Programming MTurk experiments via *JSEXP*

Florian Jaeger with help from Linda Liu, Zach Burchill, Wednesday Bushong, and Xin Xie

October 22, 2021

Contents

1 Overview

Make sure you have completed the tutorials on *Setting up Mechanical Turk* and *Conducting Mechanical Turk experiments*. This tutorial walks you through how to program your own experiment using *JSEXP*, which is a collection of Javascript programs originally written by Dave Kleinschmidt and then extended and modified by Zach Burchill, Wednesday Bushong, Esteban Buz, Florian Jaeger, Linda Liu, Xin Xie, and others.

2 Disabling CORS errors when testing locally

Avoiding CORS Errors on Localhost (in 2020)

To disable web security for Chrome, for example:

```
open -n -a /Applications/Google Chrome.app/Contents/MacOS/Google Chrome --args
↪   --user-data-dir="/tmp/chrome_dev_test" --disable-web-security
```

3 Making your own experiment

3.1 What needs to be in the HTML file?

- A form element that posts the resuls to MTurk when the experiment is done.

```
<form id="mturk_form" method="POST"
↪  action="https://www.mturk.com/mturk/externalSubmit">
    <!-- some MTurk specific information -->
    <textArea id="assignmentId" name="assignmentId" ></textArea><br />
    <textArea id="practiceResp" name="practiceResp" ></textArea><br />
    <input type="hidden" id="userAgent" name="userAgent" />

    <!-- hidden fields for the URL parameters that we want to be stored. -->
    <input type="hidden" id="label" name="label" />
    <input type="hidden" id="condition" name="condition" />
    <input type="hidden" id="reverse" name="reverse" />
    <input type="hidden" id="reverse2" name="reverse2" />
    <input type="hidden" id="list_num" name="list_num" />
    <input id="submitButton" type="submit" name="Submit" value="Submit" />
</form>
```

- A link to whatever Javascript code you're using. For example, the example experiment in this tutorial uses js-adapt, a collection of code originally written by Dave Kleinschmidt, and then extended by Linda Liu, Zach Burchill, Wednesday Bushong, and Xin Xie. That code provides object definitions for experiments, different types of blocks, stimulus lists, etc. These definitions can then be evokes to create the specific experiment, like the last line in the following example, in which experiment-A.js would contain the code for the experiment (code that references code in the other .js files).

```
<!-- general javascript, incl. general definitions of objects
     for experiment, block, and stimulus lists  //-->
<script src="js-adapt/jquery-1.10.1.min.js" type="text/javascript" ></script>
<script src="js-adapt/modernizr.min.js" type="text/javascript"></script>
<script src="js-adapt/stimuli.js" type="text/javascript" ></script>
<script src="js-adapt/labelingBlock.js" type="text/javascript" ></script>
<script src="js-adapt/exposureBlock.js" type="text/javascript" ></script>
<script src="js-adapt/experimentControl2.js" type="text/javascript" ></script>
<script src="js-adapt/soundcheckBlock.js" type="text/javascript"></script>
<script src="js-adapt/mturk_helpers.js" type="text/javascript"></script>
<script src="js-adapt/progressBar.js" type="text/javascript"></script>
<script src="js-adapt/logreg.js"  type="text/javascript"></script>
<script src="js-adapt/utilities.js" type="text/javascript"></script>
<script src="get_stim.js" type="text/javascript"></script>

<!-- Here is where your experiment javascript file is specified -->
<script src="experiment-A.js" type="text/javascript"></script>
```

- Any HTML objects references in the Javascript code. This includes, for example, various ¡div¿s that the Javascript code will fill with content at different points during the experiment.

4 Example experiments using *JSEXP*

- The code for the unsupervised and supervised learning experiment in Kleinschmidt et. al (2015) [1] can be cloned from https://bitbucket.org/hlplab/nrtmodule/src/master/. If you'd like to get a sense of a full version of the experiment, you can test it out at https://www.hlp.rochester.edu/mturk/mtadapt/sup-unsup/.

[1] https://mindmodeling.org/cogsci2015/papers/0200/paper0200.pdf

- The code for a web-based version of the priming experiment from Xie & Myers (2017) can be found at https://github.com/xinxie-cogsci/Online-experiments. If you'd like to get a sense of a full version of the experiment, you can test it out at, for example, https://www.hlp.rochester.edu/mturk/xxie/io_perception/exp1/exp1_transcription.html?condition=experimental&speaker=M15&order=2&visual=1&block=2 (for information on the URL parameters, see the readme.md file in the github repository).
- And here is a demo of a whole number of different paradigms available on request: https://www.hlp.rochester.edu/mturk/lliu/demos/.