

Game dev: Dijkstra to A*

Ricard Pillosu - UPC



Solution A*



A*

- A* is like Dijkstra, but it will take in account distance to goal
- It is considered a [Greedy Algorithm](#) (focus only on the local problem)
- Just by adding a distance heuristic to priority we are done:
 - Manhattan distance
 - Squared root distance
 - Distance

TODO (before)

- Make sure you finish Dijkstra and that you understand it 100%
- Concept of “Cost so far” must be implemented since A* can revisit nodes

TODO 1

- We want to stop propagation as soon as we find the goal tile
- When clicked remember in a property the goal position in tile coordinates
- Stop propagation as soon as goal is found
- You can test with Dijkstra

TODO 2

- Create a `PropagateAStar()` as a copy from Dijkstra
- Now to add a neighbor in the frontier:
 - Add to the cost that goes into the queue the distance heuristic of choice
- The heuristic should be one of three:
 - Manhattan distance
 - Square root distance
 - Distance

Homework

- Create a full new module for A* pathfinding:
- It should receive an abstraction of the scene (all navigation data)
- It should remember the last path generated