Behavorial Experiments with oTree and Python

Moinul Islam

9/15/2022

Table of contents

Preface

oTree is an open-source and online software for implementing interactive experiments in the laboratory, online, the field or combinations thereof. The basic experimental setup in oTree consists of (i) an experiment written within oTree, (ii) a server computer, which can be a cloud server or a local laptop and (iii) subjects' devices with a web browser. oTree creates an experimental session on the server, as well as links for all the participants and the experimenter.

oTree is a framework based on Python that lets you build:

- Multiplayer strategy games, like the prisoner's dilemma, public goods game, and auctions
- Controlled behavioral experiments in economics, psychology, and related fields
- Surveys and quizzes

This tutorial will use Python to develop the experiment by using oTree software. For Python code generation, we will use **Pycharm Professional** software.

1 Installation and environments

1.1 Install Python

- Open the python website and install python3 for your OS
- Choose the most recent version of Python

1.2 Install Pycharm

- Open Pycharm website. There are different version of Pycharm.
- I recommend to install the Pycharm professional because it has better syntax highlighting. To have the professional version for academic use, you need to have an academic email address.
- Alternatively you can install the community version of Pycharm which is free of charge.

1.3 Run Pycharm

- Create a new Pycharm project. Choose pure python. Give a name of the project. Pycharm will create the project for you.

1.4 Install oTree

Install oTree in your computer by using the terminal in Pycharm.

```
pip install otree
```

1.4.1 Upgrading/reinstalling oTree

```
pip install -U otree
```

- Recommend upgrading every couple of weeks.

1.5 oTree setup

- From your command prompt, create your **IGG** project

```
otree startproject IGG
```

- Move into the folder you just created

```
cd IGG
```

- Run the server

```
otree devserver
```

- Open your browser to http://localhost:8000/. You should see the oTree demo site.
- To stop the server, press **Control+C** at your command line.
- To create a new app, run

```
otree startapp app_name
```

- Session configs are defined in **settings.py**

2 Structure of an oTree project

2.1 App

- One app can be an experiment or a part of experiment. A project is collection of all different types of App.
- If your experiment consist of a prisoner's dilemma game and a survey, then you need to create two apps, one app for prisoners dilemma and another app is for survey.
- Later you can combine these two apps and run one session.
- App is like one experiment and project is like a container of different experiments.
- How you structure the projects is up to you. You can choose the sequence.
- Now if you see the structure of the examples, you can see that there are various different types of folder.
- The following two folders are known as global settings as they are same level of the project

  _Static

- Images, sounds, videos

  _Template

- Design how web page look like. If you want a specific style of your webpage, you can include a template

  settings.py

- It is also global option. From this setting file, you can chage various global options, for instance Your currency, language, interface
- In this **setting.py** file you will find the session_configs where you can setup the sequence of apps for your project.

2.2 How to create a new app.

- Navigate your working folder by using **cd**
- For instance you are using the project **IGG**. Then use

```
cd IGG
```

To navigate in this folder

- Now you can create a new app from the terminal.

If you want to create a app, name “SVO”

```
otree startapp SVO
```

- You will notice that a new app named “SVO” is listed on your Pycharm
- If you open the app, you can see that it contains three different types of files:
  - _init.py
  - MyPage.html
  - Results.html

If you open the _init.py you can find several different class.

You can edit it based on your game. Two most important classes are:

- class (Constants)
- class (Player)

Everything we want to analyze and store in data should be stored in Player.

3 App in oTree

3.1 App

1. To create an application named **game_app** move to the oTree folder

- cd oTree

2. Create the application

- otree startapp game_app

3. Move to the folder **game_app**
4. In this folder, you will find the following files as default

- models.py
- pages.py
- tests.py

5. In this folder, you will also find a subfolder

- templates/game_app
  - Mypage.html
  - Results.html

3.2 Models.py

A model is basically a database. Here we define the structure of the data. For instance, in a three data models. This is python **class**

- Subsession
- Group
- Player

1. class Subsession(BaseSubsession):
    - pass
2. class Group(BaseGroup):
    - pass
3. class Player(BasePlayer):
    - pass

3.3 Pages.py

- Pages that the participants see are defined in pages.py

  Logic for how to display the HTML templates

  when, how, and what to display
- page_sequence gives the order of pages

  If there are multiple rounds the sequence is repeated

For instance,

1. class MyPage(Page):

   pass
2. class ResultsWaitPage(WaitPage):

   def after_all_players_arrive(self):

   pass
3. class Results(Page):

   pass

- page_sequence = [MyPage, ResultsWaitPage, Results]

4 Create project by oTree hub

5 Chapter 5

6 Chapter 6

7 Chapter 7

8 Chapter 8

9 Chapter 9

10 Chapter 10

References