Overall, I thought the Pacman game was fun. I implemented the whole project on my own, though a little help with debugging from my friends who graduated from CS years ago (and working in industry) for things I was really stuck on. I think I’ve accomplished a lot given I’ve not coded much in Python and only coded primarily in Java. There was a lot I learned, even just considering reading and understanding someone else’s code in a language I’m not familiar with.

In terms of working alone, there wasn’t much of a dynamic to worry on. Since the project is relatively short in length and number of classes/methods we need to implement, having a partner or more would likely slow down the progress since it is hard for 2 people to work on a single method. However, what would have helped is to pick someone else’s brain at the same time while trying to solve some of the problems, like coming up with a heuristic.

Probably the one that stumped me the most was finding all corners, Question 5. Although it doesn’t use a heuristic yet, I was surprisingly stuck on trying to figure out how to get Pacman to explore new areas and revisit positions he’s been to before. Many times, he would get stuck in a position. It didn’t occur to me to change the state so that he would not recognize the positions as having been visited already if he already passed through it once to get to a corner. However, this is necessary since the searches takes into account visited positions, but we should refresh the positions when Pacman reaches one of the goals.

Second, heuristics were relatively challenging, though fun. Coming up with some of the heuristics, especially Question 7, took a lot of trial and error. However, it was definitely interesting since it was somewhat like a game in itself to try different heuristics and see how they improve Pacman (or make it worse!). It was also hard to figure out if my heuristic was consistent and admissible as well. Although there are checks created by Berkeley, I am a bit worried on how to accomplish this check myself in the future. I am also a little doubtful that my heuristic is truly consistent and admissible even though it passed all the tests; I probably will be a little more reassured if I studied the autograder and how it tests your heuristic for Question 7 to see if it truly captures all scenarios possible.

Probably the one I hated most was Question 5 because it was frustrating to get stuck on figuring out the logic of the problem of having Pacman revisit old positions after reaching a corner. I felt I really shouldn’t have gotten as stuck as I did, and it really made me deflated.

Overall, it was just surprising how many methods I got working when I look back at the code. Although breadth first search and depth first search is review, the rest is pretty new. It’s also surprising on applying these searches to Pacman given I’ve never really applied this to a real-life problem. That in itself is quite exciting to see what you code be put in action.