

# Smart Grid Simulation

**A modular C++ library with PyQt interface for simulating smart electrical grids.**

## **Key Features:**

- ✓ Dynamic energy management: Add/remove producers (solar, wind, etc.) and consumers (houses, factories).
- ✓ Battery storage system: Simulate charge/discharge cycles with adjustable capacity.
- ✓ Real-time simulation: Balance production/consumption and calculate energy needed to cover the battery capacity.
- ✓ User-friendly PyQt interface: Visualize and interact with the grid in real time.
- ✓ Extensible architecture: Easy to integrate new energy sources or custom algorithms.

# Setting up the grid

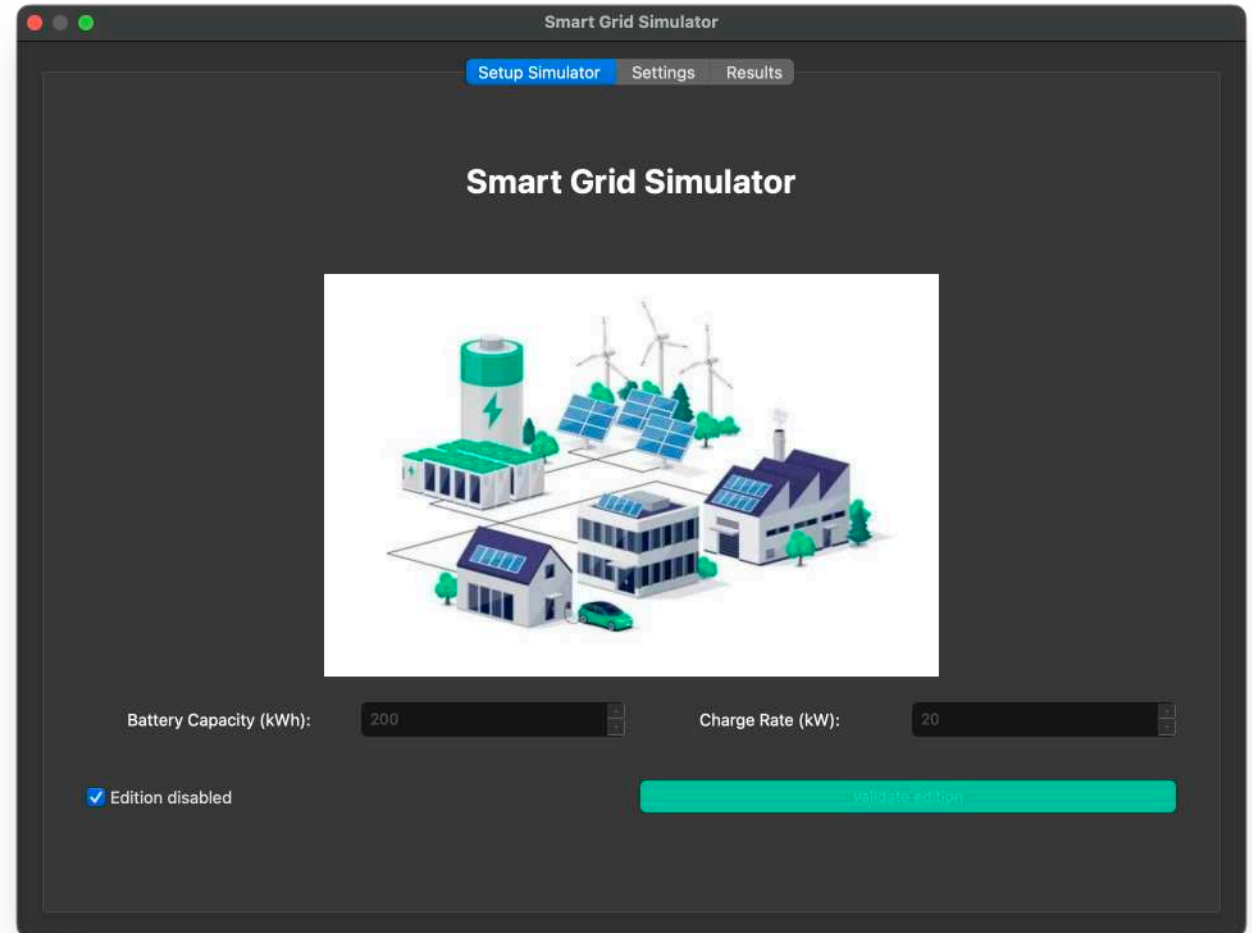
## Feature :

Set the maximum battery capacity and the charge rate.

## Description:

The battery allow the grid to provide energy to consumers when the production is not enough. If the battery is discharged, the app "purchased" energy from outside the grid.

The goal is to have a grid well set with enough energy production for the different consumers. And the battery should help with a peak of energy demand.



# Manage the Producers and Consumers of the grid

## Feature :

Add or remove Producers and Consumers in the grid. Also setting their capacity and base demand.

## Description:

The Producers provide energy to the grid and Consumers use it. The main goal here is to maintain a balance between producers and consumers. The battery should only provide energy to handle peaks of energy demand. So the grid don't waste or purchase too much energy.

The image shows a software interface titled "Smart Grid Simulator" with three tabs: "Setup Simulator", "Settings", and "Results". The "Settings" tab is active, displaying "Producers and consumers settings".

On the left, under "Add Producer", there is a dropdown menu set to "wind", a numeric input field set to "50", and a green "+" button. Below this is a table with two rows of existing producers:

	Type	Capacity (kW)	Action
1	solar	50	<button>Delete</button>
2	wind	50	<button>Delete</button>

On the right, under "Add Consumer", there is a dropdown menu set to "industry", a numeric input field set to "150", and a green "+" button. Below this is a table with two rows of existing consumers:

	Type	Base Demand (kW)	Action
1	household	50	<button>Delete</button>
2	industry	150	<button>Delete</button>

At the bottom of the interface is a wide green button labeled "Confirm Settings".

# Visualization of the simulation

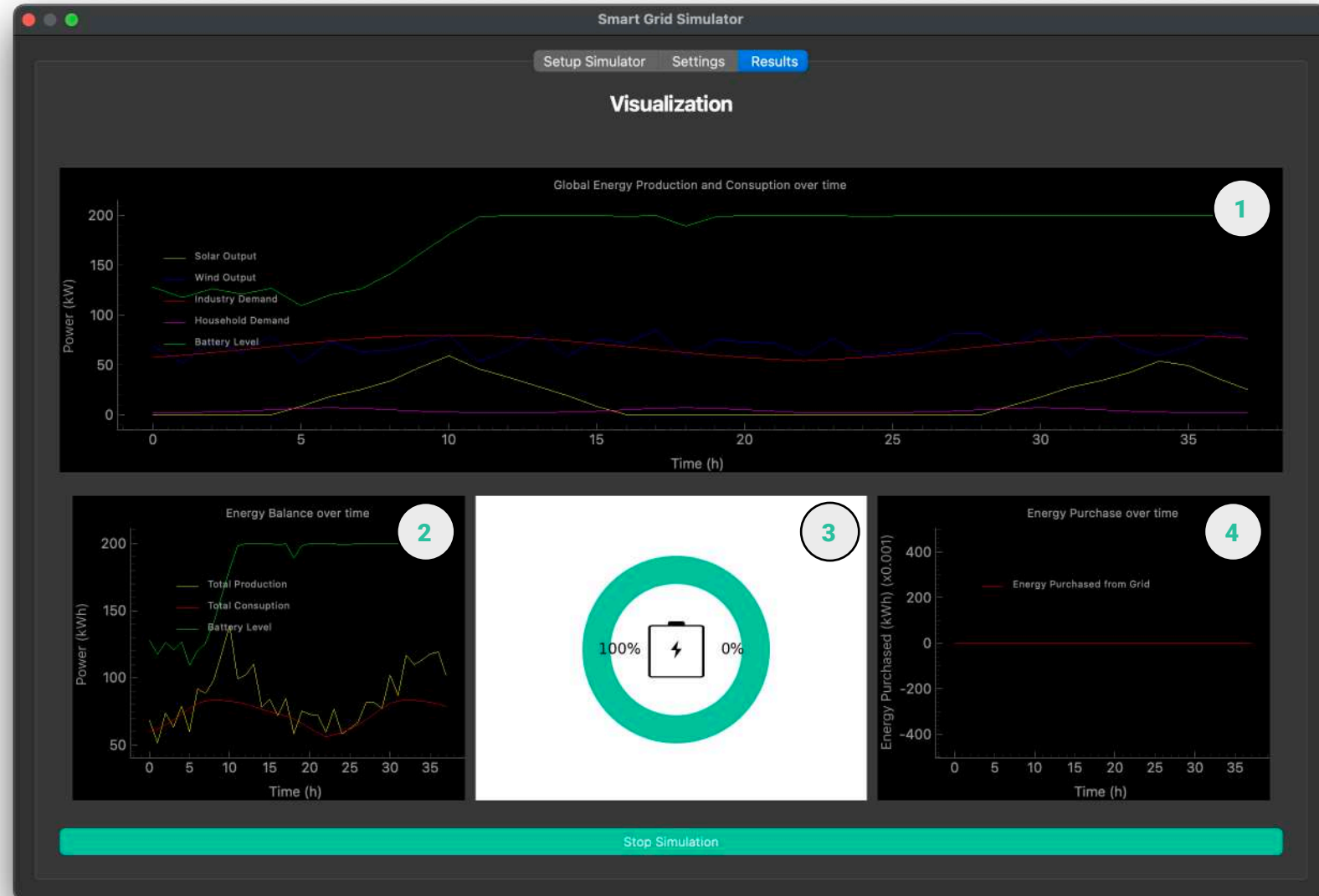
## Feature :

Visualize the real-time simulation.

## Description:

The app is configured to display an hour of simulation every second for demonstration purpose.

- 1 Shows the producers and consumers in detail and the level of the battery over time (every hour).
- 2 Shows the sum of the producers and consumers with the battery level over time. It shows directly the level of energy compared with the demand.
- 3 The level of the battery over time, the goal is to never be completely discharged.
- 4 Shows the amount of energy purchased over time



# Possible evolution

---

The application uses simple algorithms for :

- simulating the demand (household and industries)
- the production overtime (randomly for wind production and simple equation for solar production with a peak at noon)

Those algorithms could be more complexe or based on real data. For example, the solar production and the wind production could be based on real-life weather database.

The energy demand could be based on real data or machine learning model.