function THINK(player, enemies, maplayout, end)

if openList is not empty and the path is not yet found

current = node in openList with lowest f value

if the player is at the end

create the path, in reverse, using the previous attribute of each tile

pathFound = true;

move the current tile from openList to closedList

get all of the neighbors of current tile

for each neighbor of current

if the neighbor is not in the closedList and it is not a wall

bestG = current g + 1

isBestG = false

if the neighbor is not yet in the openList

neighbor.h = heuristic(neighbor, end)

add to openList

isBestG = true

else if the g of neighbor is worse than the bestG

isBestG = true

if you found the best G

previous of neighbor = current

g of neighbor = bestG

f of neighbor = g of neighbor + h of neighbor

else if the path is found

if the player is on the tile of the most recent path location

move(player, maplayout, end, path)

function initAI(player, enemies, maplayout, end)

openList = []

closedList = []

path is not yet found

generate a replica of the map

add all the neighbors to each tile

add the start position to the openList

function move(player, enemies, maplayout, end)

get the current coordinates of player

get the next position of the path

check if the next position is above, below, to the left, or to the right of the player

move the player

function heuristic(current, end)

return Math.abs(end.x – current.x) + Math.abs(end.y – current.y)