CSC3206 Artificial Intelligence A1 Assignment 1

Group: Pleb Bois

Student Name	Student ID
Chang Jin Hung	18034033
Yap Tzec Tom	18031161
Marcus Ho Kai Shuen",	18022293
Ho Hui Sien	15071459

Introduction

Efficient pathing is time consuming done by humans, yet takes mere seconds for computers to compute. This paper shall demonstrate the methods used by machines to efficiently path to a goal.

Finding a path to the snack can be as simple as traversing through every single location in the game. However, that would be inefficient in terms of time and would prove to be a lack of use of processing power. Thus, the pathing algorithms demonstrated in this paper are based on Breadth-first Search for the uninformed search and the Greedy Best-first ~~Search for~~ the informed search. The former takes the shallowest path while the latter takes the shortest path always. Both of which are more efficient than traversing through every single location in the maze.

Parameters accounted for

Accounting for the problem state given, we had to account for the location of the snake, the location of the snack acting as the goal, and the borders of the maze in our algorithm. The table below shows the purpose each of these factors plays in our algorithm.

Location of snake	Act as starting point and locations to avoid
Location of the snack	Act as the goal and indicator for a new solution to be generated
Borders of the maze	Act as the boundaries and locations to avoid

The algorithms take into account these three parameters and will generate a path to the goal.

Challenge 1

General Characteristics of Algorithm

1. Initially, the starting location of the snake shall be considered as the frontier.
    a. The frontier is a collection of locations that should be explored.
    b. Exploration is used as a means of finding out where the snake can go without exiting the boundaries of the maze.
2. The explored location shall be recorded down as the children.
    Each child will be embedded with their parents.
3. However to prevent loopy or redundant paths, each time a location is explored, the algorithms shall make sure that it is not among the frontier and have not been explored upon before.
4. After that, the explored frontier shall be removed while the newly explored children shall be labelled as the new frontier.
5. Among the children in the new frontier, one shall be chosen to be explored upon.
6. The process continues until the snack location has been explored upon.
7. The solution is generated by tracing from the goal back to the starting location of the snake through information embedded which is the parent.
8. A list of directions (n,s,w,e) is obtained through the previous step and is given to the game for the snake to path to the goal.
9. A search tree is generated as well and given to the game application.

Differences

The difference between the uninformed and informed search algorithm differs in terms of the way the frontier is chosen at Step 5.

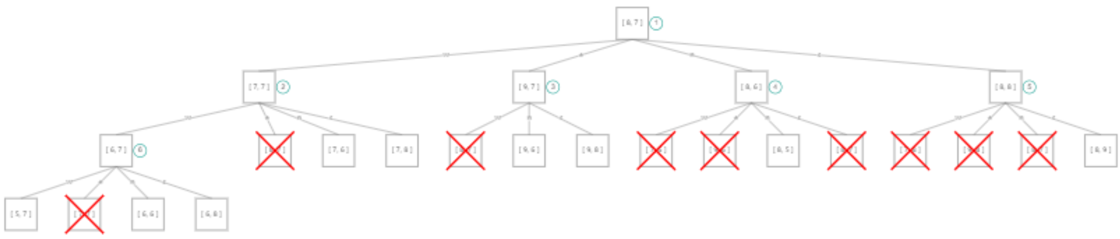
The uninformed search algorithm that uses Breadth-first search shall always choose the first index in the frontier to be explored upon while the informed search algorithm that uses

Greedy best-first search always chooses the location in the frontier with the lowest cost to the goal. The latter is accomplished by embedding the distance to the goal whenever a child is found in Step 2.

Achievability of Goal
The algorithms implemented for Challenge 1 are all successful and could be run without error.

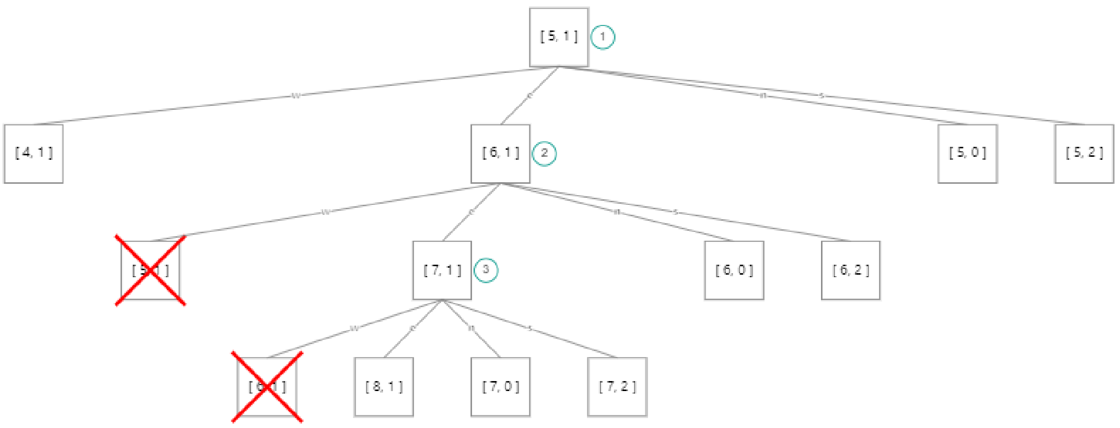
Analysis of Solution

Step 7



Search tree generated by Breadth-first search algorithm when distance to goal is only 3 blocks apart.

Step 9



Search tree generated by Greedy best-first search algorithm when distance to goal is only 3 blocks apart.

As can be seen from the solution, both of the algorithm's solutions will go all the way in a certain direction before changing it. In other words, once a snake previously going north, goes east currently, it will never go back north in the future. Not only that, Breadth-first search will always explore more locations compared to Greedy best-first search and will take longer to path to the goal. This is because the former has no guide to the goal while the latter has and can intelligently pick the best frontier to explore.

— this is due to the location of the food

For the application of both the informed and uninformed search, it can be observed that the solution will be the same in either case. This is because the shallowest path is coincidentally also the shortest path. This occurs as there are no obstacles between the snake and the snack as well as the layout of the maze.

Challenge 2
Changes from Challenge 1
To account for the dynamic size of the snake, the algorithm is modified so that any explored children shall be checked that they are not among the locations that the snake occupies at Step 3. This avoids the snake from colliding into each other. However, the limitation of this approach is that when backed up into a corner or when the snake is too long, a solution can no longer be generated and would output an error. The error would only occur once the length of the snake exceeds either the minimum value of the width or length of the maze. Not only that , the algorithms in this challenge also fail to account for the moving body of the snake. Our team did not make the snake account for not trapping itself in.

Achievability of Goal
Both of the algorithms run fine while the length of the snake is still smaller than either the minimum value of the number of rolls and columns of the maze.

Analysis of Solution
Since the initial snake's body locations are only accounted for, the snake will avoid those locations until a new snack is generated. It is because of this, it does not take into account the tail of the snack being moved. Thus, this creates a situation where the solution may not be generated even though there is clear space for the snake to path itself. Not only that, the solution generated also lacks the ability to detect whether it will trap the snake in the next round. This could be attributed to the nature of the algorithm being used which only considers the shallowest and shortest path in the short-term. Another issue is that the snake is not necessarily taking the shortest path as it only bars entry into the initial location of the snake and lacks the ability to detect whether any of these initial locations is passable due to the movement of the snake. This causes the snake to take an even further path to get to the snack even though there is a much shorter path.

A way to solve these issue when the snake fails to generate an appropriate solution even though there is a way for the snake to untrap itself or a shorter way to the snack is that for every single step the snake takes in a certain direction, the algorithms will be forced to regenerate a new solution taking into account the new current locations of the snake and the empty locations in the maze.

Conclusion
This paper demonstrated the techniques used for effective pathing in a snake game application. While both Breadth-first search and Greedy best-first search can effectively path to the snack, both of them do not consider its movement's effect on trapping the snake in the long-term in Challenge 2. Thus, these two algorithms should only be used when a solution needs to be found immediately but the solution itself brings about no consequences for the future.