

Künstliche Intelligenz, SoSe21

Project 1

Jonny Lam & Thore Brehmer

23. Juni 2023

1 Q1: Is the exploration order what you would have expected? Does Pacman actually go to all the explored squares on his way to the goal?

- Yes we did expect is this way. Pacman goes the direct way to the goal, which was found by our algorithm. If we wanted to see how Pacman also goes into dead ends, we would have had to implement our algorithm differently. (e.g. save all routes and also backtracking routes)

2 Q2: What difference you observe from the other applied search strategies??explain your observation.

```
1 #1 python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
2 #2 python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
3 #3 python pacman.py -l mediumScaryMaze -p StayWestSearchAgent
4 #4 python pacman.py -l mediumMaze -p SearchAgent -a fn=GreedyBestFirstSearch
5 #5 python pacman.py -l mediumDottedMaze -p SearchAgent -a fn=GreedyBestFirstSearch
6 #6 python pacman.py -l mediumScaryMaze -p SearchAgent -a fn=GreedyBestFirstSearch
```

- The difference between uniformCostSearch and GreedyBestFirst Search compared to the previous strategies (dfs, bfs) is that these two strategies also take the costs of the transitions into account.
- Execution #1 is exactly the same as when using bfs. Since every transition in this version has a cost of 1. (So in this case looking at the costs makes no difference)
- With version #2 however, the StayEastSearchAgent changes the costs so that all routes that do not lead to the east are more expensive. This is how PacMan succeeds in finding the right way through the mediumDottedMaze. Since all of the food points are in the east.
- #3 is very similar. Here PacMan manages to pass the mediumScaryMaze, as the west provides a safe route past the opponents.
- However, it does not work so well with the 4-6# versions. All are similar to the 1#. Although the GreedyBestFirstSearch algorithm is used here, the cost for each transition is always 1 for all courses. So its like running bfs.
- Therefore, the 4# execution also manages to find the shortest path to the destination.

- However, 5# and 6# fail. The shortest path to the destination is still run here (the destination for the SearchProblem). However, neither the food nor the opponents are taken into account, as these are not considered as costs on the transactions and are not taken into account by the SearchProblem.

3 Q3: What happens on openMaze for the various search strategies? Replace manhattanHeuristic with euclideanHeuristic , what difference you notice? which heuristic is better here and why?

- The Euclidean distance calculates the direct path to the target, it can also go diagonally and does not calculate the walls. In contrast to the Manhattan distance, which only follows horizontal and vertical lines to get to the target.

4 Q4: Describe your heuristic used in the foodHeuristic method.

- We calculate the distances to the current position for every food on the field. Then we return the maximum of these values.
- Colloquially, this means that the higher the heuristic, the greater is the distance from the most distant eating position, from this state. The smaller the heuristic, the closer is the most distant eating position, from this state.