So, for this project, we needed a method of differentiating particles in different areas of the map. We achieved this using masks. Lots and lots of masks. 32 masks in total, to be exact.

To get these masks, we simply took a map of QUT. Using the image thresholder tool in Matlab, we then lifted a mask of the paths and the buildings. The buildings mask took two different masks, as D block is a different colour from the rest. After that, we used a median filter to clean up the noise.

We needed to develop different masks for different paths, and buildings. Matlab was a little primitive for this kind of task, so we just used paint to mask out areas we didn’t want to see.

The maze random walk already includes code that tells you if you hit a barrier. However, the code itself is really inefficient, and we found that it took ages for Matlab to compute. So, we adopted a new strategy.

The particles have a certain (x, y) position on the Cartesian plain. We convert these co-ordinates to array positions, rows and columns. We then compare this position to the mask array. If it’s a zero, or black, nothing happens. But if it’s a one, or white, it registers as a hit.

This new strategy is much more efficient that the last, but it has some limitations. One: the particles can’t go outside of the mask, otherwise the function breaks. To solve this, we added another function, to check if the particles is outside of the array. Two: the particles co-ordinates must be integers. This means that the starting points, and delta x and y must also be integers.