Journal of the Economic Science Association (2019) 5:136–141
https://doi.org/10.1007/s40881-019-00065-1

ORIGINAL PAPER

μCap: connecting FaceReader™ to z-Tree

Leonard Doyle[1] · **David Schindler**[2]

Received: 3 December 2018 / Revised: 9 April 2019 / Accepted: 29 April 2019 / Published online: 9 May 2019
© The Author(s) 2019

Abstract

μCap is a software package (citeware) for economic experiments enabling experimenters to analyze emotional states of subjects using z-Tree and FaceReader™. μCap is able to create videos of subjects on client computers based on stimuli shown on screen and restrict recording material to relevant time frames. Another feature of μCap is the creation of time stamps in csv format at prespecified screens (or at prespecified points in time) during the experiment, measured on the client computer. The software makes it possible to import these markers into FaceReader™ easily. μCap is the first program that significantly simplifies the process of connecting z-Tree and FaceReader™ with the additional benefit of extremely high precision. This paper describes the usage, underlying principles as well as advantages and limitations of μCap. Furthermore, we give a brief outlook of how μCap can be beneficial in other contexts.

Keywords Experiment · Software · FaceReader™ · z-Tree

JEL Classification C90 · C91 · C99

This paper resulted from work during our joint time at MELESSA. Many thanks to the whole team for offering such a friendly and constructive environment. We thank René Cyranek and Martin Kocher for their encouragement and input. Additionally, we are grateful for the many valuable comments we received from Florentin Krämer, Konstantin Lucks, Simeon Schudy and Mark Westcott.

✉ David Schindler
d.schindler@uvt.nl

[1] Ludwig Maximilian University of Munich, Munich, Germany

[2] Department of Economics, Tilburg University, PO Box 90153, 5000 LE Tilburg, The Netherlands

Downloaded from https://www.cambridge.org/core. 07 Sep 2026 at 03:38:19, subject to the Cambridge Core terms of use.

1 Introduction

In economics, conducting laboratory experiments to determine causal relationships has become increasingly fashionable over the last few decades. A large number of laboratory experiments are nowadays conducted using computers as a result of the spread of computerized laboratories at major universities and research institutions. Urs Fischbacher's experimental software z-Tree (Fischbacher 2007) is very popular around the globe and is among the most-used experimental softwares, having been cited more than 8000 times as of 2019 according to Google Scholar.

z-Tree is versatile and yet easy to learn—starting with version 3.5.0—users can even record the current time on the client machine. Part of z-Tree's popularity is presumably due to its simplicity and the convenience of creating even complex experiments with only very little programming effort. The analysis of video footage created during these experiments, however, was relatively inconvenient so far, since the capability of z-Tree to interact with other software packages was limited until recently. For example, consider an experimenter interested in facial expressions of subjects after they observe a certain screen, for example a donation decision in a dictator game. To perform their analysis, experimenters so far needed to employ a software to record videos, create and export the time stamps (i.e., the times at which the events of interest occur) into z-Tree (using table dumpers)[1], and manually match time stamps and video footage in the program of their choice. It is apparent that these and similar methods might be very tedious to the experimenter, since they rely on manually repeating a large number of tasks. This might be especially burdensome if experimenters have to deal with a large number of subjects and sessions. One of these situations for example concerns the analysis of emotions using FaceReader™.

FaceReader™ is a software package developed by the Dutch company Noldus (http://www.noldus.com), it analyzes facial expressions from photographs and videos with respect to six basic emotional states. The software lays a grid of more than 500 key points over images of each participant's face and identifies emotions by distinct muscle movement that is associated with a change in emotions. According to Noldus, the software performs equally well as trained annotators in describing emotional states of subjects. Researchers in economics have recently started to use FaceReader™ to investigate how emotional states correlate with economic decision making. For example, Noussair and Nguyen (2014) investigate the role of emotions in decision-making under risk, and Breaban and Noussair (2017) look at emotions in the context of asset markets. Furthermore, van Leeuwen et al. (2017) investigate the relationship of anger and rejections in ultimatum games, and Breaban et al. (2016) describe a positive correlation between more negative states and greater prudence.

μCap is the first tool to allow experimenters to create a fully automated connection between z-Tree and FaceReader™ in a straightforward way. It is barely noticeable by subjects and imprecisions are kept very limited with only a few milliseconds

[1] As already said, this is only possible since version 3.5.0. If the experimenter was to use earlier versions, additional workarounds would have to be used (e.g., switching lights on and off to vary brightness).

Downloaded from https://www.cambridge.org/core. 07 Sep 2026 at 03:38:19, subject to the Cambridge Core terms of use.

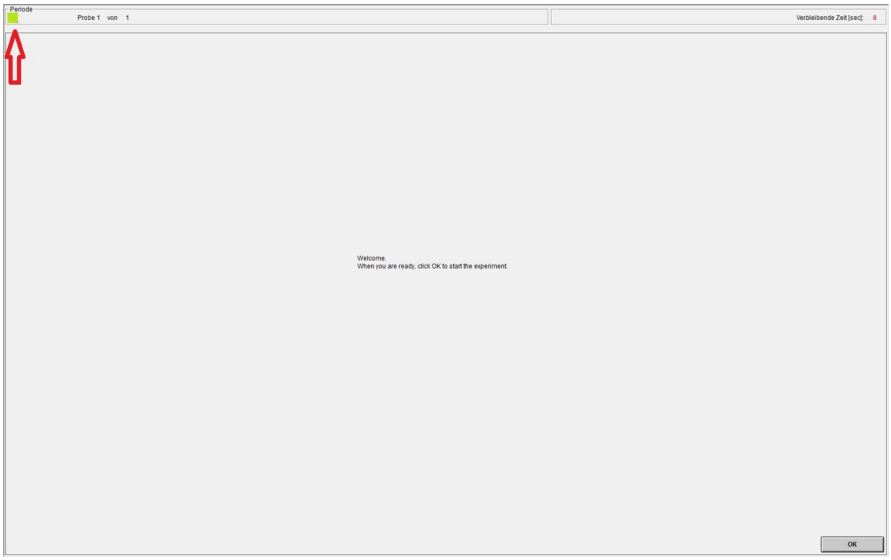

Fig. 1 Participant's screen when using μCap, red arrow added to highlight the colored rectangle (color figure online)

delay. As we will point out later, μCap can easily be implemented to improve experimenters' work flow and has possible applications beyond its original purpose. μCap has found first applications, such as Kugler et al. (2018), who use the tool to synchronize decisions from a trust game programmed in z-Tree with videos of participants' facial expressions.

2 μCap

μCap is a bundled software package and comes with two additional separate applications: μConfig and μProject. The governing principle of μCap is based on repeatedly reading a specific pixel from the top left corner of the participants' screen (for an example screen, see Fig. 1). Whenever this pixel changes color, μCap will create a time stamp in a csv file.[2] By doing so, μCap handles each participant separately, such that the progress of participants in different stages of the experiment will still be recorded precisely. This would matter, for example, if participants in a dictator game arrive at the donation decision screen at different points in time, for example, due to a preceding questionnaire. This procedure requires a minimum of additional programming effort by the experimenter. On the one hand, the tool μConfig renders the creation of hand-written configuration files (i.e., files that define the association of colors and time stamps) unnecessary, as it offers experimenters a graphical interface to define colors μCap will react to and to attach labels to events. On the other

[2] csv is short for comma-separated values, a simple file format to store tabular data.

Downloaded from https://www.cambridge.org/core. 07 Sep 2026 at 03:38:19, subject to the Cambridge Core terms of use.

hand, additional programming in z-Tree is limited to adding a small box of 20 × 20 pixels in the top left corner of the screen.[3]

In a typical experiment, experimenters will use μConfig when they program their experiment in z-Tree. A set of colors (using the RGB scheme) has to be defined in μConfig and rectangles with the corresponding colors have to be placed in the top left corner of the z-tree screen. Virtually unlimited numbers of markers (i.e., color event combinations) can be defined to distinguish all parts of interest within the experiment. If the same part occurs more than once, markers can be re-used. Note that using the RGB scheme also enables users to only marginally vary the color of the rectangle (e.g., from 250, 250, 250 to 252, 250, 250) in a way such that μCap recognizes the difference, but the participant does not. This might be especially valuable in a setting where the experimenter does not want the participant to know (or guess) what the instances of interest are.

After μConfig has been set up correctly, μCap will not only create video files of participants (including starting and stopping the recording at prespecified color changes)[4], but also correctly create the necessary time stamps. For μCap to function properly, it simply needs to be executed on all clients (just like z-Leaf, the client program of z-Tree). After the start of the experiment, μCap will be executed in the background, not visible to subjects.

As soon as video files and time stamps have been collected from the client computers[5], μProject will help to automatically create a new project in FaceReader™ (i.e., a file that will allow to batch process the analysis of facial expressions from the video material). Not only will this project automatically create the total number of subjects, but also load the video files and time stamps. Especially in projects with a large number of subjects, the automated processing of data using μProject can substantially reduce experimenters' workload.

In a typical experiment, and to generate the desired output, an experimenter would typically follow these steps:[6] First, when programming the experiment, ensure that all screens in z-Tree include the small rectangular box with the desired RGB color code. Second, using μConfig, generate the configuration csv file defining the markers for these color codes. Third, store the csv in the same folder as μCap and make sure that all client computers can access this folder. Fourth, when running the experiment, make sure that all clients run μCap and z-Leaf to generate the marker and video file for each subject. Fifth, collect all files and load them in μProjectCreator, which in turn creates an frx file to be loaded in FaceReader™. Sixth, let FaceReader™ execute the frx file and save the resulting data file that indicates

[3] Although μCap only monitors one specific pixel, different screen types and resolutions might make it necessary to increase the monitored area to a rectangle of size 20 × 20 pixels. This takes up only a small part of the entire screen but still very likely includes the pixel of interest. We recommend every lab to pre-test the optimal rectangle size and position before first use.

[4] An obvious prerequisite for the recording feature is of course that a video camera was installed on the respective computer. μCap was designed such that every standard web cam will do and no specific camera type or drivers will be required.

[5] At the University of Munich we created a tool (MELESSA Video Collector) for automating the process, it is however specifically tailored to our IT infrastructure. If you are interested in the tool, please get in touch.

[6] A more detailed description can be found in the user's manual accompanying the software package.

Downloaded from https://www.cambridge.org/core. 07 Sep 2026 at 03:38:19, subject to the Cambridge Core terms of use.

the strength of each emotion at every point in time. μCap puts an additional column of data in the file that indicates the event markers.

μCap can be used free of charge. We however kindly ask you to cite this paper in any academic publication or presentation when μCap has been used (citeware). The most recent version of the package and documentation can be downloaded from http://mucap.david-schindler.de.

3 Advantages, limitations, and possible extensions

The biggest advantage of using μCap for experimenters is the reduction of performing time-consuming manual tasks to link video footage and experimental data. This can prove extremely valuable in settings with a large number of subjects. While other options require repeating many small steps, μCap offers a worry-free solution that automatizes many of the necessary tasks and enhances workflow. Another big advantage of μCap is the ability to only record parts of interest. In a long experiment consisting of several parts, maybe only the facial expressions in one part, for example the donation decision in a dictator game, are of interest to the experimenter. Since FaceReader™ needs computational power to process the videos files, useless footage will drastically increase the time FaceReader™ will need for completing the analysis. In our experience, an average workstation needs about 3 min to process a 1-min video file. Many experiments nowadays consist of more than 100 participants and last up to 2 h. If a researcher was interested in only 15 min of the recordings, the use of μCap could reduce computation time drastically from 600 h to only 75 h.

An important issue for experimenters is probably the imprecision resulting from the use of μCap. Although we have not taken the effort to measure the exact latency μCap induces, we are very confident that the incongruence of timestamp and video frame is less than a frame (i.e., less than 40 milliseconds).[7] Apart from many important advantages, using μCap may have some downsides, which we would like users to be aware of. Although we tried to keep additional programming very limited, the use of μCap will always mean additional work to the experimenter. We think that our solution is very efficient in requiring as little effort as possible, but still, colored rectangles will have to be implemented in z-Tree and the system needs to run both, z-Tree and μCap, which might be more burdensome than using z-Tree alone. On the μCap website, we provide a template treatment that can help to reduce programming effort. Additionally, μCap is a program written for users of Microsoft Windows. We do not see ourselves able to offer the package for any other operating system, but μCap is compatible with versions of Windows XP up to Windows 10. Given that z-Tree requires Windows as an operating system as well, we deem this constraint not binding. Support on other operating systems is probably possible using emulations.

[7] The screen sensor monitors the pixel of interest with 50–100 frames per seconds which induces a latency of 7–20 milliseconds. One may add a few milliseconds for Microsoft Windows-specific latencies. We understand that there exist settings where this latency is unacceptable (and in those cases μCap should not be used), we, however, deem it small enough most applications involving the use of z-Tree. We also recommend to test the actual latency in your IT environment before using the tool, as the delay may depend on features of your infrastructure.

Downloaded from https://www.cambridge.org/core. 07 Sep 2026 at 03:38:19, subject to the Cambridge Core terms of use.

While μCap was developed to work with FaceReader™, it is potentially interesting to use the tool in other environments or with different software packages, where timestamps in csv format are needed. In addition to the presented application, one could think of linking experimental data to a variety of physiological measures, like eye-tracking, skin conductance and the like. Since most of those physiological measures are optimized to analyze individual behavior, μCap can offer a simple extension for the analysis of interactive behavior, especially in large groups. Also, the principle of placing colored rectangles on screen can be implemented in basically any experimental software that supports such a customization and therefore does not only limit μCap to be used with z-Tree.

The principle of changing colors to create precise timestamps has since been picked up by Perakakis et al. (2019), who suggest the use of external photo sensors to feed changes in luminosity directly to external devices. While their approach is equally promising in terms of effectiveness, it requires setup costs: monetary (purchasing photo sensors) and non-monetary (installing and maintaining such a system). Most importantly, however, it does not offer the degree of automation that μCap offers for the specific combination of connecting FaceReader™ and z-Tree.

4 Conclusion

In this paper we have introduced μCap, a software package for experimental economists to link FaceReader™ and z-Tree. We have explained the governing principle of μCap and discussed its advantages and weaknesses. Furthermore, we have suggested possible extensions. We heavily rely on users' feedback to improve and enhance the program and therefore ask all users to contribute by sending in ideas.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

Breaban, A., & Noussair, C. N. (2017). Emotional state and market behavior. *Review of Finance*, *22*(1), 279–309.

Breaban, A., van de Kuilen, G., & Noussair, C. N. (2016). Prudence, emotional state, personality, and cognitive ability. *Frontiers in Psychology*, *7*, 1688.

Fischbacher, U. (2007). z-Tree: Zurich toolbox for ready-made economic experiments. *Experimental Economics*, *10*(2), 171–178.

Kugler, T., Ye, B., Motro, D., & Noussair, C. N. (2018). On trust and disgust: Evidence from face reading and virtual reality. Working paper.

Noussair, C. N., & Nguyen, Y. (2014). Risk aversion and emotions. *Pacific Economic Review*, *19*(3), 296–312.

Perakakis, P., Guinot-Saporta, J., Jaber-Lopez, T., García-Gallego, A., & Georgantzis, N. (2019). A technical note on the precise timing of behavioral events in economic experiments. *Journal of Behavioral and Experimental Finance*, *21*, 10–14.

van Leeuwen, B., Noussair, C. N., Offerman, T., Suetens, S., van Veelen, M., & van de Ven, J. (2017). Predictably angry–facial cues provide a credible signal of destructive behavior. *Management Science*, *64*(7), 3352–3364.

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

Downloaded from https://www.cambridge.org/core. 07 Sep 2026 at 03:38:19, subject to the Cambridge Core terms of use.