**CS 205 Artificial Intelligence**

**PacMan Project Report**

**Group Members:**

* Ritik Kapoor
* [Pooja Patil](mailto:ppati010@ucr.edu)
* Manan Birhmmaan

**Introduction**

PacMan is a very popular game that almost everyone is aware of, implementing an AI agent to traverse through the game is a quite interesting problem statement. When we initially visited the problem site and saw the project in action it felt quite overwhelming with all the different maze sizes and different food dot patterns and ghost agents to reduce the chances of our PacMan winning. Soon we realized that instead of looking at the problem from the top, if we break it up into smaller, different problems, we can succeed in solving these smaller problems by implementing all the algorithms that we’ve been introduced to during the classes. While the search algorithms help solve the food search amd path finding problem, the adversarial problems could be used for dealing with the ghost agents.

**Challenges and Accomplishments**

The first week of the project started with a basic problem of finding the shortest path to one particular position in the grid. Though we had been familiar with the Depth-First Search and Breadth-First Search Algorithms previously, trying to implement them for this particular problem took some effort as we had to find our way through the project code in order to implement them for traversing through a grid (unlike our previous implementations where we normally worked with graphs). The UCS and A\* algorithms were entirely new and the project helped us understand how much of a difference calculating the cost so far or estimating the future costs via heuristics can make in finding the most optimal path. The questions and hints provided in the problem statement have been very insightful to push us to find the best solution. The introductions of minor challenges like the Eight puzzle search problem helped us understand how a single algorithm written with good generalization can be used to solve problems with different definitions.

The second week of the project took the complexity level up a notch. Instead of finding just one position in the grid now we had to find the shortest path to visit all four corners of the grid. This also meant we had to devise our own non-trivial heuristic which turned out to be the most fun part. We faced a little bit of setback while defining the goal state. We initially started with just checking the length of visited corners without considering the order in which they were visited. This resulted in a path with a slightly longer route than what was expected. The hints in the problem statement stating exploring the closest corner might not result in the shortest path prompted us to take the order of visited corners into consideration. The highlight of this week was to actually see the real power of A\* which wasn’t very obvious in the first week as BFS managed to match the performance of UCS and A\* in PositionSearchProblem. But in CornersProblem we could clearly see the difference. Coming up with a good heuristic which was admissible and consistent was a bit challenging but nevertheless very enlightening. Like the Professor usually mentions in the class nobody can tell us which is the best heuristic for a task which makes it a more difficult task but we believe we were able to achieve a good heuristic as it passed the sanity test and worked well for the different problems described in the project. Having to work in a team felt like a boon during this week as we were able to bounce off ideas one another for different heuristics, figure out why certain heuristics didn’t work while the others did.

The last week seemed lightweight compared to the second week. As almost the major backbone of the functionality was ready we could concentrate on devising the heuristics this week. Our most prized accomplishment was that our algorithm managed to solve the eating all dots problem by exploring less than 7000 nodes and hence gained additional points for the same. The next part of this week’s deliverable was to realize that there will be instances when A\* and a good heuristic might fail to find an optimal path and in these cases the greedy algorithms come to our rescue.

**Team Dynamics:**

With every member having a different schedule it was difficult to meet in person to collaborate for the project so we tried to figure out a schedule to meet online over zoom and discuss each other’s challenges. Ritik being really good at Algorithms was a great help to the team with the implementations of the other teammates’ shares as well. Like mentioned earlier, getting different perspectives for the heuristic values was very helpful in coming up with optimal solutions.

**Conclusion:**

The project was the perfect blend between coding and logic. It feels like our personal ‘Hello World problem’ (what we programmers normally like to refer to) into the world of AI. It helped us understand the algorithms and realize the advantage of heuristics first hand. It also showed us how generalizing the algorithm would make it more versatile. Overall it was a good learning experience.