# Open Source Implicit Association Test

Written in PsychoPy, but more flexible than the version bundled with it.



## License



## Version

1.2.1 (2016/04/16)

## Description & purpose

The IAT is an implicit measure of automatic behaviour or automatic biases (Greenwald, Nosek & Banaji, 1998). This implimentation of the IAT has high fidelity to the procedure described in Nosek et al. (2007: the IAT at 7), and to the standard IAT Inquisit script provided by Millisecond. See block layout below. IMHO, fidelity is higher than other freely available alternatives, such as

the FreeIAT or OpenIAT (e.g., the latter has a different block layout, and also uses a combination of free responding and accuracy feedback, where the IAT almost invariably uses one or the other but not both).

# Requirements

- PsychoPy v1.90 or above
    - NB if running the .py file, you must use the Python 2 version of PsychoPy and not the Python 3 version. Opening the .psyexp file in either version and then compiling a .py file from this should produce a .py file that will run in the current PsychoPy version.
    - A free and open source program for delivering psychology experiments written in Python. See here for documentation.
    - PsychoPy runs locally on Windows, Mac, and Linux. It's not possible to run PsychoPy scripts online.
- RStudio, a very user friendly interface for R that integrates Knitr and RMarkdown documents, which are used by the processing script to score the data.

# Usage

- You can run either the `.psyexp file` or the `.py` file inside PsychoPy. However, if using the .py file, you must use the Python 2 version of PsychoPy and not the Python 3 version. Opening the .psyexp file in either version and then compiling a .py file from this should produce a .py file that will run in the current PsychoPy version. The task was written in version 1.82.01 but has been tested to run in later versions.
- The included stimulus file employs pictures as category stimuli and words as attribute stimuli. However, this implimentation can display any combination of words and/or images for both categories and attributes. To do this, edit the highlighted rows in the excel files: if using text stimuli, put "blank.png" in the appropriate image exemplar column in the stimuli.xlsx file. If using image stimuli, put a single space character in the appropriate column, as empty cells will cause the task to crash and Psychopy will throw an undefined variable error.
- The escape key quits the task at any time.
- The order of presentation of blocks within the task (e.g., whether participants get flowers-positive/insects-negative or flowers-negative/insects-positive first) is determined by the participant code. Odd numbered participants get the former, even numbered participants get the latter. Be careful that this counterbalancing does not covary with your counterbalancing of other experimental conditions.
    - If you don't enter a participant number the task will crash and refuse to run!
- Block length is fixed by the code and the `block_layout.xlsx` file, and is independent of the number of stimuli exemplars.
- The number of exemplars can vary freely; it's currently 6 but can be more or less without this affecting the block length. However, it must be identical across categories.

- `.psydat` and `.csv` files are produced for each participant. The `.csv` file is sufficient to most analyses (e.g., calculation of D scores).
- R script in the analysis folder produces D1 scores (and D1 scores for each stimulus category parcel), as well as accuracy and latency summary data.
- All stimuli and instructions can be altered by editing the excel files. Indeed, all strings presented within the task are variables, so translating the task into other languages only requires changes to the stimuli and instructions files.
- ITI is set to 250 ms (see Nosek et al., 2007: the IAT at age 7).

# Block layout

The current version follows the block layout described in Nosek et al. (2007: the IAT at age 7). The contents of each screen follows the standard Inquisit IAT distribution very closely. However, the Inquisit IAT uses the older block layout of 20 trials in block 5.

- Block 1 (categories) 20 Trials
- Block 2 (attributes) 20 Trials
- Block 3 (both) 20 Trials
- Block 4 (both) 40 Trials
- Block 5 (categories reversed) 40 Trials
- Block 6 (both reversed) 20 Trials
- Block 7 (both reversed) 40 Trials

# Localisation and customisation

All stimuli and instructions within the task are set via the `stimuli.xlsx` and `task.xlsx` files.

PsychoPy has Unicode support, so translating the task into other languages (Spanish, Polish, Japanese, etc.) only requires changes to these excel files.

- NB a poorly documented bug is that if you zip and unzip excel files using archive utility on Mac OS X, Unicode characters are no longer correctly displayed and will throw an ASCII error in PsychoPy. Make new excel files to correct the issue.

# Output

## Data files

`.psydat`, `.csv` and `.log` files are produced for each participant. The `.csv` file alone is sufficient to most analyses (e.g., calculation of D scores). To my understanding, the format of the `.csv` output files are Tidy Data compliant (Wickham, 2014) and therefore easy to analyse (e.g., in R) with little to no processing needed.

## Data processing

The included `processing.rmd` R script produces accuracy and latency summary data and *D*1 scores for each participant (including "overall" *D*1 scores, *D*1 scores for each trial-type, and split-half overall *D*1 scores).

Very little familiarity with R/RStudio is needed to use this script.

1. Open the script in RStudio.
2. Click "Knit", this will run the script, save the processed output, and create a html file record.

### Interpretation of D1 scores

Positive *D*1 scores refer to faster responding on the block where response mappings are shared between trial types 1 & 3 and 2 & 4 (e.g., flowers-positive/insects-negative) than the block where response mappings are shared between trial types 1 & 4 and 2 & 3 (e.g., flowers-negative/insects-positive).

# Known issues

1. If participants get 100% of trials correct throughout the whole task then three incorrect response columns will not be created for that participant. This is highly unlikely, and futhermore is not a problem if your data processing workflow merges data files across participants based on column header names (e.g., with R using readr's read.csv) rather than column positions (e.g., a SPSS script using a GET command, or some other R commands which assume equivalent table shapes).
2. Block order is not recorded in the data file, but is derived from the participant code (odd number vs even number).
3. “Empty” cells in the instructions file must actually include a whitespace character or task will crash. If text stimuli are used, put “blank.png” in the image stimulus box.
4. Requires 5 exemplars per category, as the stimuli file conflates block structure with exemplars to be used. I have a method to change this, but haven't implimented it yet.
5. Stimuli presentations are based on timing rather than screen frames. This is relatively easy to alter, but requires more from the user in terms of setting the script up with their specific hardware. Marginal timing accuracy trade off.

# Timing accuracy

PsychoPy is technically capable of millosecond timing, depending on design choices by the researcher (see Garaizar & Vadillo, 2014).

The current implementation is written to be at least as accurate as other implementations of the IRAP (i.e., accurate to within a frame or c.17ms). The stimuli to be presented within a block are generated on the pre-block rule screen, and then `pop()`'d on each trial.

If you're looking for higher accuracy (e.g., for EEG/fMRI work) you'll want to change all timings to frames rather than seconds. You may also want to remove the presentation of images if you're not using them.

- NB no assessment of jitter has been conducted for the current implementation.

# Issues

If you have any issues, find bugs, or observe any unwanted divergences from other implementations please report them using GitHub's issue reporting system. Doing it online rather than by emailing me allows other people to benefit from your experience.

If there are additional features or refinements you would like to see please feel free to contribute to the project yourself by branching, editing, and submitting a pull request on Github. You can also email me at ian.hussey@ugent.be.

# To do

- Add internal consistency calculation to R script

# Changelog

1.2.1

- Checked to work in the latest version of psychopy (1.90) and updated the readme.

1.2

- Removed calculation of D1 scores for trial types (e.g., calculated only from insects trials), as this is uncommon in the literature.
- Corrected calculation of D1 scores. R script did not take block order into account. Thanks to Jamie Cummins for feedback on this.
- Corrected order of blocks 1 and 2. Task is now in better in line with Nosek et al. (2007).
- R script auto now installs its depencies, and produces split half D scores for internal reliability estimations.

1.1

- Seperated stimuli file into exemplars and block_layout. Multiplier is determined by the number of exemplars. As such, the block lengths are fixed by the code, but the number of exemplars can vary freely (although they must be equal across trial types, e.g., all must have N).

1.0.1

R script updated.

1.0

Used in a completed experiment and bug tested. No changes made relative from 0.10, which I've upgraded to a release copy.

0.10

Added new code components so that only a single stimulus file and single blocks file is needed. The contents the category labels and the correct and incorrect responses are determined before each trial based on the trial type, block order and current block. This limits the scope for human error when putting together the stimuli files.

0.9.3

- Corrected ITI from 300ms to 250ms.
- Allows the researcher to display any combination of word or picture stimuli.

0.9.2

- Rewritten from scratch to make the data files follow Hadley Wickham's Tidy Data standards. Loops now include nested references to the stimuli files, thus allowing for a highly simplified flow and greatly decreased number of routines. The py script has also gone from 3250 lines to 450, and is far more readable. However, this change shifts complexity to the stimulus files: there are now 12 where there were 2.
- This rewriting also allowed me to add the ability to choose the block order based on a variable entered in the popup before the experiment (i.e., block order: 1 vs 2).

0.9.1

- changed the instuctions1and5 from "is trials" = true to "is trials" = false so that a superfluous line is not recorded in the data file.
- changed the endLoop from is trials" = true to "is trials" = false so that a superfluous line is not recorded in the data file.
- until now, trial blocks 3&4 and 6&7 used the same routines (repsectively) because the contents of these blocks' trials was identical. However, it makes for easier analysis if the data is output in different columns.
- renamed keyboard components in trial blocks to blockX and blockXwrong. When combined with the above, D scores can be calculated by examining the means and standard deviations of the block1.rt to block7.rt variables, and accuracy exclusions from the block1wrong.corr to block7wrong.corr variables. NB here, the 1s refer to incorrect answers rather than correct answers. The variables specified in the code components for accuracy feedback were also changed appropriately.