Journal of Behavioral and Experimental Finance 37 (2023) 100762

Contents lists available at ScienceDirect

Journal of Behavioral and Experimental Finance

journal homepage: www.elsevier.com/locate/jbef

Full length article

Zurich Trading Simulator (ZTS) — A dynamic trading experimental tool for oTree☆

Sandra Andraszewicz [a,b,*], Jason Friedman [a,**], Dániel Kaszás [a], Christoph Hölscher [a,b]

[a] Chair of Cognitive Science, ETH Zurich, Clausiusstrasse 59, 8092 Zürich, Switzerland
[b] Singapore-ETH Centre, Future Resilient Systems, CREATE campus, 1 CREATE Way, #06-01 CREATE Tower, Singapore 138602, Singapore

ARTICLE INFO

Article history:
Received 7 July 2022
Received in revised form 11 October 2022
Accepted 16 November 2022
Available online 1 December 2022

Keywords:
oTree
Dynamic trading software
Experimental finance app
Trading behaviour

ABSTRACT

Recent literature on the intersection of economics, finance and psychology indicates the benefits of simulated experience in tools measuring decision making. Here, we present the Zurich Trading Simulator (ZTS) first used by Andraszewicz et al. (2022) to test the impact of upward social comparison on trading activity. ZTS is a free application for oTree designed to create dynamic investment experiments suitable for measuring trading activity and risk taking. The software is ready-to-use applying the default settings. It can also be freely adapted in the source code depending on the experimenter's needs. Price paths developed for experiments with the ZTS are freely available online. We also outline recommendations and possibilities for future studies and extensions.

1. Introduction

Stock markets can be conceived as dynamically evolving complex social systems driven by human decisions, emotions and interaction among the stock market participants. "Luck, uncertainty, and surprise are the most fundamental physical forces in the world of investing" (Zweig, 2015, p. xi). Investing in a stock market is a dynamic decision problem where a series of decisions is required to reach the goal (gain) and the consequent decisions depend on the previous decisions creating an evolving decision environment (Brehmer, 1992). Therefore, investment decisions are characterised by high complexity, which cannot be fully captured by static experimental tasks. While field studies have high external validity, they may offer too little control to accurately measure the relationships between variables influencing dynamic decisions (Brehmer and Dörner, 1993). Developments of computers and computing power since the 1990s enabled the design of so-called microworlds, which bridge the gap between controlled laboratory experiments and the realism of complex and dynamic decision problems. Microworlds are computerised experimental tasks which simulate real-life decision situations in changing environments (Edwards, 1962; Rapoport, 1975). They allow an experimenter to systematically control or vary some aspects of the decision environment and to interpret the casualty among measured variables (Brehmer and Dörner, 1993). This type of tasks have been used to investigate various applied decision problems (see Gonzalez et al., 2005, for a review).

However, dynamic experimental investment tasks measuring individual decision making under uncertainty are scarce. Murphy et al. (2016) developed an investment task, in which the uncertainty about the outcome evolves over time, but the consecutive decisions are independent. In contrast, Lejarraga et al. (2016) implemented a sequential decision making task with dependent decisions, to measure investment risk taking during market crashes and booms. They presented information about a risky asset as chunks of historical prices of the Spanish stock market index (IBEX). After viewing each chunk, participants would decide on the percentage of their available cash that they want to invest in the risky asset. All participants completed the same number of trials. The presented historical prices were displayed on static price charts. In a similar fashion, Grosshans and Zeisberger (2018) presented price paths leading to the same return but with different pattern (up–down vs. down–up) in a two-stage process. They found that the pattern of the price path impacts investors' financial decisions and their satisfaction from these decisions despite leading to the same returns. Along the similar lines, Borsboom and Zeisberger (2019) demonstrated that price paths with the same historical volatility but different shapes result in a different perceived riskiness. This research shows that the trajectory of historical prices determines people's perceived riskiness of a stock and their satisfaction from investing

☆ This software is free of charge if you cite this paper. The application is available at https://github.com/Zurich-Trading-Simulator/OtreeZTS. Price patterns and other experimental materials are available at the Open Science Framework at https://osf.io/we48d/.

* Corresponding author at: Chair of Cognitive Science, ETH Zurich, Clausiusstrasse 59, 8092 Zürich, Switzerland.

** Corresponding author.

E-mail addresses: sandraszewicz@ethz.ch (S. Andraszewicz), j.friedman@mail.ch (J. Friedman).

https://doi.org/10.1016/j.jbef.2022.100762
2214-6350/

in the stock. Also, Huber et al. (2022) documented that investment behaviour of professionals is negatively related to price changes, while student traders' perception of stock riskiness is most sensitive to the frequency of negative returns.

Bradbury et al. (2015) asked their participants to choose a financial product to invest in after sampling potential returns. In the sampling procedure, the authors randomly drew a number of static price paths and their corresponding returns from a pre-defined pool. This sampling procedure is a form of a simulated experience (i.e., simulating the potential price paths and returns coming from the same financial asset). Bradbury et al. (2015) showed that simulated experience improves people's understanding of risk of financial products. They aimed at closing the description-experience gap (Hertwig and Erev, 2009) between making choice investments from description and by experiencing the potential consequences of these decisions. Peters and Slovic (2000) explains the description-experience gap in risky decision-making by the fact that personal experience of adverse consequences of risky decisions should have a stronger effect on people's consequent decisions than mental simulation of these consequences. Several studies showed that simulated experience leads to more accurate probability judgements (Hogarth et al., 2015) and statistical information communication (Hogarth and Soyer, 2015). Altogether, these results lead to the conclusion that "experience has a crucial role in learning and forming judgements" (Hogarth and Soyer, 2015, p.1801). Therefore, static experimental tasks depart from the complexity of real dynamic systems such as stock markets.

In contrast, the seminal work by Smith et al. (1988) and its various extensions (see Palan, 2013, for a review) create a dynamically evolving market. However, these tasks are designed to measure the collective market behaviour rather than an individual trading behaviour. Experimental software which provides a simulated trading experience is limited. DiFonzo and Bordia (1997) developed a sequential investment task to investigate the impact of rumours on trading strategies. They displayed historical market prices of 60 ticks, each tick presented every 20 s. They found that the trading strategy of participants experiencing rumours noticeably departed from the most optimal buy-low-sell-high strategy.

In this paper, we present an experimental tool – the Zurich Trading Simulator (ZTS) – that provides the simulated experience in a form of a dynamically evolving market for individual investment decisions. ZTS is a one-to-one copy of the design developed by Andraszewicz et al. (2022), further programmed as an oTree app for the purpose of sharing it with the scientific community. ZTS differs from the previous designs investigating the impact of price patterns on risk taking and risk perception (i.e, Borsboom and Zeisberger, 2019; Bradbury et al., 2015; Grosshans and Zeisberger, 2018; Huber et al., 2022; Lejarraga et al., 2016) by including a dynamic chart compared to static (i.e., a picture) graphical price display. In addition, in ZTS, one's earlier investment decisions determine the decision environment of the subsequent decisions.

ZTS also differs from the dynamic experimental asset markets in a market setting that are designed to measure interaction among players rather than to measure trading activity of an individual. ZTS is not meant for studying interaction among market participants. It is designed to investigate an individual's reaction to market and trading conditions. ZTS participants are the market takers, which mimics actual situations of most individual investors in real markets. Therefore, an experimenter can use ZTS to measure the impact of various market features on an individual investor rather than on the market as a whole.

ZTS software is implemented as a free app for oTree, for investigating risk taking and trading activity in a dynamically evolving environment. In contrast to the other experimental tasks, ZTS allows the participants to experience asset price movement "live" in a dynamic setting. This setting resembles the real-world trading environment which is continuously developed over time rather than presented in chunks. Thanks to its dynamic setup, ZTS allows for measuring trading activity (i.e., volume, number of trades, value of the traded assets, etc.). Additionally, ZTS measures risk taking as a proportion of a risky asset in portfolio. ZTS is open-source allowing experimenters adapt the standard setup to meet the needs of their experiment.

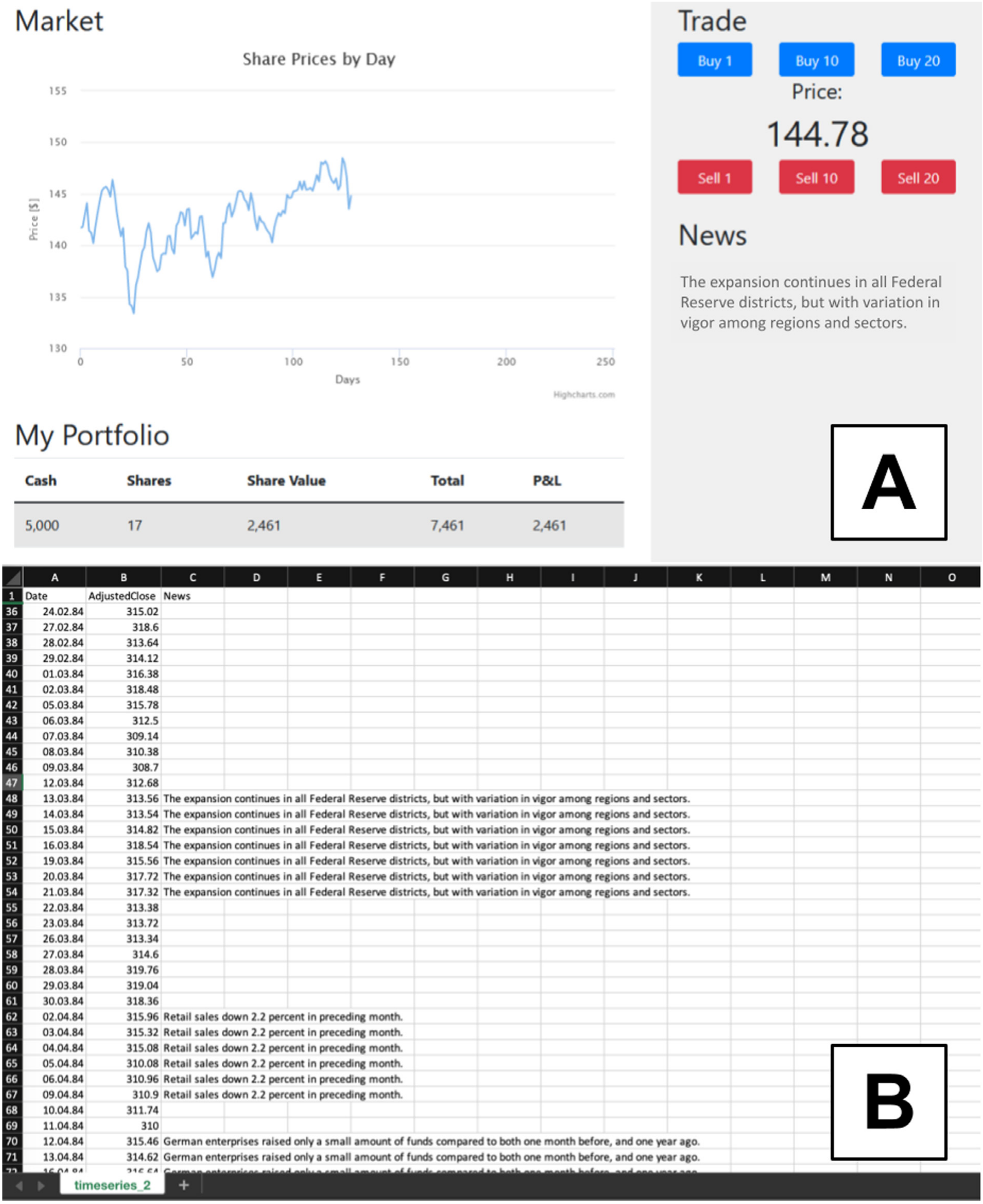
2. Other software for investigating trading and investment decisions

Research domains such as experimental finance, experimental economics and cognitive psychology have greatly benefited from standardised and amendable experimental software for efficiency reasons because programming and testing can be very costly. In order to share experimental methods, platforms such as z-Tree (Fischbacher, 2007), oTree (Chen et al., 2016) and PsychoPy (Peirce, 2007) were developed to constitute general backbones for online experiments conducted in the laboratory and in the field. Researchers build libraries and apps working on these standard platforms to increase the accessibility of various experimental designs within the scientific community. These apps provide an important contribution to dissemination of experimental research in a number of ways. First, they enhance data handling by increasing data security (Jiang and Li, 2019) and improving text processing (Saral and Schröter, 2019). Second, some libraries and applications link specialised proprietary software for measuring eye-tracking (Niehorster and Nyström, 2020) and analysing emotional states based on facial expressions (Doyle and Schindler, 2019) with the standard platforms such as oTree and PsychoPy. While some libraries provide a toolbox for creating standardised experimental stimuli (see Peirce, 2009), others offer standard experimental paradigms. For example, Crosetto et al. (2019) offers software for measuring social value orientation. Holzmeister (2017) and Holzmeister and Pfurtscheller (2016) created a tool for eliciting risk preference. Pichl (2019) developed an app providing a complete package of standardised economic games. Aldrich et al. (2020) created a framework for multiplayer artificial financial markets.

Apart from making use of the standard experimental methods and tools easier, researchers can use platforms such as oTree to share their newly developed and tested experimental designs and procedures. For example Huber (2019) created an adaptable ready-to-use app based on the methodology developed by Moinas and Pouget (2013). Along the same lines, Holzmeister and Kerschbamer (2019) implemented a novel Equality Equivalence Test (Kerschbamer, 2015), while von Bülow and Liu (2020) shared their complete experiment for investigating climate change adaptation behaviour. Following these standards, we converted the experimental design used by Andraszewicz et al. (2022) to an oTree app. Here, we document the technical features of the ZTS software, provide information about the potential uses and application of ZTS and outline all necessary information for setting up and customising an experiment.

3. ZTS software and experimental design features

The ZTS is a web-based trading and investment task. It consists of several trading rounds, where the number of the rounds can be freely chosen by the experimenter. In each round, a participant is presented with a simple trading user interface dynamically displaying the price of one asset (see Fig. 1 and a movie showing the experimenter and the participant view of the software https://youtu.be/lxGa6PWYmak). A running demo version is available



News

The expansion continues in all Federal Reserve districts, but with variation in vigor among regions and sectors.

My Portfolio

Cash	Shares	Share Value	Total	P&L
5,000	17	2,461	7,461	2,461

	A	B	C
1	Date	AdjustedClose	News
36	24.02.84	315.02	
37	27.02.84	318.6	
38	28.02.84	313.64	
39	29.02.84	314.12	
40	01.03.84	316.38	
41	02.03.84	318.48	
42	05.03.84	315.78	
43	06.03.84	312.5	
44	07.03.84	309.14	
45	08.03.84	310.38	
46	09.03.84	308.7	
47	12.03.84	312.68	
48	13.03.84	313.56	The expansion continues in all Federal Reserve districts, but with variation in vigor among regions and sectors.
49	14.03.84	313.54	The expansion continues in all Federal Reserve districts, but with variation in vigor among regions and sectors.
50	15.03.84	314.82	The expansion continues in all Federal Reserve districts, but with variation in vigor among regions and sectors.
51	16.03.84	318.54	The expansion continues in all Federal Reserve districts, but with variation in vigor among regions and sectors.
52	19.03.84	315.56	The expansion continues in all Federal Reserve districts, but with variation in vigor among regions and sectors.
53	20.03.84	317.72	The expansion continues in all Federal Reserve districts, but with variation in vigor among regions and sectors.
54	21.03.84	317.32	The expansion continues in all Federal Reserve districts, but with variation in vigor among regions and sectors.
55	22.03.84	313.38	
56	23.03.84	313.72	
57	26.03.84	313.34	
58	27.03.84	314.6	
59	28.03.84	319.76	
60	29.03.84	319.04	
61	30.03.84	318.36	
62	02.04.84	315.96	Retail sales down 2.2 percent in preceding month.
63	03.04.84	315.32	Retail sales down 2.2 percent in preceding month.
64	04.04.84	315.08	Retail sales down 2.2 percent in preceding month.
65	05.04.84	310.08	Retail sales down 2.2 percent in preceding month.
66	06.04.84	310.96	Retail sales down 2.2 percent in preceding month.
67	09.04.84	310.9	Retail sales down 2.2 percent in preceding month.
68	10.04.84	311.74	
69	11.04.84	310	
70	12.04.84	315.46	German enterprises raised only a small amount of funds compared to both one month before, and one year ago.
71	13.04.84	314.62	German enterprises raised only a small amount of funds compared to both one month before, and one year ago.

Fig. 1. A: User interface of the ZTS software; B: CSV-file used as stimulus to display price of a risky asset and news. The news is displayed for *n* experimental days.

at https://zts.otree.ethz.ch/ which demonstrates the application's functionality.

Participants can use six quick-trade buttons (three for buying and three for selling) to quickly execute small, medium, and large trades. They can monitor their portfolio and they can read news displayed in the news box . Each new price data point (i.e., tick) is displayed every one second using the standard set-up. The speed of displaying price information can be edited in the source code, depending on the experimenter's requirements (see Appendix C). When displaying only price information, Andraszewicz et al. (2022) used the refresh rate of 0.8 s, while experiments which presented news, apart from the price information, used the refresh rate of 1.4 s (see Section 5). In the ZTS user interface, the ticks are referred to as "experimental days". The exact price information is displayed numerically between the quick-trade buttons, while the price chart labelled "Market" displays a dynamically evolving price pattern. Every experimental day *i*, the vertical axis of the price chart is adjusted automatically, such that the upper boundary of the vertical axis set to $Price_{t=0} + MaxDist_{t=i}$, lower boundary of the vertical axis set to $Price_{t=0} - MaxDist_{t=i}$, where $MaxDist_{t=i} = max_{j<i}(1.05 \cdot |Price_{t=j} - Price_{t=0}|)$. The boundaries are rounded by the chart library to the nearest 100 or 10 (i.e. 2039 would be rounded to 2000 in case of the lower limit, or to 2100 in case of the upper limit).

The initial endowment set by the experimenter for the whole experiment is equally split between cash and the risky asset.[1] In Andraszewicz et al. (2022), the values of medium and large quick-trade buttons automatically adapt to the asset price, prior to each round, such that the value is divisible by 10. The values of the middle quick-trade buttons correspond to the rounded 10% of the maximum shares that could be bought with the cash endowed at the beginning of each round. The values of the large (i.e., rightmost) quick-trade buttons correspond to rounded 20% of the maximum shares that could be bought with the cash at the start of the game. In the ZTS application linked to this paper, the values of the quick-trade buttons can be selected in the session configuration.

The "News" box displays the news text information aligned with specific price data. The duration of the display of the news is measured in experimental days. For example, if an experimenter aims to display the news for eight experimental days, the news should be included in eight consecutive rows of the CSV-file with experimental stimulus (see panel B of Fig. 1 and Appendix B). One piece of news cannot be longer than 500 characters. Longer texts are automatically cropped. The font size of the news is 16px.

In the panel labelled "My Portfolio", a participant can monitor the performance of their portfolio. The information in "My Portfolio" is updated at the same speed as the price information and when the participant executes a trade. The portfolio information includes the cash value (i.e., Cash), number of shares of the risky asset (i.e, Shares), value of these shares calculated by multiplying the share price by number of shares (i.e., Share Value), the total portfolio value including the cash value and the share value (i.e., Total), and the profit and loss of the portfolio wrt. the initial endowment. All values are expressed in the experimental currency units. The endowment is the same in all rounds. After each trading round, the portfolio is reset.

Apart from the trading app, the ZTS contains a survey app. The survey part consists of a single page with a "Proceed to Survey" button that redirects the players to a survey under the link specified in survey_link variable from the session configurations. The survey can also be directly implemented in oTree or in another commercial or non-commercial survey software. In oTree an, app is an independent module of the entire experiment. Therefore, ZTS consists of two oTree-apps. When a participant starts a session (i.e., one batch of data collection), they are redirected to the instructions page. By default, the instructions are empty and can be edited by the experimenter. The instructions page is displayed only before the first trading round.

4. Output data

The ZTS app outputs several specific data fields, apart from the standard oTree data.[2] For each experimental day, the following data fields are logged:

- *cur_day* — the experimental day
- *cash* — the amount of cash that a participant holds
- *shares* — the number of shares a participant has
- *share_value* — the amount of cash that the held shares are worth
- *portfolio_value* — total worth of the held portfolio
- *pandl* — profit and loss in the current round
- *action* — trading action:
  - BUY: A participant bought an amount of shares.
  - SELL: A participant sold an amount of shares.
  - HOLD: A participant did not perform any action during a tick.
  - START: Marks the start of a round in the trading round (day 0)
  - END: Marks the end of a round in the trading round (last day)
- *timestamp* — the exact timestamp at which the action occurred
- *quantity* — if current action is BUY or SELL, this specifies the number of shares bought/sold
- *owned_shares* — the number of shares a participant has; value of this variable is the same as "shares" and prefix "owned" is added for clarification in the custom report
- *asset* — asset name (configurable in session configurations); this variable is important, when stimuli price data corresponds to more than one asset
- *roi* — Return on Investment expressed as the percentage of *portfolio_value* at the experimental day 0 wrt. the last experimental day in the round.

The data is available in "Excel" (.xlsx) or "Plain" (.csv) format. One data file contains the entire data for each experimental session of the ZTS app.

5. Previous data collections and potential extensions

In order to test functionality and applicability of ZTS as an experimental and didactic tool, in twelve studies, we collected data from 2132 participants total. These data collections took place in three university laboratories at ETH Zurich (behavioural and neuroscience labs) and at the University of Basel (behavioural lab). We collected data online via crowdsourcing platforms such as Amazon Mechanical Turk and Cloud Research, from employees of Swiss banks and from economics and from management students of Radboud University, Nijmegen. We also ran a series of small-scale pilot studies that served for software bug testing. In Appendix D, we share insights from the development of the software and experimental design. Our final choice of several software features was motivated by intermediate empirical observations. Other researchers adapting ZTS to their needs or designing their own experimental software may find these insights useful.

5.1. Recommendation of software settings

In all experiments, we exposed participants to 1–7 rounds preceded by one practice round. Experiments with a larger number of rounds (5–7 rounds) appear to be more appropriate for laboratory (rather than online) studies, because longer experiments require continuously staying focused for a longer period of time. We recommend implementing up to three experimental rounds in online studies if the overall time for the experiment does not justify requiring participants to come to a laboratory. Based on several pilot and actual experiments, we conclude that a practice round should employ an artificial price path that displays no clear pattern and does nor resemble a real-life market price pattern. Previous research (Borsboom and Zeisberger, 2019; Grosshans and Zeisberger, 2018; Nolte and Schneider, 2018) showed that

[1] Number of endowed shares is rounded down to the nearest integer. It is impossible to own or trade a fraction of a share.

[2] For the documentation of the oTree standard data fields, check here: https://otree.readthedocs.io/en/latest/models.html.

viewing or experiencing price paths with the same returns but different patterns can impact one's investment decision and risk perception. Therefore, we generated a price path that should induce no priming. During this practice round, we provide participants with instructions about the software and task features in the news box. By using a straight-line pattern in the practice trial, we ensure that participants are unbiased wrt. to the upcoming experimental stimuli.

We recommend using price paths with similar starting price such that in all experimental rounds within one experiment, the quick-trade buttons and the scale of the *y*-axis can stay the same. Exceptions from this rule should be motivated by the research question of the study. We suggest adjusting the initial endowment to the values range of the price paths to control for the cash-to-price ratio. We also recommend adjusting the initial endowment to the sample profile. For example, we settled on endowing participants with 10'000 ECU corresponding to an amount of money a typical individual investor would have available. We also normalise the stock price to start with about 140 ECU.

5.2. Price paths

In our ZTS studies, we only used historical price patterns coming from European market indices. Using an algorithm based on a rolling-window of 252 trading days, we selected price patterns from the history of closing prices of DAX and SIX market indices, searching for particular return-on-investment between the first and the last day of about a certain amount and fixed average annual volatility range. Based on these criteria, we selected four *flat price* scenarios characterised by relatively low average annual volatility. Using the same search algorithm, we selected two *exogenous crashes* (i.e., market crashes caused by an external event such as a pandemic, a piece of important news, elections etc.) characterised by a sharp price decline of at least 10% from the maximum to the minimum price within less than 7 days. One of the exogenous crashes was characterised by the price recovery to yield a very similar return-on-investment as the flat price paths, while the other exogenous crash ended with a negative return. We also selected one *endogenous crash* (i.e., a market crash caused by a slow price decay due to internal processes of the financial system) characterised by similar average annual volatility and price decline as the endogenous crashes, but the price pattern resembles a bubble-and-crash scenario (Sornette, 2006). These price paths constituted the base for further modifications of these price paths. Modifications included price normalisation and price scaling (i.e., artificially inducing an exogenous crash following the bubble of the endogenous crash). We provide all relevant price paths and their documentation on the Open Science Framework platform.

5.3. Previous findings with the ZTS

In an experiment with 807 online participants, Andraszewicz et al. (2022) induced upward social comparison induced via presenting outcomes of high-performing participants in a similar fashion as it occurs on social trading platforms. In their experiment, upward social comparison increased risk taking and trading activity of other participants. They also found that participants exposed to an upward social comparison were less satisfied with their own trading performance.

Also, a working paper which is included in the doctoral thesis of Dániel Kaszás.[3] describes a study investigating the relation between overconfidence in a form of overplacement and risk taking and trading activity. Before mTurk participants completed the ZTS task, they completed a financial literacy quiz and received feedback about their financial knowledge in a form of a range within which their actual score lays. Half of the participants received a range, in which their actual score was exactly in the middle of the range, while the other half of participants received a range, in which their actual score was at the lower end of the range. This resulted in a successful induction of overplacement that showed a small influence on trading activity.

5.4. Further extensions

ZTS offers a number of future extensions using the standard setup, as well as customisation of the experiment by changing configuration features or changing the source code. The standard setup allows for investigating risk taking and trading activity under various compensation schemes, such as a fixed salary, watermark payment, bonus compensations etc. In a between-subject design, participants in different experimental conditions would perform the same trading task but they would be compensated differently in each condition. ZTS would allow for measuring at which point during the trading participants endowed with different compensation schemes change their trading behaviour, depending on whether they reach their goal or not. In a similar fashion, one could investigate the effect of framing on trading activity, where participants would be put in a situation of an asset manager for an external firm vs. a person managing their own assets. Other extensions could involve investigating the influence of individual traits or skills, such as financial literacy, numeracy, personality, etc. on trading activity. For example, an experiment could compare a group of financially more literate participants (i.e., finance students who passed their core courses) and a group of financially less literate participants (i.e., students of non-finance related study programmes). Alternatively, one could use a median-split design to, for example, investigate the impact of impulsivity on trading activity. It would also be possible to investigate the impact of various news, presented during the same market conditions, on trading activity, risk-taking and performance.

Another type of extensions would involve using various price paths and market scenarios, aligned with different pieces of news. ZTS would allow to replicate the results of Lejarraga et al. (2016) in a dynamic setting that also allows for measuring trading activity. An experimenter could upload price patterns corresponding to a historical boom and boost, either the same as in Lejarraga et al. (2016) or modified, to test not only risk taking, but also trading activity during these two contrasting market scenarios. ZTS could also be linked with software measuring physiological reaction to various market scenario such as market crashes. Previous research (see Coates and Herbert, 2008; Coates et al., 2010; Cueva et al., 2015, for example) documented the link between risk taking on the stock market and trader's physiology, such as hormonal levels of testosterone and cortisol. ZTS allows for a controlled measurement of the link between hormonal levels and trading behaviour such as trading activity and risk taking. Also, due to its price display features, ZTS enables to accurately link arousal measured with the skin conductivity response and bodily emotional response to various market scenarios. In a field experiment, Lo and Repin (2002) measured skin conductance, blood volume pulse, heart rate, electromyographical signals, respiration and body temperature of ten professional traders and found not only physiological reaction to high-volatility market periods, but also individual differences in the baseline reaction and difference between the baseline and the stressful situation. ZTS offers a setup to measure physiological reactions to strictly designed market patterns and their accompanying news.

[3] The thesis is available at https://www.research-collection.ethz.ch/handle/20.500.11850/528421.

Further, an experimenter could modify the source code to investigate features of the interface of the trading software. Some simple manipulations would involve changing location or colours of elements of the trading interface. Other manipulations would involve the influence of the scale of *y*-axis scale on trading behaviour. Huber and Huber (2019) demonstrated that the scale of the y-axis impacts the perceived riskiness of an assets and an investor's willingness to invest in the asset. ZTS offers a natural extension of their study in which actual trading behaviour could be measured depending on the scale of the price path. This could be done by changing the way *y*-axis adjusts to the asset price. One could also change the code of the price chart to display more than one asset to investigate the impact of viewing the market index next to the traded asset.

It is out of scope of this paper to list all possible experimental designs that would profit from ZTS' setup. Here, we aimed to demonstrate the variety of applications of a dynamic trading software such as ZTS. Information about how to customise an experiment by adapting the configuration and the source code is provided in Appendix A. We also believe that making the experimental software freely available can encourage the scientific community to conduct replication studies.

6. Concluding remarks

Experiments on the intersection of finance, economics and psychology require a wider range of experimental designs accompanied with software that is easy to use and flexible for implementing variations of the design. Programming and testing such software can be time-consuming and costly. In this paper we present the Zurich Trading Simulator (ZTS) for oTree, which is a dynamic tool in which individuals can trade an asset as a price taker, similar to most individual investors. The tool differs from previous designs in several aspects. First, it resembles real-world investing by displaying continuously changing price ticks. Second, participants can complete various numbers of trades, while most previous designs impose a fixed number of investment decisions such that the number of trades is the same for all participants. In ZTS, every participant can develop their own 'investment path", as it occurs in real markets. Therefore, ZTS enables the researchers to investigate how and which individuals make gains and losses, trade more or less actively, and take more or less risk, under the same market conditions. "An understanding of how consequential and nonconsequential experiences affect risk taking will [...] be necessary to further improve the communication of financial risks " (Lejarraga et al., 2016, p. 376). Thanks to its dynamic feature, ZTS also allows for analysing measures of trading activity such as volume, value of traded portfolio, size of trades and frequency of trades. In contrast, most previous experimental designs focus on measuring portfolio risk or a choice between more or less risky investment. We provide the software as freeware. Anyone who would like to use the software, is free to do so if they cite this paper. We encourage other researchers to further use and modify the software depending on their research questions.

CRediT authorship contribution statement

Sandra Andraszewicz: Conceptualisation, Methodology, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualisation, Supervision, Project administration, Funding acquisition. **Jason Friedman:** Software, Writing – review & editing. **Dániel Kaszás:** Conceptualisation, Methodology, Formal analysis, Investigation, Resources, Writing - review & editing. **Christoph Hölscher:** Conceptualisation, Writing - review & editing, Supervision, Funding acquisition.

Acknowledgements

We would like to thank Dinesh Pothineni, Mariyana Koleva, Lin Cong, Armin Grossrieder, David Hofer and Stefan Wehrli for their help in the implementation of the software. We also would like to thank Stefan Zeisberger for very useful comments on the manuscript.

This research was primarily funded by Cooper Fonds grant No. ETH-08 17-1 and ETH Grant No. ETH-15 16-2. This research was partially supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme, Future Resilient Systems (FRS).

Appendix A. Python and oTree installation

Make sure that a recent Python version (3.x) is installed on your machine before continuing with the next steps. Find out if a Python version is installed by issuing the following command in the terminal and checking the response:

```
$ python --version
Python 3.7.5
```

If Python is not yet installed follow the steps for your specific operating system listed on the following website: https://realpython.com/installing-python/

Follow the instructions on the oTree Documentation website to install the latest version of oTree on your working machine: https://otree.readthedocs.io/en/latest/install.html.

Appendix B. ZTS installation on oTree

ZTS requires Python 3.x and oTree. Appendix A provides installation instructions for Pyhton 3.x and oTree. Once you have installed Python and oTree, download the zip-file of the ZTS source code on GitHub from https://github.com/Zurich-Trading-Simulator/OtreeZTS by clicking on Code/Download ZIP. Extract the ZIP file into your working location. Researchers familiar with GitHub can clone the repository to their working directory. Alternatively, you can download the .otreezip file of the ZTS directly on the oTreeHub from https://www.otreehub.com/projects/otree-zts/ into your working directory and unzip it using the following command:

```
$ otree unzip /path/to/workingdirectory/otree-zts.otreezip
```

To start the application locally for development and testing purposes follow the steps listed below:

Open your terminal and change to the ZTS directory you set up previously.

```
$ cd /path/to/workingdirectory/otreeZTS/
```

Install the Python dependencies (i.e., libraries) required for ZTS by running the following command:

```
$ pip install -r requirments_base.txt
```

Start ZTS in oTree on a local testing server with the following command:

```
$ otree devserver
```

Open your browser to http://localhost:8000/ and the application should appear. When you make changes to the code, save them. The ZTS application running in the browser will automatically implement your edits. We will post any potential changes to the ZTS resulting from the evolution of oTree and the corresponding documentation on https://github.com/Zurich-Trading-Simulator/OtreeZTS.

Appendix C. Setting up a custom experiment

In Appendix B, we outlined how to run a default ZTS experiment. In this section, we provide information about setting up a customised experiment. To start an experiment on oTree, an experimenter needs to set up a session. In oTree, a session is a term for the entire experiment using the same or at least similar application configurations. An example of an experimental session in ZTS would be the following situation: A number of participants in a laboratory or online are invited to complete the ZTS task, followed by a questionnaire. Participants receive payment for their participation.

In contrast, an oTree session is a series of sub-sessions. oTree sub-sessions are the "sections" or "modules" that constitute a session. For example, in ZTS, each trading round and the post task questionnaire are separate sub-sessions. Each sub-session can be further divided into groups of players. For example, you could have a sub-session with 30 players, divided into 15 groups of 2 players each. In oTree, the terms *player* and *participant* have distinct meanings. The relationship between participant and player is the same as the relationship between session and sub-session — a player is an instance of a participant in one particular sub-session. A player is like a temporary "role" played by a participant. A participant refers to the same physical person during the whole session. The same participant can be player 2 in the first sub-session, player 1 in the second sub-session.

C.1. Session configuration

To set up an experimental session on ZTS follow these steps:

1. Click on *Sessions* in the navigation bar on the top of the startpage.
2. Click on *Create New Session* or click on an existing session to continue with an existing experiment.
3. Choose the *ZTS* session configuration and type in the desired number of participants. The session configuration specifies the course of the experiment, for example that ZTS is followed by a Survey. It is also possible to add a Survey to the beginning of the experiment by adding a new session configuration.
4. On the same page click on "Configure Session" and change the default configurations as desired. Alternatively, one can open the file "settings.py" to edit the same settings directly in the code instead of using the user interface. It is possible to configure the following session settings:
   - session_name — the name of the session.
   - survey_link — link to the survey (if required), which will automatically appear after the ZTS task.
   - timeseries_file — name of the timeseries file you want to use in this session. Only use the stem of the filename here. For example, for the timeseries files demo_1.csv, demo_2.csv, etc. type in "demo". The underscore, round number and file ending will be added automatically by ZTS.
   - num_rounds — number of ZTS trading rounds in the experiment. This number should correspond to the number of timeseries files.
   - refresh_rate — duration of one market day/tick in ms.
   - initial_cash — initial cash endowment for each round (at the beginning of each round, half of the cash is automatically invested in the risky asset).
   - random_round_payoff — if true, final payoff from the experiment will correspond to one randomly chosen round, instead of accumulating the payoff from all rounds in the experiment.
   - training_round — if true, the first round will be in practice mode, meaning that the payoff of this round will not count towards the final payoff.
   - real_world_currency_per_point — the conversion rate (i.e., a multiplier) from the experimental currency to the real currency.
5. Click on *Create*.

After you have created a new ZTS session, you will be redirected to the Session Menu, where you will be able to see the session links that can be distributed to the participants to start the experiment. The Session Menu consist of several tabs:

- Description — provides a description of the sequence of apps (e.g., ZTS, Survey).
- Links — lists distribution links for participants. There are three types of links: *session-wide link* — a single link that can be used by all participants to complete the experiment, which automatically redirects each participant to a new oTree Participant; this link makes distribution of the experiment easy, but it does not prevent the same person from participating in the experiment more than once, *persistent links* — permanent links that persist over time; this feature is only enabled when inside a room (see Appendix C.4 for more information); *single-use links* — unique links assigned to individual participants.
- Edit — enables editing some session configurations after creating a session.
- Monitor — allows the experiment administrator to monitor the experiment including the status and progress of each participant. It also allows the experimenter to enforce a page advancement for particularly slow participants.
- Data — allows tracking data during an experimental session. This feature is very useful for debugging.
- Payments — list of payments that each participant should receive at the end of the experiment. Payment is automatically calculated by the ZTS based on the session configuration. If you set *training_round* = *True*, the first round will not be considered to the payoff calculation. There are two ways of calculating the payoff. It is either calculated from a sum of all experimental rounds or one round is randomly chosen to define the payoff. The payoff is calculated from the portfolio value at the end of each round. If you want to alter the existing payoff options, modify the source code in "/ZTS/models.py", function "set_payoff".

By default, ZTS will display to the participants the ZTS task followed by a survey. In order to change this sequence, modify "SESSION_CONFIGS" in the file "/settings.py" by adding the code signalled by ### :

```
   SESSION_CONFIGS = [
   dict(
      name=’ZTS’,
      num_demo_participants=1,
      app_sequence=[’ZTS’, ’Survey’]
   ),

   ################################################################
   #### Add the following Lines to add a new session config ####
   dict(
      name=’ZTS_new’
      num_demo_participants=1,
      app_sequence=[’Survey’, ’ZTS’]
   ),
   ################################################################
]
```

C.2. Changing page content

All pages (i.e., the instructions page, the final payoff page etc.) displayed in the ZTS application are formatted in HTML and are located in "/ZTS/templates/ZTS/" or "/Survey/templates/Survey/". These pages are editable templates. In addition to standard HTML syntax, oTree also allows template syntax on their pages, as stated in https://otree.readthedocs.io/en/latest/templates.html.

C.3. Changing and customising timeseries files

As introduced above, ZTS allows for presenting participants with customised price patterns. To upload a new price pattern, upload it in a CSV-file to "/_static/ZTS/timeseries_files/". The files that you place in this directory should have the same stem name and should follow this naming procedure "stemname_[roundnumber].csv". Make sure to use the same stemname in the session configuration.

The CSV-file should follow a very specific format. It should be comma delimited and it must contain two columns "Adjusted-Close" corresponding to the asset prices for each experimental day and "News" needed to display the text in the News Box. All stimuli provided along this paper contain the "Date" column, which is not required for ZTS to run, but may provide useful information to the experimenter.

C.4. Rooms

OTree gives the possibility to open virtual experiment rooms, that offer some additional features compared to normal sessions. These features include generating links that an experimenter can assign to participants or laboratory computer, which stay constant across sessions. Another feature is a "waiting room" that lets an experimenter see which participants are currently waiting to start a session. The third feature of oTree rooms are short links that are easy for participants to type and work well for quick live demos. You can find more information on Rooms and how to set them up in the following oTree documentation https://otree.readthedocs.io/en/latest/rooms.html.

C.4.1. Deployment

Running the application locally, is sufficient for testing purposes, but the user must be aware that in order to perform any experiments the application needs to be run in a production environment, to guarantee safety and correctness. To deploy the ZTS for production follow the oTree documentation steps under https://otree.readthedocs.io/ja/latest/server/intro.html.

C.4.2. Amazon Mechanical Turk (mTurk)

In order to use ZTS on mTurk, go to oTree's admin interface and publish your session to mTurk. Afterwards, mTurk workers can participate in your session. From oTree's admin interface, you send each participant their participation fee and bonus (payoff). This oTree documentation provides details on setting up an experiment on mTurk https://otree.readthedocs.io/en/latest/mturk.html.

Appendix D. Insights from the ZTS development process

First, in early pilot studies and experiments, we made a few observations regarding the user interface of the software. We found it more sensible to implement quick-trade buttons instead of an open field to enter a custom number of shares to trade, due to the pace of the evolving market. We also found that the vertical axis should re-scale with every new tick in order to make it impossible to guess whether the upcoming price would increase or decline. In the standard ZTS setup, the price path starts in the middle of the y-axis. Very minor changes to the vertical axis of the price chart can result in substantial changes in risk taking of the trading participants. Therefore, we suggest using the standard display when comparing results across various experiments. Also, initially, we included a button that would allow a participant view the history of their trades. We later removed this button because participants did not use it implying that in our fast-pace dynamic task, participants did not reflect on their past trades, presumably because they did not want to miss any price movement.

References

Aldrich, E.M., Demirci, H.A., Vargas, K.L., 2020. An otree-based flexible architecture for financial market experiments. J. Behav. Exp. Finance 25, 100205. http://dx.doi.org/10.1016/j.jbef.2019.03.007/.

Andraszewicz, S., Kaszás, D., Zeisberger, S., Hölscher, C., 2022. The Influence of Upward Social Comparison on Retail Trading Behavior. Working Paper Available on the Open Science Framework, http://dx.doi.org/10.31219/osf.io/48deq.

Borsboom, C., Zeisberger, S., 2019. What makes an investment risky? An analysis of price path characteristics. J. Econ. Behav. Organ. 169, 92–125. http://dx.doi.org/10.1016/j.jebo.2019.11.002.

Bradbury, M.A.S., Hens, T., Zeisberger, S., 2015. Improving investment decisions with simulated experience. Rev. Finance 19, 1019–1052. http://dx.doi.org/10.1093/rof/rfu021.

Brehmer, B., 1992. Dynamic decision making: Human control of complex systems. Acta Psychol. 81, 211–241. http://dx.doi.org/10.1016/0001-6918(92)90019-A.

Brehmer, B., Dörner, D., 1993. Experiments with computer-simulated microworlds: Escaping both the narrow straits of the laboratory and the deep blue sea of the field study. Comput. Hum. Behav. 9, 171–184. http://dx.doi.org/10.1016/0747-5632(93)90005-D.

Chen, D.L., Schonger, M., Wickens, C., 2016. oTree – an open-source platform for laboratory, online, and field experiments. J. Behav. Exp. Finance 9, 88–97. http://dx.doi.org/10.1016/j.jbef.2015.12.001.

Coates, J., Gurnell, M., Sarnyai, Z., 2010. From molecule to market: steroid hormones and financial risk-taking. Philos. Trans. R. Soc. Ser. B 365, 331–343. http://dx.doi.org/10.1098/rstb.2009.0193.

Coates, J.M., Herbert, J., 2008. Endogenous steroids and financial risk taking on a London trading floor. Proc. Natl. Acad. Sci. 105 (16), 6167–6172. http://dx.doi.org/10.1073/pnas.0704025105.

Crosetto, P., Weisel, O., Winter, F., 2019. A flexible z-tree and otree implementation of the social value orientation slider measure. J. Behav. Exp. Finance 23, 46–53. http://dx.doi.org/10.1016/j.jbef.2019.04.003.

Cueva, C., Roberts, R., Spencer, T., Nisha, R., Tempest, M., Tobler, P., Herbert, J., Rutschini, A., 2015. Cortisol and testosterone increase financial risk taking and may destabilize markets. Sci. Rep. 5 (11206), 1–16. http://dx.doi.org/10.1038/srep11206.

DiFonzo, N., Bordia, P., 1997. Rumor and prediction: Making sense (but losing dollars) in the stock market. Organ. Behav. Human Decis. Process. 71, 329–353. http://dx.doi.org/10.1006/obhd.1997.2724.

Doyle, L., Schindler, D., 2019. μCap: connecting FaceRender^{TM} to z-tree. J. Econ. Sci. Assoc. 5, 136–141. http://dx.doi.org/10.1007/s40881-019-00065-1.

Edwards, W., 1962. Dynamic decision theory and probabilistic information processings. Hum. Fact. J. Hum. Fact. Ergon. Soc. 2, 59–67. http://dx.doi.org/10.1177/001872086200400201.

Fischbacher, U., 2007. z-Tree: Zurich toolbox for ready-made economic experiments. J. Exp. Econ. 10, 171–178. http://dx.doi.org/10.1007/s10683-006-9159-4.

Gonzalez, C., Vanyukow, P., Martin, M.K., 2005. The use of microworlds to study dynamic decision making. Comput. Hum. Behav. 21, 273–286. http://dx.doi.org/10.1016/j.chb.2004.02.014.

Grosshans, D., Zeisberger, S., 2018. All's well that ends well? On the importance of how returns are achieved. J. Bank. Financ. 87, 397–410. http://dx.doi.org/10.1016/j.jbankfin.2017.09.021.

Hertwig, R., Erev, I., 2009. The description-experience gap in risky choice. Ignite. Sci. 13, 517–523. http://dx.doi.org/10.1016/j.tics.2009.09.004.

Hogarth, R.M., Lejarraga, T., Soyer, E., 2015. The two settings of kind and wicked learning environments. Curr. Direct. Psychol. Sci. 24, 379–385. http://dx.doi.org/10.1177/0963721415591878.

Hogarth, R.M., Soyer, E., 2015. Communicating forecasts: The simplicity of simulated experience. J. Bus. Res. 68, 1800–1809. http://dx.doi.org/10.1016/j.jbusres.2015.03.039.

Holzmeister, F., 2017. oTree: Ready-made apps for risk preference elicitation methods. J. Behav. Exp. Finance 16, 33–38. http://dx.doi.org/10.1016/j.jbef.20217.08.003.

Holzmeister, F., Kerschbamer, R., 2019. oTree: The equality equivalence test. J. Behav. Exp. Finance 22, 214–222. http://dx.doi.org/10.1016/j.jbef.2019.04.001.

Holzmeister, F., Pfurtscheller, A., 2016. oTree: The "bomb" risk elicitation task. J. Behav. Exp. Finance 10, 105–108. http://dx.doi.org/10.1016/j.jbef.2016.03.004.

Huber, C., 2019. oTree: The bubble game. J. Behav. Exp. Finance 22, 3–6. http://dx.doi.org/10.1016/j.jbef.2018.12.001.

Huber, C., Huber, J., 2019. Scale matters: risk perception, return expectations, and investment propensity under different scalings. Exp. Econ. 22, 76–100. http://dx.doi.org/10.1007/s10683-018-09598-4.

Huber, C., Huber, J., Kirchler, M., 2022. Volatility shocks and investment behavior. J. Econ. Behav. Organ. 194, 56–70. http://dx.doi.org/10.1016/j.jebo.2021.12.007.

Jiang, M., Li, J., 2019. J. Behav. Exp. Finance 22, 90–92. http://dx.doi.org/10.1016/j.jbef.2019.02.001.

Kerschbamer, R., 2015. The geometry of distributional preferences and a non-parametric identification approach: The equality equivalence test. Eur. Econ. Rev. 76, 85–103. http://dx.doi.org/10.1016/j.euroecorev.2015.01.008.

Lejarraga, T., Woike, J.K., Hertwig, R., 2016. Description and experience: How experimental investors learn about booms and busts affects their financial risk taking. Cognition 157, 365–383. http://dx.doi.org/10.1016/j.cognition.2016.10.001.

Lo, A.W., Repin, D., 2002. The psychophysiology of real-time financial risk processing. J. Cogn. Neurosci. (3), 323–339. http://dx.doi.org/10.1162/089892902317361877.

Moinas, S., Pouget, S., 2013. The bubble game: An experimental study of speculation. Econometrica 81, 1507–1539. http://dx.doi.org/10.3982/ECTA9433.

Murphy, R.O., Andraszewicz, S., Knaus, S.D., 2016. Real options in the laboratory: An experimental study of sequential investment decisions. J. Behav. Exp. Finance 12, 23–39. http://dx.doi.org/10.1016/j.jbef.2016.08.002.

Niehorster, D.C., Nyström, M., 2020. SMITE: A toolbox for creating psychophysics toolbox and psychopy experiments with SMI eye trackers. Behav. Res. Methods 52, 295–304. http://dx.doi.org/10.3758/s13428-019-01226-0.

Nolte, S., Schneider, J., 2018. How price path characteristics shape investment behavior. J. Econ. Behav. Organ. 154, 33–59. http://dx.doi.org/10.1016/j.jebo.2018.07.018.

Palan, S., 2013. A review of bubbles and crashes in experimental asset markets. J. Econ. Rev. 27, 570–588. http://dx.doi.org/10.1111/joes.12023.

Peirce, J.W., 2007. PschopPy – psychophysic software in python. J. Neurosci. Methods 162, 8–13. http://dx.doi.org/10.1016/j.jneumeth.2006.11.017.

Peirce, J.W., 2009. Generating stimuli for neuroscience using PsychoPy. Front. Neuroinform. 2, 10. http://dx.doi.org/10.3389/neuro.11.010.2008.

Peters, E., Slovic, P., 2000. The springs of actions: Affective and analytical information processing in choice. Pers. Soc. Psychol. Bull. 26 (12), 1465–1475. http://dx.doi.org/10.1177/01461672002612002.

Pichl, B., 2019. RAM: A collection of mechanisms for (indivisible) resource allocation in oTree. J. Behav. Exp. Finance 23, 133–137. http://dx.doi.org/10.1016/j.jbef.2019.05.006.

Rapoport, A., 1975. Research paradigms for studying dynamic decision behavior. In: Wendt, D., Vlek, C. (Eds.), Utility, Probability, and Human Decision Making: Selected Proceedings of an Interdisciplinary Research Conference, Rome, 3–6 September, 1973. Springer Netherlands, Dordrecht, pp. 349–369. http://dx.doi.org/10.1007/978-94-010-1834-0_22.

Saral, A.S., Schröter, A.M., 2019. zBrac – A multilanguage tool for z-tree. J. Behav. Exp. Finance 23, 59–63. http://dx.doi.org/10.1016/j.jbef.2019.04.006.

Smith, V.L., Suchanek, G.L., Williams, A.W., 1988. Bubbles, crashes, and endogenous expectations in experimental spot asset markets. Econometrica 56 (5), 1119–1151. http://dx.doi.org/10.2307/1911361.

Sornette, D., 2006. Endogenous versus exogenous origins of crises. In extreme events in nature and society. In: Albeverio, S., Jentsch, V., Kantz, H. (Eds.), The Frontiers Collection. Springer, Berlin, Heidelberg, http://dx.doi.org/10.1007/3-540-28611-X_5.

von Bülow, C.W., Liu, X., 2020. Ready-made otree applications for the study of climate change adaptation behavior. J. Behav. Exp. Finance 88, 101590. http://dx.doi.org/10.1016/j.socec.2020.101590.

Zweig, J., 2015. The Devil's Financial Dictionary. Public Affairs, New York.