
current_node = heapq.heappop(open_list)
if current_node.position == goal:
path = []
while current_node:
path.append(current_node.position)
current_node = current_node.parent
return path[::-1] # Return reversed path from goal to start
closed_set.add(current_node.position)
# Explore neighbors (up, down, left, right)
for dr, dc in [(-1, 0), (1, 0), (0, -1), (0, 1)]:
new_row = current_node.position[0] + dr
new_col = current_node.position[1] + dc
new_pos = (new_row, new_col)
if (0 <= new_row < rows and 0 <= new_col < cols and
grid[new_row][new_col] == 0 and new_pos not in closed_set):
new_node = Node(
new_pos,
current_node,
current_node.g + 1,
heuristic(new_pos, goal)
)
heapq.heappush(open_list, new_node)
return None # No path found
Activate Windows