Artificial Intelligence Assignment: PacMan AI

Shivam Swarnkar, ss8464@nyu.edu

October 26, 2016

Abstract

Most algorithms failed to generate answers in given time limit. However, with few modifications, almost every algorithm succeeded to perform decently. The best algorithms (with modifications) were BSF, A*, Iterative Deepening and Evolutionary algorithm without random mutation.

1 Algorithms and Performances

Every algorithm was initially designed by following the standard structure, but few modifications were made to improve the performance. Also, a data structure to hold game state, move which resulted in this current state, and parent state was designed, created and used in following algorithms.

1.1 BSF

Performed well. Best performance : Cleared Level 1, Avg. Score range: 2000-3000. Used standard BSF with number of nodes limit.

1.2 DFS and Iterative Deepening

DFS: Performed below average. Best Performance : 980, Avg. Score range : 250- 750. Used standard DFS with recursion and depth limit. Iterative Deepening: Performed average. Best Performance : 1700, Avg. Score range : 750-1300. Used recursive DFS for repetition.

1.3 A*

Performed average. Best Performance: 1600, Avg. Score range: 650-1300. When a simple Heuristic function was used then performance was average because it was running out of time, but with a better heuristic function (which calculated distance from (edible/non-edible) ghosts, time and other logics) performed better than BSF.

1.4 Hill Climbing

Performed below average. Best Performance : 680, Avg. Score range : 250-450. Used standard HC algorithm, but it got stuck easily in corners and when neither pills or ghosts were around.

1.5 Simulated Anealing

Performed below average. Best Performance : 980, Avg. Score range : 250-350. Used standard SA algorithm, but many times randomly taken bad steps made situation worse. Probably a better heuristic and schedule function will improve the performance.

1.6 Evolutionary Algo 1 and 2

Performed Average, perform better with a better evaluation function. Created a population of 20 and 30 respectively. One algo selected the best, and kept progressing them after elimination while second mutated few top populations using randomly generated different actions