A* Pathfinding

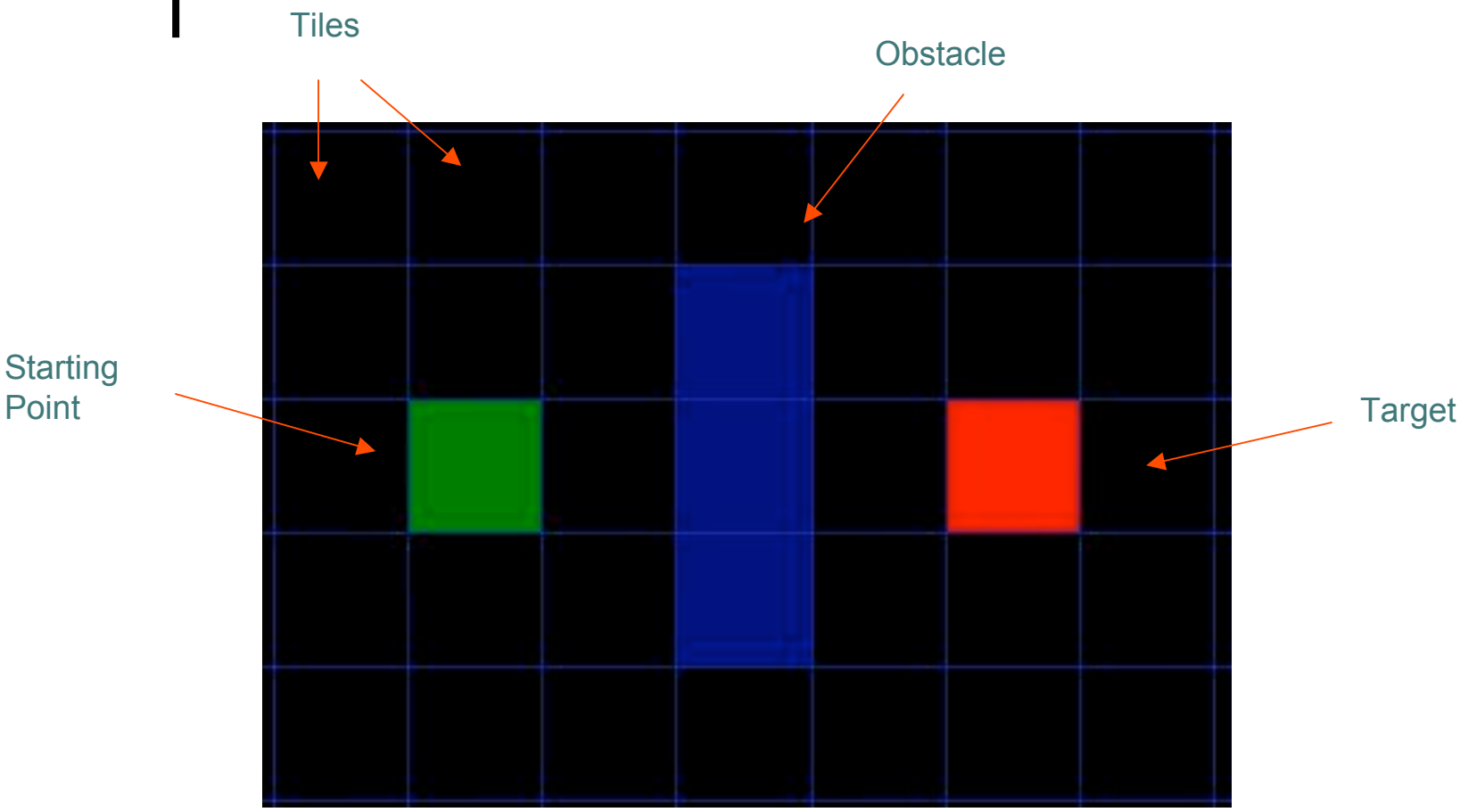
- The A* pathfinding algorithm is an exhaustive search algorithm which is guaranteed to find the shortest path between two points.
- The basic assumption is that the search area is tiled (e.g. square tile) and that the animat moves from the center of one tile to the center of the next tile.

A* Algorithm

- Add the starting square to the open list.
- Repeat the following:
  - Look for the lowest F cost square on the open list. We refer to this as the current square.
  - Switch it to the closed list.
  - For each square adjacent to this current square
    - If it is not walkable or if it is on the closed list, ignore it.
    - If it is not on the open list, add it to the open list. Make the current square the parent of this square. Record the F, G, and H costs of the square.
    - If it is on the open list already, check to see if this path to that square is better, using G cost as the measure. If so, change the parent of the square to the current square, and recalculate the G and F scores of the square.
  - Stop when you:
    - Add the target square to the closed list, in which case the path has been found or
    - Fail to find the target square, and the open list is empty. In this case, there is no path.
- Save the path. Working backwards from the target square, go from each square to its parent square until you reach the starting square.

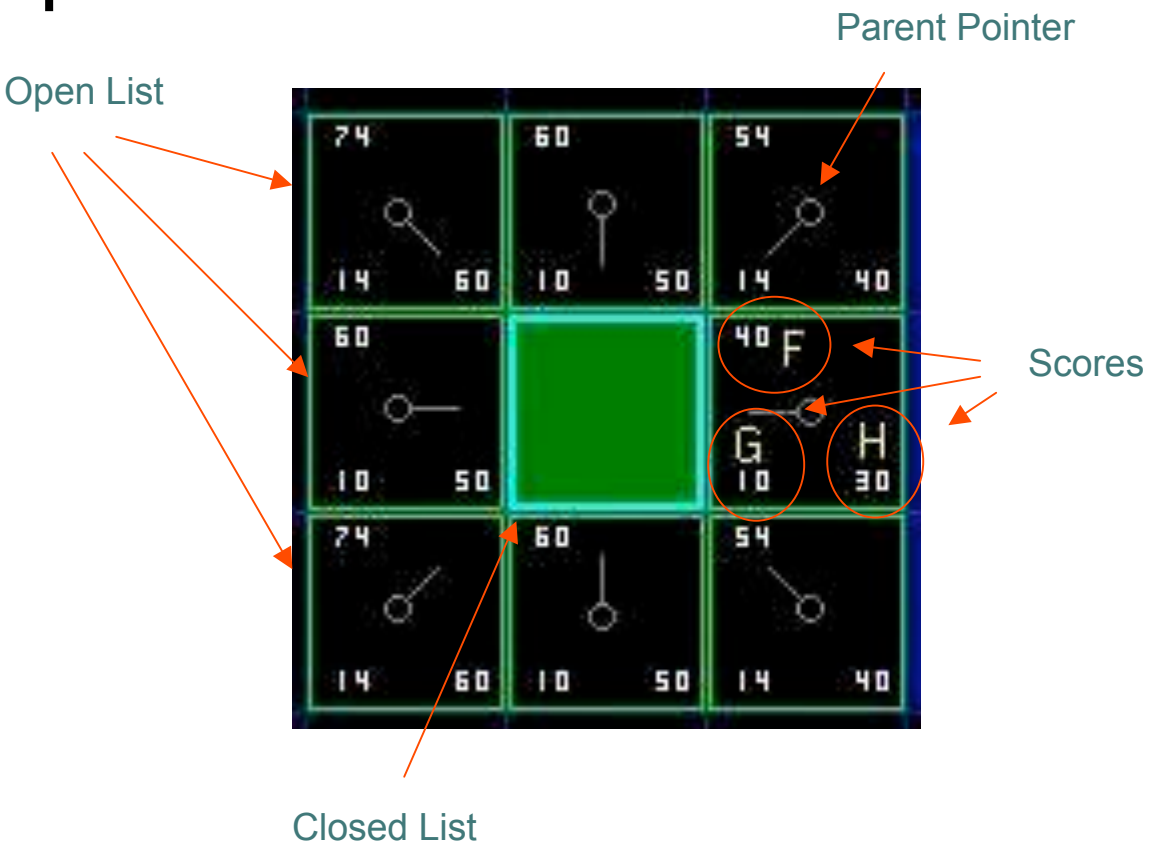
Source: *A* Pathfinding for Beginners*, Patrick Lester, http://www.policyalmanac.org/games/aStarTutorial.htm

The Search Area



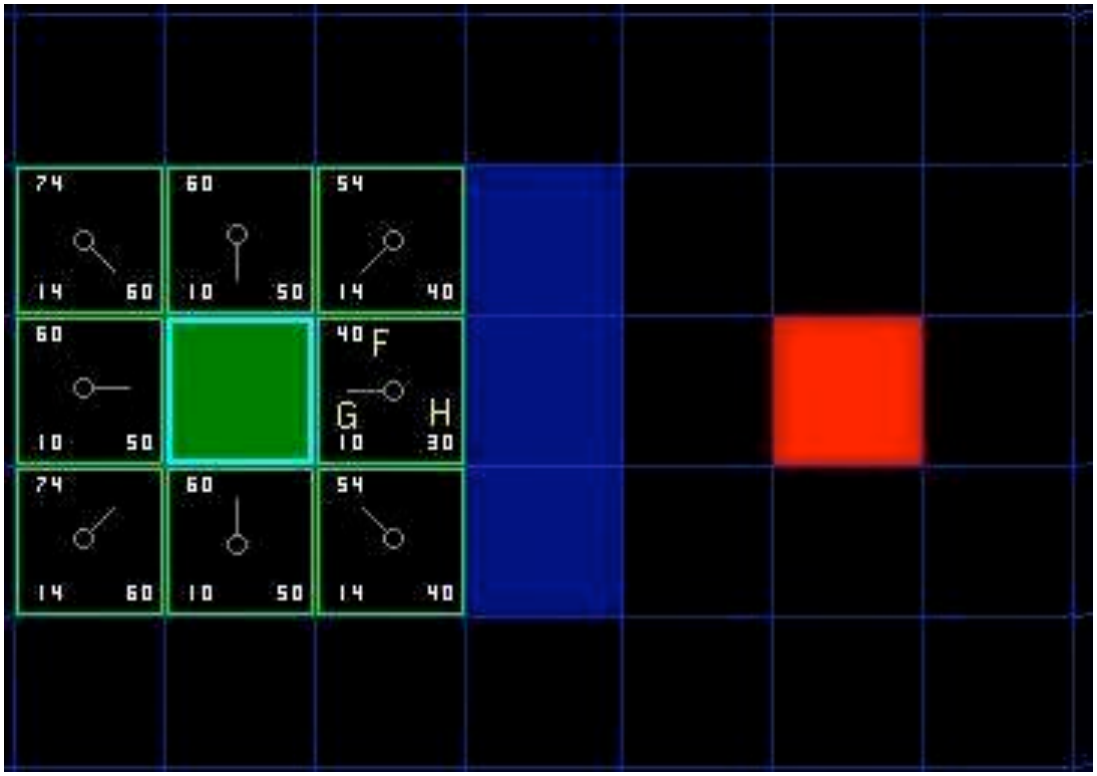
What is the shortest path from the starting point to the target?

Data Structures

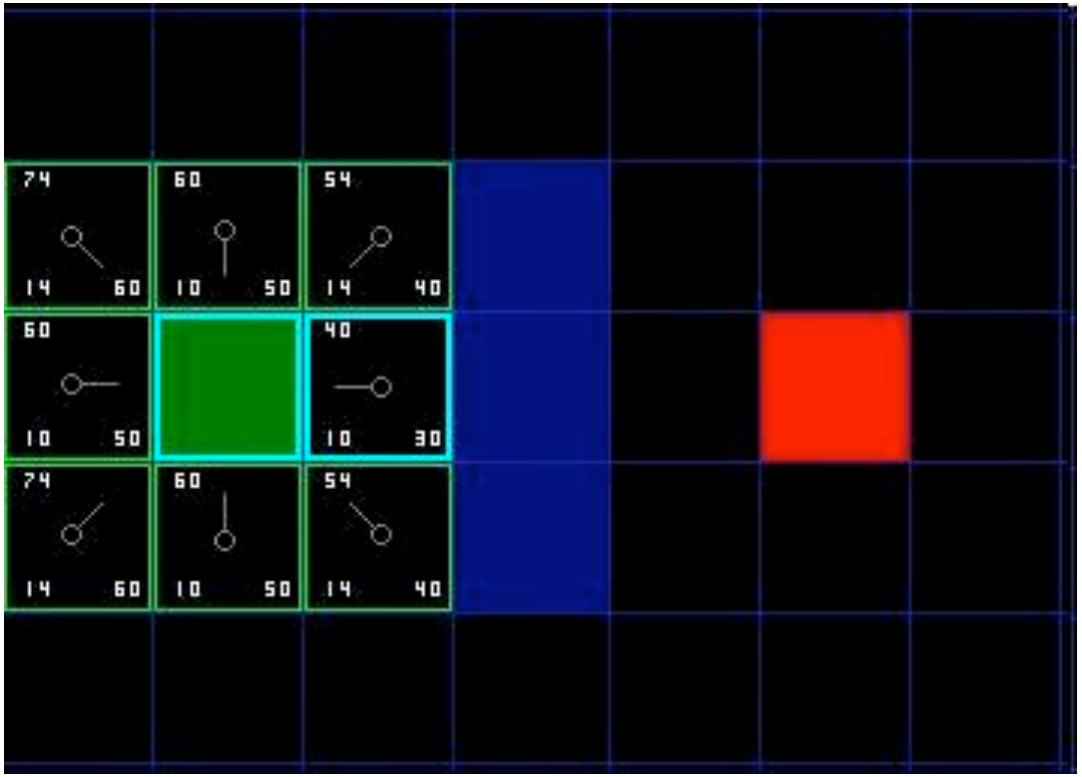


Scores:
F = G + H
G = the movement cost to move from the starting point to the given square following the generated path.
H = the estimated movement cost to move from the given square to the target.

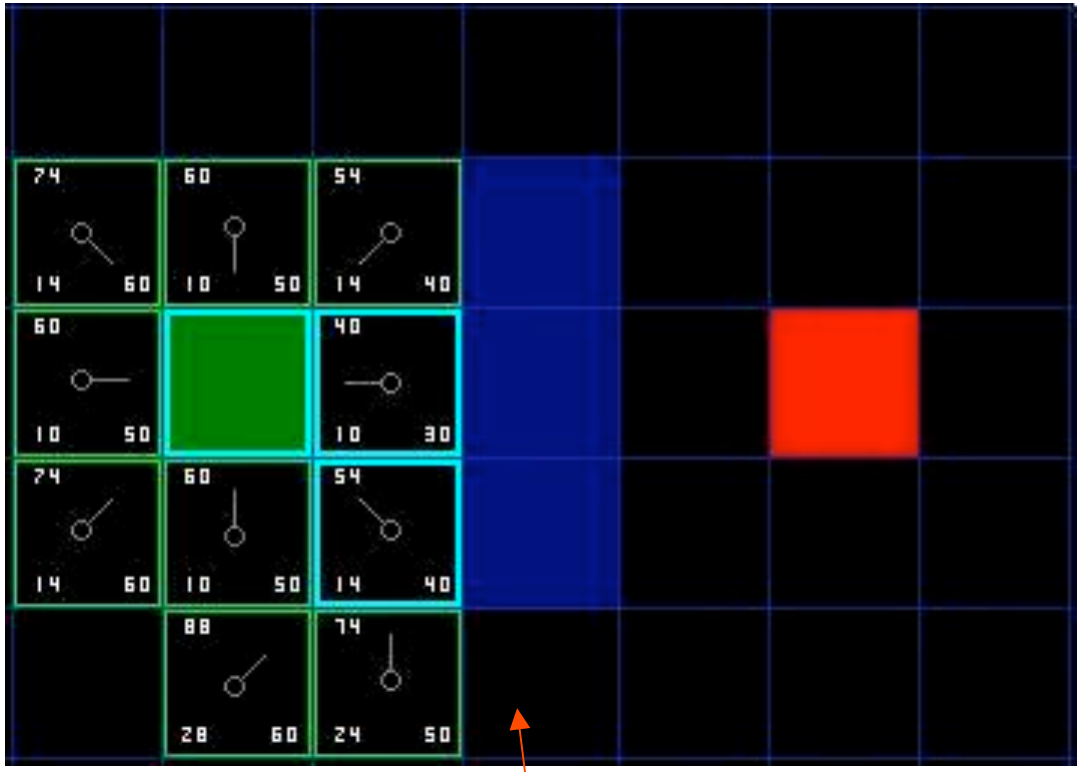
First Iterations



First Iterations

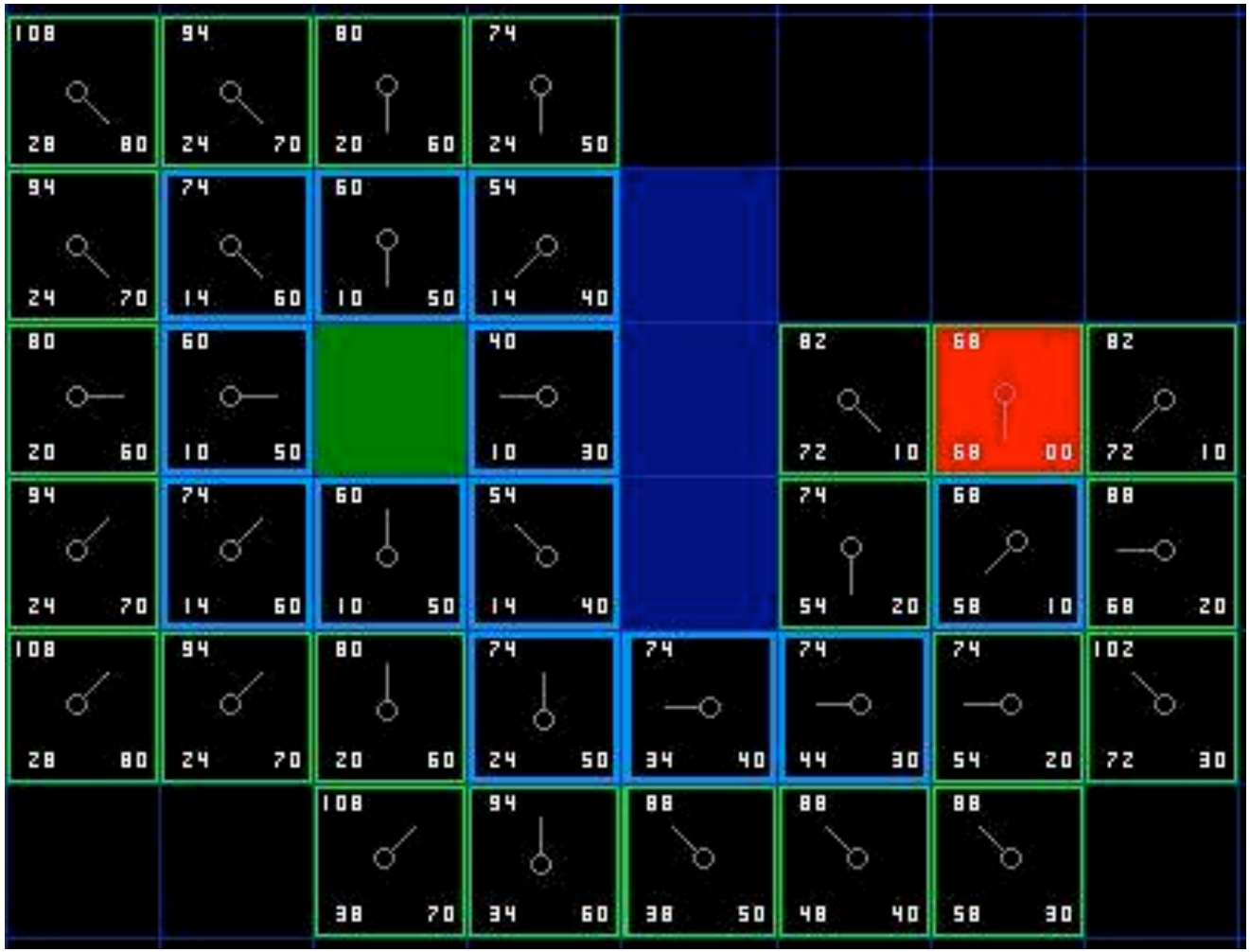


First Iterations

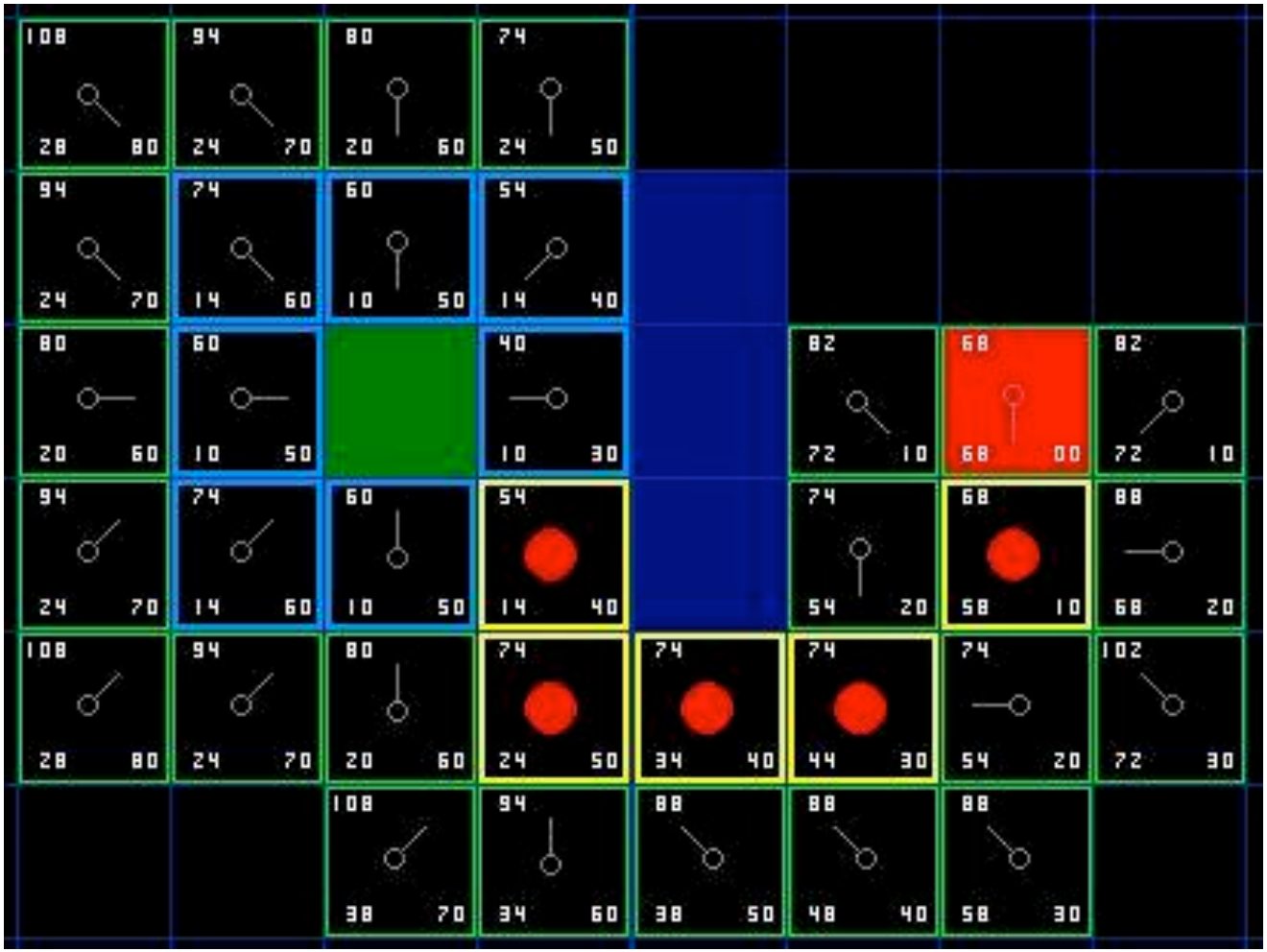


not added – would cut across the corner of the wall

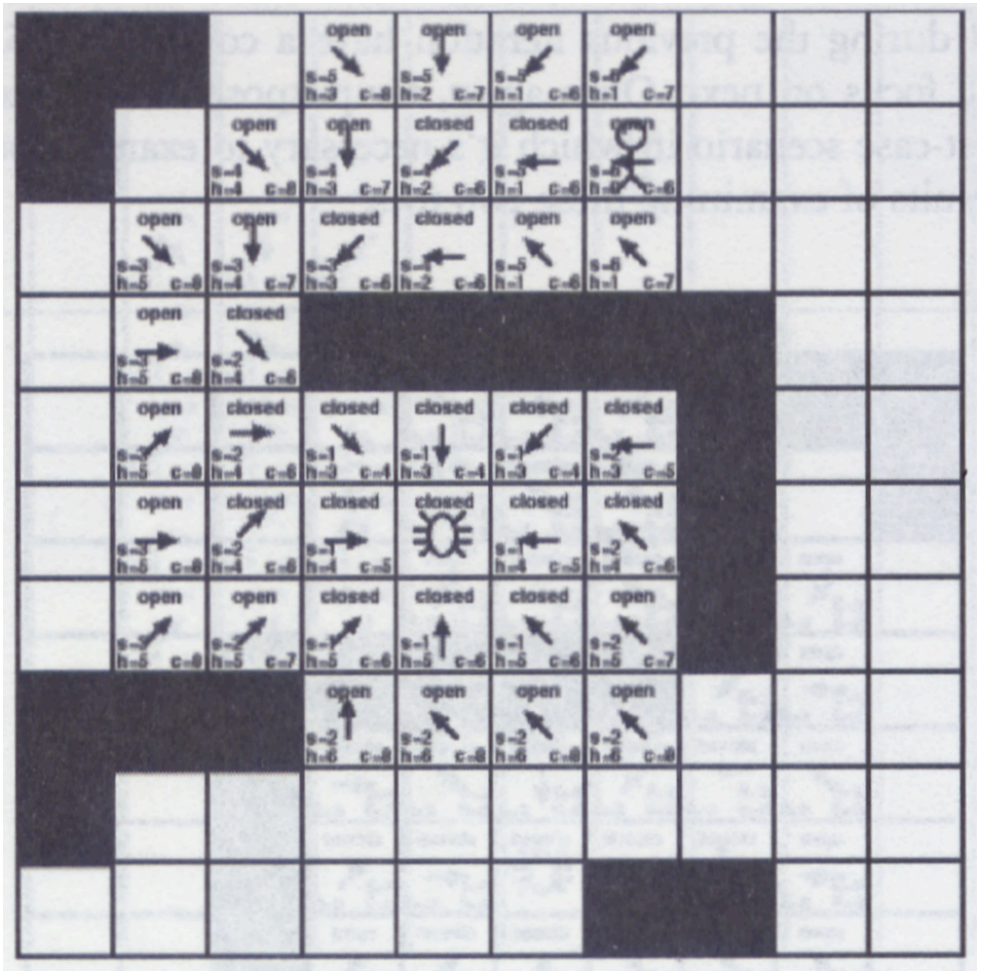
Final Iterations



Final Iterations



Another Example



Source: *AI for Game Developers,* D. Bourg and G. Seemann, O'Reilly, 2004

Another Example