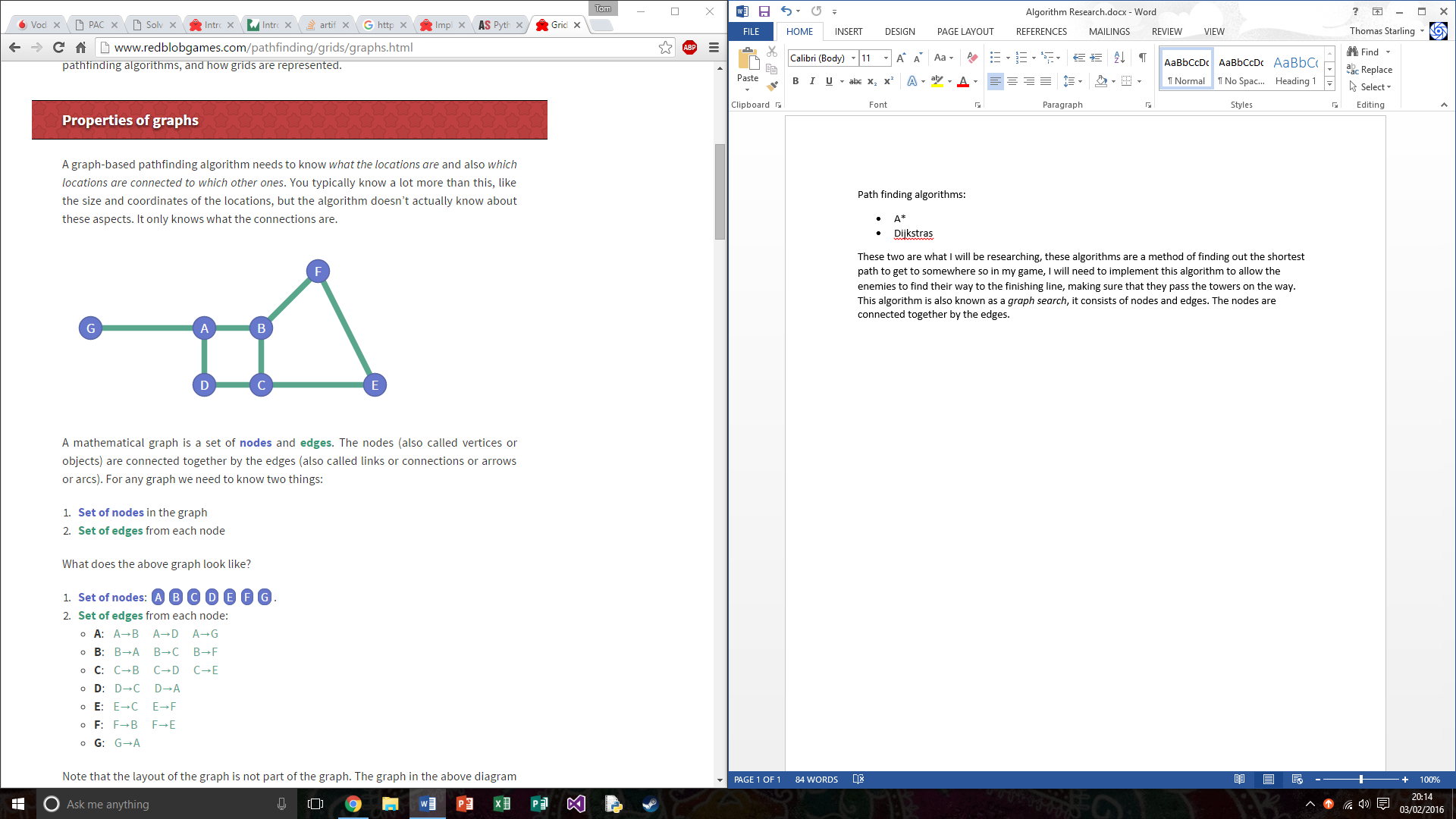
Path finding algorithms:

The A\* algorithm is what I will be researching, the algorithm id a method of finding out the shortest path to get to somewhere so in my game, I will need to implement this algorithm to allow the enemies to find their way to the finishing line, making sure that they pass the towers on the way. 

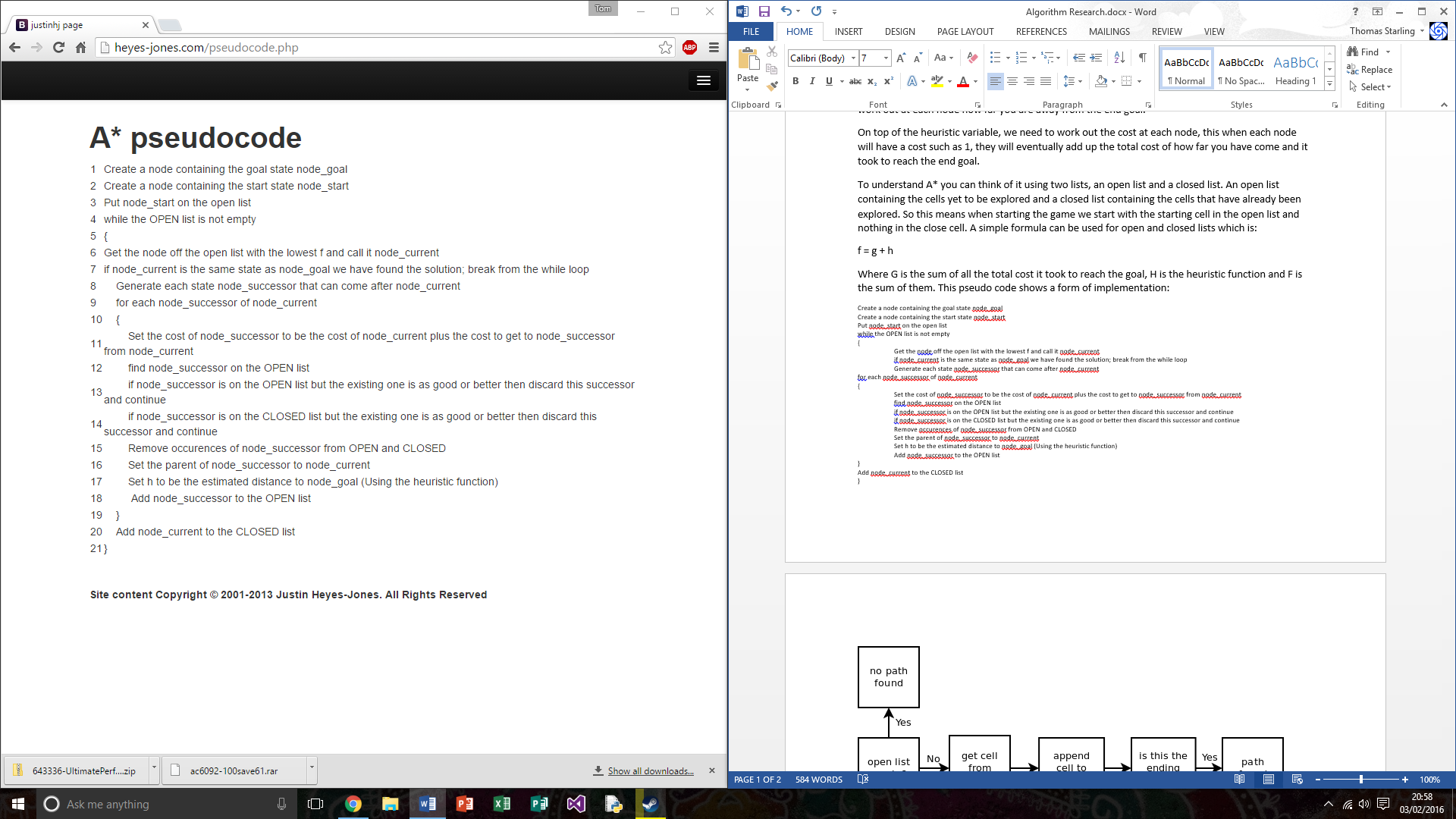
This algorithm is also known as a *graph search*, it consists of nodes and edges. The nodes are connected together by the edges. There are also different types of graphs, the first is an ‘undirected’ graph, this graph is when it has edges that go in both directions such as C->B, B->C. The next type of graph is a directed graph, this is when the edges can go in one direction but not the other. The last variant of graph is multigraph, this is when you have multiple edges between the nodes, for example is we take node ‘B’ and it is connected to C and D this makes the graph multigraph.

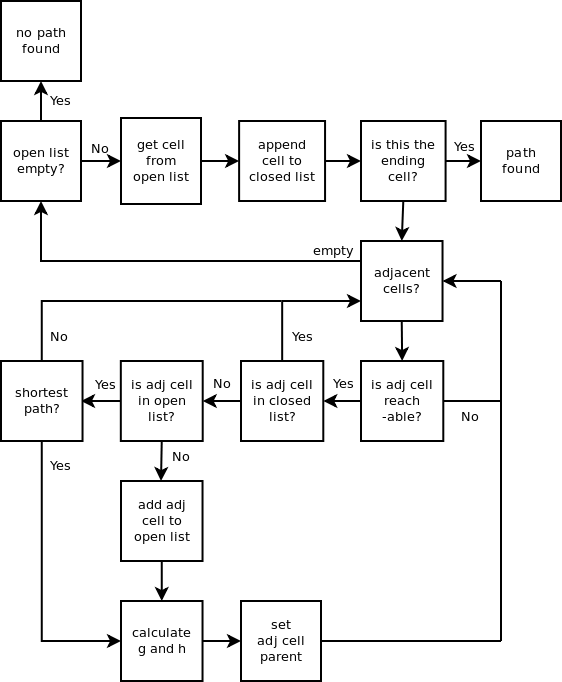
A heuristic is like an algorithm however it isn’t one, the heuristic is a possible solution to solve a problem for which there is no algorithm. This is helpful to cut down on searching, this is used to work out at each node how far you are away from the end goal.

On top of the heuristic variable, we need to work out the cost at each node, this when each node will have a cost such as 1, they will eventually add up the total cost of how far you have come and it took to reach the end goal.

To understand A\* you can think of it using two lists, an open list and a closed list. An open list containing the cells yet to be explored and a closed list containing the cells that have already been explored. So this means when starting the game we start with the starting cell in the open list and nothing in the close cell. A simple formula can be used for open and closed lists which is:

f = g + h

Where G is the sum of all the total cost it took to reach the goal, H is the heuristic function and F is the sum of them. This pseudo code shows a form of implementation: 



<http://www.laurentluce.com/posts/solving-mazes-using-python-simple-recursivity-and-a-search/>

<http://www.redblobgames.com/pathfinding/a-star/introduction.html>

<http://www.redblobgames.com/pathfinding/grids/graphs.html>

<http://heyes-jones.com/astar.php>