CSC3206 Artificial Intelligence

2021 Assignment 1 Report

Project Title: Informed and Uninformed search algorithm on A.I. snake game

Lecturer: Dr. Richard Wong Teck Ken

Date: 19-05-2021

Group Name: Banana

Group members:

1) Sarah Low Ren Ern 18122614
2) Low Jun Jie 19041409
3) Sean Suraj Nathan 17003807
4) Mukesh Rajah A/L Michael Rajah 19028281

The purpose of the project

In this particular objective, there is only one similar goal for the two algorithm to achieve, that is to automate the snake game to play on itself using both the informed and uninformed search. Hence, the problem which the algorithm needs to solve would be given the snake's location and food's location with its current direction, how would the algorithm using informed and uninformed search to return the direction in the form of (north, south, east and west) from which the snake is to the food. With this direction, the snake will automatically slither its way to the food. This is the state of goal of our project. Besides, the algorithm would also need to produce a search tree for the respective search algorithm depicting how the algorithm has performed its search.

The algorithm is composed so that it receives few parameter produced necessary for the function to solve its algorithm's problem. Firstly, the snake game which we try to automate will only be possible with an agent. The agent composed of a class player where to initiate, few parameters are to pass on. The setup parameter which is the settings taken from the maze such as the maze size and the Boolean value of static snake length. The agent also has a run function where the main algorithm produced the solution from ~~for~~ the problem. To do so, this is where the second problem parameter fits into the algorithm, the run function takes in a problem parameter which are snake's locations, snake's current direction and food's locations. Each time there is a food on the maze to eat the agent calls run function and given with the problem parameter, the algorithm solves by returning the direction in such to navigate the snake to the food.

Obviously, different algorithm has its unique function to achieve goal. Below, will be explaining how the two algorithms were coded to understand the problem and retrieve its assignment.

Uninformed search algorithm

The uninformed search algorithm that we chose to solve the problem is breadth-first search algorithm. The algorithm starts off by creating a virtual dynamic map that can be changed via user command. Then, after creating this map, the algorithm is able to locate the food and the snake own position. After creating the virtual map, the algorithm creates the initial node and adds it to the search tree. Using the addition to the frontier list, the children of the initial node is added to the explored list. After removing the node, we check if the node was expended/generated previously before testing if it has reach the goal yet. If it has reached the goal, then the frontier will append to the child and add it to the frontier list.

The algorithm starts by creating a virtual maze that is manipulatable by selecting the size. Once the maze is generated, the algorithm will create the initial node and the end node as well as initialize the frontier and the explored array. This is so the algorithm can keep track of where the snake has explored. The initial node is then added into the frontier. The algorithm will then proceed to obtain the current node that is in the frontier that has the lowest f value. The current node is then popped off the frontier and added to the explored array. If the search tree has reached its goal, the algorithm will return the solution. The algorithm also has a way to prevent the snake from biting itself and ending the game. The algorithm generates children which the algorithm can loop through. If a child is on the explored array and is the explored child, the loop would end there.

Informed search algorithm

As for the informed search algorithm, we decided to go with A* search algorithm. The A* algorithm used works as such; firstly, it takes in the settings as setup dictionary; snake's locations, food's locations and current direction of the snake as problem dictionary from the frontend file. To begin, we extract the maze_size information from the setup dictionary to produce a virtual maze in the run function as a playground to input the snake location and food location in order for the algorithm to gauge the heuristic function. After, the algorithm initialises the initial node which is the first value in the "snake_locations" array from the problem dictionary and the end node which is the first value in the "food_locations" array from the same dictionary.

After retrieving all relevant and necessary information, the algorithm starts by appending the initial node to the frontier then expand on the nodes in the frontier which contains the lowest f value. If the selected lowest f value node is equivalent to the end node which is the food's location, the algorithm stops and return the solution with the search tree. The algorithm repeats again with new problem input. If end node yet to reach, the algorithm generates the children of the selected node and store the children in an array. From the children array, the algorithm loops through it to examine whether the children nodes generated exist in the explored array where explored nodes are stored. The algorithm continues to expand on the second lowest f value node if children nodes are already in the explored array, else examine whether the children nodes generated are in the frontier. If this is true, the algorithm will then compare the g values between the similar nodes from children array and frontier, if the similar node in frontier has a lower g value the algorithm continues to on next selected node. The algorithm will then append the children in the frontier if all conditions fail to meet and continues the loop until there are no nodes in the frontier.

try not to explain by just stating the code in sentences

Unfortunately, only the informed search algorithms produced were able to achieve its designated goal. Though it is not perfected yet, but it is definitely sufficient for the purpose of this project. The agent was able to play the snake game artificially at the very least. With that, we can confidently conclude that the algorithm has achieved the aim of solving the problem. To summarize, the performance of the algorithms for the foundation has no error in the sense where the game never ends. This is achievable in terms of the snake length will not increase and only one food at any time is produced. On top of that, the performance has reached yet another heights. All in all, the algorithm was able to achieve the minimum requirement request where the snake length will increase after eating a food, only one food is produced at any time and minimum 10 food is eaten. We would like to deeply express our sincerely apology due to miscommunication and some complication, the uninformed algorithm was not expected. Though the algorithm was not achieved but we fully understood how it would solve the problem, just not enough knowledge.

Just saying cannot achieve is not analysis.