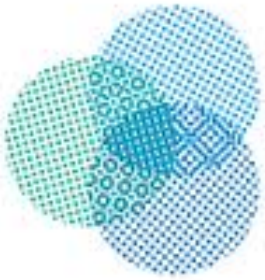


CENTER FOR
Brains
Minds+
Machines

Biological
Discovery
in Woods Hole

USING MECHANICAL TURK WITH PSITURK

CBMM Summer School

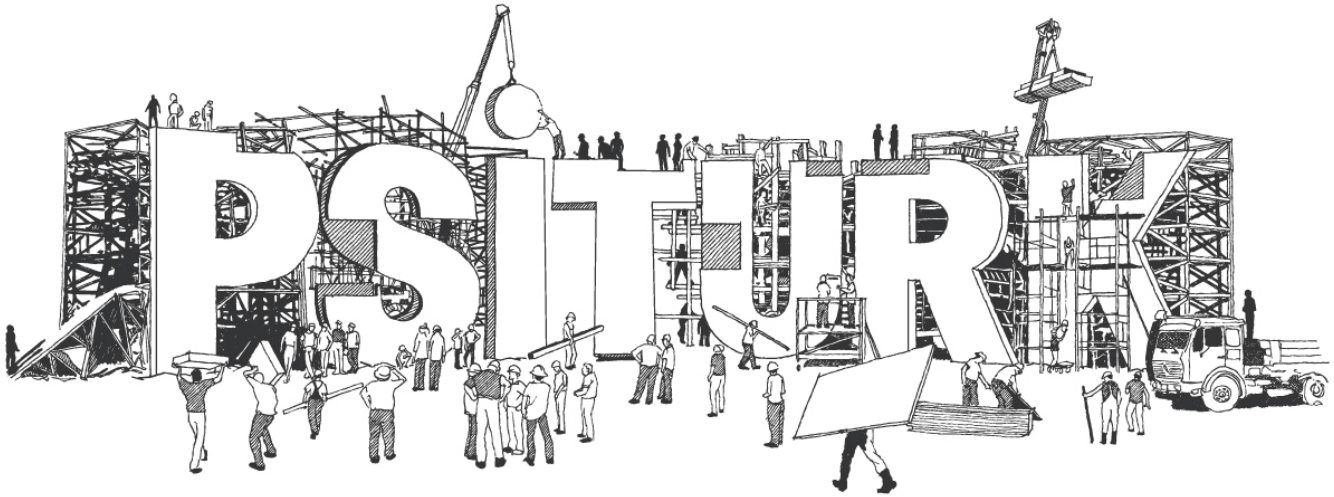
Aug 22, 2017

Roadmap

- What is psiTurk?
- Setting up psiTurk
- Running psiTurk
- Structure of a psiTurk project
- JavaScript psiTurk commands
- Extracting / dealing with data
- Exercises

WHAT IS PSITURK?

What is psiTurk?



- Wrapper around Mechanical Turk for psychological studies
- Handles database management, web serving, counterbalancing, worker / HIT management
- Allows you to run experiments from your own computer

https://psiturk.org/

Why use psiTurk?

Pros

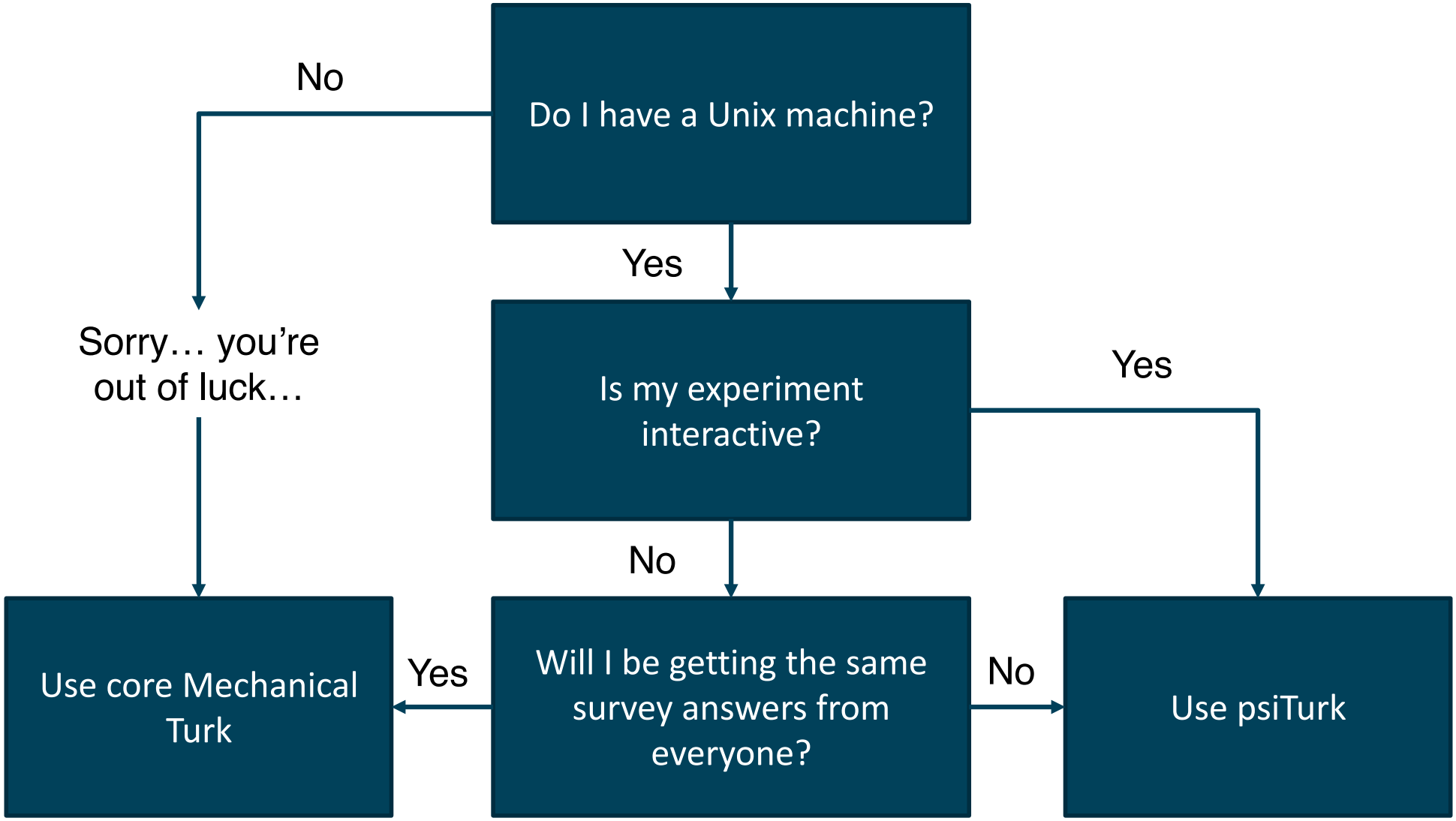
- Runs locally; don't have to worry about web backend (hosting / database)
- Better worker management (approvals, exclusions, etc.)
- Don't need to build web-based counterbalancing logic
- Easy interface with Mechanical Turk

Cons

- Some JavaScript / HTML knowledge & scripting required
- Only Unix-based (Mac / Linux)
- For local running, can require some configuration with your home / university network (e.g., at MBL)

https://psiturk.org/

Should I be using psiTurk?



SETTING UP PSITURK

Setting up psiTurk

- Make sure you have python (v2.x) and pip installed (see https://pip.pypa.io/en/stable/installing/)
  - If on a Mac, install C compiler (e.g., Xcode from AppStore)

- In a terminal, run "`pip install psiturk`"

- Optional (needed for posting experiments, not tutorial):
  - Get a Mechanical Turk account
  - Get a psiTurk account

- For more details: http://psiturk.readthedocs.io/en/latest/install.html

Getting a Mechanical Turk account

- Make an Amazon Web Services account @ https://aws.amazon.com/ (NOTE: requires a credit card & phone number)

- Get account keys:
  - Go to the AWS Management Console @ https://console.aws.amazon.com/iam/home?#security_credential
  - Open the Access Keys tab
  - Click the "Create New Access Key" button
  - Open the "`.psiturkconfig`" file in your home directory (may be hidden, or you may need to create it)
  - Copy those keys into the relevant "`.psiturkconfig`" fields

- Make Mechanical Turk Requester account
  - Go to https://requester.mturk.com/
  - Click on "Create an Account" and enter the same email you used for AWS
  - Go to https://requester.mturk.com/developer, click "Link your AWS Account" and log in with your AWS credentials
  - Repeat those steps on the Sandbox at https://requestersandbox.mturk.com/

- For more details: http://psiturk.readthedocs.io/en/latest/amt_setup.html

Getting a psiTurk account

- Make a psiTurk account @ https://psiturk.org/register
  (note: requires lead time to get an invite)

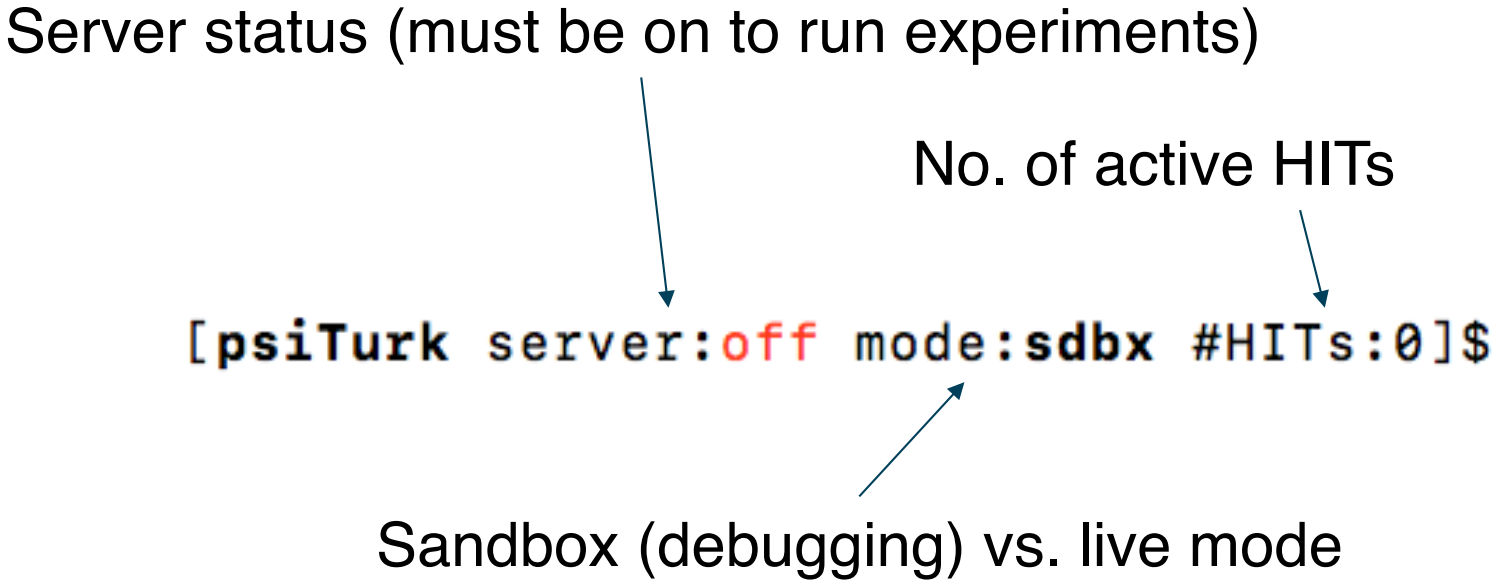
- Get account keys:
  - Log in to the psiTurk account page @ https://psiturk.org/login
  - Select the dropdown menu under your email address (top right side), and select “API keys”
  - Copy those keys into the relevant “`.psiturkconfig`” fields

- For more details:
  http://psiturk.readthedocs.io/en/latest/psiturk_org_setup.html

RUNNING PSITURK

Running psiTurk

- Type ‘psiturk’ in the terminal once you have entered the cbmm-psiturk-exercise folder



Trying out your experiment

- Start up the server

```
> server on
```

- Post a debug link / start in browser

```
> debug
```

Try this out (but don’t do the experiment just yet)!

Posting your experiment (sandbox)

- Create a HIT

> hit create <number> <payment> <time limit (hrs)>

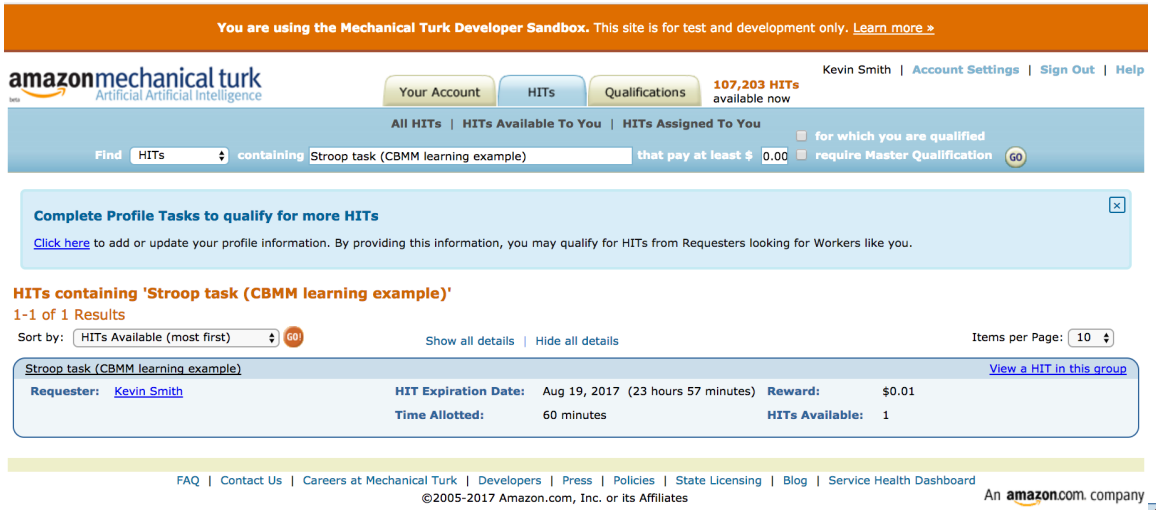
```
[******************************
   Creating sandbox HIT
     HITid: 3IYI9285WSKI3P26ILJH3DBYC5ZCJL
     Max workers: 5
     Reward: $1.00
     Duration: 1.0 hours
     Fee: $1.00

     ------------------------
     Total: $6.00
   Ad URL: https://sandbox.ad.psiturk.org/view/dPqhRB4N8cMqdjYZajU3vC?assignmentId=debugQHA343&hitId=debugCAT7RM&workerId=debug29J7H0
Note: This url cannot be used to run your full psiTurk experiment.  It is only for testing your ad.
   MTurk URL: https://workersandbox.mturk.com/mturk/searchbar?selectedSearchType=hitgroups&searchWords=Stroop+task+%28CBMM+learning+example%29
Hint: In OSX, you can open a terminal link using cmd + click
Note: This sandboxed ad will expire from the server in 16 days.
```

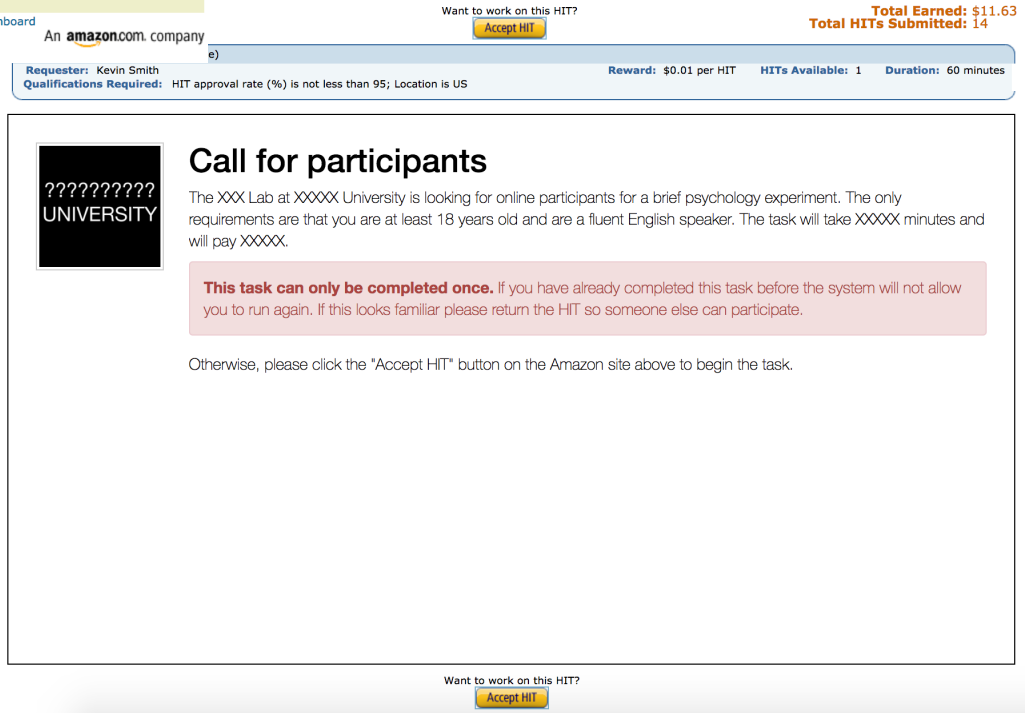
- Show your HITs

> hit list

```
Stroop task (CBMM learning example)
        Status: Assignable
        HITid: 3IYI9285WSKI3P26ILJH3DBYC5ZCJL
        max:5/pending:0/complete:0/remain:5
        Created:2017-08-16T18:45:14Z
        Expires:2017-08-17T18:45:14Z
```

Appearance on the Sandbox





Live-posting your experiment

- First, switch to live mode and (re)start the server:

```
[psiTurk server:off mode:sdbx #HITs:0]$
```

```
> mode
> server on
```

```
[psiTurk server:on mode:live #HITs:0]$
```

- Then any HIT you make will be posted to the live server

```
> hit create 5 1.00 1
```

Worker management

- Terminology:
  - Worker ID: a unique identifier for each worker
  - HIT ID: a unique identifier for each HIT
  - Assignment ID: an identifier for one worker doing one HIT (used for approving / bonuses)

- Find the workers who performed your task

```
> worker list --hit  <HIT ID>
```

- NOTE: you need to be in ‘live’ mode to review live HITs

Approving (paying) workers

- By individual

```
> worker approve <Assignment ID>
```

- For everyone in a HIT

```
> worker approve --hit <HIT ID>
```

- Providing a bonus

```
> worker bonus --amount <$$$> <Assignment ID>
```

Getting your data

- Download it from the database

```
> download_datafiles
```

- Makes the following files:
  - trialdata.csv: structured trial data
  - questiondata.csv: unstructured event data
  - eventdata.csv: captures users' browser events (resizing windows, bringing others to front, etc.)

Final useful function

- See the money you have in your MTurk account

```
> amt_balance
```

- Note: only works in live mode (shows $10k in sandbox)

- Get out of psiTurk

```
> exit
```

STRUCTURE OF A PSITURK PROJECT

Structure of a psiTurk project

- **config.txt** – configuration for:
  - Experiment metadata for MTurk website
  - Restrictions on workers (location / browser)
  - Number of conditions / counterbalancing
  - Information on server / database
- **templates** – stores HTML files for experiment
- **static** – stores other files (JavaScript, images, css, etc.)
- **Other** (advanced):
  - participants.db
  - config.py
  - server.log

Gureckis et. al., 2015

Configuration file

config.txt

```
[HIT Configuration]
title = Stroop task (CBMM learning example)
description = Judge the color of a series of words.
amt_keywords = Perception, Psychology
lifetime = 24
us_only = true
approve_requirement = 95
contact_email_on_error = youremail@gmail.com
ad_group = Default psiTurk Stroop Example
psiturk_keywords = stroop, learning
organization_name = CBMM Summer School
browser_exclude_rule = MSIE, mobile, tablet

[Database Parameters]
database_url = sqlite:///participants.db
table_name = cbmmdemo

[Server Parameters]
host = localhost
port = 22362
cutoff_time = 30
logfile = server.log
loglevel = 2
debug = true
login_username = examplename
login_pw = examplepassword
threads = auto
secret_key = 'this is my secret key which is hard to guess,
i should change this'
#certfile = <path_to.crt>
#keyfile = <path_to.key>

[Task Parameters]
experiment_code_version = 1.0
num_conds = 1
num_counters = 1

[Shell Parameters]
launch_in_sandbox_mode = true

# If you are not using the psiturk ad server, set
`use_psiturk_ad_server` to `false` and point `ad_location`
to your proxy server <host> and <port>. Format the
ad_location like this:
#
#   https://<host>:<port>/ad

use_psiturk_ad_server = true
ad_location = false
```

Title, description, contact info & exclusion criteria

Database info (no need to change by default, except table name)

Server info (only need to change host to “0.0.0.0” by default)

Condition / counterbalance info

Linking info (no need to change by default)

http://psiturk.readthedocs.io/en/latest/configuration.html#local-configuration-file

Important HTML files

- ad.html: The first thing Workers will see when they click on your experiment in Mechanical Turk (not seen in debug)
- consent.html: The consent agreement – this is determined by your IRB
- instructions/*: Individual pages for text / images to introduce the experiment
- exp.html / stage.html: The pages that contains the HTML your code will work on and loads the JavaScript (including psiTurk)
- postquestionnaire.html: Instructions after the experiment for getting feedback (ratings, forms, etc.)

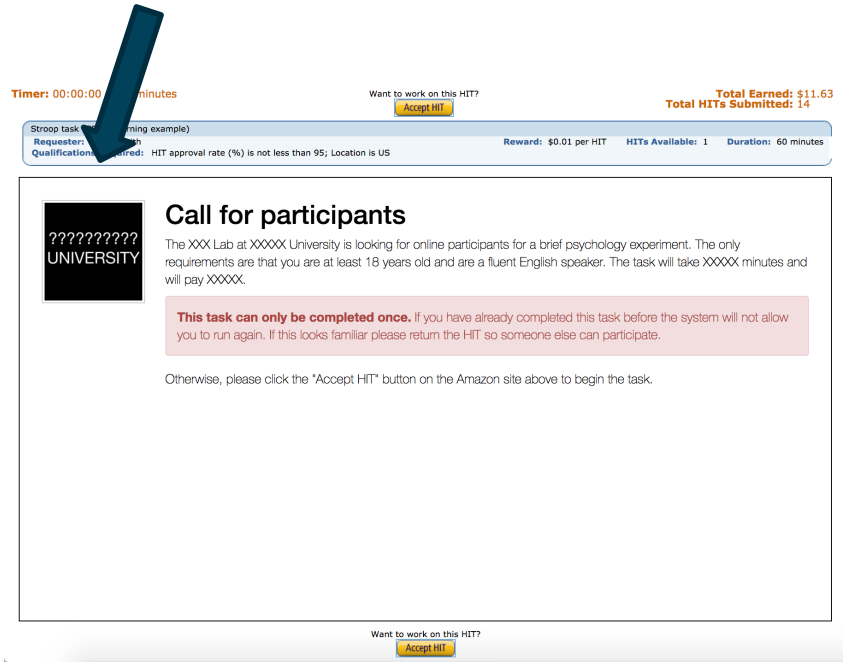
http://psiturk.readthedocs.io/en/latest/file_desc/templates_folder.html

ad.html

```
        </style>
    </head>
    <body>
        <div id="container-not-an-ad" class="media">
            <div class="media-left pull-left" href="#">
                        <!-- REPLACE THE LOGO HERE WITH YOUR  UNIVERSITY, LAB, or COMPANY -->
                        <img id="adlogo" src="{{ server_location }}/static/images/university.png" alt="Lab Logo" />
            </div>
            <div class="media-body">

                            <!--
                                If assignmentid is "ASSIGNMENT_ID_NOT_AVAILABLE"
                                it means the participant has NOT accepted your hit.
                                This should display the typical advertisement about
                                your experiment: who can participate, what the
                                payment is, the time, etc...

                            -->
                            {% if assignmentid == "ASSIGNMENT_ID_NOT_AVAILABLE" %}

                                <h1>Call for participants</h1>
                                <p>
                                    The XXX Lab at XXXXX University is looking for online participants
                                    for a brief psychology experiment. The only requirements
                                    are that you are at least 18 years old and are a fluent English
                                    speaker.  The task will take XXXXX minutes and will pay XXXXX.
                                </p>
                                <div class="alert alert-danger">
                                    <strong>This task can only be completed once.</strong>
                                    If you have already completed this task before the system will not
                                    allow you to run again. If this looks familiar please return the
                                    HIT so someone else can participate.
                                </div>
                                <p>
                                    Otherwise, please click the "Accept HIT" button on the Amazon site
                                    above to begin the task.
                                </p>


                            {% else %}

                                <!--
                                    OTHERWISE
                                    If assignmentid is NOT "ASSIGNMENT_ID_NOT_AVAILABLE"
                                    it means the participant has accepted your hit.
                                    You should thus show them instructions to begin the
                                    experiment ... usually a button to launch a new browser
                                    window pointed at your server.

                                    It is important you do not change the code for the
                                    openwindow() function below if you want you experiment
                                    to work.
                                -->
                                <h1>Thank you for accepting this HIT!</h1>
                                <p>
                                    By clicking the following URL link, you will be taken to the experiment,
                                    including complete instructions and an informed consent agreement.
                                </p>
                                <script type="text/javascript">
                                    function openwindow() {
                                        popup = window.open('{{ server_location }}/consent?hitId={{ hitid }}&assignmentId=
                                    }
                                </script>
                                <div class="alert alert-warning">
                                    <b>Warning</b>: Please disable pop-up blockers before continuing.
                                </div>
```

Edit this part

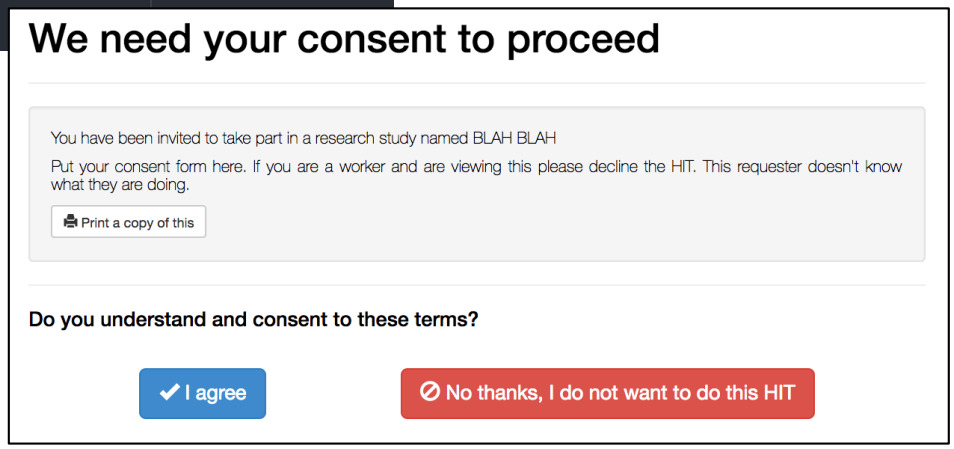
Change university.png in static/images for this picture



consent.html

```
<html>
    <head>
        <meta charset="utf-8" />
        <title>Psychology Experiment - Informed Consent Form</title>
        <link rel="stylesheet" href="/static/css/bootstrap.min.css" type="text/css" />
        <link rel="stylesheet" href="/static/css/style.css" type="text/css" />
        <script type="text/javascript">
            function onexit() {
              self.close(); // no harm, no foul here
            }
        </script>
    </head>
    <body>
        <div id="container-consent">
            <div id="consent">
                <h1>We need your consent to proceed</h1>
                <hr />
                <div class="legal well">
                    <p>
                    You have been invited to take part in a research study named BLAH BLAH
                    </p>
                    <p>
                    Put your consent form here.  If you are a worker and are viewing this
                    please decline the HIT.  This requester doesn't know what they are doing.
                    </p>

                    <button type="button" class="btn btn-default btn-sm" onClick="window.print();">
                    <span class="glyphicon glyphicon-print"></span> Print a copy of this
                    </button>
                </div>

                <hr />
                <h4>Do you understand and consent to these terms?</h4>
                <br />

                <center>
                    <button type="button" class="btn btn-primary btn-lg" onClick="window.location='/exp?hitId={{ hit
                    <span class="glyphicon glyphicon-ok"></span> I agree
                    </button>
                           
                           
                           
                    <button type="button" class="btn btn-danger btn-lg" onClick="onexit()">
                    <span class="glyphicon glyphicon-ban-circle"></span> No thanks, I do not want to do this HIT
                    </button>
                </center>

            </div>
        </div>
    </body>
</html>
```

Edit this part



Instructions

```
<div id="container-instructions">

    <h1>Instructions</h1>

    <hr />

    <div class="instructions well">

        <p>
            This is the first page of instructions.
        </p>

        <p>
            Be helpful, clear.
        </p>

        <p>
            Use images and concrete examples.
        </p>

        <script type="text/javascript">
        //console.log(psiTurk.getInstructionIndicator())
        </script>

    </div>

    <hr />


    <div class="instructionsnav">
        <div class="row">
            <div class="col-xs-2">
                <!-- no previous button on first screen -->
            </div>
            <div class="col-xs-8">
            </div>
            <div class="col-xs-2">
                <button type="button" id="next" value="next" class="btn btn-pri
                Next <span class="glyphicon glyphicon-arrow-right"></span>
                </button>
            </div>
        </div>
    </div>

</div>
```

Edit this part

Plus in Javascript (task.js file), set the pages to use:

```
var instructionPages = [ // add as a list as many pages as you like
    "instructions/instruct-1.html",
    "instructions/instruct-2.html",
    "instructions/instruct-3.html",
    "instructions/instruct-ready.html"
];
```

```
$(window).load( function(){
    psiTurk.doInstructions(
        instructionPages, // a list of pages you want to display in sequence
        function() { currentview = new StroopExperiment(); } // what you want
    );
});
```

postquestionnaire.html

```
<div id="container-questionnaire">
    <h1>Task Complete</h1>

    <hr />

    <p>You are finished!  Thank you for your contributions to science. You will be eligible for full payme

    <div class="instructions well">

        <form id="postquiz" action="debrief" method="post">


            <!-- beginning of a question -->
            <div class="row question">
                <div class="col-md-8">
                    On a scale of 1-10 (where 10 is the most engaged), please rate how <b>ENGAGING</b> you
                </div>
                <div class="col-md-4">
                    <select id="engagement" name="engagement">
                        <option value="10">10 - Very engaging</option>
                        <option value="9">9</option>
                        <option value="8">8</option>
                        <option value="7">7</option>
                        <option value="6">6</option>
                        <option value="5" SELECTED>5 - Moderately</option>
                        <option value="4">4</option>
                        <option value="3">3</option>
                        <option value="2">2</option>
                        <option value="1">1</option>
                        <option value="0">0 - Not engaged</option>
                    </select>
                </div>
            </div>
            <!-- end of a question -->


            <!-- beginning of a question -->
            <div class="row question">
                <div class="col-md-8">
                    On a scale of 1-10 (where 10 is the most difficult), please rate how <b>DIFFICULT</b>
                </div>
                <div class="col-md-4">
                    <select id="difficulty" name="difficulty">
                        <option value="10">10 - Very difficult</option>
                        <option value="9">9</option>
                        <option value="8">8</option>
                        <option value="7">7</option>
                        <option value="6">6</option>
                        <option value="5" SELECTED>5 - Moderately difficult</option>
                        <option value="4">4</option>
                        <option value="3">3</option>
                        <option value="2">2</option>
                        <option value="1">1</option>
                        <option value="0">0 - Not difficult at all</option>
                    </select>
                </div>
            </div>
            <!-- end of a question -->
        </form>
    </div>

    <hr />

    <div class="instructionsnav">
        <div class="row">
            <div class="col-xs-2">
            </div>
            <div class="col-xs-7">
            </div>
            <div class="col-xs-3">
                <button type="button" id="next" value="next" class="btn bt
                    Continue <span class="glyphicon glyphicon-arrow-right"
                </button>
            </div>
        </div>
    </div>


</div>
```

Change these questions / options as needed – any ‘select’ or ‘textarea’ element will have its responses recorded as unstructured data if you use the default Questionnaire code

Task Complete

You are finished! Thank you for your contributions to science. You will be eligible for full payment once you answer the following questions.

On a scale of 1-10 (where 10 is the most engaged), please rate how **ENGAGING** you found the learning task: 5 - Moderately

On a scale of 1-10 (where 10 is the most difficult), please rate how **DIFFICULT** you found the learning task: 5 - Moderately difficult

Continue ➡

JAVASCRIPT PSITURK COMMANDS

Flow of Stroop task.js

1. Set up psiTurk and load data / stimuli (lines 8-73)

2. Run instructions then continue to the main experiment (lines 238-243)

3. Build a StroopExperiment object (lines 75-172) that:
   - Loads the ‘stage.html’
   - Runs through each of the stimuli on that stage and waits for keypress
   - Records data

4. Calls the Questionnaire, then finish the experiment (lines 179-230)

Important psiTurk.js functions

- Setup / shut-down:
  - `psiTurk = new PsiTurk(uniqueID, adServerLoc, mode)` (line 8)
  - `psiTurk.preloadPages([<LIST OF PAGES>])` (lines 14-24)
  - `psiTurk.completeHIT()` (line 222)

- Instructions:
  - `psiTurk.doInstructions([<INSTRUCTION PAGES>], callback)` (lines 237-239)

- Saving data:
  - `psiTurk.recordUnstructuredData(key, value)` (line 77)
  - `psiTurk.recordTrialData(data)` (lines 128-135)
  - `psiTurk.saveData(...)` (line 220)

http://psiturk.readthedocs.io/en/latest/api.html

psiTurk.js: Setup

- Setting up / shutting down:
  - Initialize psiTurk (lines 7-8)

```
7    // Initalize psiturk object
8    var psiTurk = new PsiTurk(uniqueId, adServerLoc, mode);
```

  - Preload pages that you will be using (lines 14-24)

```
14   // All pages to be loaded
15   var pages = [
16       "instructions/instruct-1.html",
17       "instructions/instruct-2.html",
18       "instructions/instruct-3.html",
19       "instructions/instruct-ready.html",
20       "stage.html",
21       "postquestionnaire.html"
22   ];
23
24   psiTurk.preloadPages(pages);
```

  - End and return to Mechanical Turk (line 223)

```
psiTurk.completeHIT();
```

http://psiturk.readthedocs.io/en/latest/api.html

psiTurk.js: Instructions

- Displaying instructions:
  - Tell psiTurk which pages to use (lines 26-31)

```
26    var instructionPages = [ // add as a list as many pages as you like
27        "instructions/instruct-1.html",
28        "instructions/instruct-2.html",
29        "instructions/instruct-3.html",
30        "instructions/instruct-ready.html"
31    ];
```

  - Run the instructions, then start the expt. (lines 238-43)

```
235   /********************
236    * Run Task
237    ********************/
238   $(window).load( function(){
239       psiTurk.doInstructions(
240           instructionPages, // a list of pages you want to display in sequence
241           function() { currentview = new StroopExperiment(); } // what you want
242       );
243   });
```

http://psiturk.readthedocs.io/en/latest/api.html

psiTurk.js: Saving data

- Saving data:
  - Recording unstructured data (line 77)

```
psiTurk.recordUnstructuredData("mode", mode);
```

	A	B	C
1	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	difficulty	10
2	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	engagement	0
3	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	mode	sandbox

  - Recording trial data (lines 128-135):

```
psiTurk.recordTrialData({'phase':"TEST",
                         'word':stim[0],
                         'color':stim[1],
                         'relation':stim[2],
                         'response':response,
                         'hit':hit,
                         'rt':rt}
                        );
```

A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	8	1.5031E+12	{"rt": 784, "hit": true, "color": "blue", "relation": "congruent", "phase": "TEST", "word": "BLUE", "response": "blue"}

  - Saving data to the database (lines 220-226):

```
psiTurk.saveData({
    success: function(){
        psiTurk.computeBonus('compute_bonus', function() {
            psiTurk.completeHIT(); // when finished saving
        });
    },
    error: prompt_resubmit});
```

http://psiturk.readthedocs.io/en/latest/api.html

Extra variables set by default

- **uniqueId:** The AssignmentID
- **condition:** The condition randomly assigned (maximum set in config.txt; zero-indexed)
- **counterbalance:** As above for counterbalancing
- **adServerLoc:** Passed to psiTurk object so it knows how to return to Mechanical Turk
- **mode:** String that is either “live”, “sandbox”, or “debug” depending on how the experiment is accessed

EXTRACTING / DEALING WITH DATA

Extracting your data

- trialdata.csv (use parse_trialdata.py for nicer version)

	A	B	C	D
1	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	0	1.5031E+12	{"phase": "INSTRUCTIONS", "templates": ["instructions/instruct-1.html", "instructions/instruct-2.html", "instructions/instruct-3.html", "instructions/instruct-ready.html"], "
2	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	1	1.5031E+12	{"phase": "INSTRUCTIONS", "indexOf": 0, "viewTime": 1181, "action": "NextPage", "template": "instructions/instruct-1.html"}
3	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	2	1.5031E+12	{"phase": "INSTRUCTIONS", "indexOf": 1, "viewTime": 545, "action": "NextPage", "template": "instructions/instruct-2.html"}
4	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	3	1.5031E+12	{"phase": "INSTRUCTIONS", "indexOf": 2, "viewTime": 448, "action": "NextPage", "template": "instructions/instruct-3.html"}
5	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	4	1.5031E+12	{"phase": "INSTRUCTIONS", "indexOf": 3, "viewTime": 719, "action": "FinishInstructions", "template": "instructions/instruct-ready.html"}
6	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	5	1.5031E+12	{"rt": 1358, "hit": true, "color": "green", "relation": "incongruent", "phase": "TEST", "word": "BLUE", "response": "green"}
7	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	6	1.5031E+12	{"rt": 1118, "hit": true, "color": "green", "relation": "congruent", "phase": "TEST", "word": "GREEN", "response": "green"}
8	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	7	1.5031E+12	{"rt": 855, "hit": true, "color": "blue", "relation": "incongruent", "phase": "TEST", "word": "RED", "response": "blue"}
9	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	8	1.5031E+12	{"rt": 784, "hit": true, "color": "blue", "relation": "congruent", "phase": "TEST", "word": "BLUE", "response": "blue"}
10	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	9	1.5031E+12	{"rt": 783, "hit": true, "color": "green", "relation": "unrelated", "phase": "TEST", "word": "MONKEY", "response": "green"}
11	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	10	1.5031E+12	{"rt": 696, "hit": true, "color": "red", "relation": "unrelated", "phase": "TEST", "word": "SHIP", "response": "red"}
12	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	11	1.5031E+12	{"rt": 774, "hit": true, "color": "red", "relation": "incongruent", "phase": "TEST", "word": "GREEN", "response": "red"}
13	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	12	1.5031E+12	{"rt": 744, "hit": true, "color": "red", "relation": "congruent", "phase": "TEST", "word": "RED", "response": "red"}
14	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	13	1.5031E+12	{"rt": 886, "hit": true, "color": "blue", "relation": "unrelated", "phase": "TEST", "word": "ZAMBONI", "response": "blue"}
15	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	14	1.5031E+12	{"phase": "postquestionnaire", "status": "begin"}
16	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	15	1.5031E+12	{"phase": "postquestionnaire", "status": "submit"}

- questiondata.csv

	A	B	C
1	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	difficulty	10
2	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	engagement	0
3	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	mode	sandbox

- eventdata.csv

	A	B	C	D	E
1	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	initialized	0		1.5031E+12
2	A1NF0ETG2CJCYZ:3Y5140Z9DY08PNPYVAPO0VQMA73PIO	window_resize	0	[1280, 850]	1.5031E+12

Tips for making psiTurk experiments

- Build project around psiTurk structure (consider jsPsych)
- Use psiTurk debug to iron out issues
- Test on multiple browsers
- Use the sandbox to ensure MTurk integration
- After all that, go live and get data fast!

EXERCISE

Exercise

You should be able to make the following changes to the sample code so that you can see the effects of the experiment in debug mode

1. Run through the experiment once in debug mode
2. Edit the ad & consent forms to display fake info & the CBMM logo
3. Make some quick, fake instructions. Use at most two instruction pages (hint: pay attention to task.js line 26)
4. There are two stimulus sets (task.js; lines 48-70). Edit the configuration and task.js (~line 81) to randomly choose one of these two sets. For extra credit, record the participant's condition as unstructured data
5. Download the data and analyze it. What was the average reaction time of people for the incongruent condition? (Hint: run `` `python parse_trialdata.py` `` to get your raw trial data into a more readable form)
6. Extra credit: add a free-form feedback response to the post-questionnaire