

Iterated Drawing Application

Web application for Professor De Boer @ VUB AI Lab, concerning language evolution. The Django framework is used, following a model-view-controller model. Verification and storage of data is handled by the backend part of the project.

The source code of the project is contained within the *src* folder.

Setup

In the *src* folder, the *activate_experiment.sh* file can be executed to start the Django server. **Make sure variables in *Experiment.py* are correct before setting up the project. These are mainly paths to directories like "cases" and "experiments" in order to load the sessions.**

1. Open terminal in project directory and run the *activate_experiment.sh* script. This will create a virtual python runtime environment for the application to run, installed with the necessary modules.
2. The initial cases are loaded into the experiment (from the *data/cases* folder). This can be changed in the *Experiment.py* when other cases have to be loaded for the first time.
3. Go to 127.0.0.1:8000 in the browser for interaction on the local machine. For local network access, open the browser on the client machine (tablet or phone) and go to the IP address of the host (using the command *ip addr* to find the IP of the host), including port number 8000.
4. After an experiment is completed, the data is saved as described in the **Data** section.
5. To close the project: press ctrl+c in the console.

Frontend

The statics folder contains all static elements like JavaScript, CSS and images. The *canvas.js* file contains the draw application as well as path recording functions during the Session.

Backend

The backend uses several abstractions in order to execute an experiment. An **Experiment** class consists of a **TrainingSession** and a **TestingSession** class. Experiment is responsible for the correct loading of case.

A Session consists of several Cases that could be loaded into the experiment. Each Case also consists of Trial classes who're responsible for evaluating the input of every user trial.

Form

In *src/form/models.py*, the form can be defined for registering a new user. Default, first/second name, mail and age are asked.

RESTful API

A simple RESTful API is used for data transfer between client and host while on a page. For both kind of Session, a **get_case** and a **post_case** method. JSON objects and strings are used as the transferred data.

Call	Arguments	Response
training/get_case	none	JSON: {action:String, path:Array{x:Int, y:Int, t:Int}}
training/post_case	JSON: Array{x:Int, y:Int, t:Int}	String:"tolerated", "not tolerated", "session done"
testing/get_case	none	JSON: {action:String, path:Array{x:Int, y:Int, t:Int}}
testing/post_case	JSON: Array{x:Int, y:Int, t:Int}	String: "tolerated", "session done"

Data
