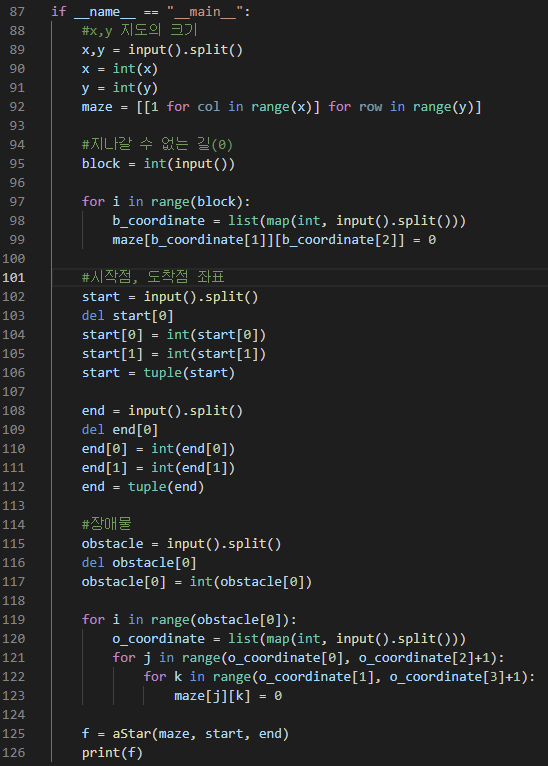
PA05\_NAVI\_알고리즘 풀이

201825153 김지영

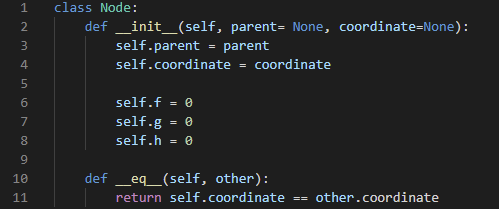
1. 사용자 입력 코드



먼저 값을 입력 받는 코드이다. 89~92코드를 통해 지도의 크기를 x, y값으로 입력 받는다. 그리고 x\*y크기의 지도(maze)를 2차원 배열로 생성한다. 94-99줄을 통해 지나갈 수 없는 길(0)인 셀의 개수와, 그 좌표를 입력 받아 바로 maze에 적용하여 준다. 101-106, 108-112코드를 통해서는 시작점의 좌표(start)와, 도착점의 좌표(end)를 입력 받는다. aStar 알고리즘 함수를 호출할 때 튜플 형식으로 넘겨줄 것이기 때문에 list형을 tuple형으로 변환시켜 준다. 115-123 줄을 통해 장애물의 개수와, 장애물의 좌표를 입력 받는다. 이때 장애물은 지나갈 수 없는 길의 종류이기 때문에 지나갈 수 없는 길을 처리해주었던 것과 같이 maze에 바로 적용해준다.

125-126은 매개변수(지도, 시작점, 도착점)을 aStar 알고리즘 함수로 넘겨주고 함수를 실행하여 도착점에서의 평가함수 값 f(x)을 return 받아 출력한다.

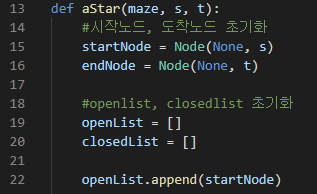
1. Node 클래스 정의



노드 클래스를 사용하여 aStar알고리즘을 실행할 때 각 도착점 별로 f, g, h값의 계산이 용이하도록 재사용 한다. Node는 지나온 부모 노드를 저장하는 parent, 현재 좌표를 저장하는 coordinate, g+h값을 저장하는 f, 출발점에서 현재 점 까지의 맨하탄 거리인 g, 현재 점에서 목표점까지의 유클리드 거리인 h가 있다.

1. aStar 알고리즘

* 변수 및 노드 설정

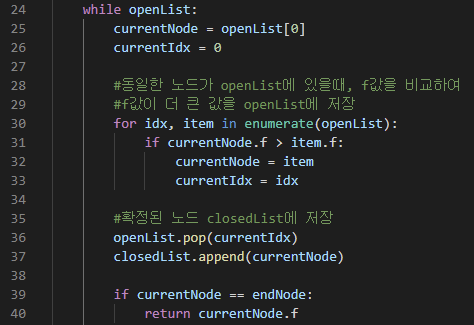


가장 먼저 시작 노드와 도착 노드를 초기화 시켜준다. 이때 시작 점과 도착 점의 좌표 값을 각각 s, t로 받았으므로 coordinate값에는 s와 t를 주고, 시작 노드는 부모 노드가 없으므로 None을, 도착 노드는 부모 노드가 결정되지 않았으므로 None 값을 준다.

또한 현재 노드에서 갈 수 있는 경로를 저장하는 openList와 지나온 노드를 저장하는 closedList를 만들고, 배열로 초기화 시켜준다.

시작 노드인 startNode를 openList에 저장한다.

* ClosedList에 값 저장하기

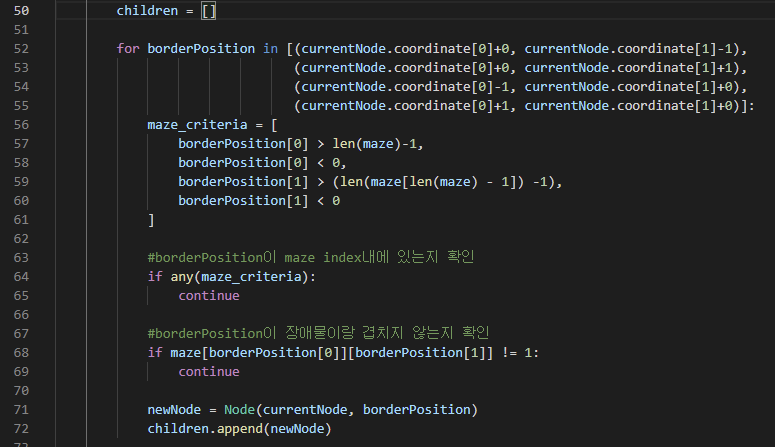


앞으로의 과정은 openList에 값이 있는 동안 반복해서 일어난다. 먼저 현재 노드를 가리키는 currentNode를 openList의 첫 값으로 초기화 시켜준다. openList의 값을 돌며 f값이 동일한 노드가 있으면 더 큰 값을 currentNode로 지정하고 openList에 저장한다.

이렇게 결정된 currentNode를 openList에서 삭제하고, closedList에 저장한다.

만약 currentNode가 endNode값과 같으면, 도착점에 도달한 것이므로 출력 값인 f값을 return해주고 함수를 종료한다.

* 주변 노드 거리 탐색과정

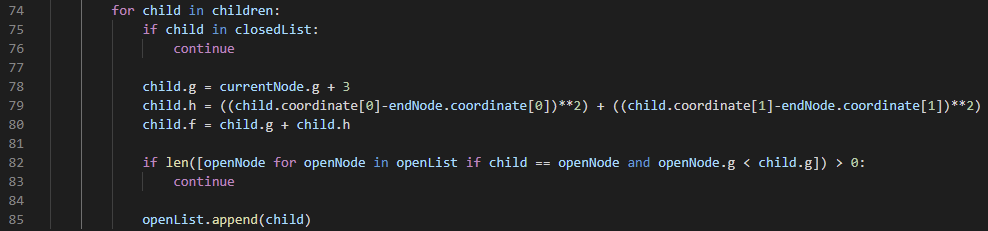


먼저 지나갈 좌표를 모두 저장할 children배열을 생성한다.

현재 노드인 currentNode의 상, 하, 좌, 우의 좌표에 대하여 for문을 돌며 다음을 실행한다. 문제의 조건에서 대각선으로는 이동할 수 없다고 하였으므로 4개의 경우에 대해서만 검사한다. 먼저 이동할 좌표의 x나 y값이 음수인 경우, 혹은 maze의 범위를 벗어나는 경우, 지나갈 수 없으므로 maze\_criteria의 검사를 통해 확인한다. 또한 장애물이나 지나갈 수 없는 좌표(0)도 지나갈 수 없으므로 검사를 통해 통과한다.

이렇게 검사를 거쳐 남은 노드들을 children배열에 append한다.

* Children 중 closedList에 넣을 값 검사



위의 과정을 통해 생성된 children배열을 돌며 child가 이미 closedList에 없는 노드들에 대해 다음을 수행한다. 먼저 g값은 출발점에서 현재 셀까지의 맨하탄 거리이므로 기존 노드의 g값에 3을 더한다. 문제 조건에 의해 각 셀 간의 거리가 3이므로 3을 더해주었다. H값은 유클리드 거리를 나타내므로 현재 child의 좌표와 도착 좌표의 유클리드 값을 계산해 준다. F값은 앞에서 계산한 h와 g값을 더해주어 계산한다.

Child의 값이 openList에 이미 존재하고, openList의 해당 노드의 g값이 child의 g값보다 작은 경우, continue로 지나친다.

이렇게 결정된 child를 openList에 append해준다.