LOCATION PROBLEM

NJRB2 | MVS9 | FA296

# Problem Description

The drone should aim to locate the Roomba, using any specified features, and successfully guide the Roomba to the pen. The pen will be a small area within the predefined arena. To complete the project’s aims, we should be able to drive the Roomba through messages sent from our side. We are not allowed to contact the Roomba using the drone.

One problem regarding the location part is having to search the specified area for the drone. Complications can include the drone missing the Roomba as it searched the perimeter, as well as the drone flying over the boundaries of the zone (which is not allowed). Furthermore, we also must make sure that the Roomba does not cross said boundaries. In case it happens, the mission has failed.

Another problem is the following action after the drone has successfully found the location of the Roomba. Due to the movement of the Roomba, it may be difficult to follow the it with no latency. Therefore, we need to figure out what the drone should do as soon as the Roomba is found. One of the solutions could be to follow the Roomba whilst staying at a minimum height.

It has been declared that this is one of the most significant issues we will have to resolve. To help us conclude, as well as to give us ideas, we will be reading through a variety of articles regarding this topic.

# Use Cases

|  |  |
| --- | --- |
| *Use Case Name* | *Location Problem* |
| *Participating Objects* | Drone, Pen, Roomba |
| *Description* | We want the drone to be able to guide the Roomba to the location of the pen |
| *Flow of Events* | 1. Drone enters the zone 2. Drone to know the location of the pen 3. Drone to find the Roomba using image analysis 4. Drone to manoeuvre the Roomba to the pen |
| *Pre-condition* | * Roomba to have a QR code * Roomba is only moving within the zone |
| *Post-condition* | * Roomba must at least reach or go through the pen for it to be successful |

|  |  |
| --- | --- |
| *Use Case Name* | *Drone-Pen Acknowledgement* |
| *Participating Objects* | Drone, Pen |
| *Description* | Drone to know the location of the pen before attempting to find the location of the Roomba |

|  |  |
| --- | --- |
| *Use Case Name* | *Access to Roomba* |
| *Participating Objects* | Roomba, Drone |
| *Description* | Drone can send messages to the Roomba, however, must be simple and cannot be complex directions. Messages can be:   * Rotate 90 degrees * Stop |

|  |  |
| --- | --- |
| *Use Case Name* | *Time to Accomplish Task* |
| *Participating Objects* | Drone, Roomba |
| *Description* | There is no set time for the task to be finished. |

|  |  |
| --- | --- |
| *Use Case Name* | *Attachments* |
| *Participating Objects* | Drone, Pen, Roomba |
| *Description* | All participating objects allowed to have as many attachments needed to accomplish the task |

|  |  |
| --- | --- |
| *Use Case Name* | *Low Battery Roomba* |
| *Participating Objects* | Roomba |
| *Flow of Events* | 1. Roomba will realise it is low battery 2. As an automatic reaction, Roomba will find the nearest charging source |
| *Pre-condition* | 1. Charging station is nearby 2. Roomba needs to be low in battery |

|  |  |
| --- | --- |
| *Use Case Name* | *Speed of the Roomba* |
| *Participating Objects* | Roomba |
| *Description* | Roomba will have a constant speed throughout the task to enable accurate image analysation |

|  |  |
| --- | --- |
| *Use Case Name* | *Drone Specifics* |
| *Participating Objects* | Drone |
| *Description* | Drone will have the following:   * 2 cameras: one vertical camera, and one horizontal camera * Gyroscope * Accelerometer * Magnetometer * Pressure sensor * Altitude ultrasound sensor * Wi-Fi |

|  |  |
| --- | --- |
| *Use Case Name* | *Roomba Messages* |
| *Participating Objects* | Drone, Roomba |
| *Description* | Drone will have the following:   * 2 cameras: one vertical camera, and one horizontal camera * Gyroscope * Accelerometer * Magnetometer * Pressure sensor * Altitude ultrasound sensor * Wi-Fi |