**Computational Issues**

As mentioned above, there are certain practices that are adopted so that we limit the burden of computational overhead when trying to infer things about the maze. Like the professor said in the announcement, new data / observations are not necessarily important for all over the board. In fact, most times, those new pieces of information are only important to the immediate neighborhood (possibly the neighbors of neighbors, but that’s specifically in Agent 4’s territory). With this in mind, we purposefully designed the inference agent to operate on a “ripple” principle. To clarify, when the agent gets a new piece of information, we check around us to see if anything is affected (certainly observations will be), and if another cell is able to be confirmed either blocked or empty, we can begin a ripple effect out from there as well. With this approach, we don’t have to look at every cell on the board, which can be quite a large number, and it gives us some flexibility as to how far we infer, based on the momentum of the information that’s being uncovered (it can propagate out really wide if that was a key piece of information almost like a large rock or it can stop at a really short radius like a small gnat landing on the water).

This also led to a key choice in the design of Agent 4. As mentioned in the Agent 4 section, we didn’t devote the time and resources to trying to find the potentially perfect synthesis of equations that may lead us to infer a random spot on the board. This would be very cumbersome with a very limited payoff on average. For example, another principle we observed when testing and refining the project is that the information we could determine in the maze spread out behind us on our previous path outward like a small wake off the stern of a ship moving through water (apologies for all of my amphibious analogies). It’s held in by the lack of sensing that we’re able to do outside of our path. With this in mind, we can see that trying to determine information for what’s to come (the cells unexplored) is an uphill battle. Much like in the game Minesweeper, it is often difficult to find a breach point in the nebulous cloud of unconfirmed cells, and it takes a lot of brain / computing power to figure out (and even then, sometimes you have to guess). This is why we chose to simply try to focus on seeing if we could determine if our next step in the path is safe or not. This directed us towards a goal, much like only inferring for the immediate neighborhood, and we get that added benefit that the other neighbors may be inferred because of their proximity to the next cell in the path. The drawback is that we aren’t inferring everything there is to be inferred, and this could potentially lead us to have information that’s at our fingertips slip through the cracks. However, we made a judgment call that it would be better to not compromise speed for a potentially slight improvement in performance from what we’re already experiencing.