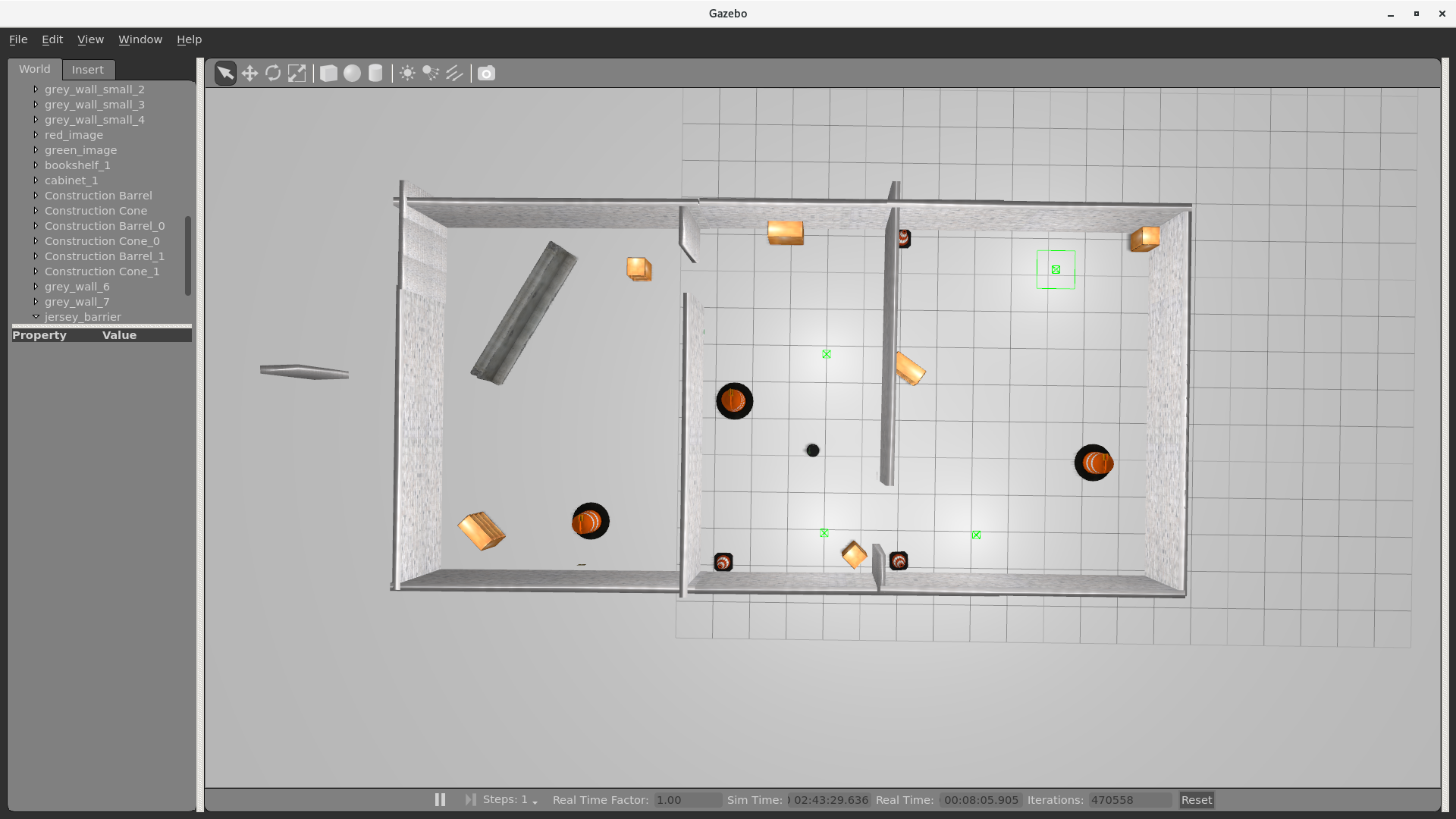
Implementation of a new simulated environment

We implemented a new simulated environment to test the robustness of our programme. We designed the new environment by changing the location and orientation of the existing objects and adding new objects. The result is an environment where the starting room (where the default location of the robot is) is in the middle between two other rooms. The green room and red room are twice as large as the rooms provided in the example, which increases the difficulty for the programme to find the cluedo image because the image is now further away from the centre point. We also deliberately placed the cluedo image behind a cone barrier, making it out of direct line of sight after the robot reaching the room. We specifically designed a path strategy which instructs the robot to go in a pattern that resembles an asterisk shape. That way our robot can bypass the obstructions placed between the robot and the cluedo image.

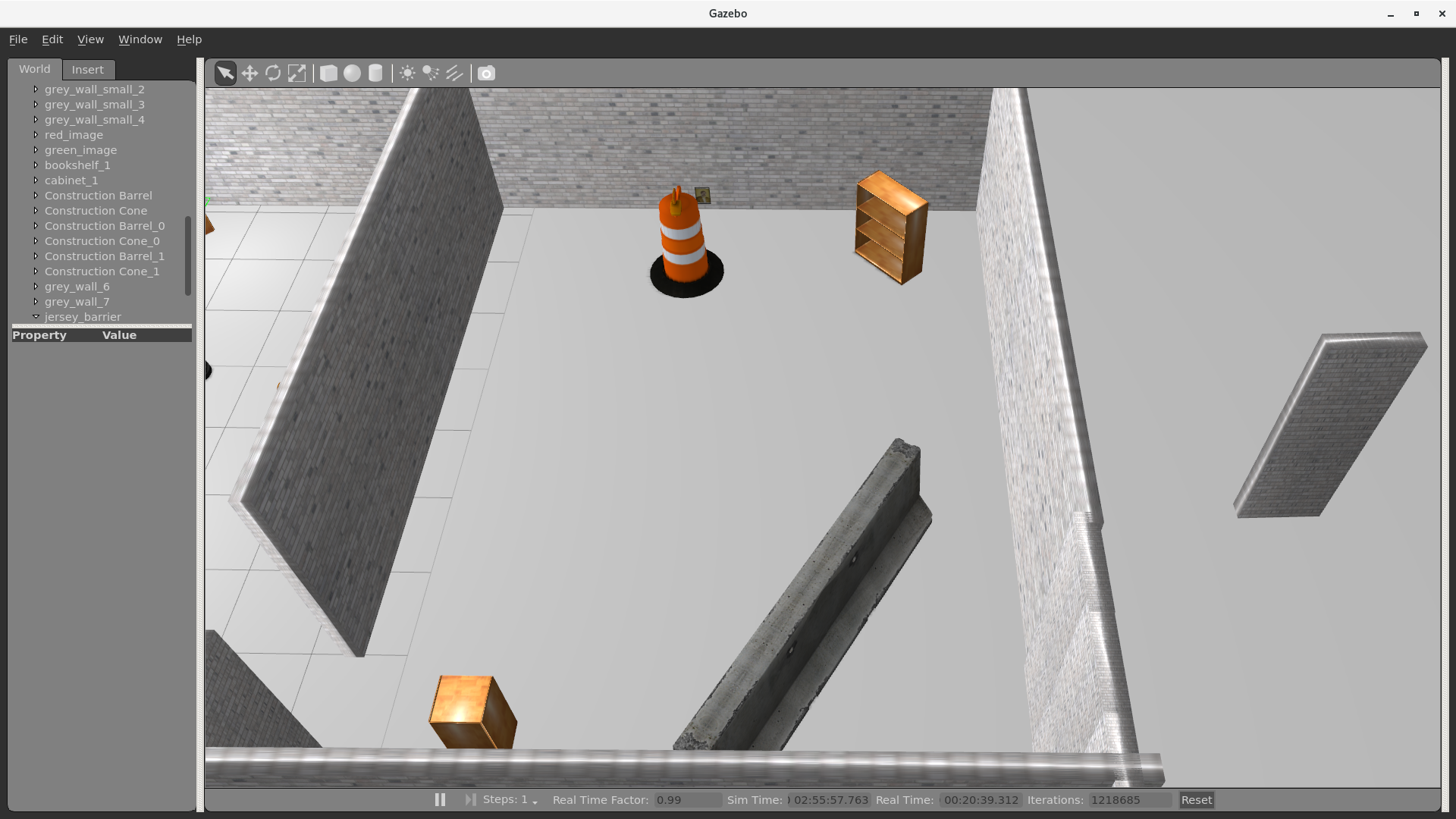
Following are a few screenshots showing the new environment.

Overview of reconstructed environment:



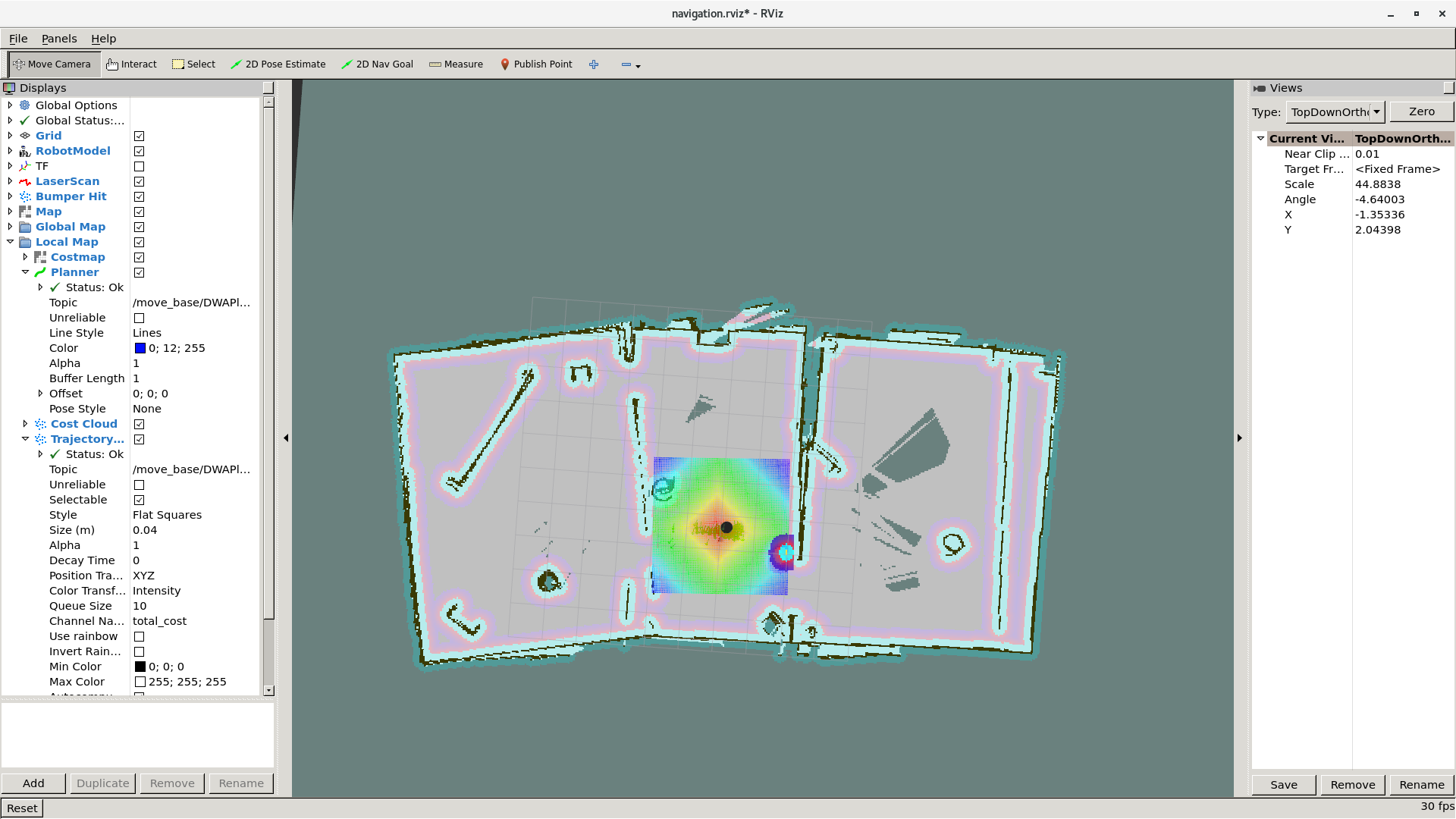
Starting point of the robot is next to the cone barrier in the room in the middle.

Overview of the green room (cluedo is at the back of the room, behind the cone barrier)



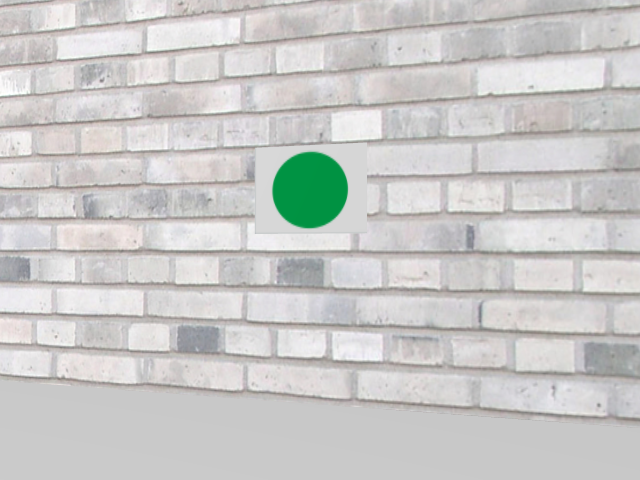
To make the robot run in the simulated environment, we also need to create a map such that the robot can navigate itself. We ran the mapping function and manually walked the robot through all rooms, thus obtained a map of the environment.

Sensor map of the new environment:

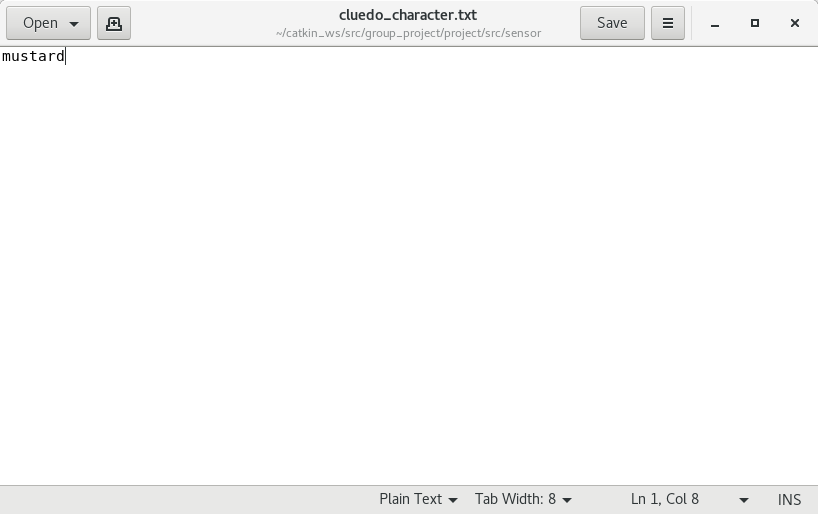


Output of the programme after running in the new simulated environment:

Sensor image of the green circle and cluedo image:



Cluedo\_character.txt



The programme identified the cluedo image as mustard, which is correct.

Testing in real environment

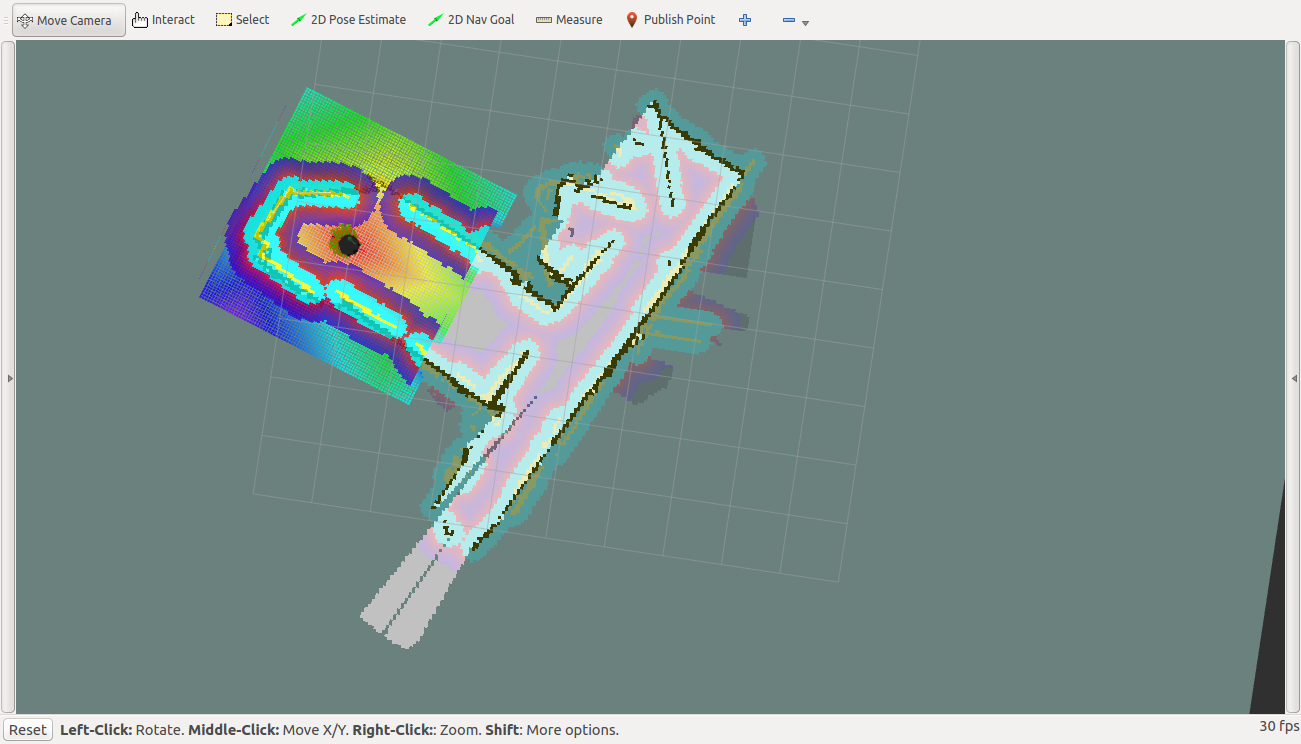
We tested our programme in the rooms provided in Logik space.

Before testing the programme on real robot, we need to map the real environment, obtain the four coordinates (room1 entrance, room 2 entrance, room 1 centre, room 2 centre), and make sure the robot can move to any given point on map as instructed.

It’s also important that our programme can identify green circle, red circle and cluedo images in real environments without confusing them with other objects. Due to different lighting situation, sensor configuration and colour differences of image on paper etc, we need to re-adjust the filter boundaries by testing them against real images on paper.

After these preparations, our programme should be able to work in real environments.

The sensor map we created using the mapping function (robot is in the green room in the picture):



The four points were obtained by “publish points” function provided by rviz, and validated by putting them in go\_to\_specific\_point\_on\_map.py provided in Lab4

We ran the programme in the real environment and recorded the whole process. Please access our recording at: https://youtu.be/hChltdCDT2w

The programme successfully identified red circle at the entrance of the first room, and then decided to go to the entrance of the second room. After spinning for two rounds, the programme found the location of the green circle and decided to go to the centre point of the second room. At the centre of the second room, the programme managed to identify the cluedo image and terminated execution.

Green circle captured at the entrance of the green room.



Cluedo image captured after reaching the centre point of the green room



The robot successfully identified the circles at the entrances and went to the correct room, captured the images which completely contains the green circle and cluedo image. The running time was within reasonable range. Therefore, the overall performance and accuracy was up to expectation.