TD4136 - Introduction to Artificial Intelligence

Assignment 2: Applying the A* Algorithm

Pablo Zalla Quintanilla and Marc Soler Colomer
pabloaz - 567362 / marcso - 566522

Autumn 2021-2022
30/09/21

NTNU

Norwegian University of
Science and Technology

Introduction

In order to complete this assignment, we used different sources to implement our own A* algorithm. We used the pseudocode and the information of the *Essentials of the A* Algorithm* document provided by the NTNU, along with the programming tips offered in the assignment lecture. To complement this information, we have also been inspired by some internet articles and source codes:

- A* Search Algorithm in Python - https://www.annytab.com/a-star-search-algorithm-in-python/
- Understanding A* Path Algorithms and Implementation with Python - https://towardsdatascience.com/understanding-a-path-algorithms-and-implementation-with-python-4d8458d6ccc7
- Easy A* (star) Pathfinding - https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2

The source code of our A * algorithm can be found attached in the zip file. There are two versions, *Astar1.py* and *Astar2.py*, depending if we consider weighted or unweighted cells.

For both versions, we have used the “Manhattan Distance” as our heuristic (h).

$$\left|x_1 - x_2\right| + \left|y_1 - y_2\right|$$

We strongly believed that is the best heuristic for this case, keeping in mind that diagonal movements are not allowed, and the assignment results confirmed our hypothesis by getting in all tasks the shortest path.

Part 1 - Grid with obstacles

Task 1

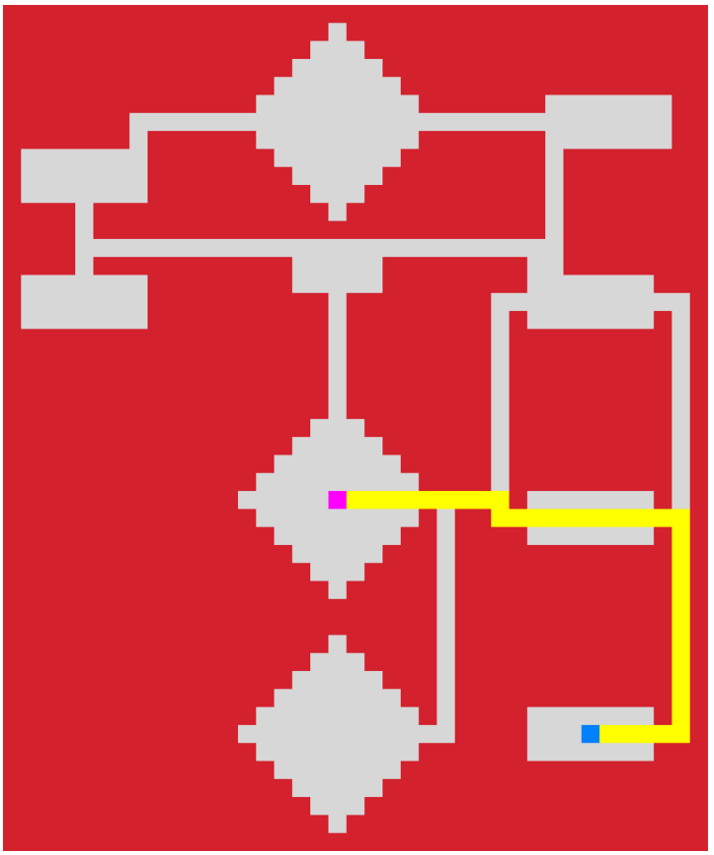
You and your friend arrived at Samfundet only five minutes ago, but you've already managed to get separated. Being the resourceful person that you are, you call your friend, which tells you that they went looking for you and are currently located at Strossa. Your task is therefore to find the shortest path from Rundhallen (your location) to Strossa using your implementation of the A* algorithm.

If we run our algorithm code, Astar1.py, on the Windows Command Prompt, we get the following path, with a cost of 38.

```
-------  PATH    -------
[[27, 18], [27, 19], [27, 20], [27, 21], [27, 22], [27, 23], [27, 24], [27, 25], [27, 26], [27, 27],
 [28, 27], [28, 28], [28, 29], [28, 30], [28, 31], [28, 32], [28, 33], [28, 34], [28, 35], [28, 36],
 [28, 37], [29, 37], [30, 37], [31, 37], [32, 37], [33, 37], [34, 37], [35, 37], [36, 37], [37, 37],
 [38, 37], [39, 37], [40, 37], [40, 36], [40, 35], [40, 34], [40, 33], [40, 32]]
------------------------

PATH COST:     38
```

On the following image there is a visual representation of the path taken, represented with the yellow colour.



As we can see, this is the shortest path to get from the starting position (pink spot) to the goal position (blue spot). There aren't any alternative paths for this task. So we can affirm that the algorithm works properly for this task.

Task 2

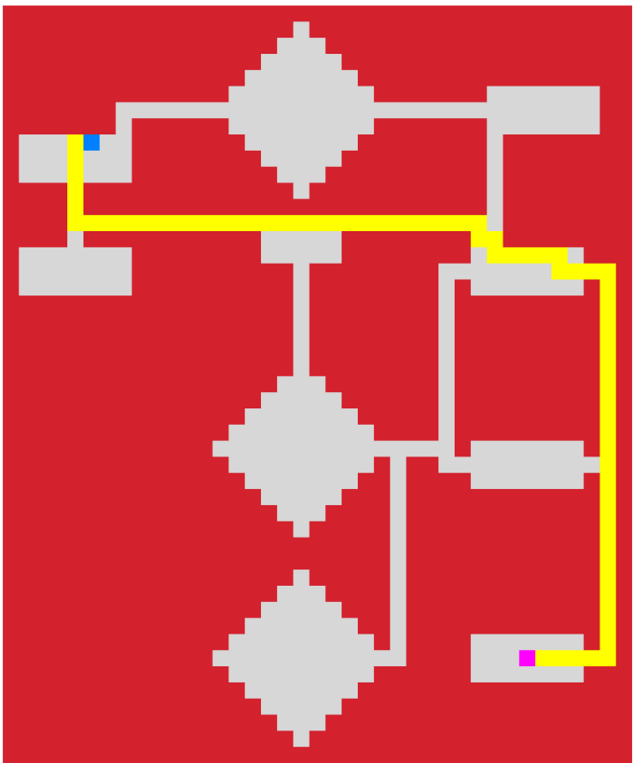
When you arrive at Strossa, your friend is nowhere to be found. Applying your intellect, you deduce that they have probably moved on and you missed them in the stairs. You call your friend again and find out that they are now at Selskapssiden. Your task is now to use your A* implementation to find the shortest path from Strossa to Selskapssiden.

If we run our algorithm again, Asta1.py, code on the Windows Command Prompt, we get the following path, with a cost of 72.

```
-------  PATH    -------
[[40, 32], [40, 33], [40, 34], [40, 35], [40, 36], [40, 37], [39, 37], [38, 37], [37, 37], [36, 37],
 [35, 37], [34, 37], [33, 37], [32, 37], [31, 37], [30, 37], [29, 37], [28, 37], [27, 37], [26, 37],
 [25, 37], [24, 37], [23, 37], [22, 37], [21, 37], [20, 37], [19, 37], [18, 37], [17, 37], [16, 37],
 [16, 36], [16, 35], [16, 34], [15, 34], [15, 33], [15, 32], [15, 31], [15, 30], [14, 30], [14, 29],
 [13, 29], [13, 28], [13, 27], [13, 26], [13, 25], [13, 24], [13, 23], [13, 22], [13, 21], [13, 20],
 [13, 19], [13, 18], [13, 17], [13, 16], [13, 15], [13, 14], [13, 13], [13, 12], [13, 11], [13, 10],
 [13, 9], [13, 8], [13, 7], [13, 6], [13, 5], [13, 4], [12, 4], [11, 4], [10, 4], [9, 4], [8, 4], [8
, 5]]
------------------------

PATH COST:     72
```

As before, we can see a visual representation of the path taken, represented with the yellow colour.

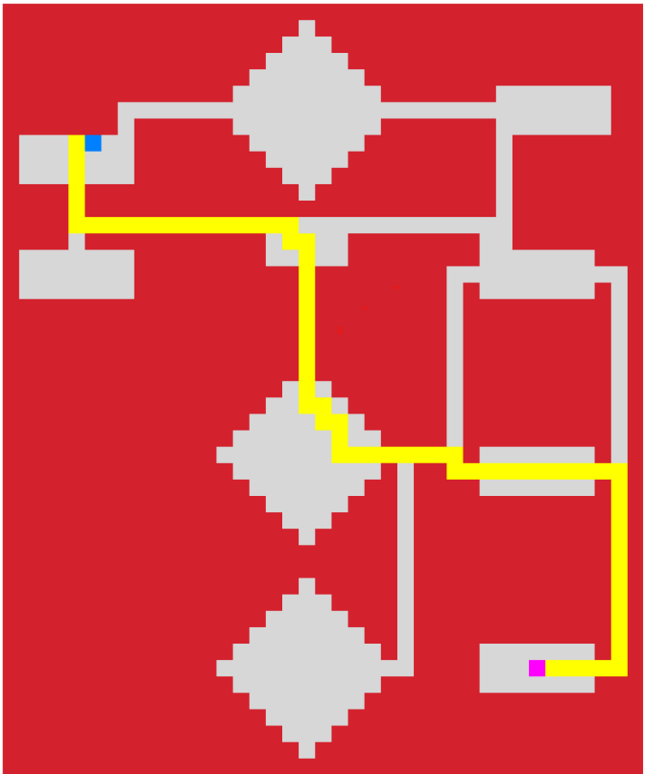


As we can see, this is one of the shortest paths to get from the starting position (pink spot) to the goal position (blue spot). So we can affirm that the algorithm works properly for this task too.

We would like to mention that we have found a different path with the same cost (72), when we tried to run our weighted cells A* algorithm in this task (*Astar2.py)*. The result can be seen in the following images.

```
-------  PATH    -------
[[40, 32], [40, 33], [40, 34], [40, 35], [40, 36], [40, 37], [39, 37], [38, 37], [37, 37], [36, 37],
 [35, 37], [34, 37], [33, 37], [32, 37], [31, 37], [30, 37], [29, 37], [28, 37], [28, 36], [28, 35],
 [28, 34], [28, 33], [28, 32], [28, 31], [28, 30], [28, 29], [28, 28], [28, 27], [27, 27], [27, 26],
 [27, 25], [27, 24], [27, 23], [27, 22], [27, 21], [27, 20], [26, 20], [25, 20], [25, 19], [24, 19],
 [24, 18], [23, 18], [22, 18], [21, 18], [20, 18], [19, 18], [18, 18], [17, 18], [16, 18], [15, 18],
 [14, 18], [14, 17], [13, 17], [13, 16], [13, 15], [13, 14], [13, 13], [13, 12], [13, 11], [13, 10],
 [13, 9], [13, 8], [13, 7], [13, 6], [13, 5], [13, 4], [12, 4], [11, 4], [10, 4], [9, 4], [8, 4], [8
, 5]]
-------------------------

PATH COST:     72
```



With this results, we can conclude that our A* algorithm with weighted cells, also works in this unweighted cells map, resulting in a better general approach for pathfinding since it gives us the best solutions in both environments, and it's the code that we'll use in the next part of the assignment.

Part 2 - Grids with different cell costs

Table 1: Cell types and their associated costs.

	Char.	Description	Cost
	.	Flat Ground	1
	,	Stairs	2
	:	Packed Stairs	3
	;	Packed Room	4

Task 3

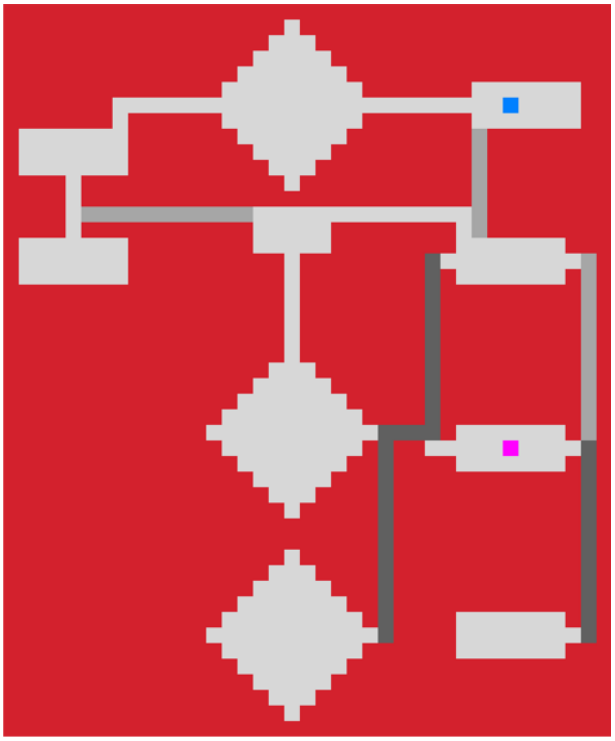
Tonight you are going to a concert at Samfundet. The concert is held at Klubben and will start at 21. You arrived early to enjoy a Lyche-Burger with some friends before going to the concert. The time is 20:45 and you should get going. The stairs from Rundhallen to Edgar have become packed with all the concert goers arriving. Use your A* implementation to find the path from Lyche to Klubben with the least cost.

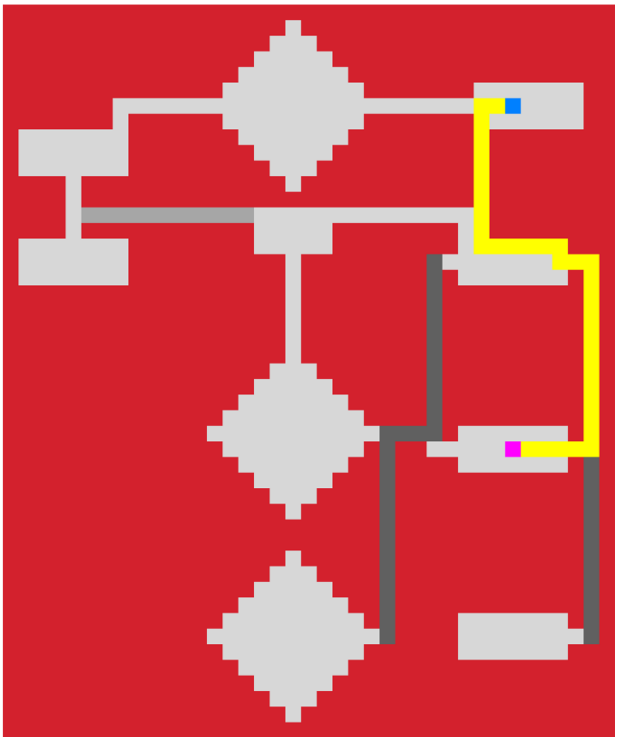
If we run our algorithm code, Astar2.py, on the Windows Command Prompt, we get the following path, with a cost of 37.

```
-------  PATH    -------
[[28, 32], [28, 33], [28, 34], [28, 35], [28, 36], [28, 37], [27, 37], [26, 37], [25, 37], [24, 37],
 [23, 37], [22, 37], [21, 37], [20, 37], [19, 37], [18, 37], [17, 37], [16, 37], [16, 36], [16, 35],
 [15, 35], [15, 34], [15, 33], [15, 32], [15, 31], [15, 30], [14, 30], [13, 30], [12, 30], [11, 30],
 [10, 30], [9, 30], [8, 30], [7, 30], [6, 30], [6, 31], [6, 32]]
------------------------

PATH COST:     37
```

On the following images, there is a visual representation of the path taken, represented with the yellow colour, and we can observe that our algorithm has chosen the right-stairs, instead of the left-starts, which are packed, resulting in a less costly path.





Task 4

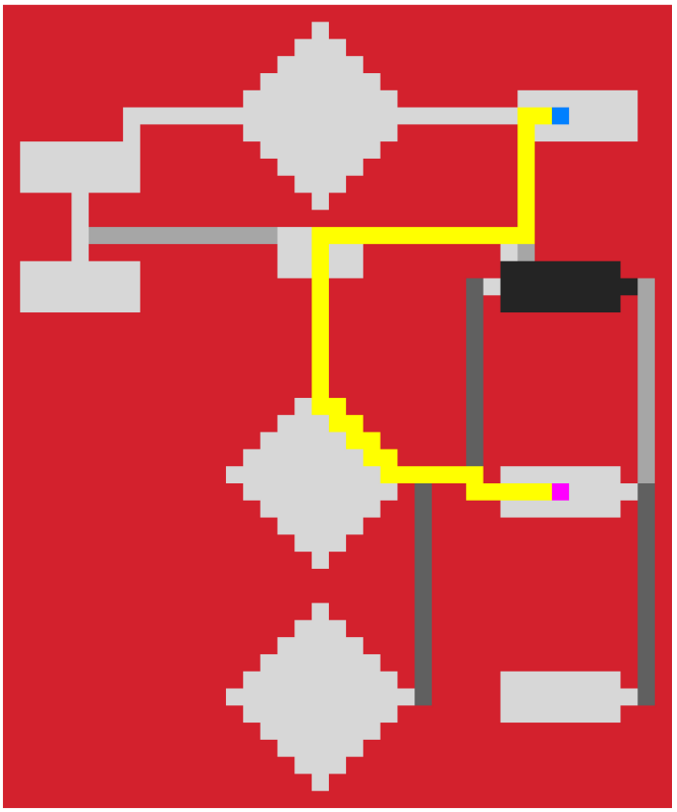
As you start walking you remember seeing a poster announcing a free chocolate cake party at Edgar this very evening. Edgar is therefore filled with hungry students scrambling to eat as much cake as possible. Use your A* implementation to find the new least-cost path from Lyche to Klubben, now considering the cake party at Edgar.

If we run our algorithm code on the Windows cmd, we get the following path, with a cost of 51.

```
-------  PATH    -------
[[28, 32], [28, 31], [28, 30], [28, 29], [28, 28], [28, 27], [27, 27], [27, 26], [27, 25], [27, 24],
 [27, 23], [27, 22], [26, 22], [26, 21], [25, 21], [25, 20], [24, 20], [24, 19], [23, 19], [23, 18],
 [22, 18], [21, 18], [20, 18], [19, 18], [18, 18], [17, 18], [16, 18], [15, 18], [14, 18], [13, 18],
 [13, 19], [13, 20], [13, 21], [13, 22], [13, 23], [13, 24], [13, 25], [13, 26], [13, 27], [13, 28],
 [13, 29], [13, 30], [12, 30], [11, 30], [10, 30], [9, 30], [8, 30], [7, 30], [6, 30], [6, 31], [6,
32]]
------------------------

PATH COST:     51
```

On the following image there is a visual representation of the path taken, represented with the yellow colour.



As we can see, we take a different path and avoid the cake party at Edgar. We can see that this is the least cost solution for the conditions that are given.