

@Navigation Mesh

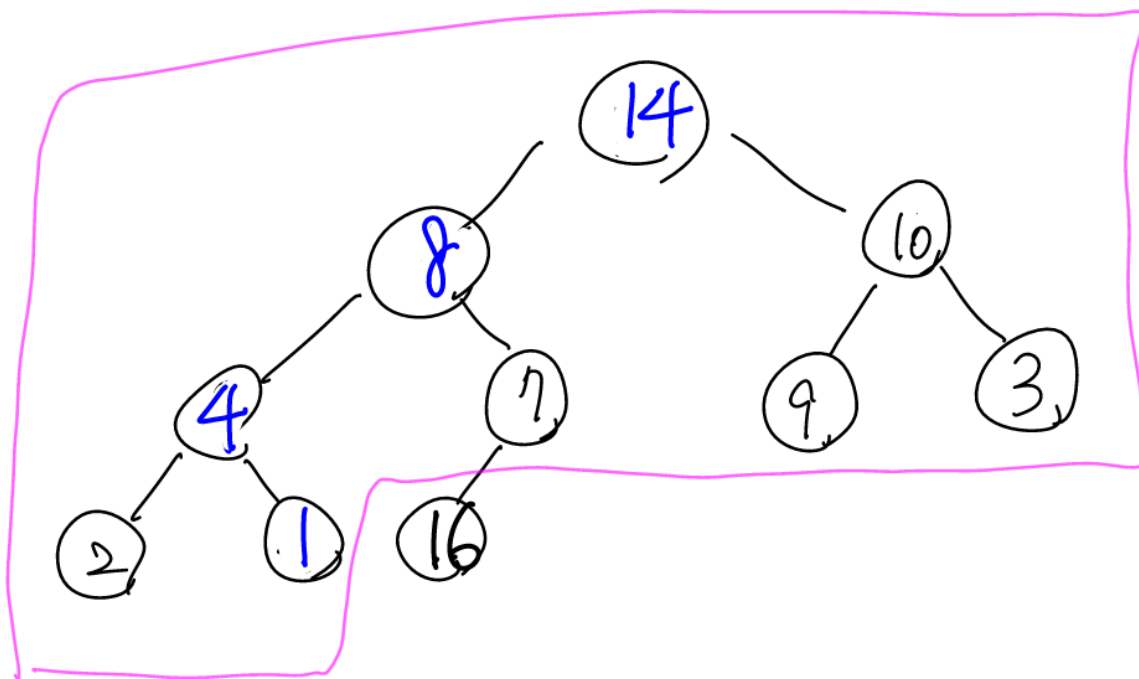
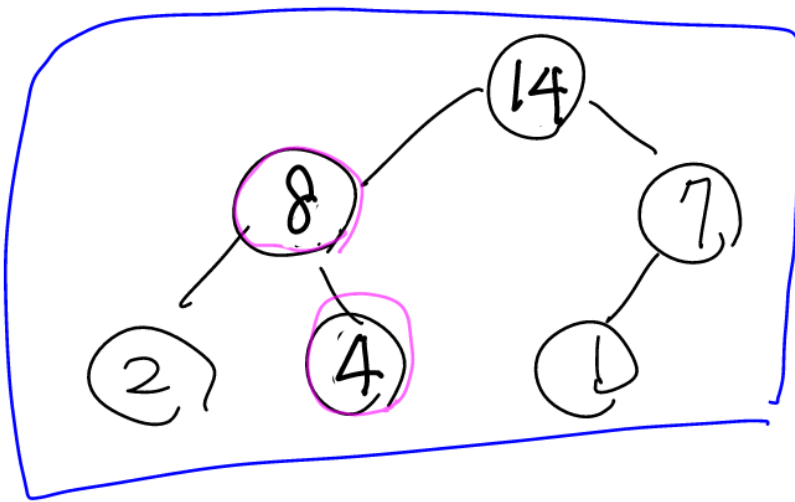
heap (data structure)

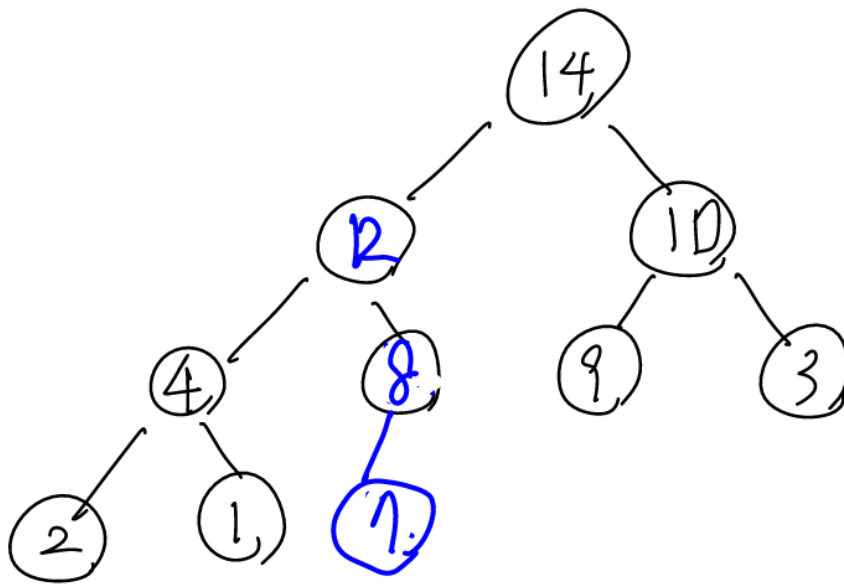
priority queue

hash container (unordered_set)

A* algorithm

Navigation Mesh in Unity

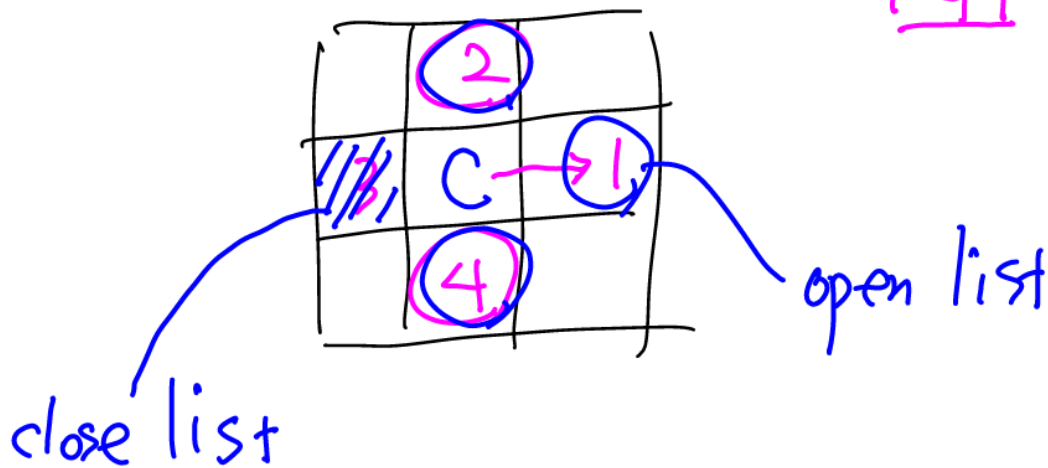




`std::priority_queue < >`
 ↳ uses heap

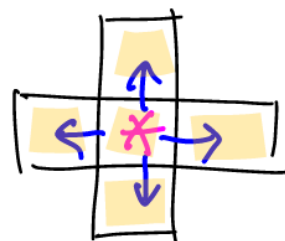
`std::priority_queue < AStarNode >`

[G]



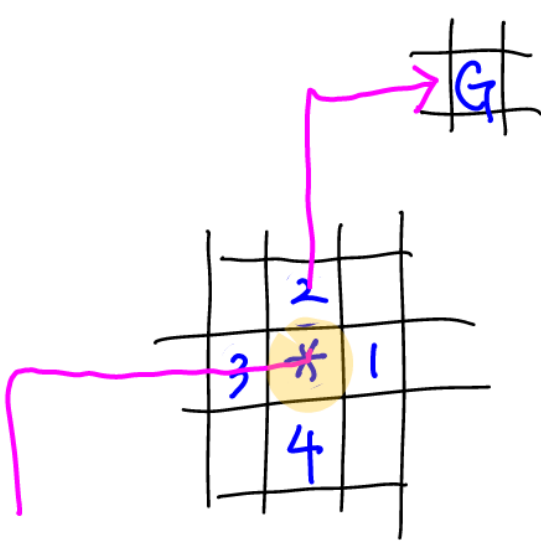
`std::unordered_set < AStarNode >`

@ AStar algorithm
 (basic)

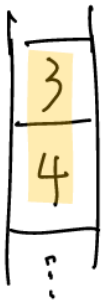


4-connected
 neighbors

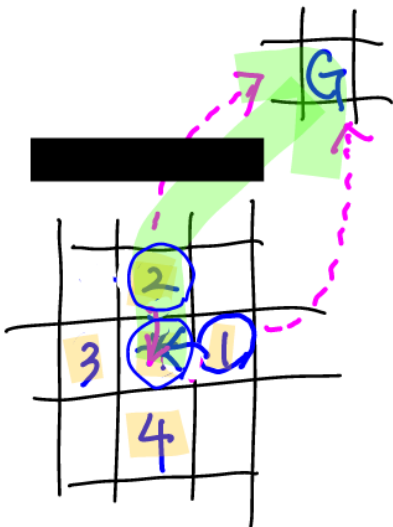
(3)



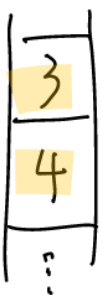
close list



open list (priority queue)



close list



open list (priority queue)



new cost
new heuristic

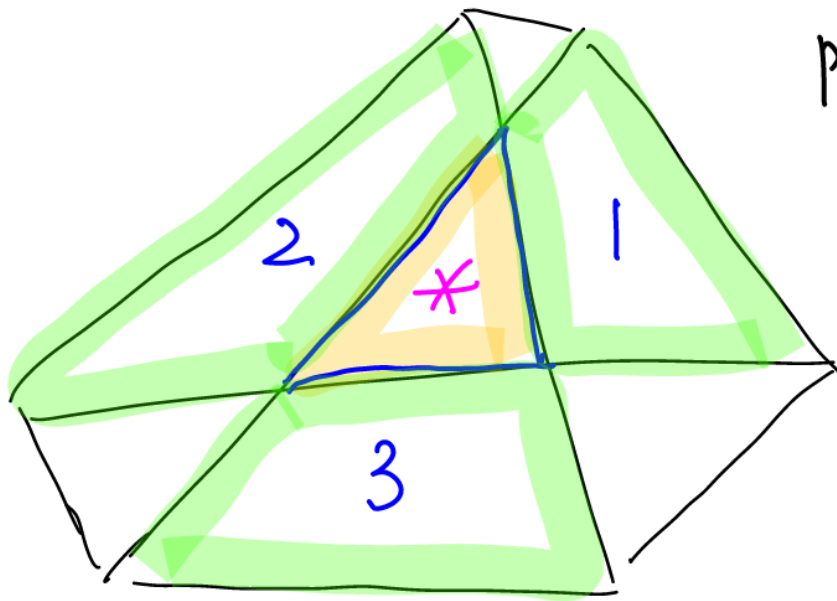
(2018.5.4)

@ Extending neighbors

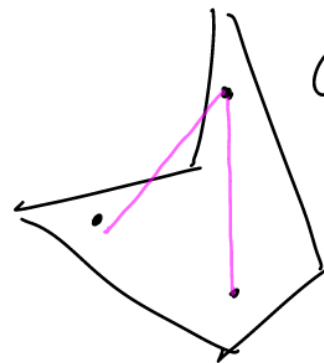
(4)

	2	
3	*	1
	4	

4	3	2
5	*	1
6	7	8



* convex polygon



concave.

@ A* algorithm (detail)

(5)

						G
	S					

Close
(null)

Open
S

						G
	2					
3	S	1				
	4					

close
S

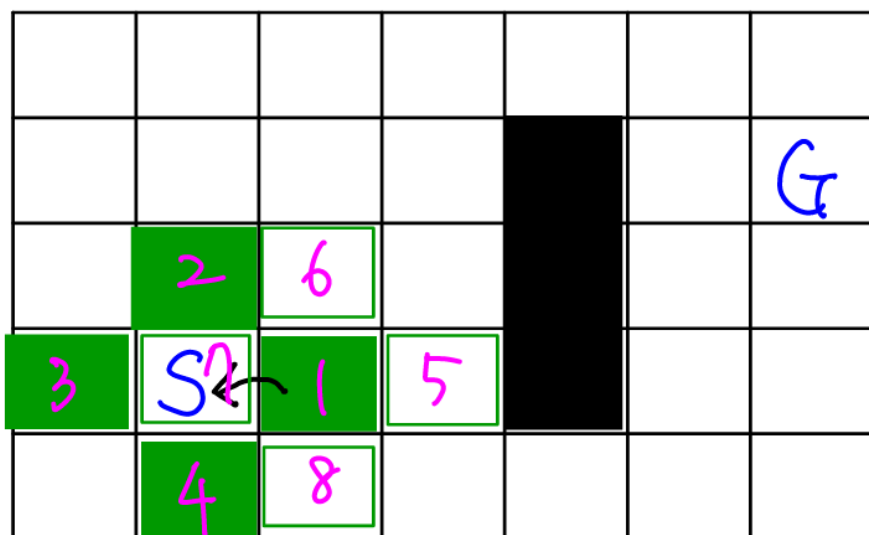
Open
1
2
3
4

						G
	2					
3	S ← 1					
	4					

close
S
1

Open
~~1~~
2
3
4

(b)



close

S

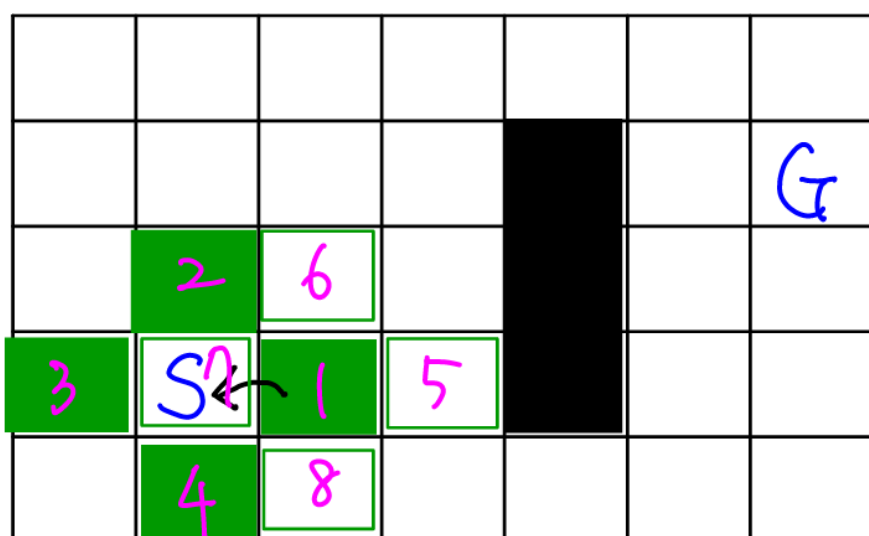
1

Open

2

3

4



close

S

1

Open

2

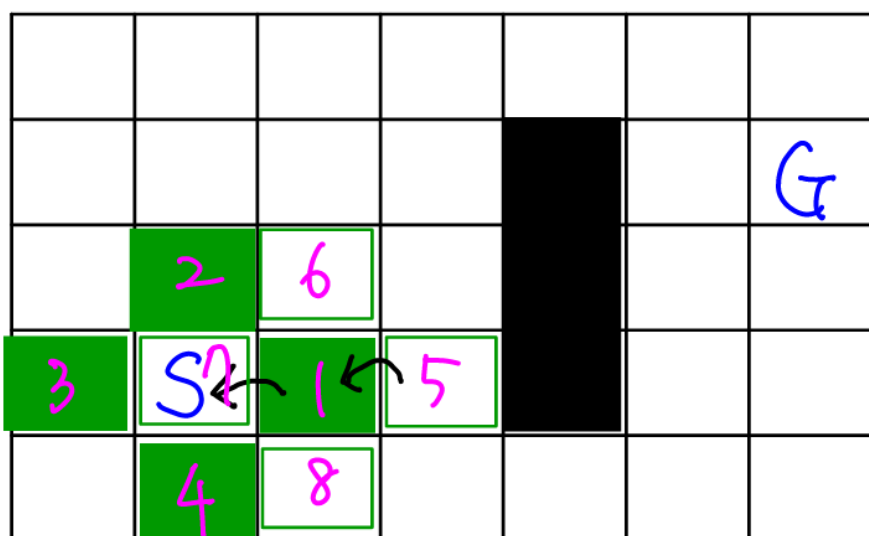
3

4

5

6

8



close

S

1

5

Open

2

3

4

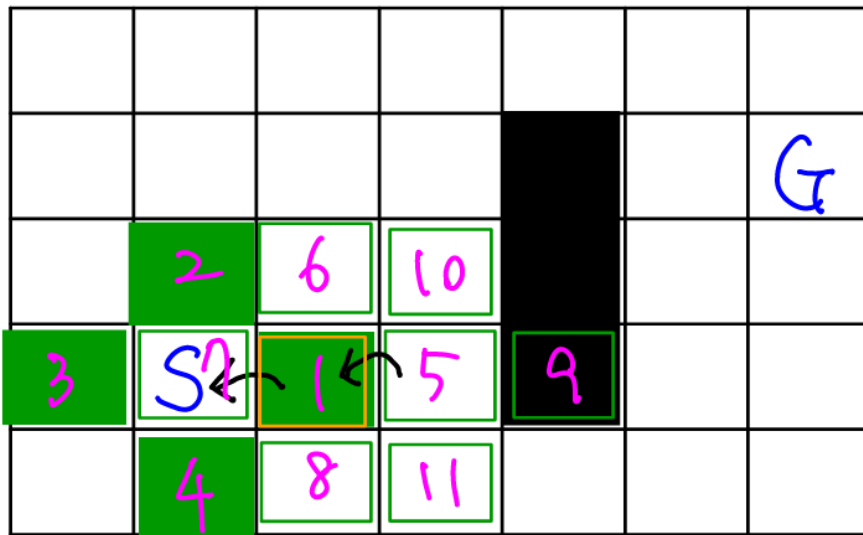
~~5~~

6

8

7 ≡ S

(1)



close

5

1

5

Open

2

3

4

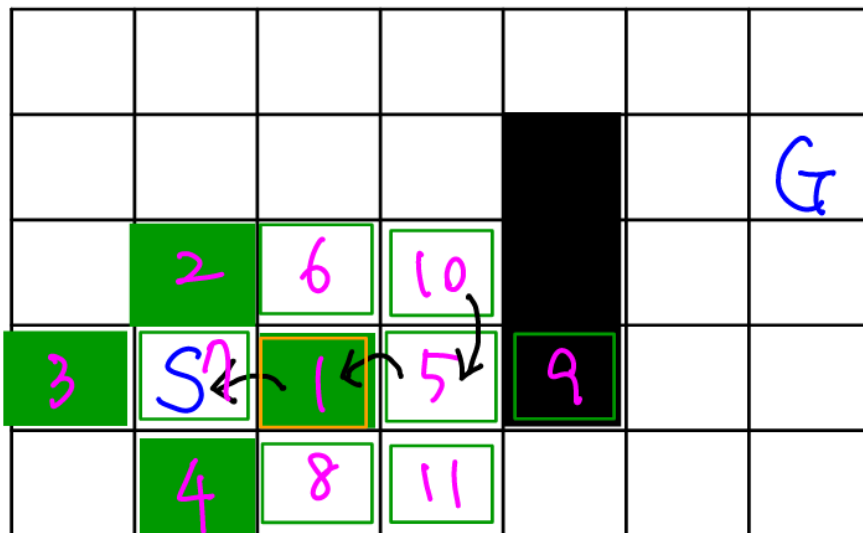
6

8

10

11

5's neighbors \equiv 9, 10, 1, 11
 9 = wall (not a neighbor)



close

5

1

5

10

Open

2

3

4

6

8

~~10~~

11

			15			
		16	13	14		G
	2	6	10	12		
3	S	1	5	9		
	4	8	11			

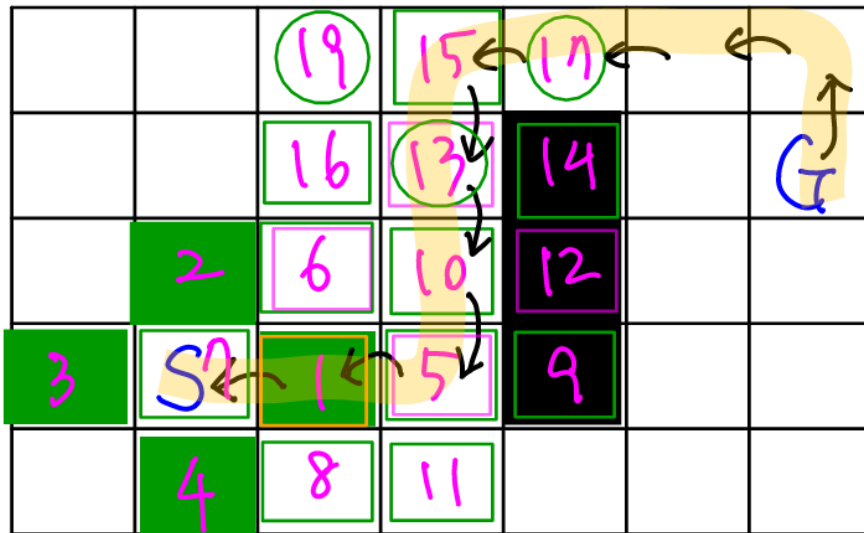
close	Open
S	2
1	3
5	4
10	6
13	8
	11
	15
	16

18

		19	15	17		
		16	13	14		G
	2	6	10	12		
3	S	1	5	9		
	4	8	11			

close	Open
S	2
1	3
5	4
10	6
13	8
15	11
	15
	16

* final result



close

Open

5

2

1

3

5

4

10

6

13

8

15

11

17

16

:

19

:

:

(2018.5.6)