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

Daniel L. Chen, Martin Schonger, Chris Wickens

chris@otree.org

www.oTree.org

What is oTree?

A platform for lab, web and field experiments

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

Named in homage to z-Tree, c.f. Unix → Linux, Java → JavaScript, C → C++

Background

- We originally created oTree for our own research
- We needed to run experiments on AMT and in the lab
- It was more fun to create a general, good tool, than to fiddle around with new specific code every time

Open Source and Free

- suits the spirit of science
- oTree is a public good, open source allows everyone to contribute
- sustainability

Usage so far

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MaXLab
Magdeburger Experimentallabor für Wirtschaftsforschung

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
DECISION SCIENCE LABORATORY

amazonmechanical turk
beta
Artificial Artificial Intelligence

Usage so far

- RAs/freelancers have programmed 30+ games
- More than 1,000 subjects in lab experiments, plus many more on AMT

Papers using oTree:

"Social Preferences or Sacred Values?
Theory and Evidence of Deontological Motivations"
(Chen/Schonger) nber.org/~dlchen/papers/Deontological.pdf

"A Theory of Experiments:
Invariance of Equilibrium to the Strategy Method and Implications for Social Preferences" (Chen/Schonger) www.n.ethz.ch/~scmartin/theoryofexperiments.pdf

"Ambiguity Aversion With Three Or More Outcomes
- Experimental Evidence And A Formal Observation" (Chen/Schonger)

"Ambiguity Aversion without Asymmetric Information" (Chen/Schonger)

Usage scenarios

No software to install, just open a link on any device. Minimal IT constraints.
Server can be in the cloud or local (you need either Internet or LAN).

1.) Lab
- Can run in any lab (set browsers to kiosk mode)
- Multiple labs simultaneously
- lab/online hybrid approach

2.) Online
- AMT integration
- Large numbers of simultaneous participants (e.g., 500)

3.) Field
- Internet access not necessary
- bring 30 tablets and server laptop

4.) Classroom
- Post a URL/QR code and BYOD and play live in class
- Grading/feedback on homework
- administer surveys etc.
- Use instead of clickers to flip the classroom

Sample games

Simple sample games available at: **demo.otree.org**

- play them from your browser (even right now)
- download the program, modify, use as template
- Programmed by RAs (econ, CS, applied math students) and oDesk -freelancers

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

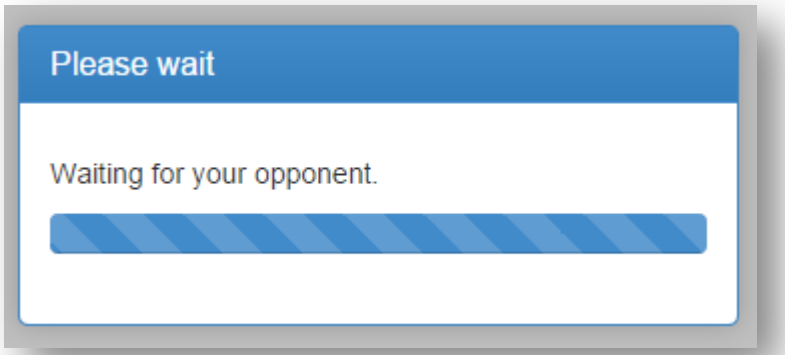
2 x 2 Matrix Game (Symmetric)
2 x 2 Matrix Game (Asymmetric)

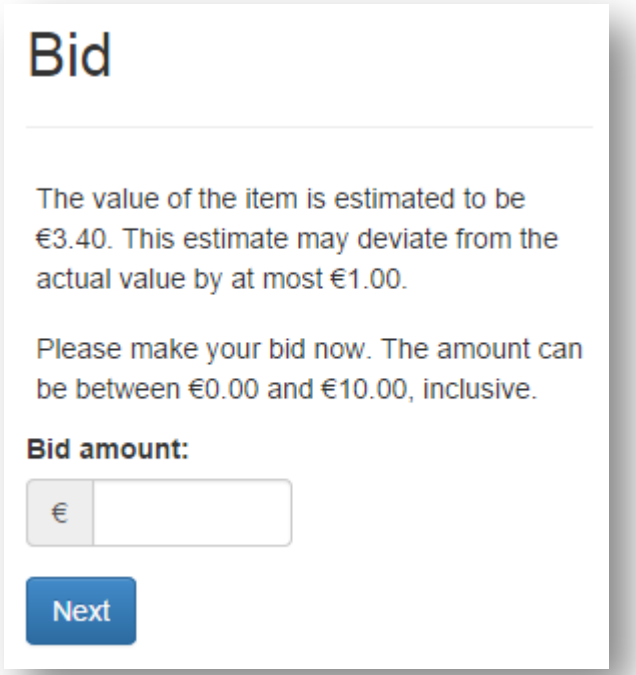
Demo

- You can play live at demo.otree.org

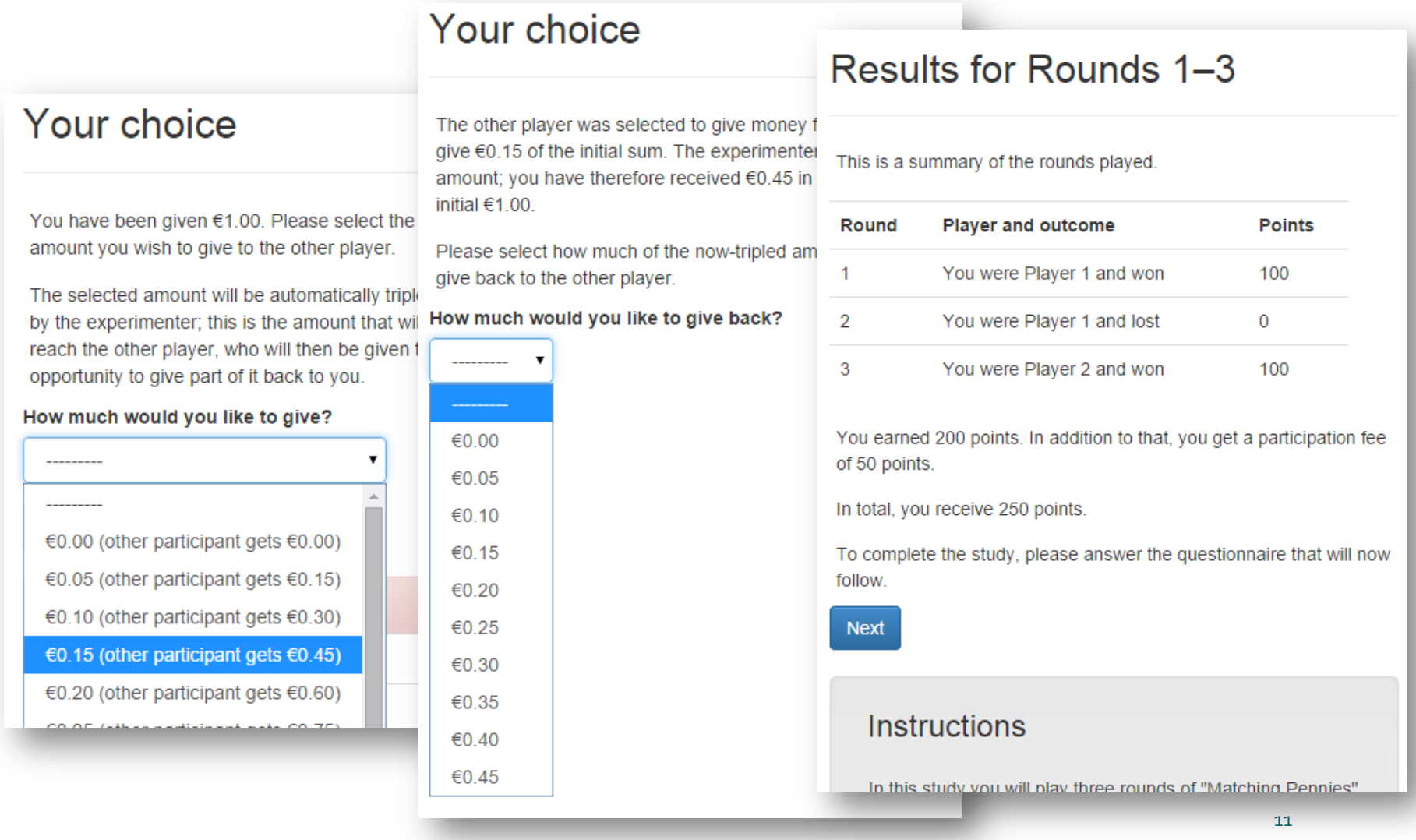
UI

- Auto-scales to any device and screen size





UI


Your choice
You have been given €1.00. Please select the amount you wish to give to the other player.
The selected amount will be automatically triple by the experimenter; this is the amount that wil reach the other player, who will then be given t opportunity to give part of it back to you.
How much would you like to give?

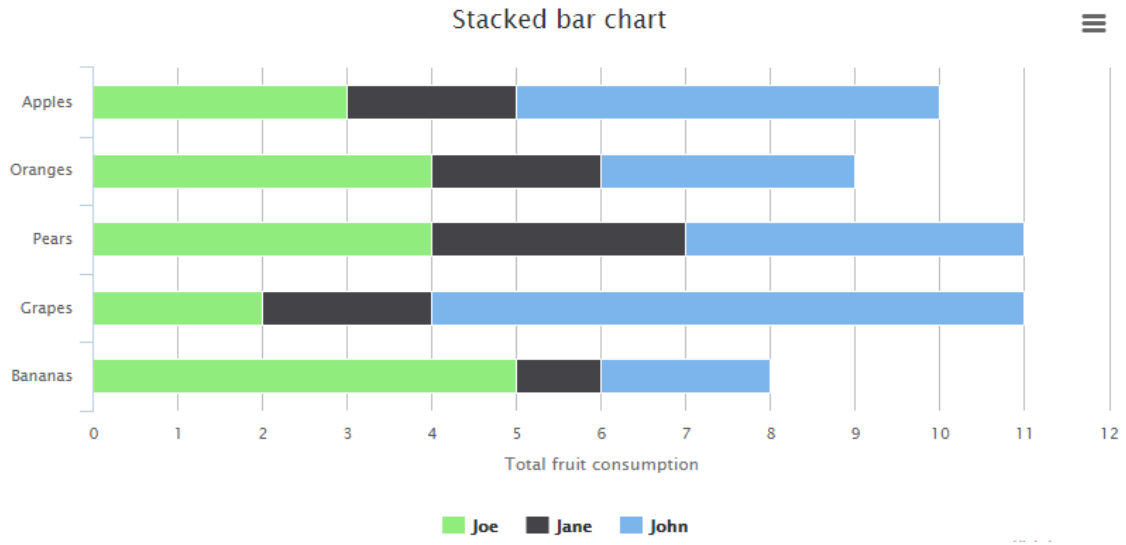
€0.00 (other participant gets €0.00)
€0.05 (other participant gets €0.15)
€0.10 (other participant gets €0.30)
€0.15 (other participant gets €0.45)
€0.20 (other participant gets €0.60)
Your choice
The other player was selected to give money f give €0.15 of the initial sum. The experimente amount; you have therefore received €0.45 in initial €1.00.
Please select how much of the now-tripled am give back to the other player.
How much would you like to give back?

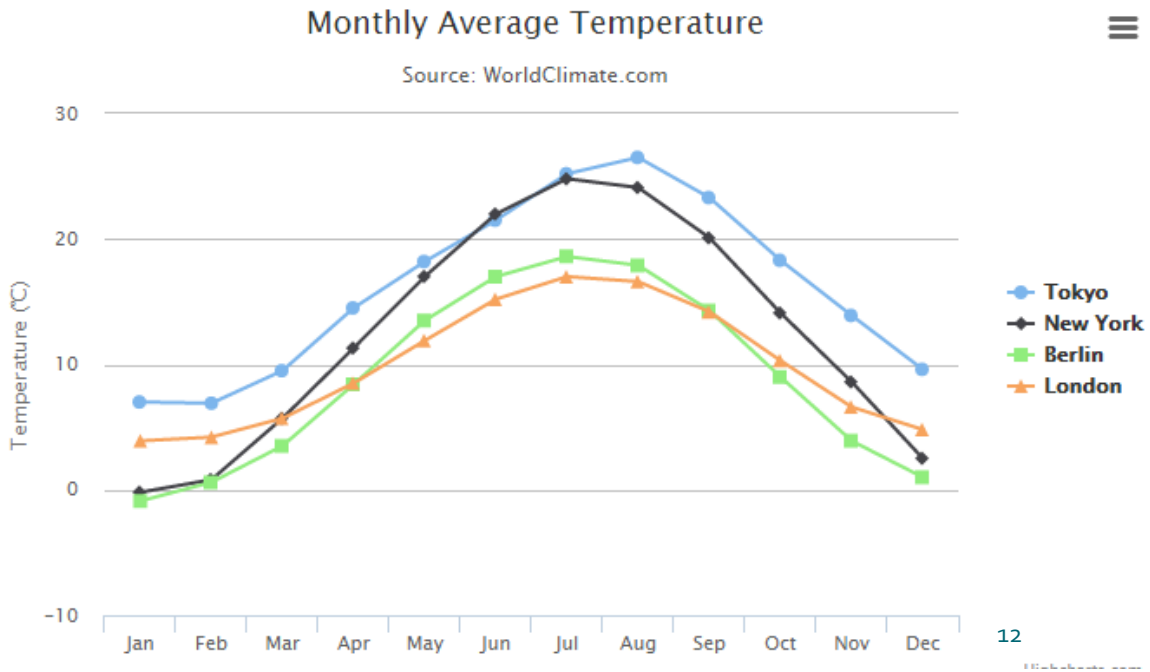
€0.00
€0.05
€0.10
€0.15
€0.20
€0.25
€0.30
€0.35
€0.40
€0.45
Results for Rounds 1–3
This is a summary of the rounds played.
Round
Player and outcome
Points
1
You were Player 1 and won
100
2
You were Player 1 and lost
0
3
You were Player 2 and won
100
You earned 200 points. In addition to that, you get a participation fee of 50 points.
In total, you receive 250 points.
To complete the study, please answer the questionnaire that will now follow.
Next
Instructions
In this study you will play three rounds of "Matching Pennies"

Pre-loaded charts

HIGHCHARTS

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





Programming

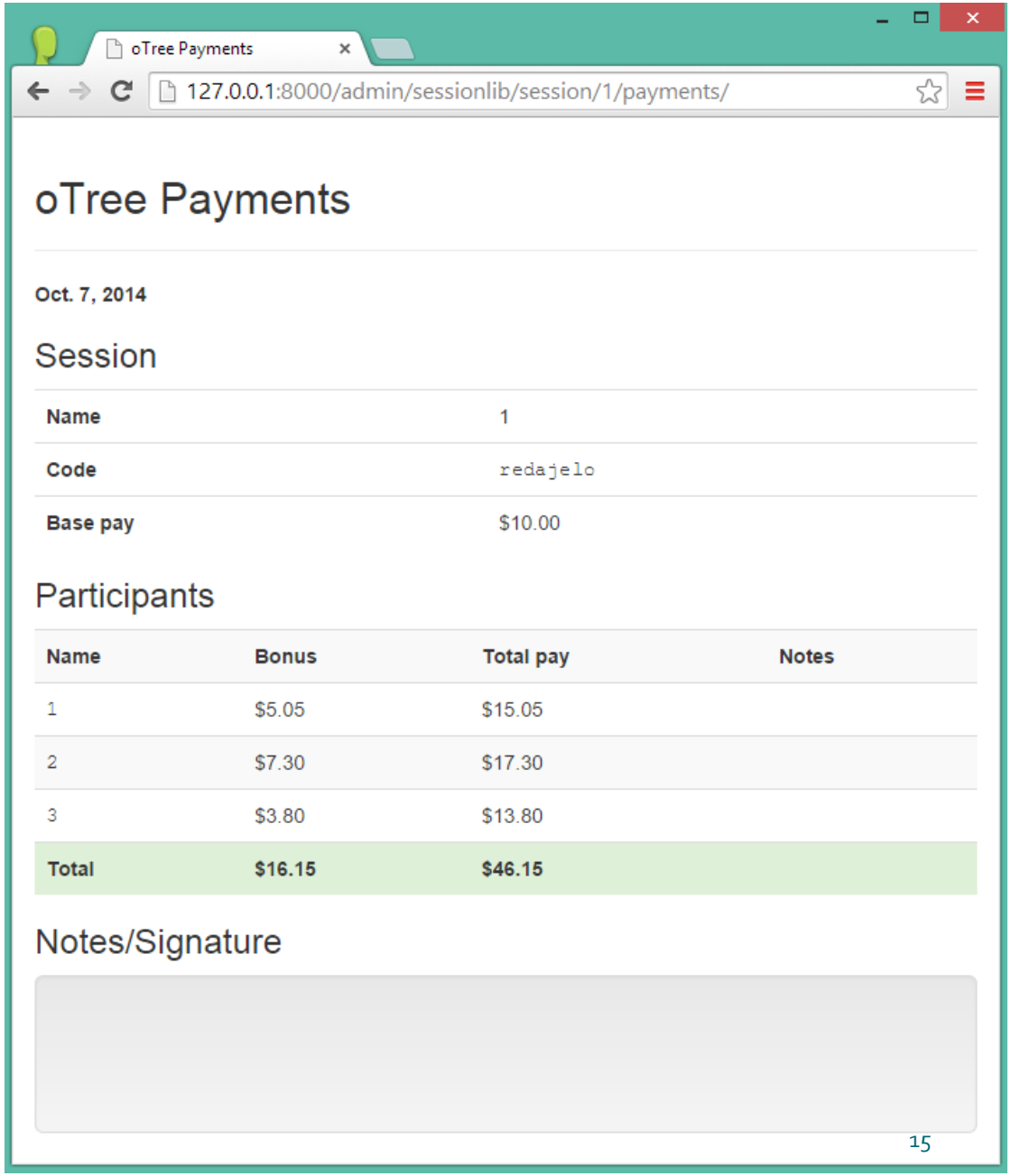
- Python is one of the most common languages in academia, great for beginners, used in many intro CS courses
- Python skills are reusable (web development, scientific computing, statistics, text & natural language processing, scraping data from websites)
- Easy to recruit developers with Python/HTML expertise
  - We have used remote freelancers

Admin Dashboard

- monitor participant progress , detect trouble
- multiple experimenters can open it in their browsers

Total return	Agent fixed pay	Agent return share	Agent work effort	Agent work cost	Contract accepted
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)
(None)	(None)	(None)	(None)	(None)	(None)

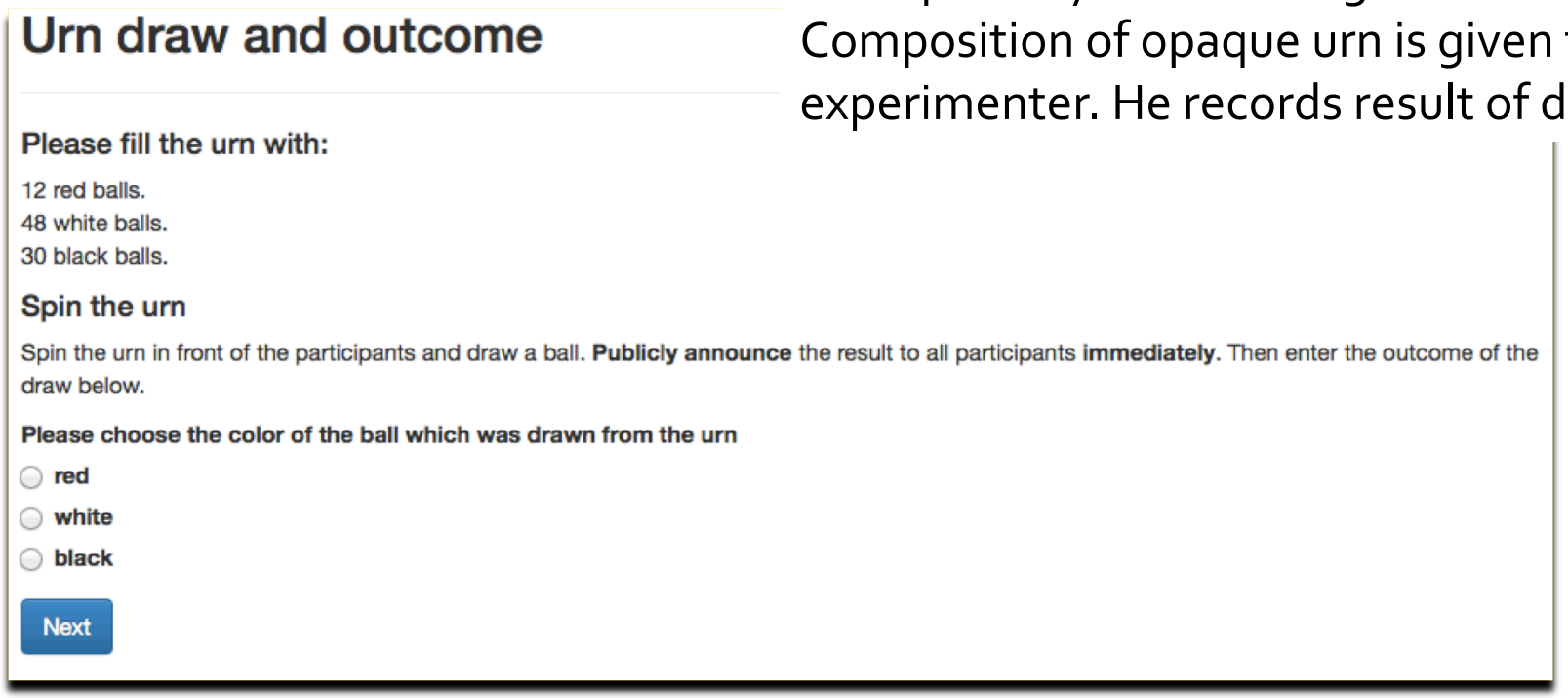
Payments PDF for lab



Study-interface

- some experiments require input from the experimenter.
- experimenter could use tablet for this
- "fake participant"

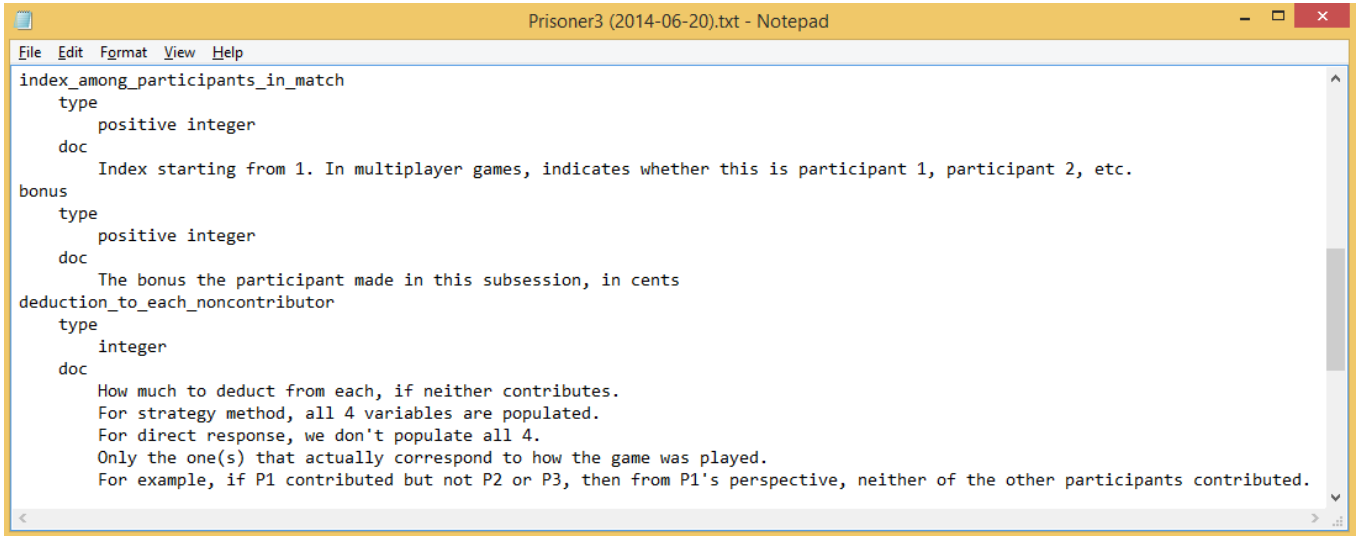
Example: Physical Ellsberg urn is used. Composition of opaque urn is given to experimenter. He records result of draw.



Export data in CSV format

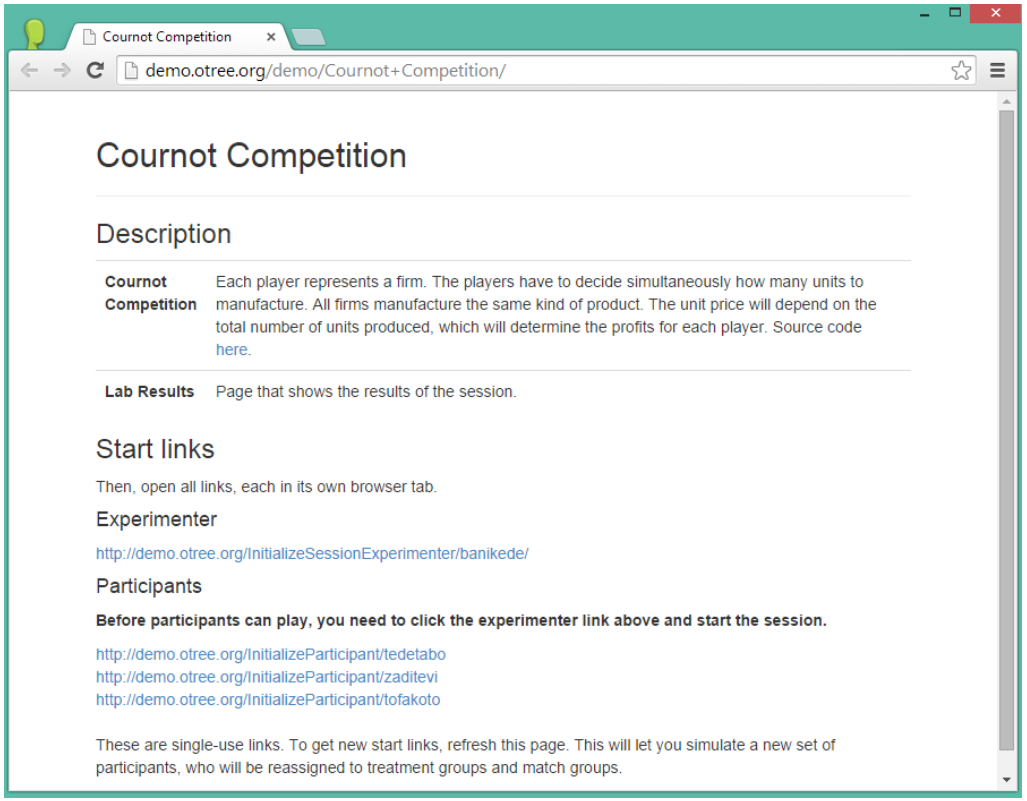
	A	B	C	D	E	F	G	H
1	name	visited	index	bonus	contributed	deduction_noncontrib	deduction_contrib	deduction_1_contrib
2	1	1	3	420	1	-3	-4	-7
3	2	1	3	300	1	-8	-9	-5
4	3	1	1	260	1	-11	-4	-6
5	4	1	2	430	1	2	10	9

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**



Future work

- Out-of-the box/one-click setup
- More comprehensive Wiki

- Continuous-time games
- Millisecond precision
- Measuring reaction time, mouse movements, keystrokes
- Integration of physiological measurements (pupil dilation, eyetracking)

- Adapt for functional illiterates (touch-screen optimized, library of symbols for yes/no, cell phone keypad)

- Classroom module: already can be used for the classroom, but to make it more out-of-the-box

Thank you

Bonus slides
Optional slides

AMT integration

amazonmechanical turk beta
Artificial Artificial Intelligence

AMT—Amazon's online labor market:

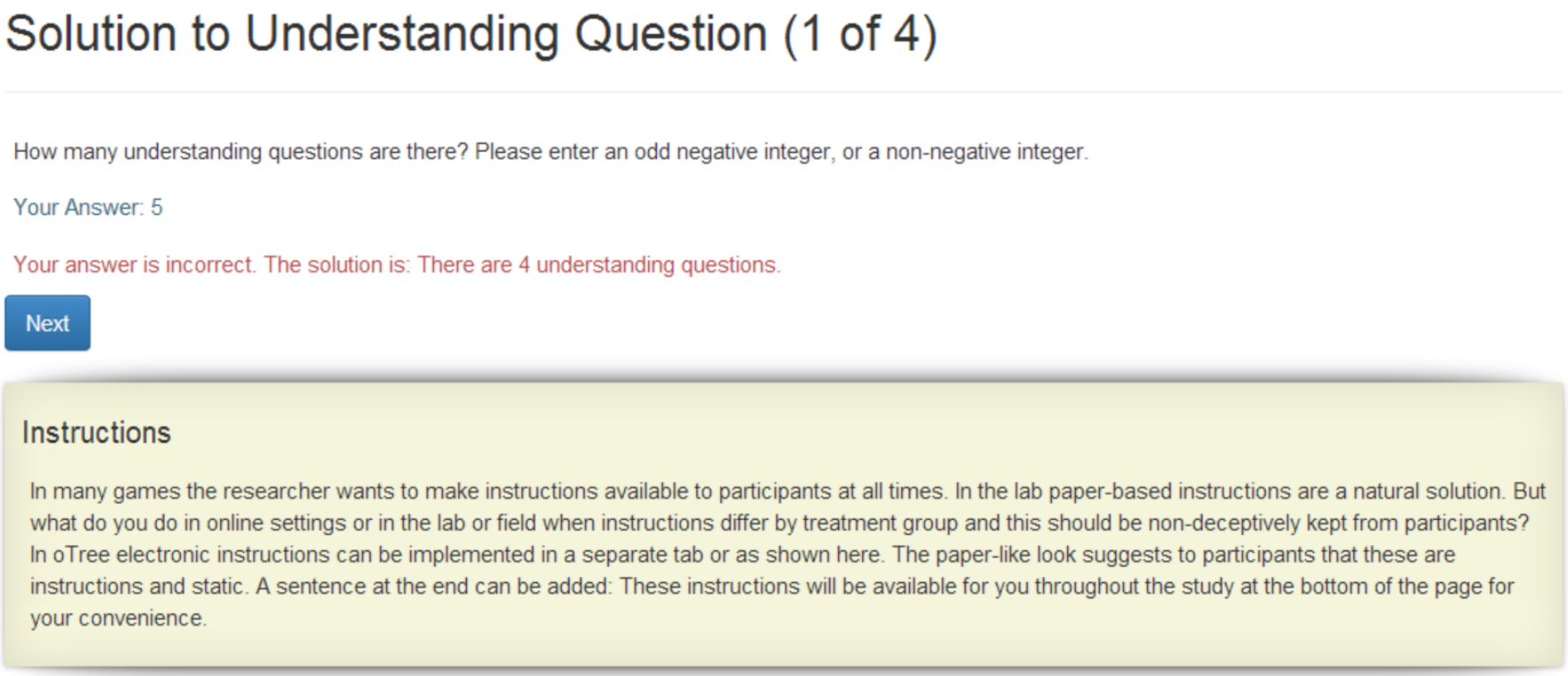
Here is your redemption code:

cegavokiz

Please enter this code on the HIT page.

Paper-like instructions

- Saves handing out paper instructions
- Allows instructions to differ by treatment without participants observing this (non-deceptive obfuscation of the research question to minimize experimenter demand)



Automated testing with bots

- Bots can simulate hundreds of participants (deterministic or Monte Carlo)
- Discover bugs prior to the launch of the experiment
- Detect programming errors as well as design errors (e.g. negative payoffs)
- Saves time by reducing manual testing and re-testing
- All games in our library have bots

```
class ParticipantBot(ParticipantMixin, ptree.test.ParticipantBot):

    def play(self):

        # basic assertions
        assert (self.treatment.amount_allocated == 100)
        assert (self.match.participants_per_match == 2)

        # start game
        self.submit(views.Introduction)

        # if p1, play send page
        if self.participant.index_among_participants_in_match == 1:
            self.play_p1()

        # else p2, play send back page
        else:
            self.play_p2()

        # finally, show results
        self.submit(views.Results)

    def play_p1(self):
        self.submit(views.Send,
                    {"sent_amount":
                     random.choice(self.match.send_choices())})

    def play_p2(self):
        self.submit(views.SendBack, {'sent_back_amount': 200})
```

Group matching

```
groups = [[P, P],
          [P, P],
          [P, P],
          [P, P],
          [P, P]]

for group in groups:
    group.reverse()
```

Groups are represented as a matrix, which you can permute any way you want

Example: P1 and P2 swap roles

Easily reproduce an old experiment

Each data export file notes the version of the code that was used to run the experiment

V
session.git_commit_timestamp
Wed, 04 Jun 2014 22:37:47 +0900

```
$ git checkout [timestamp ID]
$ pip install -r requirements.txt
```

Installs the original version of oTree (and other dependencies)

requirements.txt

```
dj-database-url==0.2.2
dj-static==0.0.5
django-ptree==0.2.112
django-toolbelt==0.0.1
django-vanilla-views==1.0.2
gunicorn==18.0
requests==2.0.1
static==0.4
wsgiref==0.1.2
django-countries==1.5
django-debug-toolbar==0.11.0
```

Next, you can re-run test bots to make sure the app still works the same way

Summary

- Works on **any device** (Windows/Mac, tablets, smartphones)
- **No installation**; open a link and play instantly
- Library of 20 **pre-made games**
- **Modern look & feel**; embed live updating charts, visualizations, or multimedia
- **Demo mode**: put your games online and share with co-authors, referees, readers, students
- Easy programming with **Python**
- **Bots** can simulate participants, save time spent testing manually

Flexible

- Players can progress through pages & rounds individually, in groups, all together, or in any combination thereof
- Randomization and re-matching groups is very easy
- Access to all data from previous rounds