

## Navigation Feature documentation

----Jingwei Zhang (z1771485029@gmail.com)

### 1. What is Node? (Object in /constants/nodes.js)



Nodes are points taking up all possible paths. It helps us convert navigation problem to a graph problem. The navigation result contains a list of nodes, representing the optimal steps from the start point to the end point.

Main Nodes: represents the paths on the wood path, which facilities encourage us to walk on

Sub Nodes: represents the paths crossing desks which we may want to avoid if possible

Main nodes on 1st floor: nodes with id < 200, floor = 1, type = "main"

Main nodes on 2nd floor: nodes with id < 200, floor = 2, type = "main"

Main nodes on 3rd floor: nodes with id < 200, floor = 3, type = "main"

Sub nodes on 1st floor: nodes with id: 1\*\*\*, floor = 1, type = "sub"

Sub nodes on 2nd floor: nodes with id 2\*\*\*, floor = 2, type = "sub"

Sub nodes on 3rd floor: nodes with id 3\*\*\*, floor = 3, type = "sub"

#### -How do we get node in code?

Map.js -> getNodeFromId(id, floor)

#### -What is node.nexts?

Node.nexts are a list of nodes which are reachable from the current node. It's defined manually and can be changed in nodes.js. (We cannot using a search algorithm to define nexts because it is hard to detect walls and it may not be reliable. So we decided to define them manually)

### 2. Which algorithm do we use to calculate the best path?

A\* search Algorithm ([https://en.wikipedia.org/wiki/A\\*\\_search\\_algorithm](https://en.wikipedia.org/wiki/A*_search_algorithm))

Basically, it maintains a PriorityQueue called frontier. Everytime it pops a path with lowest heuristic and check whether the end node can be reached from it. If it does, then return the path and it's guaranteed to be the optimal one. If it cannot reach the end node, for every node in nexts, create one deeper path and add them back to the frontier. Continue the loop until we reach the end node.

### **3.How do we handle cross-floor navigation?**

We introduce "transition node" to help us. A transition node could be an elevator or a stair. In case of cross-floor navigation, the navigation result will be two lists of nodes, one for each floor.

getOptimalTransitions(start, end) in Map.js will return the proper transition node given the start and end points.

### **4. What is Canvas?**

Canvas is a translucent layer in which we can draw lines and render nodes. It is useful because we can resize all the elements on that by resizing the canvas itself. (If we don't use that to draw lines, it would be very painful when resizing. )

### **5. What's the execution sequence of navigation?**

navigate() : the startpoint of navigation, it's checking valid start/end point before calculation

→ computeNavigationPath(): Use different logic for same-floor nav and cross-floor nav

→ calulateBestPathOnSameFloor(): In this function, we perform the A\* search algorithm to find the path.

### **FAQ:**

#### **Why did you use canvas for the map pin and not just an image?**

It's okay to use an Image instead, it's just personal preference. If you switch to use image, remember to adjust the position when map is resized.

#### **Why do you have to use priority queue, can't you just use a vanilla JS data structure and find the smallest value to remove?**

PriorityQueue is best in our case, it takes  $O(\lg n)$  to get the lowest heuristic value path at each time.

#### **Why did you divide desk["x"] and desk["y"] by 10?**

The desk points are extracted from map of size 13280 \* 7380.

#### **What is MAP\_X and MAP\_Y:**

MAP\_X and MAP\_Y is the size of map from which nodes are extracted. It helps us converting node.x and node.y to percentage. It could be removed if we have a function in Utils to do the transformation or we can convert all nodes in nodes.js using percentage rather than pixels.

**HOW TO EASILY CONVERT COORDINATES in XXX.jsons TO PERCENTAGE:**

nodes.json: x/1328, y/738

Stairs-elevators-nodes.json: x/1328, y/738

desks.json: x/13280, y/7380

Rooms.json: x/1328, y/738

**-If you have any other questions, just contact me using the provided email address.**



