

COSE361 Final Project

2019320137

컴퓨터학과 황상민

a. Screen capture of the result

```
(COSE361) C:\Users\bestm\Desktop\Programming\Python\COSE361 AI\minicontest1>python pacman.py --agent MyAgent --layout test71.lay
Pacman emerges victorious! Score: 805
Average Score: 805.3758567810064
Scores:      805.3758567810064
Win Rate:    1/1 (1.00)
Record:      Win
```

b. Description of my agent

- Search algorithm으로 Uniform cost search를 사용합니다. 팩맨의 현재 위치를 기준으로 cost가 가장 적은 Food로의 Path를 계산하여 다음 행동을 결정합니다.
(ClosestDotAgent와 유사)
- ClosestDotAgent에서는 한 칸=1로 하여 edge cost를 계산했던 것과는 다르게, Edge cost를 customize하여 휴리스틱을 추가하였습니다. 둘 이상의 팩맨이 뭉쳐서 같은 food를 찾게 되면 비효율적이므로, '자신으로부터 가장 가까운 팩맨까지의 manhattan distance'에 반비례하는 값을 휴리스틱 값으로 넣어 주변 Pacman과 너무 가까워지지 않도록 했습니다.
- 또한 ClosestDotAgent에 비해 Path search 횟수를 줄였습니다. getAction()을 호출할 때마다 새로 path를 계산하는 대신, getAction 호출이 10회 일어날 때마다 path를 계산하고, 10회 동안은 이전에 계산해둔 path 상으로 이동하도록 했습니다.

c. Discussions

1) Cases where my agent is better than the baseline

Food가 맵 전역에 흩뿌려져 있을 때 제가 구현한 agent가 평균적으로 더 성능이 높을 것입니다. Baseline은 팩맨 간 거리를 고려하지 않고 모두가 가까운 food만을 찾아 움직이기 때문에, 여러 팩맨이 같은 food를 찾아 이동하는 경우가 발생할 수 있습니다. 반면에 제가 구현한 agent는 팩맨끼리 거리를 두고 맵 구석구석에 있는 food를 찾아 움직이도록 유도하기 때문에 그러한 비효율을 방지할 수 있습니다.

2) Cases where my agent is worse than the baseline

Food가 맵 한쪽에 몰려 있을 때 제가 구현한 agent가 성능이 더 낮게 나올 가능성이 있습니다. 팩맨이 뭉쳐져서 비슷한 지역의 food를 탐색해야 하는데, 휴리스틱 값으로 인해 팩맨이 서로 떨어져 움직이게 되어 길을 돌아 간다든가 하는 경우가 발생할 수 있습니다.

3) Custom question - 매 Action마다가 아닌 일정 주기마다 Path를 갱신하여 움직이는 구현 방식에서 발생할 수 있는 문제점은?

매 Action의 결과를 즉각적으로 반영하며 움직이는 것이 아닌, 몇 step 이전의 결과를 가지고 움직이는 것이므로 팩맨이 이미 획득한 food로 움직인다고 하는 문제가 발생할 수 있습니다.