

# Evaporative Cooling Simulator

A simulator to generate the data to train an ML system and visualise the simulation result.

## Features

- Generate simulation data based on the time, temperature and occupancy count
- Visualise the animated simulation result with charts
- Export simulation result as image

## Tech

The program is written in Python3 and required the following library installed:

- adjustText==0.7.3
- matplotlib==3.1.3
- pandas==1.0.1
- numpy==1.18.1

## File Structures

```
├─ data
│   ├── images
│   │   └─ image1..n
│   ├── melbourne_temperature.csv
│   ├── melbourue_temperature_occupancy.csv
│   └─ melbourne_temperature_calculated.xlsx
├─ configure.ini
├─ requirments.txt
├─ simulator.py
├─ matrix_generator.py
├─ helper_function.py
├─ animation_player.py
└─ ReadMe.md
```

File	Description
image 1..n	All the export images will be stored under data/images
melbourne_temperature.csv	Simulator input file which include the hourly <b>Time</b> and <b>Temperature</b> data from 9AM to 17PM. This input file is required when the occupancy number is randomly generated.
melbourue_temperature_occupancy.csv	Simulator input file which include the hourly(10 minutes interval) <b>Time</b> , <b>Temperature</b> and <b>OccupancyCount</b> data from 9AM to 17PM. This input file is required when the occupancy number is provided.
melbourne_temperature_calculated.xlsx	This is the matrix of the simulation data which include the stage 1, 2 and 3 corrections.
configure.ini	This is where user set up the behavior of the simulator
requirments.txt	All the dependencies required for this python program,

	see the Installation section to insall the dependencies.
simulator.py	Simulator python script, it contains the logic of preparing and animating the graphs.
matrix_generator.py	It contains the logic of loading the raw data and calculate the stage 1 2 & 3 matrixs.
helper_function.py	It contains the helper functions that used to calculate occupancy positions, nozzle's postion and status.
animation_player.py	It contains the python Class for animation player and the tool bar.
ReadMe.md	Documentation file

## Installation

Install the dependencies.

```
pip install -r requirements.txt
```

## Run the Simulation

After install the dependencies libraries, you can run the simulator using following command.

```
python simulator.py
```

## Simulator Configuration

Below is the content of the `conigure.ini`, the settings in this file will affect the behaviors of the simulator.

```
[SETTING]
AUTO_GENERATE_OCCUPANTS=False
[CHART]
MAIN_TITLE=Evaporative Cooling System Simulation
ROOM_TITLE=Room Occupancy
CHART2_TITLE=Temperature & Occupancy VS Time
CHART3_TITLE=Release Periods & Active nozzles VS Time
[PARAMETERS]
MAX_OCCUPANCY=10
[CHARTEXPORT]
EXPORT=False
START_FRAME=0
END_FRAME=10
```

**[SETTING]** The values of `AUTO_GENERATE_OCCUPANTS` could be `True` or `False` .

- When `AUTO_GENERATE_OCCUPANTS = True` , the simulator will take `melbourne_temperature.csv` as input and generate random occumancy number between 1 and `MAX_OCCUPANCY` for each 10 minutes time interval.
- When `AUTO_GENERATE_OCCUPANTS = False` , the simulator will take `melbourue_temperature_occupancy.csv` as input, and use the predefined occupancy count for each 10 minutes interval.

**[CHART]** The value of `MAIN_TITLE` , `ROOM_TITLE` , `CHART2_TITLE` , `CHART3_TITLE` could be any text value. It will affect the heading text of the charts in the visualisation.

**[PARAMETERS]** The value of `MAX_OCCUPANCY` should be an interger large than 1. It will affect the max number of random occupancy count for each time interval.

**[CHARTEXPORT]** The value of `EXPORT` could be `True` or `False` .

- When `EXPORT = True` , the script will generate the images for frame number between `START_FRAME` and `END_FRAME` . For example, if the `START_FRAME=0` and `END_FRAME=10`, the script will generate 11 images from frame 0 to frame 10. **Each time interval is considered as a frame.**
- When `EXPORT = False` , the script won't generate images and will only start the visualisation.

The value of `START_FRAME` and `END_FRAME` are integer.

- The integer value should be between 0 and (total frame number-10)
  - Based on the visualisation design, the default slider window is 10 time intervals values. The simulator visualisation will stop at the last 10 frames, because there are no further values to display.
- `START_FRAME` should be smaller than `END_FRAME`
- Each frame image takes around 1 seconds to generate

## Developer Notes

This simulation program contains three main parts:

- Matrix calculation for stage 1,2 & 3. Anything related to matrix please update the code in `matrix_generator.py`
- Create Graphs - Currently the simulation include three graphs, for any future update about adding graphs/elements please check the code before `plot_simulation` function in `simulator.py`
- Update the Graphs(Animation) - The graphs are update based on the data in matrix, if you want to change how the graph is animated please check the `plot_simulation` function in `simulator.py`

**Additional links** Subplot

- [https://matplotlib.org/stable/api/as\\_gen/matplotlib.pyplot.subplot.html](https://matplotlib.org/stable/api/as_gen/matplotlib.pyplot.subplot.html)
- [https://matplotlib.org/stable/gallery/subplots\\_axes\\_and\\_figures/axes\\_margins.html#sphx-glr-gallery-subplots-axes-and-figures-axes-margins-py](https://matplotlib.org/stable/gallery/subplots_axes_and_figures/axes_margins.html#sphx-glr-gallery-subplots-axes-and-figures-axes-margins-py)

Graph share x axis

- [https://matplotlib.org/stable/gallery/subplots\\_axes\\_and\\_figures/two\\_scales.html](https://matplotlib.org/stable/gallery/subplots_axes_and_figures/two_scales.html)

Animation

- <https://stackoverflow.com/questions/44985966/managing-dynamic-plotting-in-matplotlib-animation-module/44989063#44989063>

Matplotlib Library - where you can find code examples

- <https://matplotlib.org/stable/gallery/index.html>