Journal of the Economic Science Association
https://doi.org/10.1007/s40881-020-00087-0

EXPERIMENTAL TOOLS

LIONESS Lab: a free web-based platform for conducting interactive experiments online

Marcus Giamattei[1,2,3] · **Kyanoush Seyed Yahosseini**[4] · **Simon Gächter**[2,5,6] · **Lucas Molleman**[2,4,7]

Received: 25 March 2019 / Revised: 28 May 2020 / Accepted: 6 June 2020

Abstract

LIONESS Lab is a free web-based platform for interactive online experiments. An intuitive, user-friendly graphical interface enables researchers to develop, test, and share experiments online, with minimal need for programming experience. LIONESS Lab provides solutions for the methodological challenges of interactive online experimentation, including ways to reduce waiting time, form groups on-the-fly, and deal with participant dropout. We highlight key features of the software, and show how it meets the challenges of conducting interactive experiments online.

Keywords Experimental software · Interactive online experiments · Experimental standards

JEL Classification C90

✉ Marcus Giamattei
m.giamattei@berlin.bard.edu

[1] Chair in Economic Theory, University of Passau, Innstraße 27, 94032 Passau, Germany

[2] Center for Decision Research and Experimental Economics, University of Nottingham, University Park, Nottingham NG7 2RD, UK

[3] Bard College Berlin, Platanenstr. 24, 13156 Berlin, Germany

[4] Center for Adaptive Rationality, Max Planck Institute for Human Development, Lentzeallee 94, 14195 Berlin, Germany

[5] Institute of Labour Economics, Schaumburg-Lippe-Straße 5-9, 53113 Bonn, Germany

[6] Center for Economic Studies, Poschingerstraße 5, 81679 Munich, Germany

[7] Amsterdam Brain and Cognition, University of Amsterdam, Nieuwe Achtergracht 129b, 1018 WT Amsterdam, The Netherlands

Published online: 27 June 2020

1 Introduction

A rapidly growing number of behavioural researchers use online experiments to study human decision-making. Online labour markets, such as Amazon Mechanical Turk (MTurk; www.mturk.com) and Prolific (www.prolific.co), allow researchers to conveniently recruit participants for experiments and compensate them for their efforts. The quality of data from online experiments is generally deemed comparable to data obtained in the laboratory (Berinsky et al. 2012; Buhrmester et al. 2011; Hauser and Schwarz 2016; Mason and Suri 2012; Paolacci and Chandler 2014; Paolacci et al. 2010; Snowberg and Yariv 2018; Thomas and Clifford 2017; but see Hergueux and Jacquemet 2015), making online experimentation a promising complement to laboratory research. However, online experiments have typically used non-interactive tasks that participants complete on their own, either using survey software (e.g., SurveyMonkey, Qualtrics) to document decisions or emulating social interactions by using the strategy method and matching participants post hoc. Online studies using designs with live interactions between participants have typically employed tailor-made software (Egas and Riedl 2008; Gallo and Yan 2015; Nishi et al. 2015; Schmelz and Ziegelmeyer 2015; Suri and Watts 2011; Wang et al. 2012).

A number of software platforms are currently available for conducting experiments via the Internet, at varying stages of development.[1] Typically, the use of these platforms for interactive experiments online requires considerable programming skills, and involves substantial installation and setup times. Moreover, these platforms do not provide integrated methods to address the specific logistic and methodological challenges of conducting interactive experiments with participants recruited online (Arechar et al. 2018). As a result, the online use of interactive designs has thus been largely restricted to experimenters with advanced technical skills or considerable financial resources, limiting the potential of online experimentation for behavioural research.

Here we introduce LIONESS Lab, a free web-based platform that enables experimenters to develop, test, run, and share their interactive experiments online. The software is developed and maintained at the Centre for Decision Research and Experimental Economics (University of Nottingham, UK) and the Chair of Economic Theory (University of Passau, Germany) and can be accessed via https://lioness-lab.org. LIONESS stands for *L*ive *I*nteractive *ON*line *E*xperimental *S*erver *S*oftware. LIONESS *Lab* provides an intuitive online interface to develop LIONESS *experiments*. LIONESS experiments include a standardized set of methods to deal with the typical challenges arising when conducting interactive experiments online (Arechar et al. 2018). These methods reflect current 'best practices', e.g., for preventing participants to enter a session more than once, facilitating on-the-fly formation

[1] These software platforms include BreadBoard, ConG (Pettit et al. 2014), MobLab, NodeGame (Balietti 2016), oTree (Chen et al. 2016), Psynteract (Henniger et al. 2017), SMARTRIQS (Molnar 2019), SOPHIE (Hendriks 2012), and UbiquityLab; see Chan et al. (2019) for a recent comparison of web-based software platforms with respect to various features. A comparison of desirable features between LIONESS Lab and the most prominent similar software platforms—z-Tree and oTree (see Table 1 in the Appendix).

of interaction groups, reducing waiting times for participants, driving down attrition by retaining attention of online participants and, importantly, adequate handling of cases in which participants drop out.

A key distinguishing feature of LIONESS Lab is that researchers require only minimal programming skills to develop and conduct their own interactive online experiments. The LIONESS Lab platform provides a user-friendly environment to create and edit LIONESS experiments in a point-and-click fashion. No installation is needed, and programming is only required for calculations inherent to the researchers' specific experimental design. At the same time, LIONESS Lab supports adding JavaScript to experiments, allowing for great flexibility in implementing design features (for example, dynamic elements or conditional display). Experimenters can create new experiments starting from scratch or import existing designs (e.g., created by other researchers) through a repository. The repository enables experimenters to share their experimental designs with co-workers and other colleagues, allowing researchers to base their experiments on designs of others and facilitating transparency and replicability of research (Camerer et al. 2016; Munafò et al. 2017; Open Science Collaboration 2015).

Testing LIONESS experiments is facilitated by a test mode including a 'robot' feature that simulates participant responses (or can be programmed to generate custom responses). Experiments can be downloaded and used on the experimenter's own server. Participants access the experiment through a link (e.g., posted on MTurk or Prolific). Experimenters can monitor the progress of a session through a control panel. Upon session completion, data can be exported as a spreadsheet ready for analysis. This spreadsheet includes a tab for automating the performance-based payment of participants through online labour markets.

2 Overview

Figure 1 provides a general overview of LIONESS Lab. Experimenters access the platform using a free-of-charge account, where they can develop, test and share their LIONESS experiments. Developing, testing, and sharing is done on the LIONESS Lab server to allow for an easy start with no need to set up a server. Once a LIONESS experiment has been developed, it can be downloaded and used on a personal server, giving the experimenter full control over their experimental data and allowing researchers to adjust server capacities according to their needs.[2] Each experiment is a standalone program comprising a set of files defining the experimental screens that participants will navigate during a session.[3] This program incorporates standardised methods for dealing with participant dropout, one of the most challenging

[2] Guidelines for setting up a server are available in the online documentation (https://lioness-lab.org/documentation).

[3] Downloaded experiments include support files to set up a database to store the participants' decisions, control the flow of the experiment, regulate communication between the server and the online participants, and allow the experimenter to monitor a session's progress through the control panel.

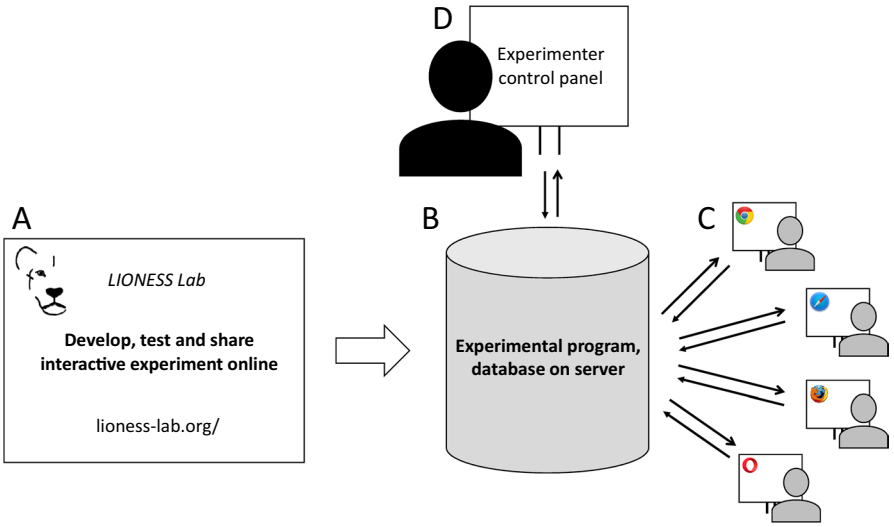


Fig. 1 Basic structure of LIONESS Lab. The online user interface of LIONESS Lab facilitates easy development and testing of interactive experiments (A). LIONESS Lab produces LIONESS experiments (B), which contain the software and a database to conduct sessions with participants who can be recruited from online platforms such as Amazon Mechanical Turk (MTurk) or Prolific. Participants access experiments through a web link and interact via their web browsers (C). The experimenter can monitor the progress of a session via the control panel (D)

aspects of interactive online experimentation (Arechar et al. 2018). These methods have been thoroughly tested in a range of different interactive and non-interactive experimental designs, conducted by various research groups and with participants recruited online and in the field (for a list of publications based on LIONESS experiments, see Table 2 in the Appendix; ongoing projects include LIONESS experiments with online groups of up to 16 participants and over 1000 participants per session). For a demo experiment, see https://lioness-lab.org/demo.

Researchers and experimental participants use LIONESS Lab online; no installation procedures are needed. Participants can complete experiments on devices with an internet connection, including laptops, tablets, and smartphones. LIONESS Lab was specifically designed for conducting interactive experiments online with participants recruited from crowd-sourcing platforms or from the participant pool of research institutions. However, it can also be used in the laboratory (Arechar et al. 2018; Molleman and Gächter 2018). An important advantage of this portability is that the screens of experimental participants are exactly the same in the physical lab and online, facilitating comparisons between the two. Furthermore, it enables experimenters to complement online studies with data from the physical lab, for example, to test the robustness of their results in more highly controlled lab conditions (a request that may well occur during the referring process of papers conducted solely online).

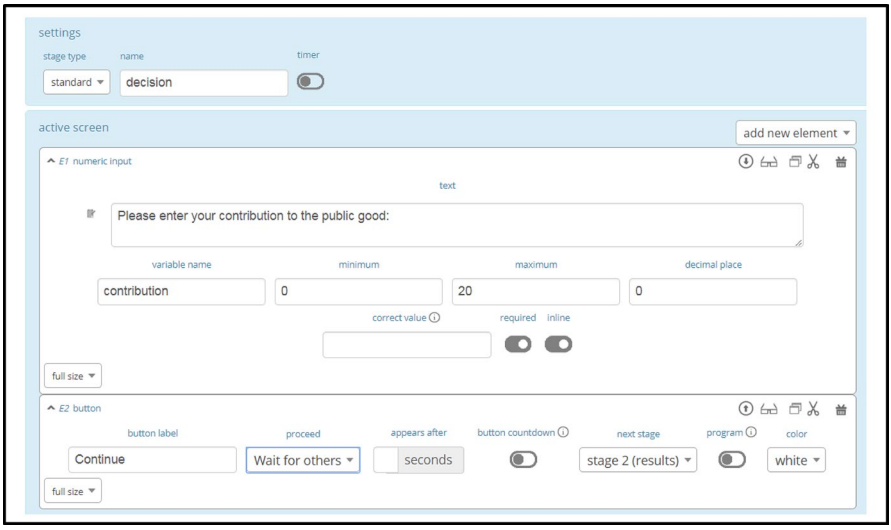


Fig. 2 Defining a stage by adding elements to experimental screens. The top horizontal box is used for specifying basic features of a stage (its name or an optional timer). The section below defines the 'active screen' (see main text), containing two elements. The first element (E1) defines a numeric input which prompts participants to input their contribution to a public good (an integer between 0 and 20). The second element (E2) defines a button which allows participants to proceed (to a stage called 'results') as soon as all group members have submitted their decision

3 Developing experiments

Experimenters can register for an account on https://lioness-lab.org. Upon login, one can choose to either start developing a LIONESS experiment from scratch, or to build upon existing experiments which can be imported through the repository (Sect. 6).

LIONESS Lab aims at making the development of experiments as simple and intuitive as possible. To this end, the user interface allows for developing experiments by creating them stage by stage.[4] Each stage corresponds to an experimental screen which participants will navigate during a session. Experimenters can define each stage in a point-and-click fashion by adding 'elements' in the order they want them to appear on the participants' screens. The development interface largely displays these elements in the way they will be shown to experimental participants.

Figure 2 illustrates how stages are specified in LIONESS Lab. In many economic experiments, participants are required to provide numerical input (e.g., place a bid in an auction, or make a contribution to a public good). To specify such an input stage,

[4] The interface is based on classEx, a widely used online tool for classroom experiments (Giamattei and Lambsdorff 2019). The development of experiments by stages—and adding elements to those stages—largely follows the logic of z-Tree (Fischbacher 2007).

```
contributions = getValues('contribution');
sum = 0;
for (i=0; i<3; i++) sum += contributions[i];
share = sum * multiplier / 3;
```

Fig. 3 An example of a JavaScript program. This code reflects the logic of a three-player public goods game. The function getValues() retrieves a vector of values from the database, containing the contributions of all members of a focal participant's group in the current period. With the for loop, all contributions are summed up, and the focal participant's share is calculated by multiplying this sum (using a custom parameter multiplier) and dividing it by the group size

an experimenter needs two elements: a 'numeric input' and a 'button' to submit the input and continue to the next stage. Experimenters add these elements to a stage with a dropdown menu and specify accompanying texts to show to participants. In the button element, the experimenter specifies which stage participants will be directed to next, and adds a condition upon which participants can continue to that next stage (e.g., 'as soon as possible', or 'wait for others').[5]

A range of different types of elements can be added to stages. These include text boxes to display information, a chat box, and various input boxes for recording participants' responses such as numeric or text input fields, sliders, radio buttons, and discrete choices. Participants' input is automatically validated before it is written to the database which stores all responses in the experiment.[6] For each input element, experimenters can specify 'display conditions' so that the elements are only displayed when certain conditions are met. This is often useful, e.g., when defining treatment variations.

Dynamic features can be added to stages with JavaScript elements, which allow for manipulation of variables. JavaScript elements provide predefined functions to access data from (and write data to) the experiment's database, and can be used to perform any calculations needed for the specific experimental design at hand. An example of a simple JavaScript program is shown in Fig. 3. JavaScript is the most widely used language for programming dynamic webpages and is supported by all modern web browsers. Due to its large user base, solutions for programming issues can be found easily. Experimenters with more advanced programming skills can use the great flexibility of JavaScript to incorporate their own custom-made components into experimental stages.

Stages have two different 'screens': an 'active screen' and an optional 'waiting screen'. Many experimental designs require that, at times, participants will only be allowed to proceed to the next stage when all members of their group have submitted

[5] Participants cannot navigate the experimental pages at will (e.g., by using their browsers' 'back' and 'forward' buttons). Each time participants are directed to a new stage, the browser history will be overwritten so that participants cannot navigate back. When participants enter an experiment, a script automatically checks whether their browser allows for overwriting the history. If not, the participant will be prompted to enable this functionality.

[6] Validation procedures on the client-side check whether the responses of experimental participants are in the right format (for example, they cannot submit a decimal number when an integer is required). In addition, the PHP code for processing these requests on the server side contains standardized checks to prevent PHP injection. See Sect. 7 for more information about data security.

their response in the current stage. In these cases, participants will be directed to the waiting screen of that stage after making their decision. As soon as all group members have made their decisions, they will all be directed to the next stage.[7]

4 Test experiments

At any point during the development of their LIONESS experiment, users can choose to compile and test the experiment. Once the compilation process has finished—usually within less than a second—the control panel of the experiment opens in a new browser tab. The control panel is the centre of a LIONESS experiment, responsible for general coordination (Sect. 5). It includes a switch to activate a 'test mode' designed to facilitate testing during development. The test mode allows experimenters to start multiple mock participants (called 'test players') within the same browser, or to start a 'robot' participant generating random responses in each of the stages. Using robots is particularly useful for testing designs for larger groups, so that the researcher can test the experiment from the perspective of one group member, while the responses of the other group members are generated automatically.

While testing, experimenters can check the design of their experimental screens, and make adjustments as they go through them by updating their experimental screens in LIONESS Lab and refreshing the experimental screen they are viewing as a test player. In the control panel, experimenters can verify whether all variables are correctly recorded in the database. Experimenters can share the link to the experiment with collaborators, who can then instantly view the implemented design. The development interface also allows for downloading all experimental screens as a single webpage which can then be shared with collaborators for checking.

Once experimenters have finished specifying their LIONESS experiment, they can download it by clicking a button.[8] This will yield a standalone program that can subsequently be uploaded to a privately-owned server. This setup ensures that experimenters have full control over the data generated by their participants and that they can tailor server capacity to the prospective number of participants.[9]

[7] Note that this setup again follows the logic of z-Tree (Fischbacher 2007).

[8] LIONESS Lab generates LIONESS experiments as a set of PHP files downloadable in zip format. These files include JavaScript and HTML code for displaying the experimental screens in the participants' browsers and an SQL file to set up the database. This setup is very common for websites and can be readily deployed on a web server using a so-called LAMP stack, which is freely available as an off-the-shelf server that can be set up in a few mouse clicks. Detailed instructions are available in the documentation (https://lioness-lab.org/documentation).

[9] There are many user-friendly cloud services available to set up a private virtual server (e.g., on Google Cloud and Amazon Web Services).

5 Conducting experiments online

Once the experimental screens have been specified and the LIONESS experiment has been uploaded to the server of the experimenter, participants can be invited (e.g., via MTurk or Prolific) and the data collection process can start. Here, we briefly describe how LIONESS experiments deal with the challenges of conducting interactive experiments, from starting up a session, through the interaction phase, to payment of participants. We focus on the technical aspects of LIONESS lab; an extensive methodological discussion of conducting interactive experiments online can be found in Arechar et al. (2018).

Start-up phase Before starting a session, experimenters go to the control panel of their LIONESS experiment, where they find a web link through which participants can access the experiment. Once the experiment is 'activated' by clicking a button in the control panel, the link directing prospective participants to the experiment can be posted as a 'job' on a crowd-sourcing website.[10] By default, LIONESS experiments store participants' IPs (after one-way encryption to meet data privacy requirements) to block participants who try to enter a session more than once. After a further check for browser support,[11] a participant can enter the experiment. After reading instructions, participants typically complete control questions. LIONESS Lab allows for adding 'quiz' stages, automatically recording the number of attempts that participants needed for solving each item. It is also possible to specify a maximum number of failed attempts, after which a participant is excluded from further participation in the experiment.

Once participants have read the instructions and successfully completed the quiz, they are ready to be matched. Matching takes place in a 'lobby' stage in which they wait for other participants with whom they will form a group. Experimenters can choose to inform participants in the lobby about the number of participants necessary to form a group. In case participants cannot be matched into a group within a predefined time limit, they are directed to a screen where they can choose to either return to the lobby or to leave to another experimental stage (e.g., ending the experiment, or presenting an alternative task).

Experimenters can choose from various pre-set matching procedures. As the number of participants in online experiments often cannot be determined exactly, each of these matching procedures flexibly forms groups 'on-the-fly'. In online experiments, statically predefining the composition of matching groups is often infeasible as some participants may drop out before being matched. The default option 'first-come, first-served' ensures that groups will be formed as soon as a sufficient number of participants are in the lobby. This minimizes waiting time and reduces dropout (Arechar et al. 2018).[12] Available standard matching procedures can be based on

[10] LIONESS experiments are not embedded in MTurk or Prolific (or any other platform), but function as a standalone program to which participants are directed. At the end of the experiment, participants return to the crowd-sourcing website.

[11] Most importantly, this procedure checks whether JavaScript is enabled, and prompts a participant to switch it on if disabled. NB: LIONESS experiments do not support Internet Explorer, which is an outdated browser and has a relatively small (and declining) share of users.

[12] One concern might be that faster participants are more likely to be grouped together as they finish the instructions earlier than others. Such effects of 'assortment' are mitigated by the fact that participants enter the experiment at different times: slow participants who entered early might be matched with fast participants who entered later.

treatments (grouping participants from the same treatment) or unique roles (grouping participants with different roles). Alternative matching procedures—such as stranger matching, rotation, or matching based on responses prior to the lobby—can be implemented with custom JavaScript code. Various implemented examples of custom matching procedures, including (perfect) stranger matching, are publicly available from the LIONESS repository.

Interaction phase Once a group has been formed, participants are directed to the first stage of a period. The flow of an experiment is centrally regulated by the control panel, ensuring that all group members move through the periods of the experiment in synchrony.[13] In the control panel, experimenters can track the progress of participants during an experimental session (Fig. 4).

Participant dropout is a great challenge for online experimentation (see, e.g., Zhou and Fischbach 2016), particularly for interactive designs (Arechar et al. 2018). Experimenters can reduce dropouts by using attractive pay rates, conditioning payment upon completion, and carefully managing participants' expectations from the outset (see, e.g., Horton et al. 2011). Complementing these general measures, LIONESS has methods in place to further reduce participant dropout in interactive experiments and, importantly, to deal with logistical issues should dropouts occur. The time that participants wait on others (and associated waning attention; a major source of dropouts) can be mitigated by adding timers to experimental screens, keeping up the pace of the experiment.[14] By default, waiting screens display an animated spinning wheel to ensure participants that the experiment is still running. Furthermore, if an experiment progresses to a new stage while a participant has the experimental screens in the background of their device (which can occur when participants are waiting for others and get distracted), an overlaying notification will be shown.

Despite these measures, dropouts cannot be avoided altogether. The controller algorithm detects when a participant loses connection or fails to respond in time. Experimenters can select pre-set options defining what happens to an unresponsive participant as well as the other members of this participant's group. By default, unresponsive participants are removed from the session[15] and the other members can continue in a group reduced in size. This way, the earnings of remaining participants will likely match their expectations (which would not happen if the whole group was terminated or stalled after a dropout of a group member; management of participant expectations is key for the online reputation of experimenter accounts on crowd-sourcing websites, which can be easily damaged by disappointed participants). Alternative options to handle dropouts—e.g., terminating the interaction

[13] Advanced users may tweak the experimental flow with JavaScript to allow more complex design features required for e.g., market experiments, where some players of a group may drop out after having traded while others are still trading.

[14] Participants can be asked to respond before the timer reaches zero, and if they fail to do so, they can be directed to another stage, or be excluded from further participation in the experiment.

[15] In case they try to return to the experimental web pages they are led to a screen telling them that their session is over.

PGGame – Control panel

Address for participants: lioness-lab.org/PGGame/_beginParticipant.php

core | decisions | globals | logEvents | session

> Display options

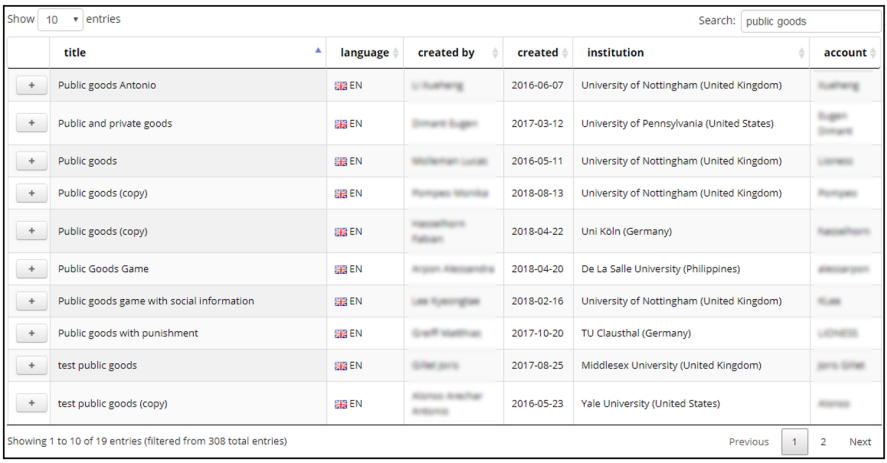
playerNr	groupNrStart	period	onPage
6	2	4	*** decision ***
7	2	4	- decision -
8	2	4	- decision -
1	1	11	*** earnings ***
2	1	11	*** earnings ***
4	1	11	*** earnings ***
3	0	1	*** control questions ***
5	0	1	*** Lobby ***
9	0	1	*** instructions ***

Fig. 4 Experimenter control panel. The experimenter can track the progress of a session and browse the tables of the database underlying the experiment (with the tabs under the web link for participants). The 'core' table shown contains useful variables to monitor participants' progress. The 'display options' button opens a menu that allows the experimenter to filter and sort the variables they want to see. This example experiment involves a three-player ten-period public goods game (called 'PGGame'). Nine participants have entered the session. The top three rows in this table show a group which is currently in period 4. Two participants of this group have already submitted their choice and are in the waiting screen of the 'decision' stage (indicated with the hyphens). One player is in the active screen of that stage (indicated with the asterisks). Another group has finished the experiment, and all members are in the 'earnings' screen. Three participants are not yet in the interaction phase of the experiment. One of these participants is currently in the 'instructions' stage, one is in the 'control questions' stage, and one has already completed these, and is waiting in the 'lobby' until a group of three can be formed

phase of a group when one member drops out, directing the other group members to an alternative stage or directly to the end of the experiment—are available.[16] Using JavaScript, experimenters can also define programmed responses to replace group members who dropped out (e.g., based on previous responses of other participants). A number of example experiments implementing such 'robots' are available from the repository (see Sect. 6 below).

Payment phase At the end of an experiment, participants typically receive a unique completion code, which they can enter on the crowd-sourcing platform to get paid. For performance-specific payment, LIONESS experiments link the amount of 'bonus payments' to the completion codes.

[16] When dropouts are unlikely to happen (e.g., when running a LIONESS experiment in the physical laboratory, or when using non-interactive tasks), measures related to dropout handling can also be deactivated.


Show 10 entries Search: public goods

	title	language	created by	created	institution	account
+	Public goods Antonio	EN	[redacted]	2016-06-07	University of Nottingham (United Kingdom)	[redacted]
+	Public and private goods	EN	[redacted]	2017-03-12	University of Pennsylvania (United States)	[redacted]
+	Public goods	EN	[redacted]	2016-05-11	University of Nottingham (United Kingdom)	[redacted]
+	Public goods (copy)	EN	[redacted]	2018-08-13	University of Nottingham (United Kingdom)	[redacted]
+	Public goods (copy)	EN	[redacted]	2018-04-22	Uni Köln (Germany)	[redacted]
+	Public Goods Game	EN	[redacted]	2018-04-20	De La Salle University (Philippines)	[redacted]
+	Public goods game with social information	EN	[redacted]	2018-02-16	University of Nottingham (United Kingdom)	[redacted]
+	Public goods with punishment	EN	[redacted]	2017-10-20	TU Clausthal (Germany)	[redacted]
+	test public goods	EN	[redacted]	2017-08-25	Middlesex University (United Kingdom)	[redacted]
+	test public goods (copy)	EN	[redacted]	2016-05-23	Yale University (United States)	[redacted]

Showing 1 to 10 of 19 entries (filtered from 308 total entries) Previous 1 2 Next

Fig. 5 LIONESS Lab repository. Experimenters can choose to share their LIONESS experiments with other researchers. In this screenshot we have used the 'search' field (top right) to look for experiments that contain the key words 'public goods'. Once experimenters import an experiment to their own account (by clicking the '+' sign), it can be viewed, copied and edited

At the end of a session, experimenters can download their data via the control panel. By clicking a button, the browser will download an Excel file. This file includes each of the tables of the database underlying the experiment, which are shown in separate tabs. The downloaded file also contains tabs to help experimenters automating the bonus payment of online participants in a few simple steps. An extensive description of how to do this for payments on MTurk can be found in the online documentation (https://lioness-lab.org/documentation).

6 Sharing experiments through a repository

Users of LIONESS Lab can choose to share their experiments with others through the repository (Fig. 5). Sharing experiments with co-authors facilitates collaboration during development and testing. Moreover, the repository enables other colleagues to view the experiment and replicate results once the data has been collected and a study has been completed.[17] The repository aims to promote transparency and replicability of research, which is essential to the reliability of scientific research in general (Camerer et al. 2016; Munafò et al. 2017; Open Science Collaboration 2015), and to the relatively young field of behavioural online experimentation in particular (Stewart et al. 2017). Finally, by making their experiments publicly available to other LIONESS Lab users, experimenters contribute to the range of experimental designs available for others to view, edit, and build upon. Experimenters can avoid

[17] In addition to sharing their experiment through the repository of LIONESS Lab, researchers can share the downloaded experimental files in any way they wish, e.g., add them to any online scientific data repository along with their experimental data and analysis code upon publication of their paper.

reinventing the wheel and copy solutions to common issues in developing their designs, helping speed up the interaction between theory and empirics. In addition, the repository is frequently used to ask for help in the online discussion forum (see Sect. 8) and to provide example solutions to raised issues.

7 Data security and privacy

LIONESS Lab runs on a server which is professionally maintained by the University of Passau IT services, to guarantee minimal downtimes and high stability. To register for an account, experimenters have to accept the terms of use, which include references to the EU General Data Protection regulation (GDPR) to regulate the collection of personal data (from experimenters). LIONESS Lab does not store any personal data of participants. LIONESS Lab further implements the standard EU cookie policy with a disclaimer on the login screen. Communication is secured via standard HTTPS protocols. Intellectual property rights are clearly defined in the terms of use.

Basic aspects of LIONESS experiments, such as the flow of participants, matching in groups, synchronization between participants and screen timers are controlled on the server side, and by the control panel (which runs in the experimenter's browser). For participant screens, however, LIONESS experiments depend to a large extent on JavaScript, which runs in the participants' browsers. This has a range of advantages (very high flexibility, most programming issues are easily addressed by a Google search, not having to rely on web sockets, which are often blocked by institutions) but also has a number of drawbacks. For example, experimental participants with advanced web programming skills might be able to interact with the web pages and view restricted information using the JavaScript console of their browser. LIONESS experiments cannot completely rule out such attempts, but they do make it hard (and generally not worthwhile, or feasible under time pressure) for participants to try and circumvent such restrictions. Display conditions are enforced every 100 ms so that manipulation requires more advanced skills than just changing the display setting of an element on a page. Finally, external data manipulation attempts are curbed on the server side by using internal identifiers for the database and its tables are hard-coded into an experiment's PHP files, and standard measures are in place to prevent SQL injections.

8 Conclusion and future outlook

In this paper we introduce LIONESS Lab, a free platform for the development of interactive online experiments with minimal need for programming experience. By allowing researchers to conveniently develop, test, run, and share their interactive experimental designs, LIONESS Lab aims at helping online behavioural research reach its full potential.

LIONESS experiments include thoroughly tested measures to deal with the methodological and logistical challenges of conducting interactive experiments online (Arechar et al. 2018). Most importantly, these measures drive down participant dropouts and adequately deal with situations in which dropouts do occur.

LIONESS Lab enhances the potential of online behavioural research and increases the number of ways in which it can complement experimental research conducted in the physical lab. Research could benefit from systematic comparisons between results obtained in the physical lab and results obtained online. There is a lot of research on this topic for non-interactive tasks (Buhrmester et al. 2011; Paolacci et al. 2010; Snowberg and Yariv 2018; Stewart et al. 2017), and first comparisons for interactive designs look encouraging (Arechar et al. 2018). However, it remains to be established to what extent behavioural results of laboratory studies across the broad range of possible experimental designs are replicable with participants recruited online, and to what extent the methodological and conceptual differences lead to different results.

LIONESS Lab has a growing user base of experimenters who can access and build on each other's code through the repository. For technical aspects of the software and common coding issues, users can refer to the extensive online documentation. Furthermore, there is an active dedicated forum on which programming issues are discussed (https://lioness-lab.org/forum). Finally, there is an email address (info@lioness-lab.org) where users can ask for help. As developers of LIONESS Lab, we regularly update the software and its documentation, now and in the foreseeable future. This way, LIONESS Lab will accommodate recurrently emerging issues, incorporate solutions and new features (sometimes suggested by users), and respond to the changing needs of the world wide web. Should institutional support for LIONESS Lab unexpectedly cease, the developers of LIONESS Lab will make the full platform open-source, so that it can be deployed elsewhere. This information can also be found in the terms of use.

Acknowledgements We thank Antonio Alonso Aréchar, Benjamin Beranek, Urs Fischbacher, Johann Graf Lambsdorff, José Guinot Saporta, KyeongTae Lee, Katrin Schmelz, Nina Sedarevic, Shruti Surachita, Jungsun Yoo, and the members of the Ecology, Evolution and Biodiversity Conservation lab of the University of Leuven for useful comments and discussions. We also thank the users of the beta version of LIONESS Lab for suggestions for improvement and Susanna Grundmann for proof-reading our manuscript. We also thank the editor and two anonymous referees for their valuable comments. This work was supported by the European Research Council Grant ERC-AdG 295707—COOPERATION, the Economic and Social Research Council Network for Integrated Behavioural Sciences (ES/K002201/1) and by the Chair in Economic Theory, University of Passau, Germany. Lucas Molleman is supported by Open Research Area grant ASTA ID: 176.

Appendix

See Tables 1, 2.

Table 1 Comparison of LIONESS Lab to other software platforms

	LIONESS Lab	z-Tree (Fischbacher 2007)	oTree (Chen et al. 2016)
Platform independent	Yes	No only stationary PCs	Yes
Optimized for mobile use	Yes	No	Yes
Programming language for user	JavaScript	z-Tree specific	Python
Required programming knowledge	Basic	Basic	Advanced
Modular backend system	Yes	Yes	No
Number of participants	Automatic flexibility	Fixed at the start of the session	Flexibility programmable
Experiments with large groups (N > 100)	Yes	Yes	Unknown
Run on own server	Yes	Yes	Yes
Type of software	Free service	Free proprietary software	Open source

We focus on comparing our platform to z-Tree (Fischbacher 2007) and oTree (Chen et al 2016). For a more extensive overview comparing various web-based experimental software platforms, see Chan et al (2019). Users of LIONESS Lab require little programming skills. Experiments can be developed with a modular backend system to set up stages and elements. To develop simple experiments, users can start right away, without having to own a web server. While oTree is distributed as software, LIONESS runs as a service on dedicated servers. LIONESS experiments have been successfully run with hundreds of simultaneous participants recruited from Amazon MTurk (e.g. Molleman et al. 2019), we do not find evidence for oTree in the literature on large groups so far. In addition, LIONESS experiments can account for dynamically determined data quantity (after the start of the session), while oTree requires complex and deep changes to the data structure (Konrad 2018)

Table 2 Publications based on LIONESS experiments

Arechar, A. A., Gächter, S., & Molleman, L. (2018). Conducting interactive experiments online. *Experimental Economics*, *21*(1), 99–131

de Quidt, J., Fallucchi, F., Kölle, F., Nosenzo, D., & Quercia, S. (2017). Bonus versus penalty: How robust are the effects of contract framing? *Journal of the Economic Science Association*, *3*(2), 174–182

Fallucchi, F. & Nosenzo, D. (2020). The coordinating power of social norms. CeDEx Discussion Paper Series 2020-14

Fongoni, M, Hepp, J & Waltl SR. (2019) Social comparison, wage inequality and procedural fairness: an experimental investigation. https://www.aiel.it/cms/cms-files/submission/all20190615171533.pdf

Fornwagner, H., Pompeo, M., & Serdarevic, N. (2020). Him or her? Choosing competition on behalf of someone else. CeDEx Discussion Paper Series 2020-13

Gächter, S., Huang, L., & Sefton, M. (2016). Combining "real effort" with induced effort costs: the ball-catching task. *Experimental Economics*, *19*(4), 687–712

Glowacki, L., & Molleman, L. (2017). Subsistence styles shape human social learning strategies. *Nature Human Behaviour*, *1*, 0098

Li, X., Molleman, L., & Van Dolder, D (2020). Conditional punishment: descriptive norms drive negative reciprocity. CeDEx Discussion Paper Series 2020-05.

Molleman, L., Kanngiesser, P., & van den Bos, W. (2019). Social information use in adolescents: The impact of adults, peers and household composition. *PLoS ONE*, 14(11)

Molleman, L., Kurvers, R. H., & van den Bos, W. (2019). Unleashing the BEAST: a brief measure of human social information use. *Evolution and Human Behavior*, 40(5), 492–499

Molleman, L., Kölle, F., Starmer, C., & Gächter, S. (2019). People prefer coordinated punishment in cooperative interactions. *Nature Human Behaviour*, 3(11), 1145–1153

Molleman, L., Tump, A. N., Gradassi, A., Herzog, S., Jayles, B., Kurvers, R., & van den Bos, W. (2020). Strategies for integrating disparate social information. *PsyArXiv*, https://psyarxiv.com/wgzna

Molleman, L., & Gächter, S. (2018). Societal background influences social learning in cooperative decision making. *Evolution and Human Behavior*, 39(5), 547–555

Stagnaro, M. N., Arechar, A. A., & Rand, D. G. (2017). From good institutions to generous citizens: Top-down incentives to cooperate promote subsequent prosociality but not norm enforcement. *Cognition*, *167*, 212–254

Tonning, H, & Underhaug, M. (2019). In bots we (dis)trust. https://uis.brage.unit.no/uis-xmlui/bitstream/handle/11250/2618905/Underhaug_Martin.pdf

Weidinger, L., Gradassi, A., Molleman, L., & van den Bos, W (2019). Test–retest reliability of canonical reinforcement learning models. *Proceedings of the Annual Conference on Cognitive Computational Neuroscience*. https://ccneuro.org/2019/proceedings/0000513.pdf

Yahosseini, Reijula, Molleman, & Moussaid (2018); Yahosseini, K. S., Reijula, S., Molleman, L., & Moussaid, M. (2018). Social information can undermine individual performance in exploration–exploitation tasks. *Cognitive Science Conference Proceedings*, *17*. https://doi.org/10.31234/osf.io/upv8k

These experiments include experiments conducted online (e.g., de Quidt et al. 2017), in the lab (e.g., Gächter et al. 2016) and in the field (e.g., Glowacki and Molleman 2017). The above list was compiled on 11 June 2020. For an up-to-date list, see https://lioness-lab.org/publications

References

Arechar, A. A., Gächter, S., & Molleman, L. (2018). Conducting interactive experiments online. *Experimental Economics, 21*(1), 99–131.

Balietti, S. (2016). nodeGame: real-time, synchronous, online experiments in the browser. *Behavior Research Methods, 49*, 1–20.

Berinsky, A. J., Huber, G. A., & Lenz, G. S. (2012). Evaluating online labor markets for experimental research: Amazon.com's Mechanical Turk. *Political Analysis, 20*(3), 351–368.
Buhrmester, M., Kwang, T., & Gosling, S. D. (2011). Amazon's Mechanical Turk a new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science, 6*(1), 3–5.
Camerer, C. F., Dreber, A., Forsell, E., Ho, T.-H., Huber, J., Johannesson, M., et al. (2016). Evaluating replicability of laboratory experiments in economics. *Science, 351*(6280), 1433–1436.
Chan, S. W., Schilizzi, S., Iftekhar, M. S., & Rosa, R. D. S. (2019). Web-based experimental economics software: how do they compare to desirable features? *Journal of Behavioral and Experimental Finance, 23*, 138–160.
Chen, D. L., Schonger, M., & Wickens, C. (2016). oTree—An open-source platform for laboratory, online, and field experiments. *Journal of Behavioral and Experimental Finance, 9*, 88–97.
de Quidt, J., Fallucchi, F., Kölle, F., et al. (2017). Bonus versus penalty: How robust are the effects of contract framing? *Journal of the Economic Science Association*, *3*(2), 174–182. https://doi.org/10.1007/s40881-017-0039-9.
Egas, M., & Riedl, A. (2008). The economics of altruistic punishment and the maintenance of cooperation. *Proceedings of the Royal Society of London B: Biological Sciences, 275*(1637), 871–878.
Fischbacher, U. (2007). z-Tree: Zurich toolbox for ready-made economic experiments. *Experimental Economics, 10*(2), 171–178.
Gächter, S., Huang, L., & Sefton, M. (2016). Combining "real effort" with induced effort costs: the ball-catching task. *Experimental Economics*, *19*(4), 687–712.
Gallo, E., & Yan, C. (2015). The effects of reputational and social knowledge on cooperation. *Proceedings of the National Academy of Sciences, 112*(12), 3647–3652.
Giamattei, M., & Lambsdorff, J. G. (2019). classEx—an online tool for lab-in-the-field experiments with smartphones. *Journal of Behavioral and Experimental Finance, 22*, 223–231.
Hauser, D. J., & Schwarz, N. (2016). Attentive Turkers: MTurk participants perform better on online attention checks than do subject pool participants. *Behavior Research Methods, 48*(1), 400–407.
Hendriks, A. (2012). SoPHIE—software platform for human interaction experiments. University of Osnabrueck, Working Paper.
Henninger, F., Kieslich, P. J., & Hilbig, B. E. (2017). Psynteract: a flexible, cross-platform, open framework for interactive experiments. *Behavior research methods, 49*(5), 1605–1614.
Hergueux, J., & Jacquemet, N. (2015). Social preferences in the online laboratory: A randomized experiment. *Experimental Economics, 18*(2), 251–283. https://doi.org/10.1007/s10683-014-9400-5.
Horton, J. J., Rand, D. G., & Zeckhauser, R. J. (2011). The online laboratory: conducting experiments in a real labor market. *Experimental Economics, 14*, 399–425.
Mason, W., & Suri, S. (2012). Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods, 44*(1), 1–23.
Molleman, L., & Gächter, S. (2018). Societal background influences social learning in cooperative decision making. *Evolution and Human Behavior, 39*(5), 547–555.
Molleman, L., Kölle, F., Starmer, C., & Gächter, S. (2019). People prefer coordinated punishment in cooperative interactions. *Nature Human Behaviour*, 3(11), 1145–1153.
Molnar, A. (2019). SMARTRIQS: a simple method allowing real-time respondent interaction in qualtrics surveys. *Journal of Behavioral and Experimental Finance, 22*, 161–169.
Munafò, M. R., Nosek, B. A., Bishop, D. V., Button, K. S., Chambers, C. D., du Sert, N. P., et al. (2017). A manifesto for reproducible science. *Nature Human Behaviour, 1*, 0021.
Nishi, A., Shirado, H., Rand, D. G., & Christakis, N. A. (2015). Inequality and visibility of wealth in experimental social networks. *Nature, 526*(7573), 426–429.
Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science, 349*(6251), aac4716.
Paolacci, G., & Chandler, J. (2014). Inside the Turk: Understanding Mechanical Turk as a participant pool. *Current Directions in Psychological Science, 23*(3), 184–188.
Paolacci, G., Chandler, J., & Ipeirotis, P. G. (2010). Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making, 5*(5), 411–419.
Pettit, J., Friedman, D., Kephart, C., & Oprea, R. (2014). ConG, Software for continuous game experiments. *Experimental Economics, 17*(4), 631–648.
Schmelz, K., & Ziegelmeyer, A. (2015). Social distance and control aversion: Evidence from the Internet and the laboratory. Research paper series Thurgauer Wirtschaftsinstitut.
Snowberg, E., & Yariv, L. (2018). *Testing the Waters: Behavior across Participant Pools (No w24781)*. Cambridge, MA: National Bureau of Economic Research. https://doi.org/10.3386/w24781.

Stewart, N., Chandler, J., & Paolacci, G. (2017). Crowdsourcing samples in cognitive science. *Trends in Cognitive Sciences, 21*, 736–748.

Suri, S., & Watts, D. J. (2011). Cooperation and contagion in web-based, networked public goods experiments. *PLoS ONE, 6*(3), e16836.

Thomas, K. A., & Clifford, S. (2017). Validity and mechanical turk: An assessment of exclusion methods and interactive experiments. *Computers in Human Behavior, 77*, 184–197.

Wang, J., Suri, S., & Watts, D. J. (2012). Cooperation and assortativity with dynamic partner updating. *Proceedings of the National Academy of Sciences, 109*(36), 14363–14368.

Zhou, H., & Fishbach, A. (2016). The pitfall of experimenting on the web: How unattended selective attrition leads to surprising (yet false) research conclusions. *Journal of Personality and Social Psychology, 111*, 493–504.