**1 Experiment Overview**

This experiment program is designed to replicate the public goods experiment of Hauser et al. (2019). The public goods game (PPG) has been extensively studied to investigate human cooperation and free-riding behaviour (Carpenter, 2004; Fehr & Gächter, 2000; Page et al., 2005; Shirsendu Podder et al., 2021). In the PPG, several players could choose how many units they would like to contribute to the common pool (i.e. public goods). Their contribution will be multiplied by a preset, certain factor before added up (multiplier). Then all the players would share the sum of multiplied contribution equally regardless of how much they contributed. The payoff of each player would thus be the sum of the share per person and the units they did not contribute. Therefore, we could understand human cooperation by analyzing their actions in the PPG.

Most previous studies have assumed that players have equal initial units and that their contributions are multiplied by a common factor. However, Hauser et al. (2019) proposed that people's endowments and productivity may be different in our real life. Therefore, PPG should take the inequality into consideration. To replicate their experiments, this program allows experimenters not only to conduct a PPG experiment with multiple participants, but also to customize the initial units and multiplier for each participant.

**2 Description of procedure**

Each group consists of a total of *n* participants. After all *n* participants have arrived at the computer lab, each of them is assigned a computer (client computer). The experimenters will run the python program on a server computer first, which communicates with client computers, calculate their payoffs in each trial, and records their responses and demographics. Then the experimenters help participants run the experiment on the client computers.

When the experiment begins, participants will first be shown a consent form, only after signing which could they continue the experiment. Then they will be asked report their demographic information including age, gender, education level and race. Next, they need to read through the instructions. After all the participants have read the instructions and clicked the ‘NextPage’ button, their computers will be connected with the server computer and the trials will begin.

In each trial, they are asked to contribute a certain amount of units to the common pool (Fig. 1). After all the participants have contributed, their responses will be sent to the server computer to calculate the group contribution and the share per person. Then the contribution of each player and the total payoff will be displayed on the client computers (Fig. 2). By clicking the ‘continue’ button, they could enter a new trial.

Each group of participants will finish at least **X** trials. When they have completed all the trials, they will be thanked for participating in this experiment. Each participant will receive a basic participant fee £x and a bonus based on the numbers of units that they accumulated over all trials.

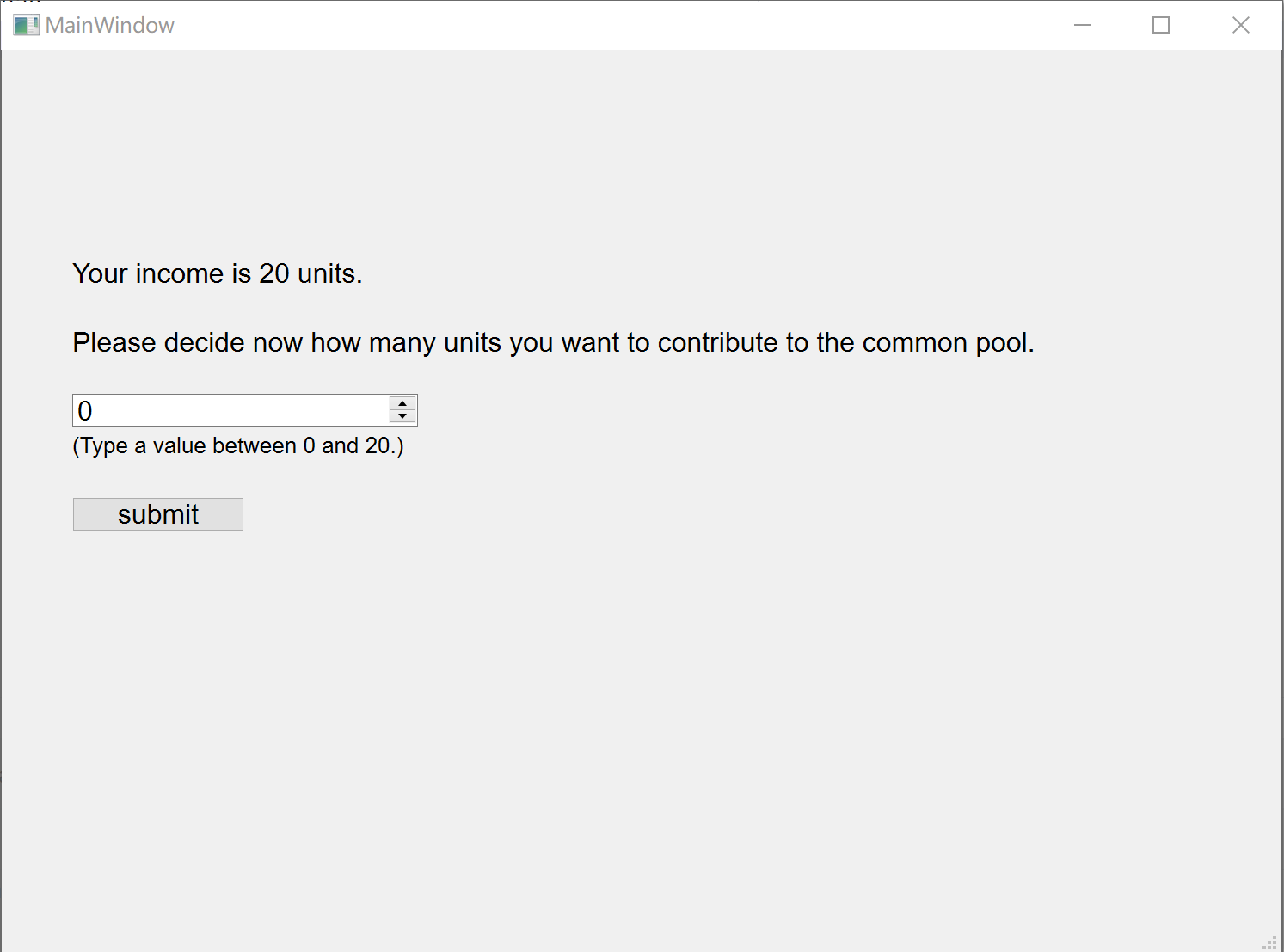


Fig.1 Contribution Page

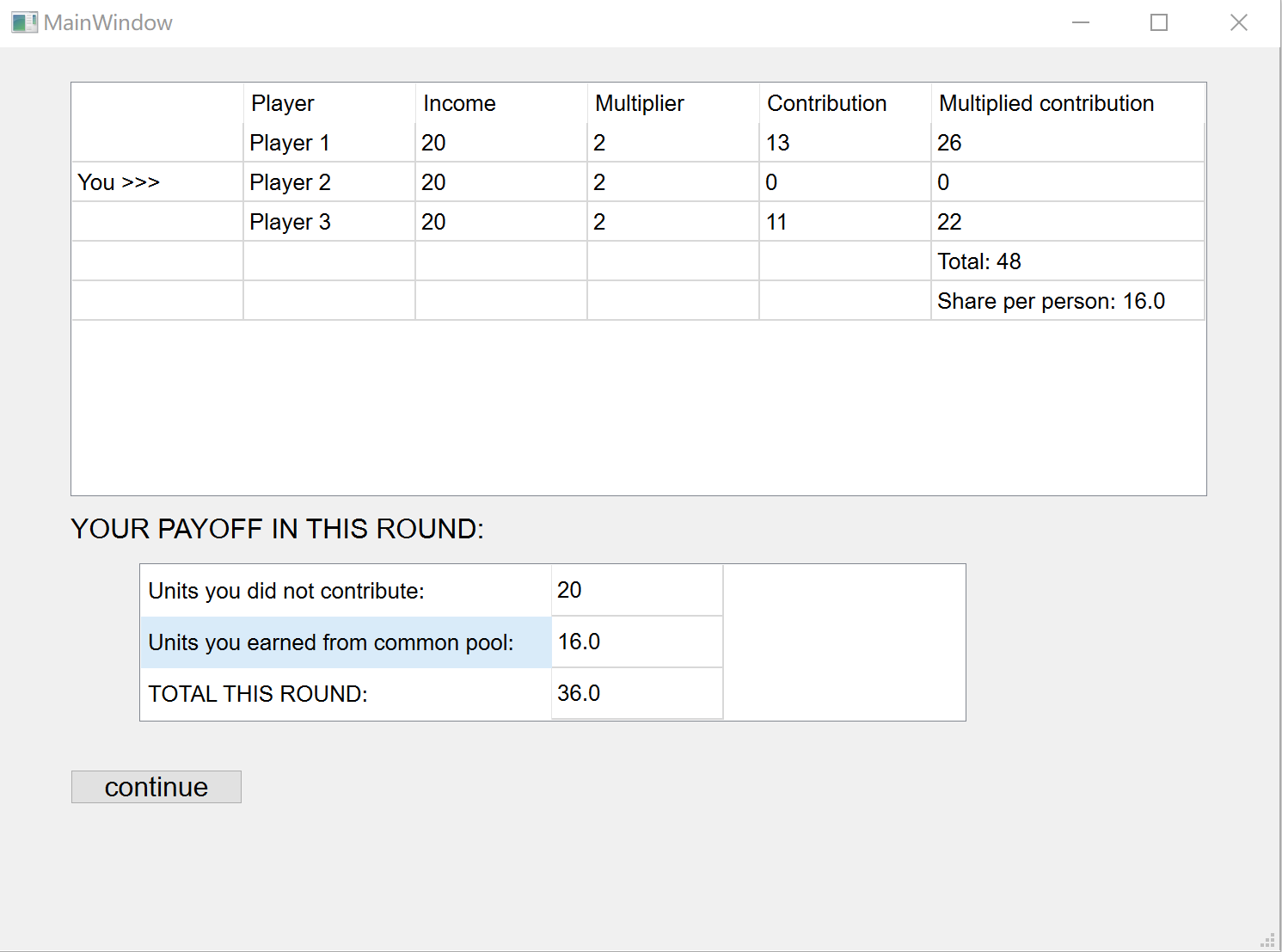


Fig. 2 Results Page

**3 Experimenter’s manual**

**To run a PPG experiment with *n* participants:**

1. Copy the files from the ‘server’ folder to the server computer, and copy the files from the ‘client’ folder to *n* client computers. Make sure all the computers have installed the latest version of python and relevant libraries on which the experiment depends. The client computers should have installed Qt Designer.

2. You could modify some default experiment settings in the 'server\_setting.py' module, such as the number of players per group, the number of trials, the paths for data files, etc. Participant settings, such as their initial units, multiplier, UI file path and the IP address of the server computer, could be modified in the ‘client\_setting.py’ modules on each client computer.

3. Run the ‘get\_hostname.py’ in the ‘server’ folder to get the IP address of the server computer. Then set this IP address in each ‘client\_setting.py’ to connect to the server computer.

4. Run the ‘server\_main.py’ program to start the server program and then help participants run the ‘client\_main.py’ to begin the experiment.

**To test and mark this experiment locally:**

Usually, running a PPG experiment with *n* participants will require *n+1* computers. But you could also test this experiment locally with only one computer. To achieve this, you do not need to copy the files to different computers. Instead:

1. Copy the ‘client\_main.py’ file in the ‘client’ folder for *n* times (it depends on how many players you want to simulate) in the same folder.

2. Set the number of players in ‘server\_setting.py’ and set the initial units and multiplier for each player in the ‘client\_main.py’ when instantiating a Client Object.

3. Run ‘server\_main.py’ and each ‘client\_main’ file.

4. Fill in information and decide how many units to contribute in *n* UI windows, just like participants.

**Output data file:**

The path of output data depends on your settings. The default path for output data is ‘server/data.csv’. Consider a PPG experiment with *n* participants and *m* trials. Each row records the information of each participant in one trial. Every *n* rows record the information of all participants in one trial. Every *m*\**n* rows record the information of all trials in one experiment session. There are 17 variables in this CSV file:

1. Session: experiment session. After you run a new experiment, the session will be plus one.

2. Trial: the sequence of trial in this session.

3. Player number: number of players in this session

4. Player name: name of each player

5. Income: income of each player in each trial

6. Multiplier: multiplier of each player

7. Contribution: contribution of each player in each trial

8. Multiplied contribution: multiplied contribution of each player in each trial. The product of Multiplier and Contribution.

9. Remains: units not contributed of each player in each trial. The different between Income and Contribution

10. Group contribution: total contribution of each group.

11. Group multiplied contribution: total multiplied contribution of each group.

12. Share per person: share for each player.

13. Payoff: payoff of each player in each trial. The sum of Remains and share per person

14. Age: age of each player

15. Gender: gender of each player

16. Education: education level of each player

17: Race: race of each player

**4 Program highlights**

**1 Communication between devices**

This program used ‘socket’ library to realize the communication between multi devices.

**2 Multi-Thread**

This program used ‘threading’ library to realize multi threads because the ‘app.exec()’ function blocks the main thread.

**3 Pandas & dataframe**

This program used ‘pandas’ library and dataframe to record experiment information and convert the dataframe to CSV files.

**4 Replication and extensible**

This program not only replicated the original experiment, but also became more extensible. It allows for more than two participants to take part in the experiment simultaneously (while the original experiment had only two participants for each session).

**5 Future Improvements**

Due to a limit of time, I have not created a program that I consider perfect. Therefore, I have uploaded it to my GitHub and plan to make some improvements in the following aspects in the future.

**1 Waiting**

When one player has made their decision, they have to wait for other players. While waiting, the UI will be not interactable. If the player continuously clicks on the UI interface, the program may experience a crash.

**2 Instructions and comprehension questions**

In the original study, there were more detailed instructions. The players were required to answer several comprehension questions. Only after correctly answering these questions, could they enter the main part of the experiment.

**3 Reaction to dropout**

Participants may drop out the experiment sometimes. If that happens, the program should end the experiments of other players and record the experiment information.

**4 Readability, organization and comments**

The codes should be more readable and organized. More comments should be made in the codes.

**References**

Carpenter, J. P. (2004). Punishing Free-Riders: How Group Size Affects Mutual Monitoring and the Provision of Public Goods. SSRN Electronic Journal. <https://doi.org/10.2139/ssrn.285001>

Fehr, E., & Gächter, S. (2000). Cooperation and Punishment in Public Goods Experiments. American Economic Review, 90(4), 980–994. <https://doi.org/10.1257/aer.90.4.980>

Hauser, O. P., Hilbe, C., Chatterjee, K., & Nowak, M. A. (2019). Social dilemmas among unequals. Nature, 572(7770), 524–527. <https://doi.org/10.1038/s41586-019-1488-5>

Page, T., Putterman, L., & Unel, B. (2005). Voluntary Association in Public Goods Experiments: Reciprocity, Mimicry and Efficiency. The Economic Journal, 115(506), 1032–1053. <https://doi.org/10.1111/j.1468-0297.2005.01031.x>

Shirsendu Podder, Righi, S., & Pancotto, F. (2021). Reputation and punishment sustain cooperation in the optional public goods game. Philosophical Transactions of the Royal Society B, 376(1838). <https://doi.org/10.1098/rstb.2020.0293>