Human task

Run the human task

Read the readme in the main folder for the intention and the instructions about how to run the task.

In monkeylogic, the classical way to set up a task is to write a general script where instructions about how the task goes are written, and other smaller scripts that are specific of one part of the task. In our case, the general script is **task\_human.m** and the smaller scripts are **step\*\_human.m**

**Task\_human.m**

This script is the main script of the task. It contains all the parameters of it, allows to save what happened during the task, and to change the steps depending on several parameters until the task is completed. It is called directly from the monkeylogic interface, you don’t need to run it on the command window.

Here is a list of all the parameters that can be changed :

***L3*** : timingfile = the list of all the smaller scripts used during the task. If one script is added or removed, you need to change timingfile accordingly

***L10*** : TrialRecord.User.nb\_of\_conditions = the total number of conditions the subject will see during the task.

***L12*** : TrialRecord.User.min\_success = the minimal number of successful trials to reach the next step in learning phases

***L13*** : TrialRecord.User.list\_behaviors = the list of all the behaviors that can be shown. */!\ we thought of new behaviors and I haven’t implemented them yet, this list therefore need to be changed if the behaviors change !*

***L146*** : proba > 0.5 = in the test phases, we present sometimes a real test (with unknown pictures), and sometimes it is the learning picture that is shown (to reinforce the learning). The ratio between these two possibilities is determined by this line in test 1

***L223*** : proba > 0.7 = the same as above but for test 2

**Step1\_1\_human.m**

This script is used for the fixation phase between the different conditions. It only displays a yellow triangle on which the subject has to click several times, to notify him/her that a new condition will start.

***L24*** : max\_reaction\_time = the max time within which the subject has to click on a picture, otherwise the trial stops

***L29*** : Threshold = the size of the margins around the fixation object in which a click is considered as a click on the object

***L82*** : idle(700) = the time between the end of this trial and the beginning of the following if the trial was not a success

**Step1\_2\_human.m**

This script is used for the fixation phase between the different conditions. It only displays a blue triangle on which the subject has to click several times, to notify him/her that a new condition will start.

***L24*** : max\_reaction\_time

***L29*** : Threshold

***L82*** : idle(700)

**Step2\_human.m**

This script is used for the learning1 phase. After a fixation step, it displays 1 picture in a random position, the picture of the behavioral category selected in the current condition.

***L59*** : max\_reaction\_time

***L65*** and ***L85*** : Threshold

***L77*** : wthb.Duration = the time between fixation and presentation of the pictures (so that if the subject clicks a little too long on the fixation dot, it doesn’t count as a click on a picture because it had appeared too soon)

***L147*** : idle(700)

**Step3\_human.m**

This script is used for the learning1 phase. After a fixation step, it displays 3 pictures in random positions, the picture of the behavioral category selected in the current condition, and two others taken at random in the other relationship categories.

***L75*** : max\_reaction\_time

***L81***, ***L101***, ***L108*** and ***L115*** : Threshold

***L93*** : Wthb.Duration

***L183*** : idle(700)

**Step4\_human.m**

This script is used for the test1 phase. After a fixation step, it displays 3 pictures in random positions, one from the behavioral category selected in the current condition, the two others taken at random in the other relationship categories.

***L80*** : max\_reaction\_time

***L87***, ***L107***, ***L114*** and ***L121*** : Threshold

***L99*** : wthb.Duration

***L177*** : idle(700)

**Step5\_human.m**

This script is used for the learning2 phase. After a fixation step, it displays 1 picture in a random position, from the behavioral category selected in the current condition.

***L54*** : max\_reaction\_time

***L61*** and ***L81*** : Threshold

***L73*** : wthb.Duration

***L145*** : idle(700)

**Step6\_human.m**

This script is used for the learning2 phase. After a fixation step, it displays 3 pictures in random positions, one from the behavioral category selected in the current condition, the two others taken at random in the other relationship categories.

***L76*** : max\_reaction\_time

***L83***, ***L103***, ***L110*** and ***L117*** : Threshold

***L95*** : wthb.Duration

***L185*** : idle(700)

**Step7\_human.m**

This script is used for the test2 phase. After a fixation step, it displays 3 pictures in random positions, one from the relationship category selected in the current condition, the two others taken at random in the other categories.

***L70*** : max\_reaction\_time

***L89*** : wthb.Duration

***L97***, ***L104*** and ***L111*** : Threshold

***L165*** : idle(700)

**Step\_end\_human.m**

This script displays an ending screen when the task is finished, which lasts 10 s and says that the task is finished and displays how many coins were gained.

*Things to do :*

*In the main script, there’s something wrong with the order of the conditions (for instance if there are 3 conditions, for now it does : C1 T1, C2 T1, C3 T1, C3 T2, C2 T2, C1 T2). I haven’t found yet why.*

Analyze the data obtained from the human task

A .bhv2 file is created each time the task is finished. In this file, a lot of informations about what happened during the task are stored. To make easier the processing of the results, I created a script (**get\_stimuli\_presented.m**) which compute a .mat file with the important informations. Then, several scripts are used to control if there was not a biais in the way the subject did the task : **comparison\_errors.m, mean\_color\_picture.m, plot\_errors.m, plot\_reaction\_time.m, plot\_touch.m, proportion\_touch.m, show\_pictures.m, time\_responses.m**

A script can be used to create all these control figures at once for each subject : **control\_figures.m**

A script was written to concatenate all the .mat files created, in order to analyze all at once to get the “final” results of this task : **concatenate\_files.m**

Other scripts are being written to get these results : **errors\_global.m** and **reaction\_time\_global.m**, with a script to do them at once : **results\_figure.m**

*These last scripts are far from being finished and need to be worked on*

**Get\_stimuli\_presented.m**

Inputs:

* Name\_file = the name of the .bvh2 file, complete, with quotation marks
* Saving = 0 if you don’t want to save, 1 if you want to save

The script creates a .mat file to store all the important informations from the task. The .mat file created from the .bhv2 file is a table where each row is a trial of the task.

Here is a list of what is stored in each column :

C1 : the « good » picture presented

C2 : the name of the first distractor picture if 3 pictures were presented, or an empty cell

C3 : the name of the second distractor picture if 3 pictures were presented, or an empty cell

C4 : the position where the « good » picture was presented

C5 : the position of the first distractor picture if 3 pictures were presented, or an empty cell

C6 : the position of the second distractor picture if 3 pictures were presented, or an empty cell

C7 : the position of the touch when the trial is considered as done (ie a click on a picture was made), or a NaN if no touch was detected

C8 : the name of the picture touched if 3 pictures were presented, or an empty cell if no touch was detected

C9 : the phase number (20 = learning1 1picture, 30 = learning1 3pictures, 40 = test1, 50 = learning2 1picture, 60 = learning2 3pictures, 70 = test2)

C10 : reaction time

C11 : error kind (0 = successful trial, 1 = no touch in fixation, 2 = no touch on a picture, 3 = incorrect picture in learning, 4 = incorrect picture in test1, 5 = incorrect picture in test2)

C12 : phase kind (learning, test1 or test2)

C13 : the name of the condition

C14 : condition number

C15 : name of the subject (so that when we concatenate the different files, we know which is which)

***L2*** : addpath = path to change depending on the computer you use

***L88*** : conditions = all the different conditions possible in this task */!\ you need to change this if you change the different categories (which should be done as new categories have been defined)*

***L90*** : cond = all the possible names of the pictures, so that the script can match these names with a condition */!\ you need to change this if you change the different categories (which should be done as new categories have been defined)*

**comparison\_errors.m**

Inputs:

* Stimuli = the .mat file from get\_stimuli\_presented where all the useful informations are stored

This script show the distribution of the reaction times during the different phases of the task, depending if the trials were successful or not. This is useful to see if the subject took more time to answer when she failed or not.

**mean\_color\_picture.m**

Inputs:

* Stimuli = the .mat file from get\_stimuli\_presented where all the useful informations are stored

This script analyses the mean color of the “right” pictures presented and of the pictures clicked on. It is useful to control if there is a bias toward a certain kind of picture (for instance, if the subject clicked more on pictures that contain a lot of green).

***L5*** : cd = change the current folder to go to the folder where all pictures are stored, you must change this path depending on the computer you use

***L54*** : cd = same as above

**plot\_errors.m**

Inputs:

* Stimuli = the .mat file from get\_stimuli\_presented where all the useful informations are stored

This script plot the kind of errors that occurred across time, as well as the proportion of errors in each phase and the proportion of errors depending on the condition number. This is useful to know if the subject did more errors in the first condition or not (for instance because she needed time to understand properly the task, or on the opposite because the subject got bored in the end).

**plot\_reaction\_time.m**

Inputs:

* Stimuli = the .mat file from get\_stimuli\_presented where all the useful informations are stored
* All\_trials = 0 if you want to look only at the test trials, 1 if you want to look at all trials

This script plots the distribution of reaction times for the different phases, taking all trials into account or only the test trials. This is useful to know if there are differences in the time needed to choose the picture depending on the phase.

**plot\_touch.m**

Inputs:

* Stimuli = the .mat file from get\_stimuli\_presented where all the useful informations are stored

This script plots where the subject touched the screen during the task.

**proportion\_touch.m**

Inputs:

* Stimuli = the .mat file from get\_stimuli\_presented where all the useful informations are stored
* All\_trials = 0 if you want to look only at the test trials, 1 if you want to look at all trials

This script plots the proportion the “right” picture is presented in each position, and the proportion the subject clicked on each position, taking all trials into account or only test trials. This is useful to know if there’s a bias toward one of the positions (for instance the middle one), everytime or only during test phases.

***L10*** : squares = the different positions where the pictures were presented. If the screen on which the task is made changes, these positions might change, so you need to change this

**show\_pictures.m**

Inputs:

* Stimuli = the .mat file from get\_stimuli\_presented where all the useful informations are stored

This script is not finished at all, it is meant to show all the pictures presented and clicked on during the task, to see visually if there seems to be a bias toward certain kinds of pictures. *It should be worked on!*

**time\_responses.m**

Inputs:

* Stimuli = the .mat file from get\_stimuli\_presented where all the useful informations are stored

This script plots a whisker plot of the reaction times across phases and across success or failure. It is useful to see if some phases are harder than others.

**control\_figures.m**

Inputs:

* Name\_file = the name of the .bvh2 file, complete, with quotation marks

This scripts allows the user to compute all the above functions at once and to store all the figures in the same folder.

***L2, L3, L4, L8, L9, L10, L35*** : all the lines where paths are written, that have to be changed depending on the computer you’re using

*Things to do :*

* *Change the folder pictures in control figure to put the new pictures in it*
* *In get\_stimuli\_presented, you need to change L84 : it is this way because in the script of the task the order is not good, but if you find how to change it in the task script, you’ll have to change this line aswell!*
* *In the end of the task, a “end screen” is presented. It was not taken into account when writing the get\_stimuli\_presented, but that may change the length of everything. You’ll need to check it!*