



# SimplePhy: An open-source tool for quick online perception experiments

Miguel A. Lago<sup>1</sup>

Accepted: 19 November 2020 / Published online: 14 January 2021  
© The Psychonomic Society, Inc. 2021

## Abstract

Because of the COVID-19 pandemic, researchers are facing unprecedented challenges that affect our ability to run in-person experiments. With mandated social distancing in a controlled laboratory environment, many researchers are searching for alternative options to conduct research, such as online experimentation. However, online experimentation comes at a cost; learning online tools for building and publishing psychophysics experiments can be complicated and time-consuming. This learning cost is unfortunate because researchers typically only need to use a small percentage of these tools' capabilities, but they still have to deal with these systems' complexities (e.g., complex graphical user interfaces or difficult programming languages). Furthermore, after the experiment is built, researchers often have to find an online platform compatible with the tool they used to program the experiment. To simplify and streamline the online process of programming and hosting an experiment, I have created SimplePhy. SimplePhy can save researchers' time and energy by allowing them to create a study in just a few clicks. All researchers have to do is select among a few experiment settings and upload the stimuli. SimplePhy is able to run most psychophysical perception experiments that require mouse clicks and button presses. In addition to collecting online behavioral data, SimplePhy can also collect information regarding the estimated viewing distance between the participant and the monitor, the screen size, and the experimental trial's timing—features not always offered in other online platforms. Overall, SimplePhy is a simple, free, open-source tool (code can be found here: <https://gitlab.com/malago/simplephy>) aimed to help labs conduct their experiments online.

**Keywords** Online · Perception · Psychophysics · Experiment · Open-source

## Introduction

In the age of online services, many researchers use the Internet to conduct their experiments. Compared to face-to-face experiments, running an experiment online has obvious advantages. For example, researchers are able to collect data from a diverse population (Reinecke & Gajos, 2015), avoid scheduling constraints, and have the potential for a significant increase in the number of participants. However, when it comes to online platforms, there are many challenges that psychophysical perception experiments face. Many perceptual decisions are sensitive to variables such as to monitor specifications, stimulus size, viewing distance, room illumination, and monitor brightness calibration (Birbaum & Birbaum, 2000; de Leeuw &

Motz, 2016). While some of these inconsistencies might balance out when running more participants (e.g., the differences in monitor sizes), many of these other factors require extra steps to alleviate it that are unreasonable to expect participants to perform (e.g., downloading software to get the monitor brightness or color calibration). To transition from in-person to online experimentation, researchers need to be aware of the advantages and disadvantages of running online experiments, carefully select the experiment builder that matches their requirements, and find a way to make their experiments publicly available on the Internet (see (Grootswagers, 2020) and (Sauter, Draschkow, & Mack, 2020) for a comprehensive review of the challenges in online experiment design and deployment).

Currently, there are many options to build psychophysical perception experiments online. The most popular options are PsychoJS (Peirce, 2007), OSWeb (Mathôt, Schreij, & Theeuwes, 2012), jsPsych (de Leeuw, 2015), PsyToolkit (Stoet, 2010), and lab.js (Henninger, Shevchenko, Mertens, Kieslich, & Hilbig, 2020). All of these tools require

---

✉ Miguel A. Lago  
lago@psych.ucsb.edu

<sup>1</sup> University of California, Santa Barbara, Santa Barbara, CA 93105, USA

**Table 1** Comparison of experiment-builders (PsychoPy, OSWeb, jsPsych, PsyToolkit, lab.js, and SimplePhy) including the presence of a graphical user interface (GUI), the coding language necessary to build experiments, and the integrated platforms to publish the experiment on the Internet

Tool	GUI	Coding	Integrated publishing platform
<b>PsychoJS</b>	✓	JavaScript (for advanced purposes)	Pavlovia
<b>OSWeb</b>	✓	JavaScript (for advanced purposes)	JATOS
<b>jsPsych</b>	✗	JavaScript	JATOS, Pavlovia, Cognition
<b>PsyToolkit</b>	✗	Own language	JATOS, PsyToolkit
<b>lab.js</b>	✓	JavaScript	JATOS, Open Lab, Pavlovia, Qualtrics
<b>SimplePhy</b>	✓	✗	SimplePhy

researchers to learn how to design and deploy their experiment with either a graphical user interface (GUI), a code (whether it is an ad hoc programming language or JavaScript), or both. Additionally, many of these tools allow the experiment design to be tweaked by changing the underlying HTML, CSS, and JavaScript code, which are the standard markup, styling, and scripting languages used to render a website. However, the major disadvantage is that they typically have a steep learning curve that keeps researchers from deploying their experiments in a timely manner.

GUI-based tools often require knowledge of the underlying programming language for more complex experiment designs (which may ultimately defeat the original intent of using a GUI-based system) and can be cumbersome to create (e.g., finding the location of the desired functionality may require a large number of repetitive actions: clicks, text input or drag-and-drop actions). Alternatively, programming-based tools require the researcher to learn the programming language or libraries and type the list of orders that the computer has to execute. For researchers, this learning is often done at the beginning of their graduate studies and may take months or even years for less experienced individuals. Because of the time constraints, deadlines, and prior obligations many researchers have, learning how to code at certain stages in one's academic career is impractical, if not impossible, for the less computer savvy. Finally, more experienced coders can also choose to program their own experiments using the standard JavaScript libraries.

After programming an experiment, researchers are then tasked with making their experiments available online for participants. Current online tools provide a variety of solutions. For example, jsPsych generates the files necessary to run the experiment and can also integrate with the platform Cognition<sup>1</sup>. Similarly, OSWeb can export the experiment to a self-hosted JATOS platform to publish it. PsychoJS and lab.js also provide a way to connect with hosting platforms like Pavlovia<sup>2</sup> to upload the designed experiments.

Alternatively, many of these tools allow for downloading the experiment files, leaving the researcher with the task of providing a web hosting service to publish the experiment. This can be beneficial for some experiments since the data collection is controlled by the researcher that can choose which platform to trust to store the data (or self-host it) but can also be inconvenient if that is not a concern. See Table 1 for a summary of experiment builders and their integrated publishing platforms.

The process of recruiting participants can also be done online. Pavlovia, Mechanical Turk<sup>3</sup>, Open Lab<sup>4</sup>, and Labvanced<sup>5</sup> are viable options that provide a place for researchers to host experiments or recruit participants. Some experiment builders (e.g., PsychoJS, OSWeb, and lab.js) can directly connect with these hosting platforms to automatically export the experiments, making this step straightforward. However, all these platforms come at a financial cost and some of them can only be integrated with specific builders. Alternatively, if the researcher is in a university, recruitment systems like SONA, a widespread recruitment and scheduling system for university subject pools, can also be used.

In this paper, I present SimplePhy, a simple open-source tool that integrates the process of creating and publishing online psychophysics experiments on the same platform. While existing tools allow for much flexibility in terms of what they can do, SimplePhy has been designed with basic perception experiments as its core and incorporates mechanisms to estimate the viewing distance, something crucial for some studies. SimplePhy has a simple GUI interface (Fig. 1) and requires no prior coding knowledge. The available options allow for a wide array of psychophysics experiment designs that either require keyboard presses or mouse clicks to indicate responses (e.g., visual search and multiple-alternative forced-choice, MAFC). Additionally, it can host the experiments for free after they have been created, something that not all platforms currently offer.

<sup>1</sup> <https://www.cognition.run/>

<sup>2</sup> <https://www.pavlovia.org/>

<sup>3</sup> <https://www.mturk.com/>

<sup>4</sup> <https://open-lab.online/>

<sup>5</sup> <https://www.labvanced.com/>

**SIMPLE\_PHY**

Try the test experiment: Enter your participant ID and click on Start. Calibration will start first.

Participant ID Start!

By clicking on Start you accept the experiment conditions and consent form (if given).

Create experiment

See tutorial

Title optional

Instructions URL optional

# ratings 8

Allowed keys Space Edit...

Randomize trials Randomize all

Multiple stimuli Sequential (default)

Calibration interval 10

Trial timeout (ms) optional

Marks Fixation Grid

Find stimuli on Google Drive

Create experiment

**Fig. 1** SimplePhy's initial screen with the Participant ID field and Start button for participants. The Create experiment box can be expanded to show the available options to create an experiment. Instructions on how to

use the latest version of the tool can be found in the GitLab repository wiki. <https://gitlab.com/malago/simplephy/-/wikis/home>

## From zero to data

To design an experiment with SimplePhy, researchers choose from a small set of options to configure the experiment and upload the stimuli images so that SimplePhy can link them (see Fig. 1 for a list of available options). Additionally, SimplePhy offers the option to estimate the participant's viewing distance by using the Virtual Chinrest method described by Li, Joo, Yeatman, and Reinecke (2020) (see Section II.F for more information). Once the experiment is generated, participants can then complete it in the same online platform by accessing the corresponding URL. Data is stored on a secure website and locked behind a password. With the known password, researchers can access and download the data at any time to analyze it.

## Experimental design

The first step in experimental design is to find what kind of experiment they need to conduct and what kind of stimuli

needs to be presented to participants. SimplePhy is compatible with stimuli in the form of static images (e.g., pictures or shapes) or videos (e.g., movies or animations). The experiment design allows for a few standard response options: mouse clicks, keyboard responses, confidence rating responses, and multiple alternative forced-choice (MAFC) responses. To facilitate participants with more information about the experiment, SimplePhy will also allow the researcher to include a link to a document with instructions<sup>6</sup>.

## Stimuli

After the options for experimental design are selected, researchers then need to create a specific folder structure containing the stimuli files that will be used by SimplePhy. The folder structure must strictly follow the organization shown in Table 2. Each experiment may have multiple conditions that

<sup>6</sup> Although the researcher is responsible for providing this document, solutions such as a public Google Document, a YouTube video or a ZIP file hosted in a service like Dropbox or OneDrive can be used.

**Table 2** Folder structure for stimuli import to be created in a Google Drive public folder

Root folder: one folder per condition	Condition level: one folder per trial	Trial level: stimulus information
		Image #1 (.jpg .webm .mp4)
Condition #1	Trial #1	<i>Image #2 (.jpg .webm .mp4)</i>
<i>Condition #2</i>	Trial #2	...
Condition #3	Trial #3	<i>Image #N (.jpg .webm .mp4)</i>
...	...	<i>Information file (.json)</i>
Condition #N	Trial #N	<i>Feedback image (feedback.jpg feedback.webm feedback.mp4)</i>

This structure may contain one or more folders for each corresponding condition. Each condition must contain one folder per trial. Each trial must contain one or more images/videos and, optionally, an information file and feedback image. Italics denote optional entries

contain multiple trials. Once the folder structure is finished, researchers must upload it to a Google Drive folder and make it publicly visible. SimplePhy will collect and save the trial files' public URL of each trial. Because stimuli files are stored in a Google Drive server, SimplePhy can remotely link those files (e.g., videos or images). Since they will be downloaded from the Google Drive server, this allows for reduced bandwidth load on the SimplePhy web server.

## Experiment options

To create an experiment, researchers need to: 1) select the options for their experiment design, 2) indicate the Google Drive folder in which they uploaded the stimuli, and 3) click on the “Create experiment” button. In order to streamline this process, SimplePhy provides a GUI startup screen that allows the users to select what specific parameters they want to use (Fig. 1).

To further customize an experiment, an optional JSON<sup>7</sup> file (a widely used format for data interchange in JavaScript applications) may be added to include additional information about the corresponding trial within each trial folder. Within this JSON file, users may include information to customize each trial's custom response options, instructions, allowed keyboard presses, and correct response. To include extra information for the results file, observers simply need to indicate what variables need to be included. For example, in a visual search task, it might be useful information about the target location within an image or the identity of the target. Any information stored in this information file will then be added to the participant's results file to facilitate the posterior analyses. Furthermore, this information file allows for advanced customization of the available responses at the end of each trial (see Fig. 2). Correct responses of each trial can also be included in the information file. If included, SimplePhy will save to the results file whether a response was correct or incorrect.

<sup>7</sup> <https://www.json.org/json-en.html>

While these options offer more flexibility to SimplePhy, this file does not interfere with GUI's simplicity. To facilitate the generation of these files, SimplePhy includes a JSON file generator GUI in case the user is unfamiliar with creating JSON files. See the GitLab wiki for more examples<sup>8</sup> of currently available advanced options.

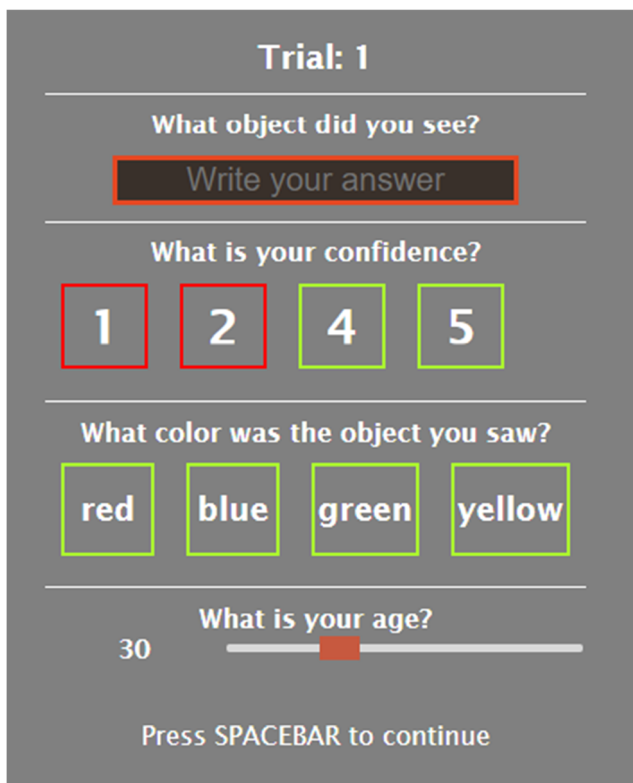
## Running an experiment

Once an experiment is created, SimplePhy will generate two URLs. The first URL is a direct link to the experiment, which the researcher can send to the participants. The second URL is a direct link to access all participants' data. In order to download participants' data, a password must be first established when entering the results section for the first time to ensure that no one can access the data without permission.

When participants receive the experiment URL, they will be shown a screen with the experiment title and an instructions link (if provided by the researcher; see Fig. 1). Participants can type in their participant ID in the text box. However, if participants do not type in a participant ID, one will be generated automatically. If the distance estimation calibration was enabled in the experiment design, the calibration screen would next be shown. After this, the web browser's full-screen mode will activate, and the participant will be presented with a trial. If the researcher uses the feedback option, a feedback screen will be provided after the participant finishes each trial. See Fig. 3 for an outline example of one trial.

In addition to all these features, SimplePhy includes safeguards to detect whether participants switched to a different software or browser tab once they have started the experiment. If the participant leaves the experiment, data collection immediately stops, and the current data is stored. If a participant returns to the experiment, a cookie in the web-browser will inform SimplePhy that the experiment needs to be resumed (trial start timestamp within the results will indicate if this

<sup>8</sup> <https://gitlab.com/malago/simplephy/-/wikis/Experiment-options>



**Fig. 2** Advanced custom response options. *From top to bottom:* text open response, confidence rating, custom options, rating slider for more than ten ratings

happens). Participants will then be asked whether they want to continue where they left off or to restart the experiment.

### Accessing the results

SimplePhy provides a results page where the researcher can view an overall summary page with the current number of participants and how much of the experiment they completed (Fig. 4). A download button will allow the researcher to download a compressed file with all participants’ data (in either JSON or CSV format). In order to keep the service free for researchers, currently, participants’ data that is older than 1 month may be subjected to be automatically removed from the storage server. Data is stored both in

SimplePhy’s web server and, for backup purposes, in a Google Firebase service. Although stored data is locked behind a password, it is not encrypted: researchers should avoid using identifiable information (i.e., use anonymous participant IDs). Researchers can download participants’ data at any time, as long as they are in the server, and are encouraged to do so at least every 30 days to avoid data loss.

### Viewing distance

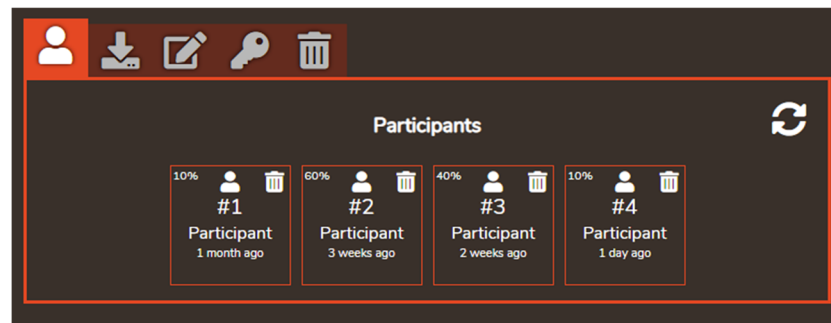
In perception experiments, controlling the participant’s viewing distance to the monitor, the monitor screen size, and the screen resolution is critical for researchers. Many perception experiments rely on a specific viewing distance or rely on knowing the stimulus size in degrees of visual angle. In order to account for these problems, SimplePhy is the first experiment builder that replicates the virtual chinrest developed by Li et al. (2020). Li and colleagues developed a way to compute viewing distance with a credit card. In particular, the credit card computes the pixel size in the screen and combines it with the human retinal blind spot to compute the pixels per degree of visual angle. In SimplePhy, researchers have the option to have participants place a credit card (which is universally the same size, 8.56 cm width) on the screen and adjust a slider to match the width of the credit card. SimplePhy uses this information to calculate how many pixels correspond to the size of the credit card, which leads to a value of pixels per centimeter for the participant’s monitor ( $px_{cm}$ ). After the measurement is taken, SimplePhy draws a cross on the right part of the screen and asks participants to fixate the cross while closing their right eye. A circle then moves from the cross leftward until it subjectively disappears, indicating that the circle is in the participant’s blind spot. Participants are then asked to press the spacebar when the circle disappears. This process is repeated five times, and the average distance is computed from these outcomes. As suggested by Li et al., SimplePhy assumes that the retinal blind spot is at 13 degrees of visual angle ( $dva$ ) (Wang et al., 2017) and use that to estimate the number of pixels



**Fig. 3** Outline example of one experiment: *Main screen:* Where participants type their participant ID. *Loading trial:* A filler screen that participants see as the files necessary for each trial are loaded. *Trial run:* Where the trial stimuli are displayed and the participants either press a key, make a mouse click, or wait for a designated timeout when viewing

the stimuli. *Response screen:* Where participants can respond to the questions regarding the previously shown stimulus. *Feedback Screen:* If the advanced feedback option is used, participants will see if their response was correct or incorrect





**Fig. 4** Example of the results page for an experiment. Researchers can set a password to see and download participants' data. From left to right, tab icons correspond to participants, download data, edit experiment configuration, change password, and delete the experiment and its associated data

per degree ( $px_{dva}$ ). Finally, using both pixels per centimeter and pixels per degree, SimplePhy infers the viewing distance of the participant via the following calculation:

$$px_{cm} = \frac{\text{credit card size (px)}}{8.56 \text{ cm}}$$

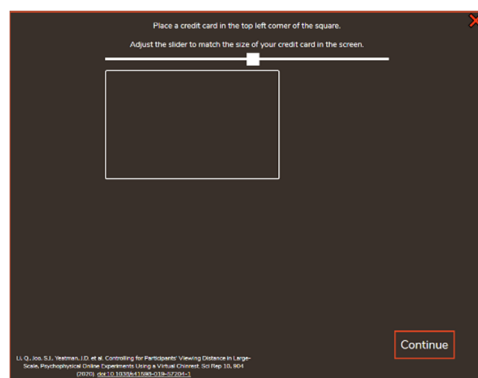
$$px_{dva} = \frac{\text{distance at which circle disappears (px)}}{13 \text{ dva}}$$

$$\text{viewing distance} = \frac{\tan(1 \text{ dva})px_{cm}}{px_{dva}}$$

This calibration process is optional and, if activated, will be completed at the start of the experiment. Additionally, a recalibration process can be required every certain number of minutes (controlled by the researcher) during the experiment. In each recalibration, a new distance is recorded, along with the time of the recalibration (see Fig. 5 for the two steps of the calibration procedure).

## Limitations

While there are many options (and customizable options) in SimplePhy, it has its limitations compared to other online experimental platforms. Here, I describe some workarounds to overcome these limitations.

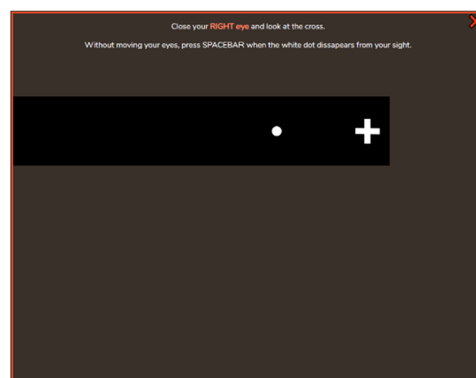


## Practice trials

A “practice experiment” can be generated using SimplePhy to present practice trials to the participant. This “practice experiment” would simply be a shorter experiment with fewer trials. Participants could then be instructed to complete the “practice experiment” before the actual experiment. Alternatively, a researcher can design an “instructions experiment” that shows non-randomized trials with screenshots or text intertwined with actual trials. With this experiment, participants can try the experiment while learning about the task and the tool. In both options, the researcher can set a threshold to assess a participant's performance. If their performance exceeds it, that indicates that the participant understood the given task, and then the researcher can send the link to the actual experiment. Unfortunately, since SimplePhy is always blind to the task, it is the researcher who has to download the participant's data and do a quick analysis to check if the participant understood the instructions. These solutions require extra micro-managing by the researcher but are ways to help ensure that participants understand the experiment.

## Between-subject and within-subject design

SimplePhy does not offer a native way to conduct between-subject experiments. However, in this case, researchers can



**Fig. 5** Credit-card calibration screen (left). Retinal blind spot calibration screen (right)

divide the experiment in two and provide the corresponding link to one of them to each participant. This comes with the cost of having extra management from the researcher. For within-subject design, SimplePhy offers different ways to control how conditions are shown (e.g., all randomized, conditions are blocked and randomized, or conditions are blocked and sequential).

### Participation control

In order to check that only authorized participants can perform the experiment, researchers can give them a specific, not easily guessed, participant ID (e.g., S1x45RL). The researcher can provide this ID to the participant and then double-check it to ensure that each personal ID is used and that it is only used once.

### Data analysis

Given the simplicity of the tool, SimplePhy does not assist researchers with data analysis (besides simple response checks, if the advanced options are used). In general, SimplePhy does not have any information about the experiment's purpose or how the participants' responses should be processed to determine if they are correct or wrong. For example, if mouse clicks are enabled, SimplePhy will record where the observers click on the screen, but not whether participants click in the "correct" location. The researcher must perform posterior analyses such as these after data is collected. However, simple responses can be analyzed by comparing participants' responses with the correct response in the trial information JSON file, in which case SimplePhy will show the participant a "correct" or "incorrect" screen if they press the correct key or choose the correct options in the response screen. An optional feedback image can be added to each trial to inform the participants what the correct answer was, regardless of their answer.

### Recruiting participants

SimplePhy leaves this task to the researcher, who will have to provide the experiment link to the participant. Since this cannot be done within the SimplePhy platform, recruiting must be outsourced to platforms like Mechanical Turk or SONA. This recruiting platforms only need a link to the experiment, making SimplePhy's generated unique link ideal for this task.

### Monitor calibrations

For all online experiments, SimplePhy included, the brightness calibration of the monitor cannot be controlled. For perception experiments that heavily rely on similar viewing conditions for all participants, this may result in noisy or biased

data and should be something to consider if choosing an online tool. Unfortunately, with a web-browser-based experiment, this is inherently unachievable in any online tool. The only option would be to ask the participant to input their monitor's model, luminance, and gamma function. While doable, this likely would create another entry barrier that might reduce the number of potential participants<sup>9</sup>.

### Other customizations

The current version of SimplePhy offers a limited amount of trial customization options (see section II.C for more details). If researchers need to control for other variables like trials that adapt to the responses, custom backgrounds, or more complex experiment logic, SimplePhy does not offer a solution for them. Although the tool might add some functionality in the future, it will maintain the simplicity as its core value: SimplePhy will never know what the purpose of the experiment is or what the meaning of correct or incorrect trial is (this would require some coding and it is something that other tools can do).

### Discussion

Designing a perception experiment can be a tricky task. Researchers are often tasked with: 1) making sure that data is effectively collected, 2) accounting/avoiding possible confounding variables that may arise from specific experimental designs (e.g., training effects, memory or vigilance decrement over time), 3) programming the scripts needed for their experiments, and 4) optimizing/fixing the code. (e.g., fixing bugs in the code, correcting non-optimized loops, recovering missing data, or accounting for timing inaccuracies). The purpose of SimplePhy is to *simplify* this entire process and reduce the time between psychophysics experiment design and data collection. Researchers only need to design their experiment, generate the stimuli, upload them, and select which options they want SimplePhy to use. The SimplePhy website handles the rest. More advanced options are available by modifying the trial information JSON file (with a helpful GUI interface that can be used to create it) to provide better feedback, add multiple questions, and choose different response types.

SimplePhy's purpose is to complement the existing experiment builders. If the researcher's experiment is simple and does not need many customizations, SimplePhy can be useful and speed up the data collection. For more complex experiments, researchers can always use the other experiment builders mentioned above, with the caveat that they might

<sup>9</sup> One can assume that monitors are calibrated with a standard gamma correction of 2.2 (Hwung, Wang, & Su, 1995) with the luminance that the participant finds more comfortable.

need to spend more time creating or programming the experiment logic. SimplePhy can also be a useful tool for piloting studies quickly, whether it is in a remote environment or the in-person lab.

## Conclusions

SimplePhy is a simple and free way to create psychophysics perception experiments without the hassle of programming and finding a host for an online experiment. It is designed to provide a seamless experience to generate simple experiment designs suitable for many labs and research groups. While current, complete psychophysics experiment builder tools provide more flexibility than SimplePhy, they come with the caveat of having to be more careful while programming the experiment. These tools may ultimately increase the amount of preparation time required by the researcher and the chance of introducing bugs into the code. If the psychophysics experiment does not require complex workflows or non-standard responses, SimplePhy is a tool that researchers should consider. Finally, SimplePhy is an open-source tool, and its source code is publicly available. It is easily extendable, free to use, and will continue to be developed in the future, adding options for more experiment designs while keeping the simplicity as its core value and appeal.

**Acknowledgements** The author thanks Dr. Stephen Adamo, Dr. Miguel Eckstein, and all the VIU lab at UC Santa Barbara for their support during the development of this tool and helpful feedback on the manuscript writing. This research was funded by the National Institute of Health grants R01 EB018958 and R01 EB026427.

## References

Birnbaum, M. H., & Birnbaum, M. O. (2000). *Psychological experiments on the Internet*. Elsevier.

- de Leeuw, J. R. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a Web browser. *Behavior Research Methods*, 47(1), 1–12. <https://doi.org/10.3758/s13428-014-0458-y>
- de Leeuw, J. R., & Motz, B. A. (2016). 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. <https://doi.org/10.3758/s13428-015-0567-2>
- Grootswagers, T. (2020). A primer on running human behavioural experiments online. *Behavior Research Methods*. <https://doi.org/10.3758/s13428-020-01395-3>
- Henninger, F., Shevchenko, Y., Mertens, U., Kieslich, P. J., & Hilbig, B. E. (2020). *lab.js: A free, open, online experiment builder*. Zenodo. <https://doi.org/10.5281/zenodo.3767907>
- Hwung, D.-J., Wang, J.-C., & Su, D.-S. (1995). *Digital gamma correction system for low, medium and high intensity video signals, with linear and non-linear correction*. Google Patents.
- Li, Q., Joo, S. J., Yeatman, J. D., & Reinecke, K. (2020). Controlling for Participants' Viewing Distance in Large-Scale, Psychophysical Online Experiments Using a Virtual Chinrest. *Scientific Reports*, 10(1), 1–11. <https://doi.org/10.1038/s41598-019-57204-1>
- Mathôt, S., Schreij, D., & Theeuwes, J. (2012). OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods*, 44(2), 314–324. <https://doi.org/10.3758/s13428-011-0168-7>
- Peirce, J. W. (2007). PsychoPy—Psychophysics software in Python. *Journal of Neuroscience Methods*, 162(1), 8–13. <https://doi.org/10.1016/j.jneumeth.2006.11.017>
- Reinecke, K., & Gajos, K. Z. (2015). LabintheWild: Conducting Large-Scale Online Experiments With Uncompensated Samples. *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 1364–1378. <https://doi.org/10.1145/2675133.2675246>
- Sauter, M., Draschkow, D., & Mack, W. (2020). Building, Hosting and Recruiting: A Brief Introduction to Running Behavioral Experiments Online. *Brain Sciences*, 10(4), 251. <https://doi.org/10.3390/brainsci10040251>
- Stoet, G. (2010). PsyToolkit: A software package for programming psychological experiments using Linux. *Behavior Research Methods*, 42(4), 1096–1104. <https://doi.org/10.3758/BRM.42.4.1096>
- Wang, M., Shen, L. Q., Boland, M. V., Wellik, S. R., De Moraes, C. G., Myers, J. S., Bex, P. J., & Elze, T. (2017). Impact of natural blind spot location on perimetry. *Scientific Reports*, 7(1), 1–9.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.