oTree: An open-source platform for lab, web, and field experiments

www.oTree.org

What is oTree?

A platform for lab, web and field experiments

- Open source
- Online (runs in browser)
- Object oriented programming

Background

- oTree started in 2013 at ETH Zurich, Daniel Chen group
- Originally created it for our own needs
  - Ease of use
  - Flexibility
- Decided to release oTree to the community
- Funded by Daniel Chen's 5-year ERC grant

Paper

- Paper recently published http://www.otree.org/oTree.pdf
- Please cite us if you use oTree for your research

OTREE - AN OPEN-SOURCE PLATFORM FOR LABORATORY, ONLINE, AND FIELD EXPERIMENTS

DANIEL L. CHEN, MARTIN SCHONGER, AND CHRIS WICKENS*

oTree is an open-source and online software for implementing interactive experiments in the laboratory, online, the field or combinations thereof. oTree does not require installation of software on subjects' devices; it can run on any device that has a web browser, be that a desktop computer, a tablet or a smartphone. Deployment can be internet-based without a shared local network, or local-network-based even without internet access. For coding, Python is used, a popular, open-source programming language. www.oTree.org provides the source code, a library of standard game templates and demo games which can be played by anyone.

JEL Codes: A20, C88, C90

Keywords: experimental economics, software, laboratory experiments, field experiments, online experiments, classroom experiments

FIGURE 1.— oTree on different devices and operating systems

Experimental economics has become an established field. The common procedure is to conduct incentivized experiments in dedicated university

*Toulouse Institute for Advanced Studies; Center for Law and Economics, D-GESS, ETH Zurich. For comments on the oTree software or the paper, we thank the people at the experimental econ laboratories in Magdeburg, Hamburg and ETH Zurich, Juan Cabral, Donja Darai, Som Datye, Gregor Muellegger, Kelly Reeve, Alexander Sandukovskiy, and Stefan Wehrli. Stefan Bucher provided superb research assistance.

4 usage scenarios

1.) Lab
- Can run in any lab
- Multiple labs simultaneously
- More than 100 simultaneous participants

2.) Online
- Surveys and interactive games
- Hybrid approaches possible (lab + online)
- Amazon Mechanical Turk integration

3.) Field
- Internet access not necessary
- e.g. remote village – bring 30 tablets and server laptop

4.) Classroom
- Post a URL and have students play on their own devices
- Play live in class

Demo

Usage

- Recent studies with oTree
  - Seoul National University, 1000 participants (80-120 per session)
  - University of Bonn, 500+ participants (up to 90 per session)
- Special collaboration with IBSEN group (EU Horizon 2020 grant)
- Researchers at 40 universities
- Many disciplines (economics, psychology, sociology, political science, neuroscience)

Sample games

22 simple games (with source code) available at:
demo.otree.org

Demo Game

Public Goods Game
Prisoners' Dilemma
Trust Game

Dictator Game

Cournot Competition
Bertrand Competition
Stackelberg Competition

Common Value Auction
Private Value Auction

Volunteer's Dilemma
Principal Agent Game

Stag Hunt
Battle of the Sexes
Coordination Game

Matching Pennies
Traveler's Dilemma
Survey
Divide a Pie
Guessing Game

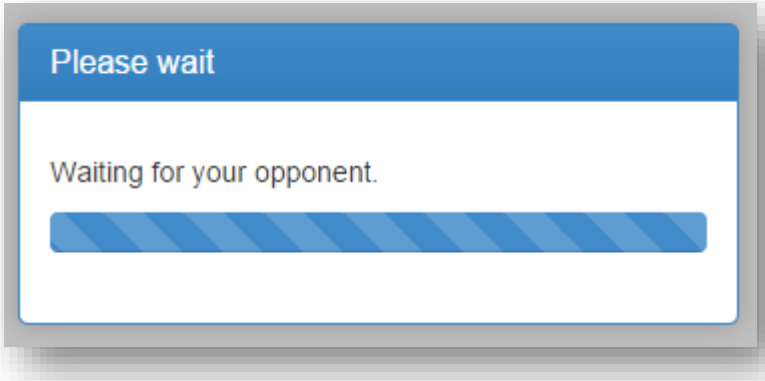
2 x 2 Matrix Game (Symmetric)
2 x 2 Matrix Game (Asymmetric)

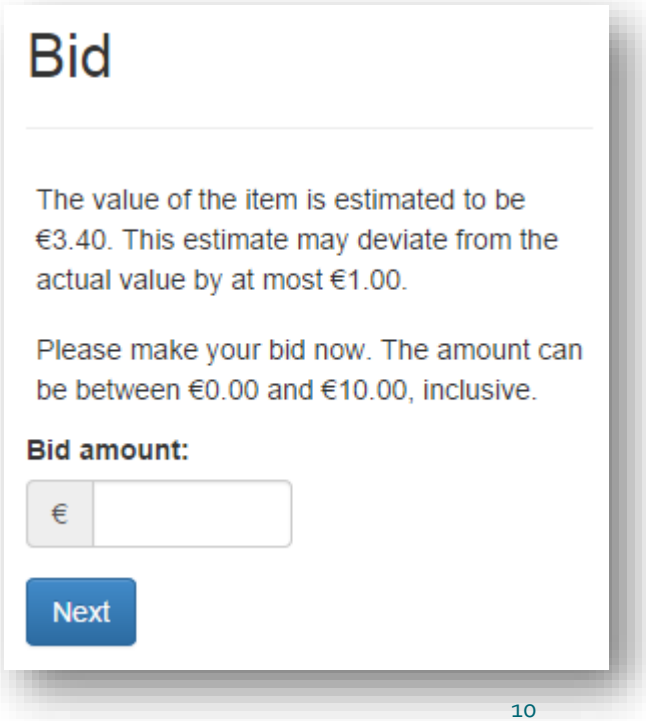
Amazon Mechanical Turk

- Publish to MTurk through oTree's admin interface
- We have run many multi-player games on Mechanical Turk
- Approve worker submissions and send payments
- Filter workers based on location, skill level, etc.

HTML5 user interfaces

- Radio buttons, dropdowns, money input
- Localization: supports any language or currency
- Customizable color theme, header, footer
  - e.g. branding

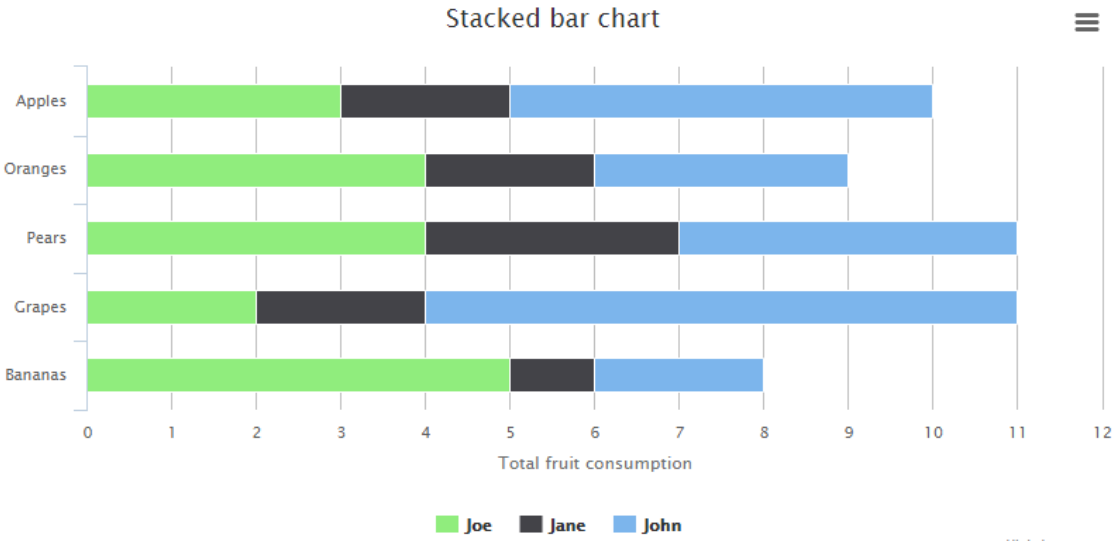


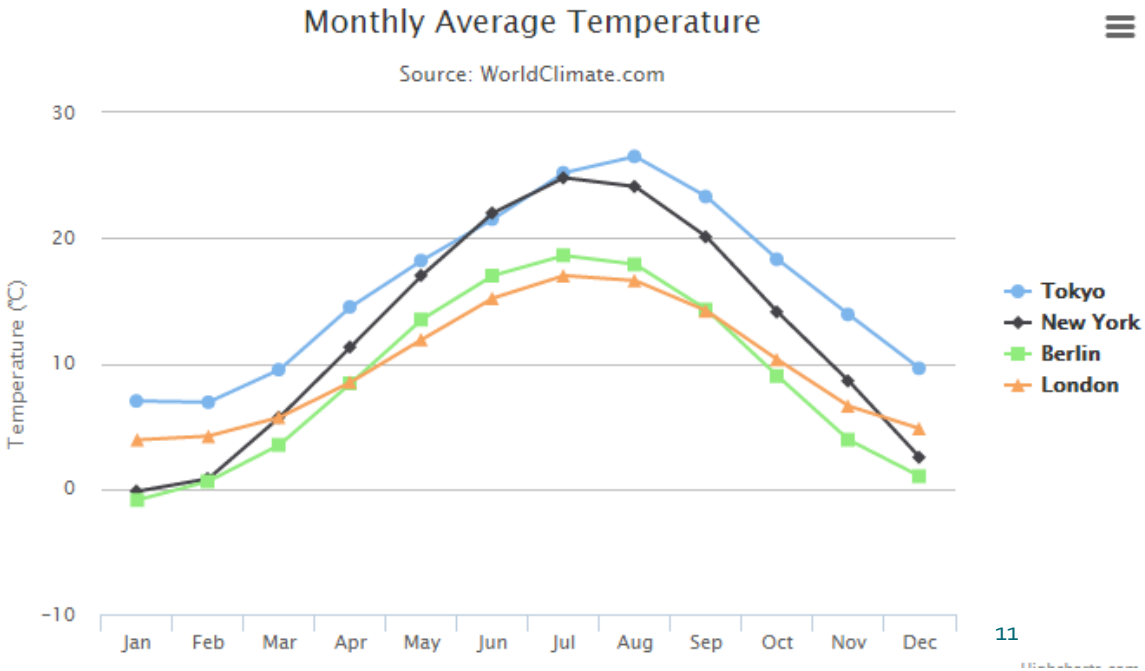


Use charts

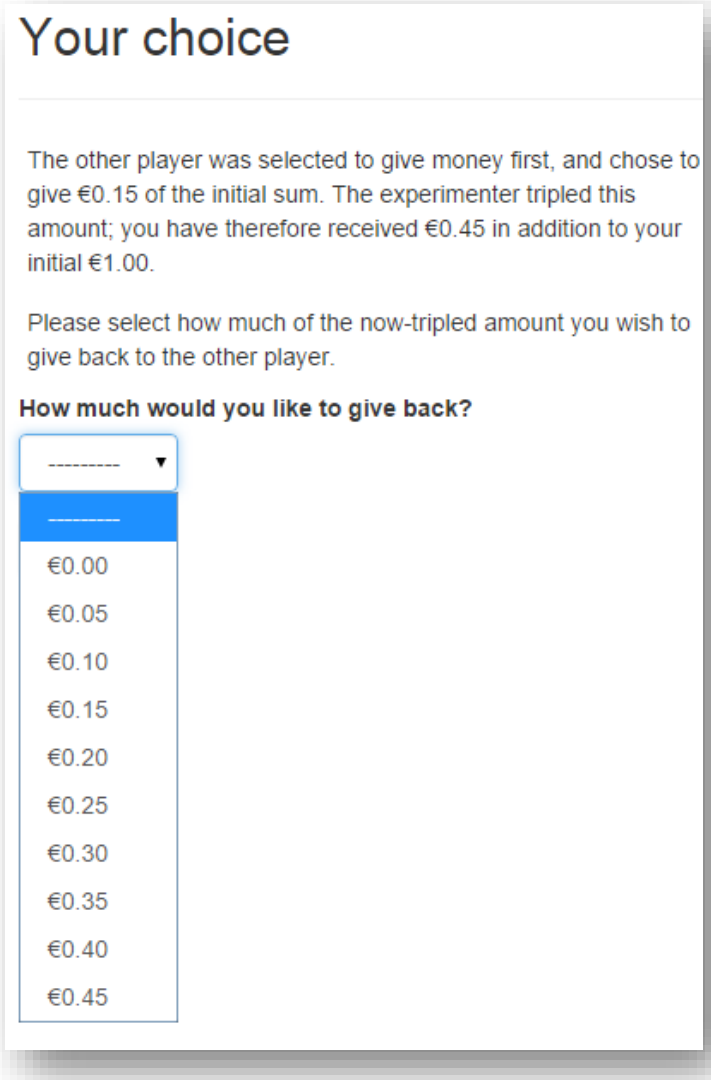
HIGHCHARTS

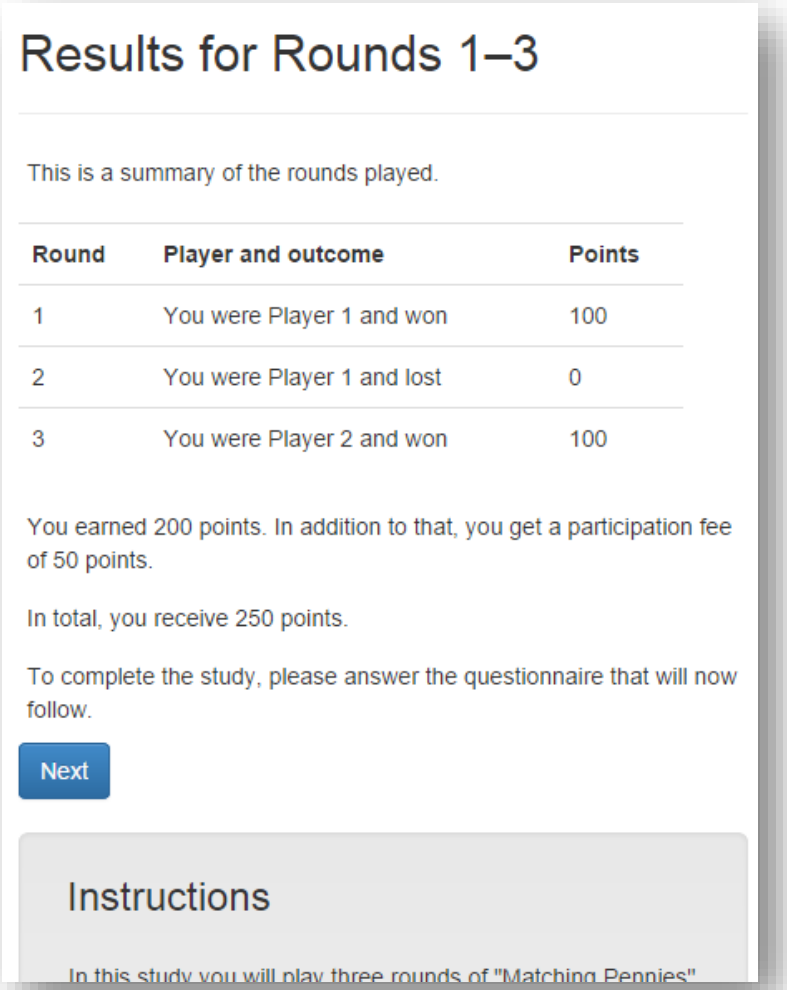
- LINE CHARTS
- AREA CHARTS
- COLUMN AND BAR CHARTS
- PIE CHARTS
- SCATTER AND BUBBLE CHARTS
- DYNAMIC CHARTS
- COMBINATIONS
- 3D CHARTS
- GAUGES
- HEAT MAPS





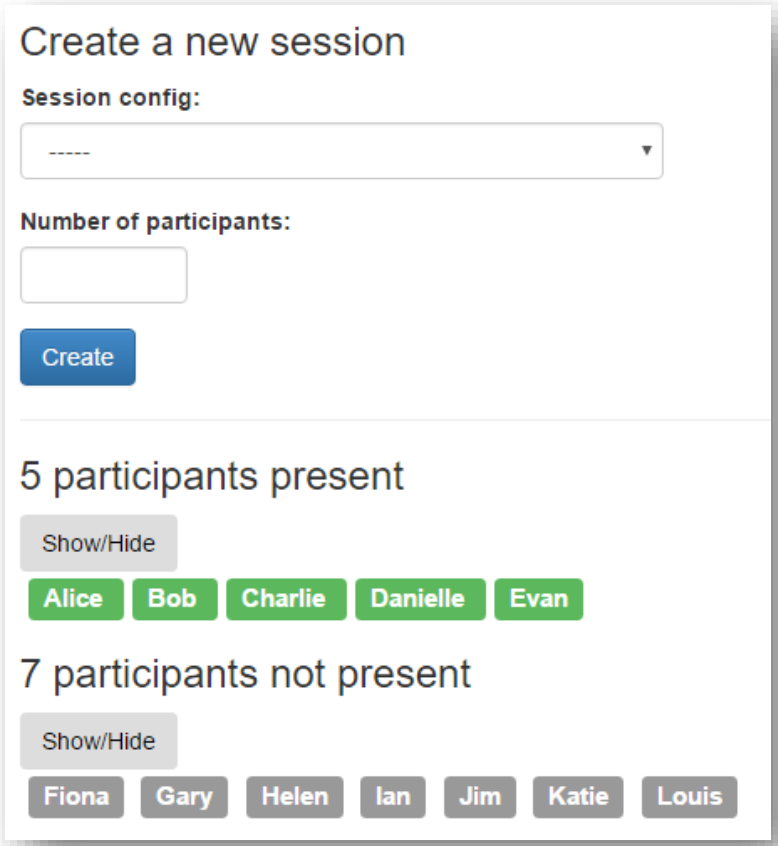
Dynamic elements

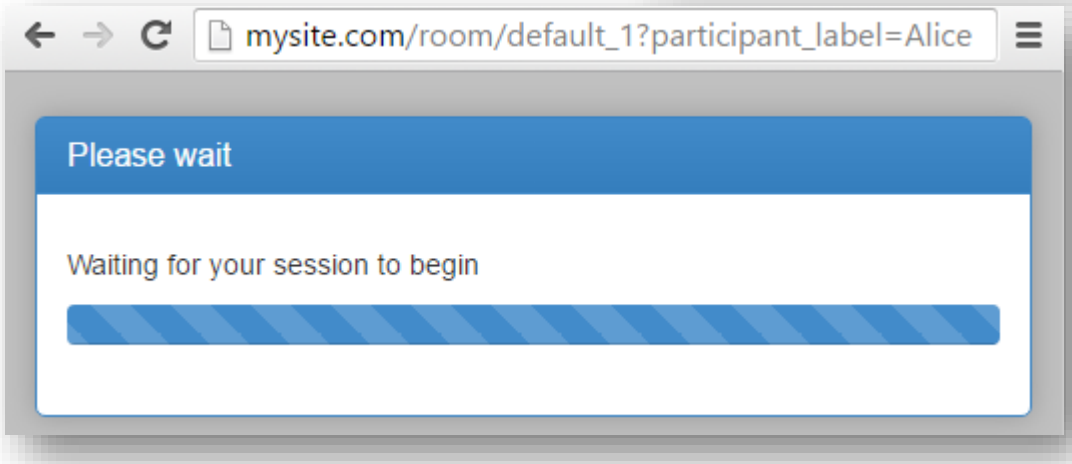




Waiting rooms

- See how many participants are ready, then create a session of appropriate size
- **Classroom**: each student gets a unique permanent URL
- **Lab/field**: each PC has a unique permanent URL
- **Spontaneous live demos**: give the audience an easy-to-type link like http://mysite.com/rooms/demo





Modern programming

- Python is easy to learn , popular, and versatile
- Good skill investment
- Can recruit developers with Python/Django expertise

Automated testing with bots

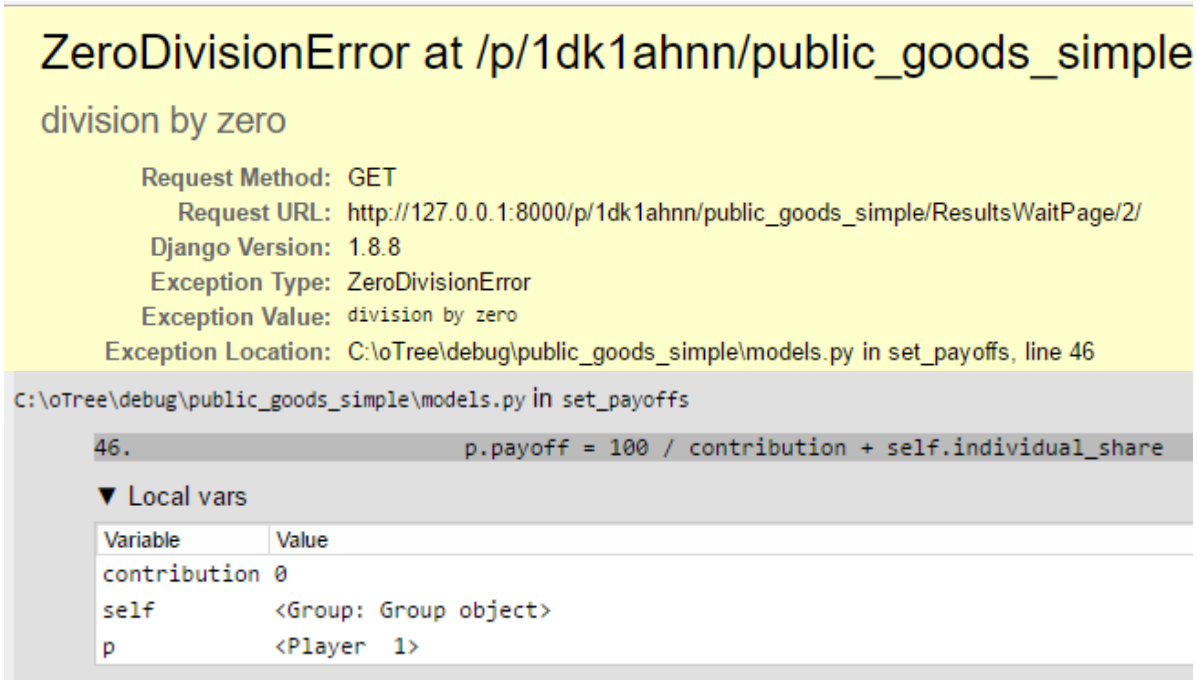
- Bots can simulate hundreds of participants (deterministic or Monte Carlo)
- Detect programming errors as well as design errors (e.g. negative payoffs)
- Saves time by reducing manual testing
- Bots run from command line or web browser

```
class PlayerBot(Bot):

    def play_round(self):
        yield (views.Contribute, {'contribution': c(1)})
        yield (views.Results)
```

```
class PlayerBot(Bot):

    def play_round(self):
        yield (views.Introduction)
        if self.player.id_in_group == 1:
            yield (views.Offer, {'amount_offered': c(10)})
        else:
            if self.group.strategy:
                yield (views.AcceptStrategy, {'response_{}'.format(
                    int(offer)): True for offer in Constants.offer_choices})
            else:
                yield (views.Accept, {'offer_accepted': True})
        yield (views.Results)
```

Informative error messages

Help you resolve programming errors more quickly by pinpointing the source of the bug



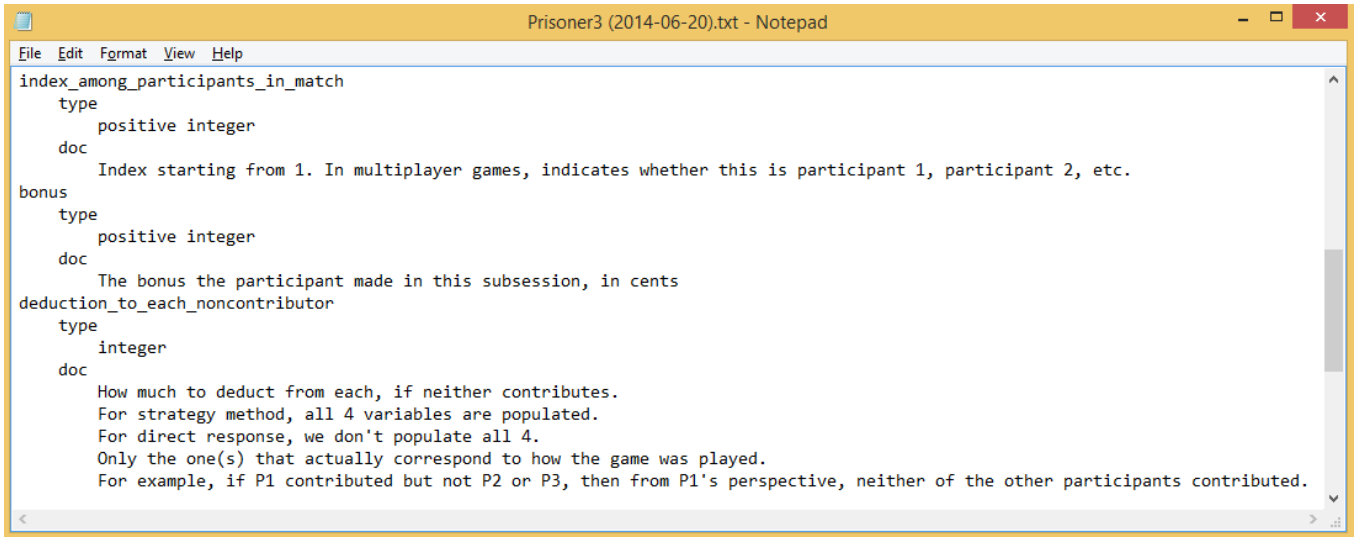
Group matching

```
>>> self.get_group_matrix()

[[<Player  1>, <Player  2>, <Player  3>],
 [<Player  4>, <Player  5>, <Player  6>],
 [<Player  7>, <Player  8>, <Player  9>],
 [<Player 10>, <Player 11>, <Player 12>]]

>>> self.group_randomly(fixed_id_in_group=True)

[[<Player  1>, <Player  8>, <Player 12>],
 [<Player 10>, <Player  5>, <Player  3>],
 [<Player  4>, <Player  2>, <Player  6>],
 [<Player  7>, <Player 11>, <Player  9>]]
```

Export data in CSV format

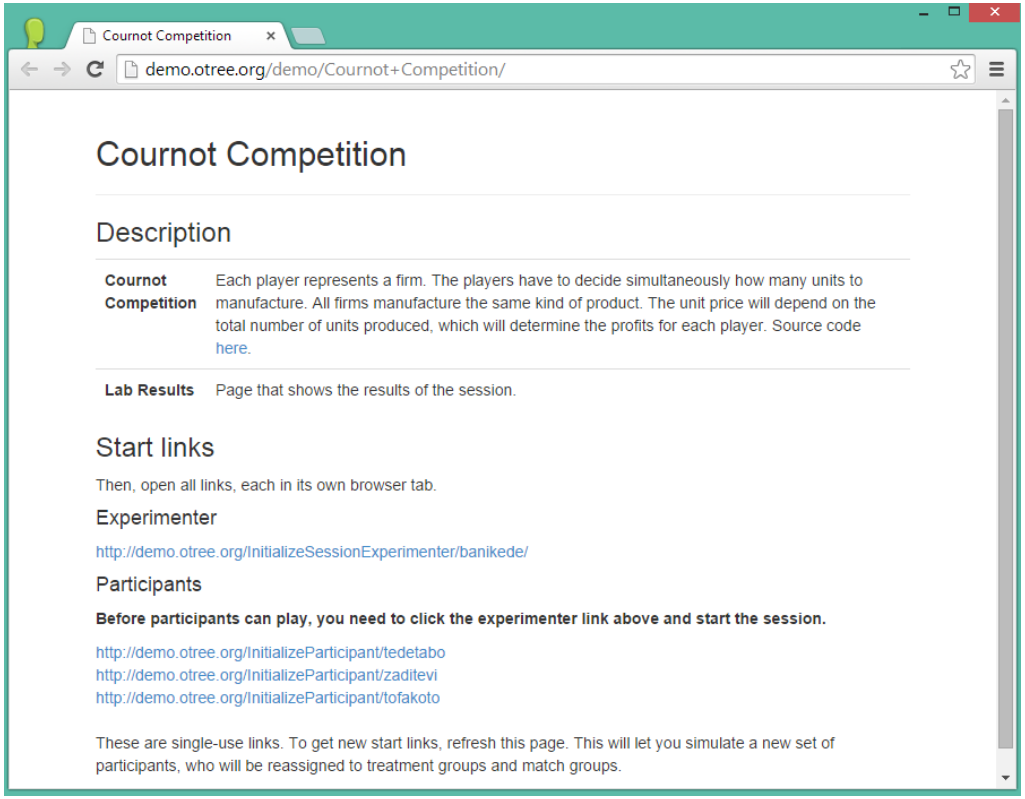
	A	B	C	D	E	F	G	H
1	name	visited	index	bonus	contributed	deduction_noncontrib	deduction_contrib	deduction_1_contrib
2	1	1	3	420	1	-3	-4	-7
3	2	1	3	300	1	-8	-9	-5
4	3	1	1	260	1	-11	-4	-6
5	4	1	2	430	1	-2	-10	0

Documentation extracted from comments in source code

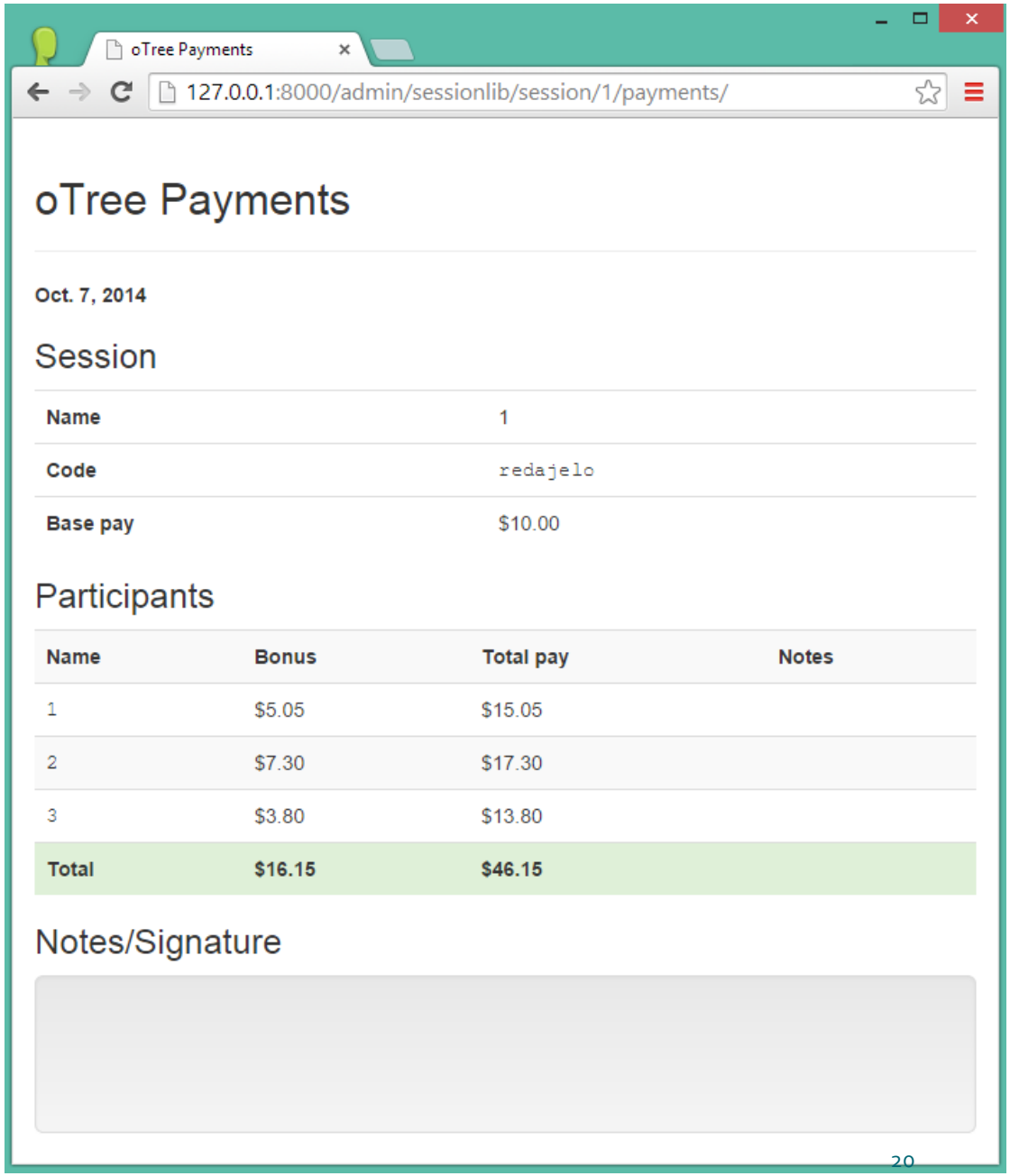


Demo mode

- Easy to put your game online (play in browser, no installation or download)
- Send link to co-authors, referees, students
- Example at **demo.otree.org**



Printable payments file



Future work

- Network games
- Measuring reaction time, mouse movements, keystrokes
- Integration of physiological measurements (pupil dilation, eyetracking)
- Continuous-time games
- Real-time chat box

Thank you

chris@otree.org