

기초 인공지능

Assignment01 - Maze Game

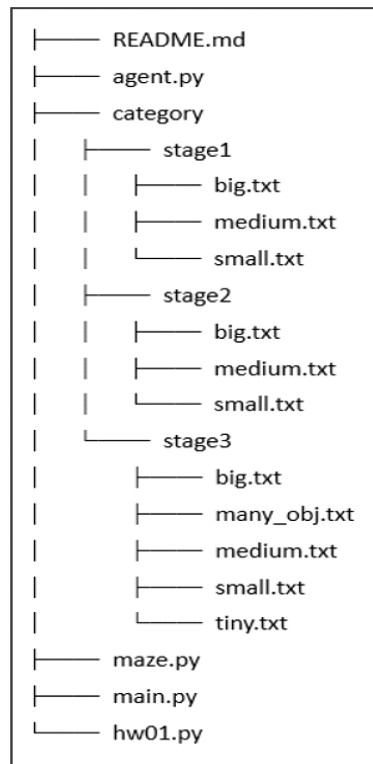
BFS, DFS와 A* Algorithm을 활용한 최단 경로 찾기

제출기한 : 2022.09.30.(금), 6:00 p.m. (Late 허용 X)

1. Requirements

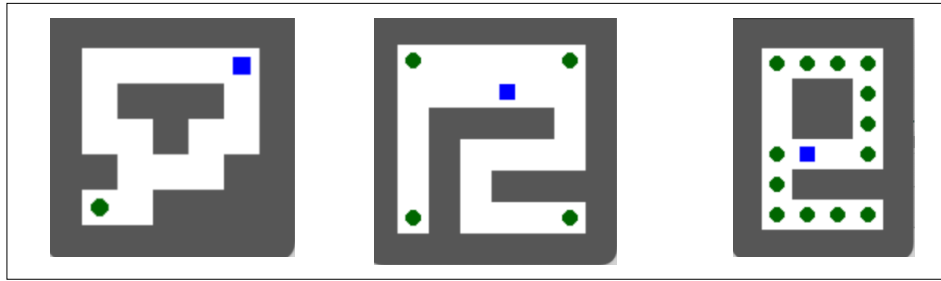
python version>=3.6, pygame library

2. 문제 설명



[그림 01] AI_assignment01 디렉토리 구성

[그림 01]과 같이 과제 디렉토리 안에는 게임 난이도를 나타내는 stage1, stage2, stage3 폴더가 있다. 그리고 각 stage1과 stage2는 크기별 3개의 txt파일이 존재하며 stage3의 경우 5개의 txt파일이 존재한다. 코드를 다운 받은 뒤, pygame을 정상적으로 설치하고 터미널 상에서 코드를 실행하면 [그림 02]와 같은 실행 화면이 나타난다. (코드 실행 방법은 README.md 참고)



[그림 02] 난이도별 초기 실행 화면 모습

파란 사각형은 출발지점이며, 초록색 원이 목표지점을 나타낸다. stage1은 목표지점이 한 개가 있고, stage2는 각 모서리마다 목표지점이 한 개씩 존재하며, 총 네 개의 지점을 반드시 지나가야 한다. 그리고 stage3는 목표지점이 여러 곳이 존재하며, 모든 지점을 통과해야 한다.

이번 과제에서는 python 언어로 BFS,DFS와 A* algorithm을 구현해 최단 경로를 탐색하는 것이 목표이다.

파일에는 hw1.py, maze.py, agent.py, main.py 4가지 소스 파일이 존재하며, 작성해야 할 파일은 hw1.py이다. (채점시 hw1.py 외에는 동일한 소스파일로 채점할 예정이며, hw1.py 외의 소스 파일내의 method 등의 수정은 허용하지 않음.)

[각 stage별 구현해야 하는 점]

▪ Stage 1

Stage 1은 BFS,DFS와 A* algorithm을 각각 이용해 최단 경로를 탐색한다. A* algorithm의 경우에는, Heuristic Function으로 **Manhattan Distance**를 사용한다.

▪ Stage 2

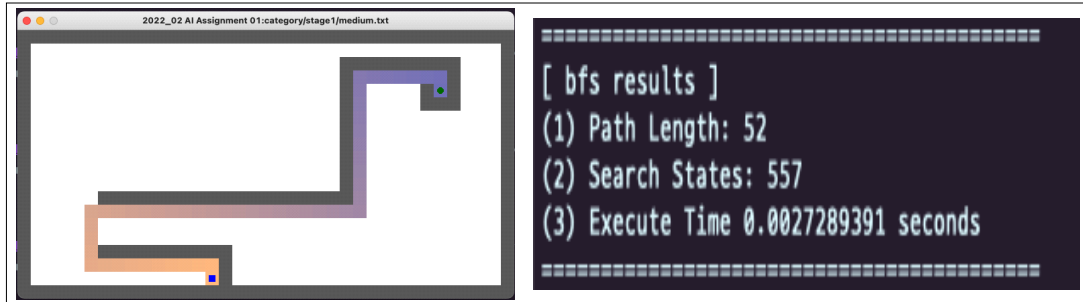
Stage 2는 A* algorithm을 이용해 최단 경로를 탐색한다. 단 Stage 2에서는 Heuristic Function은 **사용자가 정의해야 한다**. 또한 stage2/small.txt, stage2/medium.txt, stage2/big.txt 실행시간 모두 2초를 넘지 않아야 한다.

▪ Stage 3

Stage 3 역시 A* algorithm을 이용해 최단 경로를 탐색한다. Stage 3에서도 Heuristic Function은 **사용자가 정의한다**. 단, 3단계에서는 **Minimum Spanning Tree(MST)** 알고리즘을 활용한 Heuristic Function을 정의해야 한다. Minimum Spanning Tree(MST) 구현 방법은 대표적으로 Kruskal Algorithm, Prim Algorithm 2가지 방법이 있으며, 2가지 방법 중 하나를 선택하면 된다. 구현시 **Weighted A* algorithm**은 허용하지 않는다. 모든 맵의 실행 시간이 60초를 넘지 않도록 한다.

3. 보고서

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



[그림 03] 출력화면 캡처 예시

1. 사용한 라이브러리와 각 알고리즘마다 구현한 방법에 대한 간단한 설명
2. 각 stage별로 출력되는 경로 그림과, 터미널 상에서의 path length와 search states, execute time 출력 캡처 화면 ([그림 03] 참고)
즉 다음과 같은 출력화면이 포함되어야 한다.

- (1) bfs method로 실행한 stage1의 small.txt, medium.txt, big.txt 경로 그림과 출력 캡처 화면
- (2) dfs method로 실행한 stage1의 small.txt, medium.txt, big.txt 경로 그림과 출력 캡처 화면
- (3) astar method로 실행한 stage1의 small.txt, medium.txt, big.txt 경로 그림과 출력 캡처 화면
- (4) astar_four_circles method로 실행한 stage2의 small.txt, medium.txt, big.txt 경로 그림과 출력 캡처 화면
- (5) astar_many_circles method로 실행한 stage3의 tiny.txt, small.txt, medium.txt, many_obj.txt, big.txt 경로 그림과 출력 캡처 화면 캡처 화면

3. Stage2와 Stage3에서 직접 정의한 Heuristic Function에 대한 설명

4. 주의사항

라이브러리는 자유롭게 사용 가능. execute time을 제외한 코드 실행시 출력 화면과 보고서에 첨부된 화면 캡처 내용 반드시 동일해야 한다 (다를 경우 코드 실행 시의 결과를 기준으로 점수를 산정할 것). 채점 시 search states와 execute time의 결과를 고려할 예정이며 추가적인 test case가 있을 수 있음. **copy check 적발시 0점 처리.**

5. 제출

hw1.py 파일과 AI01_학번_이름.pdf 두가지 파일만을 압축해 **AI01_학번_이름.zip**으로 사이버 캠퍼스 에 업로드한다.