

기초 인공지능 Assignment02 – Pacman 보고서

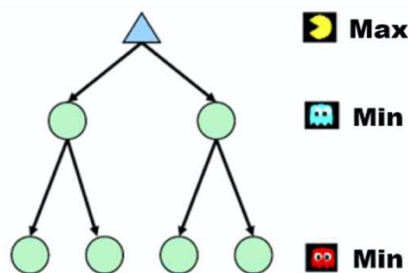
2021.10.16 11:00 PM

20181202 김수미

1. 알고리즘 구현 방법 설명

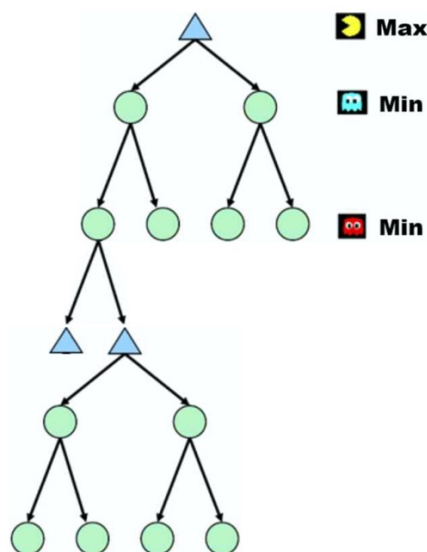
1) Minimax Agent

해당 알고리즘에서는 팩맨을 max agent로 간주하며, 유령들은 팩맨의 적으로써 모두 min agent로 간주한다. 강의자료에서 살펴본 Minimax Tree에서는 max agent와 min agent가 각각 1개씩이었기 때문에 max agent아래 1level의 min agent만 존재했지만, 팩맨 게임에서는 min agent가 2개 이상 존재할 수 있기 때문에 max agent아래 2level 이상으로 min agent들이 존재할 수 있다. 즉 모든 min agent의 action에 따른 경우의 수를 살펴봐야 하기 때문에, tree는 과제 명세서의 예시와 같은 모양으로 생성되어야 한다.



〈 그림. 팩맨에서 Minimax game tree의 예시 〉

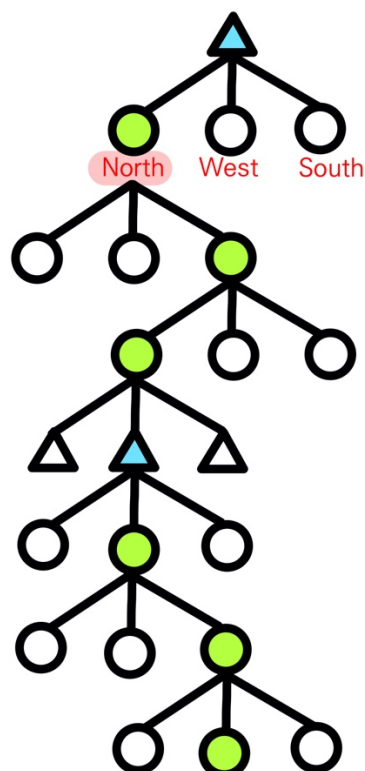
또한 일반적인 tree에서는 하나의 노드와 그 노드의 자식 사이의 깊이를 1-depth로 간주하지만, 해당 알고리즘에서는 모든 agents, 즉 팩맨과 유령의 움직임이 모두 1번씩 결정되어야 이를 1-depth로 간주한다. 즉 2-depth로 tree를 형성하는 경우 minimax tree는 아래와 같은 모양으로 형성된다.



〈 그림. 팩맨 Minimax game tree에서 2-depth tree가 형성되는 경우 〉

이 때 Minimax tree에서 최하단 말단 노드들의 효용 가치는 제공된 코드에서 evaluation function 을 제공하기 때문에 이를 사용하여 값을 구했다. 팩맨 게임에서의 효용 가치는 곧 게임 Score이다. 각 상태에서의 Score 값은 제공 코드의 self.evaluationFunction 통해 return 받을 수 있다.

각 노드의 확장은, agent가 취할 수 있는 action에 근거한다. 즉 팩맨 또는 유령이 이동할 수 있는 방향(north, south, west, east)의 개수만큼 자식 노드가 생성되고, 각 노드는 direction과 해당 위치로 이동한 경우의 Score 값을 데이터로 가진다.



〈 그림. 2-depth tree의 말단노드에서 Recursive 하게 올라와 이동 방향을 결정하는 과정 〉

팩맨은 실제로 결정한 방향으로 이동하며, 팩맨이 유령과 만나 게임이 끝나거나 팩맨이 모든 흰색 원을 먹어 게임에서 이길 때 까지 위 과정을 반복한다.

2) AlphaBeta Pruning Agent

위의 tree 그림들에서는 생략되긴 했지만, 실제로 Minimax 알고리즘의 Minimax tree에서는 가능한 모든 노드가 생성된다. 이렇게 되는 경우 depth의 깊이가 깊어지면 time complexity가 급격하게 높아지게 되는데, 이 때 AlphaBeta Pruning을 사용하여 방문할 필요가 없는 node에 대해서는 자식 node를 확장시키지 않음으로써 계산 시간을 단축할 수 있다. 이를 구현하는 방법은 어렵지 않다. Minimax Agent 코드의 일부분만 수정해주면 된다.

먼저 shared value a(alpha)와 b(beta)를 추가하여 alphaBeta, minValue, maxValue 함수들이 이를 매개변수로 받을 수 있도록 코드를 수정한다. 이 때 변수 a는 tree를 확장하면서 생성되는 노드들의 Score 값 중 가장 큰 값을 저장하며, 변수 b는 tree를 확장하면서 생성되는 노드들의 Score 값 중 가장 작은 값을 저장한다.

특정 노드가 해당 상태에서 agent가 이동할 수 있는 방향의 수 만큼 자식 노드를 형성할 때, 자식 노드의 Score 데이터 값이 a보다 작거나(max agent) b보다 큰 경우(min agent) 해당 자식 노드를 생성하지 않으며 이로 인해 해당 자식 노드부터는 더이상 tree가 확장되지 않는다. 기존 Minimax tree에서는 해당 자식 노드 쪽으로의 가지도 계속 확장되었을 것이다. AlphaBeta Pruning 알고리즘은 이처럼 어차피 선택되지 않을 자식 노드들에 대한 tree의 확장을 방지함으로써 코드 수행 시간을 단축시킨다.

3) Expectimax Agent

Expectimax Agent의 경우 Min node 대신 Expect 값을 저장하는 Expect node가 사용된다. Expect node는 자식 노드 중에서 가장 작은 Score값을 가지는 노드를 선택하는 Min node와 달리, 말단노드부터 해당 방향으로의 이동이 일어날 확률과 해당 방향으로 이동했을 때의 Score의 곱을 저장한다. 이렇게 저장된 값들 중 가장 큰 값이 Max node에서 선택되는 것이다.

팩맨 또는 유령이 이동할 수 있는 방향은 정상적인 map의 경우라면 1개부터 4개까지 존재할 수 있으므로, $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}$ 이렇게 총 4가지 확률값이 노드가 가지는 Score 값과 곱해질 수 있다.

코드 구현의 경우, Expectimax Agent에서는 위의 두 Agent 코드에서 사용한 minValue 함수 대신 expectValue 함수를 사용했으며, 이 함수는 자식 노드 중 가장 작은 Score값을 가지는 노드의 direction과 Score 값을 return 하는게 아니라, 모든 자식 노드들의 (확률 x Score)값의 합과 direction을 return 한다. maxValue 함수는 Minimax Agent의 maxValue 함수와 동일하다.

2. 실행 캡처 화면

1) Minimax Agent 승률

```
Win Rate: 66% (660/1000)
Total Time: 278.1547517776489
Average Time: 0.2781547517776489
=====
```

```
Win Rate: 61% (619/1000)
Total Time: 297.4957137107849
Average Time: 0.2974957137107849
=====
```

```
Win Rate: 64% (644/1000)
Total Time: 224.83706092834473
Average Time: 0.22483706092834474
=====
```

총 세번의 실행 결과 화면을 캡처한 것이다. 모든 실행의 승률은 50%~70% 사이를 만족한다.

2) time_check.py 출력 실행시간

① Small Map

```
Win Rate: 1% (3/300)
Total Time: 75.63834714889526
Average Time: 0.2521278238296509
=====
----- END MiniMax (depth=2) For Small Map
```

```
Win Rate: 0% (2/300)
Total Time: 56.42879319190979
Average Time: 0.18809597730636596
=====
----- END AlphaBeta (depth=2) For Small Map
```

② Medium Map

```
Win Rate: 1% (4/300)
Total Time: 137.3456995487213
Average Time: 0.4578189984957377
=====
----- END MiniMax (depth=2) For Medium Map
```

```

Win Rate: 3% (11/300)
Total Time: 126.70726776123047
Average Time: 0.42235755920410156
=====
----- END AlphaBeta (depth=2) For Medium Map

```

③ Minimax Map

```

Win Rate: 36% (366/1000)
Total Time: 67.03524208068848
Average Time: 0.06703524208068848
=====
----- END MiniMax (depth=4) For Minimax Map

```

```

Win Rate: 35% (353/1000)
Total Time: 46.26205277442932
Average Time: 0.046262052774429324
=====
----- END AlphaBeta (depth=4) For Minimax Map

```

총 세가지 Map(Small Map, Medium Map, Minimax Map)에 대한 실행 결과 화면을 캡처한 것이다. 모든 실행 결과 AlphaBeta Agent의 수행 속도가 Minimax Agent의 수행 속도보다 더 빨랐다.

3) Expectimax Agent 승률 및 Score

```

-----Game Results-----
Average Score: 4.66
Score Results: 532, 532, 532, -502, -502, -502, -502, 532, 532
, 532, -502, -502, 532, 532, 532, 532, -502, 532, -502, -502, 532
, -502, -502, 532, 532, 532, 532, -502, -502, 532, 532, -502, 532
, 532, -502, -502, -502, -502, -502, -502, 532, -502, -502, -502
, 532, 532, -502, 532, -502, -502, -502, 532, 532, 532, 532, -502
, 532, 532, -502, -502, -502, -502, 532, 532, 532, 532, 532, -502
, -502, -502, 532, 532, -502, -502, 532, -502, 532, -502, 532, 532
, -502, 532, -502, -502, -502, 532, -502, -502, 532, 532, -502, -502
, -502, -502, 532, -502, 532, 532, -502, 532
Record: Win, Win, Win, Lose, Lose, Lose, Lose, Win, Win, Win, Lose
, Lose, Win, Win, Win, Win, Lose, Win, Lose, Lose, Win, Lose, Lose
, Win, Win, Win, Win, Lose, Lose, Win, Win, Lose, Win, Win, Lose
, Lose, Lose, Lose, Lose, Lose, Win, Lose, Lose, Lose, Win, Win
, Lose, Win, Lose, Lose, Lose, Win, Win, Win, Win, Lose, Win, Win
, Lose, Lose, Lose, Lose, Win, Win, Win, Win, Win, Lose, Lose, Lose
, Win, Win, Lose, Lose, Win, Lose, Win, Lose, Win, Win, Lose, Win
, Lose, Lose, Lose, Win, Lose, Lose, Win, Win, Lose, Lose, Lose, Lose
, Win, Lose, Win, Win, Lose, Win
Win Rate: 49% (49/100)
Total Time: 0.8296568393707275
Average Time: 0.008296568393707276
=====

```

```

-----Game Results-----
Average Score: 25.34
Score Results: -502, 532, -502, 532, 532, 532, 532, -502, -502
, 532, -502, -502, -502, -502, -502, 532, -502, 532, 532, -502
, -502, -502, 532, 532, -502, -502, 532, 532, 532, 532, -502
, -502, 532, -502, 532, 532, 532, -502, -502, 532, -502, -502
, 532, 532, -502, 532, -502, -502, -502, -502, -502, -502
, -502, 532, 532, -502, 532, 532, 532, 532, 532, -502, -502
, 532, -502, 532, 532, 532, 532, 532, -502, -502, -502, -502
, 532, 532, 532, -502, 532, -502, 532, 532, 532, 532, -502, -502
, 532, -502, -502, -502, 532, 532, -502, -502, 532, 532, 532
Record: Lose, Win, Lose, Win, Win, Win, Win, Lose, Lose, Win, Lose
, Lose, Lose, Lose, Lose, Win, Lose, Win, Win, Lose, Lose, Lose, Win
, Win, Lose, Lose, Win, Win, Win, Win, Lose, Lose, Win, Lose, Win
, Win, Win, Lose, Lose, Win, Lose, Lose, Win, Win, Lose, Win, Lose
, Lose, Lose, Lose, Lose, Lose, Lose, Lose, Win, Win, Lose, Win, Win
, Win, Win, Win, Win, Lose, Lose, Win, Lose, Win, Win, Win, Win, Win
, Lose, Lose, Lose, Lose, Lose, Win, Win, Win, Lose, Win, Lose, Win
, Win, Win, Win, Lose, Lose, Win, Lose, Lose, Lose, Win, Win, Lose
, Lose, Win, Win, Win
Win Rate: 51% (51/100)
Total Time: 0.6379497051239014
Average Time: 0.0063794970512390135
=====

```

```

-----Game Results-----
Average Score: 4.66
Score Results: 532, 532, -502, 532, 532, 532, -502, 532, -502
, -502, -502, -502, -502, 532, 532, 532, 532, 532, 532, -502
, -502, -502, -502, 532, 532, -502, 532, -502, 532, 532, 532
, 532, -502, -502, -502, 532, -502, -502, -502, -502, 532, 532
, -502, 532, -502, 532, -502, 532, 532, 532, -502, 532, 532, -502
, -502, -502, -502, 532, 532, -502, -502, 532, -502, 532, -502
, -502, 532, -502, 532, 532, 532, 532, -502, 532, 532, -502, 532
, -502, 532, -502, -502, 532, 532, 532, -502, -502, -502, -502
, -502, 532, -502, -502, -502, -502, -502, 532, 532, -502, -502, 532
Record: Win, Win, Lose, Win, Win, Win, Lose, Win, Lose, Lose, Lose
, Lose, Lose, Win, Win, Win, Win, Win, Win, Lose, Lose, Lose, Lose
, Win, Win, Lose, Win, Lose, Win, Win, Win, Win, Lose, Lose, Lose
, Win, Lose, Lose, Lose, Lose, Win, Win, Lose, Win, Lose, Win, Lose
, Win, Win, Win, Lose, Win, Win, Lose, Lose, Lose, Lose, Win, Win
, Lose, Lose, Win, Lose, Win, Lose, Lose, Win, Lose, Win, Win, Win
, Win, Lose, Win, Win, Lose, Win, Lose, Win, Lose, Lose, Win, Win
, Win, Lose, Lose, Lose, Lose, Lose, Win, Lose, Lose, Lose, Lose
, Lose, Win, Win, Lose, Lose, Win
Win Rate: 49% (49/100)
Total Time: 0.5375256538391113
Average Time: 0.005375256538391113
=====

```

총 세번의 실행 결과 화면을 캡처한 것이다. 모든 실행 결과 승률은 50% 내외를 만족했고, 이긴 경우에는 532의 Score를, 패배한 경우에는 -502의 Score를 출력했다.