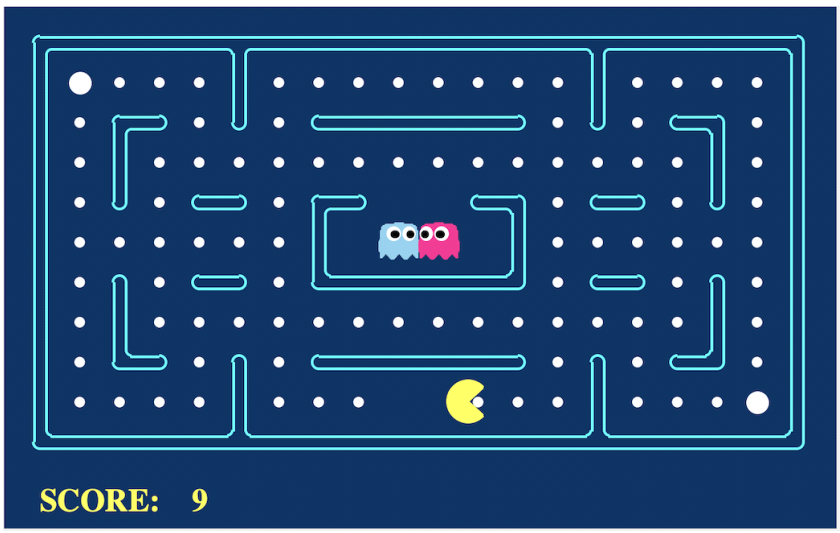
기초 인공지능

Assignment02 - Pacman

Adversarial Search를 활용한 Pacman 구현하기

제출기한 : 2022.10.21.(금), 06:00 p.m. (Late 허용 X)

1. 문제 설명

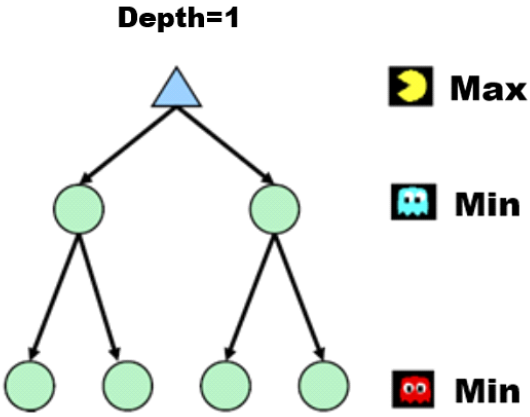


[그림 01] Mediummap 초기 실행 화면

[그림 01]과 같이 게임 맵 안에는 노란색 팩맨과 유령들이 존재한다. 게임은 다음과 같은 규칙들을 따른다. (코드 실행 방법은 README.md 참고)

[Pᗤᗧ•••MᗩN Game Rule]

1. 팩맨이 흰색 작은 원을 먹을 경우, 10 point를 획득
2. 팩맨이 한 step을 갈때마다 1 point씩 감소
3. 팩맨이 모든 작은 원을 다먹을 시 승리, 유령이 팩맨을 잡을 시 패배.
4. 팩맨의 목숨은 한 번뿐이다.
5. 모서리에 존재하는 흰색 캡슐을 먹을 경우, 유령은 흰색 모드로 변하며, 유령의 속도가 반으로 감소하고, 팩맨을 쫓지 않는다.
6. 팩맨이 흰색 유령을 잡을 시 200 point 획득
7. 게임 승리 시, 500 point 획득
8. 게임 패배 시, 500 point 감소



[그림 02] Pacman에서의 depth와 index 개념

게임 맵에 한 개의 팩맨과 두 개의 유령이 존재한다고 하면, [그림 02]와 같이 트리가 구성된다. 팩맨의 index는 항상 0이며, max agent처럼 행동한다. 반면 유령들은 min agent처럼 행동한다 (index는 각각 1과 2). 또한 일반 트리 개념과 다르게, 모든 agent가 움직여야 depth가 1이다.

파일에는 총 12가지 python 파일이 존재하며, 작성해야 할 파일은 hw02.py이다. 코드 구현시에는 pacman.py에서 정의된 함수들을 활용하여 코드를 구현하면 된다. hw02.py에 ReflexAgent를 구현한 예시가 있으므로 참고할 것.

[각 Algorithm별 구현해야 하는 점]

- **Minimax Agent**
  Minimax를 구현하면, minimaxmap에서 depth가 1,2,3,4일 때, initial state의 value는 각각 9,8,7,-492이다.

```
python pacman.py -p MinimaxAgent -m minimaxmap -a depth=4 -n 1000 -q
```

  MinimaxAgent에서는 다음 명령어 실행 시, 승률이 50%-70%를 만족함을 보여야 한다.

- **AlphaBeta Pruning Agent**

```
python time_check.py
```

  다음 명령어를 실행하면, Mediummap의 depth가 3일때의 Minimax Agent, AlphaBeta Pruning Agent 실행 시간 결과와 Minimaxmap의 depth가 4일때의 Minimax Agent,AlphaBeta Pruning Agent 실행 시간 결과가 각각 자동으로 출력된다. AlphaBeta Agent가 실행시간이 더 빠른 것을 통해, AlphaBeta Agent가 Minimax Agent보다 효율적임을 보인다.

- **Expectimax Agent**

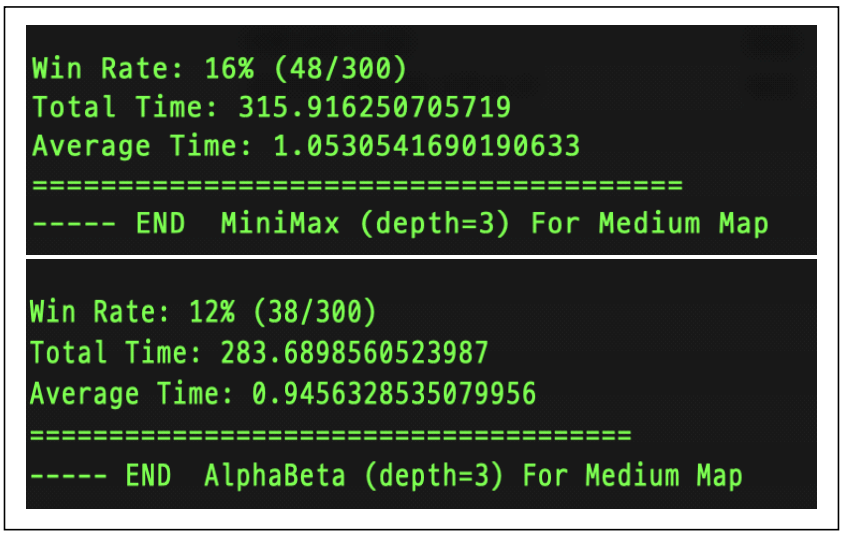
```
python pacman.py -p ExpectimaxAgent -m stuckmap -a depth=3 -n 100 -q
```

다음 명령어 실행시 Expectimax Agent가 50%내외의 승률을 보임을 보이고, 이긴 경우에는 532 혹은 531 score를 출력해야하며, 패배한 경우에는 -502 score를 출력해야 한다.

2. 보고서

보고서 분량 제한은 없으나, 반드시 다음과 같은 내용이 포함되어야 한다.

1. 각 알고리즘마다 구현한 방법에 대한 설명
2. 실행 캡처 화면
   [1] 위에서 제시한 Minimax Agent 명령어의 승률 출력 화면
   [2] time_check.py에서 출력된 실행시간 캡처 화면 (mediummap,minimaxmap 총 2개)
   [3] 위에서 제시한 Expectimax Agent 명령어의 승률 및 Score 출력 화면



[그림 03] time_check.py Medium map 출력 캡처 예시

3. 주의사항

hw02.py 외의 파일 수정 불가. 각 Agent Class 안의 Action 함수 내부에 함수 자유롭게 선언 가능.
copy check 적발시 0점 처리되므로, 반드시 각자 과제를 수행할 것.

4. 제출

hw02.py 파일과 보고서 AI02_학번_이름.pdf 두가지 파일을 압축해 **AI02_학번_이름.zip**으로 사이버 캠퍼스에 업로드한다.