

RS1: Experiment Implementation

Due: 11:59 PM CST on Mar 8, 2018

Upload submission to Blackboard.

This problem set is only for the doctoral 422D version of the class.

This problem set is the first in a three-part series (RS1, RS2, RS3) in which you will reproduce a controlled experiment of a novel user interface technique from a published paper.

1. In RS1 (this problem set), you will implement the technique and the infrastructure that you'll need to run the experiment.
2. In RS2, you'll actually run the experiment on some people and collect data.
3. In RS3, you'll analyze the data using statistical tests.

Choose the Experiment

First, you should read the three research papers below:

- L. Findlater, K. Moffat, J. McGrenere, J. Dawson. "[Ephemeral Adaptation: The Use of Gradual Onset to Improve Menu Selection Performance](#)." *CHI 2009*.
- T. Grossman, R. Balakrishnan. "[The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area](#)." *CHI 2005*.
- R. Krishna, K. Hata, S. Chen, J. Kravitz, D.A. Shamma, L. Fei-Fei, M.S. Bernstein. "[Embracing error to enable rapid crowdsourcing](#)." *CHI 2016*.

You may have to look at some of the cited work in each paper to fully understand the technique and the study design.

Next, decide which experiment you will reproduce. Here are your choices:

- Ephemeral Adaptation paper: Use the experimental methodology described in Study 1 on just the control and long-onset ephemeral conditions described in Study 2
- Bubble Cursor paper: Experiment 2

- Rapid Crowdsourcing paper: Study 1

Your version of the experiment only needs two conditions: the control condition (the normal behavior of menus or mouse selection, or self-paced image labeling) and the main experimental condition (ephemeral adaptation, bubble cursor, or rapid labeling). Other conditions in these papers, like highlighting and object pointing, do not need to be replicated.

Implement the Experiment

- Implement a web interface that supports both the control and experimental conditions
- create an experimental framework around the interface that:
 - can be visited on the web at a URL
 - shows each task to the user, with instructions if appropriate
 - sequences tasks to counterbalance and cover all factors
 - initially tells the user how long they should expect to test the interface (~15 minutes), and includes an ending message to notify users when they have finished the experiment. Good practice would be to show the user their progress in completing the experiment as they go.
 - allows adjustment of experimental parameters that affect the total duration of the experiment, like number of tasks the user has to do, or duration of each task
 - records data (e.g. accuracy or time) for each trial
 - outputs data in a delimited format for analysis in RS3
 - collects the necessary demographic information and subjective feedback from users

Note that you don't need to run users, collect data, or analyze data for this assignment. You'll do that in RS2 and RS3. When you hand in this assignment, however, your experiment implementation must be *ready* to run users.

Subjective and Demographic Data

Your goal is to replicate the study in the paper. Pay close attention to the demographic information about users that is reported in the paper, and collect the same information about your users as part of your experiment implementation. Similarly, pay close attention to any subjective judgements that users are asked to make about the

conditions, and make sure that your experiment implementation collects those judgements from its users as well.

Recording Data

Your implementation **must** record trial data (including the independent and dependent variables), which you will analyze in RS3. Your implementation must record this data automatically and store the data online (either logging directly to a Google spreadsheet or server database for later analysis).

A good data output format puts one user trial (e.g. the time to make a single selection) on each line, with other fields on the line describing the conditions of that particular trial (e.g. which user, which experimental condition, the number of the block, the number of the trial within the block).

Your implementation can record data to a Google spreadsheet using [googlesender.py](#). More information about this logging framework can be found in the [AS1 assignment](#), but note that the specific data logged by AS1 is probably *not* what you want to log for your experiment.

Or, if you prefer, your implementation can record data with a custom server and backend database that you create, install, and run. If you go this route, then you need to keep your server running continuously through RS2 and RS3, and ensure that the data is not lost. The course staff can't help with custom backends.

Your implementation doesn't need to do its own analysis or generate its own graphs. We'll use other tools for that. It just needs to record raw data in a form that will be easy to import into an analysis tool like a spreadsheet or [R](#).

In your submission, include a file named `example-data.csv` containing sample data recorded by your experiment implementation, just using yourself as a pilot participant. This file should include the subjective and demographic data your implementation collects as well.

Make a Screencast on YouTube/Vimeo

Create a **screencast** that shows your experiment implementation in action. The video should show:

- the control and experimental conditions in action
- how tasks are presented to the user

- brief screen captions saying what the video is showing (audio is not necessary)

To create the screencast, you can use the free [CamStudio](#) on Windows or Quicktime (included in OS X), a trial version of [Snapz Pro X](#), or [iShowU HD](#) on Mac. Your video should be **at most 2 minutes** long.

Upload your video to YouTube or Vimeo, so that you can hand in just a link to it rather than the entire video file. Make sure your video has a high enough resolution that you can easily see the important features of the interface.

Prepare your Turn-in

To help your grader evaluate your experiment implementation, you should prepare a **README** text file which explains how your experiment meets all of the necessary conditions.

Create a text file that you will include with your submission that contains the following information. You will lose points if it is unclear whether your experiment meets the criteria, so please be diligent in ensuring that the grader has everything they need to evaluate the submission.

- Which paper you chose
- A list of the people you discussed this assignment with, or else an **explicit statement that you had no collaborators**. This is an individual assignment, so be aware of the course's [collaboration policy](#).
- The URL of your YouTube/Vimeo screencast
- The URL of your experiment implementation
- How are tasks sequenced in your experimental implementation
- What are the experimental parameters in your implementation? How can they be adjusted?
- What data is collected for each trial run and in what format?
- What demographic data or subjective judgements do you collect from each user, and why?

What to Hand In

Package your completed assignment as a zip file that contains:

- all of your **source code**
- **example-data.csv** containing a sample of data recorded by your experiment implementation

- a **README.txt** file in plain text format

Here's a checklist of things you should confirm before you hand in:

1. Make a fresh folder and unpack your zip file into it
2. Make sure your collaborators are named in the README

Submit your zip file on Blackboard.

Grading

This assignment will be judged on three dimensions:

- Functionality (40%): does it work?
- Completeness and fidelity (40%): does it include all the necessary elements to conduct an experiment, and will that experiment faithfully reproduce the experimental design published in the paper?
- Presentation (20%): does your handin (video, README, and zip file) make it easy to judge functionality, completeness, and fidelity?