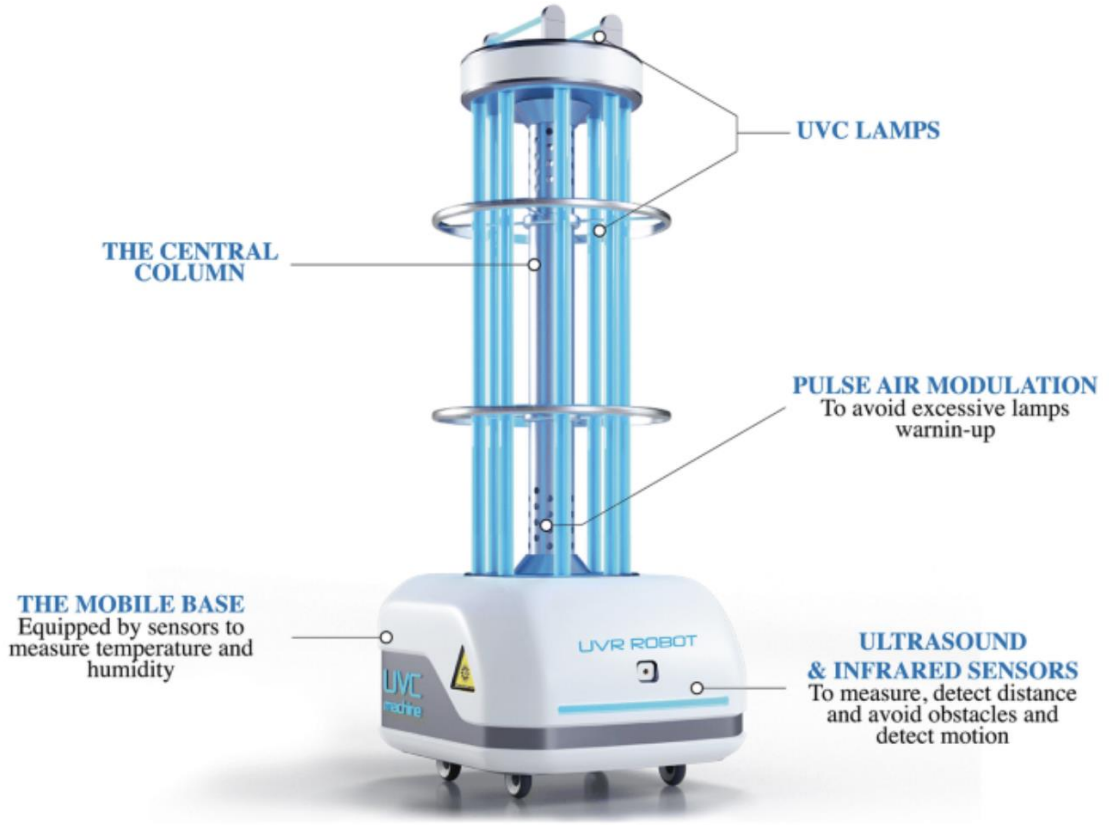
ROS 2

AUTONOMOUS AND MOBILE ROBOTICS



Sanitizer Robot Project

Group:

- Federico Fabbri
- Agatino Ricciardi

python

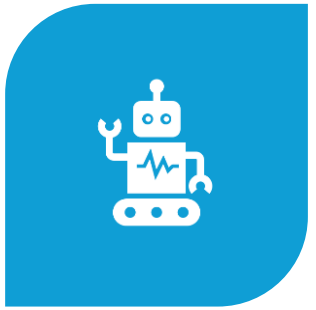
PROJECT SPECIFICS

TASK 1

SETUP OF THE TURTLEBOT3 IN THE GAZEBO BIG HOUSE ENVIRONMENT

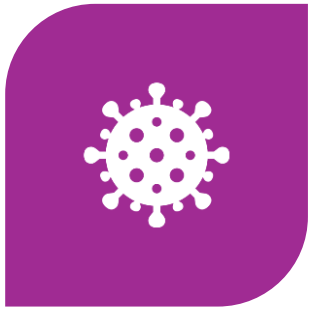
TASK 2

AUTONOMOUS EXPLORATION OF THE ENVIRONMENT TO CREATE A MAP



TASK 3

LOCALIZATION OF THE ROBOT AND NAVIGATION TO A SET OF GOALS



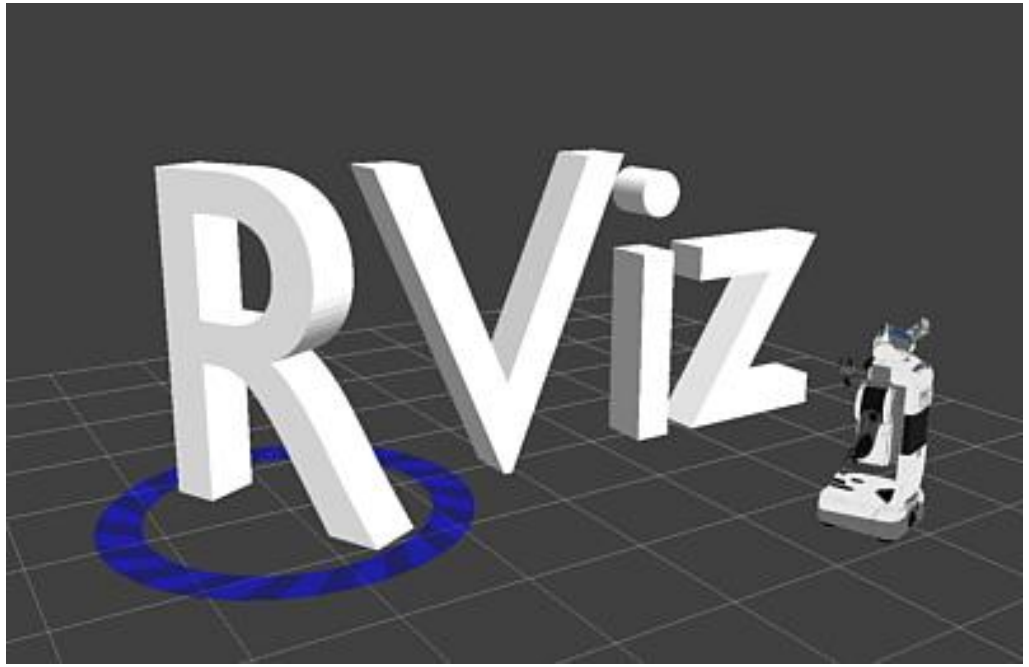
TASK 4

LOCALIZATION, NAVIGATION TO A SPECIFIC ROOM AND SANITIZATION

SIMULATION TOOLS

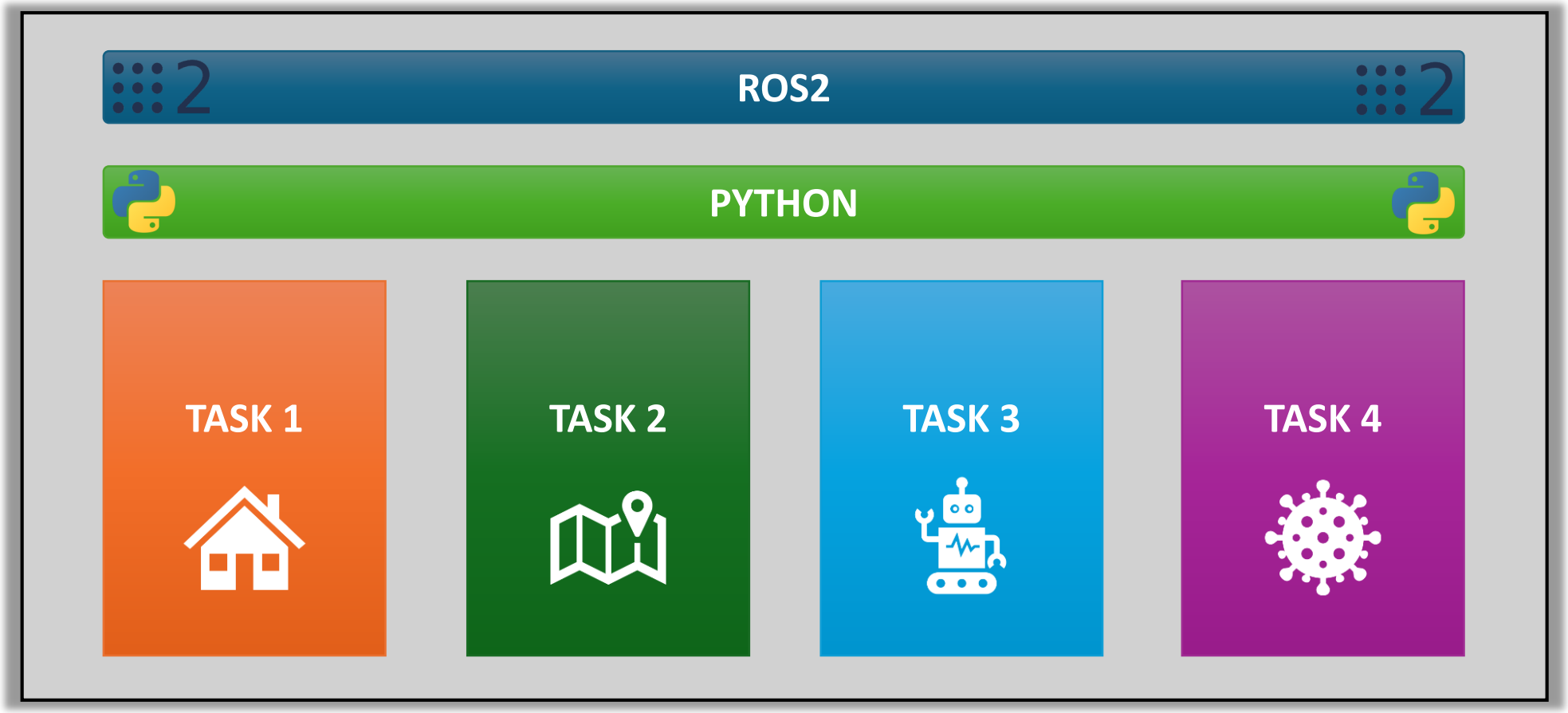
GAZEBO

Open-source 3D robotics simulator for research, design and development

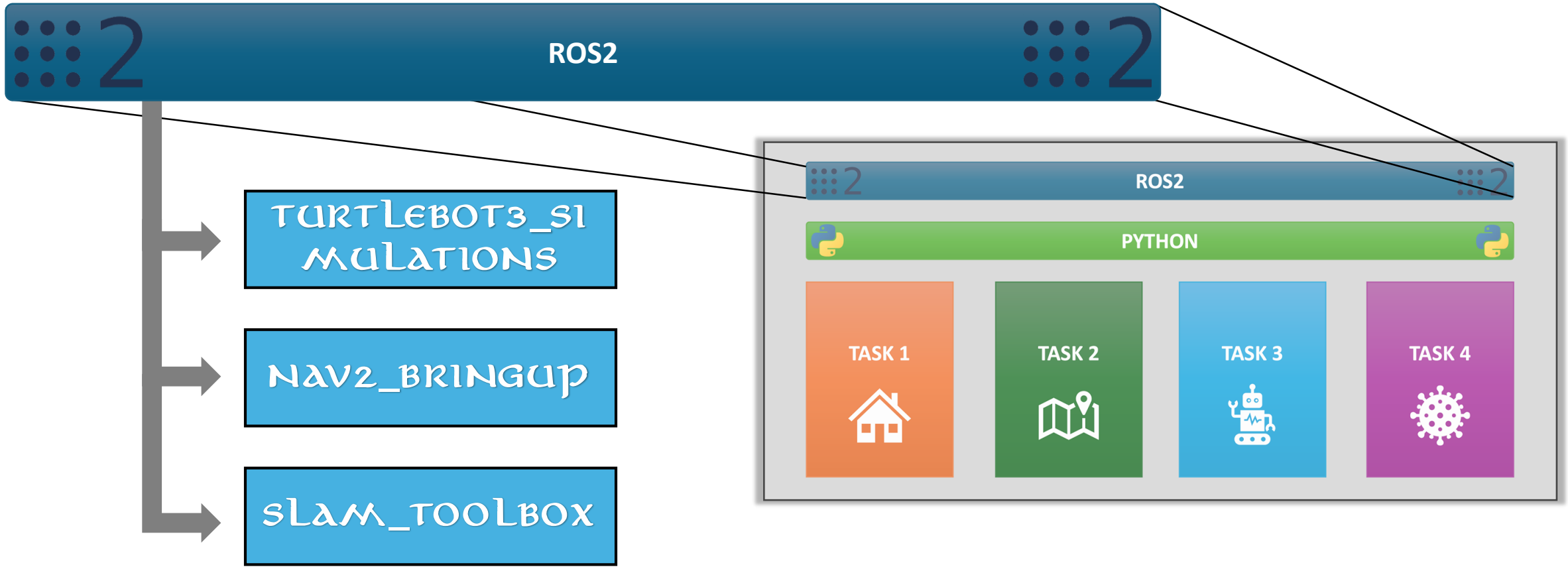


User-friendly visualization tool for ROS that allows exploration and data analysis in 3D

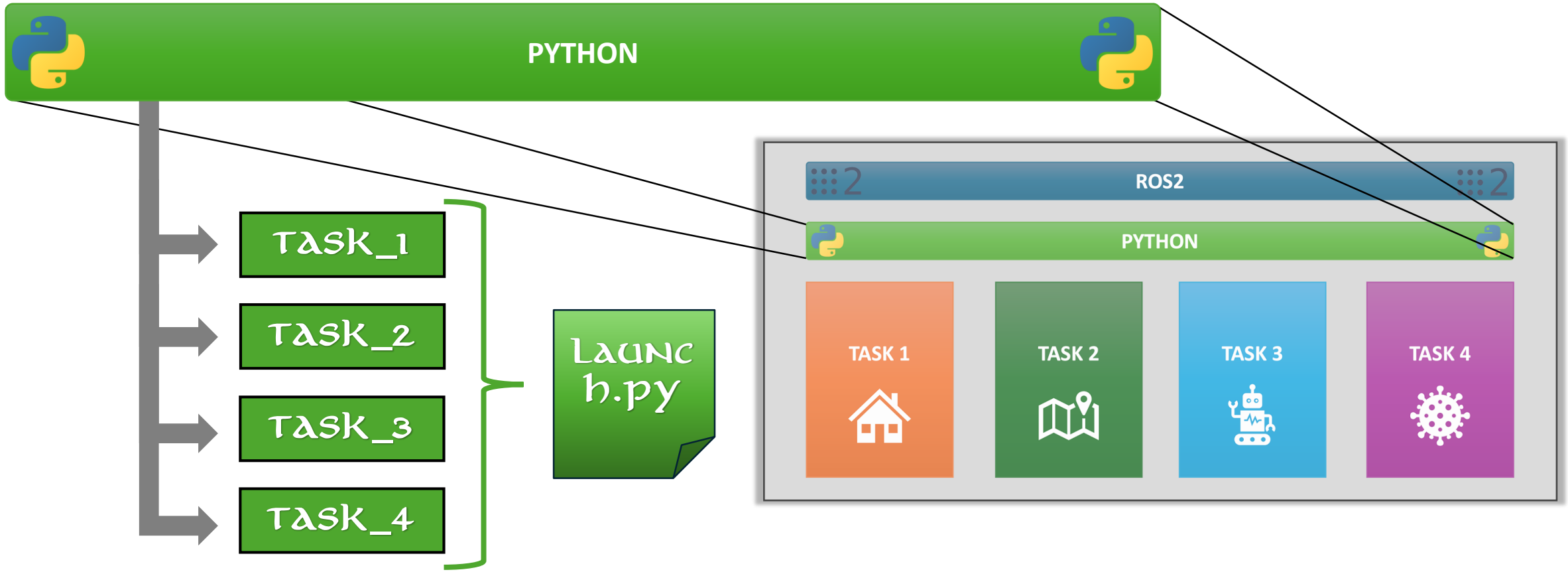
OVERALL PROJECT STRUCTURE



OVERALL PROJECT STRUCTURE – ROS2



OVERALL PROJECT STRUCTURE - PYTHON



LAUNCH FILES

TASK 1

task1.launch.py

- Gazebo Big House;
- Rviz2;
- Nav2 BringUp;
- SLAM Toolbox.

TASK 2

task2.launch.py

- M-Explore algorithm.

TASK 3

task3.launch.py

- Gazebo Big House;
- Rviz2;
- Nav2 BringUp.

task3.launch.py

- Localization task;
- Navigation task.

TASK 4

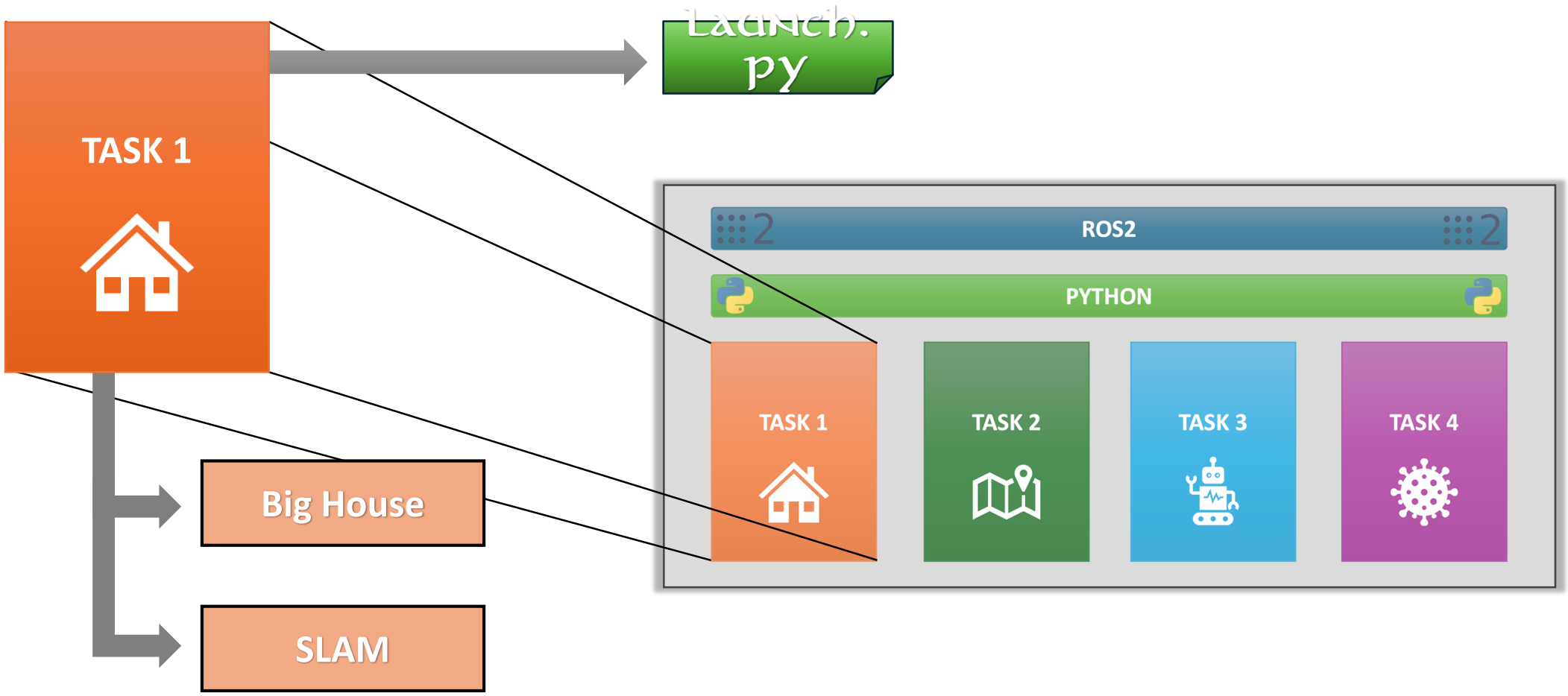
task4.launch.py

- Gazebo Big House;
- Rviz2;
- Nav2 BringUp.

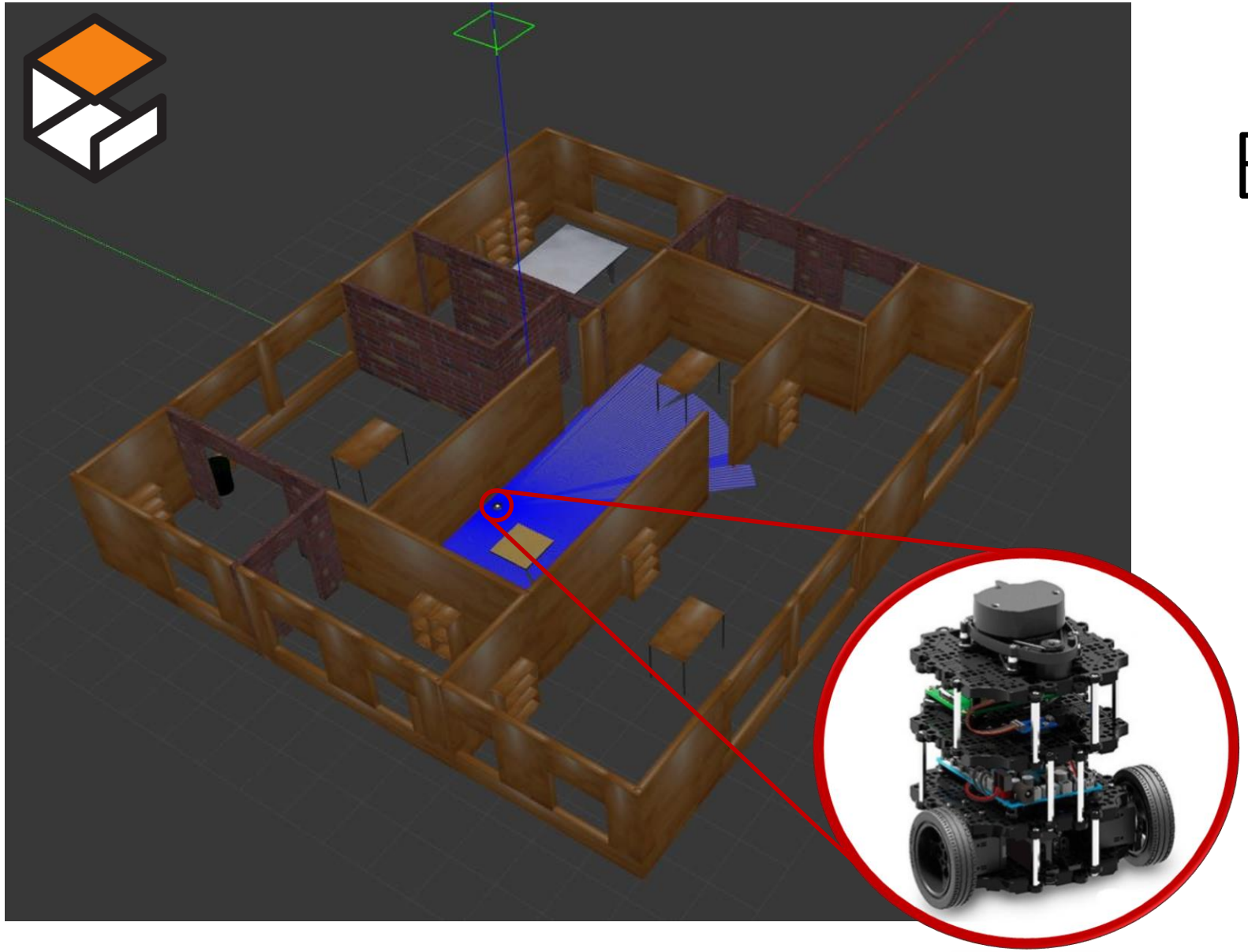
task4.launch.py

- Localization task;
- Sanification task.

OVERALL PROJECT STRUCTURE – TASK 1



BIG HOUSE ENVIRONMENT

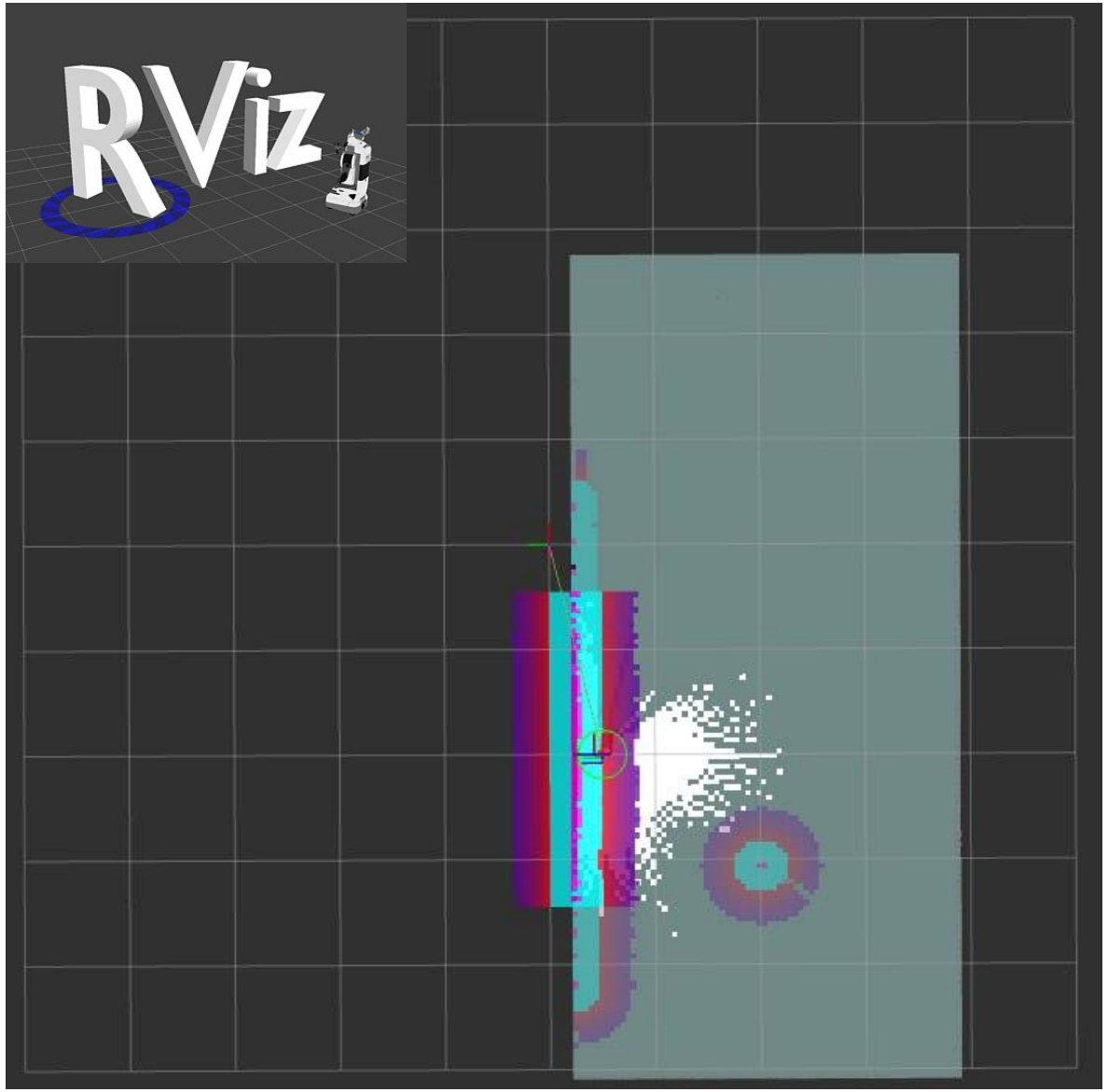


TURTLEBOT_3_MODEL = BURGER

TURTLEBOT3_BIG_HOUSE.LAUNCH.PY

NAV2_BRINGUP BRINGUP_LAUNCH.PY

Navigation network designed to help mobile robots move safely and efficiently through various environments and perform complex tasks. Provides perception, planning, control, localization, visualization and much more.



AUTONOMOUS SLAM

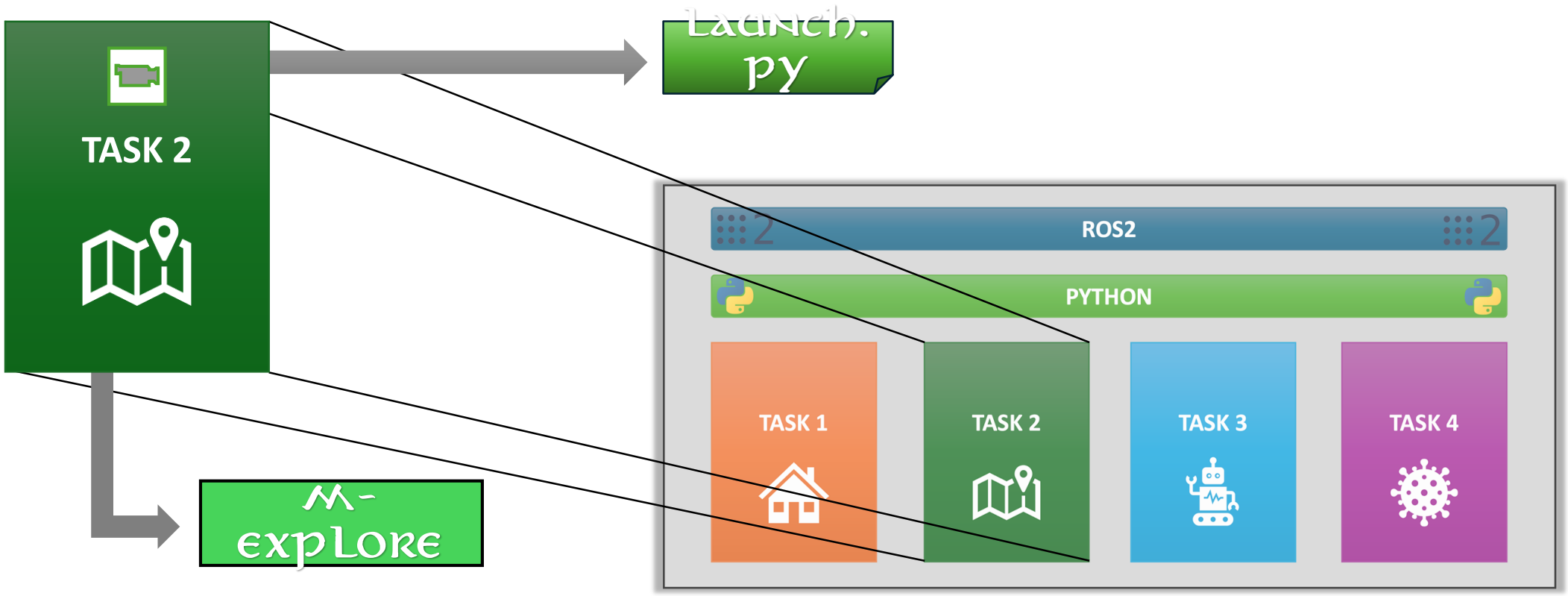
```
rviz2 -d /nav2_default_view.rviz
```

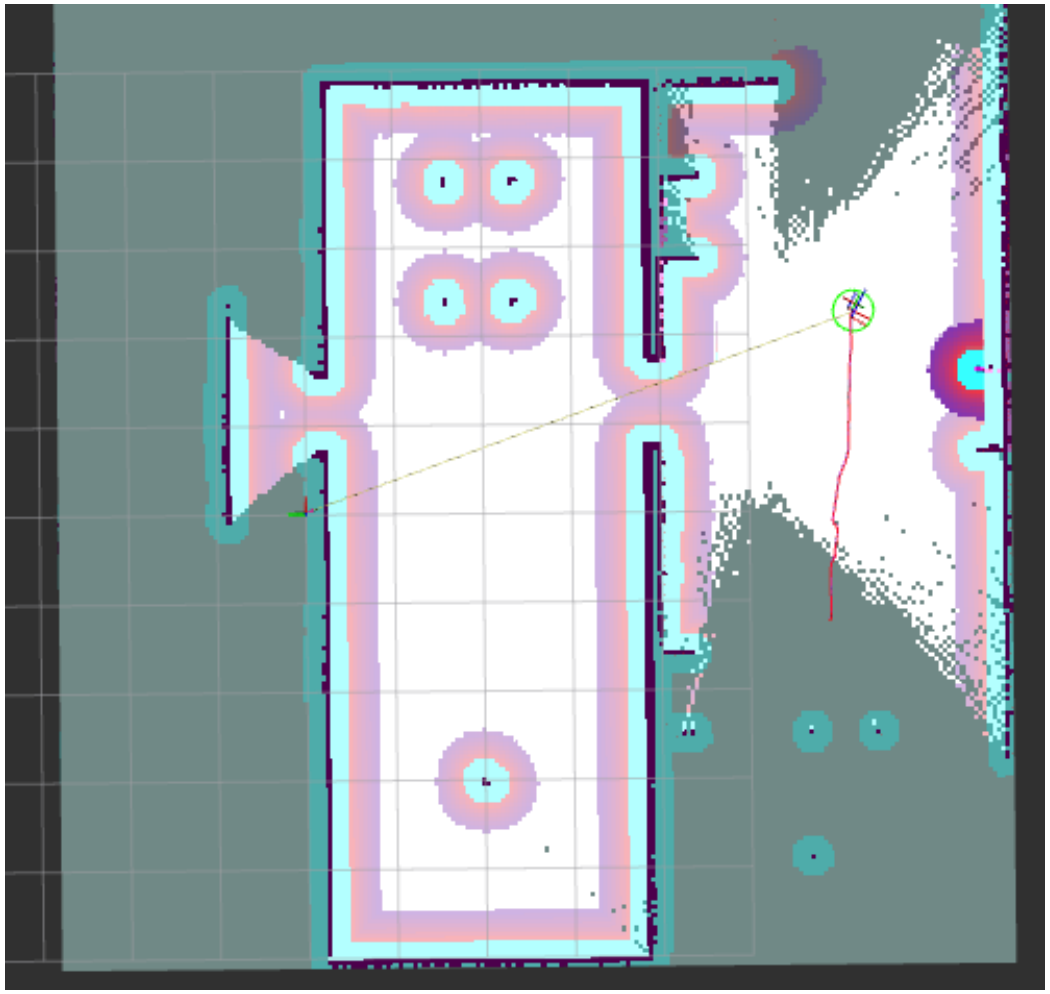
```
slam_toolbox online_async_launch.py
```

Simultaneous Localization and Mapping. Technique used to create a map of an unknown environment while simultaneously keeping track of the robot's location within it.

The async mode is recommended for online execution The mapping process is not synchronized with the robot's motion. It uses the latest available data to perform mapping.

OVERALL PROJECT STRUCTURE – TASK 2





M-EXPLORE LITE

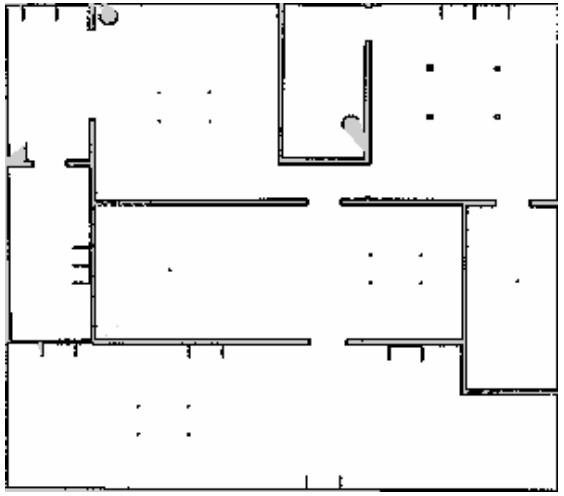
explore_lite explore.launch.py

ROS 2 package for multi-robot autonomous exploration.
It provides greedy frontier-based exploration, where the robot will greedily explore its environment until no frontiers could be found.

Modifications:

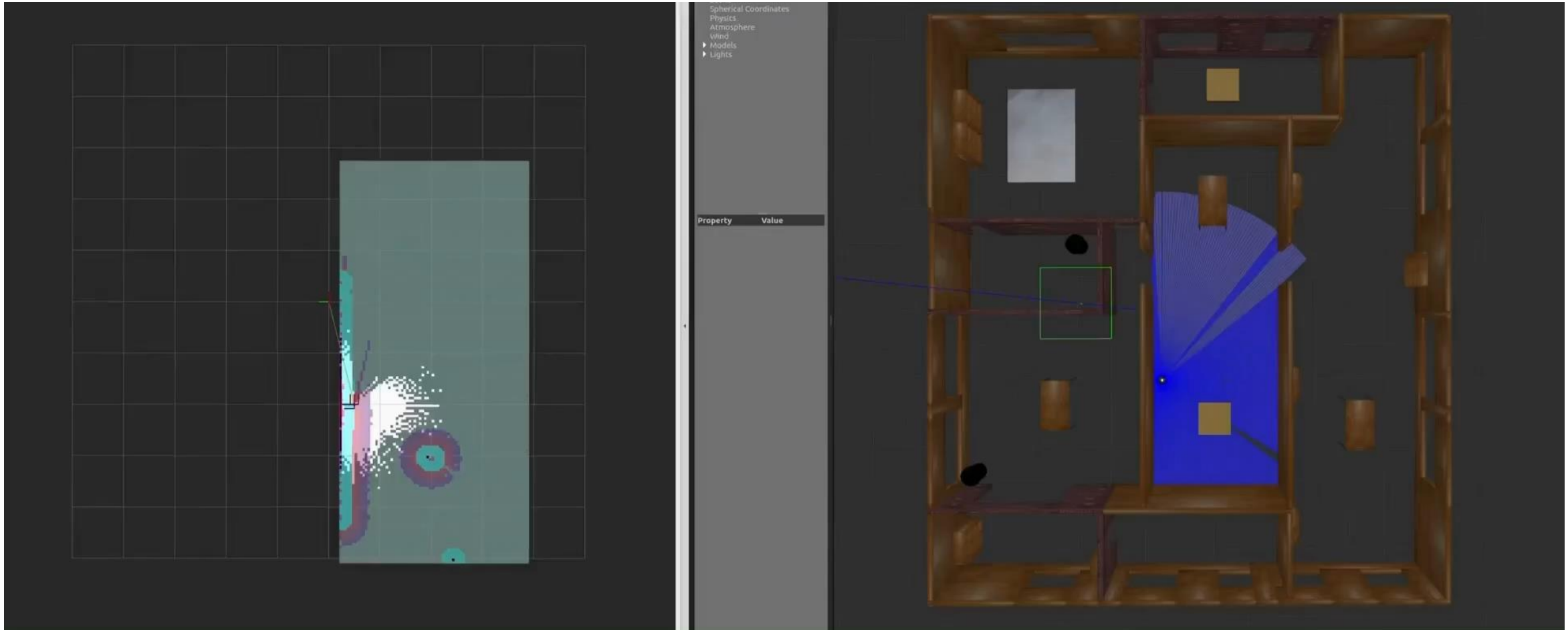
- min_frontier_size[1]: REDUCED
- progress_timeout[1]: REDUCED
- inflation_radius[2]: INCREASED
- lidar rays max range[3]: INCREASED

nav2_map_server map_saver_cli

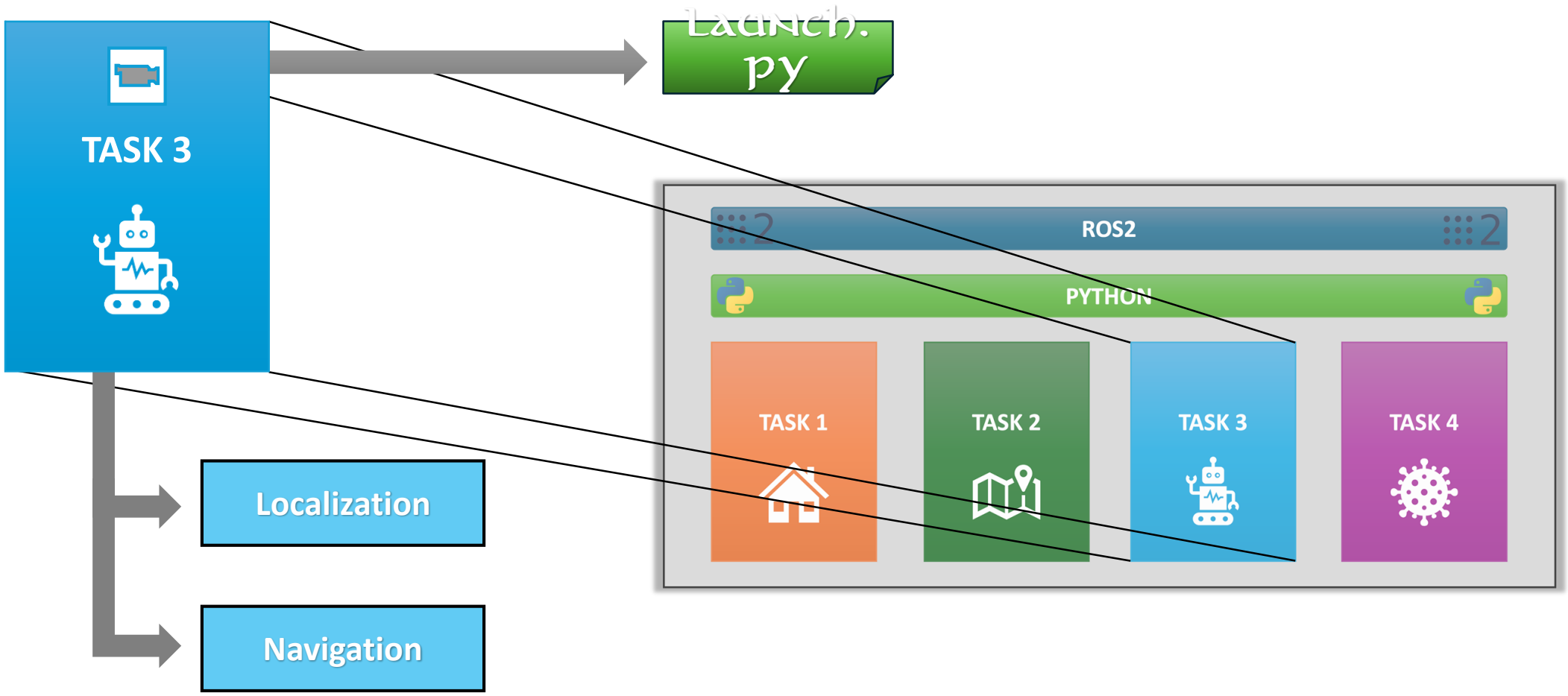


- [1]inside **explore_lite** package, in **params.yaml**;
- [2]inside **task_1**, in **nav2_config.yaml**;
- [3]inside **turtlebot3_gazebo**, in **model.sdf**;

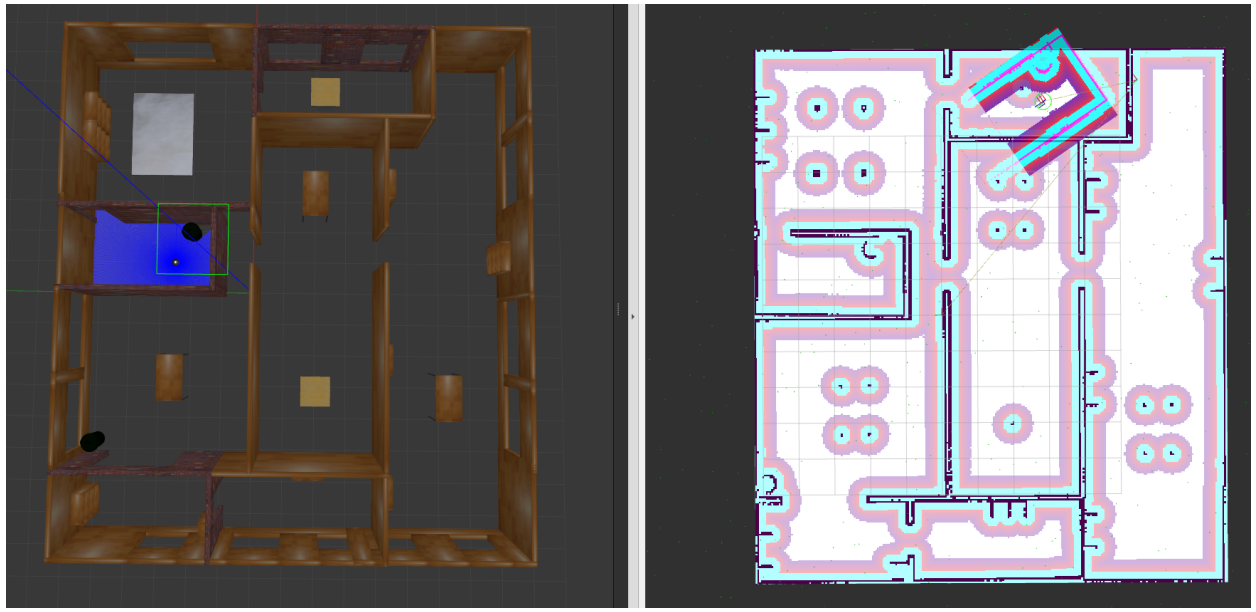
TASK 2 – VISUALIZATION



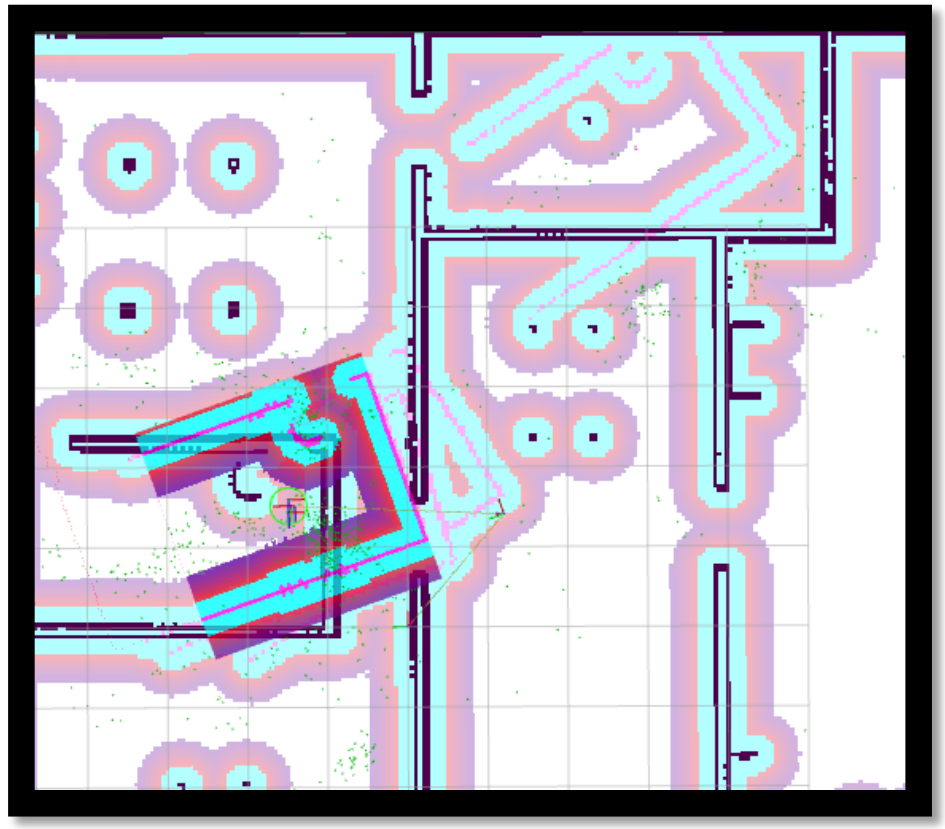
OVERALL PROJECT STRUCTURE – TASK 3



LOCALIZATION



task_3 localization.py

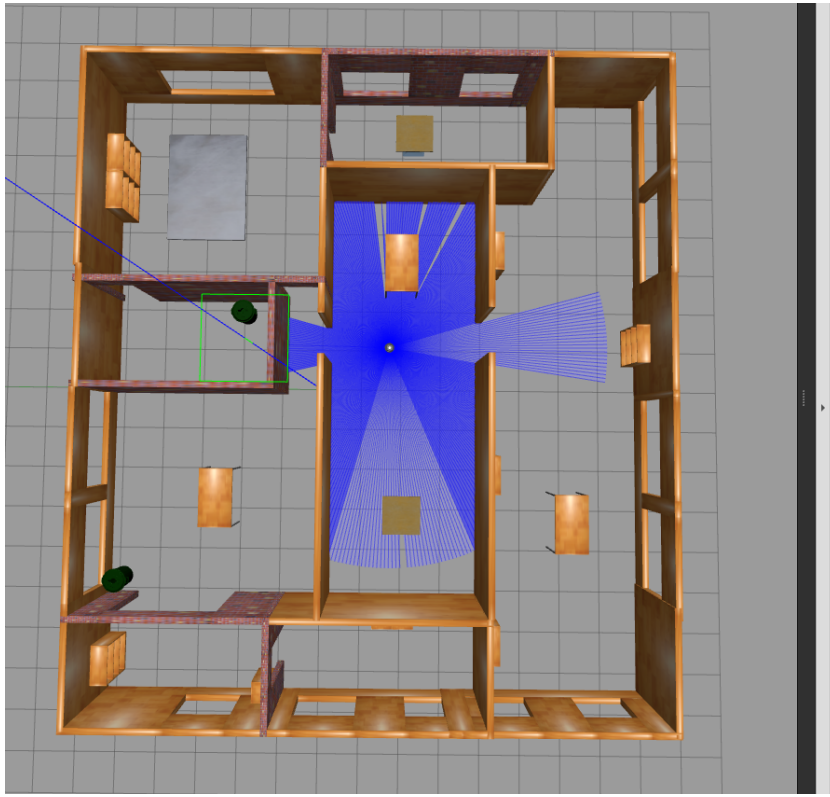


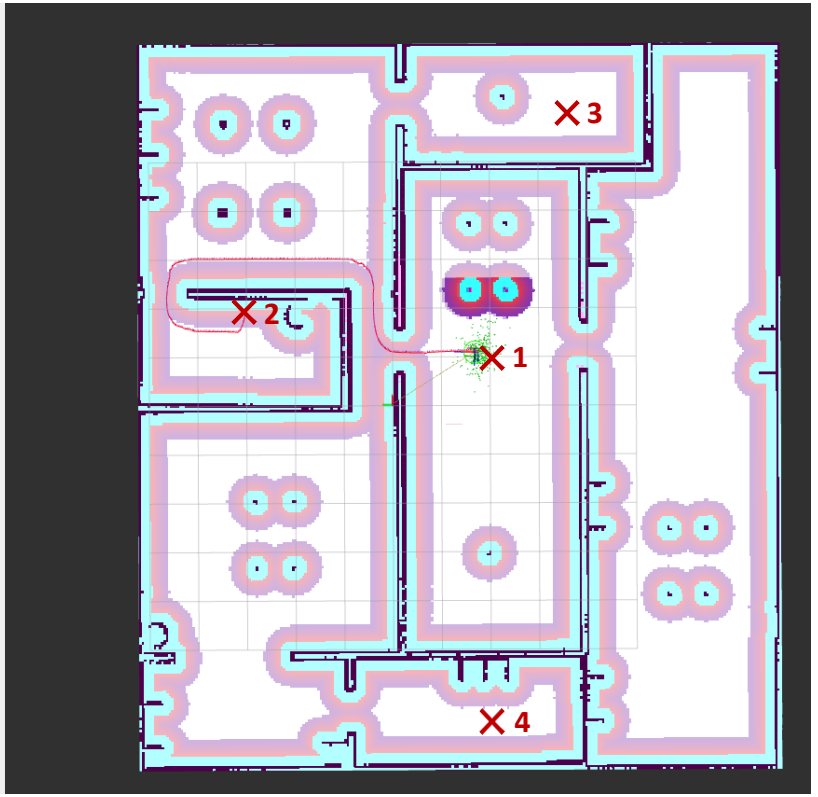
Process is done via Adaptive Monte Carlo Localization (AMCL), a probabilistic localization method for robots moving in 2D. Tracks the pose of a robot inside a known map using a particle filter.

Our solution implements a simple procedural wall follower algorithm to help the localization process. Process ends when the check on the eigenvalues of the covariance matrix is satisfied.

Modifications:

- Number of particles of the AMCL:
  INCREASED – in **nav2_config.yaml**;





NAVIGATION

task_3 route_manager.py

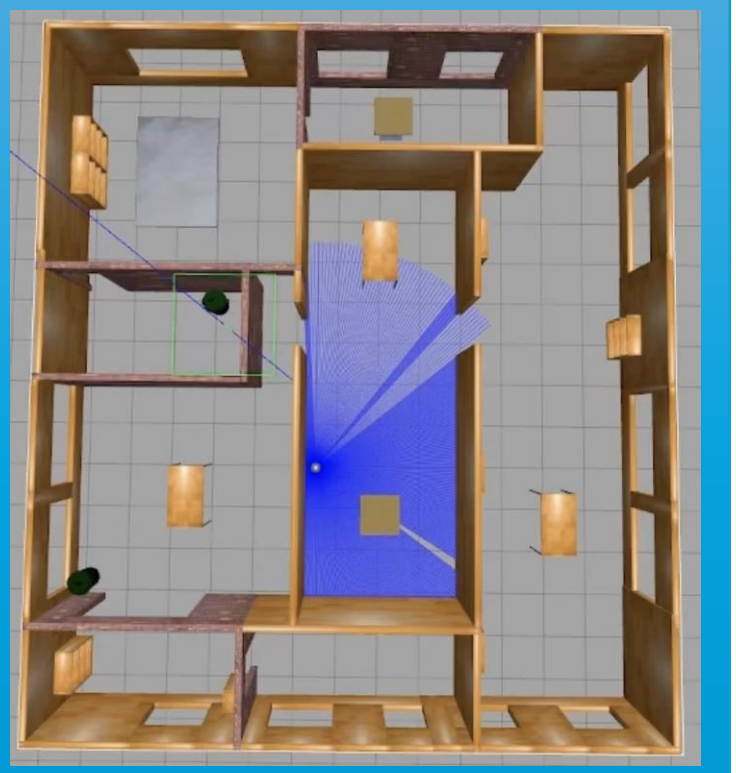
routes.yaml

```
mode: inorder
poses:
  - pose:
      position:
        x: 1.0
        y: -2.0
        z: 0.0
      orientation:
        x: 0.0
        y: 0.0
        z: 0.70
        w: 0.70
```

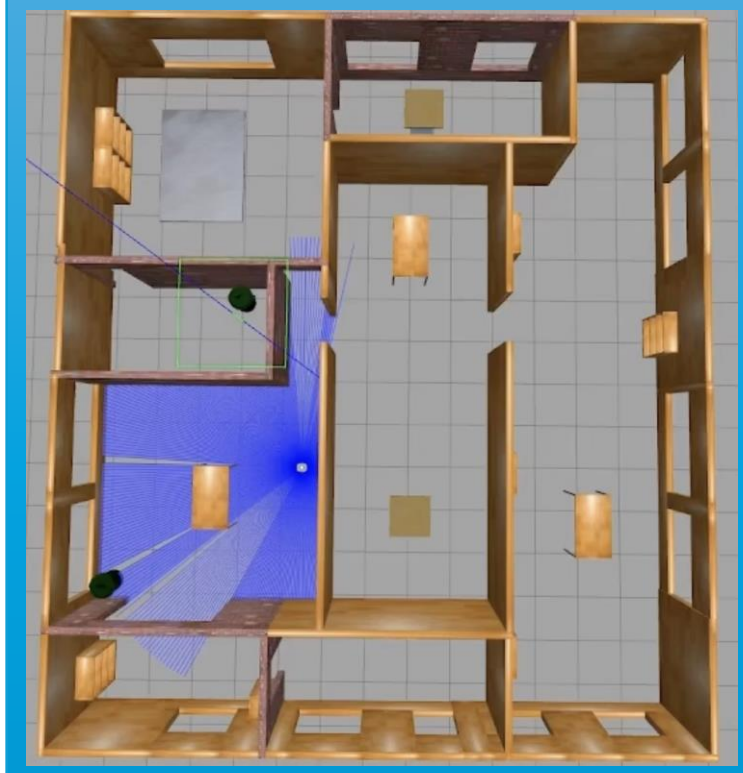
Navigation is synchronized with localization. Both processes are launched at the same time, communication between them is done via localization_callback.

Once the localization is completed, navigation starts following the route_manager.py logic, reading the goals inside the routes.yaml file and moving towards them.

TASK 3 – VISUALIZATION



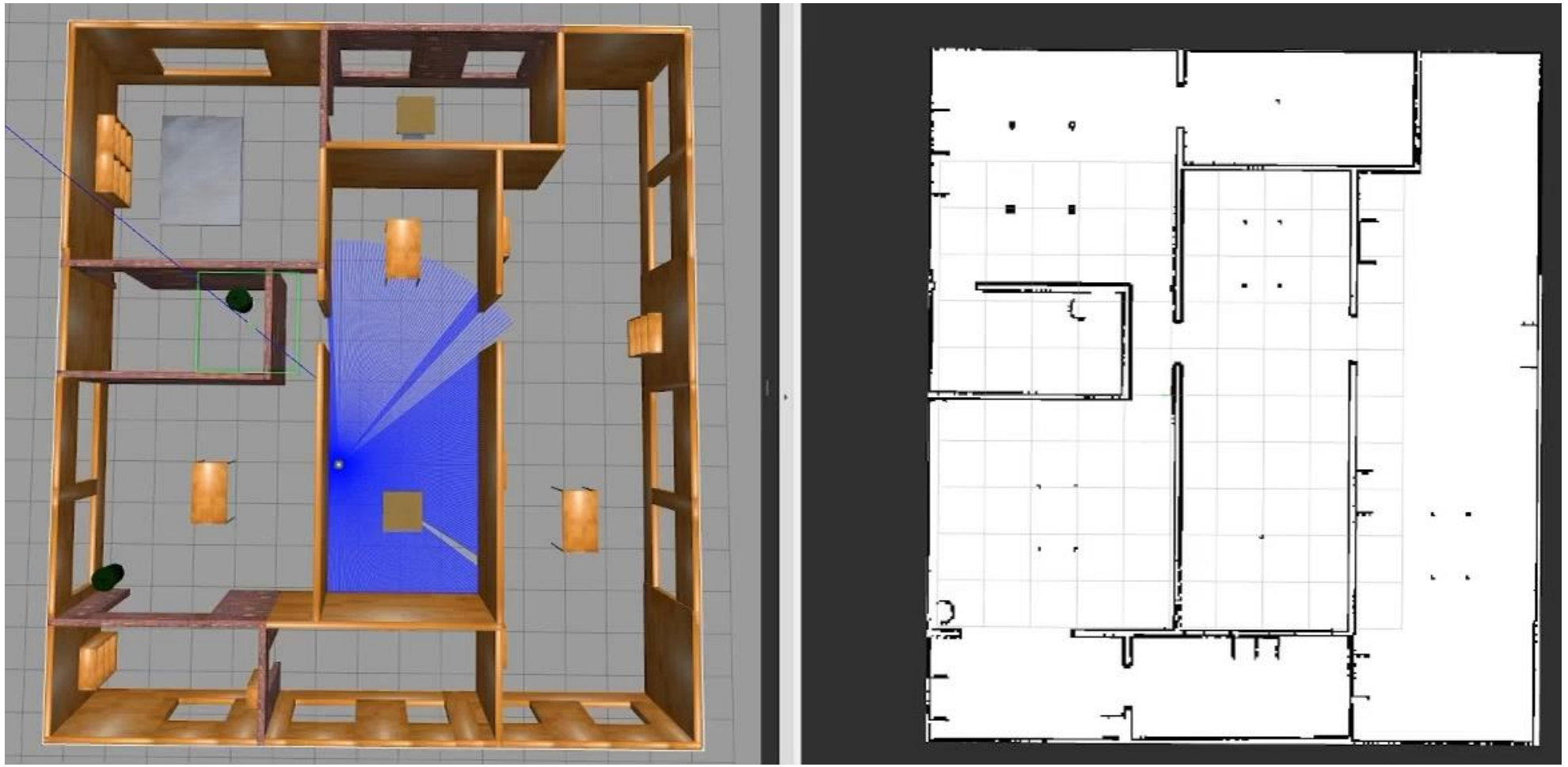
- Initial position: (-2, -0.5)



- Initial position: (-2, 0.5)

- Initial position: (1, 2.5)

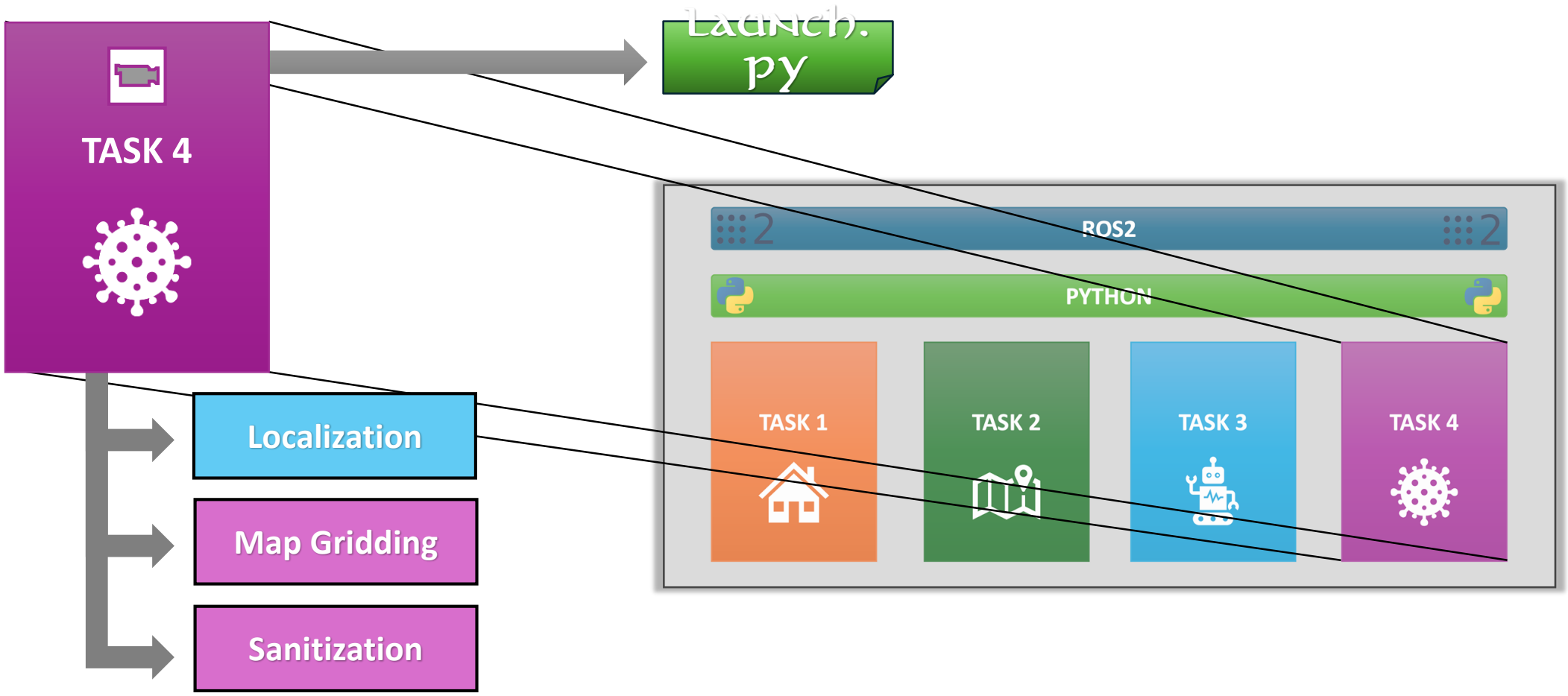
TASK 3 – VISUALIZATION 1

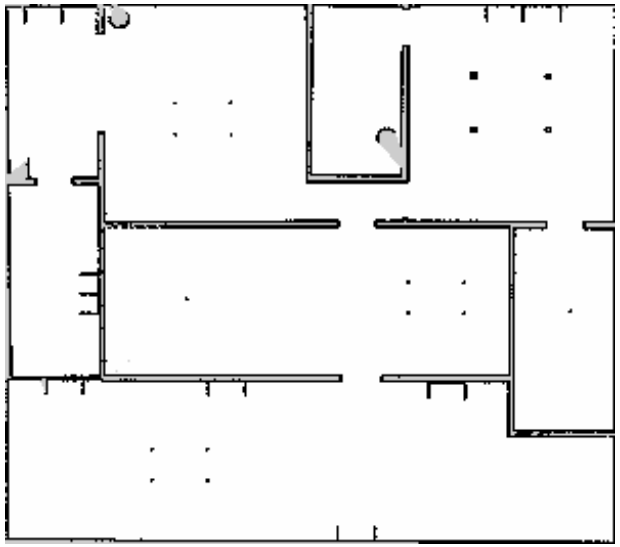


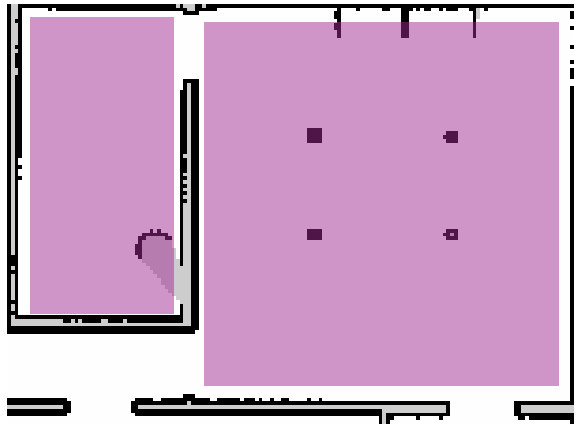
TASK 3 – VISUALIZATION 2

TASK 3 – VISUALIZATION 3

OVERALL PROJECT STRUCTURE – TASK 4





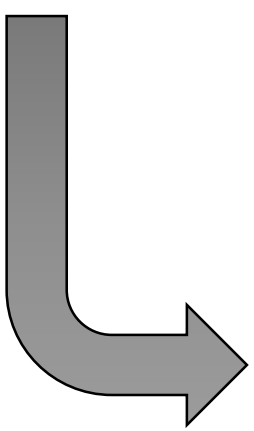


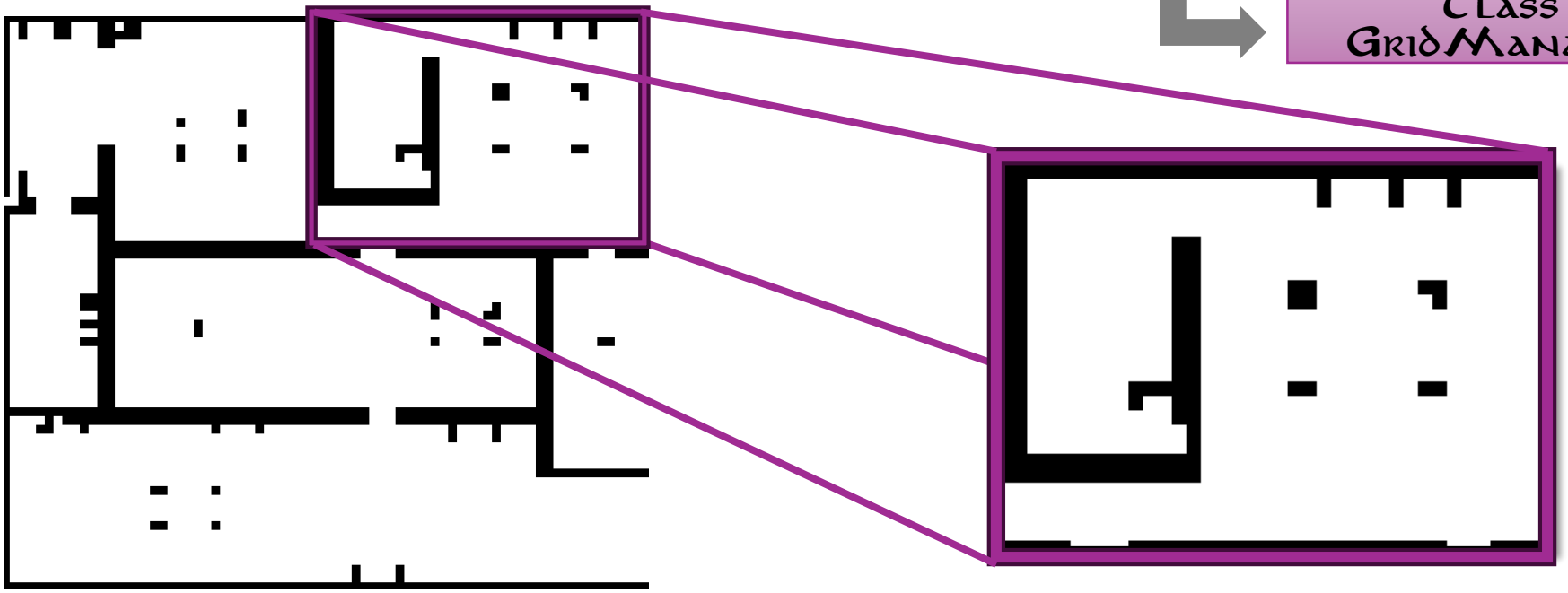
MAP GRIDDING

task_4_manager.py

Class MapManager

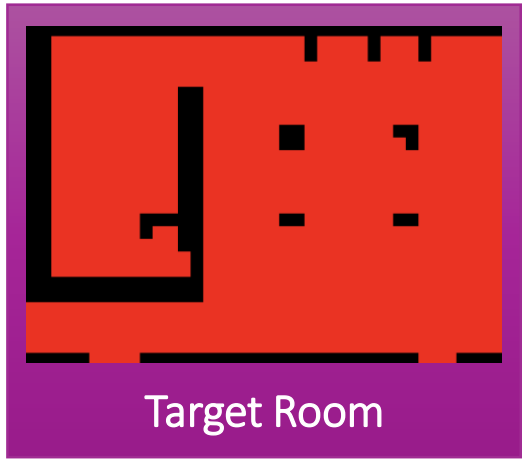
Class GridManager





SANITIZATION

- Energy Evaluation
- Cells Sanitization
- Goal Search & Navigation



Target Room

task_4 sanitizer.py

The sanitizer.py calls instances of the classes of the manager.py to perform

task_4 manager.py

- Class MapManager
- Class GridManager
- Class RoutePlanner

Sanitization is synchronized with localization. Both processes are launched at the same time, communication between them is done via localization_callback.

Once the localization is completed, sanitization starts following the sanitizer.py logic, reaching the target room and starting the sanitization process.

ENERGY EVALUATION

$$E(x, y, k) = \sum_{i=0}^{k} \frac{P_l \Delta t}{(x - p_x(i\Delta t))^2 + (y - p_y(i\Delta t))^2}$$

Data:

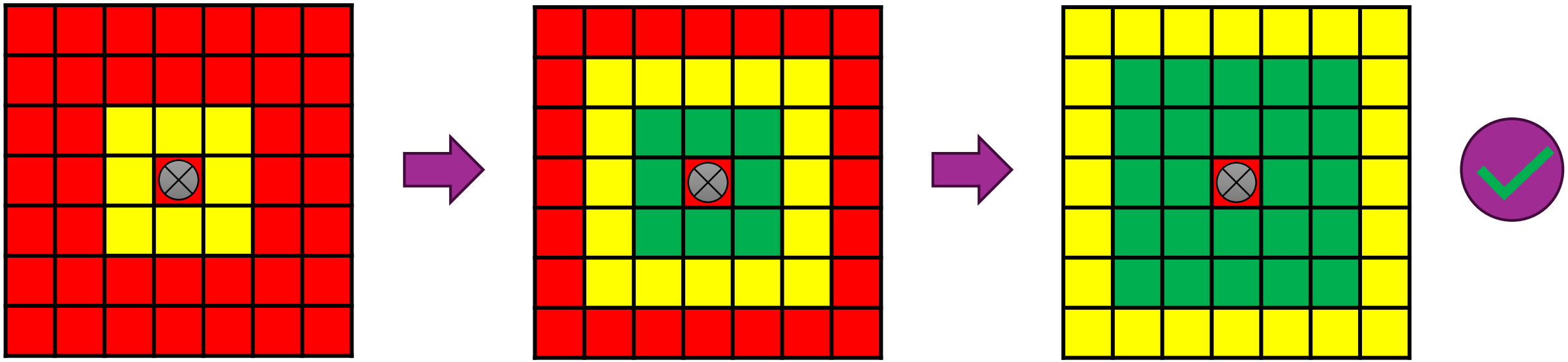
- ❑ Light Power: $P_l = 100 \mu W m^2$
- ❑ Cell position: $(x,\ y)$
- ❑ Robot position: $(p_x,\ p_y)$

× Cell Not Sanitized: $E \leq 5\ mJ$ → 🟥

➢ Cell Half-Sanitized: $5\ mJ < E < 10\ mJ$ → 🟨

✓ Cell Sanitized: $E \geq 10\ mJ$ → 🟩

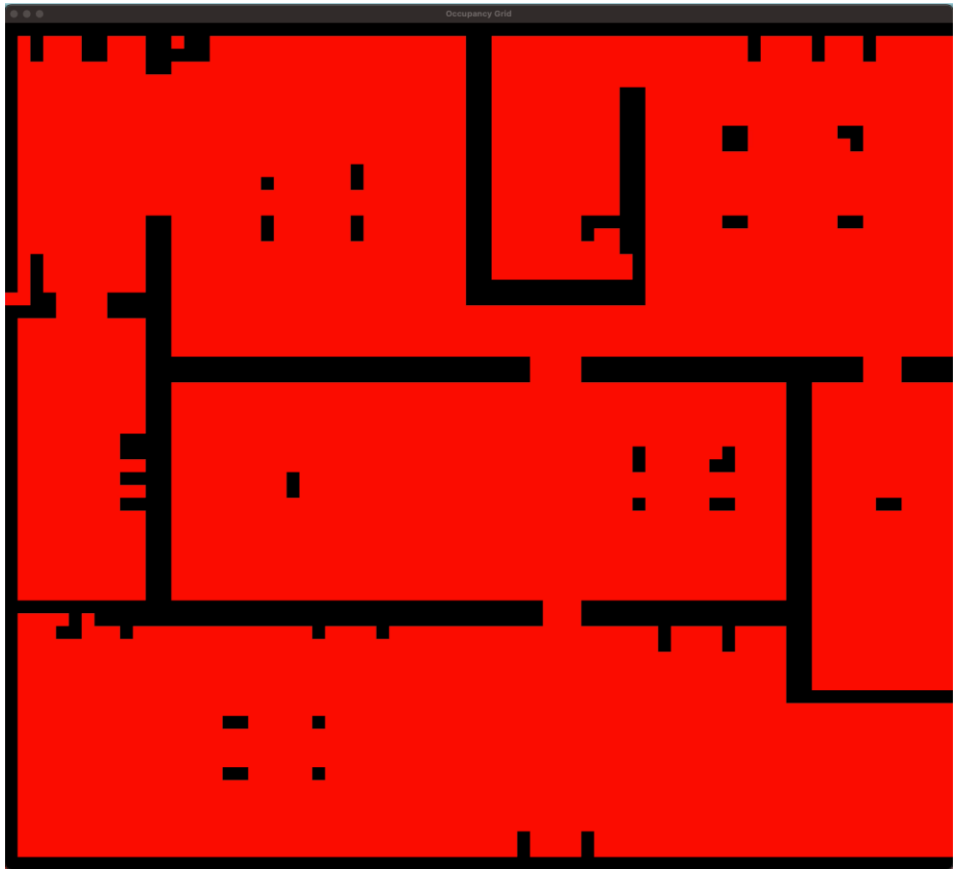
CELLS SANITIZATION



- Neighbourhood Sanitization: it can be modified to let the robot stay in current cell more/less;
- Robot Encumbrance: robot doesn't sanitize in place.

- **Ray Tracing**

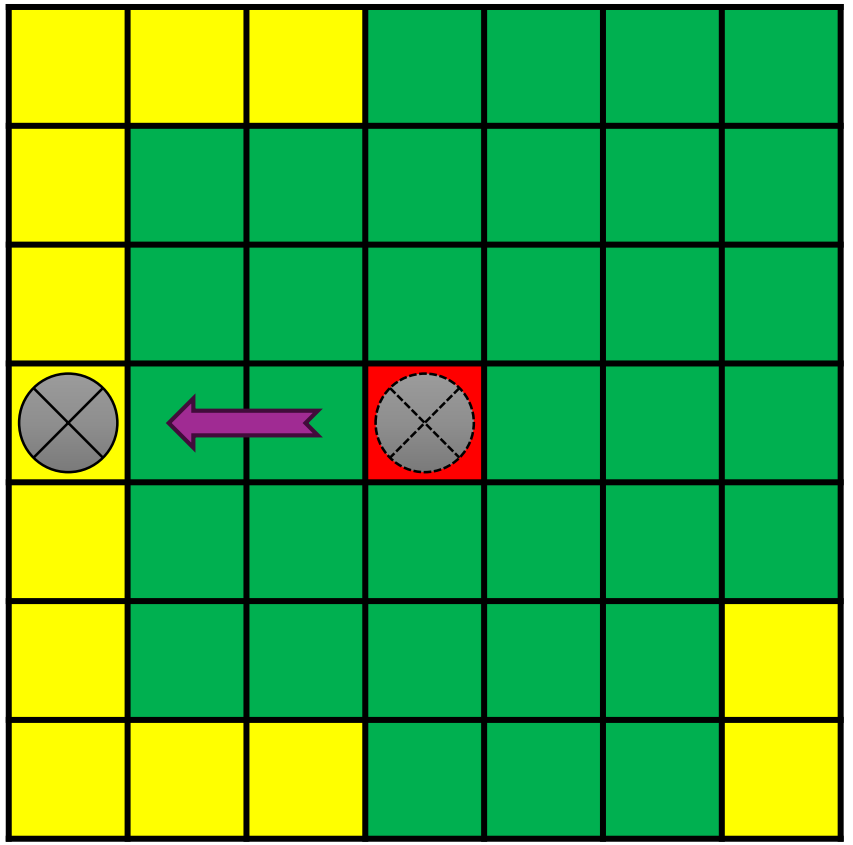
RAY TRACING – VISUALIZATION



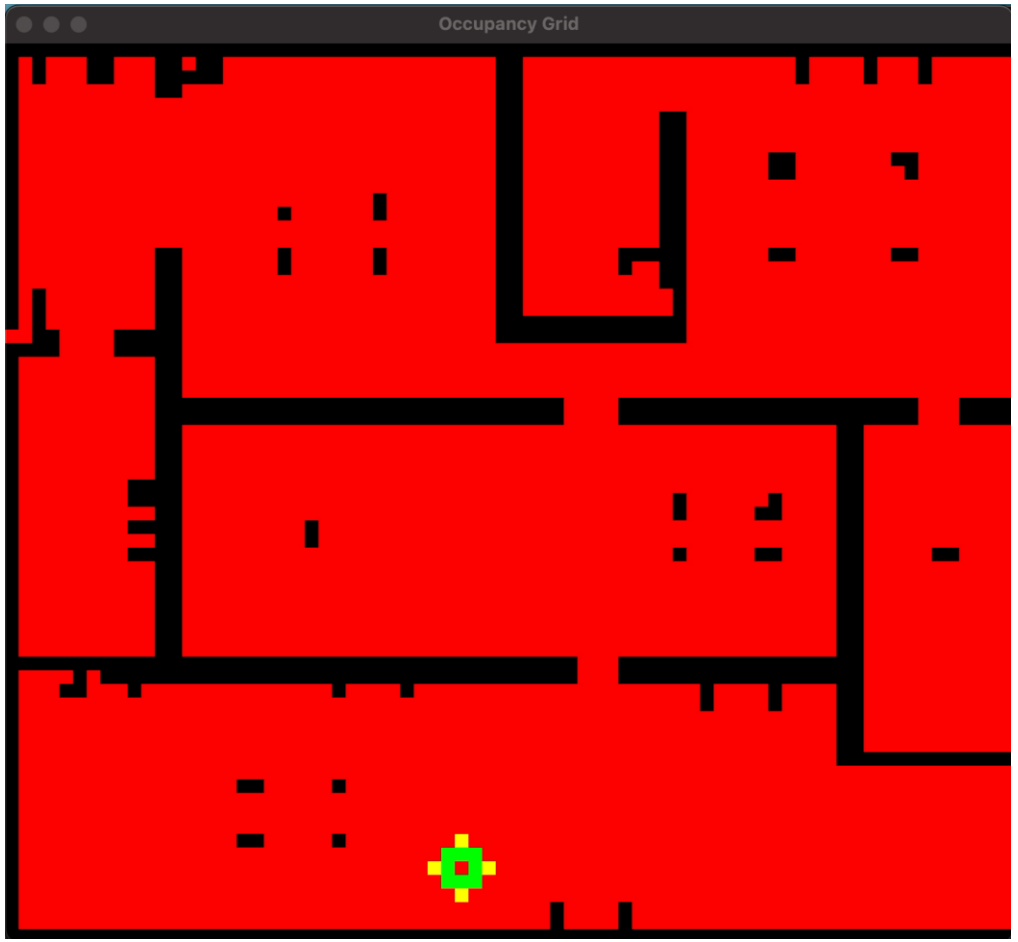
GOAL SEARCH & NAVIGATION

- **Neighbourhood Check**: when the robot can only see sanitized tiles surrounding it, starts to search for a new cell;
- **Find Closest Target**: the robot looks for the closest half-sanitized cell;
- **Navigation to Goal**: the robot starts to move to reach the defined goal;
- **Sanitization Callback**: once the new goal is reached, the sanitization restarts.

- **Policy Simulation in Big House**



BIG HOUSE POLICY – VISUALIZATION



TASK 4 – VISUALIZATION