Final Report

Team Members	Xinan Zhou	Hsi-Min Chou	Kevin Ferrer
Student ID	862206845	862254022	861272244

Summary:

Q1-Q4:
In Week 4, we learned how to implement DFS, BFS, Uniform cost search, and A*. These fundamental search algorithms emphasize the impact of how the fringe affects performance. Depending on how we manage the fringe, we can change the order in which we visit nodes, resulting in different search algorithms and changes in performance (measured in time and nodes expanded). We also gained valuable hands-on experience on Python programming. What's more, PACMAN is a lot of fun.

Q5-Q6:
In Week 5, we practiced a different challenge which is finding corners. First, we have to code the search state (i.e. to describe the search state with concrete code), which needs some manual effort. Once we finished this, we could use any search algorithm to complete the search problem. However, A* needs a strong heuristic to work well. By implementing a non-trivial heuristic, we can reach the goal state quicker with A* and with less nodes expanded, than BFS. However, as seen in problem 6, figuring out how to exactly compute the heuristic value might not be straightforward. In conclusion, we can see the difference between different search algorithms through these questions, and it's really interesting.

Q7-Q8:
In Week 6, we practiced implementing the function that aims to eat all the dots and also greedily eats the closest dot. As the professor said question 7 is the most difficult problem, we were stuck at the beginning and we didn't know how to start. We tried classic heuristics like computing the euclidean distance and manhattan distance, but realized we needed something more accurate. After a period of time thinking, we finally figured it out by applying the function that is given in the file. By taking into consideration where the walls are, we can make our heuristic more accurate by calculating the actual path. Besides, we also got extra credits since the number of expanded nodes is less than 7000.

Challenges :

- For question 6 (Corners Problem Heuristic), coming up with a good heuristic was challenging at first because there are multiple goal locations in the pacman grid. The heuristic had to accurately represent the cost to reach the overall solution, instead of just a single goal location.

Accomplishments:

- For question 7 (Eating all the Dots), we implemented a solution that was optimal enough to pass the bonus problem that was labeled as “hard”.
- For question 8 (Suboptimal search), with very few lines of code, we managed to translate the “eating all dots” problem into “greedily eats the closest dot” problem, and reached a very attractive result. It looks like a genuine AI!

Most Interesting Activities:

- Implementing the fundamental search algorithms (DFS, BFS, UCS, A*) was interesting because we got to see how changing the data structure of the fringe was enough to affect the overall performance. (Search and state representation is also critical to AI)

Most Hated Activities:

- The problem “Eating all the Dots” is our most hated activity. Since it took us some time to clear up our logic in order to get a better heuristic. However, after achieving the accomplishment, we can now flexibly use this concept and solve different problems.

Team Dynamic Challenges:

- None, we’ve never had a late submission and received perfect scores for the assignments so far.

Conclusion:

This project is really interesting and also gives us the opportunity to know more about PACMAN and practical implementations. Understanding the theory behind AI is important, but isn’t sufficient enough for real world application. In order to truly apply AI principles to real world problems, we need to be able to integrate theory into practice. The importance of the PACMAN projects stems from this motivation. After learning the fundamental theory in lecture, we apply these principles into practice through our project. In addition, learning those searching algorithms with this game can let the knowledge implant more deeply in our minds.