Gorilla in our Midst:
An online behavioral experiment builder

Alexander Anwyl-Irvine[1,2] Jessica Massonnié[3] Adam Flitton[1, 4]
Natasha Kirkham[3] and Jo Evershed[1*]

[1]Cauldron Science: Cauldron.sc
[2]MRC Cognition and Brain Sciences Unit, University of Cambridge
[3]Centre for Brain and Cognitive Development, Birkbeck College, University of London
[4]Human Behaviour and Cultural Evolution Group, University of Exeter
* corresponding author: Jo.Evershed@cauldron.sc

October 29, 2018
PREPRINT: THIS MANUSCRIPT HAS NOT UNDERGONE PEER REVIEW

+

Abstract

Behavioural researchers are increasingly conducting their studies online to gain access to large and diverse samples that would be difficult to get in a laboratory environment. However, there are technical access barriers to building experiments online, and web-browsers can present problems for consistent timing – an important issue with reaction time-sensitive measures. For example, to ensure accuracy and test-retest reliability in presentation and response recording, experimenters need a working knowledge of programming languages such as JavaScript. We review some of the previous and current tools for online behavioural research, and how well they address the issues of usability and timing. We then present The Gorilla Experiment Builder (gorilla.sc) a fully tooled experiment authoring and deployment platform, designed to resolve many timing issues, and make reliable online experimentation open and accessible to a wider range of technical abilities. In order to demonstrate the platform's aptitude for accessible, reliable and scalable research, we administered the task with a range of participant groups (primary school children and adults), settings (without supervision, at home, and under supervision, in schools and public engagement events), equipment (own computers, computer supplied by researcher), and connection types (personal internet connection, mobile phone 3G/4G). We used a simplified flanker task, taken from the Attentional Networks task (Rueda, Posner, & Rothbart, 2004). We replicated the 'conflict network' effect in all these populations, demonstrating the platform's capability to run reaction time-sensitive experiments. Unresolved limitations of running experiments online are then discussed, along with potential solutions, and some future features of the platform.

Introduction

Behavioural research and experimental psychology are increasing their use of web-browsers and the internet to reach larger (Adjerid & Kelley, 2018), and more diverse (Casler, Bickel, & Hackett, 2013) populations than has been previously feasible with lab-based methods. However, there are unique variables which are introduced when working within an online environment. The experience of the user is the result of a large number of connected technologies. Examples of this include: the server (which hosts the experiment), the internet service provider (which delivers the data), the browser (which presents the experiment to the participant and measures their responses), and the content itself – which is a mixture of media (e.g. audio/pictures/video) and text files in different programming languages (e.g. JavaScript, HTML, CSS, PHP, Java). Linking these technologies together is technically difficult, time-consuming and costly. Consequently, untilrecently, online research is often carried out – and scrutinized – by those with the resources to overcome these barriers.

The purpose of this paper is three-fold. Firstly, to explore the problems inherent to running behavioural experiments online with web programming languages, the issues this created for timing

accuracy, and recent improvements that can mitigate these issues. Secondly, to introduce Gorilla, an Online Experiment Builder that uses best practices to overcome these timing issues and makes reliable online experimentation accessible and transparent to the majority. Thirdly, to demonstrate the timing accuracy and reliability provided by Gorilla. This is achieved with data from a Flanker task – which requires high timing fidelity - collected from a wide range of participants, settings, equipment and internet connection types.

JavaScript

The primary consideration for online experimenters in the present time is JavaScript (JS), the language that is most commonly used to generate dynamic content on the web (such as an experiment). Its quirks (which are discussed later) can lead to problems with presentation time, and understanding it forms a large part of an access barrier. JS is highly dynamic - and therefore potentially unpredictable (Severance, 2012). JS was designed to be as difficult to break as possible; therefore, programming mistakes – and their consequences – can often go undetected by the designer (Richards, Lebresne, Burg, & Vitek, 2010). This is clearly not ideal for new users attempting to create controlled scientific experiments. Below we discuss two significant hurdles when building web experiments: inaccuracies in the timing of various experiment components in the browser, and the technical complexities involved in implementing an online study, including JavaScript's contributions. These complexities present an access barrier to controlled online experiments for the average behavioural researcher.

History of Timing Concerns

Timing concerns have been expressed regarding online studies (for an overview see Woods, Velasco, Levitan, Wan, and Spence (2015)), and while many of these concerns are historic for informed users – as solutions exist – they are still an issue for new users who may not be aware of them. Concerns fall into timing of stimuli – i.e. an image or sound is not presented for the duration you want – and the timing of response recording – i.e. the participant did not press a button at the time you think they did. These inaccuracies have obvious implications for behavioural research, especially those using time-based measures such as reaction time.

Several things may be driving these timing issues: firstly, in JS programs, all processes pass through a single event loop (a constantly executing list of commands to resolve). Therefore, all presentation changes are processed through this same loop, this could be: an animation frame updating, an image being rendered, a sound being produced, or an object being dragged around. Variance in the order in which computations are queued up (called the 'call stack'), due to any experiment's code competing with the hosting website and other windows, can lead to inconsistent timing. For instance, you may try and present auditory and visual stimuli at the same time, but they could end up out of synchronisation if other processes get in the way – a common manifestation of this in web videos is unsynchronised audio and video. Secondly, the current computational load on the browser will slow the event loop down; variance in timing is, therefore, dependent on different computers, browsers and computational load (Jia, Guo, Wang, & Zhang, 2018). Given the need for online research to make use of on-site computers such as in homes or in schools, this is an important issue. A laptop with a single processor, a small amount of memory, and an out-of-date web-browser is likely to struggle to present stimuli to the same accuracy as a multi-core desktop with the most recent version of Google Chrome installed. These variances can represent variance of over 100ms in presentation timing (Reimers & Stewart, 2016). Thirdly, the connection speed of the internet may also play a part – if the experiment contains lots of images or variables that must be loaded from the server during the experiment, this will increase variance in display times.

The same concerns (with the exception of connection speed) can be applied to the recording of response-times, which are dependent on a JS system called the 'event system'. When a participant presses a mouse or keyboard button, recording of these responses (often through a piece of code called an 'Event Listener') gets added to the event loop. To give a concrete example, two computers could record different times of an identical mouse response based on their individual processing loads. It must be noted that this issue is *independent* of the browser receiving an event (such as a mouse click being polled by the operating system), where there is a relatively fixed delay, shown to be equivalent to non-browser software (de Leeuw & Motz, 2016) - this receiving delay is discussed later in the paper. Timing of event recording using the browser system clock (which some JavaScript functions do) is also another source of variance – as different machines and operating systems will have different clock accuracies and update rates.

Current State of the Art

Presently, the improved processing capabilities in common browsers and computers, in concert with

improvements in web-language standards - such as HTML5 and ECMAScript 6 - offers the potential to overcome some concerns about presentation and response timings (Garaizar, Vadillo, & López-de Ipiña, 2012, 2014; Reimers & Stewart, 2015, 2016; Schmidt, 2001). This is because, in addition to standardised libraries (which improve the consistency of any potential web experiment between devices), these technologies use much more efficient interpreters, which are the elements of the browser which execute the code and implements computations. An example of this is Google's V8, which improves processing speed – and therefore the speed of the event loop – significantly (Severance, 2012). In fact, several researchers have provided evidence that response times are comparable between browser-based applications and local applications (Barnhoorn, Haasnoot, Bocanegra, & Steenbergen, 2015) even in poorly standardized domestic environments - i.e. at home (Miller, Schmidt, Kirschbaum, & Enge, 2018).

A secondary benefit of recent browser improvements is scalability. If behavioural research continues to take advantage of the capacity for big-data provided by the internet, it needs to produce scalable methods of data collection. Browsers are becoming more and more consistent in the technology they adopt, at the time of writing the standard for browser-based web apps is HTML5 and ECMAScript JavaScript. This combination, in addition to having improved timing, is also the most scalable, as it reaches the greatest number of users - with most browsers supporting them – this is in contrast with other technologies, such as Java plugins, which are becoming inconsistently supported.

Access Barriers

Often, in order to gain accurate timing and presentation, you must have a good understanding of key browser-technologies. As in any application of computer science, there are multiple methods for achieving the same goal, and these may vary in the quality and reliability of the data they produce. One of the key resources for tutorials on web-based apps – the web itself – may lead users to use out-of-date or unsupported methods; with the fast-changing and exponential browser ecosystem, this is a problem for the average behavioural researcher (Ferdman, Minkov, Bekkerman, & Gefen, 2017).This level of complexity imposes an access barrier to creating a reliable web experiment - the researcher must have an understanding of the web ecosystem they operate in and know how to navigate its problems with appropriate tools.

There are, however, tools available which lower these barriers in various ways. Libraries, such as jsPsych (de Leeuw, 2015), give a toolbox of JavaScript commands which are implemented at a higher level of abstraction - therefore relieving the user of some implementation level JavaScript knowledge. Hosting tools, like 'Just Another Tool for Online Studies' (JATOS), allow users to host JavaScript and HTML studies (Lange, Kühn, & Filevich, 2015), and present these to their participants - this enables a research-specific server to be set up. However, with JATOS you still need to know how to set it up and manage your server, which requires a considerable level of technical knowledge.

The solutions above function as 'packaged software', where the user is responsible for all levels of implementation (i.e. browser, networking, hosting, data processing) - in the behavioural research use-case this requires multiple tools to be stitched together (e.g. jsPsych in the browser and JATOS for hosting). This itself presents another access barrier, as the user then must understand – to some extent – details of the web server (for instance how many concurrent connections their hosted experiment will be able to take), hosting (the download/upload speeds), the database (where and how data will be stored, e.g. in JS object notation format, or in a relational database), and how the participants are accessing their experiment and how they are connected (e.g. through Prolific.ac or Mechanical Turk).

One way to lower these barriers is to provide a platform where all of this is managed for the user - commonly known as Software as a Service (SaaS) (Turner, Budgen, & Brereton, 2003). All of the above can be set up, monitored and updated for the experimenter, whilst also providing as consistent and reproducible environment as possible - something that is often a concern for web-research. One recent example of this is the online implementation of PsyToolkit (Stoet, 2017), where users can create, host and run experiments on a managed web server and interface - however, there is still a requirement to write out the experiment in code – representing another access limitation.

The Gorilla Experiment Builder

`www.gorilla.sc` is an online experiment builder, and its aim is to lower the barrier to access, enabling all researchers and students to run online experiments (regardless of programming and networking knowledge). As well as giving greater access to web-based experiments, it reduces the risk of introducing higher noise in data (due to misuse of browser-based technology). By lowering the barrier, Gorilla.sc aims to make online experiments available and transparent at all levels of ability. Currently, experiments have been conducted in Gorilla on a wide variety of topics, in-

cluding: cross-lingual priming (Poort & Rodd, 2017), the provision of lifestyle advice for cancer prevention (Usher-Smith et al., 2018), semantic variables and list memory (Pollock, 2018), narrative engagement (Richardson et al., 2018), trust and reputation in the sharing economy (Zloteanu, Harvey, Tuckett, & Livan, 2018), how individual's voice identities are formed (Lavan, Knight, & McGettigan, 2018), and auditory perception with degenerated music and speech (Jasmin, Dick, Holt, & Tierney, 2018). Also, several studies have pre-registered reports, including: object size and mental simulation of orientation (Chen, de Koning, & Zwaan, 2018) and the use of face regression models to study social perception (Jones, 2018). Additionally, It has also been mentioned in a paper on the gamification of cognitive tests (Lumsden, Skinner, Coyle, Lawrence, & Munafo, 2017).

Gorilla.sc provides researchers with a managed environment in which to design, host and run experiments. It is fully compliant with the EU General Data Protection Regulation (GDPR), and NIHR and BPS guidelines. A graphic user interface (GUI) is available for building questionnaires (called the 'Questionnaire Builder'), experimental tasks (the 'Task Builder') and running the logic of experiments (the 'Experiment Builder"). For instance, a series of different attention and memory tasks could be constructed with the Task Builder, and then their order of presentation is controlled with the Experiment Builder. Both are fully implemented within a web-browser and are illustrated in 1. This allows users with little or no programming experience to run online experiments, whilst controlling and monitoring presentation and response timing.

At the Experiment Builder level (Figure 1B) users can create logic for the experiment though a range of control 'nodes' that manage capabilities such as: randomisation, counterbalancing, branching, task switching, repeating and delay functions. This range of functions makes it as easy to create longitudinal studies with complex behavior. An example could be a 4 week training study with email reminders, where participants receive different tasks based on prior performance, or the experiment tree just as easily enables a one-shot between subject experiment. Additionally, Gorilla.sc includes a redirect node that allows users to redirect participants to another hosted service and then send them back again. This allows users to use the powerful Experiment Builder functionality (i.e. multi-day testing) while using a different service (such as Qualtrics) at the task or questionnaire level.

The Task Builder (Figure 1A) provides functionality at the task level. Each experimental task is separated into 'displays' that are made of a sequence of 'screens'. Each screen can be configured by the user to contain an element of a trial, be that: text, images, videos, buttons, sliders, keyboard responses, progress bars, feedback and a wide range of other stimuli and response options. The content of these areas can be either static (such as instructions text), or change on a per-trial basis (where the content is set using a spreadsheet). The presentation order of these screens are dependent on sequences defined in this same spreadsheet, where blocked or complete randomisation can take place on the trial level.

Additionally, users can extend the functionality of Gorilla through use of the scripting tools and code editor, where custom JavaScript commands, HTML templates and an application programming interface (API) are available. Therefore Gorilla.sc also can function as a learning platform where users progress on to programming - whilst providing an API that manages more complex issues (such as timing and data management) where a beginner might make errors. The code editor allows inclusion of any external libraries (e.g. animation: pixi.js, image processing: OpenCV.js, eyetracking: WebGazer.js) – so it is possible to include tasks built on the toolboxes discussed above, such as jsPsych. A full list of features is available here: `www.gorilla.sc/tools`, and a tutorial is included in the supplementary materials below.

Timing Control

A few techniques are utilised within Gorilla.sc to control timing. To minimise any potential delays due to network speed (mentioned above), several screens (trials) are loaded in advance of presentation - a process called caching. This means that fluctuations in connection speed will not lead to erroneous presentation times. The presentation of stimuli are achieved using the requestAnimationFrame() function, which allows the software to count frames and run code when the screen is about to be refreshed - ensuring screen-refreshing in the animation loop does not cause hugely inconsistent presentation. This method has previously been implemented to achieve accurate audio presentation (Reimers & Stewart, 2016) and accurate visual presentation (Yung, Cardoso-Leite, Dale, Bavelier, & Green, 2015).

Rather than assuming that each frame is going to be presented for 16ms, and presenting a stimulus for the nearest number of frames (something that commonly happens), Gorilla.sc times each frame's actual duration - using requestAnimationFrame(). The number of frames a stimulus is presented for can, therefore, be adjusted depending on the duration of each frame – so that most of the time a

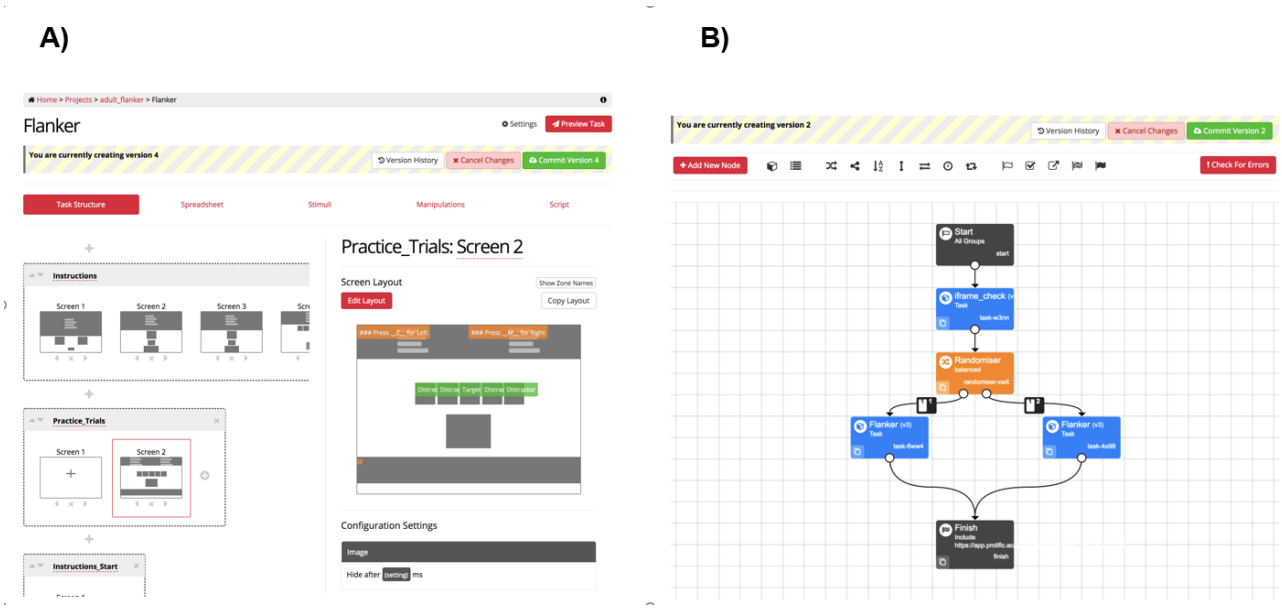


Figure 1: *An example of the two main GUI elements of Gorilla.sc. A) shows the task builder, with a screen selected, showing how a trial is laid out B) shows the experiment builder, there is a check for the participant, followed by a randomiser node which allocates them to one of two conditions, before sending them to a Finish node.*

longer frame refresh (due to lag) will not lead to a longer stimulus duration. This method was used in the (now defunct) QRTEngine (Barnhoorn et al., 2015), and to our knowledge is not used in other toolboxes (for a detailed discussion on this particular issue see this GitHub issue: `www.github.com/jspsych/jsPsych/issues/75` and this blog bost on the QRTEngine's website: `www.qrtengine.com/comparing-qrtengine-and-jspsych/`).

Reaction time (RT) is measured, and presentation time recorded using the performance.now() function, which is independent of the browser's system clock, and therefore not impacted by changes to this over time. This is the same method used by QRTEngine, validated using a photodiode (Barnhoorn et al., 2015).

Additionally, to maximise data quality, the user can restrict through the GUI which devices, browsers and connection speed they will allow the participant to have, and all this data is then recorded. This method allows the restriction of the participant environment, where only modern browser/device combinations are permitted - so the above techniques - and timing accuracy - are enforced. The user is able to make their own call in a trade-off between potential populations of participants, and restrictions on them to promote accurate timing, dependent on the particulars of the task or study.

Case Study

As a case study, an experiment was chosen to illustrate the platform's capability for accurate presentation and response timing. To demonstrate Gorilla.sc's ability to work within varied setups, different participant groups (primary school children and adults in both the UK and France), settings (without supervision, at home, and under supervision, in schools and in public engagement events), equipment (own computers, computer supplied by researcher), and connection types (personal internet connection, mobile phone 3G/4G) were selected.

We ran a simplified flanker task taken from the Attentional Networks Task (ANT) (Fan, McCandliss, Sommer, Raz, & Posner, 2002; Rueda et al., 2004). This task measures attentional skills, following the Attentional Network theory. In the original ANT papers, three attentional networks are characterised: alerting (a global increase in attention, delimited in time but not in space) orienting (the capacity to spatially shift attention to an external cue), and executive control (the resolution of conflicts between different stimuli). For the purpose of this paper and for the sake of simplicity we will focus on the "Executive control" component. This contrast was chosen as (MacLeod et al., 2010) found that it was highly powered and reliable relative to the other conditions in the ANT. Participants responded as quickly as possible to a central stimulus, one that is either pointing in the same direction as identical flanking stimuli, or in the opposite direction. Thus, there are both congruent (same direction) and incongruent (opposite direction) trials.

Research with this paradigm robustly shows that RTs to congruent trials are faster than those to incongruent trials – Rueda et al. (2004), term this

the 'conflict network'. This RT difference, while significant, is often less that 100ms, and thus very accurately timed visual presentation, and accurate recording of responses is necessary. Crump, McDonnell, and Gureckis (2013), successfully replicated the results of a similar Flanker task online, using Mechanical Turk, with letters as targets and flankers, so we know this can be an RT-sensitive task that works online. Crump et al. (2013) coded this task in JavaScript and HTML and managed the hosting and data-storage themselves; however, this current version was created and run entirely using Gorilla.sc's GUI. It is hypothesised that the previously recorded conflict RT difference will be replicated on this platform.

Experiment 1

Methods

Participants

Data was drawn from three independent groups. Group A was in Corsica, France, across 6 different primary classrooms. Group B was in three primary schools in London, UK. Group C was at a public engagement event carried out at a university in London.

In total, 270 elementary school children were recruited. Two participants were excluded for not performing above chance (< 60% accuracy) in the task. The final sample included 268 children (53.7% of females), between 4.38 and 12.14 years of age (M = 9.21; SD = 1.58). Details about the demographics for each study are provided in Table 1. Informed written parental consent was obtained for each participant, in accordance with the University's Ethics Committee.

Procedure

In all three groups, participants were tested in individual sessions, supervised by a trained experimenter. Although great care was taken to perform the task in a quiet place, noise from adjacent rooms sometimes occured in the school groups (A and B). To prevent children from getting distracted, they were provided with noise cancelling headphones (Noise Reduction Rating of 34dB; ANSI S3.19 and CE EN352-1 Approved).

The task was carried out using the web browser Safari, on a Mac OS X operating system. Because a stable Internet connection was often lacking in schools, in groups A and B, a mobile phone internet connection was used – this could vary from 3G to 4G.

Flanker Task

The Flanker task was adapted from Rueda et al. (2004). A horizontal row of five cartoon fish were presented in the centre of the screen (see Figure 2), and participants had to indicate the direction the middle fish was pointing (either to the left, or right), by pressing the "X" or "M" buttons on the keyboard. These buttons were selected so that children could put one hand on each response key. Buttons were covered by arrows stickers (left arrow for "X"; right arrow for "M") to avoid memory load. The task has two trial types: *congruent* and *incongruent*. In *congruent* trials, the middle fish was pointing in the same direction as the flanking fish. In the *incongruent* trials, the middle fish was pointing in the opposite direction. Participants were asked to answer as quickly and accurately as possible.

After the experimenter had introduced the task, there were 12 practice trials, with immediate on-screen feedback on the screen. That is to say, a red cross was displayed if children answered incorrectly, and a green tick was shown if they answered correctly. Instructions were clarified by the experimenter if necessary. After the practice trials, four blocks of 24 trials each were presented. Self-paced Breaks were provided between the blocks. For each participant, 50% of the trials were congruent, and the direction of the middle fish varied randomly between left and right. Four types of trials were therefore presented (see Figure 2): all the fish pointing to the right (25%), all the fish pointing to the left (25%), middle fish pointing to the right and flanking fish to the left (25%), middle fish pointing to the left and flanking fish to the right (25%).

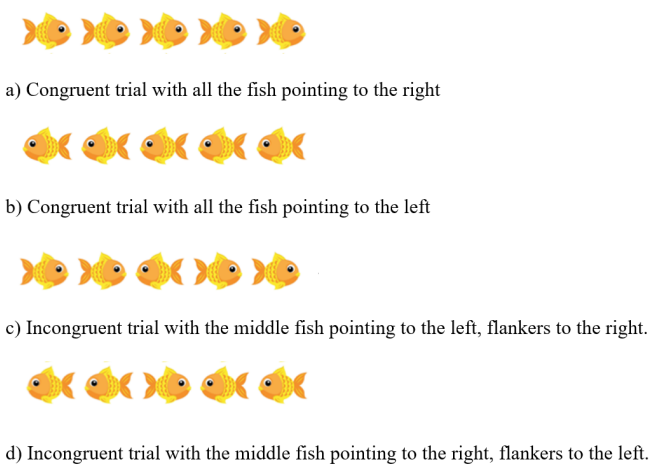


Figure 2: *Trial types for Experiment 1. The different conditions used in the Flanker Task*

As shown in Figure 3, for each trial, a fixation cross was displayed for 1700 ms. The cross was followed by the presentation of the fish stimuli, which

	Size					*Gender (% female)*
		Min	*Max*	*Mean*	*SD*	
Group A	116	7.98	11.38	9.95	.69	49.1
Group B	43	8.82	11.19	9.85	.55	60.5
Group C	109	4.38	12.14	8.18	1.93	56.0

Table 1: *Sample Size, Age and Gender of the participants per study. Age Range represented by Min and Max columns.*

stayed on screen until a valid response (either "X" or "M") was provided. A blank screen was then displayed before the next trial. The duration of the blank screen varied randomly between 400, 600, 800 and 1000ms. Overall, the task took no more than 10 minutes.

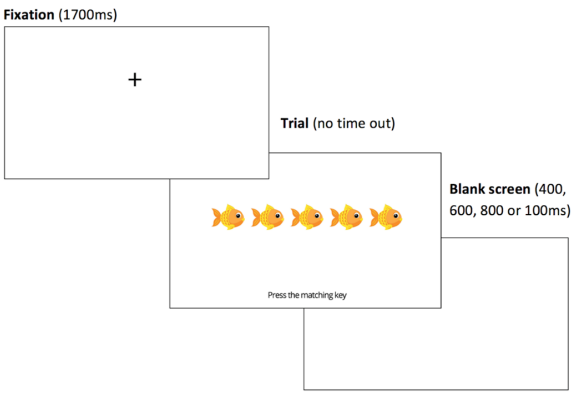


Figure 3: *The time-course of a typical trial in Experiment 1. These screens represent what the participant was seeing within their web-browser.*

Power Calculations

The main flanker effect reported in Rueda et al. (2004)'s ANT ANOVA results (Experiment 1) was F(2, 88) = 61.92; p = <0.001. They did not report the effect size, so this permits us only to estimate the effect size using partial eta squared. This was calculated using the calculator provided by (Lakens, 2013) , as η_p^2 = .58 (95% CI= .44 - .67). Using G*power (Faul, Erdfelder, Buchner, & Lang, 2009), an a priori power calculation was computed for a MANOVA with 3 groups, and a measurement correlation (congruent*incongruent) of .81 (taken from internal correlation of this measure reported in MacLeod et al. (2010)). In order to reach a power of above .95 a sample of 15 would be needed for each of our groups - we included in excess of this to increase sensitivity and provide power of <.99.

Results

	Accuracy (%)	**RT(ms)**
Congruent	97.79 (.029)	887.79 (17.10)
Incongruent	96.88 (.050)	950.12 (23.71)

Table 2: *Accuracy and Reaction Time of participants, averaged (mean) over all groups, split by Congruency. Standard Error of the means are shown in brackets*

Accuracy

A total accuracy score was computed as the proportion of correct trials throughout the task (number of correct answers / total number of trials).

A MANOVA with Congruency as a within-subject factor, Study as a between-subject factor, and Age as a covariate, revealed a significant main effect of Congruency on participants' accuracy (F (1, 264) = 9.02, p = .003, η_p^2 = .033). Although performance was at ceiling for both types of trials, participants were more accurate for congruent trials, compared to incongruent trials (see Table 2). This effect significantly interacted with participants' age, (F (1, 264) = 6.80, p = .010, η_p^2 = .025), but not with the Study they participated in (F (1, 264) = .501, p = .607, η_p^2 = .004). In order to shed light on this interaction effect, the difference in accuracy scores between congruent trials and incongruent trials was computed for each subject. This difference diminished with age (r = -.22; p < .001).

Results from the MANOVA should however be interpreted with caution, since two assumptions were violated with the present data. First, the distribution of accuracy scores in each of the three groups were skewed and did not follow a normal distribution (for Study 1: Shapiro-Wilk W = .896, p < .001; for Study 2, W = .943, p = .034; for Study 3: W = .694, p < .001). Secondly, the Levene's test for equality of variance between groups was significant (for Congruent trials: F(2, 265) = 5.75, p = .004; for Incongruent trials: F(2, 265) = 13.904, p < .001). The distribution of data is represented in Figure 4.

The non-parametric Friedman Test, however, also reveals a significant effect of Congruency on accuracy scores ($\chi_2(1)$ = 5.17, p < .023).

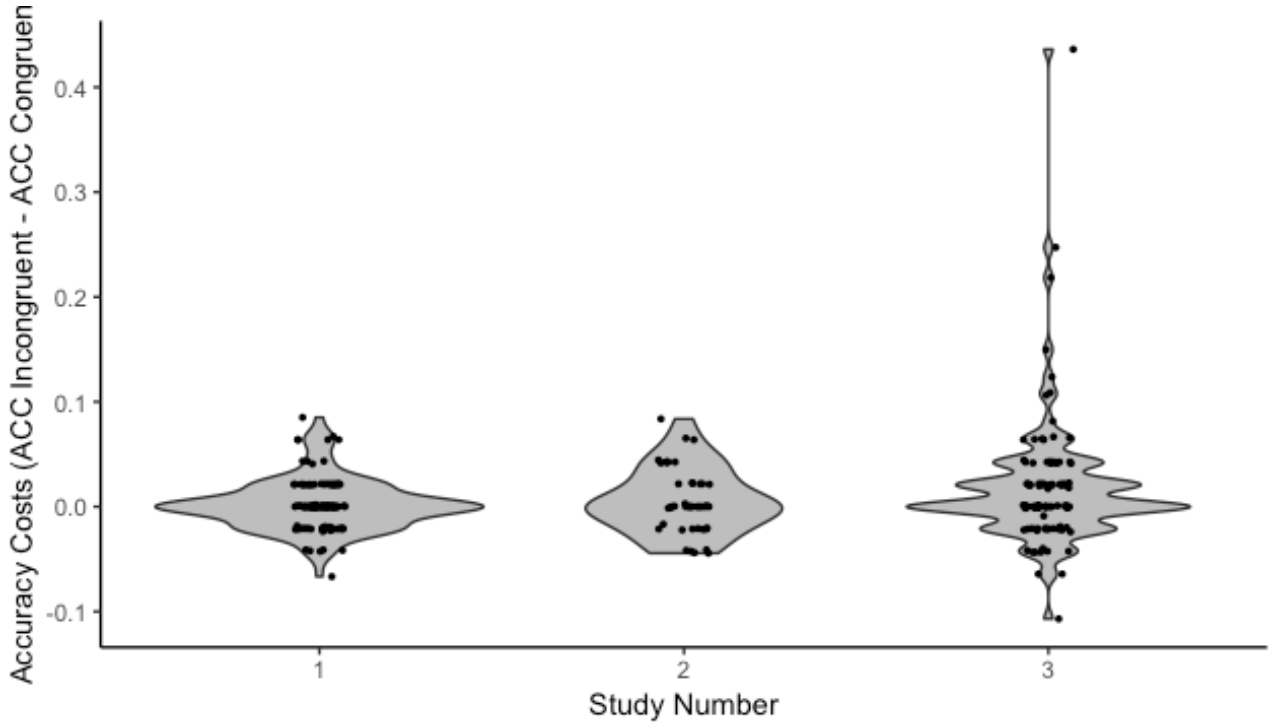


Figure 4: *Distribution of Accuracy difference between congruent and incongruent trials, for each group in Experiment 1*

Reaction Time

Reaction time scores (RTs) correspond to the mean reaction time for correct answers. Reaction times (RTs) above 3 standard deviations from the mean of each subject were excluded in order to prevent extreme values from influencing the results (in some instances, children were asking a question in the middle of the trial). RTs under 200ms were also excluded, being too short to follow the perception of the stimulus.

A MANOVA with Congruency as a within-subject factor, Study as a between-subject factor, and Age as a covariate revealed a main effect of Congruency on participants' reaction times (F (1, 264) = 18.92, p < .001, n^2p = .067). Participants took longer to provide the correct answer for incongruent trials, compared to congruent trials (see Table 2). This effect significantly interacted with Age, (F (1, 264) = 11.36, p = .001, n^2p = .041), but not with Study type (F (1, 264) = .594, p = .553, n^2p = .004). In order to better understand this interaction effect, RTs costs were calculated by subtracting the mean reaction time to the congruent trials, to the mean reaction time to incongruent trials. Higher values indicate poorer inhibitory control, in that it takes longer to give the correct answer for incongruent trials. RTs costs decreased with Age, indicating an improvement in inhibitory control over development (r = -.20; p = .001).

Similarly to the analyses for accuracy scores, RTs in each of the three groups were skewed and do not follow a normal distribution (for Study 1: Shapiro-Wilk W = .476, p < .001; for Study 2, W = .888, p = .034; for Study 3: W = .649, p < .001). Secondly, the Levene's test for equality of variance between groups was significant (for Congruent trials: F(2, 265) = 9.36, p < .001; for Incongruent trials: F(2, 265) = 7.276, p < .001). The distribution of data is represented in Figure 5.

The non-parametric Friedman Test, however, also reveals a significant effect of Congruency on reaction times for correct answers (X^2 (1) = 55.37, p < .001).

discussion

The Flanker effect was successfully replicated on a sample of 268 children tested using Gorilla.sc. This characterised the 'conflict attentional network', children taking longer to provide correct answers to incongruent trials, compared to congruent trials. This effect was lower than 100ms (being of 62.33ms on average). Crucially, there was no interaction between the Flanker effect and the specific study where the data came from, despite the fact that their set up differed greatly: two groups were taken from schools, over a mobile phone internet connection, and the third group was taken from a University setting, over a communal internet connection.

In each study, however, pupils were supervised by a trained experimenter who guided them through the task, and who checked the quality of the internet

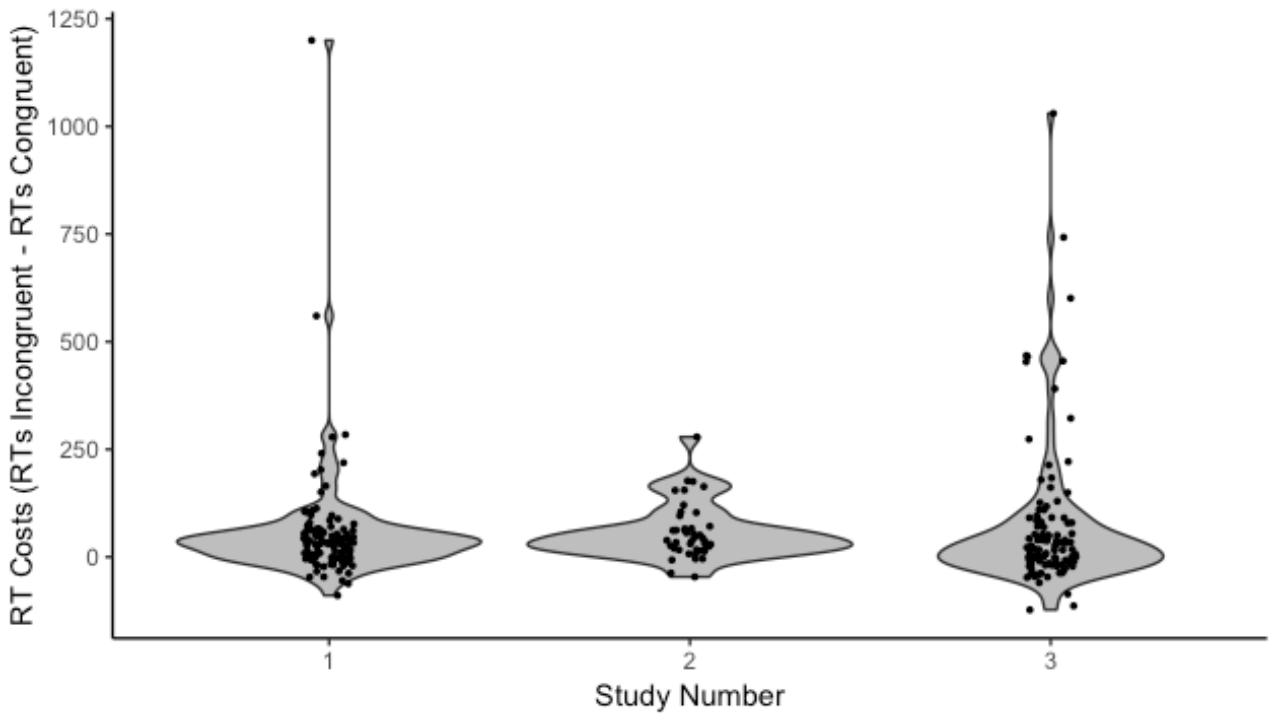


Figure 5: *Distributions of RT difference between congruent and incongruent trials, for each group in Experiment 1*

connection. One of the potential benefits of web-based research is in reaching participants in various places (e.g., their own house), allowing for broad and unsupervised testing. Experiment 2 therefore, tested whether the Flanker effect would hold under such conditions, recruiting adult participants over Prolific and without supervision.

Experiment 2

Methods

Participants

104 adults were recruited, five participants were excluded for not performing above chance (<60% accuracy) in the task (these individuals also had an accuracy in excess of 3 Standard Deviations from the mean). This left a sample of 99 of adults (57.57% female), with a mean age of 30.32 (SD = 6.64), ranging from 19 to 40 years old.

All participants were recruited online, through the Prolific.ac website, which allows recruitment and administration of online tasks and questionnaires (Palan & Schitter, 2018). All participants were based in the United Kingdom and indicated corrected to normal vision, English as a first language, and no history of mental illness or cognitive impairment. This experiment was conducted in line with Cauldron Sciences Ethics code – with complies with the Declaration of Helsinki (World Medical Association, 2013). Informed consent was obtained through an online form, participants were informed they could opt out during the experiment without loss of payment.

Compensation for the task was £0.60 GBP, which on average translated to a rate of £8.70 per hour, as participants took an average of 4 minutes and 8.36 seconds to complete the task.

In addition, the software recorded the operating system, web browser and browser viewpoint size (the number of pixels that were displayed in the browser) of the users. The breakdown is shown in Table 3 and Table 4.

	Count (Percentage)
Browser	
Chrome	75 (75.76%)
Safari	9 (9.09%)
Firefox	9 (9.09%)
Edge	3 (3.03%)
Other	3 (3.03%)
Operating System	
Windows 10	57 (57.58%)
Windows 8	4 (4.04%)
Windows 7	17 (17.17%)
Mac OS 10	16 (16.16%)
Chromium	5 (5.05%)

Table 3: *Breakdown of Operating System and Browser versions within the sample. Total percentages of the sample are included in brackets.*

	Mean (Pixels)	Std. Deviation (Pixels)	Range (Pixels)
Horizontal	1496.13	218.85	1051 - 1920
Vertical	759.40	141.66	582 - 1266

Table 4: *Viewport characteristics of the adult sample's web browsers. The viewport is the area of a browser containing the information from a site.*

Procedure

Participants completed the task on their own computers at home, and were not permitted to access the task on a tablet or smartphone. Before starting the task, participants read a description and instructions for taking part in the study, which asked them to open the experiment in a new window and note that the task would take around 5 minutes to complete (with an upper limit of 10 minutes). When the participants had consented to take part in the study on Prolific.ac, they were given a personalised link to the Gorilla.sc website, where the experimental task was presented. First, a check was loaded to ensure they had not opened the task embedded in the Prolific website (an option that was available at the time of writing), which would minimise distraction. Then the main section was administered in the browser; on completion of this they returned to Prolific.ac with a link including a verification code to receive payment.

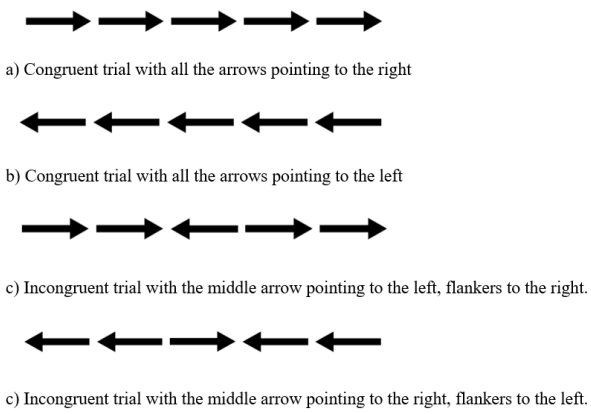


Figure 6: *Trial types for Experiment 2. The different conditions used in the Flanker Task*

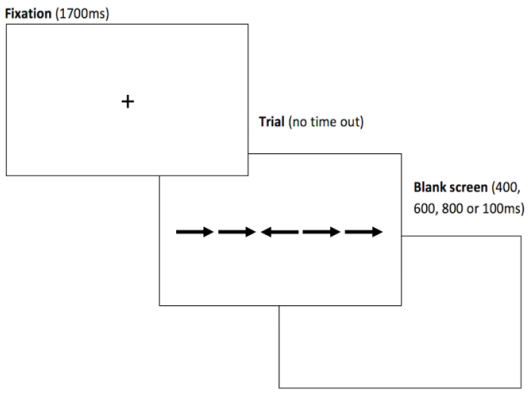


Figure 7: *The time-course of a typical trial in Experiment 2. These screens represent what the participant was seeing within their web-browser.*

Flanker Task

An adult version of the *conflict network* flanker task, taken from the ANT was used Rueda et al. (2004).The mechanics, trial numbers and conditions of this task were identical to those used in Experiment 1; however, the stimuli were altered. The fish were replaced with arrows, as is typically done in adult studies (Fan et al. (2002); see Rueda et al. (2004) for a comparison of the child and adult versions). This is illustrated in Figure 6 and the time-course illustrated in Figure 7.

Similarly, the adults were given written instructions, and then completed 12 practice trials with immediate feedback. Then moved on to complete 4 blocks of 24 trials (25% congruent-left, 25% congruent-right, 25% incongruent-left, 25% incongruent-right).

Power Calculations

The main effect of flanker reported in Rueda et al. (2004)'s adult arrow ANOVA results (Experiment 3) was $F(2, 44) = 142.82$; $p = 0.0019$. They did not report the effect size, so this permits us only to estimate the effect size using partial eta squared. This was calculated using the calculator provided by Lakens (2013), as $\eta_p^2 = .87$ (95% CI: .78 - .90).

However, as our planned comparisons for this group are simple (a t-test for mean RT and accuracy for incongruent versus congruent trials), we calculated power using the reported mean and standard deviation values from Fan et al. (2002) – Rueda et al. (2004) did not report the standard deviation, so

this was not possible. The mean RT for congruent trials was 530 ms (SD = 49), and 605 ms (SD = 59) for incongruent trials. Using an a priori calculation from the G*Power software, this gave us a calculated effect size of d=1.38 and a sample size of 26 to reach a power of .96. However, this assumes that we are working in a comparable environment, which is not the case due to the increased potential noise - our sample size is therefore much larger than the original paper to account for increased noise, giving us a calculated power of <.99.

Results

Accuracy

Accuracy scores were computed over the total number of trials for each condition (congruent and incongruent). These means are shown in Table 5. As mentioned above, 5 participants were excluded for not being above chance based on these accuracy scores. Accuracy was distributed non-normally (Shapiro-Wilk W = 0.819 p <.001), so a Wilcoxon signed-rank test was used to compare the mean accuracy across the two types of trials. This provided evidence for a significant difference between the two means (1.72% difference, W = 1242, p < .001) with a Rank-Biserial Correlation of r_{rb}=.49 (an estimation of effect size for non-parametric data, Hentschke and Stüttgen (2011)).

	Accuracy (%)	RT(ms)
Congruent	99.28 (.011)	498.72 (9.38))
Incongruent	97.56 (.033)	527.81 (10.80)

Table 5: *Average accuracy and correct trials reaction time for congruent and incongruent trials. Standard Error are in brackets.*

Reaction Time

Average reaction time was calculated for the two trial types - congruent and incongruent. Means and standard errors are reported in Table 5. Reaction Time was only calculated for correct trials, as the accuracy rates were at ceiling. As above, a Shapiro-Wilk suggested the data was distributed non-normally (W = 0.748 p <.001), so a Wilcoxon signed-rank test was used to compare the differences in mean reaction time. This test suggested a significant difference between the two means (29.1 ms difference, W = 414, p < .001) with a Rank-Biserial Correlation of r_{rb}=.83.

Discussion

The 'conflict' attentional network effect was observed and replicated. This was encouraging given the decrease in signal to noise that variance in operating system, web browser, and screen size (shown above) would contribute towards this type of task. However, the effect of 29.1ms was smaller than that observed in the original lab-based study (120ms), and still smaller than the average effect of 109 ms reported in a meta-analysis of lab studies by MacLeod et al. (2010); this is likely due to variance in a remote environment. This may not be that surprising, as MacLeod et al. (2010) found that there was large variance in the RT differences between and within participants over multiple studies – 1655 ms and 305 ms respectively. This smaller observed difference is also potentially driven by reduced RT variance. The average error in Experiment 1 was 20 ms, whereas it was around 10 ms in Experiment 2 – possibly leading to the lower than expected difference in RT. We are unable to compare our variance with the original paper's child ANT results, as standard error or deviations were not reported. As a nearest online comparison, Crump et al. (2013)'s letter flanker's difference between congruent and incongruent trials was 70 ms, which is closer to our observed difference, suggesting that online studies tend to find a smaller RT difference, however the stimuli and task structure differ significantly between our implementation and Crump et al. (2013)'s.

One potential explanation for the faster reaction times, and decreased variance in the Prolific sample we tested could be their unique setting – the framing and task goals of these participants are different to typical volunteers. Research investigating users on the Mechanical Turk platform found that they were more attentive than panel participants (Hauser & Schwarz, 2016), suggesting internet populations are measureably different in their responses. Increased attentiveness could potentially lead to less within-subject variance – this may be an avenue of research for a future study.

General Discussion

Gorrilla.sc is an Experiment Builder: a platform for the creation and administration of online behavioural experiments. It goes beyond an API, toolbox or JavaScript engine, and provides a full interface for task design and administration of experiments. It manages presentation time and response recording for the user, building on previous advances in browser-based research software without the requirement for programming or browser technology understanding. Utilising these tools, measurement of the 'conflict' attention network was successfully replicated online. The replication persisted across several different groups, children in primary schools in two countries, children at a public engagement event, and

adults taking part on their own machines at home. This demonstrates tasks built using this platform can be used in a wide range of situations – which have the potential to introduce unwanted variance in timing through software, hardware and internet connection speed – and still be robust enough to detect reaction time differences, even in a task containing a relatively low number of trials (<100 trials).

Results such as these provide evidence that could enable more researchers to undertake behavioural research on the web, whilst also offering the maintained back-end which can be kept up-to-date with changes in user's browsers - that otherwise would require a much higher level of technical involvement.

Building on these advantages, Gorilla is currently being used to teach research methods to undergraduate students in London at University College London and Birkbeck, University of London. In comparison with other software requiring specific programming skills, the teaching teams noted a lower need to provide technical assistance to students, allowing them to better focus on the research design per se.

Limitations

Whilst technical involvement is lowered, there are still some limitations with presenting a task in the browser that the user should be aware of. These are mainly limited to timing issues, which Gorilla.sc minimises but does not eliminate – there will always be an error rate, even though it is decreasing. The specific reasons for this error, and how it may be quantified or overcome in the future, are discussed below.

As with any software running on a user's device, Gorilla.sc's response time is limited by the sampling/polling rate of input devices – a keyboard for example. Unfortunately, short of installing intrusive software on the user's device, the web browser has no mechanism for directly accessing polling rate - or controlling for polling rate. Often this sits at around 125 Hz, so this can be used to inform conclusions based on Reaction Time data gathered online. Future developments may at some point allow programs running in the browser to access hardware information and adjust for this - however, this will only be important for research which aims to model individual trials on an accuracy of less than 8ms. Alternatively, developments in recruitment platforms (such as Prolific and Mechanical Turk) may enable screening of participant's hardware, allowing researchers to specify participants with high refresh monitors and high polling-rate input devices (most likely to be video-gamers).

One unique problem in remote testing is the potential processing load any given participant may have running on their computer may vary dramatically. High processing loads will impact the consistency of stimulus presentation and the recording of responses. Fortunately, the platform records the actual time each frame is presented for, against the desired time – so the impact on timing can be recorded and monitored. A potential future tool would be a processing load check - this could either work by performing computations in the browser and timing them as a proxy for load - or if browsers adopt methods already available in Node.js (an off-browser JavaScript runtime engine) for profiling CPU performance.

The use of modern browser features, such as requestAnimationFrame(), gives the best possible timing fidelity in the browser environment, and also allows for inconsistencies in frame refresh rate to be measured and accounted for. Online research will always be limited by the hardware that participants have, and because most displays have a refresh rate of 60Hz, stimulus presentation times are limited to multiples of 16ms. It is therefore advisable for users on any online platform to restrict presentation times to multiples of 16.67ms. This is spoken about in Gorilla.sc's documentation, however, a future feature may be to include a warning to users when they try and enter non-multiples of the standard frame rate.

Future Features

There are potential improvements to the platform that would make it a more powerful tool for researchers. These fall into two camps: tools for widening the range of experiments you can run, and tools for improving the quality of data you can collect.

In the authors experience, tools for researchers to run online visual perception, attention and cognition research are limited. This is perhaps a product of reluctance to use online methods, due to concerns regarding timing – which we hope to have moved towards addressing. In order to provide a greater range of tools a JavaScript-based Gabor patch generator is under development, which can be viewed using this link: `www.bit.ly/GorillaGabor`. This first asks participants to calibrate their presentation size to a credit card, and measure the distance to the screen - calculating visual degrees per pixel - and then allows presentation of a Gabor patch with size, frequency, window size in degrees. Experimenters can also set animations which change the phase and angle of these patches over time. These animations are fast

(40Hz) as the patch and window are pre-generated and manipulated to produce the animation, rather than a frame-by-frame new patch generation.

Another tool that widens online research capabilities is remote, webcam-based eye tracking. An implementation of the WebGazer.js library (Papoutsaki et al., 2016) for eye-tracking is also being integrated into the platform. This permits rough eye-tracking, and head position tracking, using the user's webcam. Recent research has provided evidence that this can be used for behavioural research, with reasonable accuracy - about 18% of screen-size (Semmelmann & Weigelt, 2018). This will also include a calibration tool, which can be run as frequently as needed, which allows the quantification of eye-tracking accuracy, and offers the ability to end the experiment if the webcam cannot be calibrated to the desired level. A prototype demo of the calibration is available here: `www.bit.ly/EyeDemo`. Additionally, WebGazer.js allows the experimenter to track the presence and changes in distance, of a user's face. This can help with data quality, as you can assess when a user is looking at the screen, and prompt them to remain attentive to the task. The impact of this type of monitoring may be particularly interesting to investigate in a task such as the one presented in this paper – perhaps participants would show an increased flanker effect if they were more attentive in the task.

Conclusion

We described Gorilla.sc as a tool which lowers the access barriers to running online experiments – e.g. understanding web development languages, servers, programming APIs – significantly, by managing all levels of implementation for the user and keeping up to date with changes in the browser ecosystem. We presented a case study, to demonstrate Gorilla.sc's capacity to be robust to environmental variance (from software, hardware and setting) during a timing task. An RT sensitive Flanker effect - Rueda et al. (2004)'s conflict network is replicated in several populations and situations. There remain some constraints in running studies online - there may be future ways of tackling some of these constraints (i.e. specialist hardware). Future improvements to the platform include: a Gabor generator, webcam eye-tracking, and movement monitoring.

Notes

Acknowledgements: We would like to thank Edwin Dalmaijer, Nick Hodges, Marie Payne, Daniel C. Richardson, Hannah Spence, and, Will Webster for their feedback, and discussion of this paper.

Declaration of Interests: Experiment 1 was hosted by Cauldron Science. Experiment 2 was hosted, and participant compensation was paid for by Cauldron Science, the creators of Gorilla.sc. AA and AF are employed by Cauldron Science and JE is the Chief Executive Officer.

References

Adjerid, I., & Kelley, K. (2018). Big data in psychology: A framework for research advancement. *American Psychologist*, No Pagination Specified–No Pagination Specified. doi: 10.1037/amp0000190

Barnhoorn, J. S., Haasnoot, E., Bocanegra, B. R., & Steenbergen, H. v. (2015, December). QRTEngine: An easy solution for running online reaction time experiments using Qualtrics. *Behavior Research Methods*, *47*(4), 918–929. Retrieved 2018-07-09, from `https://link.springer.com/article/10.3758/s13428-014-0530-7` doi: 10.3758/s13428-014-0530-7

Casler, K., Bickel, L., & Hackett, E. (2013, November). Separate but equal? A comparison of participants and data gathered via Amazon's MTurk, social media, and face-to-face behavioral testing. *Computers in Human Behavior*, *29*(6), 2156–2160. Retrieved 2018-07-06, from `http://www.sciencedirect.com/science/article/pii/S074756321300160X` doi: 10.1016/j.chb.2013.05.009

Chen, S.-C., de Koning, B., & Zwaan, R. A. (2018, Mar). *Does object size matter with regard to the mental simulation of object orientation?* OSF. Retrieved from `osf.io/26jgu`

Crump, M. J. C., McDonnell, J. V., & Gureckis, T. M. (2013, March). Evaluating Amazon's Mechanical Turk as a Tool for Experimental Behavioral Research. *PLOS ONE*, *8*(3), e57410. Retrieved 2018-07-23, from `http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0057410` doi: 10.1371/journal.pone.0057410

de Leeuw, J. R. (2015, March). jsPsych: A JavaScript library for creating behavioral experiments in a Web browser. *Behavior Research Methods*, *47*(1), 1–12. Retrieved 2018-07-19, from `https://link.springer.com/article/10.3758/s13428-014-0458-y` doi: 10.3758/s13428-014-0458-y

de Leeuw, J. R., & Motz, B. A. (2016, March). Psychophysics in a Web browser? Comparing response times collected with

JavaScript and Psychophysics Toolbox in a visual search task. *Behavior Research Methods*, *48*(1), 1–12. Retrieved 2018-07-09, from https://link.springer.com/article/10.3758/s13428-015-0567-2 doi: 10.3758/s13428-015-0567-2

Fan, J., McCandliss, B. D., Sommer, T., Raz, A., & Posner, M. I. (2002, April). Testing the Efficiency and Independence of Attentional Networks. *Journal of Cognitive Neuroscience*, *14*(3), 340–347. Retrieved 2018-07-23, from https://doi.org/10.1162/089892902317361886 doi: 10.1162/089892902317361886

Faul, F., Erdfelder, E., Buchner, A., & Lang, A.-G. (2009, November). Statistical power analyses using G*Power 3.1: Tests for correlation and regression analyses. *Behavior Research Methods*, *41*(4), 1149–1160. Retrieved 2018-08-06, from https://link.springer.com/article/10.3758/BRM.41.4.1149 doi: 10.3758/BRM.41.4.1149

Ferdman, S., Minkov, E., Bekkerman, R., & Gefen, D. (2017, June). Quantifying the web browser ecosystem. *PLOS ONE*, *12*(6), e0179281. Retrieved 2018-07-19, from http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0179281 doi: 10.1371/journal.pone.0179281

Garaizar, P., Vadillo, M. A., & López-de Ipiña, D. (2012, July). Benefits and pitfalls of using HTML5 APIs for online experiments and simulations. In *2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)* (pp. 1–7). doi: 10.1109/REV.2012.6293120

Garaizar, P., Vadillo, M. A., & López-de Ipiña, D. (2014, October). Presentation Accuracy of the Web Revisited: Animation Methods in the HTML5 Era. *PLOS ONE*, *9*(10), e109812. Retrieved 2018-07-09, from http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0109812 doi: 10.1371/journal.pone.0109812

Hauser, D. J., & Schwarz, N. (2016, March). Attentive Turkers: MTurk participants perform better on online attention checks than do subject pool participants. *Behavior Research Methods*, *48*(1), 400–407. Retrieved 2018-08-10, from https://link.springer.com/article/10.3758/s13428-015-0578-z doi: 10.3758/s13428-015-0578-z

Hentschke, H., & Stüttgen, M. C. (2011, December). Computation of measures of effect size for neuroscience data sets. *European Journal of Neuroscience*, *34*(12), 1887–1894. Retrieved 2018-08-03, from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1460-9568.2011.07902.x doi: 10.1111/j.1460-9568.2011.07902.x

Jasmin, K., Dick, F., Holt, L., & Tierney, A. T. (2018). Degeneracy makes music and speech robust to individual differences in perception. *bioRxiv*. Retrieved from https://www.biorxiv.org/content/early/2018/07/05/263079 doi: 10.1101/263079

Jia, R., Guo, H., Wang, Y., & Zhang, J. (2018, May). Analysis and test of sound delay on Web Audio under different situations. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)* (pp. 1515–1519). doi: 10.1109/ICIEA.2018.8397949

Jones, A. L. (2018, Aug). Beyond average: Using face regression to study social perception. *OSF*. Retrieved from osf.io/q5wvn

Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and ANOVAs. *Frontiers in Psychology*, *4*. Retrieved 2018-08-06, from https://www.frontiersin.org/articles/10.3389/fpsyg.2013.00863/full doi: 10.3389/fpsyg.2013.00863

Lange, K., Kühn, S., & Filevich, E. (2015, June). "Just Another Tool for Online Studies" (JATOS): An Easy Solution for Setup and Management of Web Servers Supporting Online Studies. *PLOS ONE*, *10*(6), e0130834. Retrieved 2018-07-19, from http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130834 doi: 10.1371/journal.pone.0130834

Lavan, N., Knight, S., & McGettigan, C. (2018, September). Listeners form average-based representations of individual voice identities - even when they have never heard the average. *PsyArXiv*. Retrieved 2018-10-05, from https://psyarxiv.com/bqswf/ doi: 10.31234/osf.io/bqswf

Lumsden, J., Skinner, A., Coyle, D., Lawrence, N., & Munafo, M. (2017). Attrition from Web-Based Cognitive Testing: A Repeated Measures Comparison of Gamification Techniques. *Journal of Medical Internet Research*, *19*(11), e395. Retrieved 2018-10-05, from https://www.jmir.org/2017/11/e395/ doi: 10.2196/jmir.8473

MacLeod, J. W., Lawrence, M. A., McConnell, M. M., Eskes, G. A., Klein, R. M., & Shore, D. I. (2010). Appraising the ANT: Psychometric and theoretical considerations of the Attention Network Test. *Neuropsychology*, *24*(5), 637–651. doi: 10.1037/a0019803

Miller, R., Schmidt, K., Kirschbaum, C., & Enge, S. (2018, March). Comparability, stability, and reliability of internet-based mental chronometry in domestic and laboratory settings. *Behavior Research Methods*, 1–14. Retrieved 2018-07-06, from https://link.springer.com/article/10.3758/s13428-018-1036-5 doi: 10.3758/

s13428-018-1036-5

Palan, S., & Schitter, C. (2018, March). Prolific.ac—A subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, *17*, 22–27. Retrieved 2018-07-30, from http://www.sciencedirect.com/science/article/pii/S2214635017300989 doi: 10.1016/j.jbef.2017.12.004

Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., & Hays, J. (2016, January). WebGazer: Scalable Webcam Eye Tracking Using User Interactions. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence - IJCAI 2016*. Retrieved 2018-08-06, from https://par.nsf.gov/biblio/10024076-webgazer-scalable-webcam-eye-tracking-using-user-interactions

Pollock, L. (2018, June). Statistical and methodological problems with concreteness and other semantic variables: A list memory experiment case study. *Behavior Research Methods*, *50*(3), 1198–1216. Retrieved 2018-10-05, from https://doi.org/10.3758/s13428-017-0938-y doi: 10.3758/s13428-017-0938-y

Poort, E. D., & Rodd, J. M. (2017, May). Studies of cross-lingual long-term priming. *PsyArXiv*. Retrieved from psyarxiv.com/ert8k doi: 10.31234/osf.io/ert8k

Reimers, S., & Stewart, N. (2015, June). Presentation and response timing accuracy in Adobe Flash and HTML5/JavaScript Web experiments. *Behavior Research Methods*, *47*(2), 309–327. Retrieved 2018-07-09, from https://link.springer.com/article/10.3758/s13428-014-0471-1 doi: 10.3758/s13428-014-0471-1

Reimers, S., & Stewart, N. (2016, September). Auditory presentation and synchronization in Adobe Flash and HTML5/JavaScript Web experiments. *Behavior Research Methods*, *48*(3), 897–908. Retrieved 2018-07-23, from https://link.springer.com/article/10.3758/s13428-016-0758-5 doi: 10.3758/s13428-016-0758-5

Richards, G., Lebresne, S., Burg, B., & Vitek, J. (2010). An Analysis of the Dynamic Behavior of JavaScript Programs. In *Proceedings of the 31st ACM SIGPLAN Conference on Programming Language Design and Implementation* (pp. 1–12). New York, NY, USA: ACM. Retrieved 2018-08-06, from http://doi.acm.org/10.1145/1806596.1806598 doi: 10.1145/1806596.1806598

Richardson, D. C., Griffin, N. K., Zaki, L., Stephenson, A., Yan, J., Curry, T., … Devlin, J. T. (2018, September). Measuring narrative engagement: The heart tells the story. *bioRxiv*, 351148. Retrieved 2018-10-05, from https://www.biorxiv.org/content/early/2018/09/13/351148 doi: 10.1101/351148

Rueda, M. R., Posner, M. I., & Rothbart, M. K. (2004). Attentional control and self-regulation. In *Handbook of self-regulation: Research, theory, and applications* (pp. 283–300). New York, NY, US: Guilford Press.

Schmidt, W. C. (2001, May). Presentation accuracy of Web animation methods. *Behavior Research Methods, Instruments, & Computers*, *33*(2), 187–200. Retrieved 2018-07-09, from https://link.springer.com/article/10.3758/BF03195365 doi: 10.3758/BF03195365

Semmelmann, K., & Weigelt, S. (2018, April). Online webcam-based eye tracking in cognitive science: A first look. *Behavior Research Methods*, *50*(2), 451–465. Retrieved 2018-08-06, from https://link.springer.com/article/10.3758/s13428-017-0913-7 doi: 10.3758/s13428-017-0913-7

Severance, C. (2012, February). JavaScript: Designing a Language in 10 Days. *Computer*, *45*(2), 7–8. doi: 10.1109/MC.2012.57

Stoet, G. (2017, January). PsyToolkit: A Novel Web-Based Method for Running Online Questionnaires and Reaction-Time Experiments. *Teaching of Psychology*, *44*(1), 24–31. Retrieved 2018-07-23, from https://doi.org/10.1177/0098628316677643 doi: 10.1177/0098628316677643

Turner, M., Budgen, D., & Brereton, P. (2003, October). Turning software into a service. *Computer*, *36*(10), 38–44. doi: 10.1109/MC.2003.1236470

Usher-Smith, J. A., Masson, G., Mills, K., Sharp, S. J., Sutton, S., Klein, W. M. P., & Griffin, S. J. (2018, June). A randomised controlled trial of the effect of providing online risk information and lifestyle advice for the most common preventable cancers: study protocol. *BMC Public Health*, *18*(1), 796. Retrieved 2018-10-05, from https://doi.org/10.1186/s12889-018-5712-2 doi: 10.1186/s12889-018-5712-2

Woods, A. T., Velasco, C., Levitan, C. A., Wan, X., & Spence, C. (2015, July). Conducting perception research over the internet: a tutorial review. *PeerJ*, *3*, e1058. Retrieved 2018-07-23, from https://peerj.com/articles/1058 doi: 10.7717/peerj.1058

World Medical Association. (2013, November). World Medical Association Declaration of Helsinki: ethical principles for medical research involving human subjects. *JAMA*, *310*(20), 2191–2194. doi: 10.1001/jama.2013.281053

Yung, A., Cardoso-Leite, P., Dale, G., Bavelier, D., & Green, C. S. (2015, February). Methods to Test Visual Attention Online. *Journal of Visu-*

alized Experiments : JoVE(96). Retrieved 2018-07-23, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4354665/ doi: 10.3791/52470

Zloteanu, M., Harvey, N., Tuckett, D., & Livan, G. (2018, March). *Digital Identity: The Effect of Trust and Reputation Information on User Judgement in the Sharing Economy* (SSRN Scholarly Paper No. ID 3136514). Rochester, NY: Social Science Research Network. Retrieved 2018-10-05, from https://papers.ssrn.com/abstract=3136514

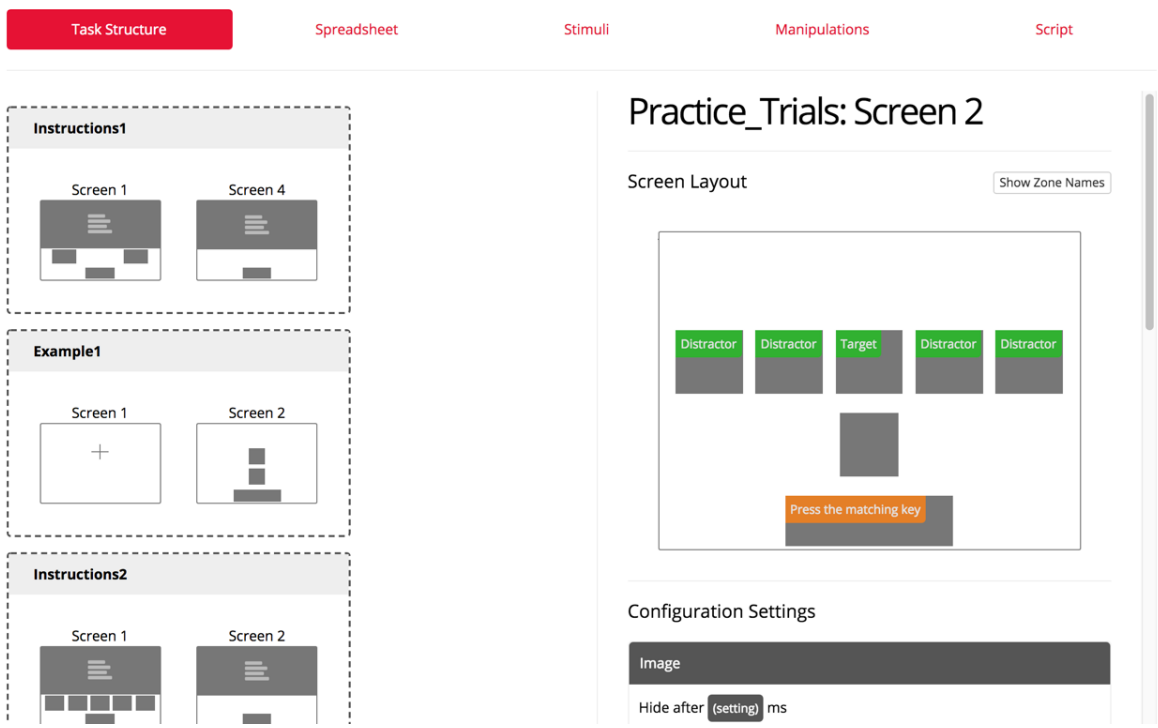
Supplementary Material: A tutorial on constructing the Flanker task

The Flanker task was built in two steps. First, we created the Task. Secondly, we incorporated this Task in an Experiment, which allowed us to recruit participants.

Creating the task

The Flanker task is a fairly simple task to program on Gorilla. Out of the five tabs of the Task Builder (Task Structure, Spreadsheet, Stimuli, Manipulations, Script), we only need to use to first three ones.

The "Task Structure" (highlighted in red below) tab allows us to define the different sections within the task, and the different screens within each section.



On the left, we can see the different sections. There is a first set of instructions ("Instructions 1"), followed by a very simple demonstration of the stimuli ("Example 1"). The instructions are fully developed in "Instructions 2" and followed by some practice trials ("Practice_Trials").

On the right, we can see more specifically how the screen is designed for the practice trials. Five images zones have been defined. Their name is written in green: the central fish is the "Target", the flanker fish the "Distractors". The rectangular zone with an orange label indicates the response modality. Participants have to press a button saying whether the central fish is pointing to the right ("m") or to the left ("z"). They will see the instructions that are written in orange on the screen: "Press the matching key". The squared zone in between the green and orange zones is used to provide feedback. We will not develop it here since it is only used during practice, and not during the main task.

Crucially, the Screen set up only has to be defined once. The images that will populate it from trial to trial are specified in the Spreadsheet.

Task Structure | **Spreadsheet** | Stimuli | Manipulations | Script

Download Spreadsheet | Upload Spreadsheet | Show spreadsheet: Simple List | Options | + Add Spreadsheet

Row	randomise_blocks	randomise_trials	display	Answer	Target	Distractor	Fixation	Type
1			Instructions1					
2		1	Example1	Left	Fish_L.png			
3		1	Example1	Right	Fish_R.png			
4			Instructions2					
5		2	Practice_Trials	Left	Fish_L.png	Fish_L.png	400	Congruent
6		2	Practice_Trials	Right	Fish_R.png	Fish_R.png	1000	Congruent
7		2	Practice_Trials	Left	Fish_L.png	Fish_R.png	600	Incongruent
8		2	Practice_Trials	Right	Fish_R.png	Fish_L.png	800	Incongruent
9		2	Practice_Trials	Left	Fish_L.png	Fish_L.png	800	Congruent
10		2	Practice_Trials	Right	Fish_R.png	Fish_R.png	600	Congruent
11		2	Practice_Trials	Left	Fish_L.png	Fish_R.png	1000	Incongruent
12		2	Practice_Trials	Right	Fish_R.png	Fish_L.png	400	Incongruent
13		2	Practice_Trials	Left	Fish_L.png	Fish_L.png	1000	Congruent
14		2	Practice_Trials	Right	Fish_R.png	Fish_R.png	400	Congruent
15		2	Practice_Trials	Left	Fish_L.png	Fish_R.png	800	Incongruent

The "Display" columns mirrors the sections that have been created in our "Task Structure". Let's focus on the "Practice_Trials" display. In the "Target" and "Distractor" columns, we indicate which image will be uploaded for each trial. On the first Practice Trial, for example (Row 5), the Target is a fish pointing to the left, and the Distractors are also pointing to the left: it is actually a Congruent trial, as written in the Column "Type". The images for each zone (here it is only "Fish_L.png") have to be uploaded in the "Stimuli" menu.

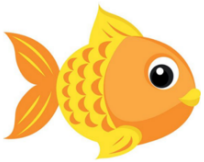
Task Structure | Spreadsheet | **Stimuli** | Manipulations | Script

+ Add New Stimuli | Showing page 1 of 1 | × Delete All Stimuli

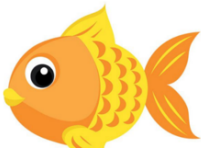
End_slide.jpg

Break_slide.jpg

Fish_R.png



Fish_L.png

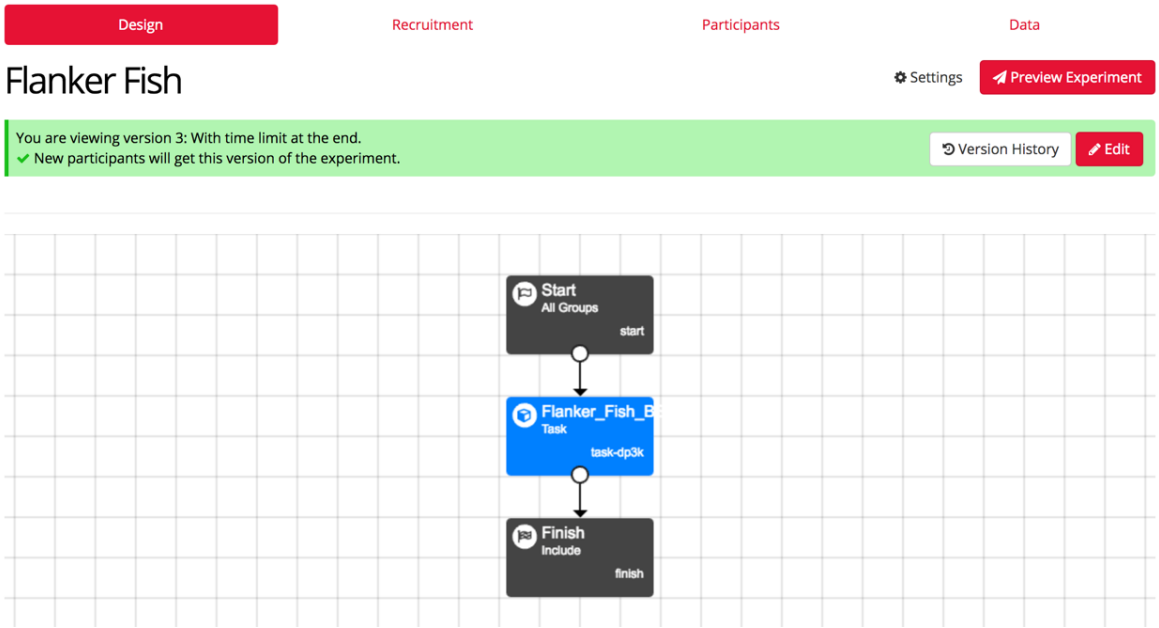


The last image is the one that is called "Fish_L.png". Once we have set up our task, using the "Task structure", the "Spreadsheet" and the "Stimuli" tabs, we need to put it in an experiment.

Putting the task into an experiment

Gorilla uses what we call an "Experiment tree". The tree specifies which tasks are used within an experiment, as well as their order. Here, the experiment (Called "Flanker Fish") is pretty simple, because we only use the

Flanker Task.



It is shown in blue, and it is surrounded by a "Start node", and a "Finish node", specifying the beginning and the end of the experiment. Many more functionalities are available in the "Experiment tree", allowing researchers to counterbalance the order of presentation for an experiment that contains several tasks, for example.

The "Recruitment tab" allows us to generate a link to share the experiment online, and to select how many participants we would like to recruit, along with any specific technical requirement (device types, browser types of connection speed that the participants would need to have). The "Participants" tab references all the participants that joined the experiment, and the "Data" tab allows us to download our data.