Web-based experiments

CRNL2024 — Hands'on session — 2024

Romain Ligneul & Daniela Domingues

Structure of the session

Talk (60 minutes)

I. Rationale for online cognitive testing
II. Regulations
III. Performance, precision and quality
IV. Online experiments beyond keyboards

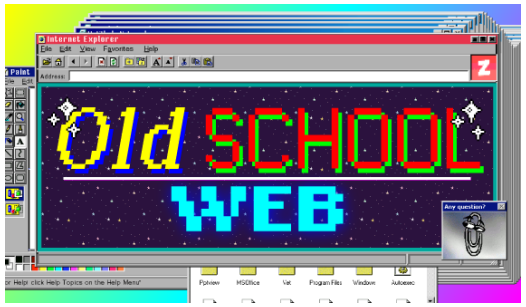
Essentials of web programming (30 minutes)

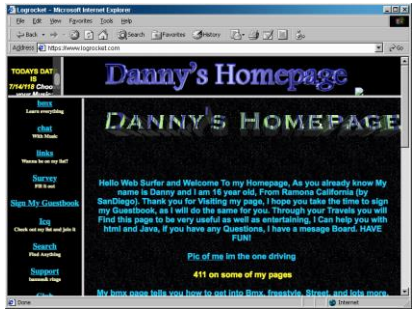
Practical (120 minutes)

I. Overview of the tutorial code
II. Modifications of the code
III. Serving the experiment online

Historical context

1990-2005
Simple things with simple code
User-made web

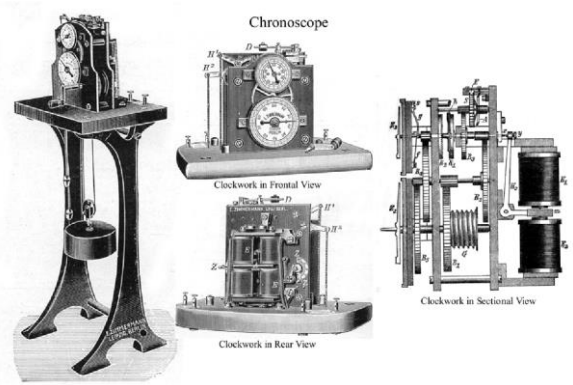




2005-2020
Complex things with complex code
Platform-made web

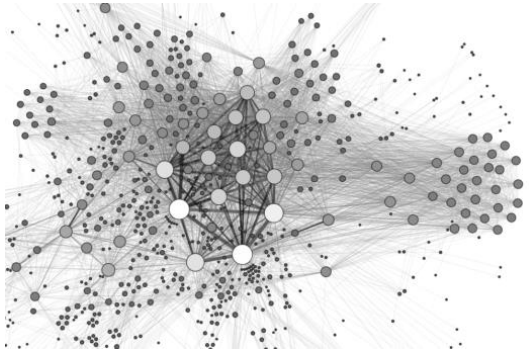
2020-
Complex things with simple code
User-made web again?

Historical context



XIXth century
Imprecise measurements…
in a handful of participants…
in the lab.

XXth century
Precise measurements…
in a few dozens of participants…
in the lab.

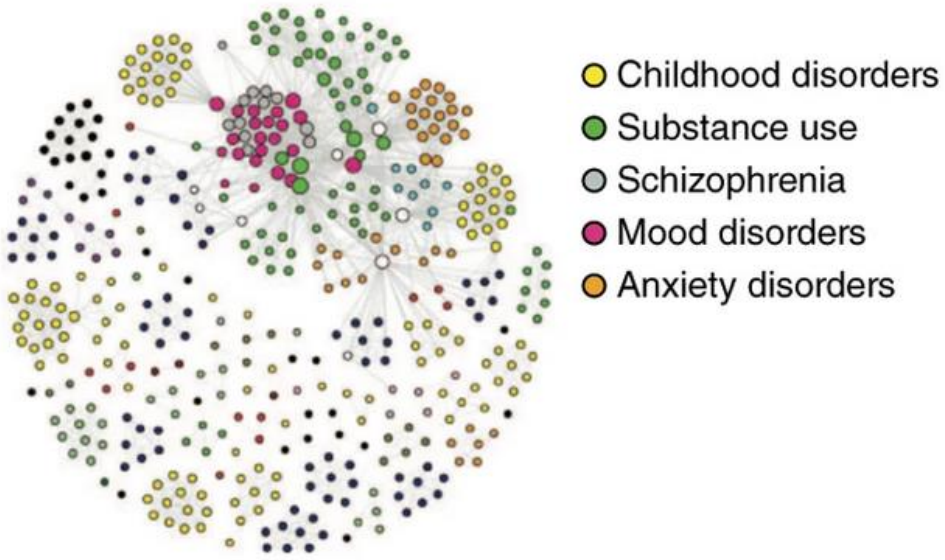


XXIth century
Precise measurements…
in a thousands of participants…
anywhere, anytime!

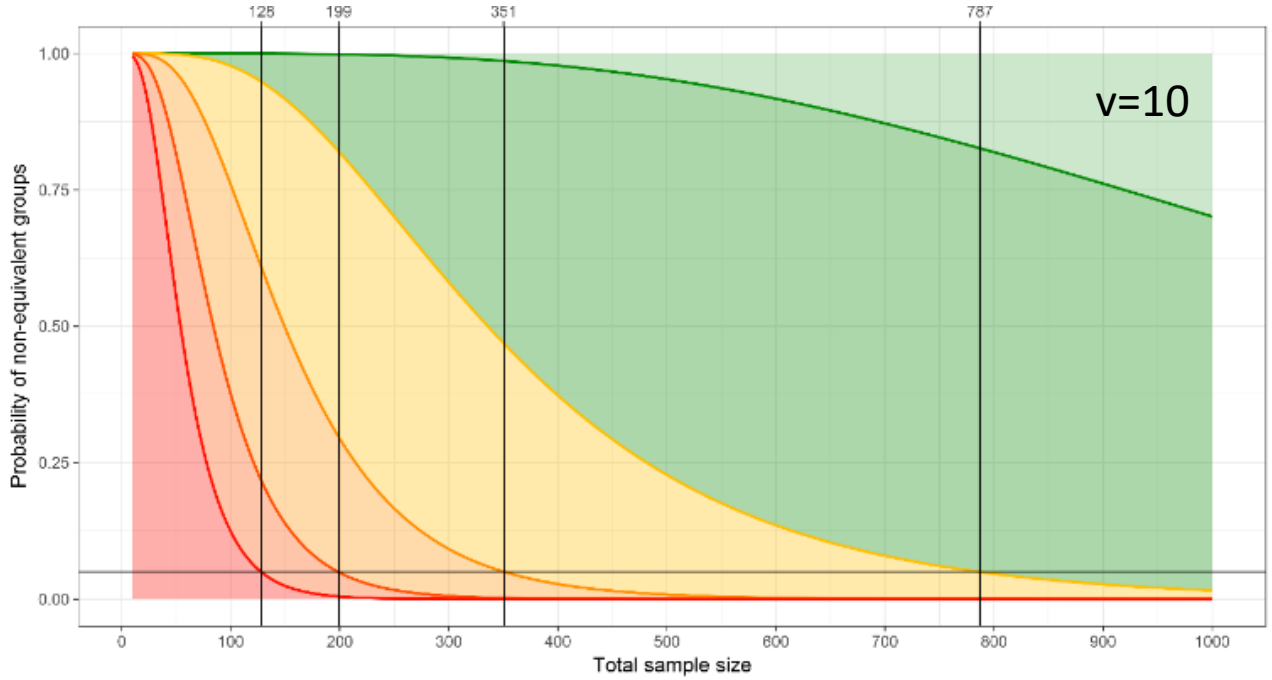
Does it matter?

Reproducible and relevant cognitive science

High-dimensionality problems

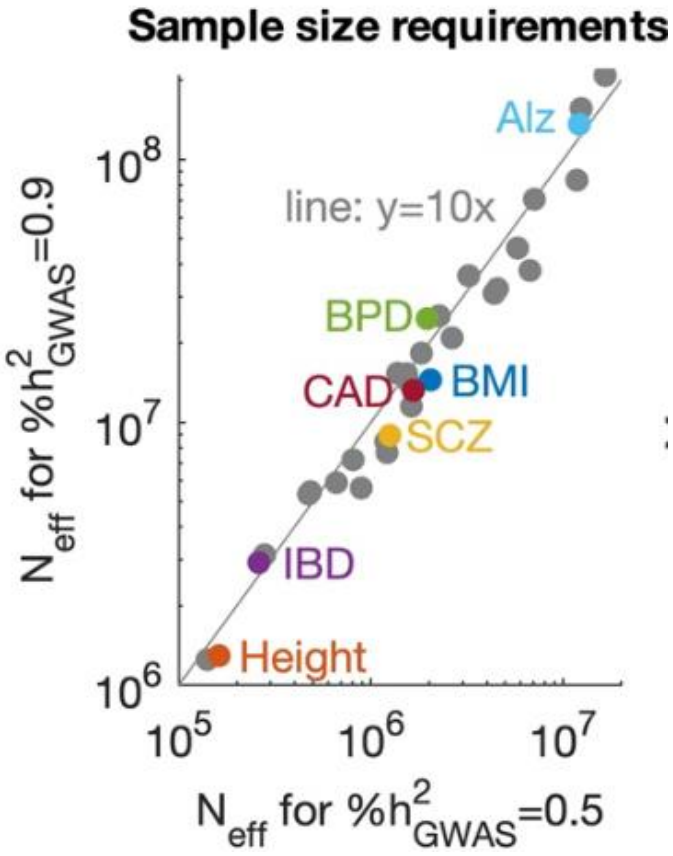


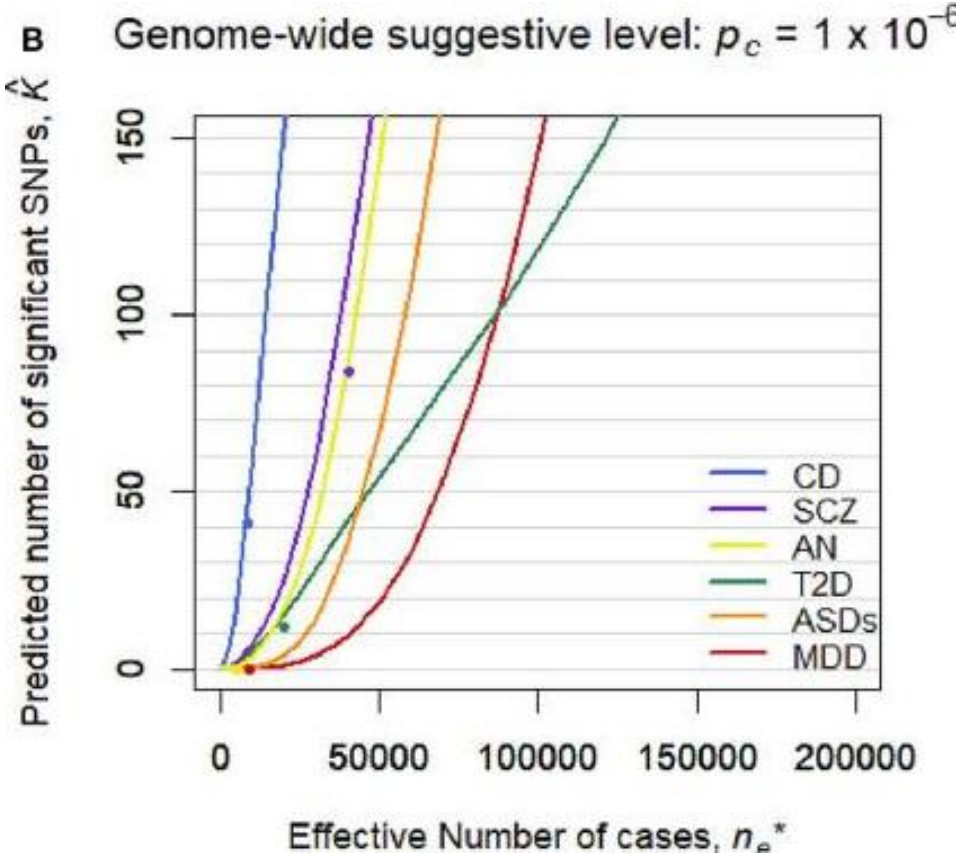
Impact of nuisance variables (v=10)



Huys et al., 2016; Gruijters & Peters, 2017

Reproducible and relevant neurogenetics

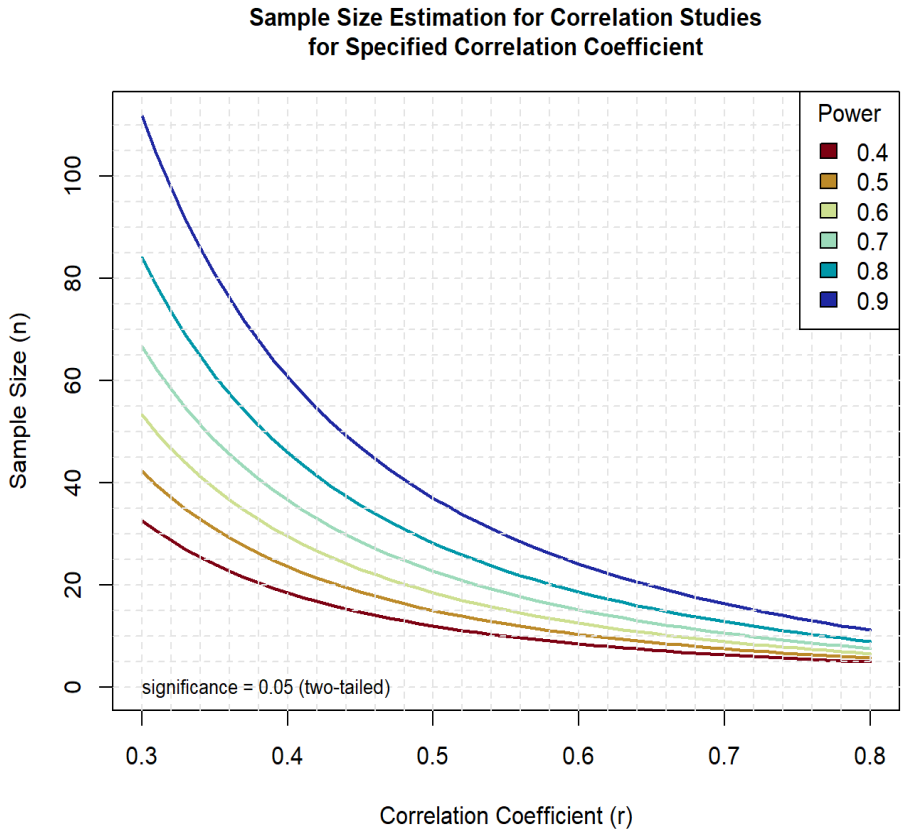




O'Connor, 2021; Nishino et al., 2018

Sample size matters

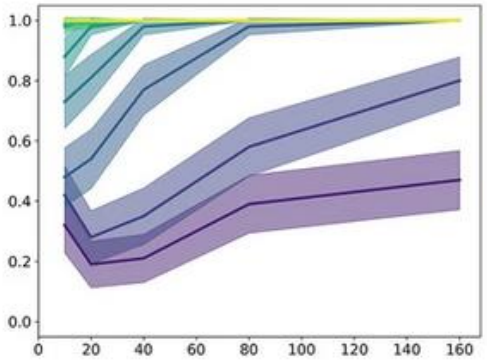
Correlation



Regression

Number of predictors	Sample sizes based on power analysis: Effect size		
	Small	Medium	Large
1	390	53	24
2	481	66	30
3	547	76	35
4	599	84	39
5	645	91	42
6	686	97	46
7	726	102	48
8	757	108	51
9	788	113	54
10	844	117	56
15	952	138	67
20	1066	156	77
30	1247	187	94
40	1407	213	110

Clustering



Rules of thumb for k-means strongly depend on expected cluster separation, cluster sizes and objectives.

⇒ 30 samples / smallest cluster frequency for well separable problems

⇒ More noisy problems can increase this requirement by an order of magnitude

⇒ Number of features also plays a (complex) role

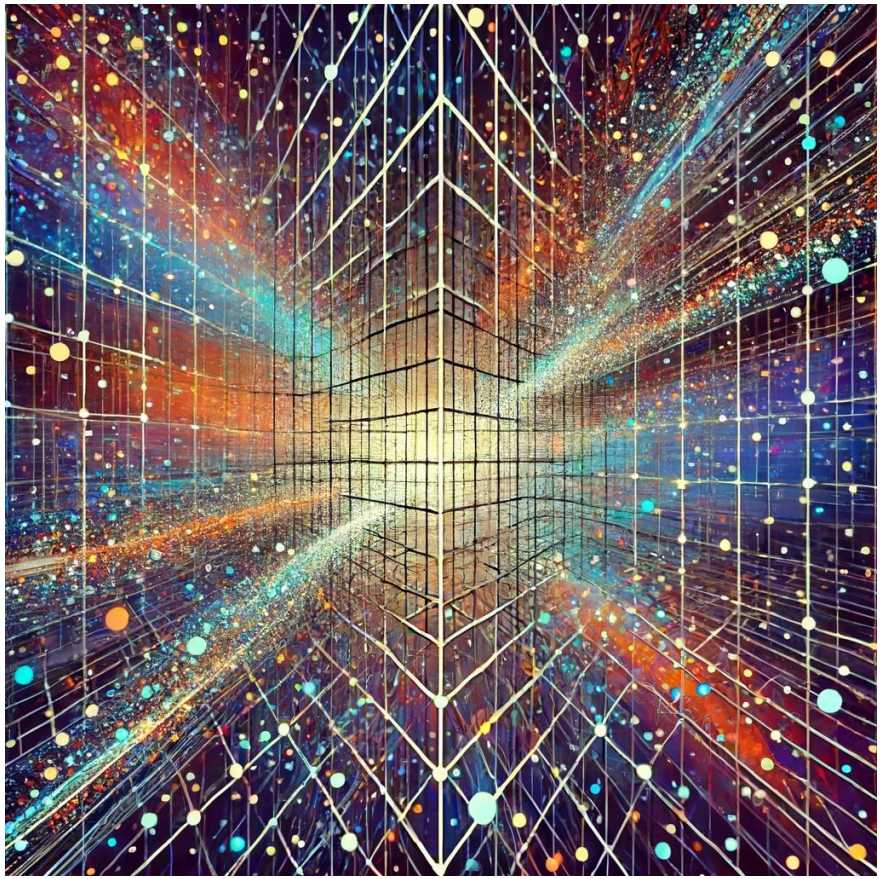
Green et al., 1991; Dalmaijer et al, 2022

Sample size matters

Larger sample sizes can…

- Address the reproducibility crisis
- Improve the estimation of effect sizes
- Provide enough data to fit hungry models
- Allow discovery of weak multifactorial associations (GWAS, risk factors)

In general, large sample sizes are key to deal with the “curse of dimensionality”



Beyond big sample sizes

But there is more to web testing than just big sample sizes…

- Access to people who rarely comes to the lab (rural areas, workers, people with physical disabilities)

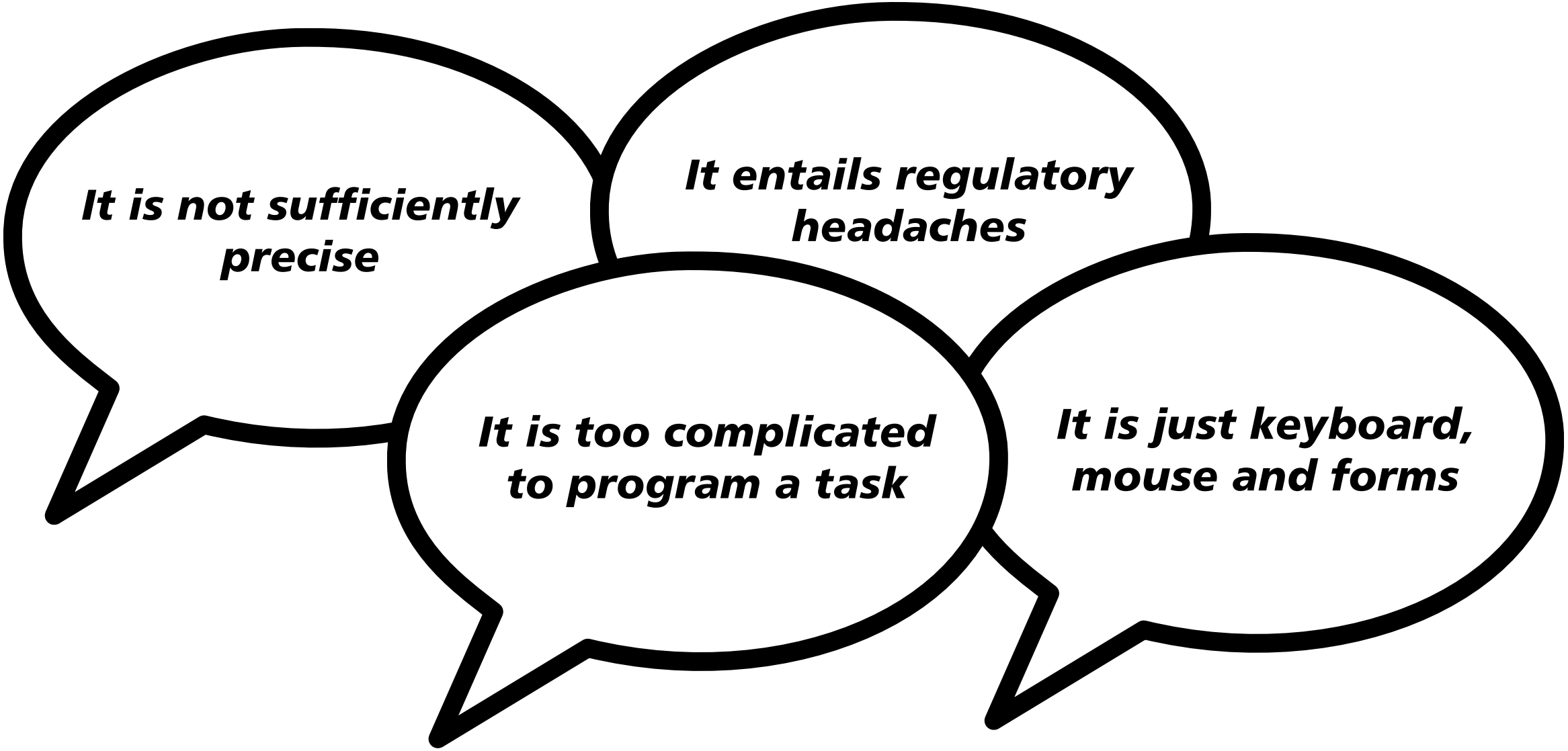
⇒ Generalizability and applicability

- Access to people in their natural habitat and avoid experimenter/contextual biases

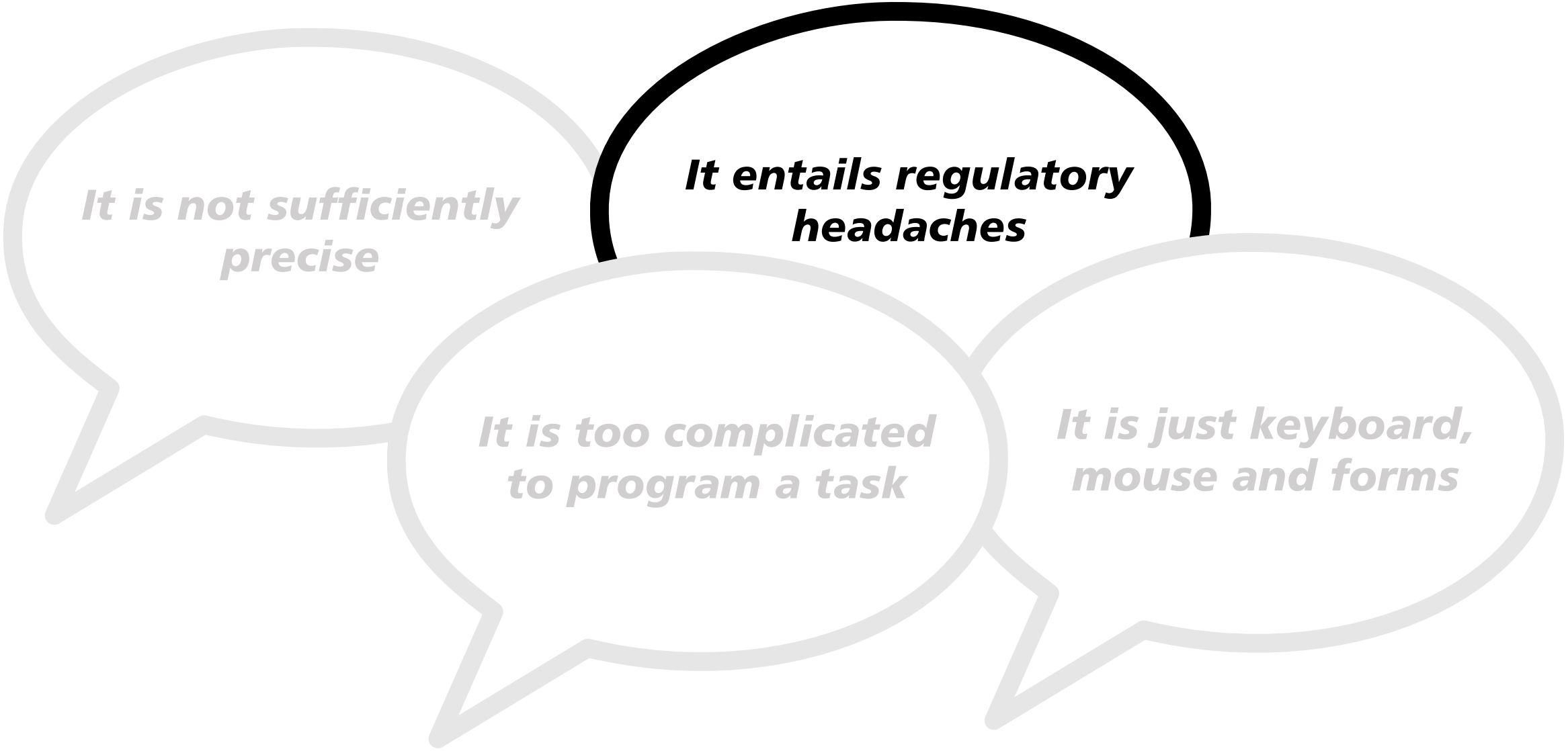
⇒ Ecological value

- Spend less student time (and money) !

Why is online testing still underdeveloped?



Regulatory considerations



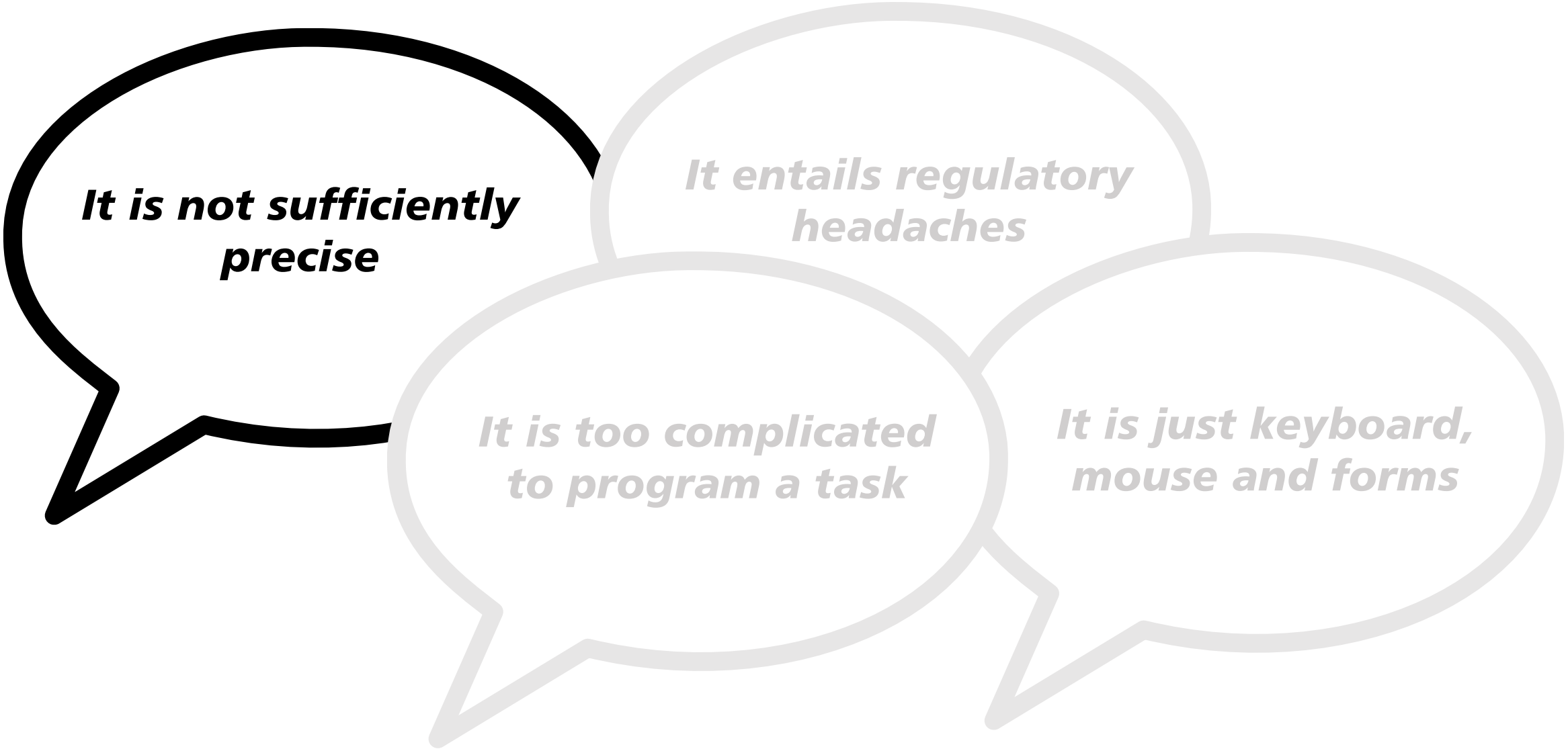
Regulatory considerations

It was indeed difficult to obtain approval from Ethics committees, CNIL, etc

- Poor understanding of web technologies and poor awareness of their benefits
- Overestimation of “offline” privacy
- Programming strategies and tools were indeed more vulnerable to breach
- Slow adjustment to GDPR requirements (General Data Protection Regulation)

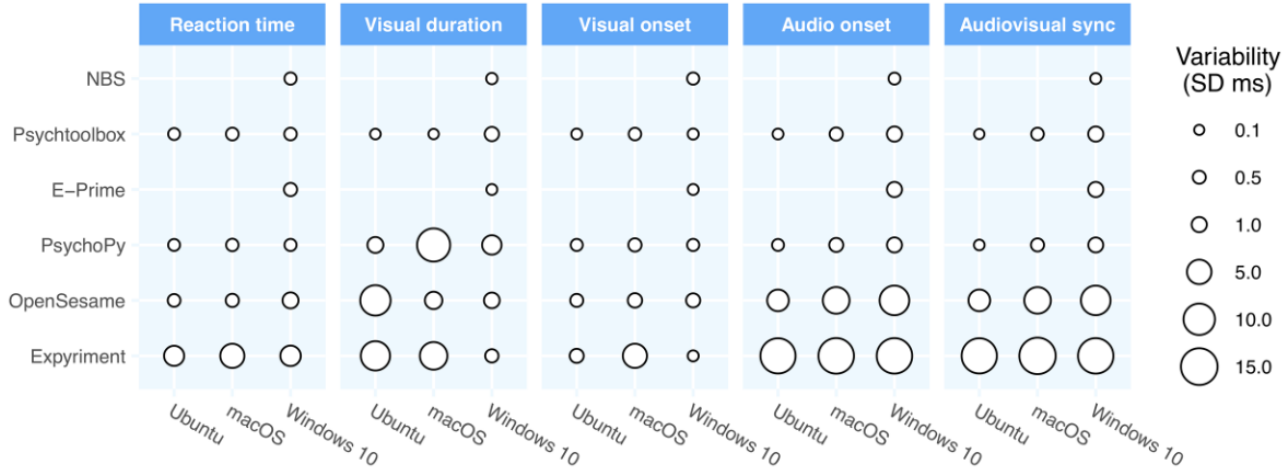
- More and more centers have blanket RIPH3 approvals (and CRNL soon?)
- A French provider (OVH) has obtained the agreement to host medical data
- Research institutes are increasing well equiped to host experiment and data
- Testing in hospital remains complex..

Why is online testing still underdeveloped?



Precision of online testing: timing

Lab software



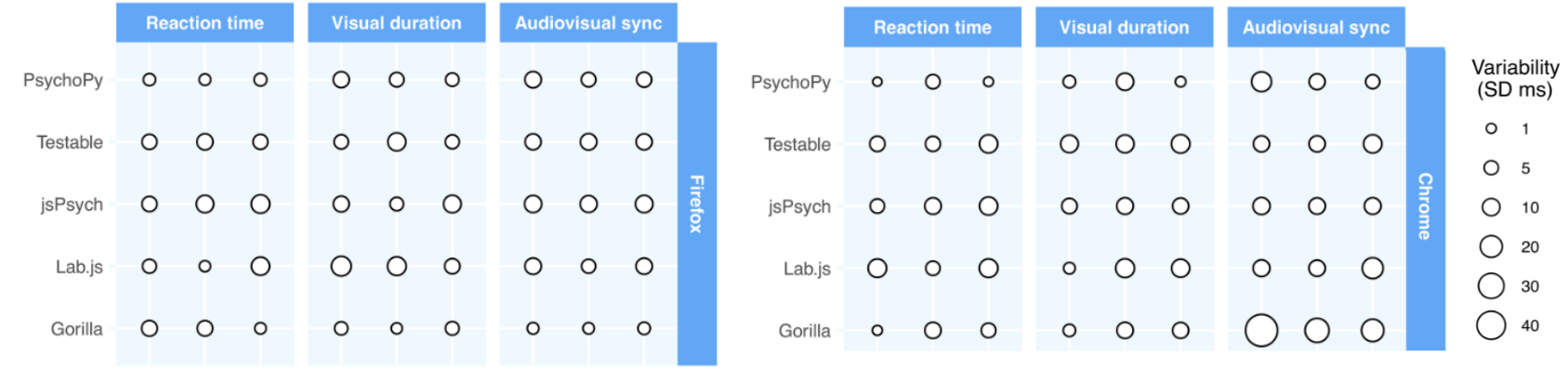
The timing mega-study: comparing a range of experiment generators, both lab-based and online

David Bridges[1], Alain Pitiot[2], Michael R. MacAskill[3,4] and Jonathan W. Peirce[1]

[1] School of Psychology, University of Nottingham, Nottingham, UK
[2] Laboratory of Image and Data Analysis, Ilixa Ltd., London, UK
[3] Department of Medicine, University of Otago, Christchurch, New Zealand
[4] New Zealand Brain Research Institute, Christchurch, New Zealand

2020 study
⇒ Ages ago!

Online software



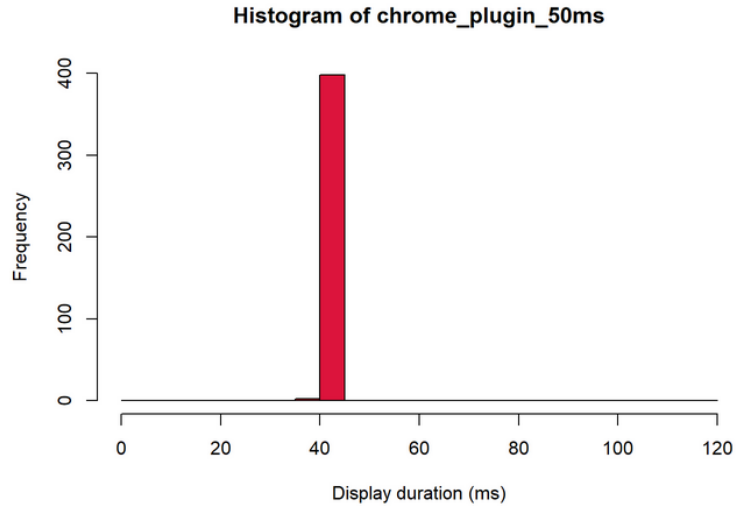
Precision of online testing: timing

- The vast majority of online behavioral paradigms will be fine using any reasonable programming approach

- Some paradigms will require timing based on the “requestAnimationFrame” methods rather than conventional display methods.

- Very few paradigms cannot be run online:
  - Specific hardware required
  - Fine-grained audiovisual synchrony
  - Very short (<20ms) display durations

- Knowing exactly when ✅
- Controlling exactly how long ✅
- Controlling exactly when ⚠️



A new jsPsych plugin for psychophysics, providing accurate display duration and stimulus onset asynchrony

Daiichiro Kuroki[1]

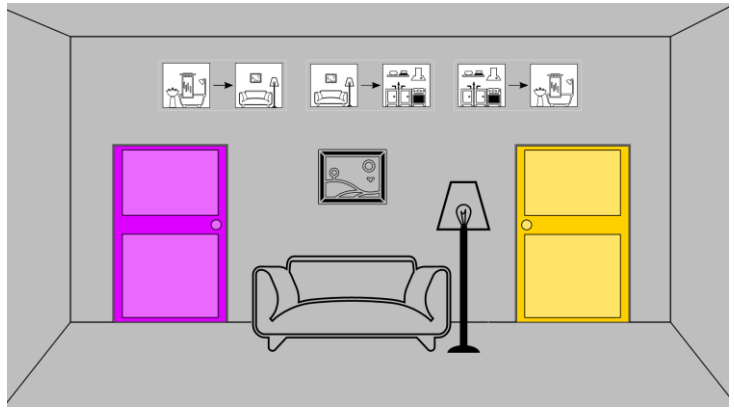
Published online: 22 July 2020

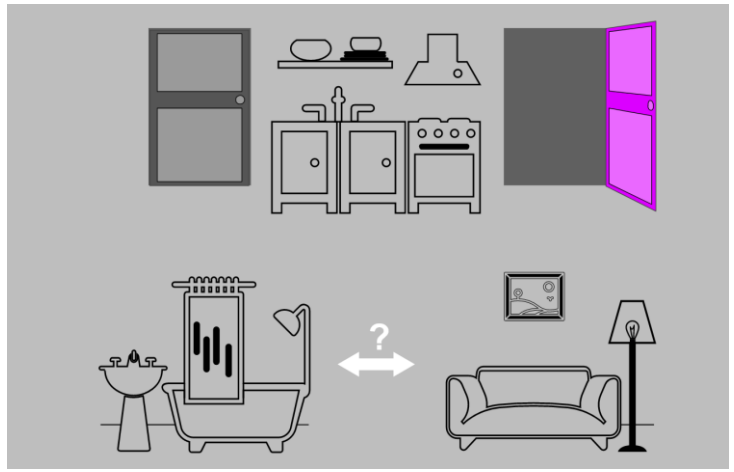
https://jspsychophysics.hes.kyushu-u.ac.jp/

See also: Gao et al. 2020, *Plos One*

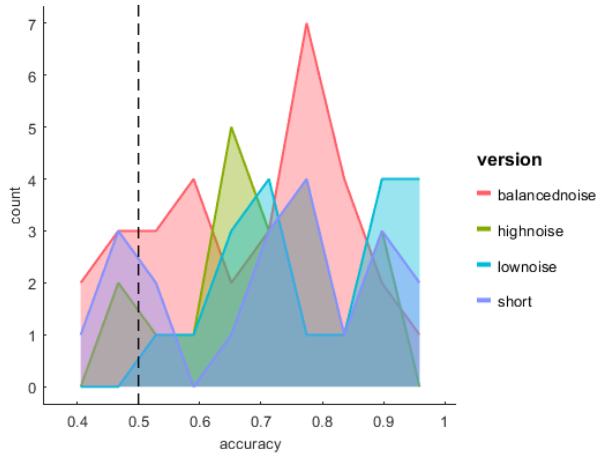
Precision of online testing: engagement

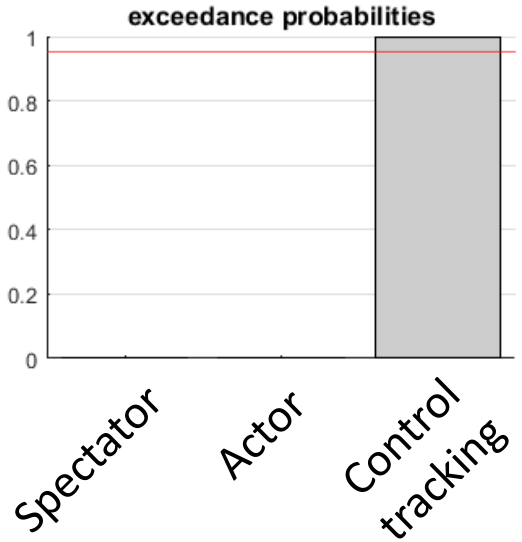
Explore and Predict task

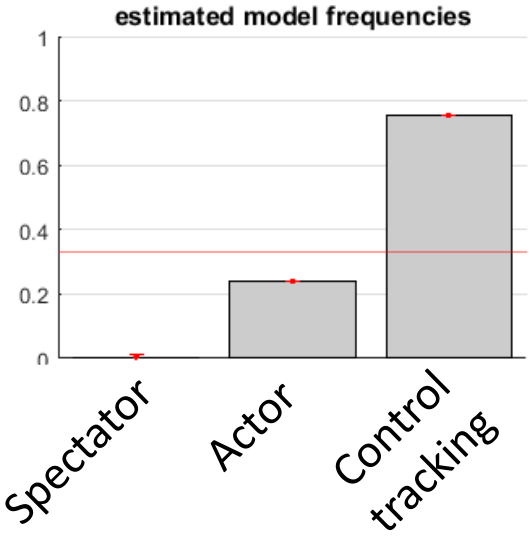




Aging adults
(55-75 years old)
Recruitment on Prolific

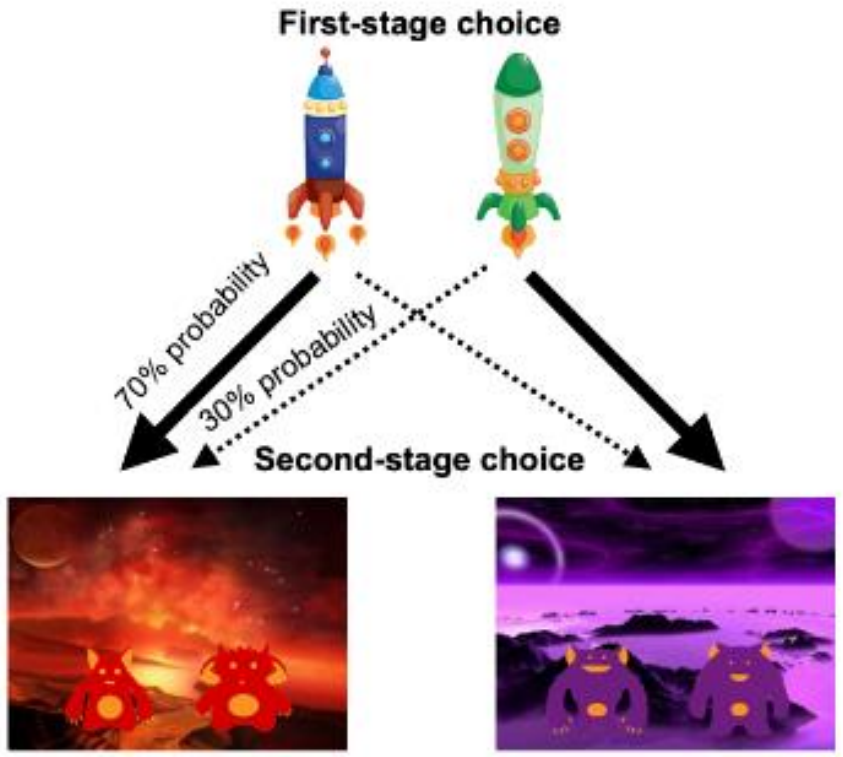


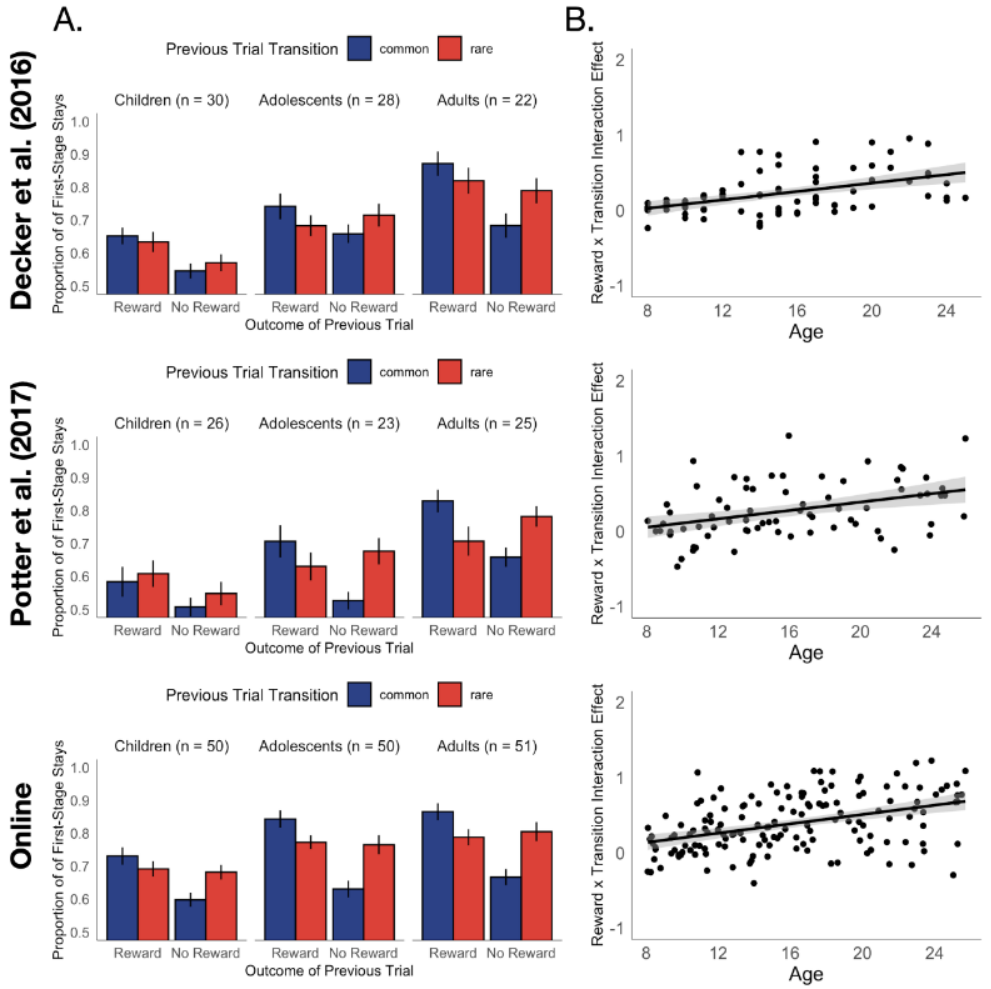




Precision of online testing: engagement

Example of the 2-step task



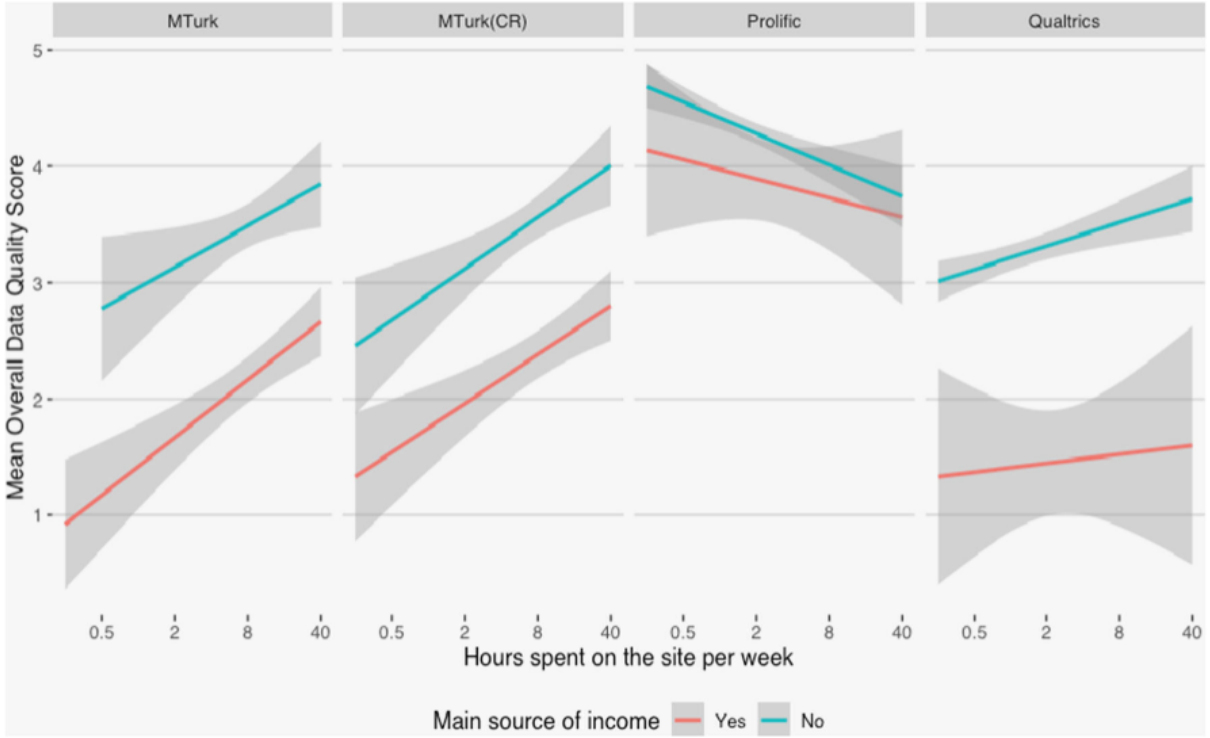


Nussenbaum et al., 2020

Includes **all** online participants (“manually” recruited)

Precision of online testing: engagement

The quality of the data is quite variable across recruitment platforms, and it is typically lower in those who complete online experiments as their main “job”.



Peer et al. 2022,
Behav Res Methods

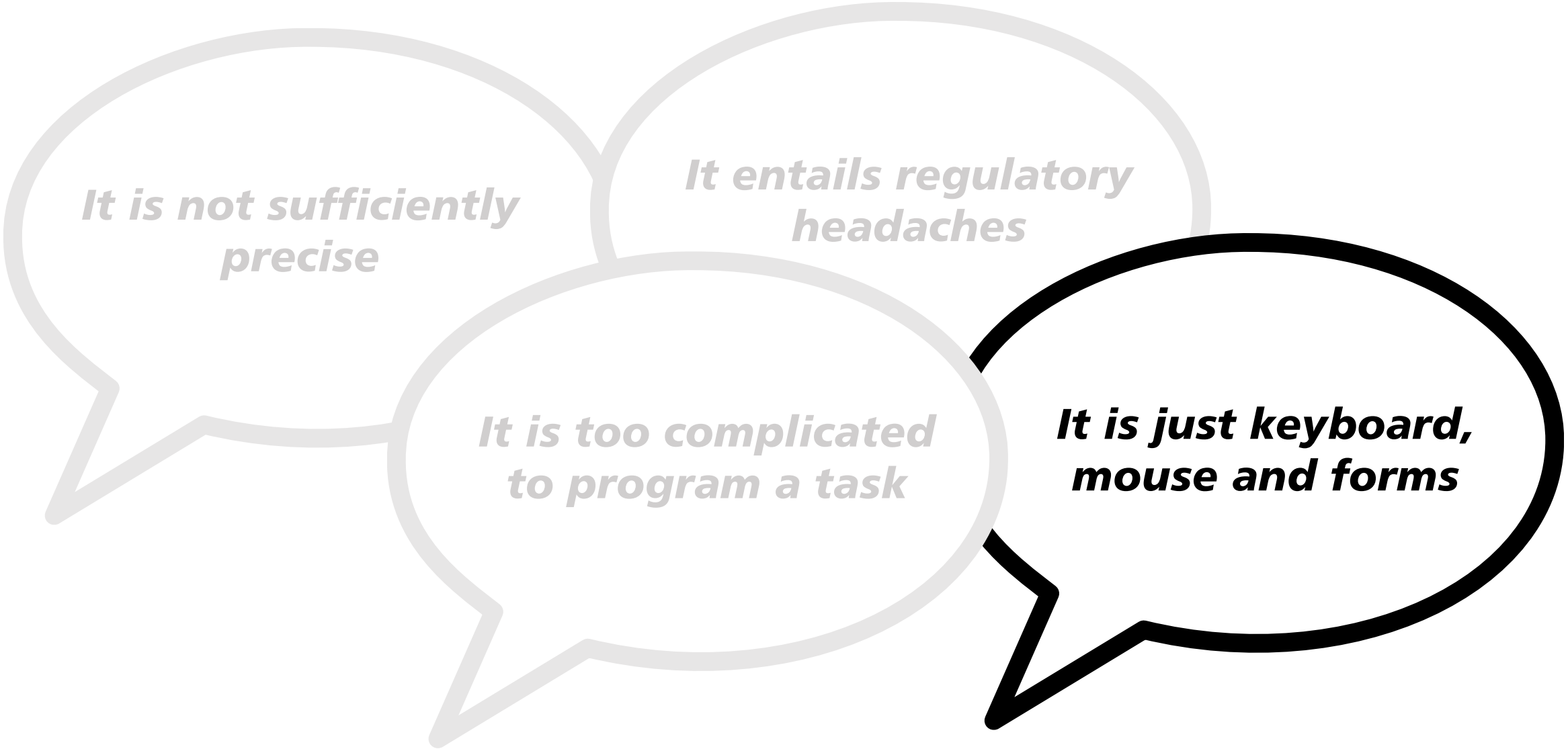
Recruitment platforms are incredibly convenient, but nothing prevents you from recruiting using your own channels (i.e., mailing lists, patient associations, etc.)

Browser performances and data quality

A few tips to maximize online data quality

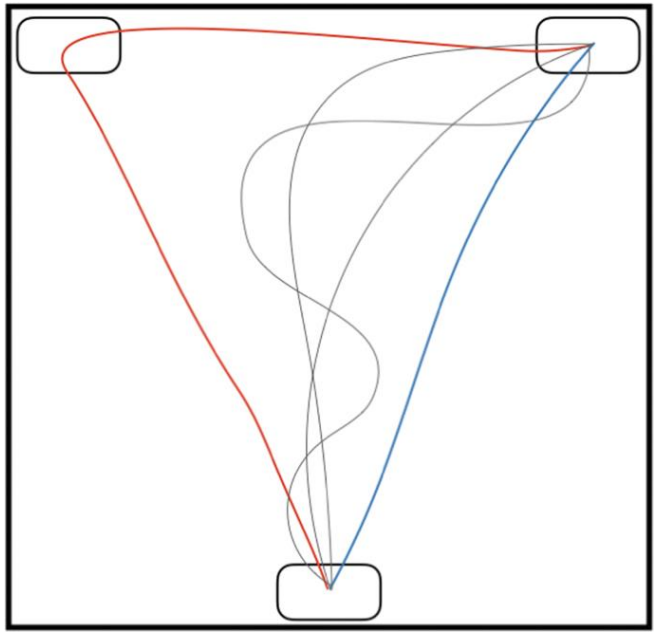
- Make your experiments cool, fluid, visually appeally and easy to understand
  ⇒ Online experiments = reality check on the quality of our average instructions…
- Prevent the use of mobile browsers (or implement custom methods for it)
- Implement **attention checks** and **comprehension questions**
  ⇒ Increase focus and make sure you can discard inattentive participants
- Include forced response trials to prevent passive viewing
- Use “responsive” visual display and preload correctly the stimuli
- Use fullscreen mode and exclude participants who exit it (straightforward to detect)
- Keep your experiment short (<15-20 minutes) and fractionate if necessary.
- If you can, notify participants that compensation is conditional upon attention
- For participants with special needs, offer “hotline” sessions (RA available for 5-10 ppl)

Why is online testing still underdeveloped?



Beyond simple tasks and measurements

Mouse-tracking
(jsPsych extension)



Fine-grained motor behaviors
Continuous readouts

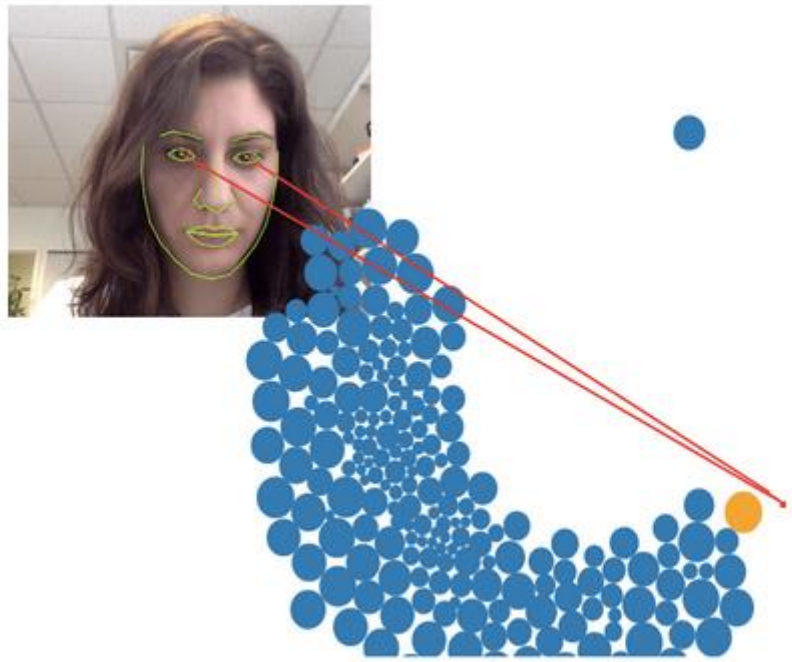
Decoupling of mouse input and
cursor position (Pointerlock API)

Sensorimotor learning
Environments with obstacles

Main limitation: no access to mouse DPI (credit card test as a poor's man solution)

Online testing beyond button presses

Webcam-based eye tracking (using for ex. Webgazer.js)

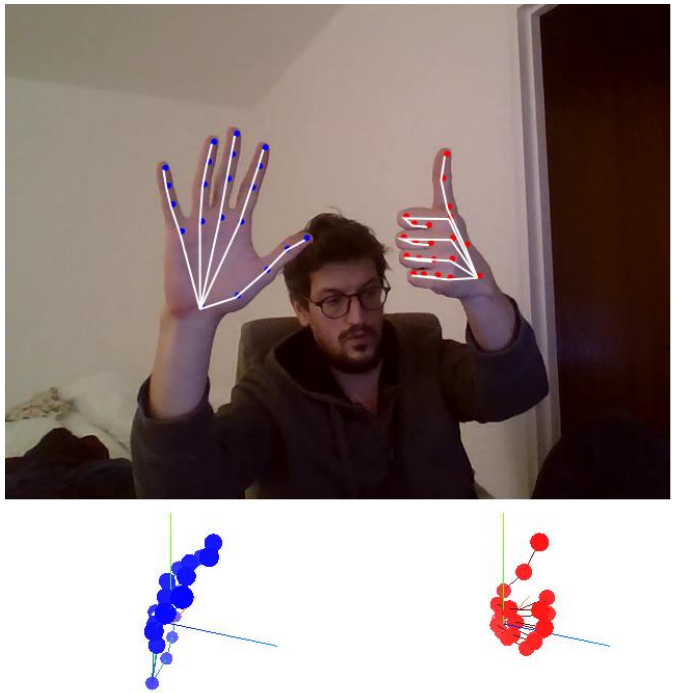


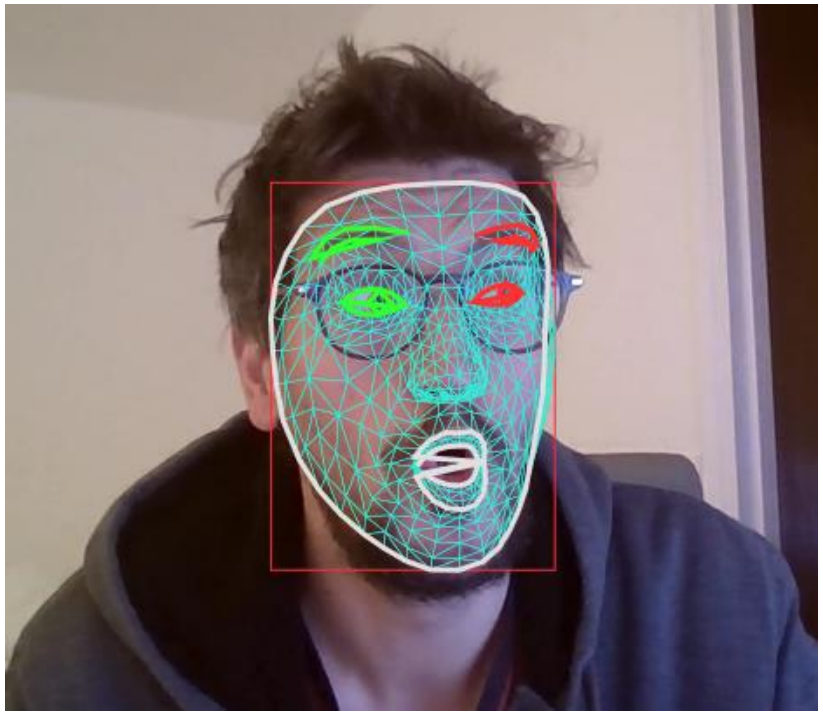
Study and assessment of attention, information-sampling processes

Main limitation: requires a lengthy calibration for a limited precision (especially if participant is allowed to move their head..)

Online testing beyond button presses

Face and body parts detection in the browser (high-dimensional action spaces)





TensorFlow.js

TF.js is an amazing library
⇒ Well documented
⇒ Check their demos

Online testing beyond button presses

Hybrid approaches combining devices

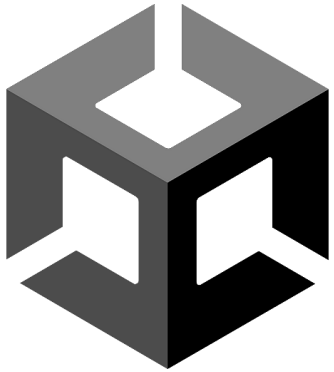
It is relatively to develop cross-platform experiments using a single development environment.

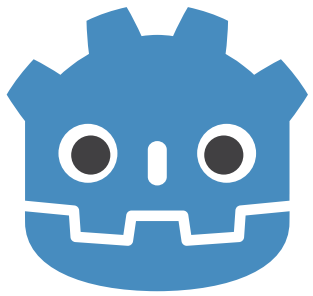
→ Web apps (ease of use)

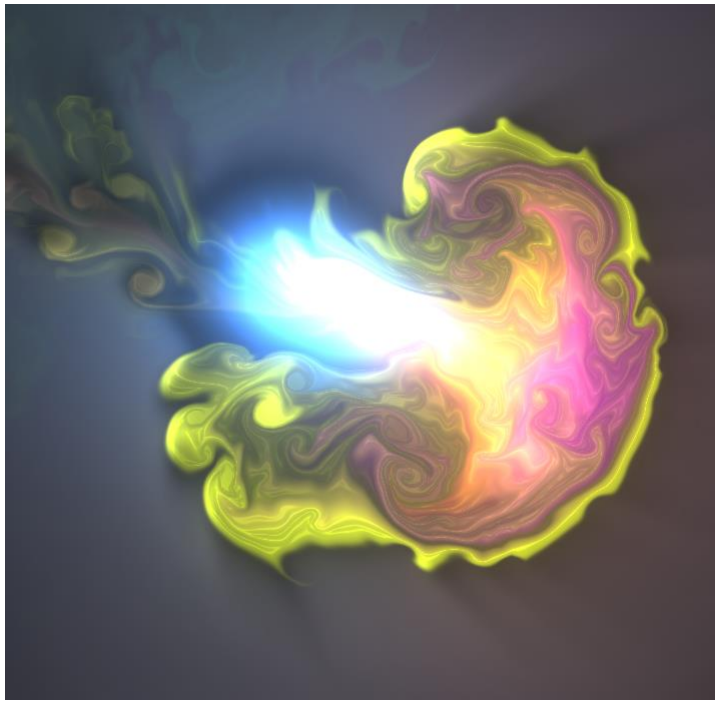
→ Desktop apps (reliability, persistence)

→ Mobile apps (sensors, mobility, reachability)

Integration with GPUs and game engines

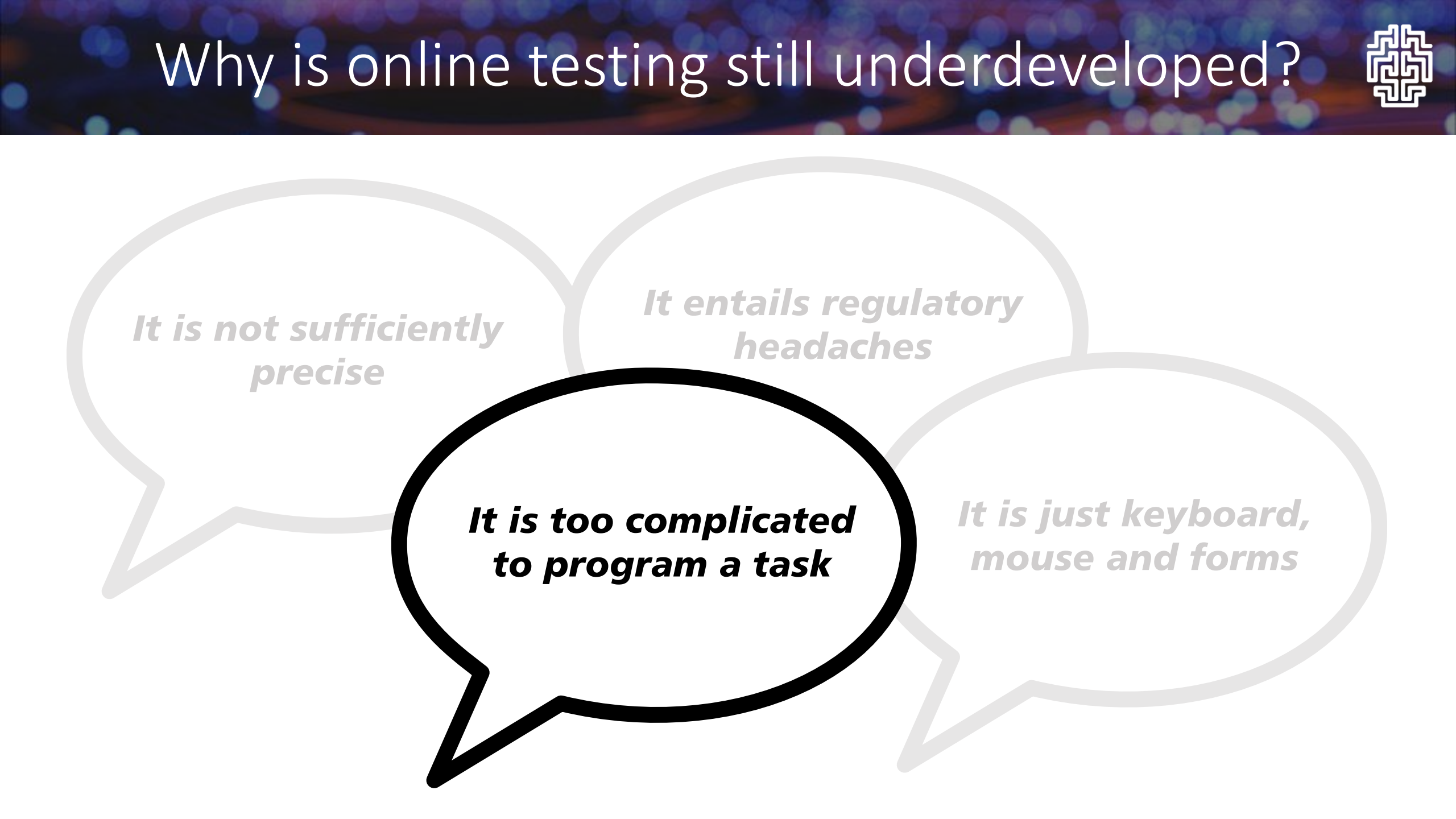




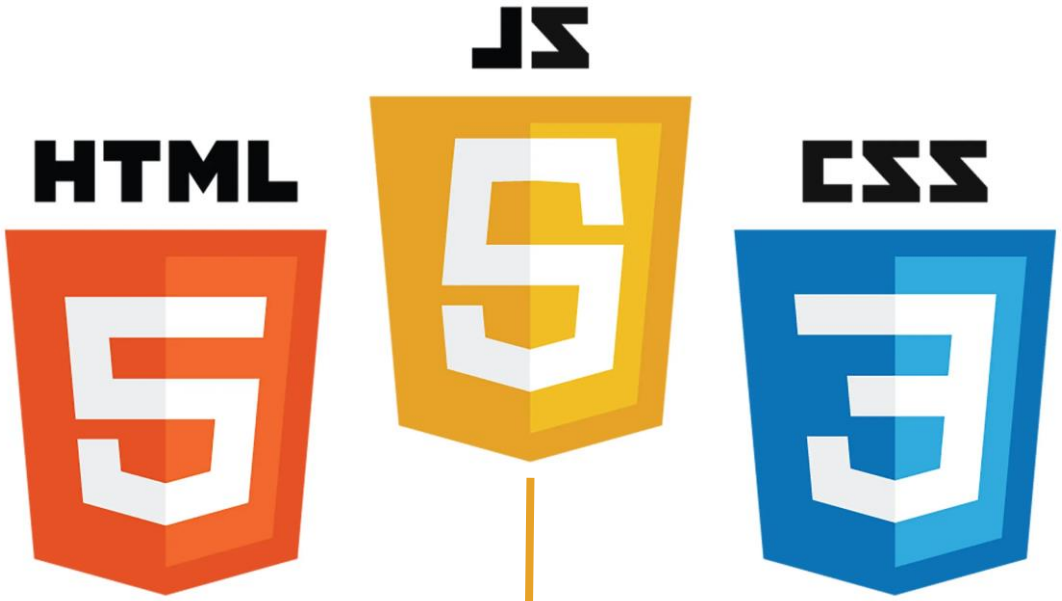


[WebGL demo]

Why is online testing still underdeveloped?



The language(s) of web programming

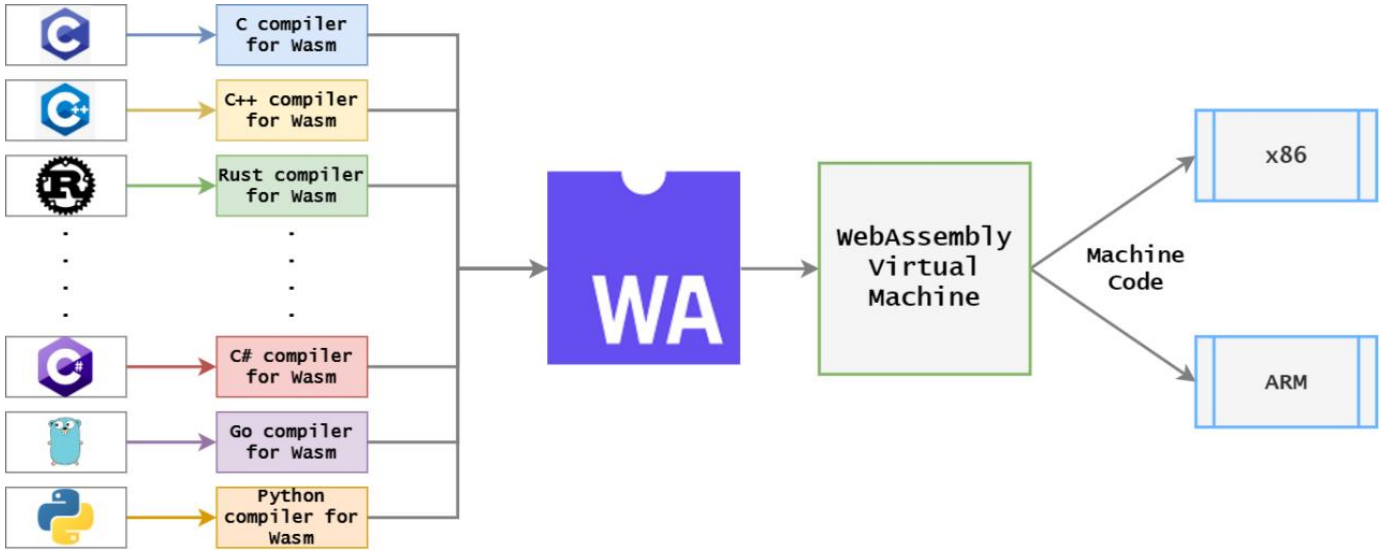


HTML
Put content in containers
Simple display command

Javascript
All the rest!
Interact with users, browsers, computers, databases & servers

CSS
Change containers' appearance
Simple animation logics

The language(s) of web programming



With the advent of cloud computing, it will become more and more meaning full to code directly in the browser (Pyscript or Pyodide may run your Python experiments alreaydy!)

WebAssembly allow to compile many languages to run them in the browser

Experiment builders versus low-level code

Several excellent **front-end** frameworks allow to create experiments in the browser.
⇒ Not more difficult than in-lab frameworks like Psychtoolbox or PsychoPy, and sometimes easier!

	**PsychoJS**	Free, comes with the same experiment builder as PsychoPy
	**Lab.js**	Free, dedicated experiment builder, newcomer
	**Gorilla**	Commercial, experiment builder, support, all-in-one solution
	**jsPsych**	Free, script-based (JS programming required), very flexible

Experiment builders versus low-level code

As compared to in-lab testing, online cognitive testing involves ingredients

- Local development but hosting service needed to serve the experiment
- A proper database must in general be used to save and retrieve the data
- A strategie to recruit a lot of participants and engage them properly

Low effort & low control → **High effort & high control**

Use a service provider to host the task

Use an experiment builder to avoid coding

Store your data in a shared database

GORILLA

Pavlovia

/pævlɒvɪə/

where behaviour is studied

qualtrics XM

JATOS

node js express

mongoDB

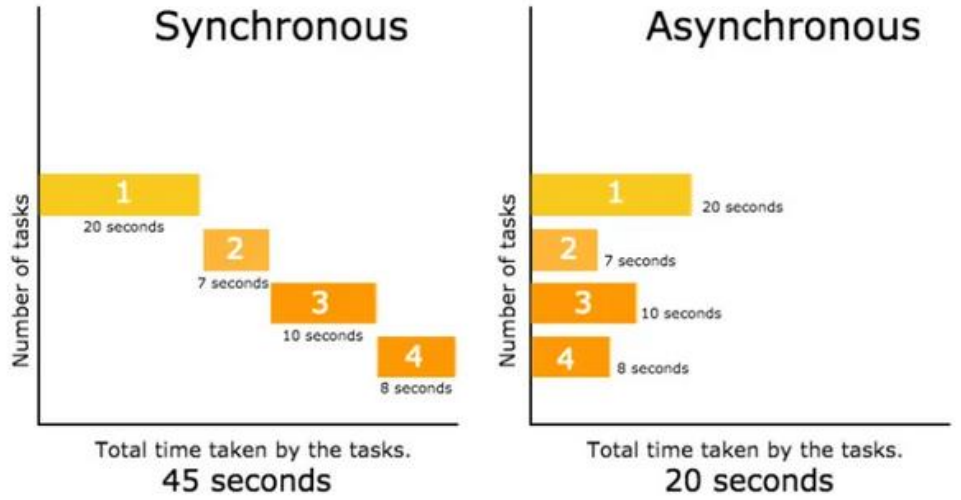
Use your own server in the cloud or onsite

Program the back- and front-end yourself

Use your own database in the cloud or onsite

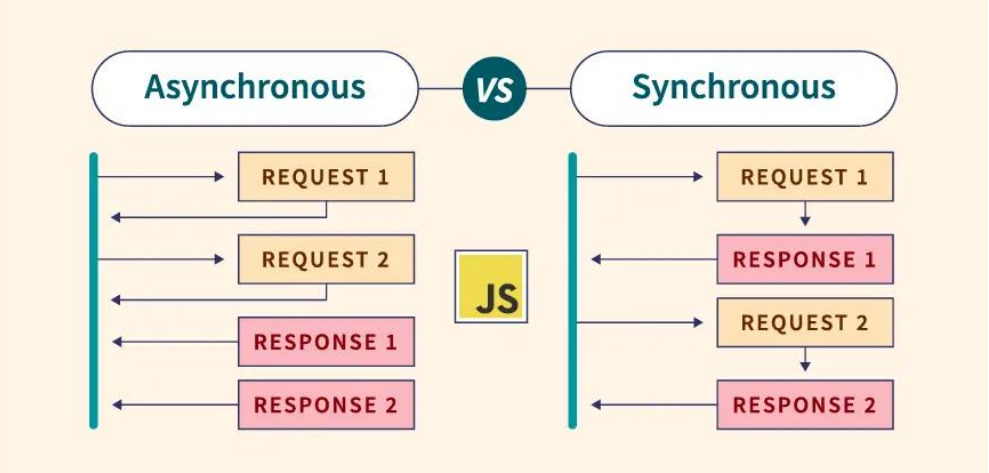
Javascript and jsPsych

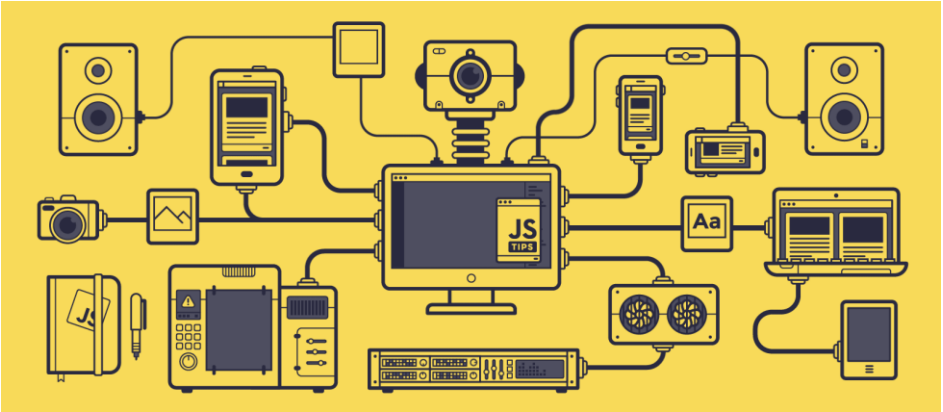
- Javascript (JS) is by far the most common language of the web.
- JS **is object-oriented** and it can be used synchronously or **asynchronously**



Javascript and jsPsych

- Javascript (JS) is by far the most common language of the web.
- JS **is object-oriented** and it can be used synchronously or **asynchronously**
- It is now used both on the **client-** and **server-**side.
- After habituation, it is very convenient for cognitive tasks because it was**created to support human-browser interactions**
- It is open-source at its core.





Javascript and jsPsych

- Like Python, JS is **extremely** modular and many functions are often explicitly loaded at runtime.
- JS has an **extremely** flexible syntax
  - Does not require prespecification
  - End of lines semicolons optional
  - Can add letter and numbers
  - Insensitive to indentation
  - Objects can contains data, code and nested objects
- JS has an **extremely** large user-base and great documentations


```
var preblock_wait = {
  type: jsPsychHtmlKeyboardResponse,
  stimulus: '<span id="clock" style="font-size:150%"">3</span>',
  on_load: function () {
    document.body.style.cursor = 'none';
    var wait_time = 3.5 * 1000; // in milliseconds
    var start_time = performance.now();
    var interval = setInterval(function () {
      var time_left = wait_time - (performance.now() - start_time);
      var minutes = Math.floor(time_left / 1000 / 60);
      var seconds = Math.floor((time_left - minutes * 1000 * 60) / 1000);
      var seconds_str = seconds.toString().padStart(1, '0');
      document.querySelector('#clock').innerHTML = seconds_str
      if (time_left <= 0) {
        document.querySelector('#clock').innerHTML = "0";
        clearInterval(interval);
      }
    }, 500)
  },
  trial_duration: 4000
}
```

Javascript and jsPsych

- But
  - JS has not been developed for scientific purposes
  - JS has been develop to manage user interactions with web pages which are very different from experiments
  - The « all-at-once » « as-soon-as-possible » philosophy of JS is not ideal for cognitive experiments
- That’s where jsPsych comes into play
  - Improved timing accuracy
  - Many useful functions for scientists
  - Great serialization of events

Large number of plugins doing that can be used to present images, sounds and a variety of HTML objects (surveys) and collect various data (keyboard, mouse, gaze, drawings)

Large number of functions commonly used in cognitive science (randomization, distribution sampling, conditional execution, loops, progress bar)

Key tools for browser and data management (full-screen methods, browser and OS identification, interaction monitoring, all the data in one place, rudimentary filtering and statistics of the data)

Javascript and jsPsych

Client code (executed by the browser)
experiment.html
.css files
.js files

The client get the .html .css .js files through a **get** request to a server route

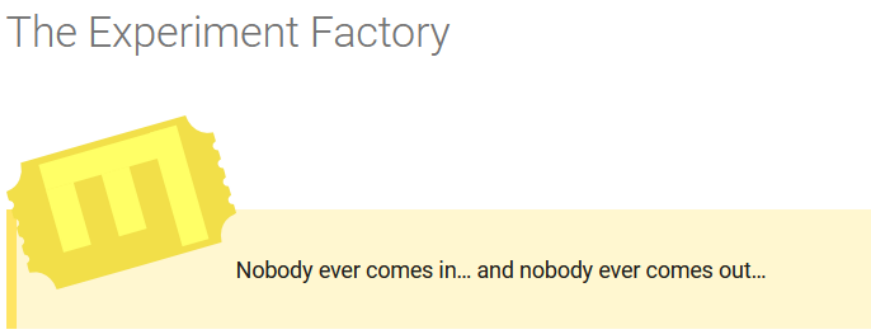
The client sends the data through a **post** request to a server route

Server code (execute by NodeJS)
app.js

Server domain
http://localhost:3000 (development)
https://example.com (production)

Server routes
ttp://localhost:3000/expNow
https://example.com/something

A few useful resources



https://expfactory.github.io/

Around 150 web experiments coded in a simple but clean fashion.

ChatGPT

Less proficient in Javascript than in other languages, but it can still be helpful for simple functions

https://www.jspsych.org/latest/

Good documentation (see also thorough YT tutorials)

https://d3js.org/

Very useful for .SVG manipulations

A few advices based on experience

- Restrict your experiment to a few common browsers (Firefox, Edge, Chrome/Chromium)

⇒ Test it on different browser / OS combinations

- Dissociate semantics from logics ++

⇒ Much easier to make adjustments when all text and parameters are in a separate file

⇒ Necessary for localization (i.e., multiple languages)

- If you are not comfortable with CSS/HTML for display, use vectorial (.svg) files and animate them with D3.js

- Make good use of URL parameters to configure your task (subject id, version, debug mode, etc.)

- When you are stuck with an experiment builder (e.g. Gorilla, LabJS, PsychoPy), it is probably time to start Javascript!

- Modern firewall are stochastic and « sniff » packets, which slows them down.

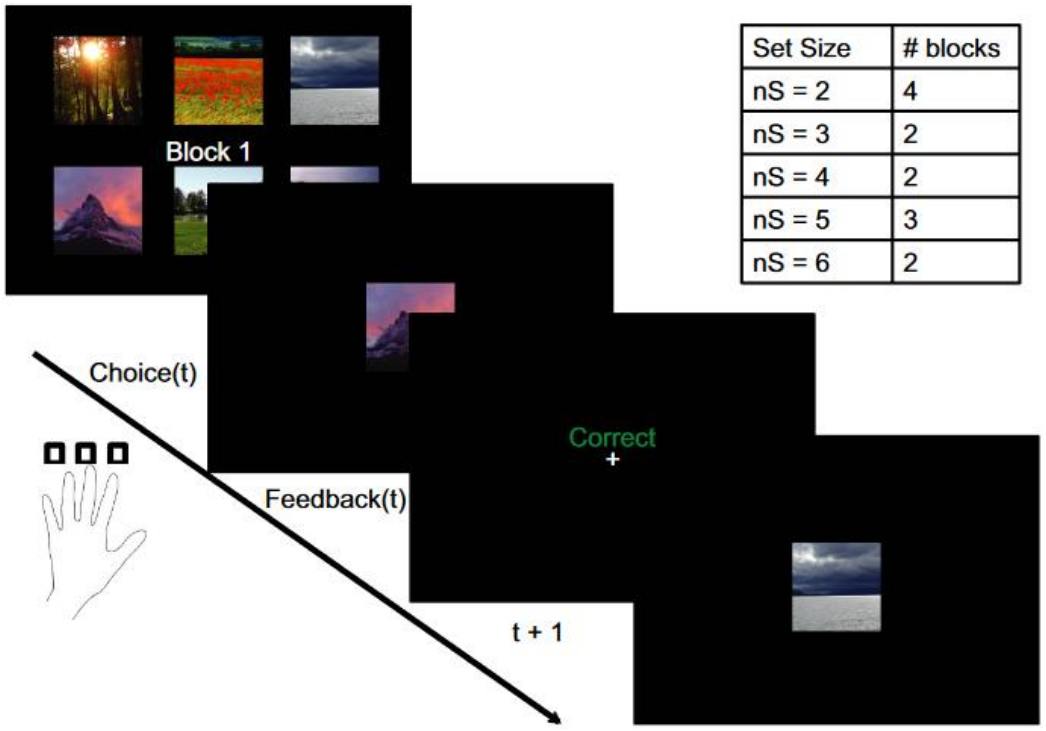
⇒ Make sure your experiment is robust to the order in which data comes in!

- Reaching participants behind organization firewalls can be tricky (e.g. hospitals, companies)

⇒ Consider packaging in a portable app

- In general, what works on your network may fail on another if you don’t understand your code…

Hands' on



Set Size	# blocks
nS = 2	4
nS = 3	2
nS = 4	2
nS = 5	3
nS = 6	2

Working Memory Contributions to Reinforcement Learning Impairments in Schizophrenia

Anne G.E. Collins,[1] Jaime K. Brown,[2] James M. Gold,[2] James A. Waltz,[2] and Michael J. Frank[1]
[1]Department of Cognitive, Linguistics, and Psychological Sciences, Brown University, Providence, Rhode Island 02912, and [2]Maryland Psychiatric Research Center, Department of Psychiatry, University of Maryland School of Medicine, Baltimore, Maryland 21201

Idea of the task

→ Small set sizes can be learned with (prefrontal) WM mechanisms only

→ Big set sizes require the contribution of (nigrostriatal) RL mechanisms too

Hands’ on

The presentation continues on your laptops

https://robustcircuit.eu/html/crnl2024-slides.html

You can also find it by browsing the website (Teaching tab)

The slides of this talk are also available