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 Python website.
- Choose the most recent version of Python

1.2 Install pip

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

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 Pycharm website to download the installation package.
- 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.

1.3.1 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 GitHub

Make sure you have some kind of version control, I use GitHub. If you crash your code, you can always go back to a previous version of the code.

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.

1.5.2 Regarding the installation of oTree in Linux

- You might need to do the proxy server settings
- In the Pycharm, you can open the terminal and install oTree as below:

```
set https_proxy address:password
pip install otree
```

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 The building blocks of oTree

Once you have installed oTree and created an app you are ready to build your own experiment. If you examine an experiment, you will see that from a programming-perspective an experiment consists of some things. These can be seen as the four building blocks of oTree.

1. **Templates**

The first building block are templates: every page in an experiment is a template with a certain lay-out and text. Templates are .html files in which you can use HTML, CSS, Javascript, and Django.

2. **Models.py**

In **Models.py** you define the variables that will be stored in your data-output. For instance, if you want to ask the age of a participant you create a variable:

```
age = models.IntegerField(label="What is your age?")
```

Here, a model is a database table and a field is one column within that table. In this case, an *IntegerField* is used, indicating that only integer numbers are valid. The label determines the text that is shown to participants next to the variable. There are six *StringField* that you can use:

- *IntegerField* for integer numbers
- *FloatField* for numbers that may include decimals
- *StringField* for strings of text
- *LongStringField* for long strings of text
- *BooleanField* for True/False values
- *CurrencyField* for currency amounts

3. **Pages.py**

In *Pages.py* you define every page that is shown to participants and specify which variables are allocated to that page. For instance:

```
class Instructions(Page):
    form_model = 'player'
    form_fields = ['Comprehension1', 'Comprehension2']
```

Here, Instructions is the name of the page, and form_model refers to the level at which the variables should be stored. form_fields includes a list of variables that can be elicited on the page. In this case, we include two comprehension checks that are further defined in Models.py. Next to defining and specifying variables for every page, Pages.py is also used to determine the page sequence.

```
page_sequence = [Instructions, Main_Page, Results]
```

Moreover, there are a battery of functions that can be used to specify conditions for each page. For example, the is_displayed() function determines conditions to which participants the page is displayed.

4. **Settings.py**

The last building block of oTree is *Settings.py*. Here you define some overarching settings of your oTree project, such as the language used, the currency used, your admin username and password, how experimental currency translates to real-world currency, and the apps that are available. For the latter, if you have an app called attribute_substitution. You can make it available in your oTree server by typing:

```
SESSION_CONFIGS = [
    dict(
       name='attribute_substitution',
       display_name="Attribute Substitution Game",
       num_demo_participants=3,
       app_sequence=['attribute_substitution']
    ),
]
```

Here, the name and display name are whatever you prefer (for name you cannot use spaces). The number of demo participants is the number of slots available to play a demo version of the experiment. The most important thing here is the app_sequence. In the app_sequence you define the sequence of apps that is displayed to participants. Suppose that you want to combine the attribute_substitution app with a public goods game or a survey, you can add, public_goods or survey, respectively.

4 Pycharm to customize oTree studio project

oTree Studio is a point-and-click interface for building oTree apps. It is free to use for small/medium sized projects.

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 oTree Hub account then login.
- Click the "+ Project" button to create a new project.
- Then you're able to visit the project configuration page where you can config the meta-data 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.

- Go to your Heroku dashboard and click “Create new app”.

6 Basic survey by using oTree

This is a basic questionnaire with page and question randomization that can be executed after or before other applications in oTree, an open-source Python framework for experimental and survey research. It includes five pages:

- `Introduction.html`
- `Survey1.html`
- `Survey2.html`
- `Survey3.html`
- `Demographics.html`

The questionnaire always starts with Introduction.html, and always ends with a Demographics.html.Between these two pages, the application randomizes other pages included under templates as "Survey.html". The order of items and questions on every page are also randomized. The questionnaire also includes a progress bar.

7 Experiment 1 with oTree

8 Experiment 2 with oTree

9 Experiment 3 with oTree

10 Conclusion

References