This is not a full task description, but describes some issues that came up during an email conversation about the SRT.

Where are the tasks:

1. SRT Learning phase (practice round, rssssr) is now in the folder “1-SRT L”
2. Retention phase (no practice, rsr, DM question, process dissociation)

Email exchange between Sayako & Jana on these tasks:

A few (unresolved, at least not resolved to my satisfaction) issues that are good to have in writing.

**Where are the tasks:**

Depending on what you’d like, the SRT Learning phase (practice round, rssssr) is now in the folder “1-SRT L”, and the Retention phase (no practice, rsr, DM question, process dissociation) is now in the folder “2-SRT Retention-RSR”, both within DAD/Methods/Tasks/E-Prime/Other tasks/SRT.

Note that I have not tested these since I don’t have a button box here. This means that if you’d like to try out the task, please use the E-Studio (.es3) file, not the E-Basic file (.ebs3) file, since the latter may not be up-to-date.

**Weird structure of the RSR task:**

(You can skip this if you’re not interested in running the retention file, or if you just wanted to extract the DM question or the Process Dissociation bit from it.)

When I received the task (from Phillip via Kaitlyn?; without process dissociation or DM question, we added those), I was very confused about how this Retention task is any different from the Learning phase task since it looks virtually identical in E-Studio. The only differences are that a) there is an easy-to-miss extra bit of code at the beginning of the retention phase file that tells E-Prime to skip the instructions and the practice and to get right to the beginning of the real trials, and b) three of the four sequence blocks are crossed off List1, so that only one block gets presented. However, and this is the confusing part until you check the bit of code and List1, instructions, practice, and the other three sequence blocks are still in the experiment file structure, they just get skipped. Doing it this way is very *useful* when you make changes to the task since you basically only need to make them once and then just save-as different versions of the file to yield your learning and your retention phase. But it’s also confusing whenever I don’t look at the file for months.

**An issue regarding the practice round:**

There is an inconsistency in this version of the task that I haven’t yet been able to solve.

In the main task, the experiment will only move on to the next trial if this trial was responded to correctly. (This is a change to how Phillip (? Kaitlyn?) had originally programmed it - in that version, the experiment moved on regardless of which button was pressed. We achieved this by making only the correct key a “termination response" key to press.) In contrast, in the practice round, the experiment moves on to the next trial regardless of which button was pressed. This was originally a mistake on my part, and we actually tried to change this (so even practice trials would only advance if the correct button was pressed) for Michael’s Hong Kong project and for another project I’m currently running in Germany.

However, changing the behavior of the practice trials (to be identical to the behavior in the real trials and to only move on when a correct key was pressed) resulted in the “incorrect!” feedback only showing up when a correct button was pressed after an incorrect one, which isn’t great.

There was also some combination of settings I tried out, which then worked fine with regards to the feedback during practice, but then the practice loop (which sends participants back for another round of practice if they make more than X% mistakes) did not work anymore. That is, even if a person got all practice trials wrong, the experiment simply advanced.

I think you may wanna play around a bit to see if you can get the practice round to work consistently with how the real task works (without sacrificing the practice loop).

And lastly, I’d double-check that everything gets collected okay, especially accuracy. I struggled a bit with that the first time around because the combination of “Only advance if a correct key is pressed” and “Collect accuracy in any case” requires a specific (non-intuitive) set of settings. But I think it should work. :) (Note that I do not know whether any RT that gets measured for inaccurately-responded-to trials is to the first (incorrect) or to the final (correct) key. I believe it may be the latter, since those values are rather high for my participants, but I never tried it out since it’s not a critical measure.)