The A\* algorithm, stripped of all the code, is fairly simple. There are two sets, OPEN and CLOSED. The OPEN set contains those nodes that are candidates for examining. Initially, the OPEN set contains just one element: the starting position. The CLOSED set contains those nodes that have already been examined. Initially, the CLOSED set is empty. Graphically, the OPEN set is the “frontier” and the CLOSED set is the “interior” of the visited areas. Each node also keeps a pointer to its parent node so that we can determine how it was found.

There is a main loop that repeatedly pulls out the best node n in OPEN (the node with the lowest f value) and examines it. If n is the goal, then we’re done. Otherwise, node n is removed from OPEN and added to CLOSED. Then, its neighbors n' are examined. A neighbor that is in CLOSED has already been seen, so we don’t need to look at it (\*). A neighbor that is in OPEN is scheduled to be looked at, so we don’t need to look at it now (\*). Otherwise, we add it to OPEN, with its parent set to n. The path cost to n', g(n'), will be set to g(n) + movementcost(n, n').

(\*) I’m skipping a small detail here. You do need to check to see if the node’s g value can be lowered, and if so, you re-open it.

OPEN = priority queue containing START

CLOSED = empty set

while lowest rank in OPEN is not the GOAL:

current = remove lowest rank item from OPEN

add current to CLOSED

for neighbors of current:

cost = g(current) + movementcost(current, neighbor)

if neighbor in OPEN and cost less than g(neighbor):

remove neighbor from OPEN, because new path is better

if neighbor in CLOSED and cost less than g(neighbor): \*\*

remove neighbor from CLOSED

if neighbor not in OPEN and neighbor not in CLOSED:

set g(neighbor) to cost

add neighbor to OPEN

set priority queue rank to g(neighbor) + h(neighbor)

set neighbor's parent to current

reconstruct reverse path from goal to start

by following parent pointers

(\*\*) This should never happen if you have an monotone admissible heuristic. However in games we often have inadmissible heuristics.