

In this project, I implemented the basic path searching algorithm A* and JPS in C++, with 3 different heuristic distance(Euler distance, Manhattan distance and Diagonal distance) and Tie Breaker.

Given a 50x50x25 grid node map with 26 connection in between:

- Influence of Tie Breaker with Euler distance as heuristic (same path):

Algorithm	Tie Breaker	Path Cost	Time
A*	No	1.2m	301ms
	$1+\text{rand}()/10^7$	221m	293ms
	$1+\text{rand}()/10^9$	25m	143ms

- Influence of different heuristic(same path):

Algorithm	Heuristic			Path Cost	Time
	Manhattan	Euler	Diagonal		
A*	X			18905m	24ms
		X		5539m	89ms
			X	1m	22ms
JPS	X			8m	3ms
		X		8m	3ms
			X	6m	1659ms

- Conclusion:

JPS works way more faster than A* in a complex map as theoretically expected, except for Diagonal distance as heuristic, still try to figure it out.

A* works best with Diagonal distance, then with Manhattan and at last with Euler, which tells Euler distance may probably be the worst choice as heuristic.

Also, a proper Tie Breaker may highly improve the performance of A*.