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

- Before installing oTree, it is required to have Python installed in your environment to run oTree.
- Open the python website and install **Python3** for your OS from this website:
- https://www.python.org/
- Choose the most recent version of Python

1.2 Install pip

If your Python environment does not have pip installed, you can install pip by following this instructions

- https://pip.pypa.io/en/stable/installation/

1.3 Install Pycharm

- PyCharm is a famous Python IDE (Integrated Development Environment). We recommend you to use PyCharm if you want to further customize your oTree app by programming. Please go to this website to download the installation package. For oTree development, a free community version should be sufficient.
- https://www.jetbrains.com/help/pycharm/installation-guide.html
- 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.

1.4 Run Pycharm

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

1.5 Install oTree

- oTree can be installed through pip. Open the terminal (Mac/Linux) or command prompt (Windows PowerShell) and type:

```
pip3 install otree
```

- You can install oTree in your computer by using the terminal in Pycharm as well.

```
pip3 install otree
```

1.5.1 Upgrading/reinstalling oTree

```
pip3 install -U otree
```

- Recommend upgrading every couple of weeks.

1.6 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 change 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 don't see a models.py in each folder, that means you are using the new no-self format.
- 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 Customize oTree hub output

4.1 Create project by oTree hub

- If you are a new, it is recommended to use oTree Studio to create and edit the project instead of hard coding. Please click this link to register an account then login.
- https://www.otreehub.com/accounts/login/?next=/studio/
- Click the "+ Project" button to create a new project.
- Then you're able to visit the project configuration page where you can config the metadata for your project.
- Go to the "download" and click the "Download .otreezip" button to download the project.

4.2 PyCharm to customize the project

- After creating and customizing your project on oTree Studio, you can also further customize your project using PyCharm or other IDE. First, you need to unpack your downloaded **.otreezip** file. To do this, run:

```
otree unzip xx.otreezip
```

- The command will produce a folder with an identical name. Use PyCharm to open that folder.
- Then, you can add python code to further customize the project. After adding your code, run the following command to re-pack the project into the .otreezip file:

```
otree zip
```

4.3 Host the App locally

- Open a terminal/Command Prompt on your computer and change the working directory to where you store your previously downloaded project file. Execute the following command to start an oTree test server:

```
otree zipserver
```

- Then, you can open http://localhost:8000/ in your browser to test your project out.

5 On Heroku from oTree

Heroku is a website hosting service where you can easily host your oTree application and open access to the public.

6 Survey with oTree

7 Experiment 1 with oTree

8 Experiment 2 with oTree

9 Experiment 3 with oTree

10 Conclusion

References