

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

Marcus Giamattei^{1,2*}, Lucas Molleman^{2,3,4}, Kyanoush S. Yahosseini³ and Simon Gächter^{2,5,6}

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, United Kingdom

³ Center for Adaptive Rationality, Max Planck Institute for Human Development, Lentzeallee 94, 14195 Berlin, Germany

⁴ Department of Psychology, University of Amsterdam, Nieuwe Achtergracht 129b, 1018 WT Amsterdam, The Netherlands

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

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

*Correspondence: marcus.giamattei@uni-passau.de

Abstract

LIONESS Lab is a free web-based platform for interactive online experiments. It 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. The intuitive, user-friendly graphical interface enables researchers to develop, test and share experiments online, with minimal need for programming experience. 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: C90

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) and Prolific¹, allow researchers to conveniently recruit participants for experiments and compensate them for their efforts. The quality of data from online experiments is generally found to be comparable to data obtained in the laboratory (Arechar et al. 2018; Berinsky et al. 2012; Buhrmester et al. 2011; Hauser and Schwarz 2016; Hergueux and Jacquemet 2015; Mason and Suri 2012; Paolacci et al. 2010; Paolacci and Chandler 2014; Snowberg and Yariv 2018; Thomas and Clifford 2017; S. Wang et al. 2015), making online experimentation a promising complement to laboratory research. However, online experiments have typically been limited to non-interactive tasks that participants complete on their own, either using survey software (e.g., Survey Monkey, Qualtrics) to document decisions or emulating social interactions by using the strategy method (Selten 1965) 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; Rand et al. 2011; Schmelz and Ziegelmeyer 2015; Suri and Watts 2011; J. Wang et al. 2012), and to date, no widely used equivalent of zTree (Fischbacher 2007) is available for conducting interactive experiments.

A number of software platforms are currently available for conducting experiments via the internet, at varying stages of development². A prominent platform among these is oTree (Chen et al. 2016), which allows for browser-based interactions between experimental participants. Typically, the use of these platforms 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). The online use of interactive designs has thus been largely restricted to

¹ MTurk and Prolific can be accessed via <https://www.mturk.com> and <https://www.prolific.ac>, respectively.

² These software platforms include BreadBoard, ConG (Pettit et al. 2014), MobLab, NodeGame (Baliotti 2016), Psynteract (Henniger et al. 2017), SMARTRIQS (Molnar 2018) and UbiquityLab.

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 Live Interactive ONLINE Experimental Server Software. 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 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.

One of the most distinguishing features 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. It includes an intuitive user interface to design experimental screens 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). Experimenters can also decide to keep their experimental designs private and not to share them with others.

Testing LIONESS experiments is facilitated by a test mode including a ‘robot’ feature that simulates participant 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.

The rest of this paper is organised as follows. In Section 2, we outline the general structure of LIONESS Lab. In Sections 3 through 5, we discuss the use of the platform throughout the main phases of an interactive experiment. Concretely, in Section 3 we describe how LIONESS Lab facilitates easy development of a custom-made interactive experiment. Section 4 offers a brief description of available tools for testing LIONESS experiments before launching an experimental session. In Section 5, we describe how to conduct an online session with LIONESS Lab. We highlight key measures that LIONESS Lab takes to deal with common logistical and methodological challenges of online interactive experimentation, such as group formation and dealing with dropouts. Section 6 shows how LIONESS Lab allows researchers to share their designs with others through an integrated online repository, facilitating coordination among collaborators and replication of experimental results. Section 7 concludes.

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. Once a LIONESS experiment has been developed, it can be downloaded and used on a server³. Each experiment is a standalone program comprising a set of files

³ Having the experiment running on one’s own server gives the experimenter full control over their experimental data generated by participants, and allows researchers to adjust the server capacities

defining the experimental screens that participants will navigate during a session. The experimental files also include a number of automatically generated 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. Most importantly, these files include standardised methods for dealing with participant dropout, one of the most challenging aspects of interactive online experimentation (Arechar et al. 2018). These features have been thoroughly tested in a range of different experimental designs⁴ and are further discussed in Section 5. More details on the structure of LIONESS Lab can be found in the online documentation, accessible via <https://lioness-lab.org/documentation>. A demonstration of a LIONESS experiment can be found at <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 institutes. However, it can also be used in a laboratory (Arechar et al. 2018; Molleman and Gächter 2018), for lab-in-the-field experiments (e.g., Glowacki and Molleman 2017), or to conduct experiments involving non-interactive tasks such as surveys or questionnaires (e.g., de Quidt et al. 2017; Gächter et al. 2016). 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 them. Furthermore, it enables experimenters to complement online studies with data from the physical lab, for example, to test the

according to their needs. Guidelines are available in the online documentation (<https://lioness-lab.org/documentation>).

⁴ LIONESS experiments have been used in a number of published studies which were conducted by various research groups with a range of different interactive paradigms (Arechar et al. 2018; de Quidt et al. 2017; Gächter et al. 2016; Glowacki and Molleman 2017; Molleman and Gächter 2018; Stagnaro et al. 2017). LIONESS experiments are also being used in a range of ongoing research projects, including experiments with groups of up to 12 participants and over 1,000 participants per session.

robustness of their results in more highly controlled lab conditions (a request that can well occur during the refereeing process of papers conducted solely online).

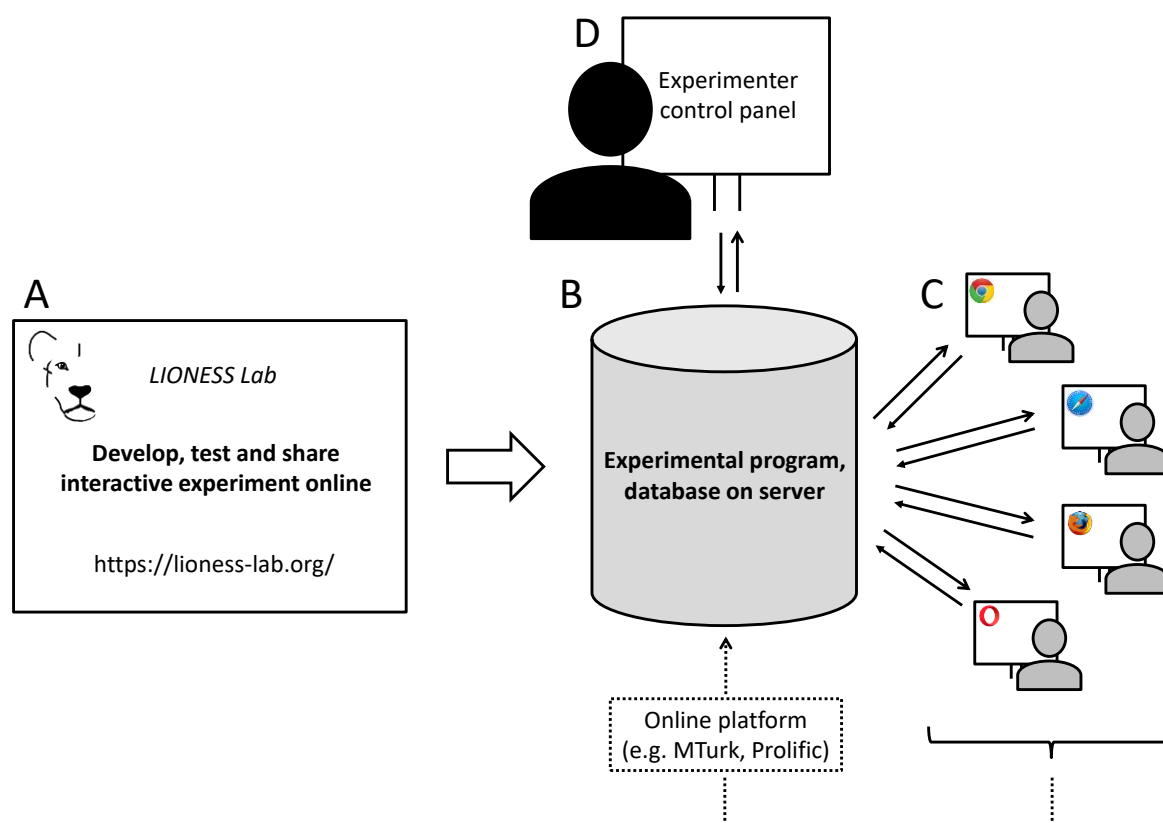


Figure 1. Basic structure of LIONESS Lab. The online user interface of LIONESS Lab facilitates easy development and testing of interactive experiments (A). LIONESS Lab produces LIONESS experiments (B), which contain the software and a database to conduct sessions with participants who can be recruited 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).

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 (Section 6).

LIONESS Lab aims at making the development of experiments as simple and intuitive as possible. To this end, the user interface (based on classEx, a widely used online tool for classroom experiments (Giamattei and Graf Lambsdorff 2015)) allows for developing

experiments by creating them stage by stage⁵. 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, an experimenter needs two elements: a 'numeric input' and a 'button' to submit the input and continue to the next stage. Experimenters select the numeric input element from the dropdown menu and add it to their screen. Inside the element, they specify an accompanying text that will be shown on the participants' screens. In the button element, the experimenter specifies which stage participants shall enter next, and adds a condition upon which participants can continue to that next stage (e.g., 'as soon as possible', or 'wait for others').⁶

⁵ The development of experiments by stages, and adding elements to those stages, largely follows the logic of zTree (Fischbacher 2007).

⁶ 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. Experimenters can choose to allow participants to go back by adding a 'back button' to a stage.

Figure 2. Defining a stage by adding elements to experimental screens. The top box is used for specifying basic features of the stage; experimenters can add a stage name or define 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 the next stage (‘results’ in this case) as soon as all group members have submitted their decision.

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 experiment⁷. 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.

⁷ Validation procedures 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). At the same time, these procedures protect the server and the data on the server from malicious response attempts, e.g., injecting code or using scripted responses.

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 Figure 3.⁸ Experimenters with more advanced programming skills can use the great flexibility of JavaScript to incorporate their own custom-made components into experimental stages.

```

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

```

Figure 3. An example of a JavaScript program. The programming reflects the logic of a 3-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.

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 of their group members have submitted 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.⁹

A typical experiment proceeds through four main phases (recruitment, session start-up, interactive decision making and payment), each of which may consist of one or more stages. Table 1 highlights key measures of LIONESS experiments to deal with the

⁸ JavaScript is the most widely used language for programming dynamic webpages. By default, it is supported by all modern web browsers. Due to its very large and active user base, solutions for programming issues are easily found.

⁹ Note that this setup again follows the logic of zTree (Fischbacher 2007).

challenges that occur in online interactive experiments in each of these phases. We discuss these measures further in Section 5.

4. Test experiments

At any point during the development of their LIONESS experiment, users can choose to compile and view the experiment. Upon compilation, LIONESS Lab generates a set of web pages that the experimental participants will navigate, as well as a number of scripts that regulate the flow of the experiment. This is a trial version of the experiment, which runs on the LIONESS Lab server and is directly available for testing. 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 global coordination; its functioning will be discussed in Section 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.¹⁰ This will yield a standalone program that can subsequently be uploaded to a private server. This 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 participants¹¹.

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 a typical session of an experiment conducted online throughout each of its four main phases: recruitment of participants, session start-up, interaction and payment. As we go along, we highlight how LIONESS experiments deal with the conceptual and methodological challenges of conducting interactive experiments online. An extensive methodological discussion of each of these phases can be found in Arechar et al. (2018). The contents of this section are summarised in Table 1.

Recruitment 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

¹⁰ 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>).

¹¹ For experimenters who do not have access to an own server, there are many user-friendly cloud services available to set up a private virtual server (e.g., on Google Cloud and Amazon Web Services). Getting a server ready for running a LIONESS experiment is described in a few simple steps in the online documentation (<https://lioness-lab.org/documentation>). The documentation also discusses basic pointers and rules of thumb for choosing server specifications depending on the details of the design and the prospective number of participants.

experiment. Once the experiment is 'activated' by clicking the button in the control panel, the link directing prospective participants to the experiment can be posted as a 'job' on a crowd-sourcing website.¹²

It is important for interactive experiments to make sure that a sufficient number of participants are recruited simultaneously, so that groups can be formed with minimal delay. Entrance rates into experiments do not only depend the availability of participants at the point in time in which the experiment is publicised, but also on the prospective earnings listed in the job posting. It is therefore often advisable to clearly communicate expected earnings and to be generous with payment rates (see Hara et al. (2018)).

When designing the job posting, crowd-sourcing websites typically allow users to set the number of participants they wish to recruit. This defines the maximum number of participants that will be paid at the end of a session. In contrast to typical sessions conducted in the laboratory, a fraction of participants may not finish the experiment and drop out along the way. Therefore, it is often useful for researchers to set the maximum number of participants on the crowd-sourcing website slightly higher than the amount of participants they want to recruit for their LIONESS experiment.¹³

Start-up phase

Duplicate participants can seriously compromise the internal validity of the experiment. For that reason, LIONESS experiments block participants who try to enter a session more

¹² 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 (cf. Figure 1). See Arechar et al. (2018), Horton and Chilton (2010); Kapelner and Chandler (2010) for discussions of effective ways to advertise a decision-making experiment on crowd-sourcing websites, and to minimize effects of selection into an experiment based on its features.

¹³ The number of additional 'slots' to be added to a LIONESS experiment strongly depends on the extent to which the experimental design causes dropouts. It is therefore hard to define a rule of thumb here. However, lengthy and complex instructions and strict compulsory comprehension questions are likely to lead to high dropout rates, especially when the expected payment for completing the experiment is relatively low.

than once.¹⁴ After a further check for browser support¹⁵, a participant can enter the experiment. Subsequently, participants are assigned a unique identifier, and the system automatically logs the time at which they entered.

After entering the experiment, participants will typically first read instructions about the task they are about to complete. Of course, the experimenter is not physically present while participants are completing an experiment, so that the experimenter is typically unable to answer any questions that may arise. To ensure that participants fully understand the instructions, LIONESS Lab allows for adding questionnaires (so-called 'quiz' stages). For these stages, LIONESS automatically documents the number of attempts that participants needed for solving each question. 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 the participants have read the instructions and successfully completed the questionnaire, they are ready to be matched. This matching takes place in a 'lobby' stage in which they wait for other participants with whom they will form a group. Experimenters can choose from various pre-set matching procedures (see below). 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 reaching the lobby stage. The default option is 'first-come, first-served', meaning that groups will be formed as soon as a sufficient number of participants are in the lobby.¹⁶ This procedure ensures that waiting times are minimized,

¹⁴ There are various third-party solutions present to prevent participants from entering different sessions (e.g., of the same study). One commonly used solution for MTurk is uniqueTurker (<https://uniqueturker.myleott.com>), which allows for adding a script to the HIT description which makes it invisible to MTurk workers who participated in earlier HITs that contained that script. This method may not work perfectly though, because a person might operate with different MTurk worker IDs at the same time. That said, MTurk acts against people having multiple worker accounts.

¹⁵ Most importantly, this procedure checks whether JavaScript is enabled, and prompts a participant to switch it on if disabled.

¹⁶ 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

which in turn drives down attrition due to inattention or random distractions while waiting for others to read the instructions and solve the control questions. Other available matching procedures allow experimenters to match participants based on treatments (such that, in experiments containing different treatments, participants from the same treatment are matched together) or based on roles (such that each group will consist of a certain number of whom has a unique role).

While participants are in the lobby, they are informed 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 the experiment and collect their participation fee. Clearly communicating information about expected waiting times helps keeping up participants' attention and driving down attrition.

Interaction phase

Once a group has been formed, participants are automatically directed to the first stage of a period. Typical experiments will have multiple periods in which participants make decisions and receive feedback. 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.¹⁷

In the control panel, experimenters can track the progress of participants during an experimental session. Figure 3 shows a screenshot of the control panel during an experiment. In this example, a 3-player public goods game (called 'PGGame') is being played and nine participants have entered the session. The top three rows in this table show a group which is currently in period 4 of the 10-period game. Two participants of this group have already submitted their choice and are in the waiting screen of the 'decision'

participants enter the experiment at different times: slow participants who entered early might be matched with fast participants who entered later.

¹⁷ 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.

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 3 can be formed.

PGGame - Control panel

address for participants: classex.uni-passau.de/classEx34/lioness/experiments/experiment75/game6333/_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 ***

Figure 4. The experimenter control panel. The experimenter can track the progress of a session and browse the tables of the database underlying the experiment. The tabs under the web link for participants represent the tables underlying the experiment. 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.

As noted above, one of the greatest challenges of conducting interactive experiments online is the possibility of participants dropping out. In some cases, dropouts cannot be prevented; when a participant loses their internet connection or leaves the experiment by closing their browser window, there is not much that any software can do. In other cases,

dropouts can be largely prevented and LIONESS has a set of measures in place to minimise participant dropout in those cases, and to deal with the logistics should dropouts occur.¹⁸

Waning attention is often a major source of dropouts: participants typically proceed through interactive experiments at the pace of the slowest member of their group. Long waiting times until the other group members have completed a stage can lead to inattention. LIONESS experiments have several measures in place to keep up attention of participants and to reduce the amount of time they have to wait for others in their group. These include the option to add timers to experimental screens. 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. This measure aims to reduce the time that participants wait for others. All waiting screens automatically display a spinning wheel as a 'waiting animation' so that participants can be that the experiment is still running. If the experimental screens are in the background of a participant's device (which can occur when participants are waiting for others and get distracted), an overlaying notification will be shown in case the experiment progresses to a new stage.

Despite these measures, dropouts cannot be avoided altogether. The controller algorithms of a LIONESS experiment detect 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 session¹⁹ 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). Letting participants continue in a group of reduced size can lead to increased costs for the experimenter, but it will avoid any potential damage to the online reputation of experimenters (which may occur when

¹⁸ For an extensive discussion of the implications of dropouts for the internal validity of study conclusions, see Zhou and Fishbach (2016).

¹⁹ In case they try to return to the experimental web pages they are led to a screen telling them that their session is over.

participants earn substantially less than they expected; they may complain about that on online forums²⁰). Alternative options to handle dropouts (e.g., terminating the interaction 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 but should be used with care. 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 dropouts can also be de-activated.

Payment phase

At the end of an experiment, participants typically receive a unique payment code, which they can enter on the crowd-sourcing platform. This verifies that a participant has actually completed the experiment, and can be paid their participation fee. In many designs, it is essential that participants are paid according to performance in the experiment. LIONESS experiments link the amount of such ‘bonus payments’ to the unique payment code which will be used to pay participants through the crowd-sourcing website.

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.²¹

²⁰ For MTurk, examples of such online forums are <https://turkopticon.ucsd.edu>, <https://www.reddit.com/r/mturk>, <https://www.reddit.com/r/HITsWorthTurkingFor>, <http://www.mturkforum.com/index.php>, <http://turkernation.com>

²¹ For MTurk, experimenters can simply copy the contents of the spreadsheet of the ‘batch’ used for the current LIONESS experiment. Pasting these contents into the tab ‘batchResults’ of the LIONESS output file will generate ready-made payment codes that can be entered into the Command Line Tool to pay bonuses. An extensive description of how to do this is in the online documentation.

6. Sharing experiments through a repository

Users of LIONESS Lab can choose to share their experiments with others through the repository. 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.²² 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 (cf. 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. Experiments can avoid reinventing the wheel and copy solutions to common issues in developing their designs, helping speed up the interaction between theory and empirics.

The screenshot shows a search results page for 'public goods' in the LIONESS Lab repository. The search bar at the top right contains the text 'public goods'. Below the search bar is a table with 10 columns: 'title', 'language', 'created by', 'created', 'institution', and 'account'. The table lists 10 entries, each with a plus sign in a grey box to its left. The entries are as follows:

	title	language	created by	created	institution	account
+	Public goods Antonio	EN	Luca Esposito	2016-06-07	University of Nottingham (United Kingdom)	luca.esposito
+	Public and private goods	EN	Shantanu Dutta	2017-03-12	University of Pennsylvania (United States)	shantanu.dutta
+	Public goods	EN	Melanie Lorenz	2016-05-11	University of Nottingham (United Kingdom)	lorenz
+	Public goods (copy)	EN	Thomas Wimmer	2018-08-13	University of Nottingham (United Kingdom)	thomas.wimmer
+	Public goods (copy)	EN	Heinz Heine	2018-04-22	Uni Köln (Germany)	heine
+	Public Goods Game	EN	Arjun Anand	2018-04-20	De La Salle University (Philippines)	anand
+	Public goods game with social information	EN	Luca Esposito	2018-02-16	University of Nottingham (United Kingdom)	luca.esposito
+	Public goods with punishment	EN	Geoff Hinton	2017-10-20	TU Clausthal (Germany)	hinton
+	test public goods	EN	John Galt	2017-08-25	Middlesex University (United Kingdom)	john.galt
+	test public goods (copy)	EN	Arjun Anand	2016-05-23	Yale University (United States)	anand

At the bottom of the table, it says 'Showing 1 to 10 of 19 entries (filtered from 308 total entries)'. On the right, there are navigation buttons for 'Previous', '1', '2', and 'Next'.

Figure 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

²² 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.

experimenters import an experiment to their own account (by clicking the '+' sign), it can be viewed, copied and edited.

7. Conclusion

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; Horton and Chilton 2010; 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.

Acknowledgements

The authors are grateful to 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. 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.

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*, 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.
- 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., 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. doi: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., & Graf Lambsdorff, J. (2015). *classEx: An online software for classroom experiments*. Passauer Diskussionspapiere: Volkswirtschaftliche Reihe.
- Glowacki, L., & Molleman, L. (2017). Subsistence styles shape human social learning strategies. *Nature Human Behaviour*, 1, 0098.
- Hara, K., Adams, A., Milland, K., Savage, S., Callison-Burch, C., & Bigham, J. P. (2018). A Data-Driven Analysis of Workers’ Earnings on Amazon Mechanical Turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 449:1–449:14). New York, NY, USA: ACM. doi:10.1145/3173574.3174023
- 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.
- Hergueux, J., & Jacquemet, N. (2015). Social preferences in the online laboratory: a randomized experiment. *Experimental Economics*, 18(2), 251–283. doi:10.1007/s10683-014-9400-5
- Horton, J. J., & Chilton, L. B. (2010). The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM Conference on Electronic Commerce* (pp. 209–218). ACM.

- Kapelner, A., & Chandler, D. (2010). Preventing satisficing in online surveys: A 'kapcha' to ensure higher quality data. In *The world's first conference on the future of distributed work* (CrowdConf2010).
- 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. doi:10.1016/j.evolhumbehav.2018.05.007
- Molnar, A. (2018). SMARTRIQS: A Simple Method Allowing Real-Time Respondent Interaction in Qualtrics Surveys. Retrieved from <https://smartriqs.com>. doi: 10.13140/RG.2.2.26386.09921
- 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). Software for continuous game experiments. *Experimental Economics*, *17*(4), 631–648.
- Rand, D. G., Arbesman, S., & Christakis, N. A. (2011). Dynamic social networks promote cooperation in experiments with humans. *Proceedings of the National Academy of Sciences*, *108*(48), 19193–19198.
- Schmelz, K., & Ziegelmeyer, A. (2015). Social distance and control aversion: Evidence from the Internet and the laboratory. *Thurgauer Wirtschaftsinstitut Research Paper Series*.

- Selten, R. (1965). Die Strategiemethode zur Erforschung des eingeschränkt rationalen Verhaltens im Rahmen eines Oligopolexperimentes. Seminar für Mathemat. Wirtschaftsforschung u. Ökonometrie.
- Snowberg, E., & Yariv, L. (2018). *Testing the Waters: Behavior across Participant Pools* (No. w24781). Cambridge, MA: National Bureau of Economic Research. doi:10.3386/w24781
- 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.
- Stewart, N., Chandler, J., & Paolacci, G. (2017). Crowdsourcing samples in cognitive science. *Trends in Cognitive Sciences* 21(10), 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.
- Wang, S., Huang, C.-R., Yao, Y., & Chan, A. (2015). Mechanical Turk-based experiment vs laboratory-based experiment: a case study on the comparison of semantic transparency rating data. In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation (pp. 53-62).
- 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(4), 493

Phase	Events / methodological challenge	LIONESS measure
<i>Recruitment:</i> participants enter the experiment through a web link		
	Exact entrance rates and attendance not known beforehand; Duplicate participants can compromise data	Participants register for the experiment through a web link Multiple participation blocked based on IP addresses (stored after one-way encryption). Maximum number of participants defined by user
<i>Session start-up:</i> participants read instructions, complete comprehension questions		
	Ensure comprehension of instruction in absence of experimenter	Compulsory comprehension questions; recording of number of attempts; optional exclusion of participants from further participation after X attempts
	Matching into groups	Lobby flexibly matches participants on-the-fly using pre-set procedures (e.g., first-come-first-serve, by treatment or by unique role);
<i>Interaction:</i> participants are matched into a group, make decisions and receive feedback		
	Inattention (especially when waiting for others)	Keep up pace and attention with optional timers; attempt 'focus stealing' when experiment not in foreground of participant's screen when a new stage begins
	Participants dropping out	Experimenter chooses from pre-set options for dropout handling (allowing remaining group members to proceed, or to direct them to a finish screen)
<i>Payment:</i> participants receive compensation by means of secure payment through crowd-sourcing website		
	Calculate and pay performance-specific bonus earnings	Generate random code for automated payment; export data as spreadsheet including tab generating payment codes (e.g., to upload to MTurk's Command Line Tools).

Table 1. The measures LIONESS takes to handle challenges of conducting a typical interactive experiment online. The table shows the four main phases of a typical experiment. Further methodological considerations for these four phases are extensively discussed in (Arechar et al. 2018).