Workshop: Section 5 - Basic oTree Experiment (Public Good Game)

Philipp Chapkovski
University of Bonn
chapkovski@uni-bonn.de

September 11th - 12th, 2023

Basic oTree Experiment: Public Good Game - Overall Game Flow

1. **Contribution Decision Page**: Each player decides how much to contribute to the public good from their initial endowment.
2. **Wait Page**: All players wait until everyone in the group has made their decision.
   - `WaitPage`: When all players arrive, the total contributions are calculated, and payoffs are set.
3. **Results Page**: Players are shown their final payoff based on their contribution and the total contributions of the group.

PGG: Variables and Parameters

- **Player-Level Variable**:
  - `contribution`: Amount each player decides to contribute.
- **Group-Level Variables**:
  - `total_contributions`: Sum of all player contributions in the group.
  - `individual_share`: Share of the total pool (after multiplication) for each player.
- **Constants / Fixed Parameters**:
  - `multiplier`: The factor by which the total contributions are multiplied.
  - `endowment`: Initial amount each player has to contribute.

Basic oTree Experiment: Public Good Game

Project and App Setup Reminder

1. **Create a New oTree Project**:
   - Run `otree startproject pggproject`
   - **Important**: Say 'No' when oTree offers to add sample games!
2. **Change Directory to Project Folder**:
   - Run `cd pggproject`
3. **Create a New App Within the Project**:
   - Run `otree startapp pgg`
4. **Register the App in `settings.py`**:
   - Add `'pgg'` to the `SESSION_CONFIGS` list in `settings.py`

Hardcoded Parameters and Session Configuration

Constants in __init__.py

- PLAYERS_PER_GROUP = 3
- ENDOWMENT = 20
- MULTIPLIER = 2

settings.py

```
SESSION_CONFIGS = [
    dict(
        name='public_goods',
        app_sequence=['pgg'],
        num_demo_participants=3,
    ),
]
```

Models: Data Structure

- **Player-Level Variable**: `contribution`

```
contribution = models.IntegerField()
```

- **Group-Level Variables**: `total_contributions`, `individual_share`

```
total_contributions = models.IntegerField()
individual_share = models.IntegerField()
```

Why Group-Level Variables? To store data that is common to a group of players, such as total contributions in a public goods game, we define variables at the group level.

Page Structure with Stubs

Decision Page: *Gather contributions from each player*

```
class Decision(Page):
    pass
```

ResultsWaitPage: *Wait for all players to make their decisions*

```
class ResultsWaitPage(WaitPage):
    pass
```

Results Page: *Show the outcomes based on contributions*

```
class Results(Page):
    pass
```

Reminder: Three Components for oTree Page

To show a page in oTree, you need three components:

1. **Class Definition**: Define a Python class corresponding to the page.
   - Example: `class Decision(Page): pass`
2. **Page Sequence**: Add this class to `page_sequence` in `__init__.py`.
3. **HTML File**: Create an HTML file with the corresponding name (e.g., `Decision.html`).
   - Tip: Copy-paste the pre-defined `MyPage.html` and rename it to `Decision.html`.

Getting Player's Contribution: Decision Page

In pages.py:

```
class Decision(Page):
    form_model = 'player'
    form_fields = ['contribution']
```

In Decision.html:

```
    <h1>Make your contribution</h1>
    {{ formfields }}
```

- `form_model = 'player'`: Specifies that we are using the `Player` model defined in `models.py`.
- `form_fields = ['contribution']`: The field(s) we want to collect from the player.
- `{{ formfields }}` in HTML: Renders the form fields in the template.

Setting Payoffs (ResultsWaitPage)

In `ResultsWaitPage`, we define a method `after_all_players_arrive` to calculate payoffs.

```
def set_payoffs(group: Group):
    num_players = C.PLAYERS_PER_GROUP
    group.total_contribution = sum([p.contribution for p in group.get_players()])
    group.individual_share = (group.total_contribution * C.MULTIPLIER) / num_player
    for p in group.get_players():
        p.payoff = p.endowment - p.contribution + group.individual_share



class ResultsWaitPage(WaitPage):
    after_all_players_arrive = 'set_payoffs'
```

Showing Results

To display results, you will use variables in your HTML file as follows:

- **Player's Endowment**: `{{ C.ENDOWMENT }}`
- **Player's Contribution**: `{{ player.contribution }}`
- **Other's Contributions**:

```
{{for p in player.get_others_in_group }}
  {{ p.contribution }}
{{ endfor }}
```

- **Group Total Contribution**: `{{ group.total_contribution }}`
- **Final Payoff**: `{{ player.payoff }}`