NAVIGATION APP FOR HAMPTON SCHOOL

Robert Ng

U6CHR

Glossary

Inception

Analysis

Design

Solution

Testing

Evaluation

**Inception**

Aim: To produce a program or application that will choose the shortest path to get from point A to point B (or in this case room A to room B) in Hampton School.

The idea for this project was initially brought up by a teacher in class (**props to Mr Ting**) which after hearing I added to my options for possible NEAs. After more considerate thought, I decided that the most challenging as well as interesting problem to solve would be to use an algorithm to navigate our school taking the shortest path like google maps for a school (although hopefully taking the best path rather than the next to best from a heuristic algorithm). This hopefully will help my client (a new teacher at Hampton) as well as other new joiners to the school in navigating our vast complicated school. As such the aim is to help as many new Hampton joiners as possible with this navigation system.

Client: Mr Thompson

**Analysis**

To clarify:

* The problem being solved is the struggle for new joiners to Hampton school in navigating to their next lessons given how big and complicated it can be
* The solution will require finding a correct path to get to the destination from a current point as well as the fastest path

As previously stated, this program will be like Google maps in that it is a navigation system that will give you a path to get from point A to point B. However differently to Google maps’ heuristic algorithm, I hope to use Dijkstra’s algorithm to return the shortest path given that I will be dealing with fewer variables, so a heuristic algorithm is unnecessary.

Regarding existing applications that solve a similar problem to mine is a website called Mapped.In. This website develops a map of your building of choice that can be navigated using its in-built path finding algorithm. The map is loaded by layering a blueprint of the building onto a geographical top-down view of the building (seems to be effectively placing the picture of a blueprint onto the building as seen on google maps). From there, you set out where doors and rooms are ===Check what comes after===. A difference and a flaw in my opinion of the website is the actual setup of the map. Their setup requires a pdf of picture form of the building’s blueprint that you may not possess and then you must place it onto the “google map” view which I found doesn’t align very well (considerably as the proportions of the building on the map are not perfect). Hence this setup was quite problematic. The aim of the website was most likely to simplify the process of mapping a building by simply putting a blueprint in, however I think this has in fact made the process more complicated and finicky. Such for my project should I develop an interface to make your own map, it will be a DIY map builder to create from scratch in contrast to the website’s method. Although simpler, this should prove to be a more functional way.

I will be taking a hybrid approach as I will be decomposing this problem into two different parts:

* The shortest path algorithm (including map coding)
* Interactive User Interface

I will divide these problems into base **low-level objectives** of the following:

1. Upon loading application, user presented initial menu with drop down box of start point for end point and a navigate button that should only work after both start and end points are inputted correctly
2. After pressing navigate, screen should swap from initial menu to a navigation map
3. Interface correctly gives the verbal (and graphical) description of the shortest path from A to B, with descriptions passed incrementally one by one
4. A “next” and “previous” button should be usable to move between step by step instructions
5. “Exit” button usable to exit navigation at any point
6. Map of building held as a graph

Should I have more time expand to **high-level objectives**.

1. Upon booting application, menu appears with
2. An interface to allow you to design your own map

Such this enables more flexibility if the existing building has changes made to it whether that’s simple room name changes or fundamental structural changes. Or even for the usage of the program to completely different buildings.

Other extended objectives include the following:

1. The ability to input your timetable to auto-navigate to your next lesson based on the time
2. Modelling traffic which will affect the weighting of certain paths such as
   * + Dependency on time of day
     + Dependency on the frequency a path is used (i.e. more frequently used paths will have larger weighting)

**Design**

I have as stated above decomposed the whole problem into the two sections of the path finding algorithm and the user interface. I will further decompose the path finding algorithm again into the algorithm itself and the graph of the map that it acts on.

**User Interface**

The user interface would require the following (already briefly stated above but to reiterate):

* A menu with 3 options

1. Start button to start navigation
2. Drop down boxes to input start point (point A)
3. Drop down boxes to input end point (point B)

* A screen providing navigation of the algorithm

1. Graphical visualisation

**Path Finding Algorithm: Coding The Map (Graphing)**

A blueprint of a house

Description automatically generatedAt first, for simplicity I will hardcode my map into the program (and should it happen later the interface to draw your own map). To help visualise this problem I took the existing map of Hampton and using draw.io I copied it onto this. A section of this is shown below.

As can be seen, I initially placed nodes at each point where the corridor spits to another path. Prior to this I considered making every room a node and navigating to them (like the simple model of the A-to-Z graph), however I quickly realized that this was much too complex and inefficient considering the number of nodes this would result in the path finding algorithm looking through. Hence why I initially place nodes at each corridor intersection. This however was still much too inefficient as there was still the problem of how going from a node to the rooms themselves within each corridor.

A computer screen shot of a blueprint

Description automatically generatedAfter much thinking, I came to the solution of now setting the nodes to the corridors themselves. The same section with corridors can be seen below

I found this to be a much more efficient and elegant solution than the formers. This way I would associate all rooms to their relevant corridors, and so to navigate to a specific room first search for which corridor this would be and from there navigate to said corridor. Although perhaps not the best solution, for my purposes it seemed the easiest to visualise and graph out. Although this does require searching for which corridor a room is within, this can be optimised later and this solution also makes the graph much simpler to create, edit and correct if needed. Another potential problem would be that this is not truly navigating to the room itself however I believe that provided the location of the corridor where the rooms is in, it wouldn’t be a huge inconvenience to then just find the room withing said corridor.