CS4408 Learning Journal 2

**Learning Journal: Reflections on Designing a Rational Agent for a Vacuum Cleaner**

**What I Did and How I Did It**

For this assignment, I had to create a rational agent that works like a vacuum cleaner in a 4x4 grid. The task was split into three main parts: defining the PEAS (Performance Measure, Environment, Actuators, and Sensors) framework, writing some pseudo-code to show how the agent would act, and finally reflecting on what I learned.

In **Part A**, I broke down the PEAS framework. This helped me figure out what the agent needed to do, where it would work, how it would move around, and how it would sense things like dirt and energy levels. I leaned on ideas from Russell & Norvig (2020) to make sure I was on the right track.

For **Part B**, I wrote Python-style pseudo-code to describe how the agent should behave. I set up a 4x4 grid to represent the environment and built functions to handle movement, dirt detection, energy management, and making sure the agent could get back home when needed. Balancing the energy use with cleaning efficiency was trickier than I thought, and I had to tweak my logic a few times to get it right.

**Part C** was all about reflecting on the process. I wrote about what I learned, how I might improve the algorithm, and what challenges I’d face if I tried to scale this up to a bigger or more complicated environment.

**My Reactions to the Assignment**

At first, I thought this would be pretty straightforward. I mean, how complicated could making a virtual vacuum cleaner be? But once I started working on the pseudo-code, I realized there was a lot more to it. Defining the PEAS framework was simple enough, but getting the agent to act efficiently took more effort than I expected.

One thing I really liked was figuring out how to manage the agent's energy. It added a fun challenge because I had to make sure the agent didn’t run out of energy before finishing its job. It felt satisfying when I finally got the logic working so the agent could clean efficiently and still have enough energy to get back home.

**Feedback and Interactions**

While working on this, I joined some discussion forums to see how my classmates were approaching the problem. One of them pointed out that it’s important to check if the agent has enough energy before moving too far from home. That tip was super helpful because I hadn’t thought about that initially.

Another classmate mentioned the A\* algorithm for better pathfinding. I didn’t end up using it for this assignment, but it got me thinking about how I could improve my agent in the future, especially if I had to deal with bigger grids or more complex environments.

**Feelings and Attitudes**

Overall, I found the assignment pretty engaging, but it definitely had its frustrating moments. I felt pretty good when I nailed down the PEAS framework, but debugging the pseudo-code when things didn’t go as planned tested my patience. There were times when my agent kept running out of energy too soon, and it took a lot of trial and error to fix that.

Still, once everything came together, I felt really proud. It was cool to see how AI concepts we’ve been learning about in theory actually worked in a practical setting. This assignment gave me a new appreciation for the complexities involved in even simple AI tasks like making a smart vacuum cleaner.

**What I Learned**

This assignment taught me a lot about how rational agents work. I realized how important it is to plan out not just what the agent should do in the moment, but how it should manage resources like energy over time.

One big thing I learned was how crucial **energy management** is. By adding checks to make sure the agent had enough energy to return home before moving too far, I made the system a lot more reliable. This mirrors real-world scenarios where robots or drones have to manage limited power (LaValle, 2006).

I also got a better understanding of **scalability**. While a 4x4 grid is easy to handle, bigger environments would need more advanced algorithms like A\* or Dijkstra’s to find the best paths (Thrun et al., 2005). Plus, in the real world, obstacles like furniture or people make things more complicated, so the agent would need extra sensors and smarter algorithms to handle that.

**Challenges Faced and Solutions**

One of the hardest parts was making sure the agent didn’t run out of energy halfway through cleaning. My first version had the agent moving in a simple right-and-down pattern, but that wasn’t very efficient. It kept running out of energy before finishing the job. To fix this, I added a **Manhattan distance calculation** to check if the agent could still make it back home after each move. This change made a big difference.

Another issue was handling the dirt bag capacity. I forgot to add a check for when the bag was full, so the agent kept trying to clean even though it couldn’t collect more dirt. Once I added a **bag capacity sensor** and made the agent return home to empty the bag when needed, it worked much better.

These problems were annoying at first, but solving them felt really rewarding. It showed me how important it is to think through all the little details when designing an intelligent system.

**Conclusion**

This assignment was a great way to apply what I’ve been learning about AI. Designing the vacuum cleaner agent helped me understand energy management, efficient decision-making, and how to handle scaling up a system. Talking with classmates gave me new ideas and perspectives, and overcoming the challenges made me more confident in my problem-solving skills.

Overall, this project showed me how even simple AI tasks require careful planning and attention to detail. I’m excited to apply these lessons to more complex problems in the future.

**References**

LaValle, S. M. (2006). *Planning algorithms*. Cambridge University Press.

Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. MIT Press.