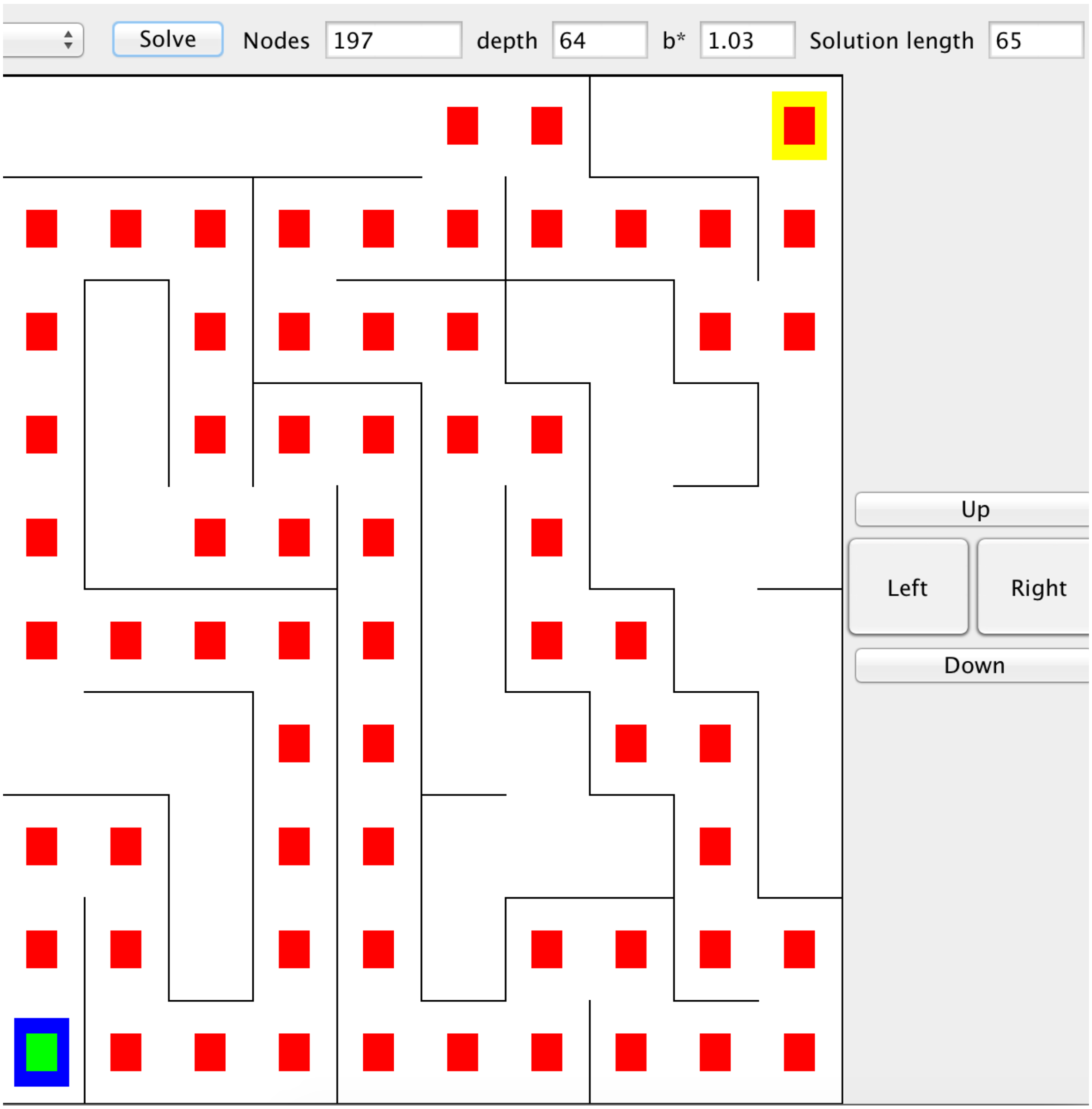
Karl Franks
kpf@aber.ac.uk

CS26110 Assignment Report

Heuristics

The Manhattan Distance of a node in this maze is definitely an admissible heuristic because it essentially gets the distance the green square has to travel to the goal if there was no walls. However, as there is the obstacle of walls, this means a more direct path cannot be taken and the true distance will always be larger than the Manhattan Distance.
For example, in this maze the Manhattan Distance from the start node is 20, but the final result avoiding the obstacles is over three times that:



Karl Franks
kpf@aber.ac.uk

The additional admissible heuristic I have implemented returns the straight line distance from the current node to the goal. This is rounded to a whole number, due to the constraint of getDistance needing to return an integer value. This is admissible, since nodes are constrained to only move in four directions and again due to the obstacle of walls.

The Manhattan Distance is arguably more informed than the straight line distance, since movement is limited to four directions. The Manhattan Distance takes this into account, and essentially returns a best-case scenario. This is not to say the straight line distance is not useful though - being admissible it will not overestimate.

The non-admissible heuristic I have implemented returns the Manhattan Distance plus the percentage of treasures found so far multipled by ten (like the admissible heuristic, it is rounded to a whole number due to returning an integer). I devised this heuristic from the limited amount of information I can call on. I decided to multiply the percentage by five, so the percentage has more of an effect on the heuristic value (since early on, before it finds any treasures it would just be the Manhattan Distance, which *is* admissible). This heuristic is non-admissible because, for example - assume I have two treasures to find, one has already been found and the Manhattan Distance = 10. The heuristic will return a value of 260, obviously a lot further than is needed to travel.

I have since realised after implementing this heuristic, that it is only non-admissible in situations where there treasures on the board and you have already found at least one treasure. Until then, it is just the Manhattan Distance (which is admissible).
I have also since realised that, since lower values are seen as the “best” in best-first search, my non-admissible heuristic does the inverse of what it should: as more treasures are found the heuristic value gets higher.
I have decided not to change this, due to time limitations, and I think it would also be interesting to see how a “bad” heuristic compares to the others.

Karl Franks
kpf@aber.ac.uk

Data Analysis

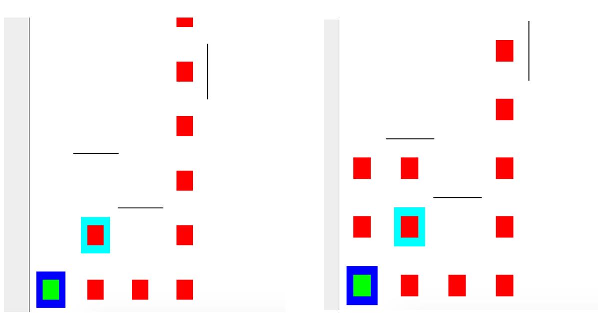
For experimentation I decided to vary the size of the grid by carrying out test in three sizes: 10 by 10, 50 by 50, and 100 by 100. I also varied the number of treasures by testing with 0, 3 and 10 treasures; as well as varying the perfection slider: set to the top, halfway down, and bottom. I believe this range of values tested represents a wide variation for suitable analysis, however I started to face some problems with the larger grids in later tests, so perhaps some spikework should have shown this program is not very efficient at large grids and to stick to smaller grids, eg up to 50 by 50.

As can be easily gleaned from the results, there can be a wide variation in the effects these heuristics have on the amount of nodes found and the resulting length of solution.

In general the Breadth-First Search (BFS) approach always produces a shortest possible route but at the cost of searching a lot more nodes. This approach does not seem worth it when handling larger amounts of data, for example: when searching a 50 by 50 grid with no treasures and the lowest perfection, both the Manhattan Distance and BFS approaches returned a solution of length 99 although BFS searched 9206 nodes before finding a solution compared with Manhattan's 1767 - over five times more. (Interestingly, they found different paths, each with the same length).
There are also examples of the Manhattan Distance finding a solution that is not the shortest, but is close enough and searches a lot less nodes than BFS. Eg, searching a 10 by 10 grid with no treasures and maximum perfection, BFS returns the shorter distance of 81 compared to Manhattan's 83, but BFS searched twice as many nodes.

My own admissible heuristic (straight line distance) performed on the whole very similar to Manhattan Distance, but generally took a small percentage more nodes to find either a very similar or slightly longer solution. If I was implenting Best-First Search myself, based on this (albeit quite small set of data) I would most likely choose Manhattan Distance over my heuristic, but (much like Manhattan) straight line distance is still a lot more efficient than Breadth First Search.

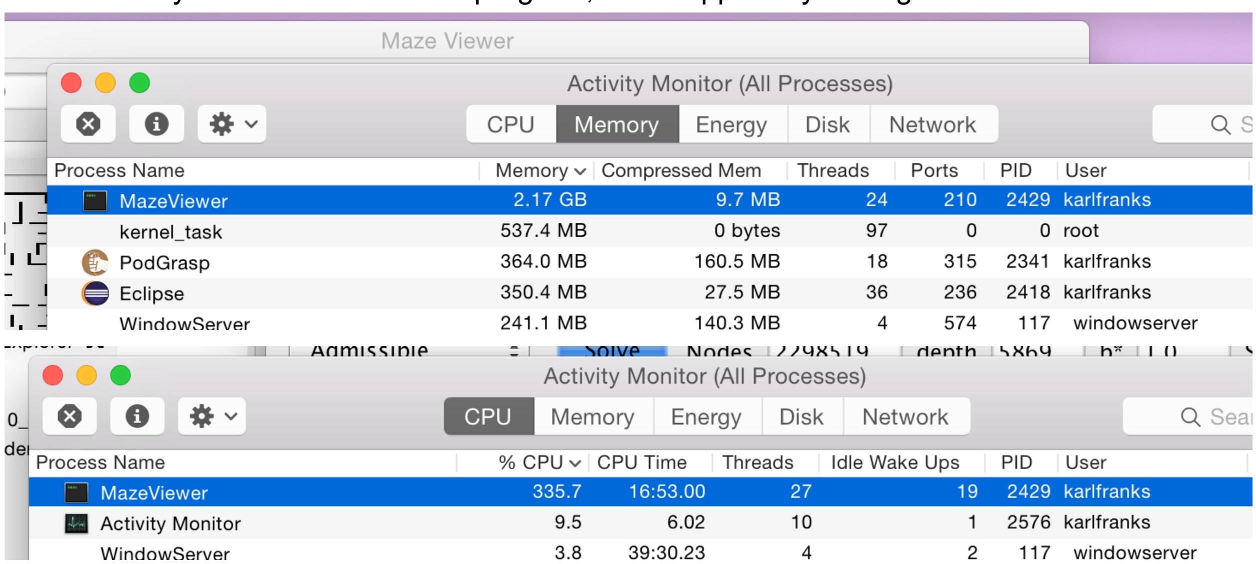
As predicted, my non-admissible heuristic was terrible. In most tests it actually searched more nodes than Breadth First Search and returned a longer solution.
An interesting effect I noticed sometimes with it, was its paths would generally follow the same path as Manhattan Distance (predictable, from the way it was derived) until the very end and it would "loop around" and take a less direct path to the goal.
For example:
(Manhattan on left, Non-Admissible on right)



Karl Franks
kpf@aber.ac.uk

The heuristics generally performed as I had expected, eg with Breadth-First searching a lot more nodes than something like the Manhattan Distance. I predicted my non-admissible heuristic would be bad, but maybe not as bad as it was with it performing worse than Breadth-First search in most tests.

As to varying the maze itself, at the smaller size differences between heuristics are less pronounced. At larger sizes (and at lower maze perfection), the program starts having to search a *lot* of nodes with some results are in the millions. At 50 by 50, with half maze perfection and 10 treasures, I was searching at least 5 million nodes and the program would free for about thirty second to a minute. I decided against even attempting the lowest perfection. I had similar issues with searching 100 by 100, with full maze perfection and 10 treasures. I attempted run the Admissible heuristic on this at half perfection, but the program froze, and upon inspection found it was using over 2 Gb of memory and claiming to use over 330% CPU cycles. I terminated the program, and stopped my testing there.



The obvious reason for this, is that that the complexity increases a lot with simply a bigger grid and more treasures to find, and also as you decrease the “perfection” meaning there are a huge amount more possible paths through the maze.

I believe my testing shows clearly how informed search, using a good heuristic, is the preferred solution over uninformed strategies such as breadth-first search.
Breadth-first can perhaps be preferred when you have a lot of computing power and time is not a limiting factor and you absolutely need the shortest, best result. However, when resources are limited and time is of the essence, having a solution which is very close to perfect but searched half the nodes, informed search is clearly the better choice.

Karl Franks
kpf@aber.ac.uk

Experiment Data

10 by 10 Grid

Zero Treasures

Heuristic	Perfection	Nodes Explored	Depth	Length
Admissible	100%	197	62	63
Non-Admissible	100%	197	62	63
Manhattan	100%	197	62	63
Breadth-First	100%	197	62	63
Admissible	50%	154	18	19
Non-Admissible	50%	96	18	19
Manhattan	50%	96	18	19
Breadth-First	50%	294	18	19
Admissible	0%	210	18	19
Non-Admissible	0%	177	18	19
Manhattan	0%	177	18	19
Breadth-First	0%	348	18	19

Karl Franks
kpf@aber.ac.uk

Three Treasures

Heuristic	Perfection	Nodes Explored	Depth	Length
Admissible	100%	752	89	81
Non-Admissible	100%	885	89	81
Manhattan	100%	748	87	83
Breadth-First	100%	821	79	79
Admissible	50%	1187	34	27
Non-Admissible	50%	2086	50	33
Manhattan	50%	1293	40	33
Breadth-First	50%	1435	21	21
Admissible	0%	499	21	21
Non-Admissible	0%	2560	37	25
Manhattan	0%	1032	27	21
Breadth-First	0%	1512	19	19

Karl Franks
kpf@aber.ac.uk

Ten Treasures

Heuristic	Perfection	Nodes Explored	Depth	Length
Admissible	100%	748	86	83
Non-Admissible	100%	1850	147	83
Manhattan	100%	725	84	83
Breadth-First	100%	1315	81	81
Admissible	50%	241293	84	71
Non-Admissible	50%	351726	118	61
Manhattan	50%	214759	91	61
Breadth-First	50%	348772	47	47
Admissible	0%	272566	80	59
Non-Admissible	0%	410346	88	73
Manhattan	0%	236700	82	71
Breadth-First	0%	402963	37	37

Karl Franks
kpf@aber.ac.uk

50 by 50 Grid
Zero Treasures

Heuristic	Perfection	Nodes Explored	Depth	Length
Admissible	100%	1893	516	517
Non-Admissible	100%	2376	516	517
Manhattan	100%	2376	516	517
Breadth-First	100%	4362	517	517
Admissible	50%	1797	98	99
Non-Admissible	50%	1283	98	99
Manhattan	50%	1283	98	99
Breadth-First	50%	7922	98	99
Admissible	0%	2253	98	99
Non-Admissible	0%	1767	98	99
Manhattan	0%	1767	98	99
Breadth-First	0%	9206	98	99

Karl Franks
kpf@aber.ac.uk

Three Treasures

Heuristic	Perfection	Nodes Explored	Depth	Length
Admissible	100%	23353	1142	1142
Non-Admissible	100%	30885	1397	1141
Manhattan	100%	13018	1140	1141
Breadth-First	100%	32275	1140	1141
Admissible	50%	21267	180	181
Non-Admissible	50%	55536	217	131
Manhattan	50%	25961	176	131
Breadth-First	50%	26253	103	103
Admissible	0%	35674	250	251
Non-Admissible	0%	64200	335	249
Manhattan	0%	34659	248	249
Breadth-First	0%	21955	103	103

Ten Treasures

Heuristic	Perfection	Nodes Explored	Depth	Length
Admissible	100%	216414	2178	1733
Non-Admissible	100%	294617	2655	1685
Manhattan	100%	172544	1975	1685
Breadth-First	100%	257385	1685	1685
Admissible	50%	5666358	653	521
Non-Admissible	50%	8170776	661	475
Manhattan	50%	6741044	652	611
Breadth-First	50%	6910696	161	161

Karl Franks
kpf@aber.ac.uk

100 by 100 Grid
Zero Treasures

Heuristic	Perfection	Nodes Explored	Depth	Length
Admissible	100%	16984	1784	1605
Non-Admissible	100%	19034	1784	1605
Manhattan	100%	19034	1784	1605
Breadth-First	100%	17926	1604	1605
Admissible	50%	4347	198	199
Non-Admissible	50%	3096	198	199
Manhattan	50%	3096	198	199
Breadth-First	50%	31990	198	199
Admissible	0%	5684	198	199
Non-Admissible	0%	4294	198	199
Manhattan	0%	4294	198	199
Breadth-First	0%	37146	198	199

Karl Franks
kpf@aber.ac.uk

Three Treasures

Heuristic	Perfection	Nodes Explored	Depth	Length
Admissible	100%	86274	4151	4059
Non-Admissible	100%	87774	4151	4059
Manhattan	100%	86348	4153	4061
Breadth-First	100%	87485	4059	4059
Admissible	50%	147515	530	477
Non-Admissible	50%	221171	591	323
Manhattan	50%	141704	567	467
Breadth-First	50%	171067	241	241
Admissible	0%	103067	344	277
Non-Admissible	0%	260976	630	477
Manhattan	0%	120166	484	477
Breadth-First	0%	93735	211	211

Ten Treasures

Heuristic	Perfection	Nodes Explored	Depth	Length
Admissible	100%	860838	7609	6463
Non-Admissible	100%	2288286	7767	5873
Manhattan	100%	778971	7767	5873
Breadth-First	100%	2298519	5869	5869