# Kate Meeting Preparation

# Algorithmic Study completed

A\* Algorithm + Heuristics (Euclidean, Manhattan….)

Dijkstra’s Algorithm

(I will go on further and document about more for project purposes)

# The next steps I need to uncover:

* How to prepare/access data for local area (detailed roads – within specific region: e.g New Cross)
* Need road names, length of road, road type relation among other roads
* What’s the best way to project this onto a node-edge like graph for use with A\*
* If I want to go from one road to another, how does that become a node to node traversal?
* Furthermore, if I wanted to go to a specific co-ordinate within that region, would that work?
* Can I get funding for a pollution sensor (Particulate Matter PM 2.5 + NO2)?
* This would be required for the evaluating part of my project

# What I need to know:

How to implement APIs within program

* I have found Google’s traffic layer useful
* I have contacted LondonAir run by Kings College who have data from sensors and also somehow are making measurements of pollution on particular roads.

Is integrating multiple API’s relating to various aspects a viable solution

* For example, taking the level of pollution, traffic and combing these to give a score of a road

What language should I use, JAVA is my strongest; will this be the best approach?

# What I’ve established

Variables that’ll contribute to dispersion of pollution

* Weather, Buildings, Plantation, Flow of traffic, Size of road (Not to be taken into account until further implementation)
* Importance of hierarchy of roads (The application is for Cyclists and Pedestrians, therefore a hierarchy of roads needs to be had. For example A roads would be at the bottom and residential roads would be near the top. (Late stage implementation)

I would like to discuss the viability of normalizing length and pollution level between 0 & 1 for a road and then add them to find the lowest score out of possible roads to choose the best one.

# Planned execution Procedure

1. Get a pathfinding algorithm to work
2. Get small region of London projected in a graph
3. Assign roads as node values – begin with zero weighting
4. Be able to traverse from one end of the graph to another with the use of pathfinding algorithm
5. Take live API data which will allow me to use traffic or pollution sensor data to apply weights to roads. (Start with static then work on real time)
6. Test method of traversal of the algorithm and fine tune with added weights
7. Add variables to improve accuracy
8. Test with real life. Does the pollutions on roads correlated with the calculated comparison on the API

# Outcomes

My first meeting with Kate was short and concise.

The meeting lasted 11minutes and was over Skype. Kate and I addressed the immediate things to tackle and didn’t delve too deep into other areas. With a bad Skype line, it was not the right opportunity or environment for me to lay everything out.

Kate recognised that the challenge would lie with integrating the API into the program and seeing what the data from the API looked like. She gave me the task of looking into the APIs that I had planned on integrating into my application. This included researching about:

* Google’s traffic layer API
* Pollution sensor monitors in London
* Any other sources

Kate further told me to try and get the APIs integrated and to have them running. This was my task over the weekend after meeting on the afternoon of January Friday 20th 2017.