

Maintenance Manual

Ioana Iulia Lucaci

2021

This appendix presents the maintenance manual for the prototype, including requirements and a breakdown of all of the prototype's files.

1 Requirements

This program was designed to be cross-platformed, and can be run on any device that can install the following tools:

- Python version ≥ 3.9 (with pip version ≥ 21.0)
- JupyterLab or Jupyter notebook
- Requirements found in `requirements.txt`

Python can be installed from the Python website¹. This will also automatically install pip as well. Instructions on how to install JupyterLab or Jupyter notebook are found on Jupyter's website². The extra requirements can be installed using pip with the command `pip install -r requirements.txt` once you are in the program's folder.

2 Compilation Instructions

The .zip file must first be unzipped. After navigating in the unzipped file's location, the extra requirements must first be installed, located in the `requirements.txt` file. This can be done using the command `pip install -r requirements.txt` in the file's location.

Afterwards, the simulation can be configured changing the `auction.txt` file's details. In the end, the program can be run by running `main.py` file using Python or by running the Jupyter notebook found in `interactive.ipynb`. JupyterLab, once installed, can be run from the command prompt using the command `jupyter-lab`. Jupyter notebook can be run from the command prompt, once installed, using the command `jupyter notebook`. Afterwards, navigate to the project's folder and open the file `interactive.ipynb`.

3 Files

This section describes the files of the program, along with their usages. Section 3.1 describes the files that can be updated to change certain aspects of the simulator. Section 3.2 contains information of all of the files present in the .zip file.

3.1 Configuration File

The file `auction.txt` is in charge of collecting all of the required information for the auction configuration. A more in-depth explanation can be seen in the User Manual. The .zip file contains a sample file with the format that needs to be followed. The results of the simulations will be saved in a .csv file in the same location as the `auction.txt` file. The file will be named based on the date, so any simulations run in the same day will override the previous information.

¹<https://www.python.org/downloads/>

²<https://jupyter.org/>

The bidders' and auctioneer's profiles are based on mathematical functions. These functions can be changed in the file `auction_information.py`. This is where the agents' rate's range can also be updated.

3.2 Code Files

Filename	Description
<code>main.py</code>	This file is the main file, where rest of the functions are connected and the simulators are run.
<code>agents_factory.py</code>	This file is in charge of populating the information for agents. This is where the bidder's parameters and auctioneer's rate can be changed.
<code>auction_information.py</code>	This is in charge of how the rate is updated. The profiles' functions can be updated here.
<code>auction.py</code>	The file holds the Auction class. This is where the agents are constructed and the metrics are collected.
<code>auctioneer.py</code>	The file holds the Auctioneer class. This is where the auctioneer's logic for the different functions can be updated.
<code>bidder.py</code>	The file holds the Bidder class. This is where the bidder's logic for the different functions can be updated.
<code>data_analyser.py</code>	The file is in charge of analysing the metrics. This is based on the previously saved .csv file.
<code>metrics_writer.py</code>	The file is in charge of writing out the metrics in a .csv file.
<code>parameters_reader.py</code>	The file is in charge of parsing the <code>auction.txt</code> file and taking the information in.
<code>parameters_writer.py</code>	The file is in charge of writing parameters in <code>auction.txt</code> . Used only when dealing with the Jupyter notebook.
<code>interactive.jpynb</code>	This contains the Jupyter notebook for this project.
<code>auction.txt</code>	The file contains the information needed to configure the simulations.
<code>requirements.txt</code>	The file contains all of the needed library requirements.

Table 1: Code files description