

App Dev

Stefano Balietti

Center for European Social Science Research at Mannheim University (MZES)
Alfred-Weber Institute of Economics at Heidelberg University

@balietti | stefanobalietti.com | @nodegameorg | nodegame.org



Building Digital Skills: 5-14 May 2021, University of Luzern



Express



Outputs of the Seminar (Part 2):

1. **More Wep App:** in NodeJS/Express.
2. **Chrome extensions:** service workers, architecture and examples.
3. **Behavioral experiment/survey:** nodeGame framework.
4. **Mobile development:** overview of different solutions, hybrid apps with Apache Cordova, intro to Ionic Framework, progressive apps (PWA).

Outputs of the Seminar (Part 2):

TODAY

1. **More Wep App:** in NodeJS/Express.
2. **Chrome extensions:** service workers, architecture and examples.
3. **Behavioral experiment/survey:** nodeGame framework.
4. **Mobile development:** overview of different solutions, hybrid apps with Apache Cordova, intro to Ionic Framework, progressive apps (PWA).

Cloud Providers

[Netlify](#) (PaaS): front-end development

[Vercel](#) (PaaS): in-between Netlify and Heroku

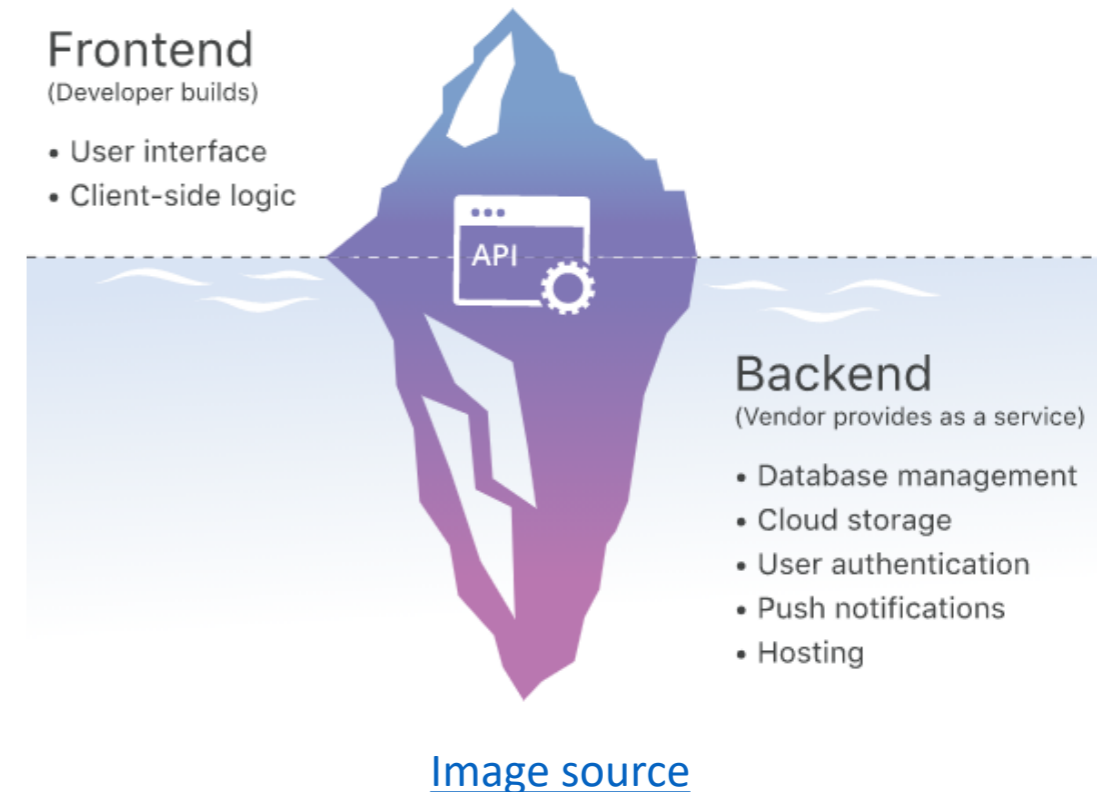
[Heroku](#) (PaaS): back-end development (apps sleep!)

[Digital Ocean](#) (BaaS and PaaS): you want to maintain your own virtual server at a very reasonable price.

[Firebase](#) (BaaS): access to Google services

[Azure](#) (BaaS): access to Microsoft services

[AWS](#) (BaaS): access to Amazon services



Securing Express

Express highlights in this page the best security practice

<https://expressjs.com/en/advanced/best-practice-security.html>

See exercise: `7_secure.js`

Securing Express

1. Never use *deprecated* or *vulnerable* versions of Express

Keep an eye on the security update [page](#)

Securing Express

2. Always use *TLS* (Transport Layer Security), enabling **https://**

TLS encrypts all data before it is sent from the client to the server

Add a route to redirect all HTTP traffic to HTTPS

Remember! POST requests do not encrypt data, more secure than GET:

- are not cached/bookmarked/browser history
- much larger data length restriction (~8Kb vs ~2Gb, but can be configured)

Securing Express

3. In doubt, put the **Helmet** on!

Helmet is a package to automatically configure Express with a higher level of security

Default security can be too tight, i.e., disabling all scripts without a hash or a nonce attribute

At least remove the "Powered by Express" header.

A Deeper Look at the Headers

Open the DevTools, click on Network tab and click on a resource (e.g., style_input.css)

WITH HELMET

The screenshot shows the Chrome DevTools Network tab with the 'Headers' sub-tab selected. The resource 'style_inputs.css' is selected in the left sidebar. The 'General' section of the headers is expanded, showing the following information:

- Request URL: `https://localhost:3000/css/style_inputs.css`
- Request Method: `GET`
- Status Code: `200 OK`
- Remote Address: `[::1]:3000`
- Referrer Policy: `no-referrer` (circled in orange)

The 'Response Headers' section is also expanded, showing the following headers:

- Accept-Ranges: `bytes`
- Access-Control-Allow-Headers: `Origin,X-Requested-With,Content-Type,Accept,content-type,application/json`
- Access-Control-Allow-Origin: `http://localhost:3000`
- Cache-Control: `public, max-age=0`
- Connection: `keep-alive`
- Content-Length: `867`
- Content-Type: `text/css; charset=UTF-8`
- Date: `Tue, 11 May 2021 07:45:04 GMT`
- ETag: `W/"363-17939670722"`

WITHOUT HELMET

The screenshot shows the Chrome DevTools Network tab with the 'Headers' sub-tab selected. The resource 'style_inputs.css' is selected in the left sidebar. The 'General' section of the headers is expanded, showing the following information:

- Request Method: `GET`
- Status Code: `304 Not Modified` (circled in orange)
- Remote Address: `[::1]:3000`
- Referrer Policy: `strict-origin-when-cross-origin` (circled in orange)

The 'Response Headers' section is also expanded, showing the following headers:

- Accept-Ranges: `bytes`
- Cache-Control: `public, max-age=0`
- Connection: `keep-alive`
- Date: `Tue, 11 May 2021 07:41:25 GMT`
- ETag: `W/"363-17939670722"`
- Keep-Alive: `timeout=5`
- Last-Modified: `Tue, 04 May 2021 22:02:03 GMT`
- X-Powered-By: `Express` (circled in orange)

Scroll down for more headers

Securing Express

4. Prevent brute-force attacks with a rate-limiter package

The [rate-limiter-flexible](#) package for instance lets you define a number of **points** that can be consumed within a given **duration**. You can associate actions to point consumption (to a logged user or to a IP address) and when it goes to zero access to resource is denied.

Securing Express

5. Whitelist IPs that can access the API

- Setting up **CORS**
- Manually **checking IP** access

Securing Express

5. Whitelist IPs that can access the API

- Setting up **CORS**:

- The [cors package](#) handles it nicely
- Can specify address, protocol, and port.
- The web server still receives and responds to requests normally.
- The browser will hide the responses if the CORS policy is not respected.

- Manually **checking IP** access:

- Completely prevent access to the resource
- IP can be masked by a proxy (e.g., Nginx) check headers 'x-real-ip'

Securing Express

6. Create a **strong access key** for the API

Use the [crypto module](#) or the [uuid package](#)

Never store the password in plain, neither in separate file nor hardcoded in code

Create an hash that can be stored outside of codebase (e.g., database or fs)

Load the hash in memory and compare incoming requests

The [bcrypt](#) module is recommended for hashing

Must use TSL to encrypt API key in incoming requests.

Securely Authenticating Users

5. If you are dealing with authenticating users in the browser [JSON Web Tokens](#) are recommended. Playground: <https://jwt.io/>

- Store them as cookies and use the HttpOnly tag.
- httpOnly tag makes the cookie unavailable to JavaScript, they are just sent with the Headers on every request.

Securely Authenticating Users

5. If you are dealing with authenticating users in the browser [JSON Web Tokens](#) are recommended. Playground: <https://jwt.io/>

- Store them as cookies and use the HttpOnly tag.
- httpOnly tag makes the cookie unavailable to JavaScript, they are just sent with the Headers on every request.

Or you may use a **plugin** auth:


- <https://auth0.com/> (Max 7000 active users for free)
- <https://magic.link/> (Max 100 active users for free)

Useful Packages

- The [passport package](#) makes it easier to integrate different authorization methods (including oauth)
- The [pm2 package](#) spawns a NodeJS child process in the background **(MUST)**.
If you run the Express server yourself you need to make sure the server is kept-alive after you close the connection/terminal.
Pm2 will restart the server upon errors, handle memory limits, and help you configure the node.js process
- The [nodemon package](#) will watch your files and restart your server whenever there is a change. Very very useful when developing a server application.
- The [dotenv package](#) is a simple package to load data (e.g., passwords and keys) into the NodeJS process

Test APIs

- <https://github.com/public-apis/public-apis>

 Cryptocurrency

API	Description	Auth	HTTPS	CORS
Binance	Exchange for Trading Cryptocurrencies based in China	apiKey	Yes	Unknown
BitcoinAverage	Digital Asset Price Data for the blockchain industry	apiKey	Yes	Unknown
BitcoinCharts	Financial and Technical Data related to the Bitcoin Network	No	Yes	Unknown
Bitfinex	Cryptocurrency Trading Platform	apiKey	Yes	Unknown
Bitmex	Real-Time Cryptocurrency derivatives trading platform based in Hong Kong	apiKey	Yes	Unknown
Bittrex	Next Generation Crypto Trading Platform	apiKey	Yes	Unknown
Block	Bitcoin Payment, Wallet & Transaction Data	apiKey	Yes	Unknown
Blockchain	Bitcoin Payment, Wallet & Transaction Data	No	Yes	Unknown
BlockFacts	Real-time crypto data from multiple exchanges via a single unified API, and much more	apiKey	Yes	Unknown

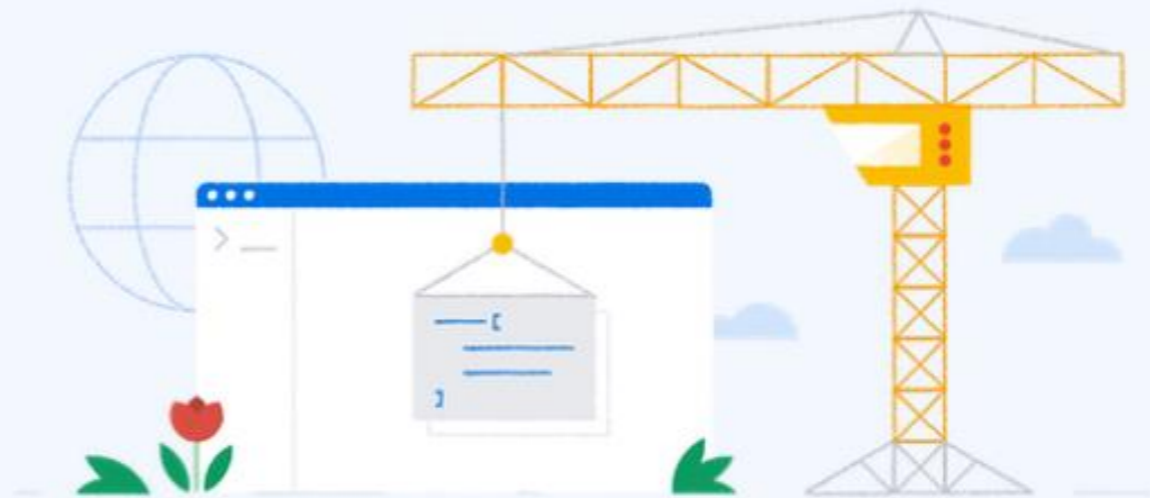
- Animals
- Anime
- Anti-Malware
- Art & Design
- Books
- Business
- Calendar
- Cloud Storage & File Sharing
- Continuous Integration
- Cryptocurrency
- Currency Exchange
- Data Validation
- Development
- Dictionaries
- Documents & Productivity
- Environment
- Events
- Finance
- Food & Drink
- Games & Comics
- Geocoding
- Government
- Health
- Jobs
- Machine Learning
- Music
- News
- Open Data
- Open Source Projects
- Patent
- Personality
- Phone
- Photography
- Science & Math
- Security
- Shopping
- Social
- Sports & Fitness
- Test Data
- Text Analysis
- Tracking
- Transportation
- URL Shorteners
- Vehicle
- Video
- Weather

Chrome Extensions

Welcome!

This is Chrome's official site to help you build Extensions, publish on the Chrome Web Store, optimize your website, and more.

Start building



<https://developer.chrome.com/>
<https://developer.chrome.com/docs/extensions/>

Module 7: Chrome Extension

Learning Goals

- Create a Chrome extension
- Understand service workers

Chrome Extensions



Extensions

Extensions are:

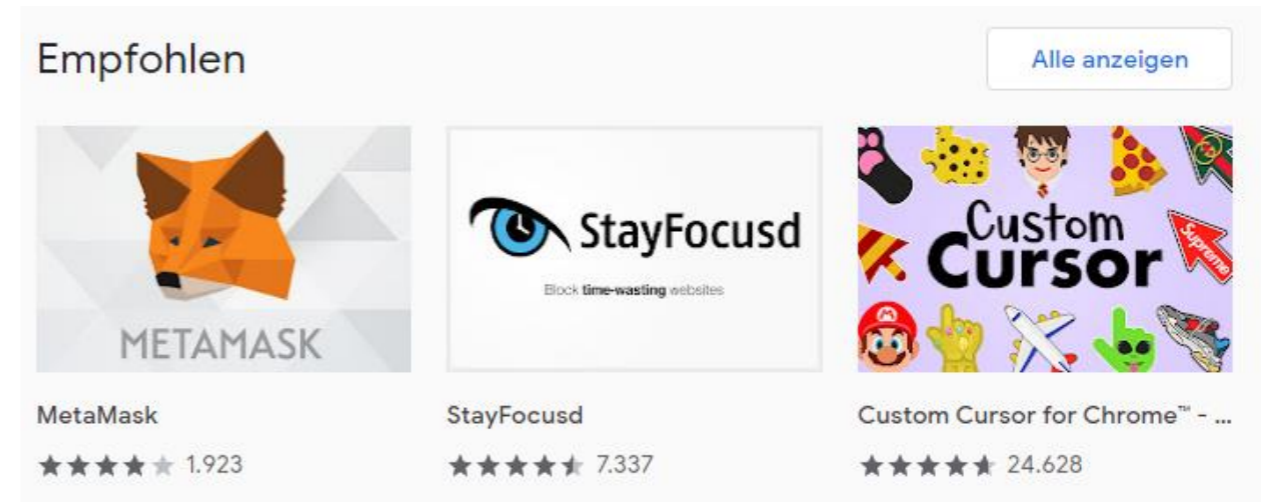
- software programs
- built on **web technologies** (HTML, CSS, and JavaScript)
- to customize the Chrome browsing experience.

Categories:

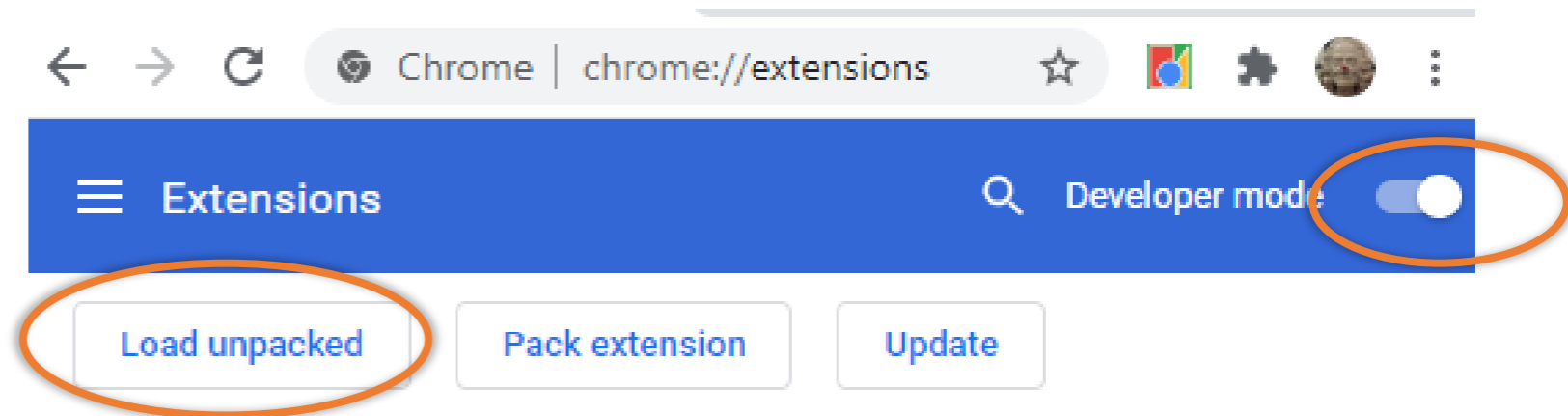
- Productivity tools
- Web page content enrichment
- Information aggregation
- Fun and games

Chrome Extensions

Packaged extensions be downloaded from the [Chrome Web Store](#)



For personal usage, development and testing, **"unpacked"** extensions can be loaded into Chrome using [extension developer mode](#).



Chrome Extensions: Components

manifest.json

```
{
  "name": "Say Hello!",
  "description": "Opens a popup that says hello!",
  "version": "1.0",
  "manifest_version": 3,
  "action": {
    "default_popup": "hello.html",
    "default_icon": "hello_extensions.png"
  },
  "commands": {
    "_execute_action": {
      "suggested_key": {
        "default": "Ctrl+Shift+F",
        "mac": "MacCtrl+Shift+F"
      },
      "description": "Opens hello.html"
    }
  }
}
```

V3 Introduced a
few months ago

Chrome Extensions: Components

manifest.json

```
{
  "name": "Say Hello!",
  "description": "Opens a popup that says hello!",
  "version": "1.0",
  "manifest_version": 3,
  "action": {
    "default_popup": "hello.html",
    "default_icon": "hello_extensions.png"
  },
  "commands": {
    "_execute_action": {
      "suggested_key": {
        "default": "Ctrl+Shift+F",
        "mac": "MacCtrl+Shift+F"
      },
      "description": "Opens hello.html"
    }
  }
}
```

V3 Introduced a few months ago

1. Action (icon)
2. Popup
3. Options
4. Background Script

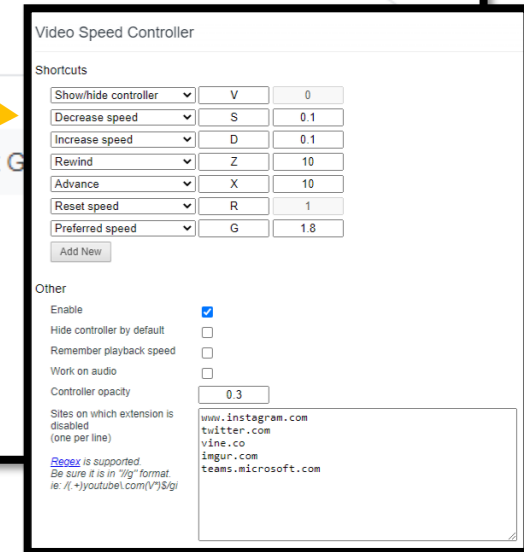
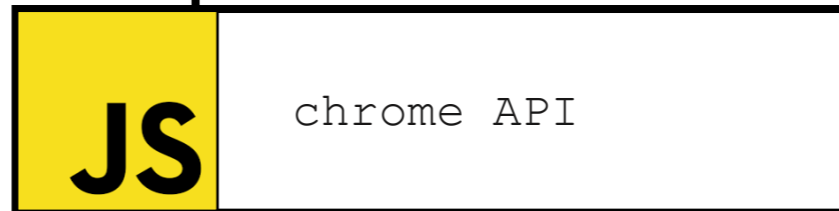
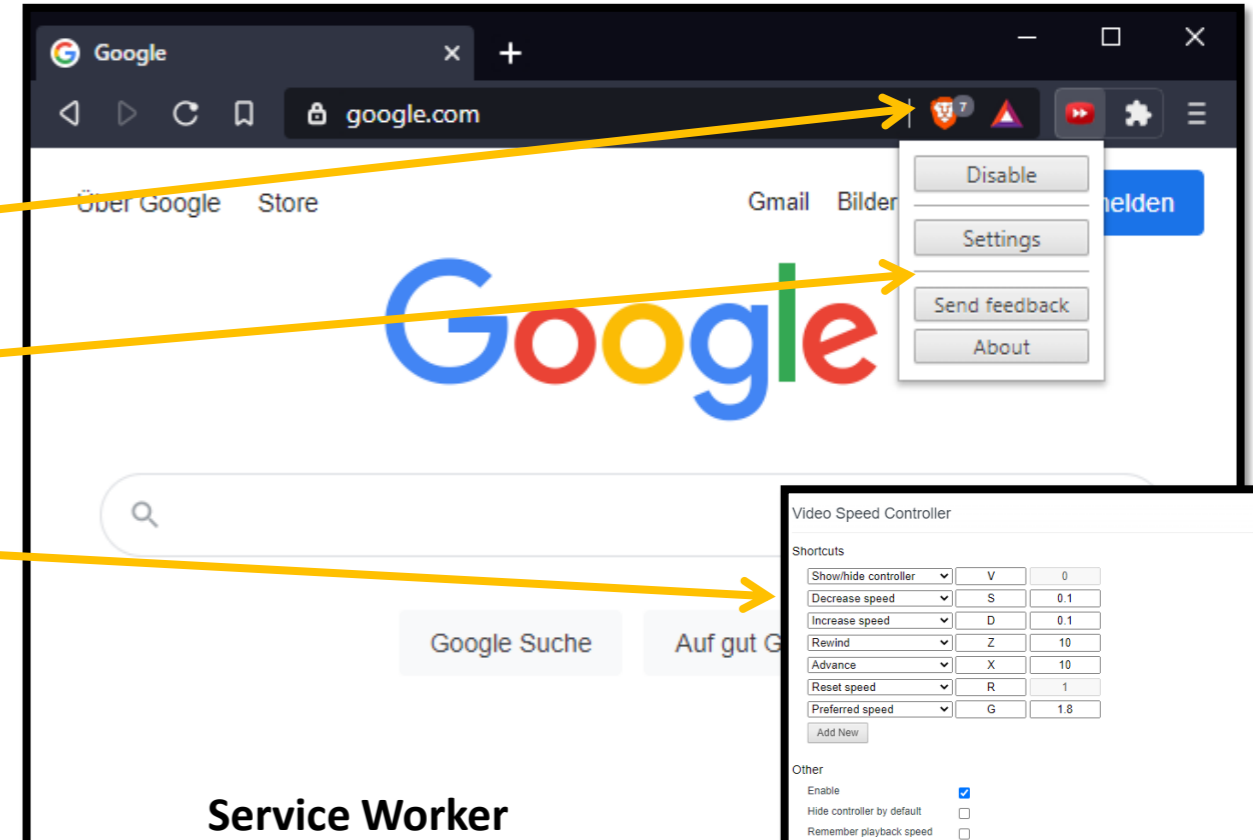
Chrome Extensions: Components

manifest.json

```
{  
  "name": "Say Hello!",  
  "description": "Opens a popup that says hello!",  
  "version": "1.0",  
  "manifest_version": 3,  
  "action": {  
    "default_popup": "hello.html",  
    "default_icon": "hello_extensions.png"  
  },  
  "commands": {  
    "_execute_action": {  
      "suggested_key": {  
        "default": "Ctrl+Shift+F",  
        "mac": "MacCtrl+Shift+F"  
      },  
      "description": "Opens hello.html"  
    }  
  }  
}
```

V3 Introduced a few months ago

1. Action (icon)
2. Popup
3. Options
4. Background Script



Service Workers

Service workers are JS processes ~~running~~ lurking in the background waiting for browser events

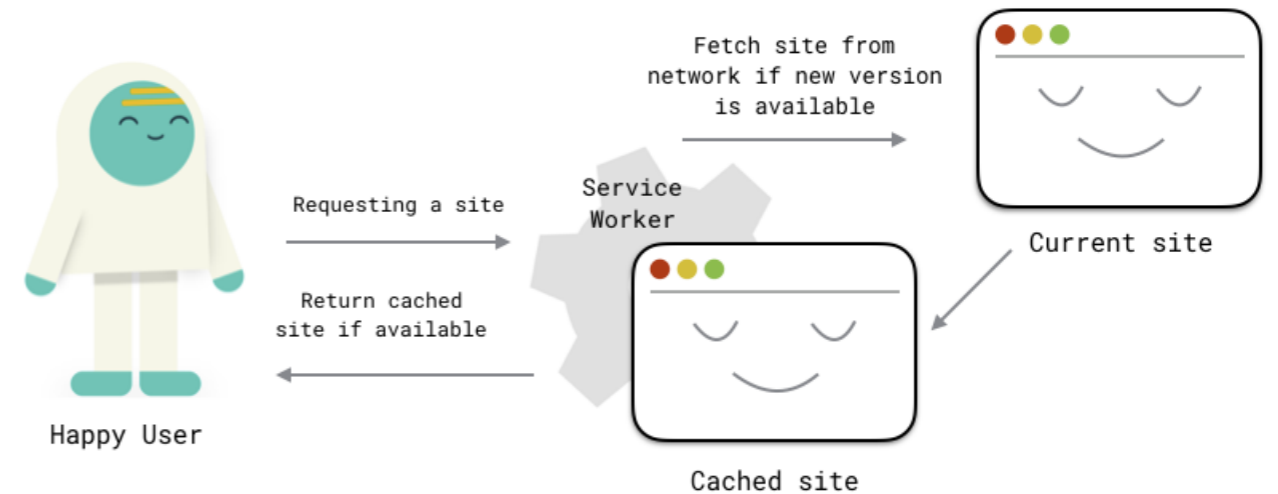
They are NOT associated with any tab

They require **HTTPS** to be executed

They are the foundation of **Progressive Web Apps (PWA)**, which can be installed on any device providing mobile-like experience

You may have some installed on your browser:

`chrome://inspect/#service-workers`



[Image source](#)

- <https://jakearchibald.github.io/isserviceworkerready/>
- <https://w3c.github.io/ServiceWorker/>

Hello Worlds Extension

- **Hello World:** opens a small popup that cheers you up
- **Hello World: Tab:** opens a new tab that cheers you up
- **Changeback:** Changes the background color of the page you are visiting

Differences from Standard Web Dev



Extensions

Generally not allowed to execute JS code directly

- **popup, options:** all `<script>` tags must have **src** attribute
- **background:** cannot execute JS directly on a tab, must inject it using the `scripting` API
- **permissions (1/2):** a JS command may either throw an error or return an incomplete output if the *permissions* are not correctly specified in *manifest.json*.
- **permissions (2/2):** to inject JS code on a page you need to add the host to the *host_permissions* in *manifest.json*

Purpose: increase security and better monitor/audit packaged extensions

Debugging

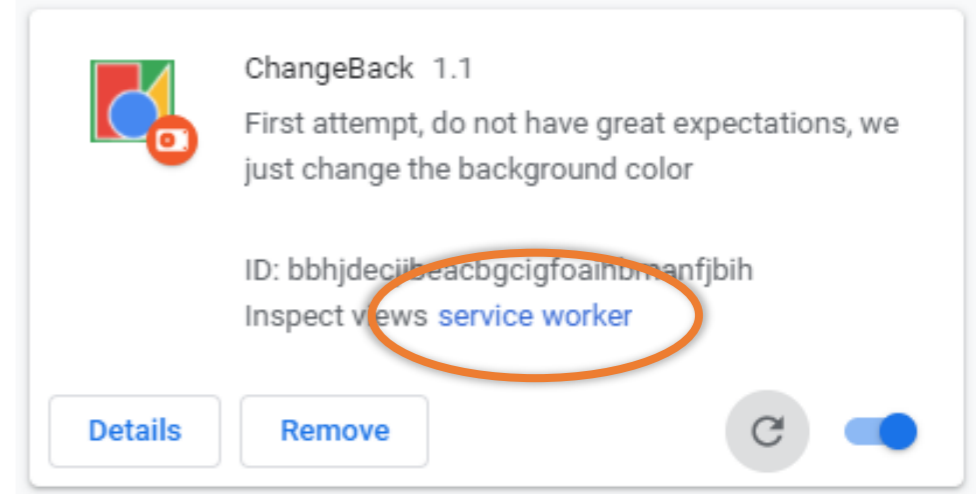
Debugging is similar as to normal web page programming, but not identical, *depending on the component*.

Popup and options: right click on them and select "Inspect:" to open a new DevTools

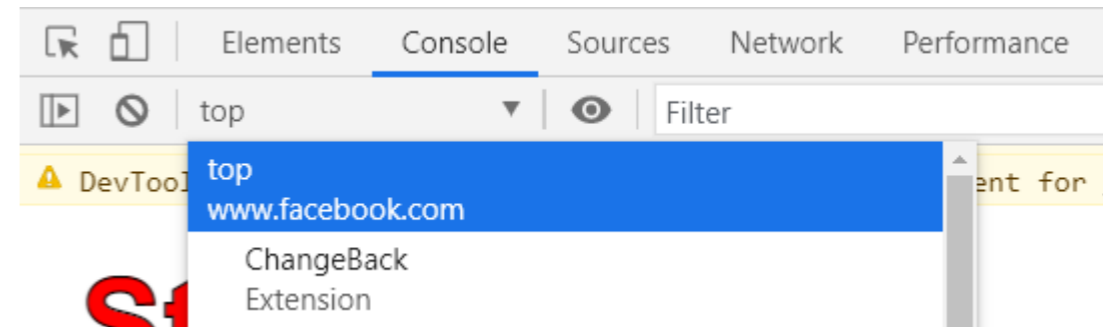
Background script: click Inspect views from the extensions tab to open a new DevTools

Injected scripts via `scripting` API: standard DevTools, however pay attention to the context of execution.

Debug background script



Context of execution of devtool console



Project: Post Block



Social media expose us to a lot of unwanted social media posts
It could be from our own friends or it could be sponsored or suggested content
Let's write a Chrome extension Post Block to control what posts we see

Project: Post Block



Social media expose us to a lot of unwanted social media posts
It could be from our own friends or it could be sponsored or suggested content
Let's write a Chrome extension Post Block to control what posts we see



Volvo Car Deutschland ✓

Sponsored · 🌐

Inkl. Versicherung, Wartung, Verschleiß, Steuern und Pannendienst. Das Auto-Abo von Volvo.

See Translation

Module 7: Resources

<https://developer.chrome.com/docs/extensions/mv3/>

<https://developer.chrome.com/docs/extensions/mv3/manifest/>

<https://developer.chrome.com/docs/extensions/reference/>

[https://developer.mozilla.org/en-US/docs/Web/API/Service Worker API/Using Service Workers](https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers)

nodeGame: Online Real-Time Synchronous Experiments



nodeGame.org

nodeGame: Online Real-Time Synchronous Experiments



nodeGame.org

- **Powerful API** to customize experiments
- Integrated JS **database**
- Fast and highly **scalable**
- Game **levels**
- **Modular** design (games, widgets, window)
- **Well-documented** (and active Forum)
- Integration with **Amazon Mechanical Turk**
- Digital Ocean Cloud **One Click Install**

What Is an Experiment?



What Is an Experiment?



- An experiment is a ***methodological procedure*** carried out with the goal of verifying, falsifying, or establishing the validity of a hypothesis.
- A ***test*** under ***controlled conditions*** that is made to demonstrate a known truth, examine the validity of a hypothesis, or determine the efficacy of something previously untried.
- An experiment is an empirical method that ***arbitrates between competing hypotheses***.

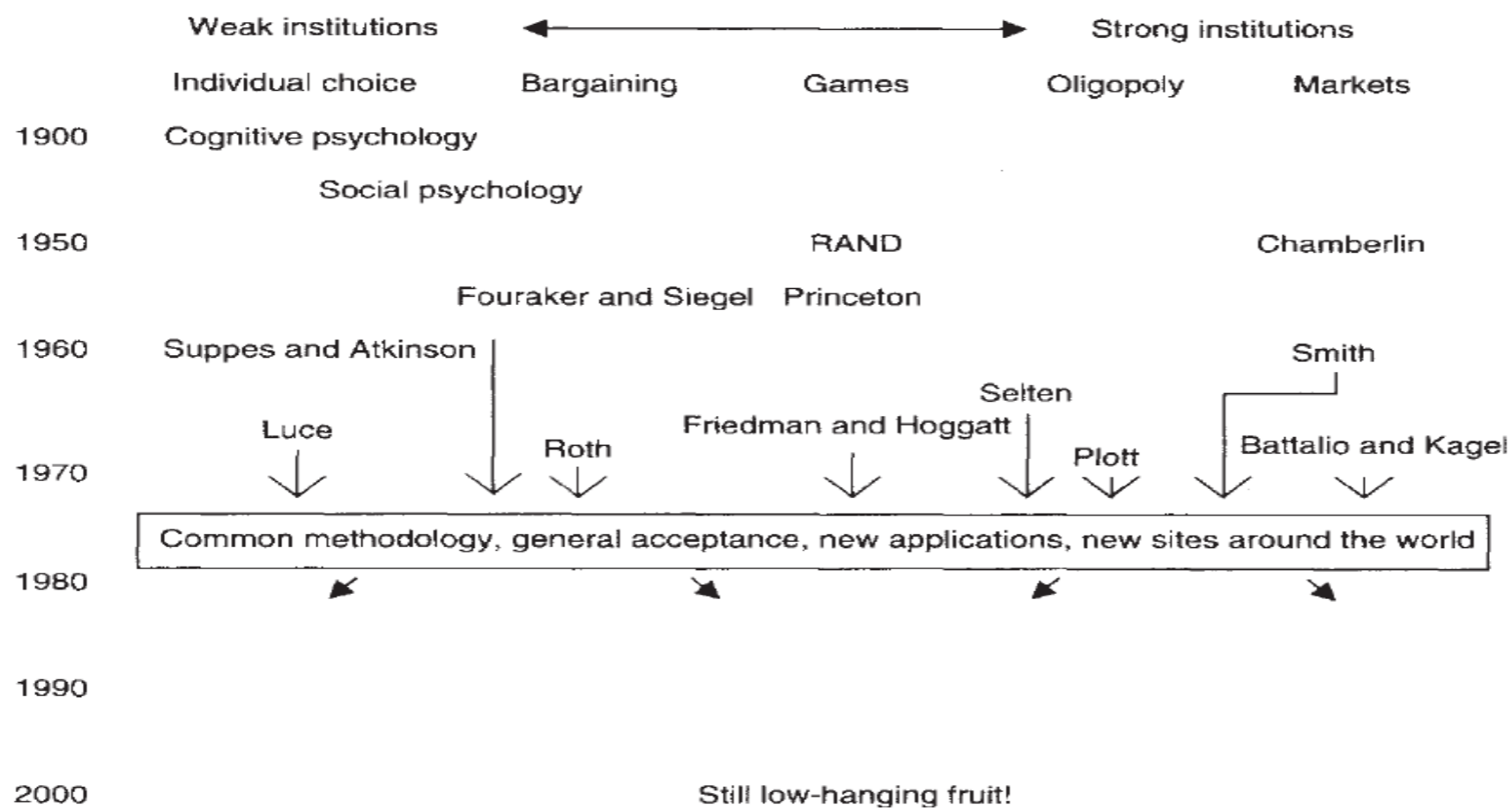
Short History of Experimental Research

“There is a property common to almost all the moral sciences, and by which they are distinguished from many of the physical; that is, that *it is seldom in our power to make experiments with them*”

John Stuart Mill, 1836



Short History of Experimental Research

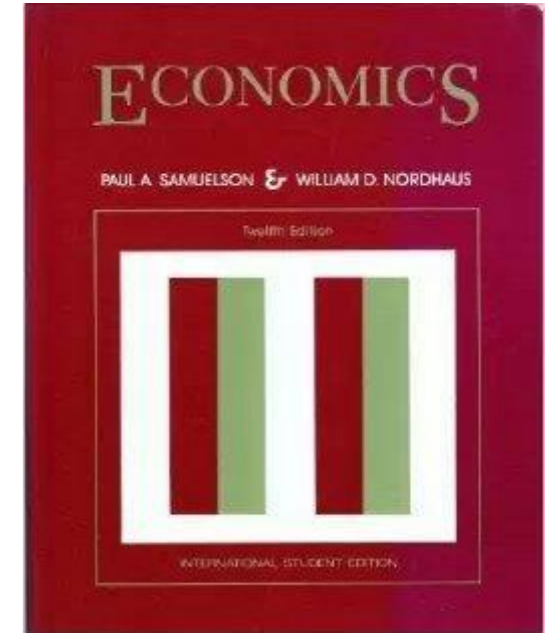


Friedman, Cassar (2004) Economists go to the laboratory. Who, what, when and why.

Short History of Experimental Research

“Economics unfortunately cannot perform the controlled experiments of chemists or biologists because [it] cannot easily control other important factors. Like astronomers or meteorologists, [it] *generally must be content largely to observe.*”

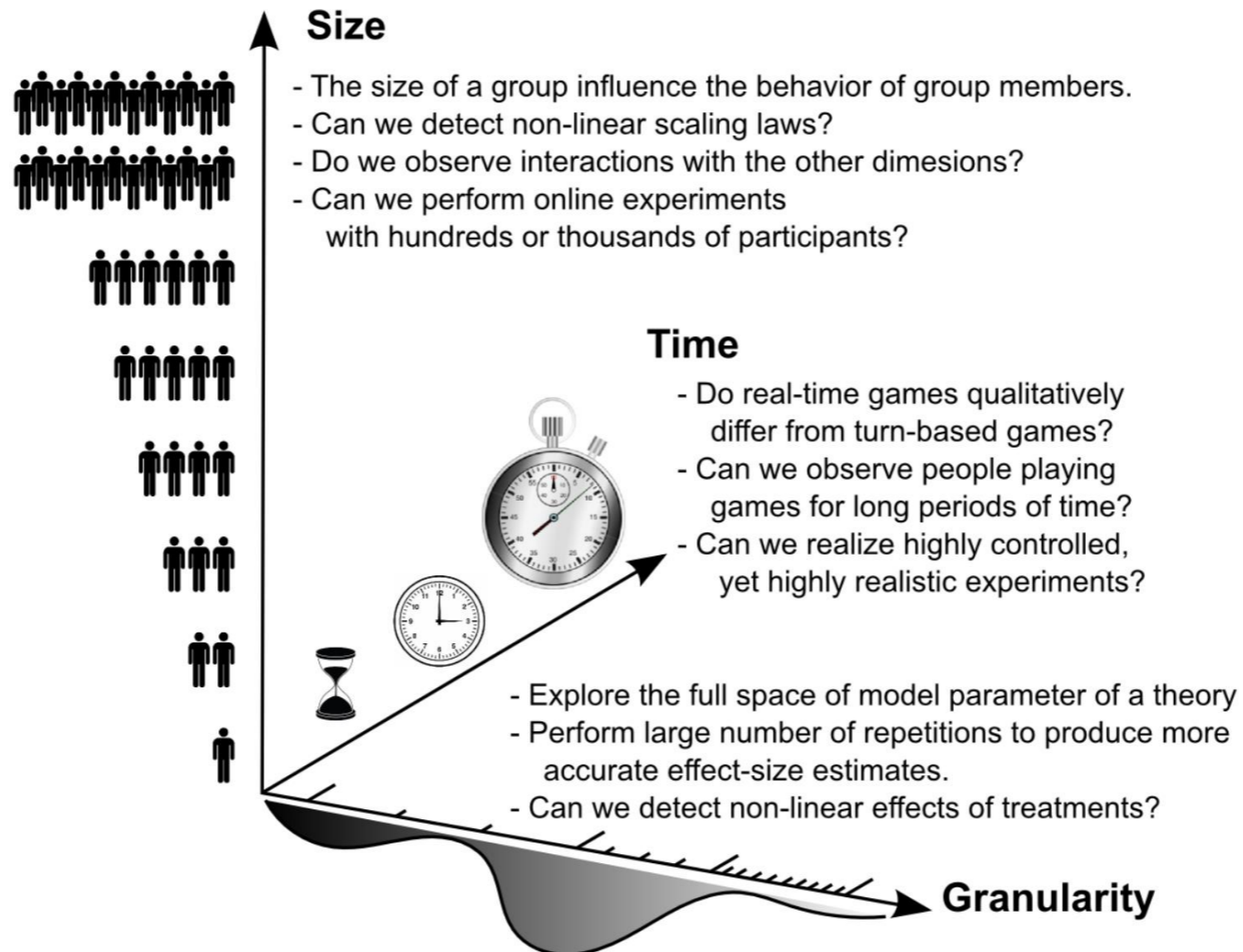
Samuelson and Nordhaus 1985, Economics Textbook (italics added)



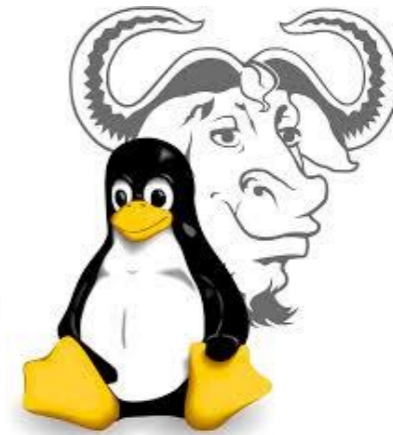
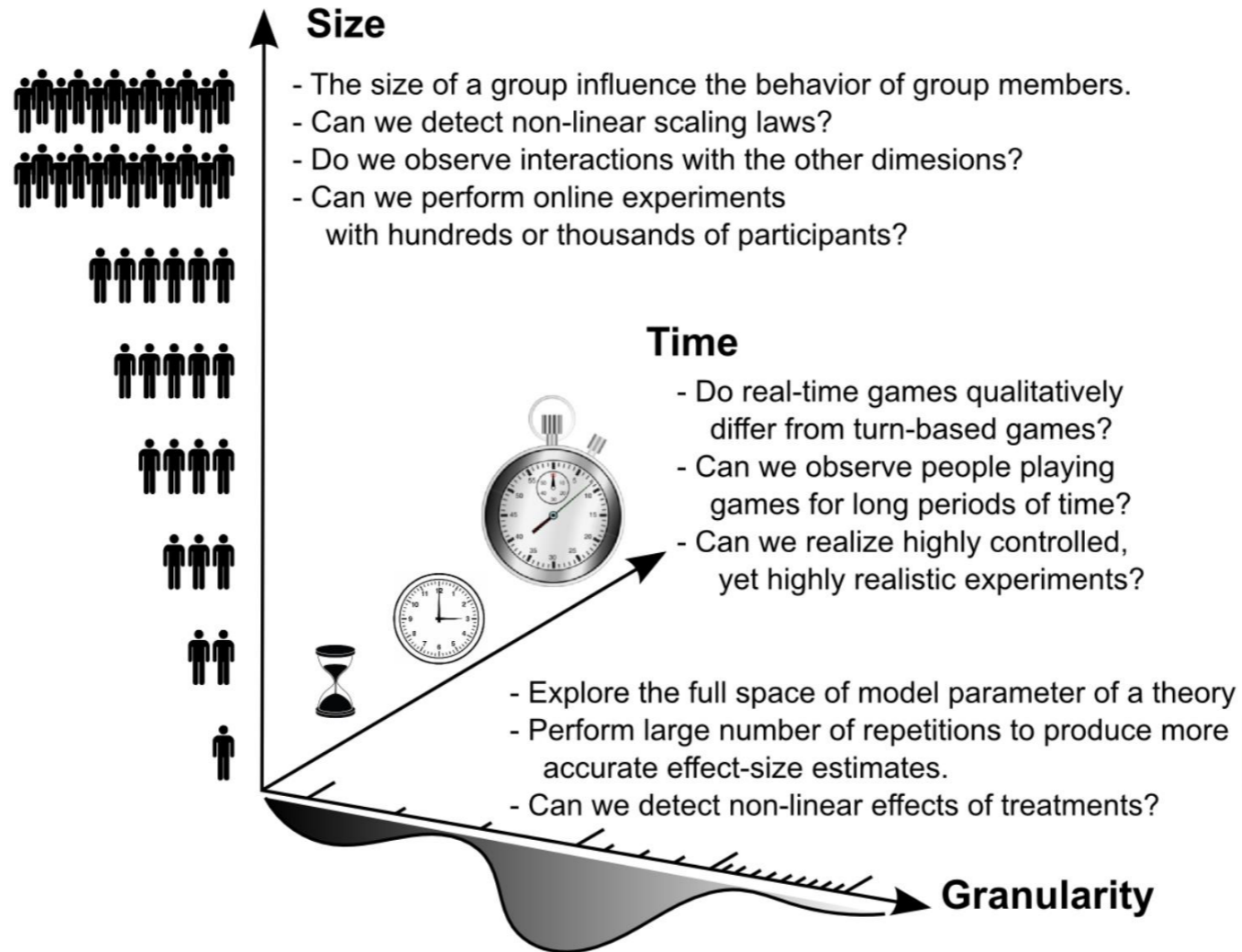
“There is no laboratory in which economists can test their hypotheses”

1993

Goals for Present and Future Behavioral Research



Goals for Present and Future Behavioral Research



express

Examples of Online Experiments

- Survey Experiments
- Game-Based Asynchronous Experiments
- Game-Based Synchronous Experiments

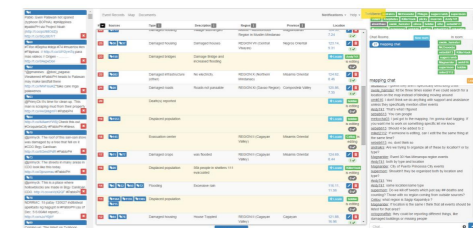
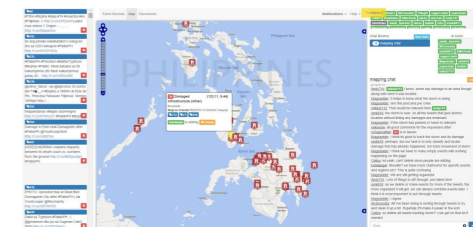
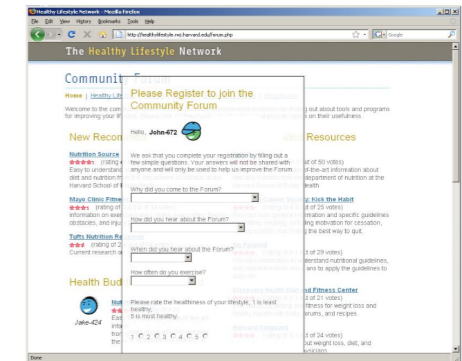
Income inequality has increased dramatically in the United States since 1980. Incomes of poorer and middle-income families have grown very little while top incomes have grown a lot.

How would YOU be doing if inequality had not increased?

The slider below shows how much each group would make if incomes had grown by the same percentage since 1980 for all groups: the poor, the middle class, and the rich. Use the slider to answer the questions below.



A household making **\$25,800** today would instead be making **\$35,200** if inequality had not changed since 1980. In other words, if growth had been evenly shared, this household would have earned **37% more**.



Examples of Online Experiments

Survey Experiments

Insert

a new block of questions, or
provide additional info

Measure preferences for outcome
variable at the *individual* level

Relatively simple to implement

Income Inequality has increased dramatically in the United States since 1980. Incomes of poorer and middle-income families have grown very little while top incomes have grown a lot.

How would YOU be doing if inequality had not increased?

The slider below shows how much each group would make if incomes had grown by the same percentage since 1980 for all groups: the poor, the middle class, and the rich. Use the slider to answer the questions below.



A household making **\$25,800** today would instead be making **\$35,200** if inequality had not changed since 1980. In other words, if growth had been evenly shared, this household would have earned **37% more**.

Kuziemko et al. (2015) How Elastic Are Preferences for Redistribution

Examples of Online Experiments

Game-Based Asynchronous Experiments

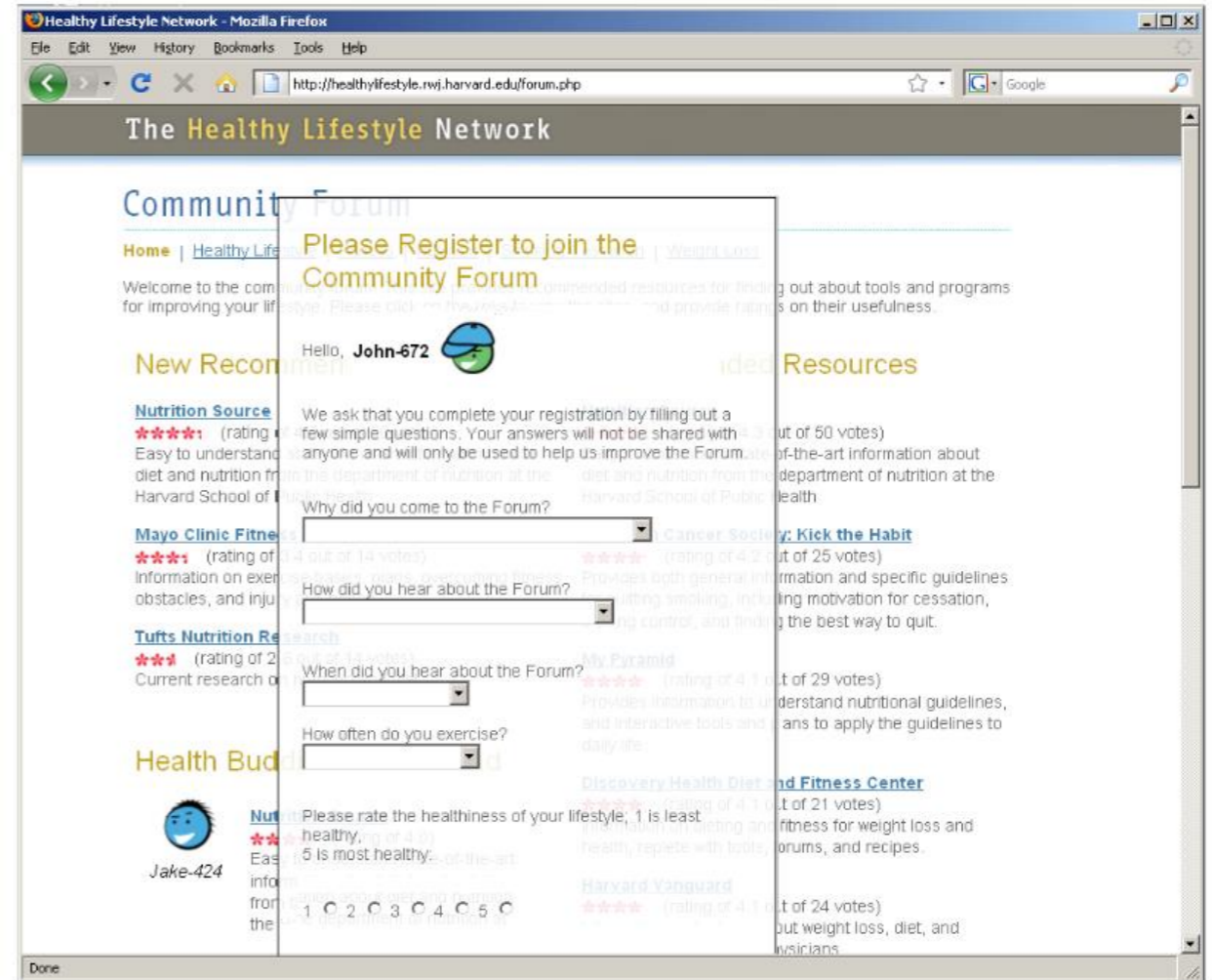
Create a common environment

where a single variable is changed (e.g. network structure)

Observe variation on outcome variable at the *group* level

Generally complex design

Can extend for long periods of time



Centola (2010) The Spread of Behavior in Online Social Networks

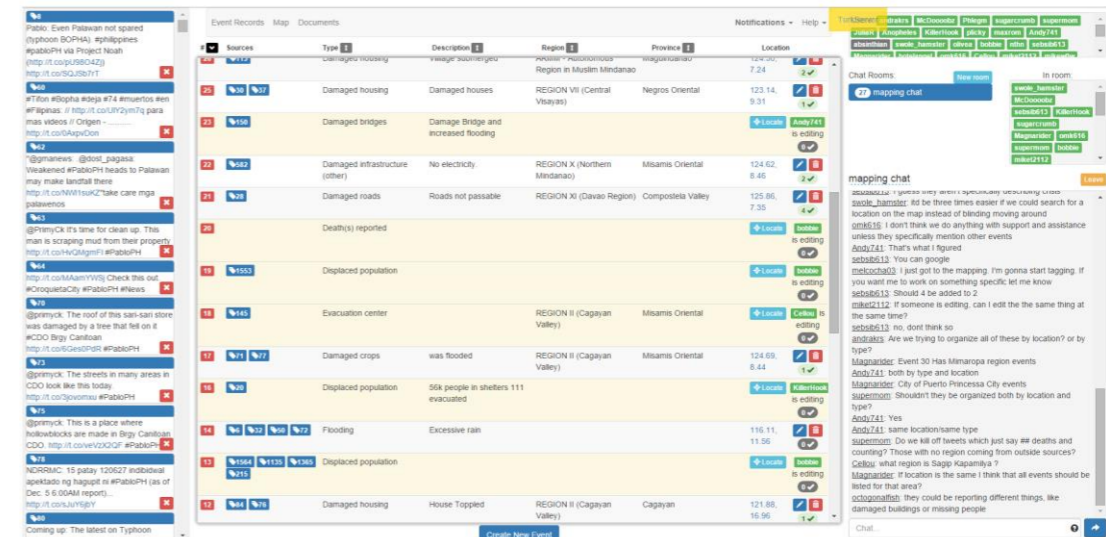
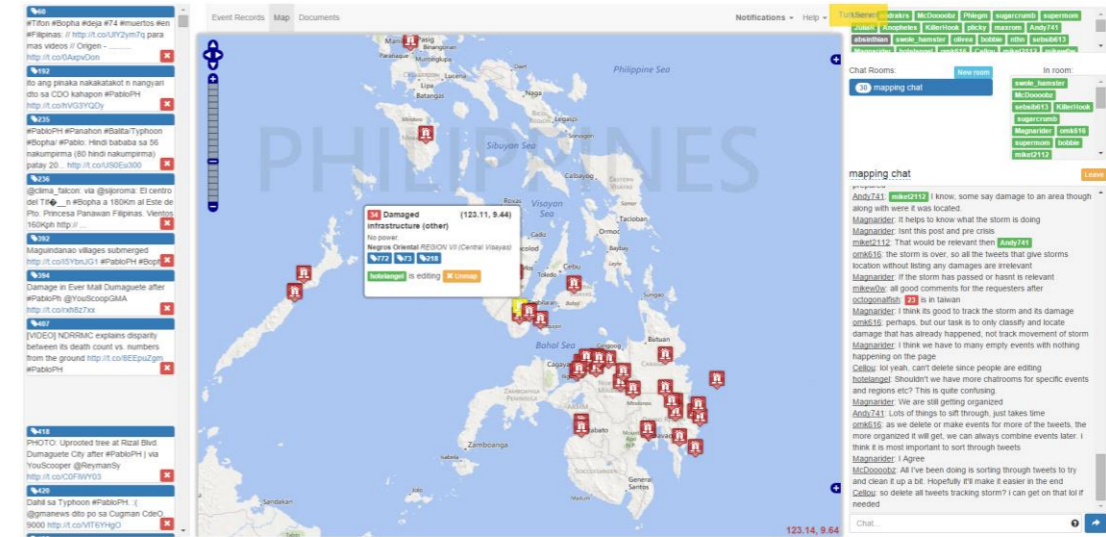
Examples of Online Experiments

Game-Based Synchronous Experiments

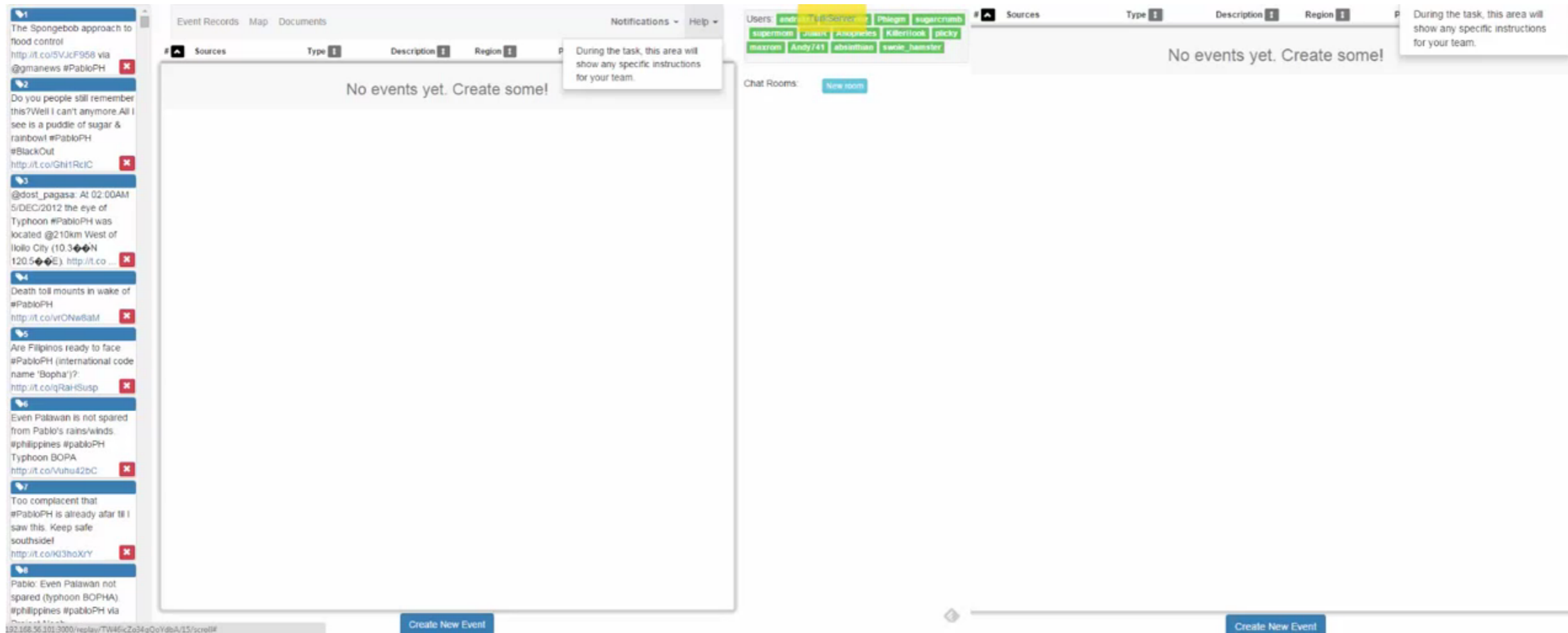
Create a common environment where a single variable is changed (e.g., group size or payoff)

Observe variation on outcome variable at the *group* level

Can be really complex or stylized



Examples of Online Experiments



Examples of Online Experiments

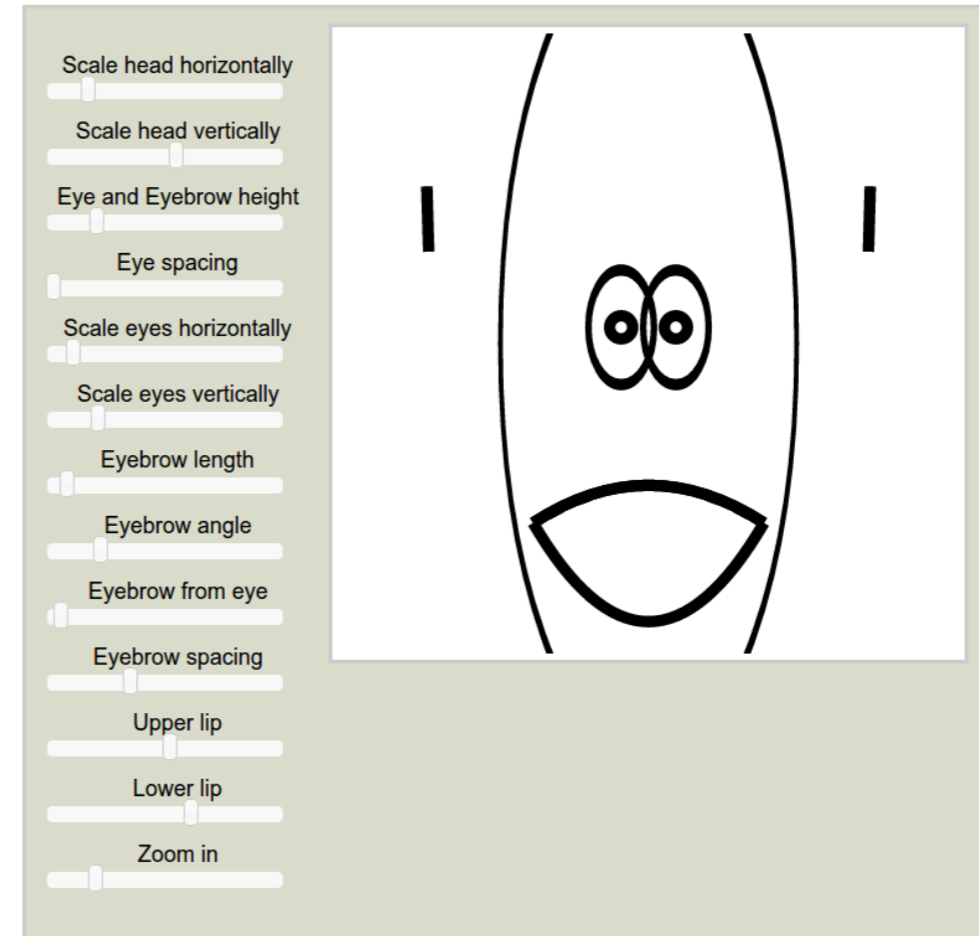
Game-Based Synchronous Experiments

Create a common environment
where a single variable is changed
(e.g., group size or payoff)

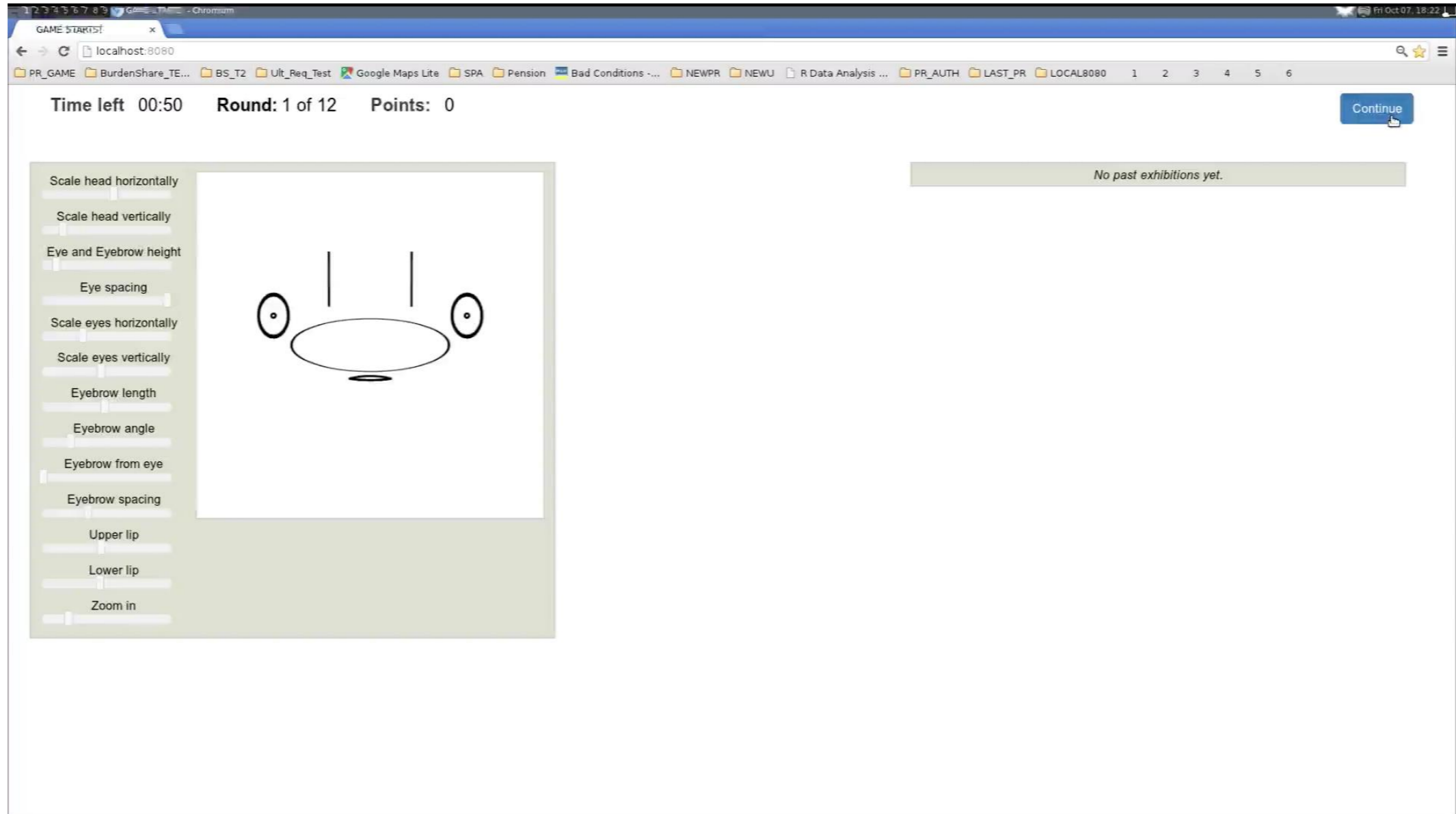
Observe variation on outcome
variable at the *group* level

Can be really complex or stylized

Time left 00:01 Round: 1 of 10 Points: 0



Art Exhibition Game



Examples of Online Experiments

Income Inequality has increased dramatically in the United States since 1980. Incomes of poorer and middle-income families have grown very little while top incomes have grown a lot.

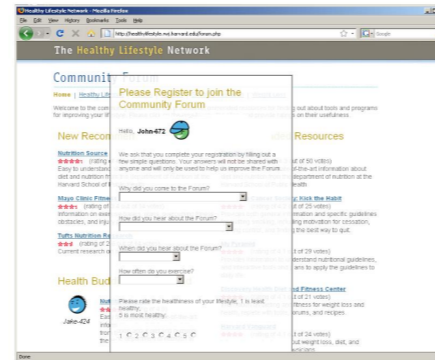
How would YOU be doing if inequality had not increased?

The slider below shows how much each group would make if incomes had grown by the same percentage since 1980 for all groups: the poor, the middle class, and the rich. Use the slider to answer the questions below.

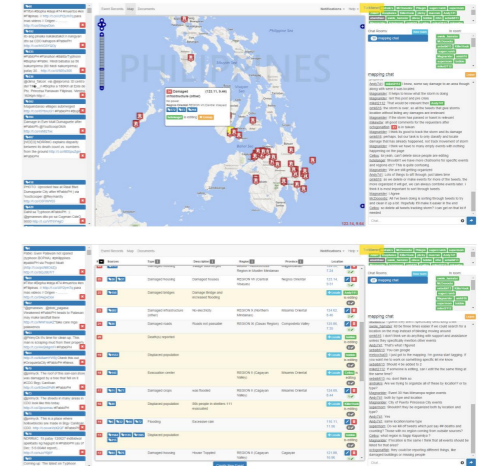


A household making **\$25,800** today would instead be making **\$35,200** if inequality had not changed since 1980. In other words, if growth had been evenly shared, this household would have earned **37% more**.

Survey
Experiments



Game-Based
Asynchronous
Experiments

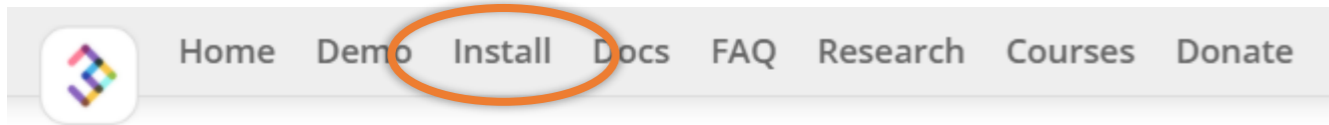


Game-Based
Synchronous
Experiments

COMPLEXITY OF EXECUTION

nodeGame Installation

<https://nodegame.org>



nodeGame 6.3.0

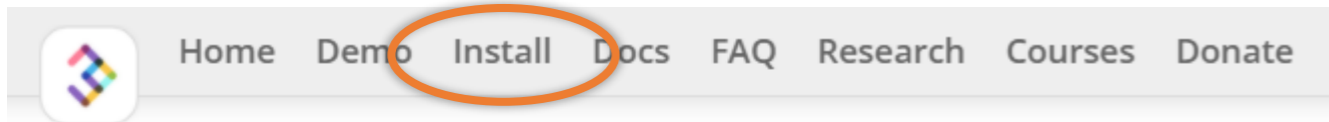
Online Real-Time Synchronous Experiments.

Fast, scalable JavaScript for large-scale, online, multiplayer, real-time games and experiments.

Demo 

nodeGame Installation

<https://nodegame.org>



nodeGame 6.3.0

Online Real-Time Synchronous Experiments.

Fast, scalable JavaScript for large-scale, online, multiplayer, real-time games and experiments.

Demo 

Installation

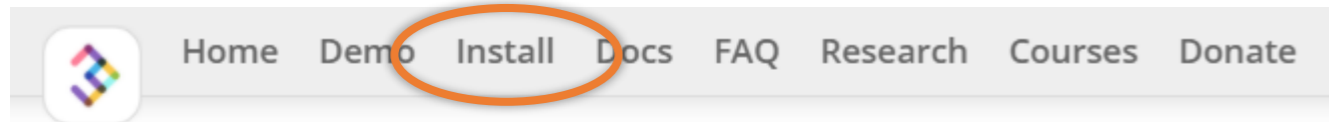
1. Download the nodegame **installer**.
2. Open a terminal and navigate to the folder where you downloaded the installer.
3. Install nodeGame with the command:

```
node nodegame-installer.js
```

4. Follow instructions on screen.

nodeGame Installation

<https://nodegame.org>



nodeGame 6.3.0

Online Real-Time Synchronous Experiments.

Fast, scalable JavaScript for large-scale, online, multiplayer, real-time games and experiments.

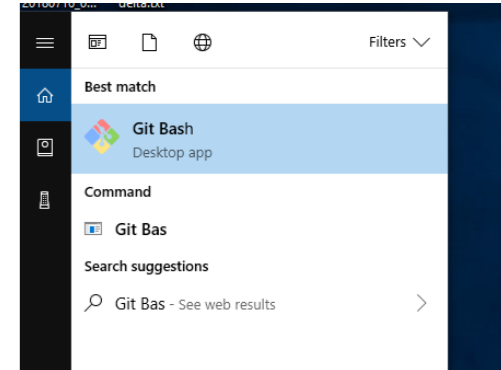
Demo 

Installation

1. Download the nodegame **installer**.
2. Open a terminal and navigate to the folder where you downloaded the installer.
3. Install nodeGame with the command:

```
node nodegame-installer.js
```

4. Follow instructions on screen.



```
balistef@mzes072 MINGW64 /tmp  
$ node nodegame-installer.js @dev
```

@dev will install latest features

List of Platforms for Online experiments

Group-Behavior

- nodeGame 😊
<https://nodeGame.org>
- Wextor (pioneer)
<https://www.wextor.eu>
- Otree (large base)
<https://www.otree.org>
- Lioness (new)
<https://lioness-lab.org>
- Breadboard (networks)
<http://breadboard.yale.edu>
- Empirica (new)
<https://empirica.ly>
- TurkServer (discontinued)
<https://turkserver.readthedocs.io>
- VeconLab (pioneer)
<http://veconlab.econ.virginia.edu/admin.htm>

Individual

- JSPsych (many plugins)
<https://www.jspsych.org>
- PsiTurk (groups possible)
<https://psiturk.org>

Recruiting Platforms

- Amazon Mechanical Turk
<https://www.mturk.com>
- TurkPrime
<https://www.turkprime.com>
- Figure Eight (ex Crowd Flower, ML)
<https://www.figure-eight.com>
- Prolific.ac
<https://www.prolific.ac>
- Reddit (for bursty access)
<https://www.reddit.com/r/WebGames>
- YouGov (country-specific)
<https://yougov.co.uk>
- PollFish (also built-in surveys)
<https://www.pollfish.com>
- Psychological Research on the Net
<http://psych.hanover.edu/research/exponnet.html>
- The Web Experiment List
<http://www.wexlist.net>
- Online Social Psychology Studies
<http://www.socialpsychology.org/expts.htm>

Citizen Science Initiatives

- Volunteer Science
<https://volunteerscience.com>
- Science@Home
<https://www.scienceathome.org>
- Zooniverse
<https://www.zooniverse.org>
- Lab in the Wild
<https://labinthewild.org>
- CitizenLab
<https://www.citizenlab.co>



Zooniverse

nodeGame Resources

- <https://www.stefanobalietti.com/teaching/online-experiments/>
- <https://nodegame.org/courses.htm>
- <https://www.youtube.com/channel/UC2esqxivayZr80QdUZujLqpA>