**Definition document**

**20.9.2013**

Game AI using the A\* algorithm. Making an implementation of the algorithm in Java to traverse an arbitrary grid (the vertices in a mesh etc).

I've found these resources quite useful when working with this algorithm.

A\*

<http://theory.stanford.edu/~amitp/GameProgramming/>

<http://www.policyalmanac.org/games/aStarTutorial.htm>

PriorityQueue

<http://pages.cs.wisc.edu/~vernon/cs367/notes/11.PRIORITY-Q.html>

ArrayList

**A\***

The algorithm chooses the next nodes based on the total score of the travelled distance and the remaining distance to the goal, calculated with a heuristics method. The score is then sorted so the method gets the lowest possible scores out of the list all the time. This uses for now **Java's own *PriorityQueue*** implementation, I'll change this later on.

The speed and effectiveness of the algorithm is strongly related to the heuristic method chosen. With a better heuristic method one will get the shortest path to the goal, but this will also take up the most resources to calculate.

**PriorityQueue**

Started to implement the priority queue from a heap structure found in the link provided at the top of this document.

**ArrayList**

The ArrayList's implementation builds on an array that doubles in size when the maximum size is reached. (It's very poorly implemented for now, just the methods needed in this project are used.)