**README Taste Test**

Table of Contents

[Introduction 2](#_Toc100745864)

[Detailed description of phases of the experiment 2](#_Toc100745865)

[Script description TUE008\_Show\_TasteTest.m 2](#_Toc100745866)

[Part 1: Preparation 3](#_Toc100745867)

[Part 2: Load stimuli images 3](#_Toc100745868)

[Part 3: Paradigm settings 4](#_Toc100745869)

[Part 4: Language, texts, instructions, and repetitions 4](#_Toc100745870)

[Part 5: Screen 4](#_Toc100745871)

[Part 6: Initialization before experiment loop 5](#_Toc100745872)

[Part 7: Start of the experiment and timing 5](#_Toc100745873)

[Part 8: Experimental loop 5](#_Toc100745874)

[Part 9: Winning trial 6](#_Toc100745875)

[Part 10: End of experiment 6](#_Toc100745876)

[Part 11: Save experiment data 6](#_Toc100745877)

[Output description 6](#_Toc100745878)

[Supporting files 7](#_Toc100745879)

[Joystick 7](#_Toc100745880)

[Folder structure 7](#_Toc100745881)

[Instructions 7](#_Toc100745882)

[Jitters 7](#_Toc100745883)

[VAS 7](#_Toc100745884)

[References 7](#_Toc100745885)

Current version: TUE008

# **Introduction**

Through this ~20 min food taste task, the consummatory reward facet of anhedonia is assessed. The task includes several phases. First, in phase I, participants will be asked to bid for the snacks to determine the participant’s willingness to pay for each snack. Next, they will be asked to anticipate the taste and taste the snacks, and to rate the experienced reward (i.e., consummatory liking) in phases II to IV. In phase V participants will repeat the willingness to pay trial, giving them the chance to increase or decrease their bid after they have tried the snacks (learning facet). Lastly, they will have the chance to receive one of the items as a reward in the winning trial:

## *Detailed description of phases of the experiment*

* **Phase I: Willingness to pay I** 🡪 Taste test food items are visually presented, and participants have to enter a price that they are willing to pay for each item.
* **Phase II: Taste test anticipation** 🡪 Participants are instructed to inspect the food items in front of them (smell, look), followed by taste test ratings.
* **Phase III: Taste test consumption I** 🡪 Participants are instructed to taste the food items in front of them, followed by taste test ratings. Every food item is assessed twice in this phase.
* **Phase IV: Taste test consumption II** 🡪 Participants are instructed to taste the food items in front of them, followed by taste test ratings. Every food item is assessed once in this phase.
* **Phase V: Willingness to pay II** 🡪 Taste test food items are visually presented, and participants have to enter a price that they are willing to pay for each item.

**\*\* The output from Phase V is stored in the output file under the name of phase 6 because the Taste test consumption I section is repeated twice, and therefore, from the second run of phase III (phase 4 in the output), the index of all the phases are shifted one number. \*\***

* **Winning trial** 🡪 Displays on the screen if an item is won or not together with the snack image, taking into account the item bid in the willingness to pay II section.

# **Script description TUE008\_Show\_TasteTest.m**

There are three general blocks in which the script can be divided:

* Experiment preparation and customization: Parts 1 – 6
* Experiment run: Parts 7 – 10
* Saving experiment data: Part 11

## *Part 1: Preparation*

In this section, the following points are taken into consideration:

* Clear workspace
* Set random seed
* Get operating system and path information
* Project and subject information: Request subject and session information from experimenter. Convert this information to the appropriate formats (e.g. num2str, abbreviations etc.). Participant ID is filled with zeros at the front of the number until the ID consists of a total of 6 digits.
* Language selection: German or English
* Controller selection: Xbox gamepad (gamepad) or mouse
* Screen preparation

The settings that are found at the beginning of the Taste Test and that can be used to customize the script. Here, elaborated in order of appearance:

|  |  |
| --- | --- |
| subj.study = 'TUE008'; | (Short) indicator of project. Is used to load/save files with appropriate project names. |
| subj.run = '1'; | The Taste Test is a single run for TUE008.  Run here means the amount of times the entire task is run in a single session (on a single day). This is required for our naming format. |
| subj.version = 3; | Identifier of the script version that was used, change this when you implement major changes |
| subj.sessionID = '1'; | Only one session in TUE008.  Otherwise: input('Session ID: ','s') |
| control\_joystick = 1; | Controller vs. mouse input could be chosen, but for TUE008 set controller always to 1 |
| debug = 0; | Set to 1 if you are debugging script.  It will define the screen type:   * settings.do\_fullscreen 🡪 0 gives a small screen useful for debugging or testing. 1 gives a full screen, which is automatically the second monitor (if one is connected) |

## *Part 2: Load stimuli images*

The snack pictures are read and stored in a specific order in a struct.

% Read image files

img.d = imread([img\_dir '184.jpg']); %pretzels

img.f = imread([img\_dir '286.jpg']); % nic nocs

img.c = imread([img\_dir '26.jpg']); %cookies

img.e = imread([img\_dir '89.jpg']); %raisins

img.a = imread([img\_dir '40.jpg']); %strawberry gummy bears

img.b = imread([img\_dir '373.jpg']); % bread rings

img.g = imread([img\_dir '217.jpg']); %rice cracker

The order of the images can be modified by changing manually the letter order. In the case that the order is altered, the name assigned to each image has to be changed, too.

% Selection of the image order

order = {'d','f','b','g','e','c','a'};

order\_names = {'pretzels','nic nocs','bread rings','rice cracker', 'raisins', 'cookies', 'gummy bears'};

% Storing images in a struct following the specified order

for pic = 1:7

all\_img{pic} = img.(order{pic});

end

## *Part 3: Paradigm settings*

This part of the code loads the jitters and specifies the following variables (in order) that can be customized:

|  |  |
| --- | --- |
| color\_scale\_background = [255 255 255]; | White |
| color\_scale\_anchors = [0 0 0]; | Black |
| screen\_offset\_y = 0.01; | Positive values move the screen towards to top, negative towards the bottom |
| scale\_offset\_y = 0.25; |  |
| min\_ISI = 0.1; |  |
| max\_waiting\_snack = 20; | In seconds. When the subject is trying the snack in the second phase, after this amount of seconds the subject is summoned to continue with ratings |
| max\_waiting\_rating = 5; | In seconds |
| VAS\_rating\_duration = 20; | In seconds |

## *Part 4: Language, texts, instructions, and repetitions*

In this section, the size of the texts and the fixation cross is specified and the supplementary file *lang.mat* is loaded, which contains all the instructions and texts that appear in the experiment both in German and English.

The number of repetitions of the consumption phase is determined. Currently, it is set to 2 repetitions. To change number of times the consumption phase should be shown to the participant, change the value of **consumption\_repetition** to number of desired repetitions. The matrix **lang.phase** loaded in **lang.mat** should be updated manually to run the experiment with the correct number of phases.

## *Part 5: Screen*

This part of the code sets all the screen settings and defines the relative location of the images and texts on the screen.

## *Part 6: Initialization before experiment loop*

In this part, some variables are initialized before the experiment starts. The textures from the stimuli images are made here before the start of the loop and stored in a struct.

## *Part 7: Start of the experiment and timing*

The starting time of the experiment is stored.

## *Part 8: Experimental loop*

**Outer loop: setting the phase**

Currently, the experimental loop consists of 5 iterations, corresponding to the five phases of the experiment: 1. Willignness to pay I, 2. Antipication, 3. Consumption I, 4. Consumption II, 5. Willingness to pay II. In the first part of the outer loop, an instructions screen is shown including the current phase number and title, the instructions, and a text indicating how to continue to the next screen (either pressing button A with the gamepad, or with a mouse click). Then, for every phase, there is an inner loop with as many iterations as there are snacks.

**Inner loop: setting the snack item**

In each phase all the snacks are presented on the screen in the previously specified order.

**Repetition loop**

|  |  |
| --- | --- |
| **Bidding phases: I and V**  **(1 and 6 in the output)** | * + - ‘willingnesspay’ type scale is displayed with stimulus image on the screen using the Effort\_VAS function.     - Snack label, rating value, controller position, rating time, and submission are stored as output.     - Fixation cross. Time that passed between experiment onset and fixation cross onset is saved. |
| **Phases II – IV**  **(2 to 5 in the output)** | **Question loop** with 6 iterations:   * + - Text to summon to go to ratings is displayed if time set to evaluate the snack is over.     - Fixation cross 2. Time that passed between experiment onset and fixation cross onset is saved.     - Selection of question type: wanting, liking, taste (intensity, sweetness, saltiness, and umami)     - Stimulus image is displayed on the screen together with the corresponfing scale type defined by the question type using Effort\_VAS function.     - Output values are stored (rating value, label, submission, timing, scale label, snack label)     - Save temporary data in backup file     - An intertrial screen is shown between the snacks telling the participant to focus on the corresponding snack.     - Question loopincludes a screen indicating to take a sip of water between snacks |
| **Anticipation phase 2** | Another screen is shown to the participant to emphasize how they must take some time to pay attention to the snack’s appearance and smell. |
| **Consumption phase 3** | Screen indicating to try the snack is displayed  For the first consumption phase, the number of iterations of the repetition loop is doubled so that the questions are asked twice. |
| **Consumption phase 4** | Screen indicating to try the snack is displayed |

## *Part 9: Winning trial*

Announces result of willingness to pay. Takes the bid of an item chosen randomly from the second bidding phase and computes a probability to win using a sigmoid function. Then, compares that probability with a random one and displays on the screen if the item is won or not together with the snack image.

## *Part 10: End of experiment*

* Shows screen with end text
* Shows cursor
* Closes screen

## *Part 11: Save experiment data*

* Saves experiment length
* Saves Data and Backup file

## *Scales and ratings*

All the scales in this task are implemented from the Effort\_VAS file.

|  |  |
| --- | --- |
| **Bidding phases** | |
| **Scale name** | **Scale range** |
| ‘willingnesspay’ | 0 – 200 (displayed as 0 – 2 Euro) |

|  |  |
| --- | --- |
| **Rating phases** | |
| **Scale name** | **Scale range** |
| ‘wanting’ | 0 - 100 |
| ‘liking’ | -100 - 100 |
| ‘taste’ | 0 - 100 |

# **Output description**

The most important data structure is **Data.mat**. It contains two subsections, **output** and **subject information.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Output** | **Bidding** | **Phase** | |
| **Snack label** | |
| **Value** | |
| **Label** | |
| **Controller position** | |
| **Timing** | |
| **Submission** | |
| **Chosen item in winning trial (win\_item\_idx)** | |
| **Chosen item’s second bid value (win\_bid)** | |
| **Reward (reward\_won)** | |
| **Rating** | **Phase** | |
| **Value** | |
| **Label** | |
| **Submission** | |
| **Timing** | |
| **Scale label** | |
| **Snack label** | |
| **Subject information**  **(subj.)** | **Study** | | |
| **Run** | | |
| **Start date** | | |
| **Date** | | |
| **Version** | | |
| **Subject ID:** both in integer (id) and string format (subjectID) | | |
| **Session ID:** both in integer (sess) and string format (sessionID) | | |
| **Time** | **Subject starting the experiment with a click (exp\_on)** | |
| **Trigger** | **Experimental loop onset (fin)** | |
| **Onsets** | **First fixation cross (fix1)** | |
| **Scales** | **Bidding (bid)** |
| **Rating (rate)** |
| **Second fixation cross (fix2)** | |
| **Durations** | **Scales** | **Bidding (bid)** |
| **Rating (rate)** |
| **End date** | | |
| **Experiment length** | | |

# **Supporting files**

## *Joystick*

If a gamepad is used as a controller, the file **JoystickSpecification.mat** is required.

## *Folder structure*

The Taste Test requires the following sub-folders to function:

* A folder named **Backup** in which temporary and backup files are stored during the experiment.
* A folder named **Data** in which the final data files are stored.
* A folder named **SnackPics** which contains all the snack pictures used in the experiment.

## *Instructions*

The instructions’ structure is called **lang.mat** and needs to be placed in the same folder as TUE008\_Show\_TasteTest.m to be loaded.

## *Jitters*

**All jitters** need to be placed in the same folder as TUE008\_Show\_TasteTest.m to be loaded. They **do not** go into a separate folder.

To create new jitters, use the function **ComputeJitter\_exp.m**.

## *VAS*

The file **Effort\_VAS.m** is required to display the rating scales.

# **References**