Volkswirtschaftliches Seminar
Tobias Werner, MSc.

Heinrich Heine
Universität
Düsseldorf

Winter term 2020/2021

MW86 Seminar

Programming Experiments in oTree: Problem set 2

2 Problem set

2.1 Public goods game

We want to program a repeated public goods game. There are three players in a group. Each player is endowed in each round with 100 tokens. Players have the choice to invest a share of $b_i^t \in [0, 100]$ tokens into a common group project. The sum of all contributions in round t, $P^t = \sum_i^N b_i^t$, will be multiplied by a factor of two and then evenly split among all group members at the end of each round. The payoff for player i in round t is given by $\pi_i^t = 100 - b_i + \frac{2P}{3}$.

2.1.1 Basic version

Program the public goods game described above. The game should be repeated for 10 rounds. After each round, players should receive feedback on their per-round payoff π_i^t and their accumulated profit $\sum_t \pi_i^t$.

2.1.2 Fixed matching with random rematching

We want to extend the existing app and add the following matching scheme:

- Group randomly at the beginning of the experiment and keep this random group composition fixed for the first five rounds.
- After five rounds, match the participants again randomly to new groups and keep this new group compositions fixed for the last five rounds.
- Add an additional page at the beginning of round six that notifies the participants that they will be in a new random group for the final five rounds.

2.1.3 set_group_matrix() matching

Play around with the custom group matching that the set_group_matrix() offers and create a creative matching scheme.
Some ideas that you could do:

- (Perfect) stranger matching.
- Change the group size from three to four in the middle of the experiment.
- Regroup after the first five rounds based on the average contributions, e.g. form new groups with high/low contributors.

2.2 Cournot game with asymmetric players

Consider the Cournot market game, which we discussed in the lecture. Copy the existing app, as we will extend it in this problem.

2.2.1 Roles & Asymmetries

As in the paper by Fischer & Normann (2018), we would like to have two player types that differ in their efficiency to produce the homogenous good. The production costs are described by $C_i(q_i) = \theta_i q_i$ with $i = L, H$. Hence, a player with type L has lower costs per unit compared to player type H. In the experiment, we want to choose the parametrization of $\theta_L = 13$ and $\theta_H = 25$ as in the original paper. Adjust the entire app to account for this asymmetry.
Some things to keep in mind:

- Assign the roles in the Constants as described in the lecture.
- Adjust the cost function.
- Change all pages/templates such that the correct information is shown to each participant conditional on the role he/she has in the experiment.

2.2.2 Switch roles

Next, we want to extend the app further to include a within-subject design. Players should play the game for five rounds and then switch their roles for the remainder of the game. Hence, the participant which has been type L in the first five rounds becomes type H and vice versa. The groups should be fixed for all ten rounds. Thus, the group should remain the same for the entire game and only the roles should be switched in the middle of the experiment. Also, add a page that notifies the participants when they switch their roles. Adjust the other pages/templates if needed.