USERS' MANUAL FAB

2019.6

Table of Contents

1. Overview

This tool is designed to run psychological or linguistic tasks that allow participants to go through sequential stimuli on their own paces, and record the time spent on each stimulus. The primary goal of this tool is to provide a free and handy instrument which simulate the eye-tracking process to explore the mechanism of human cognition.

Based on website development languages (JavaScript, HTML, and CSS), this tool does not need you to install any special software on your own computer or computers used to run the experiments. All you need is a common web browser (e.g., IE, Firefox, Chrome, Safari; Chrome is recommended). Both task generation and data collection process run locally by default, so the internet connection is not required.

You can imagine this tool as a detail-oriented consultant (the "hello" file) with a dummy machine designed to run self-paced tasks. The consultant will firstly ask you some basic questions for your experiment. For example, what color and font do you like to set up for your stimuli? Which keys do you require participants to press in order to go forward or go backward? With these questions answered, he will give you an important "card" (JavaScript file) that the machine can understand, and you will insert it on the machine (just copy it and paste to the "robot" folder).

The consultant will then tell you how to prepare the stimuli list. You can prepare a CSV according to same rules, and then ask the consultant to transform it to a second important "card", which you will also insert on the machine.

The machine is dummy. Once you insert two cards and open it, it runs, displays stimuli and records data. If you want it to run another new experiment, you can just retrieve cards and insert other cards (paste new JavaScript files to replace the existing JavaScript files). With two cards inserted, you can copy it to multiple computers in labs, just as other experimental procedures. You are welcome to rename the machine folder as you like anytime.

2. Set Up a Task

2.1 The First Card: Convey Basic Parameters

Firstly, Open the "index.html" in the "hello" file (If you see jammed codes instead of a webpage, please close, right-click, and open it with a web browser), and fill the form in "Part 1". Hopefully the webpage is already easy to follow but you could find more detailed information for each question in the following instruction.

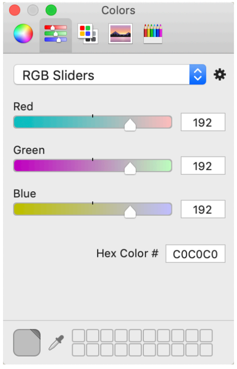
2.1.1 The Stimuli Setting

2.1.1.1 Please choose the type of stimuli

Choose the "picture" option if you only use pictures in the experiments, otherwise choose the text. The difference between the two options is that the "text" option will trigger two extra divisions to adjust the text layout or the segmentation pattern.

2.1.1.2 Please choose a background color for your screen

The default color is RGB 192,192,192.



RGB Sliders

Red 192

Green 192

Blue 192

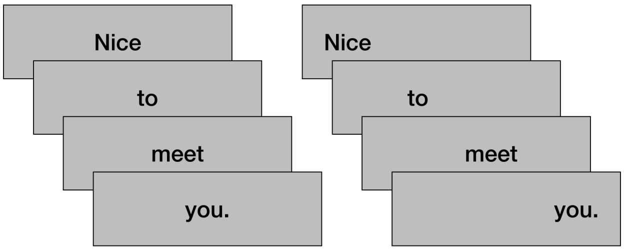
Hex Color # C0C0C0

2.1.1.3 Please set a layout for your text stimuli (only for text stimuli option)

Adjust the color, font size, font styles for the text. The stimulus layout will be identical to the example layout here. The font size will depend on the screen resolution ratio and whether you zoom with the browser, so please make sure not to zoom the browser now and at least try on the lab computer once before collecting data.

2.1.1.4 Please choose whether stimuli should display always on the center or sequentially

Differences between two display patterns are shown in the following picture. Usually we will choose "sequentially" option when doing linguistic experiments. If the "sequentially" option is chosen, you will then adjust where sequences should begin on the screen.



On the Center Sequentially

2.1.1.5 Please choose your segmentation pattern (only for text stimuli option)

There are three patterns that can help you to convey how to segment the sequences to the machine. You should choose here and then reform your stimulus list correspondingly in the next part.

- Pattern 1 -- by space -- will be usually used in space-based segmentation language and picture stimuli. You will write sentences like “nice to meet you.” in the CSV, and then the machine will display “nice”, “to”, “meet”, “you.” one by one and automatically add space between two lexicon stimuli.
- Pattern 2 -- by character -- will be used in character-based languages like Chinese, Japanese, Korean, etc. You will write sentences like “很高兴认识你。” in the CSV, and then the machine will display “很”, “高”, “兴”, “认”, “识”, “你。” one by one without space. The punctuations “。”, “，” , “、”, “？”, “！”, “：”, “；” will be displayed together with the last character rather than independently.
- Pattern 3 -- by “|” -- will be used if you want to segment the sequences using other rules. For example, if you want to display a paragraph sentence by sentence, you will write in the CSV like “Hi. |I am FAB. |Nice to meet you.”, and then the machine will display “Hi. ”, “I am FAB. ”, “Nice to meet you.” one by one. It is also useful if you want to display character languages by phrases rather than characters.

2.1.2 The Keyboard Setting

2.1.2.1 Do you allow participants to go back during the learning process?

If you choose yes, participants will be able to go back-and-forth during the task, quite like an eye-tracking process. Data would be richer but also more complicated. If you choose no, participants can never look at the previous words/pictures once they press the forward key.

2.1.2.2 Please press and save the response keys

At the “index.html” page, once you press the keyboard, the corresponding key information will be shown on the lower right “key cube”. After considering the forward/backward response key, you can press it, check its information in the key cube, and then click the saving button to save it. If you want to change the key, you can press a new key and click the saving button again. The key information is compatible since different browsers use same key code list, that is, you are confident to tell participants to press say “J” key, whenever they are using IOS or Windows, Safari or Chrome, etc. However, the physical positions of keys might differ according to keyboards.

2.1.3 Comprehension Questions and Feedback

2.1.3.1 Do you have any comprehension question for participants

If you have comprehension questions for some/all stimuli, you should answer yes. The questions will be listed in the CSV then but here you will set up how participants should answer the question.

- Yes, and it is a multichoice question -- for common comprehension questions like “yes” or “no” questions. At most four answer levels could be set up. The keyboard setting procedure is identical to forward/backward keys. Forward/backward keys are allowed to be set up as answer responses. You can set up your own labels for different answers in the dataset. If you want to record accuracy or give feedback to participants, the answer label setting is important, for the system will record the answer as correct only when it exactly matches the correct answers indicated in CSV. For example, if you use “Y”/ “N” in CSV, you should also set “Y”/ “N” labels in the textbox here (without quotation marks).
- Yes, and it is a multichoice question -- for all other questions that are more complicated. Participants will see the question followed by a textbox input, and once finishing the answer they can click the submit button to continue.
- No -- if you choose this option, this step would all finish and you do not need to indicate anything about the question in the CSV as well.

2.1.3.2 Do you want to have a special font layout for comprehension questions

You are allowed to set a special layout for comprehension questions, helping participants to distinguish the stimuli and question. All the steps are identical to the text stimuli layout setting. You can also just use pictures as questions. The details will be shown in 2.2 part.

2.1.3.3 Do you want to give feedback to participants' answer

You can give feedback to participants. After receiving the feedback, participants need to press the forward key to continue (which should be indicated in the experimental instruction). As mentioned before, the system will record the answer as correct only when it exactly matches the correct answers indicated in CSV. Therefore, the text input answers are allowed but not recommended to give feedback. The font and size of feedback will be the same as questions, and an italic style is added. The positive color will be blue (RGB 0,0,255) and the negative color will be red (RGB 255,0,0).

2.1.4 The End Massage Setting

You can input custom message here for participants when they finish the task. All other instructions or breaks will be set later in Part 2.

2.1.5 The Dataset Setting

2.1.5.1 Delete reading responses that are less than __ ms

Setting up this threshold will also get rid of all the regression-related DVs in backward reading for dwell times under that threshold, such as numbers of regress-ins and -outs, prob. of regress-ins and -outs, etc. However, you should remember that once you set up this threshold, the dataset will only contain recordings slower than the threshold and you will never have the original full data again.

2.1.6 Download the Parameter File

Now you have finished the first part and can get the first “card” for the self-paced reading machine. Click the download button and an exp.js file will be saved in the default folder of your browser (e.g., Download in mac). You should remove exp.js to the machine folder. Note that the file name should be exact exp.js, not exp(1).js, exp(2).js or anything else. If so, you need to rename it as exp.js.

Here will be a screenshot soon (after I confirm the folder and file names).

2.2 The Second Card: Convey Stimulus list

Let us prepare for the second file now. You will prepare a CSV file yourself according to the following rules. It is recommended to use office tools (e.g., Excel, Numbers) to develop an editable table and then export it into CSV.

2.2.1 Prepare a stimulus list

The system prepares some keywords for column names. All keywords are case-sensitive so make sure you give right names. The necessary column is fab_stimulus, which you input your stimuli, with “/” (but not “\”) at the begining and the end of the sentences.

(all the tables here can be replaced by screenshots in excel after discussion)

fab_stimulus
/The gardener that the homeowner envied had a coffee in the backyard./
/The bird is on the tree./
/The auditor that defied the bookkeeper bought a house downtown Chicago./

Sometimes you want to add a fixation "+" before sentences, but you do not want participants to go back to the fixation when reading backward. In this case, we can add another "/" to separate different part of the stimuli. The sequences included by two slashes are called chunks. In a back-and-forth task, participants can only go back-and-forth within the chunks.[1]

fab_stimulus
/+/The gardener that the homeowner envied had a coffee in the backyard./
/+/The bird is on the tree./
/+/The auditor that defied the bookkeeper bought a house downtown Chicago./

If you want to add special interesting area in sentences just as in eye-tracking task, you can add "*" within chunks (remember to add spaces besides asterisks if the segmentation is "by space"). Then the dataset will indicate which area the word belongs to (area 1, area 2, area 3, etc).

fab_stimulus
/+/The gardener * that the homeowner envied * had a coffee in the backyard./
/+/The bird is on the tree./
/+/The auditor * that defied the bookkeeper * bought a house downtown Chicago./

The basic table is finished now. The machine can work with just one "fab_stimulus" columns. But you might want to use more functions, as follows.

fab_stimulus	fab_question	fab_key	fab_feedback	fab_instruction	fab_qwait
/+/The gardener * that the homeowner envied * had a coffee in the backyard./	The gardener had a coffee?	Y	1	ins.jpg	
/+/The bird is on the tree./	flower.jpg	N	1		

[1] Adding slashes or fixations for all stimuli are actually simple in Excel/Numbers. You can create a new column, input: ="/+/"&A2&"/", use the fill handle to automatic apply the formula to all lines, and copy and paste this new columns as "values only" (A2 represent the blank that store the sentence).

/+/The auditor * that defied the bookkeeper * bought a house downtown Chicago./	The bookkeeper defied the auditor?	N	0	Now the practice is over and the formal experiment begins.	

- fab_question: you can add one question for each stimulus, which could be displayed by text or a picture.
- fab_key: The correct answer to the question. It is necessary if you want to record the accuracy or give feedback.
- fab_feedback: a column that if you want to customize your feedback. For example, you can only give feedback in the practice part. The feedback type (both or only negative) is chosen in Part 1.
- fab_instruction: you can add instruction before each stimulus, using a picture or sentences. Participants need to click a "next" button to continue. Scrolling is allowed here so the content could be long.
- fab_qwait: how many ms the program should wait to display the question after the participants finish the stimulus.

Now maybe you still want more information to be included in the dataset for future data analysis, you can just add them into the table as you like, and the machine will copy this information to the final dataset. Just remember that the column names cannot be identical to the existing keywords (just don't begin with fab_ or dat_).

block	trial	condition	fab_stimulus	fab_question	fab_key	fab_feedback	fab_instruction
pra	p1	a	/+/The gardener * that the homeowner envied * had a coffee in the backyard./	The gardener had a coffee?	Y	1	ins.jpg
pra	p2	filler	/+/The bird is on the tree./	flower.jpg	N	1	

1	1	b	/+/The auditor * that defied the bookkeeper * bought a house downtown Chicago./	The bookkeeper defied the auditor?	N	0	Now the practice is over and the formal experiment begins.

2.2.2 Check your lists

There is one important thing to check before export the file as CSV. Since CSV use English comma as segmentation between cells by default, we cannot include any English comma in the table, or the segmentation process would go wrong. This sounds ridiculous for linguistic tasks but there is really an easy way to solve it. Just find all "," and replace with "&#44", a code for the English comma in HTML. Participants will see comma as usual in the experiment, and the CSV segmentation will also run successfully.

2.2.3 Export the CSV and Compile the Stimulus File

the CSV file can be exported from the table in Excel/Numbers now, through file-> save as/export to -> file format: CSV. It is recommended to always revise your table in the .xlsx or .numbers file and then export to CSV rather than directly edit in CSV.

After exporting the CSV, you can now go back to Part 2 in "hello", upload the CSV file and click the download button to download the second "card"-- sti.js. You will remove it to the machine folder as you have done with the exp.js.

Here will be a screenshot soon (after I confirm the folder and file names).

2.3 The extra pictures

If you indicate any pictures for instructions, stimuli, or questions in the stimulus list, you should include them in the "media" folder under the machine folder. Please check whether file names and types are identical to the description in CSV file.

3. Running the Task

Open Robot -> "index.html"

To be continue

Draft:

Terminate and save the incomplete data: shift+q

If you want to enter the full screen mode, you can do this just as the way in other normal websites (e.g., F11 in Windows system)

Datasets are set to save in the default download folder of the browser after the experiment is complete.

4. Data collection and Analysis

To be continue

5. FAQ

To be continue

6. Appendix: Stimulus list demo

To be continue