

Graph mapping test

This is a README file for **Graph mapping**, a simple and effective test of fluid reasoning/intelligence. This computerized test requires participants to map a pair of corresponding nodes between the two mathematically isomorphic but visually different graphs. Its overall score reflects reasoning ability.

See the description of the conceptual aspects of the test and its validation details in:

[link to the article]

Contents

[Requirements](#)

[Launching the test](#)

[Closing the test](#)

[Configuration panel](#)

[Training session](#)

[Experimental session - choosing between two modes](#)

[#1 Predefined tests](#)

[#2 Randomized experiment](#)

[Provide break after n trials](#)

[Other settings](#)

[Targets](#)

[Show Feedback in training trials](#)

[Visual](#)

[Changing test's instructions and displayed texts](#)

[Main instructions](#)

[Other displayed texts](#)

[Advanced possibilities – defining new trials, designing custom tests](#)

[Defining trials](#)

[Dealing with test's results](#)

[Script for merging multiple result files](#)

Requirements

The test works on Windows and Linux operating systems.

It is implemented using the free PsychoPy package and Python 3 programming language. These need to be installed on your operating system for the test to work.

There are two basic ways to provide the above requirements:

1. **For Windows users, the simplest method** is to download and install the PsychoPy Standalone package, which also, by default, installs the Python 3 programming environment. You may use the link below:

<https://www.psychopy.org/download.html>

After the successful download and installation you should be able to launch the test.

2. **The method for advanced users, Linux users or users with Python 3.x already installed** on their operating system. Please see <https://www.psychopy.org/download.html#manual-installations> for details.

In short:

If Python 3 is not yet installed, please download and install the latest Python 3 distribution (but any other 3.x distribution should be compatible with the test). We recommend you adding Python to your operating system's PATH. This option can be selected during installation.

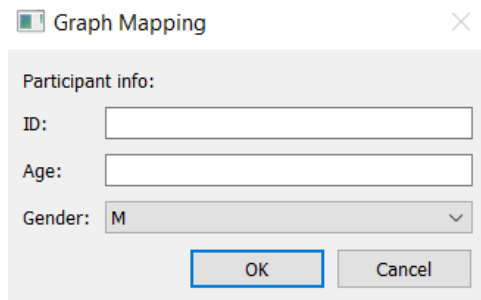
Having Python 3.x installed on your operating system, please install the "psychopy" package. One way to install the package is by providing the *pip install "psychopy"* command in your operating system's command line. This method works only with Python added to your operating system's PATH.

Launching the test

If the above requirements are fulfilled, the test can be launched using the `Graph_mapping.py` file from the main folder. You may also want to create a shortcut to the file and place it on the desktop.

You should be able to open the `Graph_mapping.py` file directly. However, it may happen that, on your system, the files with `.py` extension are associated with some application, for example Notepad. In this case, associate the `.py` files with `python.exe`, which can be found in the PsychoPy3 main folder.

After launching the test, you should see the window below.



The screenshot shows a window titled "Graph Mapping" with a close button (X) in the top right corner. Below the title bar is a section labeled "Participant info:". Under this section, there are three input fields: "ID:" (a text box), "Age:" (a text box), and "Gender:" (a dropdown menu currently showing "M"). At the bottom of the dialog are two buttons: "OK" and "Cancel".

You can run the test without providing the above information. If the information is provided, it will be saved in the `.csv` files that contain the results of the test (folder Results).

The file `Set_config.py` defines the form and sequence of the test items that will be presented. If you haven't changed the original configuration of the test, it will launch with the default configuration. See the Configuration panel section on how to customize the test.

Closing the test

To exit the program before completing the test press "q".

The program saves results even in the case of exiting the test before its completion.

Configuration panel

To customize the test please open the `Set_config.py` from the main folder. After changing the settings and saving the new configuration, the test will run using the new configuration.

To run the test with more than one configuration on a single computer, you may create copies of the main folder and set a different configuration for each copy.

The backup copy of the second to last saved configuration file is saved as `config_last_version.yaml`. You may use the file to retrieve the configuration.

After opening `set_config.py`, you should see the following window. Its options are described below:

Set config

☒ Training session

Select predefined training: 8_trials.csv

Min. required training accuracy: 0.7

Max. training attempts: 3

Experimental session

☒ Predefined test

test_44_items.csv

☐ Randomize trials order

☐ Randomize graphs

☐ Randomized experiment

Choose factor levels (max 3x2x2):

No. of edges:

☐ 3 ☐ 4 ☐ 5

Crossed edges:

☐ Graphs with crossed edges

☐ Graphs without crossed edges

Types of target vertices:

☐ Direct ☐ Indirect

No. of trials per cell:

Total n of trials: ?

Provide break after n trials: 0

Cancel

Next

Training session

Four parameters can be set for the training session. First of all, you may choose whether the training session precedes the main test or not.

Next, you may choose between the available ready-made sets of (4, 6, 8, 10, or 12) practice trials.

Min. required training accuracy [0 - 1.0] is the minimum mean accuracy a participant needs to reach in the training session to proceed to the main test. This can be set to ensure that participants properly understood the test rules before taking the test. If the required accuracy is not reached, the training session restarts from the beginning. Set 0 if you do not want to apply such a criterion.

Max. training attempts defines the maximum allowed number of attempts to complete the training session. After reaching this maximum, the program quits without proceeding to the main test (the result files are saved anyway).

Experimental session - choosing between two modes

The test can be used in two modes:

#1 Predefined tests

In this mode, you may choose from several ready-made tests (sets of predefined trials). The following tests are suitable for psychometric purposes. Each test includes a specified number of unique, predefined trials, by default presented in the fixed order of progressive difficulty:

- test_44_trials – this default validated test includes 44 trials of progressive difficulty (around 15 minutes of administration time). The test should yield a satisfactory variance of normally distributed scores and good reliability. The test has been validated in the psychometric Study 2
- test_20 – the subset of 20 trials from the validated test (around 10 minutes)
- test_30 – the subset of 30 trials from the validated test (around 12 minutes)
- test_60 – the 44 trials from the validated test + 16 additional trials of various difficulty (around 20 minutes)
- easy_26 – the test consisting of 26 easiest trials from the validated test (around 10 minutes)
- hard_36 – the test with a very high level of difficulty; you may consider increasing trial time limit as the trials are complex (20 most difficult trials from validated test + 16 additional difficult trials)

random trials order – selecting this option results in a random order presentation of the test trials. If the option is not selected, the trials will be applied in the predefined progressive order.

graphs randomization – selecting this option will result in a random visual transformation of each of the two graphs in each trial. More details on how the graphs randomization works are provided in the section below.

#2 Randomized experiment

This mode is suitable for experimental purposes as it allows control over the complexity of the trials, determining their difficulty (error rates), while using full randomization. This mode allows you to manipulate the three factors affecting the trial difficulty by applying only the trials from the selected factor levels, with the maximum of 3x2x2 experimental design with all the levels selected. The factors that define the difficulty are the following:

- the number of edges included in the graph (3, 4, and 5)

- whether one of the graphs in a trial has edges crossed (the more difficult condition) or no edge is crossed
- the type of the targets with regard to the mapping requirements, with two possible types:
 - o direct – the target nodes can be identified directly by looking at the unique degree of each target node; this type is easier
 - o indirect – the target nodes cannot be identified directly using the unique degree of each target node; to identify the targets some other nodes need to be identified directly (the graph structure must be represented); this is the more difficult type

No. of trials per cell determines the total number of trials in the experiment, shown as **Total n of trials**, as the product of the experiment cells and the no. of trials per cell.

How does the randomized experiment work?

By default, the order of trials is random.

Importantly, all the graphs with the same number of edges, across all conditions, are isomorphic (constitute structurally one and the same graph). This ensures that the graphs between different trials differ only with regard to the selected factors, protecting from any influence of potential confounding variables related to the specificities of different graphs' structures. To ensure that participants do not learn to recognize the graphs of the particular number of edges as isomorphic, and do not learn any specific solution strategies, we recommend using trials with at least two different numbers of edges. Moreover, the following types of randomization are used in this mode by default (which are optional as *graphs randomization* option in the Predefined test mode):

- random position (left/right) of each graph in a trial
- randomly switching the left button and the right button targets in a trial (in one-target variant the target is random one of the two predefined targets)
- random transformation (either rotation or mirror reflection) of each of the two graphs in each trial. Each graph in a trial is with equal probability:
 - a) randomly rotated by 0/90/180/270 degrees
 - or
 - b) reflected in a random x/y/diagonal1/diagonal2 axis

In consequence of these transformations, each resulting trial is a random variant out of 64 (eight unique variants per one graph * eight unique variants per second graph) equiprobable variants of the generic trial defined for each configuration of the three factor's levels. Notice that these two kinds of transformations (rotation and mirror

reflection) conserve the “shape” of the graph as they do not add or remove any diagonal edges. Therefore, graphs with and without edges crossed retain their respective shape.

Provide break after n trials

Set to display screen with a custom text after n trials. If left empty or set to 0, no break is applied. The default text can be changed by modifying the break.txt file in the Texts folder.

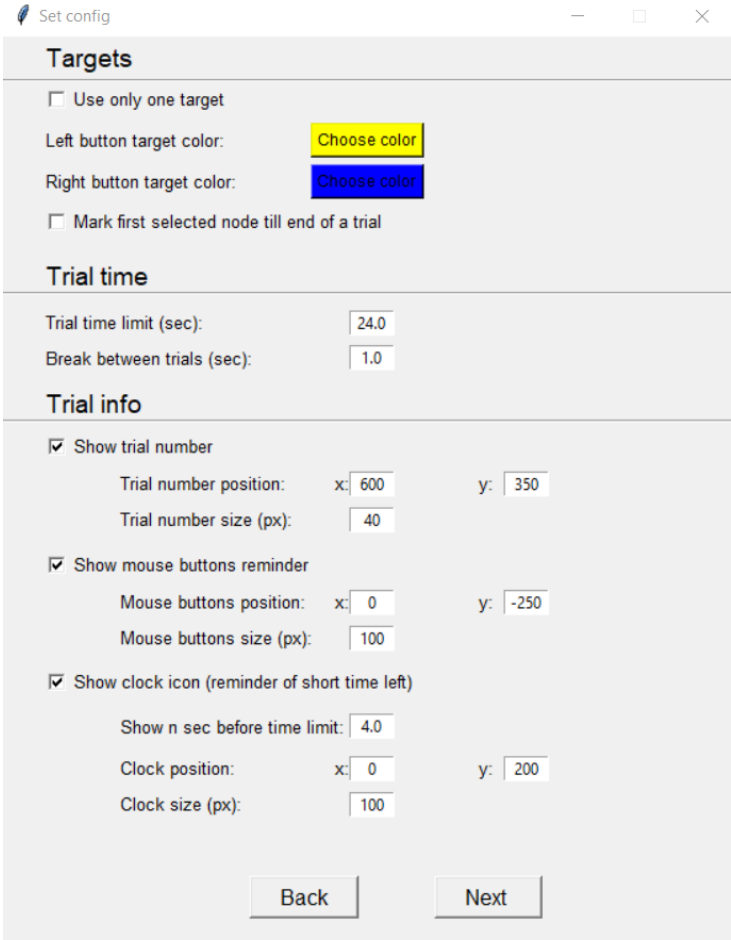
Other settings

Some settings allow you to set the **size and position** of certain elements displayed on the screen.

- The size of elements is defined in pixels - please note that their absolute size may differ depending on a monitor (monitors differ in pixel size).
- The position is specified along two axes as the distance in pixels from the center of the screen, defined as $x=0$, $y=0$.

Most of the parameters are straightforward. The other, non-trivial options are described below.

Targets



The screenshot shows a 'Set config' window with a title bar containing a feather icon, the text 'Set config', and standard window controls. The main content area is divided into sections: 'Targets', 'Trial time', and 'Trial info'. The 'Targets' section includes a checkbox for 'Use only one target', two 'Choose color' buttons for 'Left button target color' and 'Right button target color', and a checkbox for 'Mark first selected node till end of a trial'. The 'Trial time' section has input fields for 'Trial time limit (sec):' (24.0) and 'Break between trials (sec):' (1.0). The 'Trial info' section contains three checked checkboxes: 'Show trial number', 'Show mouse buttons reminder', and 'Show clock icon (reminder of short time left)'. Each checkbox has associated position and size settings. For 'Show trial number', the position is x: 600, y: 350 and the size is 40 px. For 'Show mouse buttons reminder', the position is x: 0, y: -250 and the size is 100 px. For 'Show clock icon', the 'Show n sec before time limit' is 4.0, the position is x: 0, y: 200, and the size is 100 px. At the bottom are 'Back' and 'Next' buttons.

Set config

Targets

☐ Use only one target

Left button target color: Choose color

Right button target color: Choose color

☐ Mark first selected node till end of a trial

Trial time

Trial time limit (sec): 24.0

Break between trials (sec): 1.0

Trial info

☒ Show trial number

Trial number position: x: 600 y: 350

Trial number size (px): 40

☒ Show mouse buttons reminder

Mouse buttons position: x: 0 y: -250

Mouse buttons size (px): 100

☒ Show clock icon (reminder of short time left)

Show n sec before time limit: 4.0

Clock position: x: 0 y: 200

Clock size (px): 100

Back Next

Use only one target

You may specify whether to use two targets (the default), or the one-target variant of the test. In the one-target variant, only one node is highlighted as a target, and only one target needs to be selected by using the left mouse button only. Note that the one-target variant is easier and may have a lower reliability, as the random guessing accuracy rate is larger compared to the two-targets variant

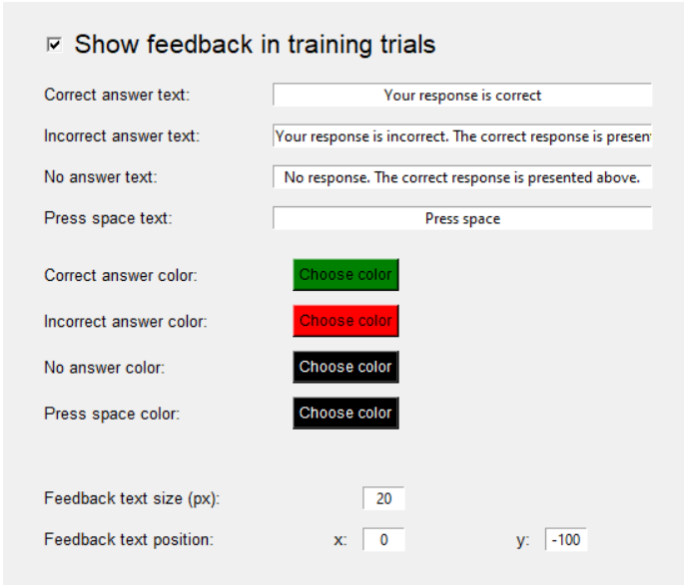
Mark the first selected node until end of trial (concerns only the two-target variant)

If this option is unchecked (by default), after clicking on the first selected node, the node is highlighted with the color assigned to the used mouse button for 200 ms, in order to assure participants that their choice was registered.

If the option is selected, the first selected node is highlighted until the end of trial. This may slightly decrease the difficulty of the test by reducing the memory load, as participants do not need to remember which node they have already selected.

This option does not apply when the *Use only one target* option is selected.

Show Feedback in training trials

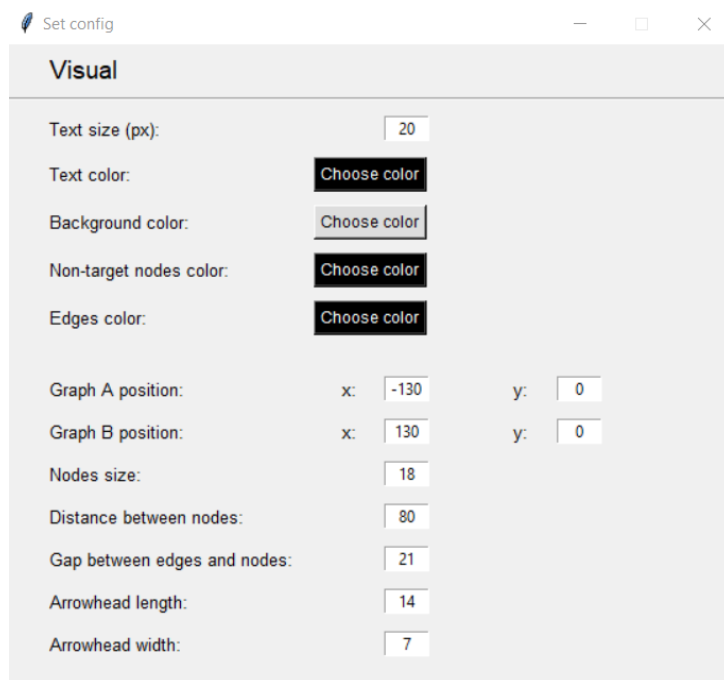


The screenshot shows a configuration window titled 'Set config' with a pencil icon and window controls. The 'Show feedback in training trials' option is checked. Below this, there are text input fields for feedback messages and color selection buttons for each feedback type. At the bottom, there are input fields for feedback text size and position.

Field	Value
Correct answer text:	Your response is correct
Incorrect answer text:	Your response is incorrect. The correct response is presen
No answer text:	No response. The correct response is presented above.
Press space text:	Press space
Correct answer color:	Choose color (green)
Incorrect answer color:	Choose color (red)
No answer color:	Choose color (black)
Press space color:	Choose color (black)
Feedback text size (px):	20
Feedback text position:	x: 0, y: -100

Select this option to provide the participants with the feedback on their accuracy on the training trials and to show the correct response.

Visual



The screenshot shows a 'Set config' window with a 'Visual' tab. The window has a title bar with a pencil icon, the text 'Set config', and standard window controls (minimize, maximize, close). The 'Visual' tab contains the following settings:

Setting	Value
Text size (px):	20
Text color:	Choose color
Background color:	Choose color
Non-target nodes color:	Choose color
Edges color:	Choose color
Graph A position:	x: -130, y: 0
Graph B position:	x: 130, y: 0
Nodes size:	18
Distance between nodes:	80
Gap between edges and nodes:	21
Arrowhead length:	14
Arrowhead width:	7

Text size; Text color

Specify the size (in pixels) and the color of the text used in messages other than main instructions and feedback (see *Changing test's instructions and messages* section)

Graph A position; Graph B position

By default, graph A (the one with the highlighted targets) is presented on the left hand side and graph B (the one in which the selection of corresponding nodes is made) is presented on the right hand side. You can change the relative positions of the two graphs, so that, for example, the graph in which the selection of corresponding nodes is made is placed to the left or below the graph with highlighted targets.

Changing instructions and texts

Main instructions

The instructions for the main test, displayed after the program launch, consist of five .png files corresponding to five consecutive pages (screens). The content of the pages can be easily modified or replaced using the provided source .pptx file (PowerPoint format) in the Images folder. Save the modified slides as the .png files and name them instruction1 etc. **(the .png file names 'instruction1' ... 'instruction5' must not be changed!)**.

To remove instruction pages or add new ones, modify the Python code in the #INSTRUCTIONS section in main.py file, by removing or adding respective lines of code containing the show_image() function. If you want to use plain text instructions instead of .png files, provide a .txt file into the 'texts' folder and load the file using show_info() function instead of show_image() function.

Other texts

Other screens displayed in the test that provide information to test takers are based on .txt files placed in the Texts folder. You are free to modify the content of these files:

- after_training – the message presented after completing the training session
- after_unsuccessful_training – the message presented if the required training accuracy is not reached
- too_many_attempts – the message presented after reaching the max number of training attempts
- break – the message presented during the break
- end – the message presented after the last trial of the test

Advanced functions – defining new trials, designing custom tests

New trials and tests can be created by editing the existing or creating the new .csv files containing the lists of trials. The files are placed in the Trials folder. It is recommended to open the files in Excel or another spreadsheet. Each file contains a list of trials in which each row defines a trial.

List of trials can be easily modified by deleting selected trials, changing the order of trials, modifying existing trials or adding new trials.

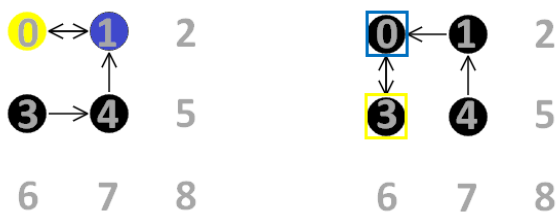
The new/modified files containing lists of trials can be then selected in the configuration panel as new tests or training sets.

Defining trials

Each trial is defined by the following columns/parameters:

	A	B	C	D	E	F	G	H	I
1	NR	FEED	TRAIN	Nodes_A	Edges_A	Nodes_B	Edges_B	Left_button_targets	Right_button_targets
2	1	1	1	0,1,3,4	[(0,1),(1,0),(3,4),(4,1)]	0,1,3,4	[(0,3),(3,0),(4,1),(1,0)]	[0,3]	[1,0]

The trial defined above is shown below. The grey numbers (positions on 3x3 virtual matrices) and the blue and yellow frames on the right (indicating the correct responses) are not displayed in the test.



Graph A

Graph B

The columns/parameters are described below:

- NR – the number of a given trial on the list, the number has only an informational purpose, and does NOT determine the order of trials
- FEED – if set to 1 this allows to display feedback if *Show Feedback in training trials* is set. It is also possible to allow feedback in experimental trials
- TRAIN – either a training or experimental trial, set 1 for training and 0 for experimental trials
- Nodes_A – provides positions (from 0 to 8) of nodes on a 3x3 virtual matrix of the graph A (the graph with targets highlighted). Each combination of nodes on the matrix is possible (each possible “shape” of the graph is allowed) and the graph can include from 2 to 9 nodes.
- Nodes_B – provides positions (from 0 to 8) of nodes on a 3x3 virtual matrix of the graph B (the graph in which participant selects the corresponding nodes). Node options are the same as for the graph A.
- Edges_A – the list of ordered pairs defining directed edges between the given nodes of the graph A (the graph with targets are highlighted); bidirectional edges are made by two corresponding ordered pairs, for example (0,1), (1,0);
- Edges_B – the list of ordered pairs defining directed edges between the given nodes of the graph B (the graph in which participant selects the corresponding nodes); bidirectional edges are made by two corresponding ordered pairs, for example (0,1), (1,0);

- Left_button_targets [A,B] – defines the pair of corresponding nodes to be matched using the left mouse button; the first digit in the pair reflects the target in the graph A, and the second digit reflects a corresponding node in the graph B;
- Right_button_targets [A,B] – defines the pair of corresponding nodes to be matched using the right mouse button; the first digit in the pair reflects the target in the graph A, and the second digit reflects a corresponding node in the graph B;

The remaining columns are descriptive, their values do not affect the procedure and do not have to be provided, **with the exception of trials used in the Randomized experiment mode** placed in trials/experiment/experiment.csv. In this mode the procedure depends on the information provided in NE, Type, and Crossed_edges columns.

- Number_of_nodes – the number of nodes in a graph
- Number_of_edges – the number of edges in a graph
- Bidirectional – the number of bidirectional edges in a graph
- Type – type of targets
 - o DI – direct; the target nodes can be identified directly by the unique degree of each target node
 - o InDI – the target nodes cannot be identified directly by the unique degree of each target node; to identify the targets other nodes need to be identified directly; the more difficult type than DI
- Crossed_edges – whether the trial includes a graph with crossed edges or not
- Name – name of a trial, may help to identify trials

Dealing with test's results

The program saves the results even in case of exiting the test before its completion. The results of each run are saved as two .csv files, one saved in the **Results/summary_results** folder and the other in the **Results/raw_results** folder.

The name of each file is composed as: ID_Gender_Age_Random number

- The file saved in **/summary_results** contains the summary information on the experimental trials only (not on training trials) such as percent of correct responses and their mean latency (RT)
- The file saved in **/raw_results** contains full data on the run of the task. It contains the list of applied training and experimental trials in the order of their presentation. The first four columns provide information about ID, age, gender of a participant, as well as the time of

launching the test. Several next columns provide the information about the set of parameters that specify each trial (see the section above for the description of these parameters).

The information important for analyzing results in Randomized experiment mode that specify the experimental conditions is provided in the following columns:

- Number_of_edges
- Type
- Crossed_edges

The information concerning response – the selected responses/nodes, RT and accuracy in each trial – is provided in the last following columns:

- LEFT_ANS – the node selected using the left button
- RIGHT_ANS – the node selected using the right button
- LEFT_CORRECT – correctness of the left button response (the only response required in the one-target variant)
- RIGHT_CORRECT – correctness of the right button response
- CORRECT – correctness of the response (correct only if both the left and the right response were correct in the default two-targets variant of the test)
- LEFT_RT – latency of the left button response
- RIGHT_RT – latency of the right button response
- RT – latency of selecting the second node

Script for merging multiple result files

Use the merge_files.py script placed in the Results folder to merge multiple results files from /raw_results and /summary_results into two .csv files.