有斐閣ストゥディア

『実験から始める経済学の第一歩』オンライン付録

花木　伸行 ・ 島田　夏美

株式会社　有斐閣
2023 年 12 月 14 日　初版
ISBN 978-4-641-15117-8

付録 A

本書で用いる実験を実施するための手順

ここでは、oTree(Chen et al., 2016) と自分の PC への環境構築について説明していきます。これらは本書の執筆時のものであることに留意して、最新の情報は各ホームページなどで確認するようにしてください。

オンラインサーバーである Heroku [*1]、Heroku との連携が可能な GitHub[*2]についても述べます。この本で紹介した実験は付録 B の GitHub の URL 上で管理されています。あなたが Heroku に登録して、GitHub と連携すれば、実験をすぐに始めることが可能になります。より端的にいえば、Python や oTree のインストール、プログラミングもせず、この本で提供されている実験をすぐに実施することが可能です。その場合は A.3 から参照してください。各種実験は、付録 B を参照してください。

[*1] 2023 年現在、Salesforce, Inc. が提供しています。URL は `https://jp.heroku.com/` です。

[*2] 2023 年現在、GitHub, Inc. が提供しています。URL は `https://github.co.jp/` です。

A.1 oTree とは

oTree とは、研究論文 (Chen et al., 2016) で発表されているオープンソースソフトウェアです。ソースコード・マニュアルも公開されており、誰でも使うことができます（日本語版のマニュアルは `https://otree.readthedocs.io/ja/latest/index.html`[*3]）。2016 年に登場し、オンラインで、参加者が実験室に来なくても安定したインターネット接続さえあれば実験が簡単にできるようになりました。コロナ禍で、実験室実験の実施が困難となったこともあり、世界中で広く使用されるようになりました。

oTree では、プログラミング言語として Python が使われています。また実験画面の表示には HTML を用います。Python は機械学習や、統計や解析ソフトウェアなどでも用いられていることから応用の範囲がとても広い言語です。学習者も多く、分からないことがあっても、書籍やインターネットでの情報が充実しているため容易に調べることができます。

oTree を自分のパソコンで動かすようにするには、Python と oTree のインストールが必要です。あとはエディタと呼ばれるソフトウェアがあるとコードを編集しやすいでしょう。エディタは、皆さんのお好みのもので良いですが、新しくインストールされる場合は、例えば、PyCharm をインストールしてみましょう。自分でプログラミングをせずに、本書が提供する実験だけ動かしたい場合は A.3 に進んでください。

A.2 環境構築

自分のパソコンで、独自に開発をできるようにすることを環境構築と呼びます。まず確認してほしいのが、自分のパソコンの OS（Windows/Mac/Linux）です。

[*3] 最新のマニュアルを必ず参考にしてください。ローカル環境構築やその後の動かし方については、Python などの知識も必要になるので学びつつ検索してみてください。

oTree のマニュアル URL にも詳しく書いてあるので、照らし合わせながらインストールしていきましょう。各種コマンドがうまくいかない場合は、oTree のマニュアルも参照してください。以下の環境構築は Windows を対象として書いているので、あなたが環境構築しようとしているパソコンが Mac や Linux の場合は適宜読み替えてください。

Python

対応している自身のパソコンと oTree のバージョンに合わせて Python をインストールします（2023 年 6 月現在 URL：`https://www.python.org/downloads/`）。Python のインストールファイルの形式は [Python-XXX-XXX99.exe] というような exe ファイルになっています。これをダブルクリックして起動し、Python をインストールします。基本的に、デフォルトのままで進んでいけばよいのですが、[Add Python X.X to PATH] にチェックを入れてください。

oTree

コマンドプロンプト（または、powershell）を開きます[*4]。Windows の検索で「CMD」または「コマンドプロンプト」と入力すると、コマンドプロンプトという図 A.1 のような黒い画面がでてきます。

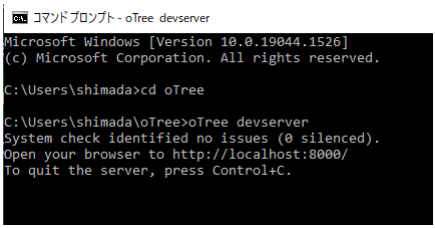


図 A.1: コマンドプロンプト:oTree 起動画面

[*4] Mac の場合はターミナルが該当します。

1. コマンドプロンプトで、[pip3 install -U otree] と入力（最後にエンターキーを押す）しましょう。ここでエラーがでるならば、まず、コマンドプロンプトで [Python -v] と入力し Python のバージョンを確認することで、Python がインストールされていることを確認しましょう。Python のインストールがうまくいっている場合は、エラーメッセージを検索にかけてみたりして解決策を見つけましょう。あなたが分からないところは、他にも分からない人、そして解決した経験がある人がいるでしょう。検索スキルも伸ばして行きましょう。
2. oTree の提供するサンプルプログラムをインストールします。[otree startproject oTree] と入力すると [Include sample games? (y or n):] のように聞かれるので [y] と入力します。サンプルプログラムが不要であれば [n] ですがあった方が便利です。
3. oTree フォルダに移動します。インストール時に設定したフォルダに移動するのでなにも変更していなければ [cd oTree] と入力して、oTree のフォルダに移動しましょう。
4. 次に、oTree を起動します。起動コマンドである [otree devserver] と入力して、コマンドプロンプトが図 A.1 のようになったら成功です。エラーが出た場合は [otree resetdb] でデータベース（実験で取得したデータなどが入っている箱）を初期化してみましょう。このコマンドは取得したデータがすべて消えるので注意して使ってください。
5. コマンドプロンプトはこのままにして、ブラウザ（Google Chrome や Firefox などのインターネットを閲覧することができるツール）を開いて、URL に [`http://127.0.0.1:8000/`] を入力してみましょう。図 A.2 のように、ブラウザ上でサンプルプログラムが見えたら、開発環境の構築が成功したことになります。図 A.2 のように画面が表示されます。この URL は、ブラウザ上でブックマークしておくとよいでしょう。
6. コマンドプロンプトに戻って、キーボードの [ctrl] キーを押しながら [C] キーを同時に押し、oTree の起動を止めることができます。する

と [http://127.0.0.1:8000/] の画面もエラー画面となります。

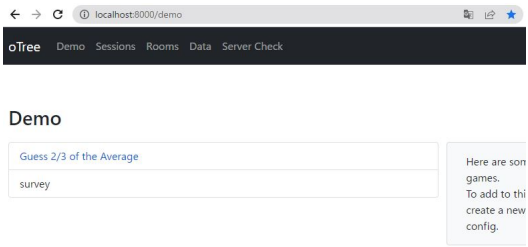


図 A.2: oTree の画面

これで oTree を用いて実験用のソフトウェアを自分のパソコンで開発できるようになりました。この自分のパソコンにある開発環境のことをローカル（環境）と呼びます。これはサーバーではないので、設定変更をしたり（不正にアクセスされたりなど）しない限り、あなたの開発環境が他の人に見られることはありません。ブラウザで見ることができているので、他のホームページなどと混同しがちですが、[`http://127.0.0.1:8000/`] で表示されている oTree の画面は、あなただけのローカルの開発環境です。例えば 2023 年 6 月では、世界中どこにいても [`https://www.iser.osaka-u.ac.jp/`] と入力すれば、誰であろうと同じく大阪大学社会経済研究所のホームページが表示されますが、[`http://127.0.0.1:8000/`] はローカル環境なので、あなたの画面と他の人の画面で見えているものが異なり、独自の開発環境が表示されます。

エディタ：PyCharm

最後に、エディタツールをダウンロードして、開発しやすい環境を整えましょう。ツールはさまざまな種類がありますが、ここでは PyCharm を導入

してみます。PyCharm[*5]をインストールしてみましょう。サイトからダウンロードした exe ファイルを起動してインストールします。

インストールしたあとは、PyCharm を起動させてみましょう。デスクトップに見つからない場合は Windows 左下にある検索で [PyCharm] と入力して探しましょう。PyCharm を起動させることができたら (左上の [File] >) [Open] でインストールした [oTree] フォルダを探してクリックし、[OK] を押します。今ブラウザ（`http://127.0.0.1:8000/`）に表示されている中身はこのフォルダの中にソースコードとして記述があります。

A.2.1 oTree プログラミングの基本

サンプルプログラムの [prisoner] を変更してみながら基本操作を確認しましょう。[prisoner] は囚人のジレンマゲームです。サンプルプログラムでは 2 人 1 組のペアとなり、それぞれが A または B を選択して結果として利得が決まるゲームになっています。ゲームの内容は、第 1 章を確認しましょう。

【PyCharm】settings.py

PyCharm で [prisoner] を開いてみましょう。図 A.3 のように、[settings.py] を開き、[SESSION_CONFIGS] に [prisoner] を追加してみてください。[SESSION_CONFIGS] に記載がある [guess_two_thirds] と [survey] は、図 A.2 の画面と対応しています。つまり、この [SESSION_CONFIGS] に記載することでブラウザからも各実験へアクセスできるようになります。なにを記載してもよいわけではなく、[settings.py] の左側、つまり oTree フォルダに入っているフォルダ名（prisoner、bargaining、dictator など）を入力してください。

プログラミングの基本はコピーすることなので、[dict] から始まるコー

[*5] 2023 年現在は、JetBrains s.r.o. が提供しています。URL は `https://www.jetbrains.com/pycharm/` です。2023 年現在は community 版だと無料なのでそちらを取得すれば十分です。

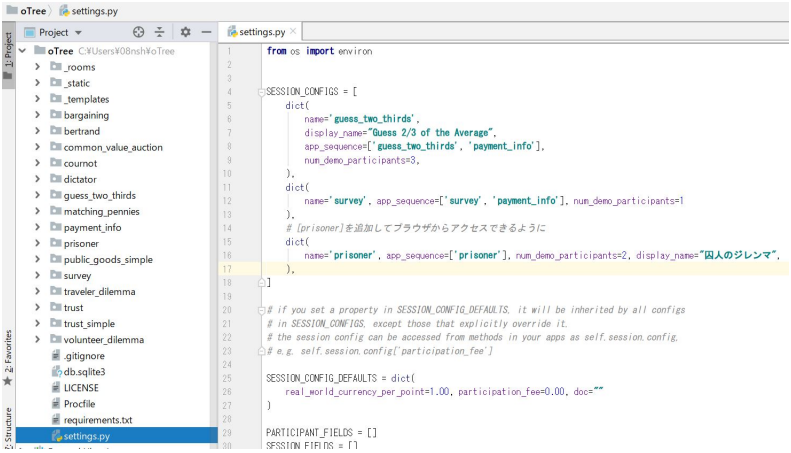


図 A.3: settings.py の変更

ドをコピーして作成するとよいでしょう。ただし、囚人のジレンマゲームは 1 章で学習した通り、2 人で行うゲームです。何人でゲームを実施するかを表す [num_demo_participants] を [=2] としてください。図 A.3 の [guess_two_thirds] にある [display_name] が画面上での表示名となるので、同じように [prisoner] にも [display_name=”囚人のジレンマ”] を追加して日本語で表示させてみましょう。うまくいかない場合は、半角全角や句読点の違いを探してみてください。うまくいけば、図 A.4 の画面になります。

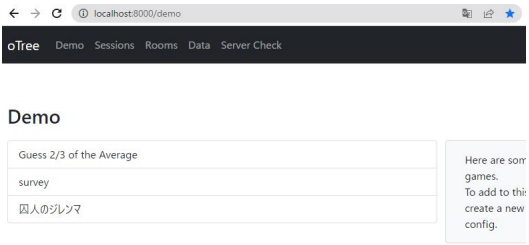


図 A.4: oTree の画面：囚人のジレンマを [display_name] ありで追加後

【ブラウザ】[囚人のジレンマ] を実施する

oTree で [囚人のジレンマ] 実験をローカル環境で実施してみましょう。図 A.4 にある、ブラウザの [囚人のジレンマ] を押し、[Play in split screen mode.] を押してください。画面が 2 つに分かれているのは、2 人 1 組のゲームなのでそれぞれのプレイヤーを表しています。画面を進めて動きを確認してみて下さい。図 A.5 のように画面が〒字に分かれていますが、〒字横線の間の部分で進捗や入力したデータを確認することができます。

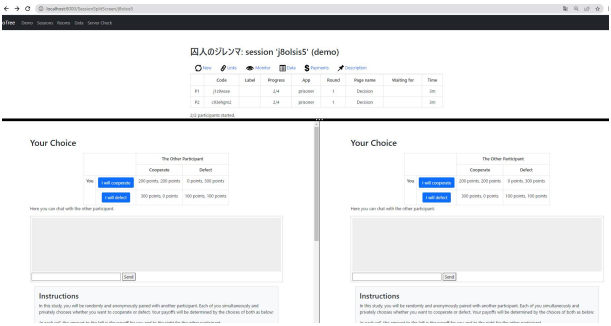


図 A.5: 囚人のジレンマ：[Play in split screen mode.]

【PyCharm】[prisoner] フォルダ

図 A.3 の左側、oTree フォルダの下にある [prisoner] フォルダを開きましょう。図 A.6 のようになります。これが oTree の各実験、この場合だと囚人のジレンマ実験の構成ファイルになります。拡張子が [.html] と [.py] となっているファイルがあることを確認してみてください[*6]。

[.html] ファイル

拡張子が html のファイルは、表示画面に関してのソースコードです。ここ

[*6] css や javascript なども使用することができ、さまざまなサーバー、データベースとも組み合わせることもできます。

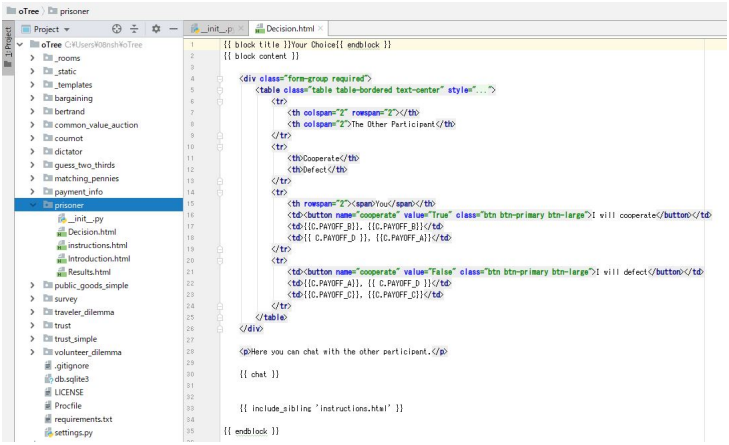


図 A.6: ファイル構成

のファイルを変更すると、ブラウザに表示される文字などを変えることができます。

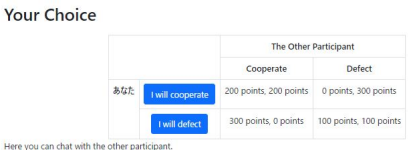


図 A.7: コード変更後 1：you をあなたに変更

図 A.8: コード変更後 2：テーブルに背景色をつける

図 A.5 の意思決定画面は、[Decision.html] と対応しています。Deci-

sion.html をブラウザを通してみると、図 A.5 の画面になるというイメージです。ためしに、[Decision.html]（図 A.6）の [You] を [あなた] と変更して、保存し、ブラウザをリロードしてみましょう。

図 A.7 のように画面上でも [あなた] となりました。Decision.html のソースコードを変更すれば、色々な画面を作ることができます。試しに、先ほど変更した「あなた」のテーブルに背景色をつけてみたものが図 A.8 になります。あなたが変更したいことがあれば、ブラウザで「html table 背景色変更」などで検索したり、専門書を読んだりして探してみましょう。すべてのコマンドを覚えることは難しいので、どういう検索をすれば目的とする結果を得られるかを意識してみましょう。

[.py] ファイル

初期の状態だと、拡張子が py のファイルは [init.py] だけなので開いてみましょう。このファイルは Python で書かれていて、内部的な処理を表しています。

＜画面の表示順＞

まず、画面に表示する内容の順番を決めている箇所を確認しましょう。html ファイルは、基本的に画面 1 枚に 1 ファイルが必要です。画面を表示するためには次の 3 つを確認してください。

(1) html ファイル
図 A.6 のように、[prisoner] フォルダの構成で、[Decision.html]、[instructions.html]、[Introduction.html]、[Results.html] の 4 つがあることを確認してください。

(2) py ファイル：page_sequence
init.py の中に書かれている [page_sequence] を探してみましょう。これは、画面の表示順を表しています。[page_sequence = [Introduction, Decision, ResultsWaitPage, Results]] となっています。Introduction は、[Introduction.html] の表示、Decision は、[Deci-

sion.html] の表示、Results は、[Results.html] の表示です。.html ファイルにはないけれど、[page_sequence] にはある [ResultsWaitPage(WaitPage)] はページではなく、処理のタイミングなどを制御します。ここでは同時手番なので相手がくるまで待つという処理（after_all_players_arrive）が書かれています。

(3) py ファイル：class 名

init.py にある [class XXX（Page）] を見てみましょう。この class 名は、(2) の [page_sequence] とも対応し、[class Decision(Page):] や [class ResultsWaitPage(WaitPage)] があり、各ページで実施する処理が書かれています。

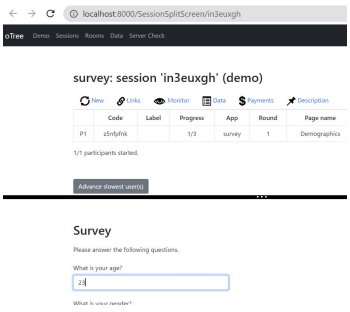
(1) ～(3) を見ると、囚人のジレンマは、実験の [Introduction] で説明を表示し、[Decision] 画面で参加者はそれぞれ入力し、(ResultsWaitPage) でペアがそろうのを待ち、[Results] で結果を表示するという流れになっていることが分かります。

これで、画面とプログラムの流れの確認までできました。html を作成して上記＜画面の表示手順＞に従って追加したり、[page_sequence] で各画面の表示・非表示を制御することができます。

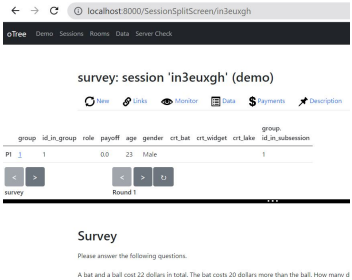
＜画面で入力した値の取得＞

では、結果を計算・格納する場所を確認し、実験結果の csv の出力まで見てみましょう。図 A.4 に表示されている [survey] のサンプルプログラムを例にして、画面入力から csv に格納されるまでの動きをまとめてみます。

図 A.4 のブラウザの画面に戻って、[survey] を押し、[Play in split screen mode.] を開いてみてください。図 A.3 の [settings.py] で survey を確認すると [num_demo_participants=1] となっているので、1 人で実施するプログラムということが分かります。そのため、[Play in split screen mode.] で開いても図 A.5 の囚人のジレンマのときのように分割はされません。図 A.9 を参照してください。



(a) 入力



(b) 格納：(a) 画面から次のページへ進み背景白の [Data] タブをクリックすると入力を確認できる

図 A.9: 画面入力から csv に格納されるまで（1）

図 A.6 で [prisoner] を開いたように [survey] フォルダを開き、構成を確認しましょう。[survey] フォルダにも [init.py] が同様にあり、各画面や [WaitPage] と対応しています。ここで、[page_sequence] にはない class [C(BaseConstants):]、[Subsession(BaseSubsession):]、[Group(BaseGroup):]、[Player(BasePlayer):] があることを図 A.10 を参考に init.py で確認してください。[prisoner] の [init.py] にも同様のクラスがあることも合わせて確認しましょう。各クラスは、役割がありますが詳細は、サンプルプロジェクトなどで手を動かして慣れてみたり、oTree の公式ページで確認をしてください。

これらの class には、[models.IntegerField][models.StringField] や [models.BooleanField] というように、models〜Field と書かれている箇所があります。これらがデータとして出力することができる箇所と対応します。データは、ブラウザ画面の [Data] をクリックすることで、csv ファイルで出力することができます。基本的に [XXX = models〜Field] とこの csv ファイルの列名 XXX が対応しています。oTree の機能として自動的に生成される値もあります。

(1) 図 A.9 は、ブラウザでみた oTree のサンプルプログラム [survey] の

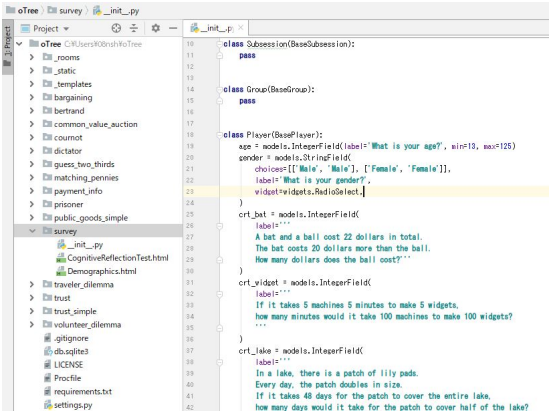


図 A.10: 画面入力から csv に格納されるまで（2）

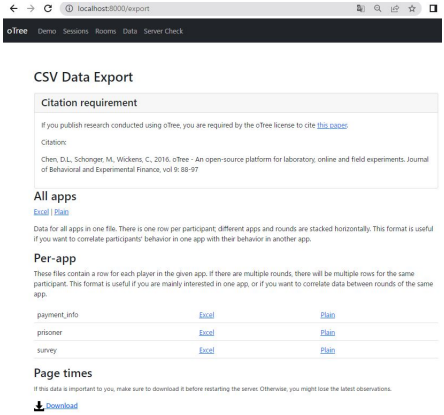


図 A.11: 画面入力から csv に格納されるまで（3）

画面です。ここで、図 A.9(a) の通り、年齢（What is your age ?）を「23」と入力したとしましょう。入力は自動保存ではなく次のページに遷移した時に格納されます。図 A.9(b) のように背景白の [Data] タブをクリックすると入力の格納を画面上から確認することができ

図 A.12: 画面入力から csv に格納されるまで（4）

ます。

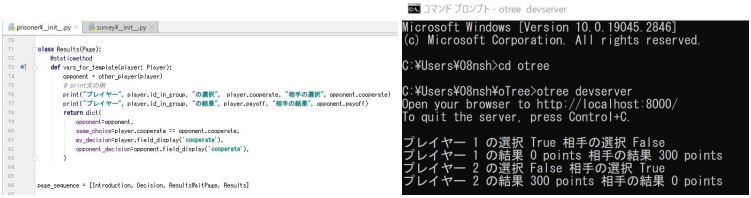
(2) (1) の入力がプログラムのどこに対応しているかというと、PyCharm の画面である図 A.10、[survey] の models.py の class Player にある age=models.IntegerField に対応しています。label で表示名の指定、min、max で入力の最小値と最大値を指定することもできます。

(3) (1) で入力した内容は格納されて csv としてダウンロードすることが出来ます。図 A.11 のように oTree をブラウザでみたときに上部にある黒い帯の [Data] をクリックすると今まで実施した実験データが格納されています。[survey] に該当している [Excel] ボタンをクリックして csv データを取得してみましょう。図 A.11 の [prisoner] には囚人のジレンマのデータが入っています。

(4) ダウンロードした csv を開いてみると図 A.12 のように表示され、入力した「23」が player.age の列に格納されていることが分かります。

＜ print 文: コマンドプロンプトに出力＞

[init.py] で通過する処理に print 文を記載し、変数名を記入するとコマンドプロンプト上に表示させることができます。



(a) print 文を記載して、その処理を通過させるようにブラウザを動かす　**(b)** 処理を通った段階で、print 文の内容がコマンドプロンプトで表示される

図 A.13: print 文の例

図は A.13(a) は [prisoner] の結果画面（class Results）で、print 文で意思決定と結果を表示するコマンドを入力しました。ブラウザ上で結果画面まで進めると、図 A.13(b) はのように、コマンドプロンプトに表示がなされます。プログラミングする際には、なぜか動かない・エラーが出るなどさまざまなことが起きます。print 文によって、解決するヒントを得ることができるのでぜひ活用してください。

＜新しいプロジェクトを作成する＞

慣れていくと自分で一から実験を構築する機会があるかもしれません。もちろん、[prisoner] や [survey] プロジェクトを変更していってもいいですが、新しいプロジェクトを oTree で作成する方法があります。ためしにここでは、新しいプロジェクト名を atarashiprojectmei としましょう。図 A.1 に戻ります。ローカル環境を構築したあなたの PC が Windows であれば、この状態から oTree を停止 ([ctrl] を押しながら [C] キーを押す) させて、[otree startapp atarashiprojectmei] と入力してください。すると、図 A.6 の oTree フォルダの直下に [atarashiprojectmei] フォルダが作成されます。図 A.3 にて [prisoner] や [survey] のように、[atarashiprojectmei] を設定して画面に表示させることができます。

＜ローカル環境にプロジェクトを作成する＞

ローカル環境に、この本のプログラムを設定する方法は、後述の A.5 節を参考にしてください。

さまざまな機能が用意されているので、サンプルプログラムや検索を見ながら、あなたの実験プログラムの作成を楽しんでください。

A.3 Heroku

プログラムはサーバーにアップロードすることで、あなただけがみることができるローカル環境から誰でも実験に参加可能な状態にすることができます。サーバーは自身で作成することもできますが、ここでは Heroku というオンラインサーバーを使った方法をご紹介します。

図 A.14 は Heroku のログインページです。初めての場合は [Sign Up] から、アカウントを作成し使用言語を聞かれたら【Python】を選択します。無料プランが廃止されたことを受け、2023 年現在は数ドルから使用できます。

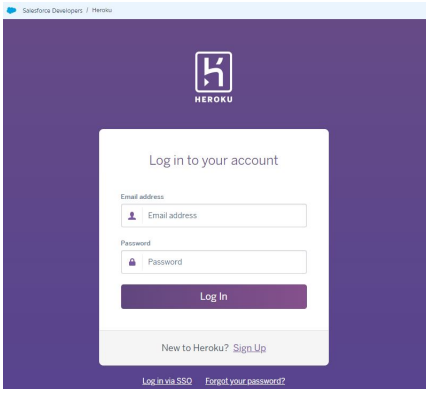


図 A.14: Heroku: ログイン画面

A.4 GitHub

GitHub（https://github.com/）はオンライン上でプログラムの管理ができるサービスです。2023 年現在、無償で提供されており、Heroku と連携を

行ってプログラムをサーバーに簡単にアップすることができます。ここでは、この本の実験をご自身の GitHub にアップして、Heroku と連携してオンラインで実験が実施できるようにします。まずは、GitHub でアカウントを作成してください。

アカウント作成してログイン後、図 A.15 を参考に、[Repositories] タブ > [New] ボタンを押して、[Create a new repository] の画面に進んでください。

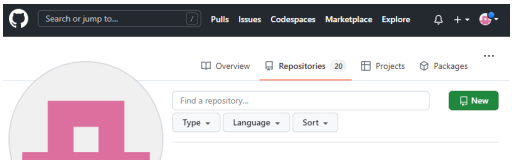


図 A.15: GitHub：Repositories 画面

Repository name は、作成するリポジトリの名前です。Heroku と連携させる際に使用するので自分で把握しやすい名前にし、そのほか公開の範囲（Private（オンライン上で非公開）、[Add a README file] をチェックを設定することが多いです）などを決定してリポジトリを作成します。これでなにもはいっていない空の状態のリポジトリができました。ここに、自分のローカルプログラムを載せて管理したり、Heroku と連携させることが可能になります。

A.5　本書での各種プログラムをセットする

ローカル環境、または、GitHub のリポジトリに、この本のプログラムを設定していきましょう。この本のプログラムは付録 B の通り、GitHub(`https://github.com/iserExperiment/IntroEconBook`) にて公開されています。URL にアクセスし、図 A.16 を参考にして、[Code] > [Download ZIP] を押してプログラム一式の zip を自分の PC にダウンロードしてくだ

さい。

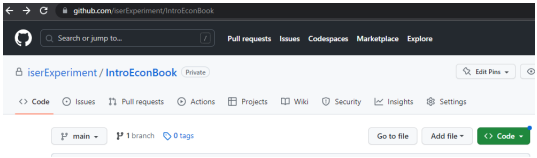


図 A.16: GitHub：実験プログラムダウンロード

ダウンロードした zip ファイルを自分の PC で解凍します。Heroku を使用して動かしたい場合は、前節で作成した GitHub の新しいリポジトリに必要なファイル一式をアップロードしてください（いろいろな方法がありますが、一番簡易的な方法はブラウザ上のリポジトリにすべてのファイルをドラッグ&ドロップすることだと思われますが、ただし一度にアップできるファイル数の制限があります）。アップしたファイルは、GitHub の画面上でも変更することができるようになります（A.7 節を参照）。ローカル環境で動かしたい場合（A.2 節を参照）は、解凍したフォルダにコマンドプロンプトで移動し、oTree を起動すればローカル環境で動かすことができるようになります。ローカル環境のファイルは PyCharm で開いて変更などしていくこともちろん可能です。

A.6　GitHub:Heroku との連携

Heroku にブラウザからログインし、ブラウザ画面右上の [New] > [Create new app] を押します。図 A.17 を参考にしてください。

画面が [Create new app] に切り替わります。App name は、作成するアプリの名前です。URL などに使われるので適当に入力して（世界中で他のユーザーとも重複があるとはじかれる場合があります）、[Create app] をクリックしましょう。

app を作成できたら、Heroku の画面にて [Settings] タブの [Domains] に

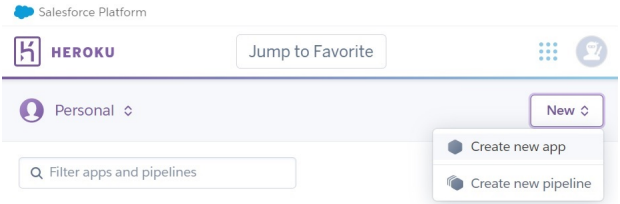
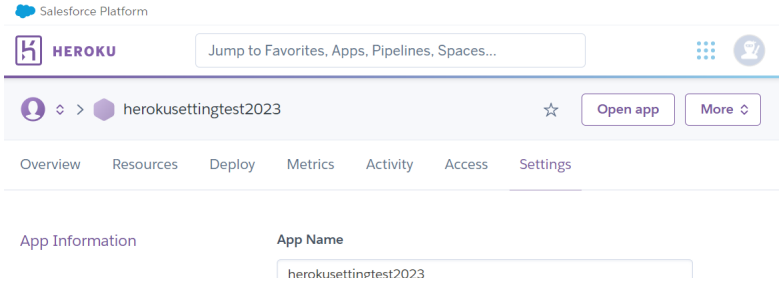
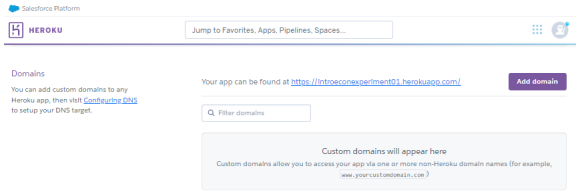


図 A.17: Heroku: アプリ新規作成

ある URL を押します。図 A.18 の (a) で [Settings] タブ位置、(b) で URL の位置を参考にしてください。



(a) Heroku:Settings タブ位置



(b) Heroku: 上の Settings タブを下にスクロール：Domains のところに URL

図 A.18: Heroku: サーバー URL

URL をクリックすると何もプログラムがはいっていなければ、ブラウザで図 A.19 のような Welcome 画面となります。まずはここまでで、Heroku の app が作成できたことを確認できます。次の GitHub のプログラムの連携の前にA.8 節を参照して、あなただけがアクセスできるようにパスワードを設定することをお薦めします。

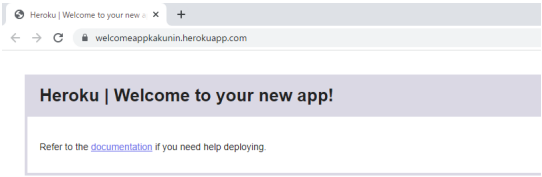
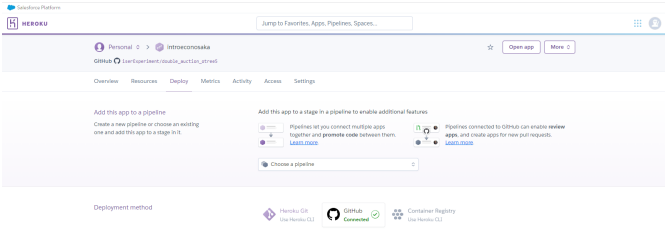


図 A.19: Heroku:Welcome 画面

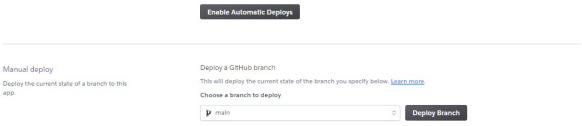
GitHub のプログラム連携

はじめて Heroku と GitHub を連携させる場合は、A.8 節を参照して他の人があなたの app にアクセスできないようにパスワードをかけておきましょう。

ここでは、Heroku の app に GitHub のプログラムを連携して実験ができるようにします。Heroku にて作成した app をクリックして画面を開きます。図 A.20a を参考に、[Deploy] タブを開きます。[GitHub] をクリックして、自分の GitHub アカウントと連携させます。連携はいつでも解除することができますし、連携させたプログラムはこの app 上で動きます。



(a) Heroku:Deploy タブ



(b) Heroku:Deploy タブを下にスクロール：Deploy Branch ボタン

図 A.20: Heroku:GitHub との連携：Deploy

スクロールして下の [Manual deploy] の [Choose a branch to deploy] からやってみたい実験をセレクトボックスから選びましょう。図 A.20b を参考にし、ひとつ決めたら [Deploy Branch] を押して連携させます。

ここでうまくいけば、図 A.18 の URL を開くと oTree の画面が表示され、設定は終了します。A.8 節にもとづいて、パスワードを設定すればログインに設定したパスワードが必要になります。

A.7　実験・授業での oTree 操作：ROOM

oTree では、実験や授業のように大人数が参加するときは ROOM という機能を用いることができます。ルームに関しての説明はオンラインマニュアル（`https://otree.readthedocs.io/ja/latest/rooms.html`）も参照してください。

ブラウザの oTree 画面の上部にある背景黒色の帯にある [Rooms] を開く

と、図 A.21 のような画面となります。

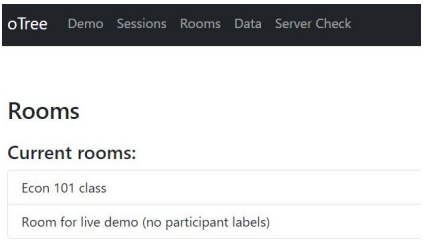


図 A.21: oTree:ROOM 画面

[Room for live demo (no participant labels)] を押します。つぎの画面で、指定の実験（Session config:）と、指定の人数（Number of participants）を設定すると、実験を実施することが可能になります。Heroku 上で、この画面に表示される [http://あなたのサーバー名/room/live_demo/] というリンクを参加者に送付する（実験室であれば各座席の PC のブラウザで開く）と指定した実験に参加してもらうことができます。ただし、この ROOM の作り方だと誰がどの入力をしたか分からない状態で結果が出力されます。本書では（学籍）番号を入力させるプログラムも提供しているので参加者の識別にはそれを活用したり、オンラインマニュアルを参考に、参加者ラベルを指定する方法（実験室であれば各座席番号を設定すること）が考えられます。

GitHub 上でコードを変更する

GitHub 上でコードを変更することが可能です。図 A.22 は、A.4 節で作成したリポジトリに A.5 節でダウンロードした本書の実験プログラム一式をセットした GitHub のアカウント画面です。ここでは、[settings.py] を変更して [新しく作成した実験] を画面に表示させてみましょう。GitHub の画面上で編集したいファイル（settings.py）を開き、[Edit this file] ボタン（図 A.22 の一番下枠の右上：鉛筆マーク、もしくは [▼] より [Edit in place]）をクリックすることで編集モードに切り替えます。図 A.22 の通りに追加編集

することで、[新しく作成した実験] を画面に表示させ、[（学籍）番号入力 (b_input_id)] > [第一章リスク実験 (ch1_1_risk)]] > [第十章独裁者ゲーム (ch10_3_dictator)] > [第二章展開形ゲーム (ch2_4_extensive)] の順番で実験を設定したものです。実験名とプログラムの対応は、付録 B：本書で用いる実験一覧を参照してください。

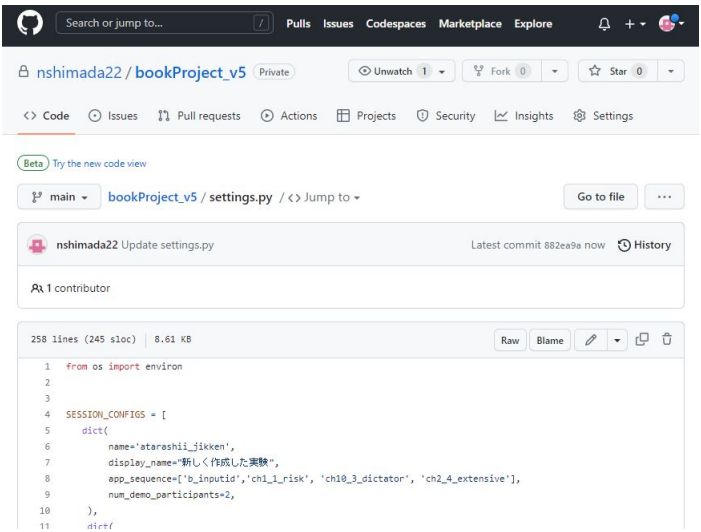


図 A.22: GitHub: コード編集画面

ファイル編集後、下にある [Commit changes] ボタンで編集を完了し、図 A.20 からの処理を実施してサーバーで動くようにしますが、その際に、ファイルが動かなくなる可能性があるので、新しく Heroku でデプロイする前には、URL から oTree 画面に行き oTree のデータファイルがあれば取得するようにしてください。もしエラーが出たら、図 A.24 なども参考に、エラーを除去してください。

[Advance slowest user(s)] ボタン

[Advance slowest user(s)] ボタンをクリックすることで ROOM でもその ROOM にいる参加者の実験のページを強制的に進めることが可能です。一番遅いページの人が一番早いページにいる人に追いつくように進められますので、図 A.23 の状態ですと P2 と P3 が P1 のページに追いつくようにページが進められます。入力は途中保存されないので、進めた場合はデータは入力されないことに注意してください。

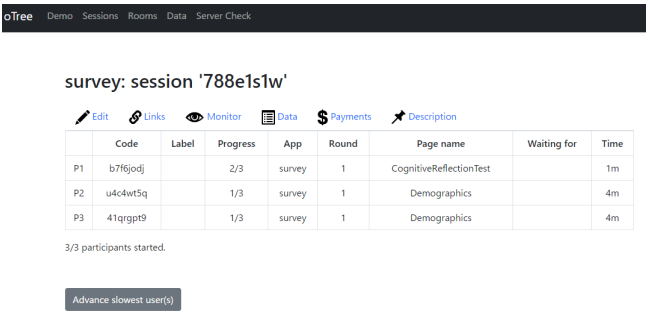


図 A.23: oTree:ROOM：Monitor タブ

図 A.9 にも [Advance slowest user(s)] ボタンがあるように、ROOM だけでなく各実験ページでも同じような役割を持ちます。

Run console:Heroku 上でのコマンドプロンプト的役割

Heroku 画面の右上の [More] > [Run console] をクリックすると図 A.24 のような画面が開きます。ここは、ローカル環境でのコマンドプロンプトのような役割を果たします。ここに [otree resetdb] と入力し、データベース（実験で取得したデータなどが入っている箱。csv で取得できるデータ）を初期化することが可能です。重複となりますが、このコマンドは取得したデータがすべて消えるので注意してください。

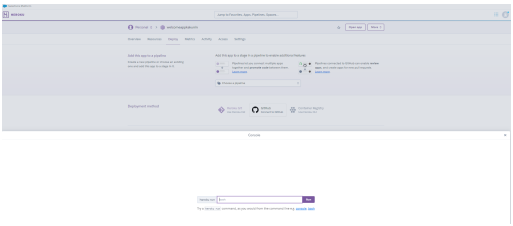


図 A.24: Heroku:Run console 画面

A.8　パスワード設定

あなたのつくった Heroku の app はあなただけの実験に使用することができます。ただしこのままだとあなたの app の URL さえ知っていれば誰でもアクセスができるようになっていますので、パスワードを設定してみましょう。

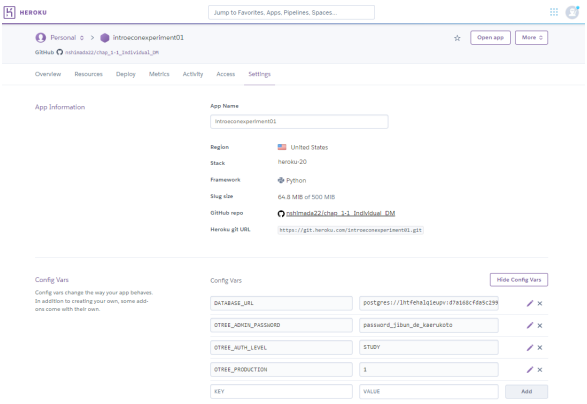


図 A.25: Heroku:config vars（環境変数）設定

Heroku の画面で [Settings] タブを開き、[Config Vars] を押すと、図 A.25 のように開きます。KEY に [OTREE_AUTH_LEVEL]、VALUE に

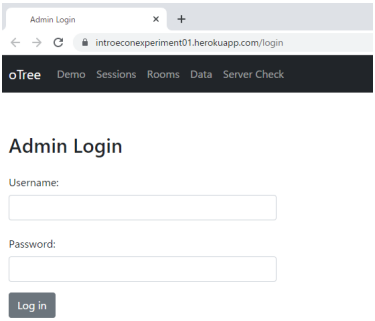


図 A.26: oTree: ログインを設定した画面

[STUDY] と入力し、[Add] ボタンをクリックします。同様に KEY に [OTREE_ADMIN_PASSWORD]、VALUE に [あなたの設定するパスワード] を入力し、[Add] ボタンをクリックしてパスワードをかけましょう（oTree のマニュアルも参照してください：`https://otree.readthedocs.io/ja/latest/admin.html`）。もちろん、[あなたの設定するパスワード] にはあなたの決めた任意のパスワードを入力するようにしてください。この設定をすることで設定したサーバー上の oTree の URL を押すと図 A.26 のようにログイン画面が現れます。この画面で、ユーザー名は（変更していなければ）[admin]、パスワードは [OTREE_ADMIN_PASSWORD] で設定した [あなたの設定するパスワード] を入力することで oTree の実験画面に入ることができるようになります。

また、実験画面の下にある [Debug info] を消したいときは上記と同様に KEY に [OTREE_PRODUCTION]、VALUE に [1] と設定します。

付録 B

本書で用いる実験一覧

実験プログラムダウンロード：URL

https://github.com/iserExperiment/IntroEconBook

提供するプログラムは、本書の作成に当たり授業で実施した内容を再編集しています。インストラクション等が別プログラムになっている実験もあります。settings.py や各種実験ファイルも参照してください。今後、適宜修正変更する可能性もあり、動作を完全に保証するものではありません。実験データも含めてご自身でもご確認ください。本に掲載している実験に基づく図表等は、当時の oTree の仕様とサーバー環境に基づいた csv データから作成しており、四捨五入等が自動的に行われている場合もあります。実施する場合は、ブラウザやローカル・サーバー環境等についてもご確認ください。最新の情報は、oTree の HP（https://otree.readthedocs.io/ja/latest/index.html）を確認してください。

実験後のグラフ

実験後、グラフが表示されるプログラムがあります。グラフの機能は、Highcharts[*1]を使用しています。oTree のグラフ説明ページ（`https://otree.readthedocs.io/ja/latest/templates.html#charts`）も参考にしてください。また、データをダウンロードして整合性も確認してください。

授業等での進め方

授業で実験を用いることは、学生の理解を深める教育的な効果もあることも示されています (藤井・大谷, 2016)。授業で教える内容に応じて実験・トピックをピックアップして頂ければと思います。

参考：Heroku の設定と金額

2023 年現在、Heroku は月額と各 app を設定使用した分の従量課金制となっています。途中で app を削除すればその時点以上の金額はかからないので、数ドルから利用することができます。

2022 年度実施の授業や実験では、[Add-ons] でデータベースとして Heroku Postgres を追加し、ときどき [Dynos] プランをアップグレードしました。参考までに月額の費用は日本円で数百円〜数千円程度でした。金額等は、oTree や Heroku のホームページを参照してください。

[*1] 2023 年現在、Highsoft AS が提供しています。URL は `https://www.highcharts.com/` です。

表 B.1: 本書で用いる実験一覧

	(学籍等) 番号入力	b_input_id
第 1 章	個人の意思決定（リスクあり）	ch1_1_risk
	囚人のジレンマ	ch1_2_prisoner
第 2 章	調整ゲーム 1	ch2_1_coordination
	調整ゲーム 2	ch2_2_coordination2
	チキンゲーム (同時手番)	ch2_3_chicken
	チキンゲーム (展開形)	ch2_4_extensive
	PK ゲーム	ch2_5_PK
第 3 章	個人の意思決定	ch3_0_shortandlong
	繰り返しゲーム（1 回）	ch3_1_repeated_oneshot
	繰り返しゲーム（有限）	ch3_2_repeated_finite
	繰り返しゲーム（無限）	ch3_3_repeated_infinite
	公共財	ch3_4_public_goods_game
	時間割引	ch3_5_time_discount
第 4 章	通常の市場	ch4_double_auction
第 5 章	公害のある市場	ch5_externality
	税のある市場	ch5_externality_tax
第 6 章	独占	ch6_individual
	寡占	ch6_mutual
第 7 章	逆選択	ch7_1_adverse_selection
	隠された情報 1	ch7_2_hidden_action
	隠された情報 2	ch7_2_hidden_action_nonlottery
第 8 章	比較優位（自給自足）	ch8_comparative_advantage1
	比較優位（同じタイプ）	ch8_comparative_advantage2
	比較優位（全員と取引可能）	ch8_comparative_advantage3
第 9 章	第一価格オークション	ch9_auction_firstprice
	第二価格オークション	ch9_auction_secondprice
	IA マッチング	ch9_matching_ia
	DA マッチング	ch9_matching_da
第 10 章	個人の意思決定（プロスペクト）	ch10_1_individual_choice
	最後通牒ゲーム	ch10_2_ultimatum
	独裁者ゲーム	ch10_3_dictator
	拡張版独裁者ゲーム	ch10_4_extended_dictator

参考文献

Chen, Daniel L., Martin Schonger, and Chris Wicken (2016) “oTree - An Open-Source Platform for Laboratory, Online and Field experiments,” *Journal of Behavioral and Experimental Finance*, Vol. 9, pp. 88-97.

藤井陽一朗・大谷剛 (2016) 「導入教育としての経済実験の有効性についての分析: ダブル・オークションを用いたアプローチ」,『大阪産業大学経済論集』, 第 17 巻, 第 3 号, 199–214 頁.