Selective Letter Attention 2D Application

Andreas Wulff-Abramsson   
*Dept. of Nutrition, Exercise and Sports  
Copenhagen University*Copenhagen Denmark  
[awa@nexs.ku.dk](mailto:awa@nexs.ku.dk)

Anne Husted Hensriksen  
*Dept. of Nutrition, Exercise and Sports  
Copenhagen University*Copenhagen Denmark  
[xbn434@alumni.ku.dk](mailto:xbn434@alumni.ku.dk)  
  
Marta Katarzyna Topor  
*Dept. of Nutrition, Exercise and Sports  
Copenhagen University*Copenhagen Denmark  
[mt@nexs.ku.dk](mailto:mt@nexs.ku.dk)

Rasmus Ahmt Hansen  
*Dept. of Nutrition, Exercise and Sports  
Copenhagen University*Copenhagen Denmark  
[sfk384@alumni.ku.dk](mailto:sfk384@alumni.ku.dk)

Jacob Wienecke  
*Dept. of Nutrition, Exercise and Sports  
Copenhagen University*Copenhagen Denmark  
[wienecke@nexs.ku.dk](mailto:wienecke@nexs.ku.dk)

*Abstract*—In this paper we present the technical underpinning and setup behind the selective letter attention test. It is created with the PsychoPy Framework, it is composed of three different scenarios where the distractor is visual, audiovisual and audio only. Each of the scenarios are composed by 1000ms Fixation cross, 200ms stimuli with filler letters and distractor, 5000ms whitespace, where the subject is able to answer what letter they have seen. Lastly, 500ms is used to provide feedback. Each of the scenarios are shown 12 times. In each trial the subject have to tell the system whether or not he/she has seen “p” or “b”, while being distracted by “p”,”b” or “g”. To avoid making the task too easy three filler letters together with the target letter are randomly assigned a predetermined position.

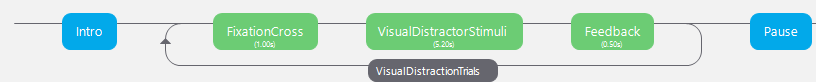
Keywords—Selective Attention, Letter Recognition, Python, 2D, PsychoPy)

# Introduction

Selective attention is the mechanism of experiencing one elements out of many possible [1], [2]. This process can be challenged by many factors [2], [3]. In the context of letter recognition it will be other letters, which could influence the selective attention processes of the subjects. In the context of our studies, we have developed this test to closely match the look and feel of the following article [4]. Just like them, we intent to get a better understand of the developing selective attention processes in children. More so are we curious about this process in the context of letter recognition as it is newly acquired skill for children in the lower grades [5]. In the following sections we will dive into the development of the program behind the Selective Letter Attention test.

# Implementation

First the program is programmed in Python 3.0 [6] using the PsychoPy Framework [7]. The Psychopy Graphical User Interface (GUI) allows us to iteratively prototype, develop and test the experiment. The development consist of five distinctive sections called Routines: Intro, FixationCross, Stimuli (visual, audiovisual or audio), Feedback and Pause (see Fig. 1).



1. Shows the cause of the first scenario presented to the subject.

## Intro

The intro is comprised of an introduction text, which explains how to behave during the experiment:

*“Velkommen til Opmærksomheds testen.*

*Du vil blive præsenteret for bogstaver*

*hold øje med b og p.*

*En af dem vil være i højre eller venstre side af skærmen.*

*Hvis du ser p klik til venstre på musen*

*Hvis du ser b klik til højre på musen*

*Klik så snart du ved hvilke bogstav der kom frem”*

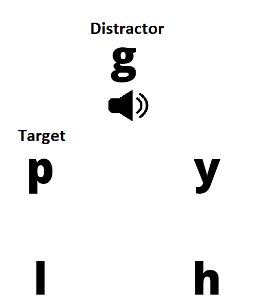
The text appears at the beginning of the experiment. It is before the subject as long as the subject desires. He/she can press the return key (Enter) to progress to the next part of the experiment

## FixationCross

In this routine a black X is shown in the middle of the screen surrounded by a white background. The System progresses by itself further to the Stimuli Stage.

## Stimuli

In the stimuli routine the stimuli is present together with either visual, audiovisual or audio only distractor (see Fig. 2).



1. Shows how the stimuli is displayed in the middle of the screen.

It is present for 200ms continued by 5000ms of blank white canvas. In all 5200ms the subject is able to press a mouse button corresponding to the target letter he/she saw. Left mouse click if it was “p” and right mouse click if it was “b”. The most interesting part of the stimuli is that it has to be shown 12 times within each of the conditions, within those 12 trials 6 different configurations have to be shown 2 times. But, the order is random and positioning of the letters within the 5 different places are random.

The five different locations are :

Distractor = (0.0 , 0.3)

Upper Left position = (-0.15 , 0.1)

Lower Left position = (-0.15 , -0.1)

Lower Right position = (0.15 , -0.1)

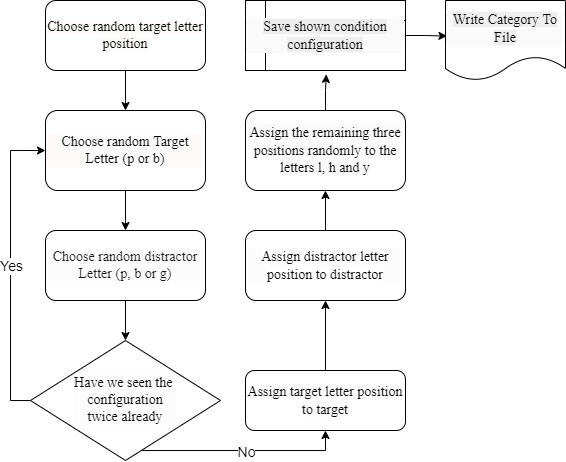
Upper Right position = (0.15 , 0.1).

The 6 different configurations can be seen in Table I.

1. Stimuli Configurations

| Configurations | | |
| --- | --- | --- |
| Target | Distractor | Category |
| p | p | Congruent |
| p | b | Incongruent |
| p | g | Neutral |
| b | b | Congruent |
| b | p | Incongruent |
| b | g | Neutral |

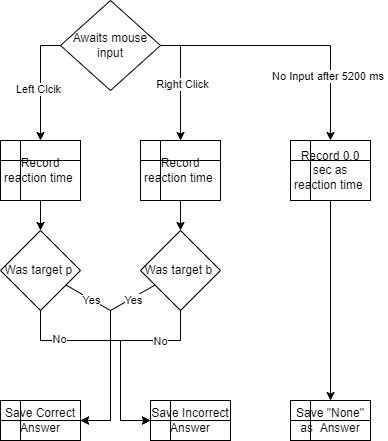
In order to accommodate the random, yet controlled nature of the stimuli the following algorithm was devised (see Fig. 3).



1. Shows the flow of how the program decides which and where stimuli shall be presented to the user

In Fig. 3 it shows that every time the stimuli is presented it ensures the configuration is saved in the program, thus after 12 trials all the wished stimuli configurations are shown.

After the stimuli has been chosen by the program it listens to the mouse for user input in order to determine how quickly the user responses to the 200ms stimuli, and whether or not it is correct. The flow of the program can be found in Fig. 4.



1. Shows the flow for how mouse input is handled when the stimuli has appeared before the subject.

From the diagram it is noteworthy to understand if the subject does not react within the time window of 5200ms the reaction will be recorded as “None” which will tell the system to prompt the user with a message expressing that the subject’s reaction shall be faster.

## Feedback

Based on the correctness of the reaction, recorded during the stimuli Routine. The feedback is present on the screen for 500ms. The following table (Table II) elucidates the different feedback permutations:

1. Feedback permutations

| Permutations | | |
| --- | --- | --- |
| Answer | Feedback |  |
| Correct | C:\psychoPy\SelectiveAttention\graphic\happyface.png |  |
| Incorrect | sadface |  |
| None | “Hurtigere” (Faster in Danish) |  |

If we look at Fig. 1 we see that FixationCross, Stimuli and Feedback are embedded in a loop, which after 12 runs continuous to Pause

## Pause

This Routine is a break between two routine loops. It occurs after the first (visual distractor) and second routine loop (audiovisual distractor), but not after the third (audio distractor). When the third has ended the whole program stops and the data is saved as a CSV file ready to be interpreted. The pause routine is a simple text stating:

*“Pause*

*tryk på en tast når du er klar til at fortsætte”*

Which means that the subject can press any button in order to continue.

# Pilot Results

To confirm that the program is reliable in providing the visual stimuli at the right time three initial pilot sessions where conducted on the involved researchers using a Lenovo X13 laptop with 32gb RAM, i7-10610U CPU 1.80GHz, 2304 Mhz, 4 Core(s), Intel UHD Graphics GPU, and ITB SSD Hard drive. And a ThinkVision P27q-20 external display 27” 2560 x 1440, 60 Hertz.

The aim were to elucidate whether or not all elements were present on the screen for the wished amount of time and whether or not all wished conditions were given. In the table below the expected time, average observed time and standard deviation is given.

1. Stimuli presentation

|  | Stimuli | | |
| --- | --- | --- | --- |
| Stimuli | Expected time in sec | Average Observed time in Sec | Standard deviation in Sec |
| Fixation Cross | 1.0 | 1.0099 | +/-0.0086 |
| Letters | 0.2 | 0.2072 | +/-0.0057 |
| Blank White | 5.0 | 5.0543 | +/-0.0134 |
| Feedback | 0.5 | 0.4566 | +/-0.0112 |

From table III it is clear that albeit the amount of time a certain element of the experiment have to be shown on the screen have been curated, then the program does not keep the time constant. Through the table III we see small discrepancies between the value we expect and the observed average. For Fixation Cross and Letters the difference is <10ms for Feedback it is <50ms, while for the Blank White screen it is <60ms. (to see the spread of the observed time please see Fig 5, Fig 6, Fig 7 and Fig 8)

1. Shows a graph displaying the time difference between when the letters appeared and when they disappeared. These differeences were recorded per trial shown the subject.
2. Shows a graph displaying the time difference between when the fixation cross appeared and when it disappeared. These differeences were recorded per trial shown the subject.
3. Shows a graph displaying the time difference between when the white screen appeared and when it disappeared. These differeences were recorded per trial shown the subject.
4. Shows a graph displaying the time difference between when the feedback appeared and when it disappeared. These differeences were recorded per trial shown the subject.

For the amount of shown condition we expect that the accumulated amount of trials for each condition over the three pilot tests would be 36 each. We can confirm through data observation that there are exactly 36 trials of each of the three conditions Congruent, Incongruent and Neutral.

# Discussion

We have visited the implementation of the Selective Letter Attention test and a small pilot test to test the reliability of the program. Through the results we observe that there are small differences between the time we have told the stimuli to be present on the screen and the observed time. This could constitute a multifactor of things.

1. The program is not efficiently enough programmed Seeing that programs is programmed in Python and sits on top of a framework, then there is a fair chance that there are many hidden computational steps that has to be computed before elements appears and disappears from the screen.
2. The hardware is not strong enough: The bottleneck could also be in the hardware itself, which is simply not strong enough to compute it at a speed, which allows the program to act as intended.
3. The screen refresh rate is not quick enough: The used screen has a refresh rate of 60Hz, which amount to 1000/60 = 16,6667ms per update or 0.016s per update. This time could explain the small discrepancies found between the expected time for the Fixation Cross and the actual time (see Fig. 6) as well as the letters (see Fig. 5). As those are within 16ms. The differences in time between expected presence of white screen and the actual time (see Fig. 7) as well as the time feedback is present (see Fig. 8) cannot merely be explained by the update rate alone. For those, there must be some computational parameters, which causes the difference between expected time presence and the average actual time presence to be > 16.66 ms.

Albeit the differences they are not experiment breaking. The most important for this experiment must be the time the letters are shown, seeing that those can be explained by the reference rate of the screen they are not deemed significant

# Conclusion

In this paper we have visited the implementation for a Selective Letter Attention test. It is programmed in Python3 using the PsychoPy Framework. It provides the researchers with three distractor scenarios, where the distractor is either visual, audio visual or audio only. Each scenario is presented 12 times in order to show 6 different types of trials two times. These trial are Congruent, Incongruent and Neutral. These are devised to understand how the distractor of different kind provokes the reaction time towards the subject pressing a mouse button corresponding to what he/she thinks he/she has experienced. The stimuli presented is comprised of a target (p or b) a distractor (p, b or g) and three filler letters (l, h and y). These where displayed for 200ms, apart from the stimulus a trial was comprised of 1000ms fixation cross, 5000ms white screen and 500ms feedback.

These expected presence times were tested to understand the reliability of the program. It appeared that fixation cross and the letter stimuli both where on the screen for the expected time +/- 10ms, which is expected with a framerate of 60 hertz. However the white screen and the feedback deviated away from their actual on screen appearance by >+/- 50ms, which is significant more and could be explained by the programming or the hardware. However, with those elements that timing discrepancy is not important, as they are not influencing the attention mechanisms of the subject.

##### Acknowledgment

We would like to thank Ruben Schachtenhaufen for his contribution with recording the sound files and as a phonetics expert consultant.

##### References

1. J. Driver, “A selective review of selective attention research from the past century,” 2001.
2. *Encyclopedia of Personality and Individual Differences*. Springer International Publishing, 2020. doi: 10.1007/978-3-319-24612-3.
3. N. Lavie, “Distracted and confused?: Selective attention under load,” *Trends in Cognitive Sciences*, vol. 9, no. 2. pp. 75–82, Feb. 2005. doi: 10.1016/j.tics.2004.12.004.
4. P. J. Matusz, R. Merkley, M. Faure, and G. Scerif, “Expert attention: Attentional allocation depends on the differential development of multisensory number representations,” *Cognition*, vol. 186, pp. 171–177, May 2019, doi: 10.1016/J.COGNITION.2019.01.013.
5. A. S. B. Malling, H. Juul, A. K. Gejl, L. Damsgaard, J. Wienecke, and A. M. V. Nielsen, “Word Reading, Letter Knowledge, and Memory Skills in Danish Children (6-Year-Olds),” *Scandinavian Journal of Educational Research*, 2021, doi: 10.1080/00313831.2021.1983646.
6. Mark. Pilgrim, *Dive into Python 3*. Apress, 2009.
7. J. Peirce *et al.*, “PsychoPy2: Experiments in behavior made easy,” *Behavior Research Methods*, vol. 51, no. 1, pp. 195–203, Feb. 2019, doi: 10.3758/s13428-018-01193-y.