**Experiment: Deel 2 (LDT & PCT)**

***Dingen in blauw mag je gewoon kopiëren van de pre-test***

***Dingen in rood zijn in deze versie erbij gekomen na overleg met promotoren (12/02/2021)***

* Van dit experiment zouden er **4 versies** moeten zijn (je kunt dat waarschijnlijk instellen dat de databank weet welke versie de laatste was die werd afgelegd). Er zijn ook **2 aparte taken** die direct op elkaar zouden moeten volgen, met de instructies voor de tweede test na afronding van de eerste taak. De combinaties van de verschillende plugins zijn:
  + *MAP\_LDT\_Stimuli\_v1.js* EN *MAP\_PCT\_Stimuli\_v1.js*
  + *MAP\_LDT\_Stimuli\_v1.js* EN *MAP\_PCT\_Stimuli\_v2.js*
  + *MAP\_LDT\_Stimuli\_v2.js* EN *MAP\_PCT\_Stimuli\_v1.js*
  + *MAP\_LDT\_Stimuli\_v2.js* EN *MAP\_PCT\_Stimuli\_v2.js*
* Experiment zou moeten starten met hetzelfde pop-up scherm als bij de pre-test (cf. *index\_html*)
* ~~Opnieuw informed consent (cf. informed\_consent.html):~~ dit hoeft niet meer, omdat ze al een document ondertekenen
* Audio\_test.html
* De instructies voor de eerste taak zijn de volgende:

var instructions = {

type: 'instructions',

pages: [

"<p>Druk op <b>'next'</b> om door te gaan naar de instructies.</p>",

'<div style="max-width:675px;"><p>In het eerste deel van het experiment zul je een reeks zinnetjes horen waarbij telkens één woord verschilt. Het is de bedoeling dat je aangeeft of dat woord als dan niet bestaat in het Nederlands.'+

"<p>Hoor je bijvoorbeeld het woord <b>appel</b> zoals in <i>Ze heeft <u>appel</u> gezegd.</i>, druk dan op de <b>J-toets</b></p>"+

"<p>Hoor je bijvoorbeeld het niet-woord <b>krasp</b> zoals in <i>Ze heeft <u>krasp</u> gezegd.</i>, druk dan op de <b>F-toets</b></p>",

'<p>Ter herinnering zal telkens herhaald worden welke toets met welk soort woord geassocieerd is.</p>'+

'<p>Merk op: niet alleen de <b>snelheid</b> maar ook de <b>nauwkeurigheid</b> van je antwoord is van belang.</p>'+

'<p>Dit eerste deel duurt ongeveer 15 minuten, pauze meegerekend.</p>',

'<p>Vooraleer het experiment te starten, kun je een aantal oefenvoorbeelden maken.</p>'+

'<p>Het is de bedoeling dat je aangeeft of het woord een bestaand of niet-bestaand woord is door op de corresponderende toets te drukken.</p>'+

'<p>Na de oefenfase start het experiment.</p>',

],

show\_clickable\_nav: true,

};

* Dus voor de eerste taak moeten de participanten drie *practice trials* (zie hieronder) maken die voor **alle 4 de versies gelijk** zijn:
  + Enkel op deze drie trials moeten ze ook feedback krijgen, ‘correct’ of ‘incorrect’ (deze code zal misschien helpen, maar je zal wel nog de drie trials [volgende pagina] op een of andere manier hieraan moeten toevoegen/koppelen):

var fixation = {

type: 'html-keyboard-response',

stimulus: '<span style="font-size:40px;">+</span>',

choices: jsPsych.NO\_KEYS,

trial\_duration: 1000

}

var lexical\_decision\_trial = {

type: 'html-keyboard-response',

stimulus: jsPsych.timelineVariable('stimulus'),

choices: ['f','j'],

stimulus\_duration: 3000,

data: {

condition: jsPsych.timelineVariable('condition'),

correct\_response: jsPsych.timelineVariable('correct\_response')

},

post\_trial\_gap: 1000,

on\_finish: function(data) {

var acc = false;

if (data.correct\_response == jsPsych.pluginAPI.convertKeyCodeToKeyCharacter(data.key\_press)) {

acc = true;

}

data.accuracy = acc;

}

};

var feedback = {

type: 'html-keyboard-response',

stimulus: function() {

var feedback\_text = '<span style="font-size:30px;color:red;">Incorrect</span>';

var last\_trial\_accuracy = jsPsych.data.getLastTrialData().values()[0].accuracy;

if (last\_trial\_accuracy == true) {

feedback\_text = '<span style="font-size:30px;color:red;">Correct!</span>'

}

return feedback\_text;

},

choices: jsPsych.NO\_KEYS,

trial\_duration: 3000

}

* + De *practice trials* zouden toch audio files moeten zijn in plaats van geschreven stimuli [zie hieronder]
  + Van deze drie trials kun je dan waarschijnlijk wel een aparte plugin maken met deze info (dit is vernieuwd):

var LDT\_practice\_trial = [

{

target: "droof",

probe: "Niet-woord <b>(F)</b><font color = 'ffffff'>dolores</font>Woord <b>(J)</b>", //droof

correct\_response: "f",

target\_audio: "js/LDT\_Audio/PRAC\_non-word\_DROOF.mp3",

condition: "prac\_non-word",

task\_part: "LDT\_prac"

},

{

target: "hamer",

probe: "Niet-woord <b>(F)</b><font color = 'ffffff'>dolores</font>Woord <b>(J)</b>", //hamer

correct\_response: "j",

target\_audio: "js/LDT\_Audio/PRAC\_word\_HAMER.mp3",

condition: "prac\_word",

task\_part: "LDT\_prac"

},

{

target: "blamps",

probe: "Niet-woord <b>(F)</b><font color = 'ffffff'>dolores</font>Woord <b>(J)</b>", //blamps

correct\_response: "f",

target\_audio: "js/LDT\_Audio/PRAC\_non-word\_BLAMPS.mp3",

condition: "prac\_non-word",

task\_part: "LDT\_prac"

}

];

* + Na afronden van deze oefentrials zou er dan gewoon dit moeten komen:

var start\_task = {

type: 'html-button-response',

stimulus: "De oefenfase is nu afgerond. Vanaf nu zul je geen feedback meer krijgen. Klik op <b>'Start'</b> om het experiment te starten. ",

choices: ['Start'],

response\_ends\_trial: true,

post\_trial\_gap: 1000

}

timeline.push(start\_task);

* Na het bovenstaande scherm start het echte experiment (deel 1) net zoals in experiment 1, dus je kunt weer dezelfde drie variabelen gebruiken (fixation, target, probe). De plugins voor deze taak zijn ***MAP\_LDT\_Stimuli\_v1.js***of ***MAP\_LDT\_Stimuli\_v2.js***(je zal waarschijnlijk wel nog *working directory* moeten aanpassen). De enige verschillen met het vorige experiment zijn dat:
  + ze hier moeten kiezen tussen **woord** en **niet-woord;**
  + er na **40** trials ipv. 50 een pauze is;
  + de eerste 5 trials **geen** **PSEUDO\_i\_X.mp3** of **PSEUDO\_ie\_X.mp3** zijn (dus diegene met condition: **amb\_crit\_word\_i(e)**);
  + er in plaats van twee *bink\_1* en *bink\_3* niet meer dan 2 trials dezelfde ***condition*** mogen hebben (dus max. twee fillers/non-words/amb\_crit\_word\_i(e) na elkaar).

Voor de rest blijft alles gelijk bij eerste taak.

* Daarna komen de instructies voor de **tweede taak,** die hetzelfde zijn als de vorige keer:

var instructions = {

type: 'instructions',

pages: [

"<p>Druk op <b>'next'</b> om door te gaan naar de instructies voor het tweede deel van het experiment.</p>",

'<div style="max-width:675px;"><p>In wat volgt zul je een reeks woorden horen met de ‘i’-klank. Het is de bedoeling dat je aangeeft welke ‘i’-klank je hoort, zonder daarbij rekening te houden of het woord al dan niet bestaat in het Nederlands.'+

"<p>Hoor je een 'i' zoals in <i>k<u>i</u>nd</i>, <i>beg<u>i</u>n</i>, <i>pr<u>i</u>ns</i> of <i>c<u>i</u>rkel</i>, druk dan op de <b>F-toets</b></p>"+

"<p>Hoor je een 'ie' zoals in <i>koff<u>ie</u></i>, <i>p<u>i</u>ano</i>, <i>d<u>ie</u>p</i> of <i>mach<u>i</u>ne</i>, druk dan op de <b>J-toets</b>.</p></div>",

'<p>Ter herinnering zal telkens herhaald worden welke toets met welke klank geassocieerd is.</p>'+

'<p>Probeer telkens zo <b>snel</b> en zo <b>accuraat</b> mogelijk te antwoorden.</p>'+

'<p>Dit laatste onderdeel duurt ongeveer 10 minuten, pauze meegerekend.</p>'+

'<p>Veel succes!</p>'

],

show\_clickable\_nav: true,

};

* De plugins voor tweede taak zijn *MAP\_PCT\_Stimuli\_v1.js* of *MAP\_PCT\_Stimuli\_v2.js*. Je mag weer dezelfde drie variabelen gebruiken (fixation, target, probe). Ze moeten hier weer aangeven of ze een ‘i’ of ‘ie’ horen. De enige verschillen met het vorige experiment zijn dat:
  + er na **40** trials ipv. 50 een pauze is;
  + dat de plugins **acht keer** herhaald worden (pas nadat één plugin volledig afgerond is, dus 8 X 20 = 160 trials);
  + er geen *wit\_1* en *wit\_3* targets onmiddellijk op elkaar mogen volgen.
* ‘Bedankt om deel te nemen aan het experiment’-scherm
* De manier waarop de data de vorige keer werden opgeslagen was super goed, dus dat mag ook weer zo.
* Zou het misschien wel mogelijk zijn om de **volgorde** van de trails te kunnen **nummeren**, zodat ik weet of een bepaalde trial eerder in het begin of op het einde voorkwam
* Ik heb van een doctoraatsstudent een code gekregen om automatisch informatie over de browser enzo van de participant te verkrijgen. Dit zou er nu ook in moeten:

*Please find the browser plugin that I used (the developer somehow named it as bowser). You should change the suffix to .js when load it. And the code to initiate browser function is as follow. You can put it at the beginning of your script.*

// detect visitor variables with the bowser js library (/js/browser.js)

jsPsych.data.addProperties({ // add these variables to all rows of the datafile

browser\_name: bowser.name, browser\_version: bowser.version,

os\_name: bowser.osname, os\_version: bowser.osversion,

tablet: String(bowser.tablet), mobile: String(bowser.mobile),

// convert explicitly to string so that "undefined" (no response) does not lead to empty cells in the datafile

screen\_resolution: screen.width + ' x ' + screen.height,

window\_resolution: window.innerWidth + ' x ' + window.innerHeight, // this will be updated throughout the experiment

});

* Een tweede code die ik heb gekregen controleert of de participant bv. niet ondertussen met andere dingen bezig is tijdens het experiment. Blijkbaar zou je daar deze code voor kunnen gebruiken:

*For the code concerning with detecting focus, you should put code in the jsPsych.init function. This is my code:*

jsPsych.init({

...

on\_interaction\_data\_update: function(data) {

// get the main trial data

var trial = jsPsych.currentTrial();

trial.data.screen\_focus = data.event;

},

....

})

* Tot slot zou er een pop-upvenster/scherm moeten verschijnen met de tekst ‘Je antwoord werd niet geregistreerd.’ als het **langer dan 4000 ms** duurt vooraleer een participant een bepaalde toets indrukt om te antwoorden.

Versie op basis van vorige

Versie en taak bij de data

Taak 2 nummer van interatie

Na 4 seconden pagina met IN het vervolg sneller antwoorden in practice, test1 en 2