Arena Choices

NJRB2 | MVS9 | FA296

This document will explain the choices made when designing the arena for the project. The arena is the region in which the drone and the Roomba are allowed to operate – crossing the arena boundaries results in a failed run. It contains the two robots, the marker for the Roomba, the marker for the Pen, and is outlined in tape.

# Markers

There are two markers we use, one for the Pen and one for the Roomba. The designs for these markers changed over the course of the project, and were only finalised in week 10. The process of iteration is detailed below.

In our first meeting with our supervisor, it was recommended that we use QR codes to detect the Roomba, in order to simplify that aspect of the project. We also decided that we should use a QR code to detect the location of the Pen, to reduce the amount of code needed. The advantages of using QR codes were that they did not rely on colour, and therefore were less susceptible to changes in lighting, and they were commonly used, meaning we were able to find a good amount of information on creating a QR code detector online.

|  |  |
| --- | --- |
|  |  |

1. Left: The QR code for the Roomba. Right: The QR code for the Pen.

During evaluation, we found that the QR codes were being detected very unreliably. The reason for this was that the drone’s camera wasn’t high-resolution enough to detect the very smallest squares on the code, particularly when outside in bright sunlight which caused the white areas to glare and cover the black areas. We decided that it was unnecessary to use QR codes because we only had two patterns we needed to detect, and that we could design alternative patterns which would work similarly to the QR codes (reducing the amount of code which needed changing), but which were more visible and useful for our needs.

|  |  |
| --- | --- |
|  |  |

2. Left: Our custom pattern for the Roomba. Right: Our custom pattern for the Pen.

These designs were much better – by only using large boxes, the detection reliability increased to a much more usable amount. They were unique enough that it was easy to differentiate between them, and they were both asymmetrical on one axis, meaning that the top of each pattern could be detected at any rotation, allowing us to see the orientation in degrees relative to the drone camera. There was, however, still one problem. Despite the fact that the detection reliability was much improved in the simulator, the sunlight would still cause glare when outside. We came up with a very simple solution of inverting the colours to reduce the amount of white space on the pattern, and making the inner box (which was now white) smaller.

|  |  |
| --- | --- |
|  |  |

3: Left: The final Roomba pattern. Right: The final Pen pattern.

The detection rates both in the simulator and in real life were now good enough to test other aspects of the project. Some issues remained when detecting patterns in real life – more information on this can be found in the Noise Reduction document.

# Boundaries

The design for the arena boundaries also changed over time, although not in the more refined style of the markers. The initial design for the arena boundaries was a hazard tape style border, with contrasting colours and a distinct pattern.

|  |
| --- |
| 1: The initial hazard tape design for the arena boundaries. |

There were three reasons behind this design being chosen:

1. It had a distinct, repeating pattern. That style of stripes is unlikely to occur in nature, so the image analysis would not get false positives in real life.
2. It had a contrast between light and dark. The image analysis for this pattern would likely have been done on a greyscale image, but the contrast between the dark black and the bright yellow would have remained and allowed for the pattern to be more quickly identified in the frame, reducing processing time.
3. We had access to hazard tape in real life. Unlike the patterns, which could be printed out onto paper once, the boundaries would have to be many metres long. Long rolls of tape would allow for these long boundaries to be set down in real life quite easily, and because hazard tape is already a real product, getting a roll of it would have been very easy.

However, the image analysis code for this design was never implemented beyond a few fairly unsuccessful tests. The reason for this is that we switched our focus to the simulator primarily. In the simulator, lighting is controlled – objects are either lit or shadowed. In order to simplify the code for the simulator, we switched to a much simpler, but less robust design.

|  |
| --- |
| 2: The final design for the arena boundaries. |

This design, a simple bright pink strip, is unlikely to work very well in real life. It is also not so easy to find as a long strip of tape. However, the code to detect it in the simulator is very quick and simple, so it works well for that purpose. It is possible that this design will be changed again once real life becomes more of a focus again.

# Layout

We chose to place some restrictions on the arena in order to simplify the possible variations we would be dealing with. The goal of this was to reduce development time to a more manageable level (given that we would also be working on other university coursework), and allow us to produce more refined solutions for the project within our restrictions.

The first restriction was that the Pen should always be in the bottom-left corner of the arena. This meant that coordinate-based search and return solutions would be easier to implement, as the bottom-left starting point could be set as coordinate 0,0.

The second restriction was that the drone always starts on the Pen. This gives us a known return location and means we only need to locate the Roomba. This is important because the Roomba has a chance of leaving the arena before we’ve found it, so finding it quickly is very crucial. However, if we found the Roomba before finding the Pen we need to guide it towards, we would have to implement some control behaviour, where the drone would go off and search whilst also not allowing the Roomba to drift away too much that we lost its location again. Due to the development time constraints (and in a certain size of arena, battery life limitations) we decided that doing a more simple search and return would be a more reasonable solution.

The third restriction was that the arena must be a rectangular or square shape. This restriction meant our search algorithms did not have to be as robust as they would be if they had to deal with circular or triangular search areas, reducing development time. It also allowed us to use 2D arrays to represent the arena locations, instead of having to come up with a more novel representation.