Global Path Planning

Brian Zhu

August 2, 2020

1 A*

- Generalization of Djikstras
- Djikstras is flawed because it dosent plan around enviorment such as obstacles. This is why its called an uninformed algorithm
- Takes in a Graph, starting cell, the goal cell, heuristic function
  - G graph
  - n_o starting cell
  - n_{goal} goal cell
  - $h(c,t) = \sqrt{(c_x - t_x)^2 + (c_y - t_y)^2}$
    - Distance from current, c, to target, t

1.1 Bookkeeping

- Unvisited Cells, seen: $O = \{n_o\}$
- Visited Cells: $C = \emptyset$
- Parent Cells: P
  - Neighbor to current cell, have lowest cost to reach from current cell
- f-score: $n_i = f^{[i]}$
  - Cost of arriving at n_{goal} through n_i
- g-score: $n_i = g^{[i]}$
  - Cost of arriving at n_i through n_o

1.2 Algorithm

1. Take cell w/ lowest f-score
   - starts as $n_o = n_{c(urrent)}$
2. Check if $n_c = n_{goal}$
   - Stopping condition
3. Remove n_i from O and add to C
4. Consider neighbors
   - For every n_i in neighbors:
   - if weve seen n_i ($n_i \in C$), continue
   - else

- add n_i to O
- calculate $g_{tentative} = g^{[n_i]} + dist(n_c, n_i)$
  - distance function is reading off of the distance to the cell, looked up in the adjacency matrix, G
  - calculated as tentative because it may be greater than the g-score that we already found which means we would discard it and move on to next neighbor since we already found shortest path to that cell
- if $g_{tentative}$ the shortest distance so far, update
- (parent) $P^{[n_i]} = n_c$
- $g^{[}n_i] = g_{tentative}$
- $f^{[n_i]} = g^{[n_i]} + h(n_i, n_{goal})$

5. go to 1.