**Find Patch Algorithm**

This is a JavaScript library to find the shortest Path from the start point to multiple target point.

Requirement conditions

* startPoint

This is the start position to find the shortest path.

Type : Array

Example : [12, 33]

* multiTargetCount

Number of targets you can find at once.

Type: Integer

Example : multiTargetCount = 3

* areaSize

This is the size of the entire grid (store)

Type: Area

Example: areaSize = [153, 50] : This is width count and height count

* tileSize

This is the size(width, height) of one square : Pixel

Type: Area

Example: tileSize = [30, 10]

(width = 30px, height = 10px)

* wall

This contains all shelve’s info:

Type: Object

Example:

Wall = {

Lager: {

pos: {

[1,3], [4,5], [21, 44]

},

col: {

‘#ff0000’

},

end: {  
 [1,4], [4, 66], [43, 11]

}

},

Tool: {

pos: {

[1,3], [4,5], [21, 44]

},

col: {

‘#ff0000’

},

end: {  
 [1,4], [4, 66], [43, 11]

}

},

}

There, Lager and Tool is the shelve ’s name.

Pos: This means all points(or squares) that consist up the shelve.

Col: This is the color to draw each shelve.

End: This mean all end points that each shelve has.

**How to use this library**

Create canvas tag in the html file

Its id must be “gridCanvas”

<canvas id="gridCanvas"></canvas>

Import JQuery library.

Import the findPatch library

Create the config variable and call setConfig function to set the config.

const config = {

startPoint: [55 ,33],

multiTargetCount: 3,

areaSize: [154, 35],

tileSize: {

titleWidth: 8,

tileHeight: 15

},

wall

}

setConfig(config);

Here, you can import the wall as the external js file. Or you can set that variable directly

Call the startFindPath function with the search value to find the shortest path

Ex). startFindPath(searchValue)

SearchValue means target’s name.