// A\* (star) Pathfinding

// Initialize both open and closed list

let the openList equal empty list of nodes

let the closedList equal empty list of nodes

// Add the start node

put the startNode on the openList (leave it's f at zero)

// Loop until you find the end

while the openList is not empty

// Get the current node

let the currentNode equal the node with the least f value

remove the currentNode from the openList

add the currentNode to the closedList

// Found the goal

if currentNode is the goal

Congratz! You've found the end! Backtrack to get path

// Generate children

let the children of the currentNode equal the adjacent nodes

for each child in the children

// Child is on the closedList

if child is in the closedList

continue to beginning of for loop

// Create the f, g, and h values

child.g = currentNode.g + distance between child and current

child.h = distance from child to end

child.f = child.g + child.h

// Child is already in openList

if child.position is in the openList's nodes positions

if the child.g is higher than the openList node's g

continue to beginning of for loop

// Add the child to the openList

add the child to the openList