**SSG script SSG\_Feb2018\_IIGI\_adjPoppy**

Irene van de Vijver, 6 februari 2018

Note: we replaced the soft ice with a calippo, but you may want to change that back for consistency (you can just replace icecream2 in the stimulus folder, but you may need to adjust the size as well in the SSG\_setup\_pictures.tem as the calippo needs to be a bit bigger than the other pics).

Note: output files are not stored in the log folder but in the main folder, because the stim\_out\_set and test\_val\_set files (see below) otherwise cannot be read back in when restarting the task (or at least I don’t know how).

**SSG\_main\_sce**: the only .sce script. Important aspects:

* It calls the template script SSG\_setup\_pictures.tem, in which all the texts, picture parts and pictures are defined.
* It defined all the trial types.
* In the pcl part, variables are defined that are used in multiple subscripts and can therefore not be defined locally.

**SSG\_main\_pcl**: this is really the core of the experiment. Here all important settings are defined and all the subparts of the experiment are called. Important aspects:

* Definition of all the task-relevant variables:
  + Training currently set at 32 blocks of 4 stim x 4 rep = 16 trials => these are 16 odd and 16 even blocks.
  + Test currently set at 4 blocks of 8 stim x 4 rep = 32 trials
* Experimenter input:
  + Condition = ii or gi
  + Task part = with which part do you want to start (see bottom of this script).

Note: you can use this to restart the task if you for example need to repeat a part. However, every participant has to start with part 1 the first time, because then the information is saved about s-r-o relations etc. which you need if you want to restart while keeping the same relations.

* Variables that are set automatically:
  + Trial order is only filled here, but shuffled later
  + The stim\_out\_set and test\_val\_set matrices are where you set the S-R-O-relations.

Stim\_out\_set is already adjusted for the 50/100 settings.

If you change the number of test blocks, test\_val\_set also needs to be adjusted.

* Inclusion of other pcl files with subroutines
* Creation of output file
  + The info file is immediately filled. It saves all the manual settings as well as the matrices.
* In the big task\_part loop, all parts of the task are presented separately from subroutines. The subroutine run\_task is the one that runs the training and test. With the settings of demoORreal and trainORtest you determine what kind of block is presented.

**SSG\_sub1a\_write\_info.pcl:** creates a logfile with all manual settings and the stim\_out\_set and test\_val\_set sets.

**SSG\_sub1b\_write\_matrices.pcl:** subroutines to store stim\_out\_set and test\_val\_set matrices to reload if task has to be restarted.

**SSG\_sub2\_create\_output\_files.pcl:** creates all output files for all separate task parts. Note that the files are created here but information is stored in them in the different subroutines. If you want to add/remove something, you need to add/remove the header here and the storage of information elsewhere.

**SSG\_sub3\_present\_instructions.pcl:** separate subroutines present separate instruction parts. Note that while you present the instructions, you can move forward with space bar and backwards with backspace. Make sure to update the size of the loops if you add or remove screens. I have added screenshots of different task parts as background during the instructions for clarification.

**SSG\_sub4\_run\_task.pcl**: run through parts of a block (included in subroutines 5a-5d).

* In this part, you can set what a block needs to look like (in a very non-elegant way): you either activate:
  + 1 SHOW AND CHECK OUTCOMES: shows outcomes in green/red boxes and presents fruitpicker, repeated if answered incorrectly
  + 1 SHOW OUTCOMES AND PRESENT IMPLEMENTATION/GOAL INTENTIONS: shows outcomes in green/red boxes and presents intentions.
  + Current settings: outcomes shown before every block, followed by intentions during training (but not test)
* run\_trials: subroutine that runs the trials in a block
* Present outcomes on screen after block
* Include a break every x blocks (set in main\_pcl).

**SSG\_sub5a\_show\_outcomes.pcl**: presents the valuable and non-valuable outcomes in the green/red boxes

**SSG\_sub5b\_check\_outcomes.pcl**: presents fruitpicker

**SSG\_sub5c\_present\_intentions.pcl**: presents implementation/goal intentions. Number of repetitions is defined in the main\_pcl.

**SSG\_sub5d\_run\_trials.pcl**: runs through trials in a block (see run\_task).

**SSG\_sub6\_SOtest.pcl**: test of explicit knowledge of S-O relations including confidence rating

I still want to add the quadrants to indicate whether people paid attention to the ice creams / ice cream vans and pressed to get points / avoided to lose points.

Output files

Info file is quite self-explanatory.

\_showoutcomes.txt

* DateTime: date and time
* DemoReal: 1 = demo, 0 = real
* TrainTest: 1 = training, 2 = test
* Block: block number
* ValSide: 1 = valuable outcomes (green box) left, 2 = valuable outcomes right
* S\_Pic1\_V1 valuable picture 1
* S\_Pic2\_V2 valuable picture 2
* S\_Pic3\_N1 non-valuable picture 1
* S\_Pic4\_N2 non-valuable picture 2

(duration of presentation of outcomes in info file)

\_checkoutcomes.txt

* DateTime: date and time
* DemoReal: 1 = demo, 0 = real
* TrainTest: 1 = training, 2 = test
* Block: block number
* C\_Pic1 first picture presented in row
* C\_Pic2 second picture presented in row
* C\_Pic3 third picture presented in row
* C\_Pic4 fourth picture presented in row
* Corr1 correct response 1 (should be given, 1-4)
* Corr2 correct response 2 (should be given, 1-4)
* Resp1 response 1 (actually given, 1-4)
* Resp2 response 2 (actually given, 1-4)
* RT1 RT1
* RT2 RT2 (RT between response 1 and response 2)

\_intentions.txt

* DateTime: date and time
* IIGI: 1 = implementation intentions, 2 = goal intentions
* DemoReal: 1 = demo, 0 = real
* TrainTest: 1 = training, 2 = test
* Block: block number
* Pic presented picture (if II)
* Outc presented outcome (if GI)
* Corr correct response
* Resp given response
* Acc accuracy
* RT RT

\_train.txt

* DateTime: date and time
* IIGI: 1 = implementation intentions, 2 = goal intentions
* DemoReal: 1 = demo, 0 = real
* TrainTest: 1 = training, 2 = test
* Block: block number
* Trial: trial number
* TrainRespreq: response required for this stimulus during training, 1=yes, 0=no
* TestRespreq: response required for this stimulus during test, 1=yes, 0=no (999 during training)
* Trainingcond: 1 = valuable, 0 = non-valuable (currently identical to trainrespreq)
* Setnr: 0 for odd blocks, 4 for even blocks
* Stimulus: stimulus
* Outcome: outcome
* Response: given response, 1=response, 0=no response
* RT: RT; 0 if no response was given
* FB: 1=pizza/ice cream/banner stayed on screen, 2=p/i/b exited with van
* Correct: was the given (non-)response correct, 1=yes, 0=no
* Currpoints: points earned on this trial
* Blockpoints: points earned in this block so far
* Totalpoints: points earned in the whole task

\_text.txt: almost identical to \_train.txt; only difference: Testcond instead of Trainingcond.

* Testcond 1 = still valuable, 2 = devalued, 3 = upvalued, 4 = still not valuable

\_SOtest.txt

* DateTime date and time
* Stim\_tr how manieth stimulus (trial)
* O\_Pic1 outcome presented in first position (top)
* O\_Pic2 outcome presented in second position
* O\_Pic3 outcome presented in third position
* O\_Pic4 outcome presented in fourth position
* Stimulus ice cream van
* Corr\_resp correct response (1-4)
* Resp given response (1-4)
* RT RT
* Accuracy Accuracy of given response
* Confidence Confidence rating, scale 0-400
* RT\_confidence RT of confidence rating