

% Documenting and Visualizing Experiments % Interaction Techniques and Technologies (ITT), SS 2016 % Session 4 (03.05.2016), Raphael Wimmer

Overview

These are slides/notes for the lecture, automatically generated from the slide set. Please extend this outline with your own notes.

Quiz: Which of the following statements is true?

- independent variables are controlled by the experimenter
- pseudo-randomization reduces the effect of random variables
- accuracy is more important than precision for quantitative experiments
- between-subjects designs are better than within-subjects designs
- confounding variables always have an effect on dependent variables

Assignment 2: Reaction Time Experiment

2.1: Whys and Hows of Experiments in HCI

- What does Hornbæk say about the relationship between self-reported and objectively measured task performance?
- Describe advantages and limitations of research hypotheses.
- Is it acceptable to conduct user studies with media informatics students as participants instead of a broad sample of potential users?
- What are advantages and limitations?
- Name one topic for a user study where it would be problematic to have only media informatics students as participants.
- Name one counter-example.

2.2 - 2.4: Plan, implement, and conduct an experiment

-
- What were your research questions and hypotheses?
 - How did you select your participants?
 - Which problems did you encounter?
 - Which results did you observe?

Quiz: What is an interaction technique?

- a method for accomplishing a task
- a tool for changing the computer's state
- communication between two humans
- a QWERTY keyboard

- undo

Interaction Techniques

...

"a way of using a physical input/output device to perform a generic task in a human-computer dialogue."

J.D. Foley, A. van Dam, S.K. Feiner and J.F. Hughes (1990), *Computer Graphics: Principles and Practice*, Addison-Wesley.

Logging Experiments

Why Logging?

- often, scientists only have a lab notebook
- we generally use computers for running experiments
- that's awesome: let's automate recording the results of experiments
- see also: Miguel Nacenta (2013) ["What and how to log in your experimental HCI software"](#)

How & What to Log (1)

- use CSV
- log everything: timestamps, IDs, raw and filtered values, ...
- make logs self-contained:
 - every row should contain all information, don't depend on context
 - include header
- * log redundantly, e.g., both category IDs and textual descriptions - eases analysis
- check log files manually
- test analysis workflow with test data before running the test - make unit tests!

How & What to Log (2)

- store everything, never delete, overwrite
- do never ever manually change anything in the logfiles
- also store raw input data (phrase sets, task order for participants, etc.)
- automate processing and analysis of log, use idempotent scripts, use a Makefile
- use version control, check in both log files and application generating the logs
- use individual files per user/session

CSV - Comma-Separated Values

- *lingua franca* of experimental data
- easy to write, easy to parse
- line-based - can also be processed by many Unix tools
- format:

...

```
~~~~~ „Timestamp“;„UserID“;„Time(ms)“ „1234“;„1“;„143“ „1532“;„1“;„543“ ~~~~~
```

- various dialects, ideally include all data in "" and use ; as a separator
- Python: module csv:

...

```
~~~~~ {#csv.py .numberLines} import csv logfile = open(„user1.csv“) d = {„id“: 1, „time(ms)“: 327}
out = csv.DictWriter(logfile, [„id“, „time(ms)“]) out.writeheader() out.writerow(d) logfile.close() ~~~~~
```

General Logging in Python

- [Python module `logging`](#)
- [Python Logging Cookbook: via Network](#)
- [Python Logging Cookbook: JSON](#)

Analyzing Experimental Data

SciPy

- <http://scipy.org>
- Collection of libraries / projects for scientific computing
- Most important for us:
 - numpy - numerical library, efficient arrays
 - scipy.stats - statistical functions
 - matplotlib - plotting data

Matplotlib & Co.

- [matplotlib](#): Python module for generating high-quality graphs
- [pyplot](#): wrapper for MATLAB-style plotting (nicer for interactive use)
- pylab: combines numpy and pyplot into one namespace as a kind of MATLAB replacement
- Alternatives:
 - [Chaco](#)
 - [Mayavi](#) (3D)

- [PyQtGraph](#) (which we will use later in the course)
- [plot.ly](#)
- [Google Chart API](#)

Using `matplotlib` in iPython Notebook

- `%matplotlib inline`
- `from pylab import *` let's try it...

Statistics with SciPy

- Student's t-test: `scipy.stats.ttest_ind()`, `scipy.stats.ttest_rel()`, `scipy.stats.ttest_1sample()`
- Linear regression: `scipy.stats.linregress()` or `numpy.polyfit()`
- Pearson's Correlation Coefficient: `scipy.stats.pearsonr()`

Outlook

Next Session

- Pointing
- Input Hardware
- Transfer Functions
- Fitts' Law and Steering Law

Course Assignment

Extend a Fitts' Law test application, conduct an experiment, analyze the data.

Goals of this assignment:

- memorize the concepts discussed in today's session
- get some practice in conducting experiments
- generate an example data set for further analysis
- get a better grasp of Python and matplotlib
- get an overview of Fitts' Law
- learn how to analyze simple datasets

ENDE

From:

<https://wiki.mi.uni-r.de/> - **MI Wiki**

Permanent link:

https://wiki.mi.uni-r.de/user/raphael_wimmer/tests

Last update: **21.04.2017 12:33**

