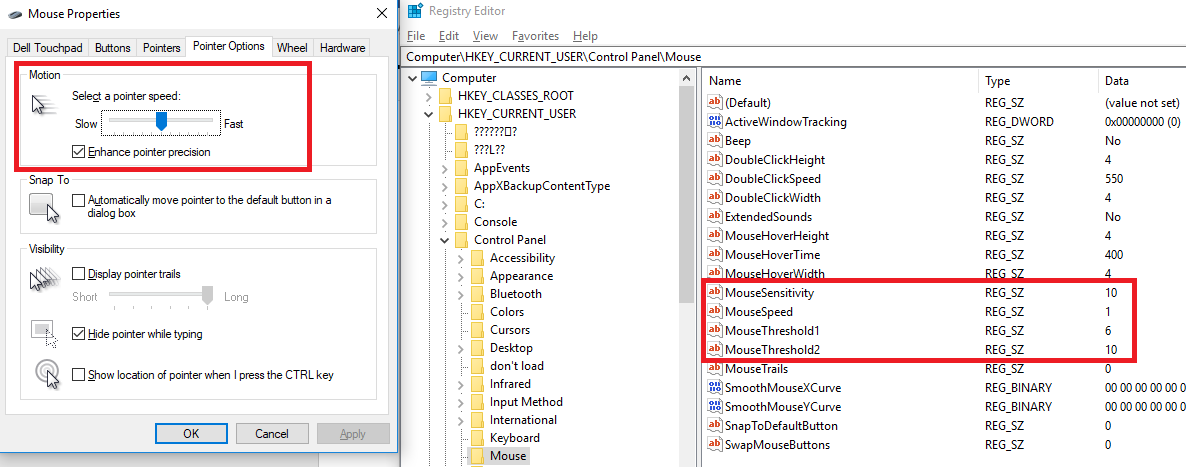


Figure 4. Left: Slider to change mouse speed, middle tick is equal to mouse speed 10 (used in the current experiment). Right: Registry editor, where the mouse speed may be assessed. Also note the mouse speed = 1, Mouse Threshold1 = 6 and Mouse Threshold2 = 10.

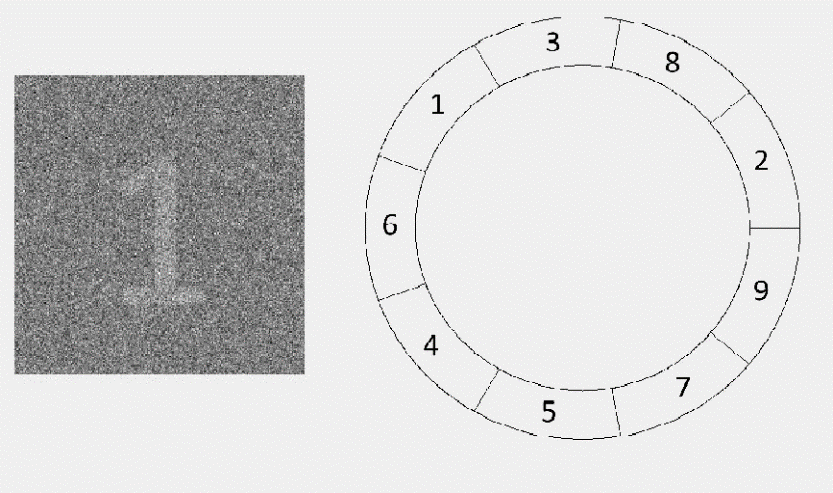
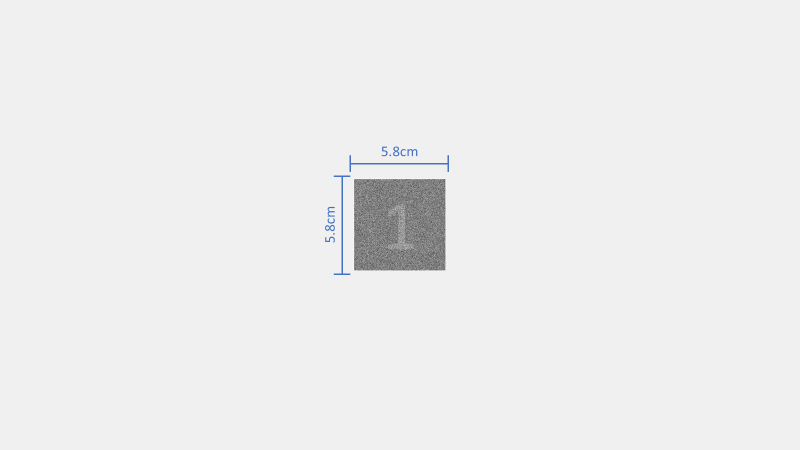


Figure 5. Example of Stimulus in noise, and example of Stimulus wheel (not to scale). Stimulus size is 5.8cm2. The outer wheel has a 27.5cm diameter, and the inner wheel a 21.1cm diameter

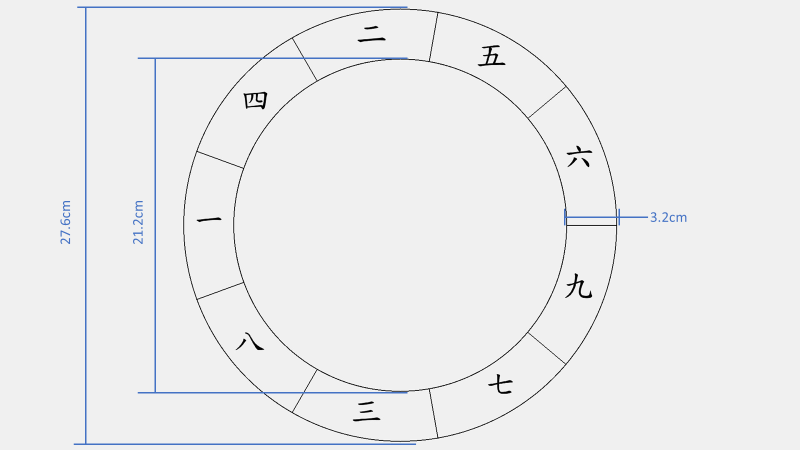
## Output File – Headings, Description and Values

|  |  |  |
| --- | --- | --- |
| **Heading** | **Description** | **Values** |
| **NativeLangNumber** | Identifies if subject was from the English or Chinese cohort | 1 = English, 2 = Chinese Cohort |
| **LanuageNumber** | Numeric, identifies which lanuage was being responded to | 0=English, 1=Chinese, 2=Thai, 3=Dots |
| **SessionNumber** | Which session has been completed i.e., A, B, C or D | 1, 2, 3, 4 |
| **Block** | Current Block | 0--13 |
| **TrialNum** | Current Trial | 1--90 (135 for practice) |
| **ConditionNumber** | Stimulus presented | 1--9 |
| **Response** | Response Made | 1--9 |
| **Accuracy** | Accuracy, Correct (1) or Incorrect (0) | 0, 1 |
| **RT** | Response Time | RT in ms, 0-8000 |
| **MovementOnsetTime** | Time before a response movement was generated | RT in ms |
| **TimeInMotion** | Time for how long the mouse was in motion | RT in ms |
| **SignalContrast** | RGB value of the stimulus signal | RBG value |
| **SignalContrastNumber** | Contrast level at which stimulus was presented | 1--5 (experimental blocks) |
| **StimDisplayTime** | Max Stimulus Display Time | 500ms |
| **MaskTime** | Mask Display Time | 200ms |
| **MaxResponseTime** | Max Time For A Response Generation | 8000ms |
| **MousePositionXY** | List of x and y coordinates, mouse at every millisecond after initial movement onset. | [ (x1,y1), (x2,y2), (x3,y3)…] |
| **MouseAngle** | Trajectory Angle at which the mouse passed over response ring (form the center) | 0-360 degrees |
| **WheelOrder** | Randomised order by which the subject viewed the stimuli, starting from 4 Oclock position and moving clockwise | 1--9 |
| **NoiseMuSigma** | Noisy Stimulus Mu and Sigma (for future reference) | (0, 0.25) |
| **ParticipantID** | Current Subject ID e.g., ENG001A, CHN001A, THAI001B, DOT005D, etc | -- |
| **LanuageTested** | Prefix of the launage that was tested | ENG/CHN/DOT/THI |
| **SessionTested** | Letter of the session currently being tested | A, B, C, D |
| **NativeLanuage** | Native Lanuage as a string | English or Chinese |
| Date | Test Date | Fri Jul 13 11:09:52 2018 |
| ResponseDevice | Response device | Mouse/Joystick |

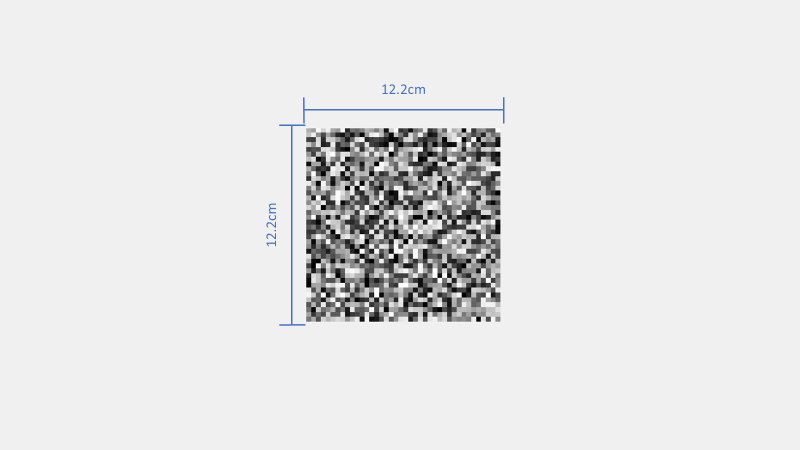
## Sizes and Dimensions (NEWCASTLE)



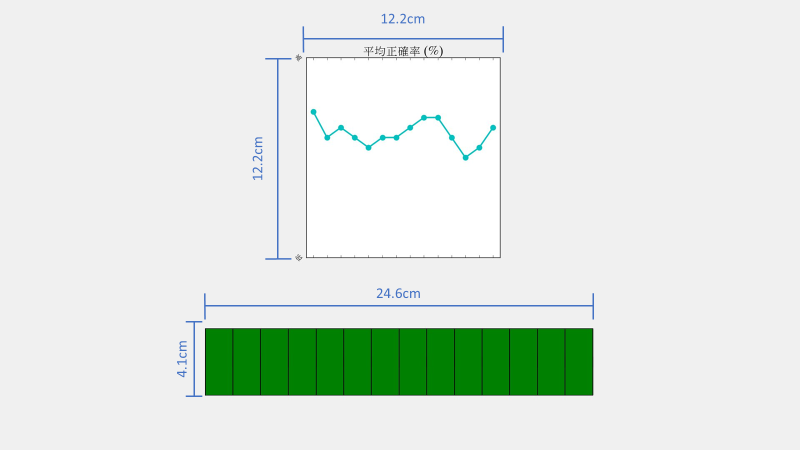
Change in Variables.py from the default value PrimaryStimSize = 100



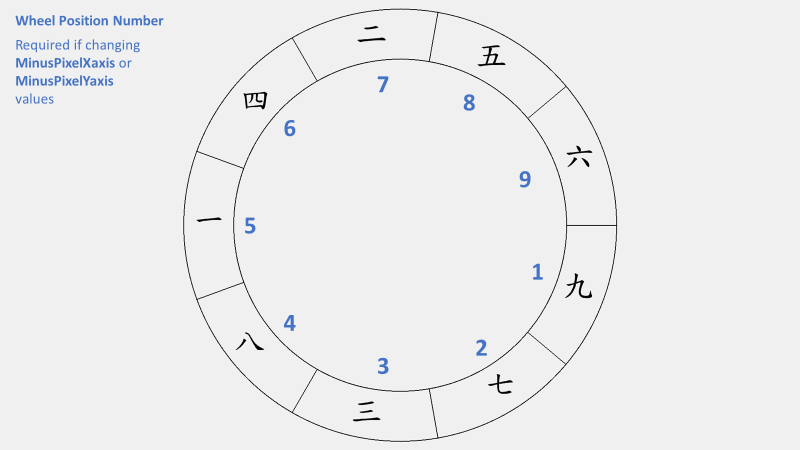
Change in Variables.py from the default value OuterRadius = 520; InnerRadius = 400



Change in Variables.py from the default value MaskSizeModifier = 6



Change in Variables.py from the default value FeedbackAccuracyPlotSize = 6; BlockProgressBarDimensions = (1200, 200) and/or BlockProgressBarSize = 3; ProgressBarTickLength = 100.



This shows the position of each stimulus as indexed in a list i.e., [1,2,3,4,5,6,7,8,9]. The stimulus positions are roughly calculated for you but they need some fine adjustments. This is where MinusPixelXaxis and MinusPixelYaxis come in. They let you adjust the positions of the stimuli using X,Y coordinates

To alter the position of the stimuli in Position 2, given that:

MinusPixelXaxis = [55,40,50,60,50,45,40,65,55]

MinusPixelYaxis = [55,60,55,60,60,60,55,60,60]

You could alter X axis from 40 to 45, and the Y axis from 60 to 65. This would move the stimuli left and up. Do this once you have the Wheel Size figured out so that all the stimuli appear in the middle of their respective wheel segments.

To view any of these screens, open the RUN.py file and comment out the RUN() command. Then uncomment any of the following to view a single screen:

* TestWheelPosition() and associated code
* TestStimSize() (and associated code)
* TestMaskSize() (and associated code)
* TestFeedbackScreen() (and associated code)

And when you are ready to test the experiment, comment out any and all of these functions, and uncomment the RUN() command. Then when you run the script, the experiment will begin.

If you want to send screenshots of any issues, uncomment the ScreenshotAllStim() function (and associated code) and run this in isolation. This will save copies of all the primary display screens into the Figures folder and you can tell me what the issue is with the screenshot to aid you. This is also good for providing examples when making presentations and publication material.