Tutorial #1 - oTree Objects - Web Technologies

Juan Cabral - jbc.develop@gmail.com

Jan, 2018

Tutorial #1: Public goods game.

www.phdcomics.com

- ▶ Open your console (Powershell, terminal, or any flaored pyton console)
- ▶ Open an editor (PyCharm, SublimeText, Kate, Atom…)
- ▶ Follow Me!

Tutorial #1: Public goods game.

This is a three player game where each player is initially endowed with 100 points. Each player individually makes a decision about how many of their points they want to contribute to the group. The combined contributions are multiplied by 2, and then divided evenly three ways and redistributed back to the players.

Tutorial #1: Public goods game.

models.Constants

Open models.py. This file contains the game's data models (player, group, subsession) and constant parameters.

First, let's modify the Constants class to define our constants and parameters – things that are the same for all players in all games.

- ▶ There are 3 players per group. So, change players_per_group to 3. oTree will then automatically divide players into groups of 3.
- ▶ The endowment to each player is 100 points. So, let's define endowment and set it to c(100).
- ▶ Each contribution is multiplied by 2. So let's define multiplier and set it to 2.

Tutorial #1: Public goods game.

models.Player

After the game is played, what data points will we need about each player? It's important to know how much each person contributed. So, we define a field **contribution**, which is a currency

Tutorial #1: Public goods game.

models.Group

What data points are we interested in recording about each group?
We might be interested in knowing the total contributions to the
group, and the individual share returned to each player. So, we
define those 2 fields.
Finally let's define our payoff function. The argument to the
function should be a group whose payoffs should be calculated.

Tutorial #1: Public goods game.

`views.py` and Templates

Now we define our views, which contain the logic for how to display the HTML templates.

Since we have 2 templates, we need 2 Page classes in views.py

1. First let's define Contribute. This page contains a form, so we need to define `form_model` and `form_fields`. Specifically, this form should let you set the **contribution field on the player**.
2. The template contains a brief explanation of the game, and a form field where the player can enter their contribution.

Tutorial #1: Public goods game.

3. Now we define Results. This page doesn't have a form so our class definition can be empty (with the `pass` keyword).
4. Now create the **Results.html** template

Tutorial #1: Public goods game.

views.py and Templates

Consideration

5. After a player makes a contribution, they cannot see the results page right away; they first need to wait for the other players to contribute. You therefore need to add a **WaitPage**. When a player arrives at a wait page, they must wait until all other players in the group have arrived. Then everyone can proceed to the next page.

Tutorial #1: Public goods game.

Finally

- Edit the `views.page_sequence`
- Define the session in **sessions.py**
- Reset the database and run

References

- http://otree.readthedocs.io/en/latest/