Alex Arnold

AP Computer Science

Game of Life Lab Reflection

I really liked this this lab because it was fun designing new patterns and creating code that allowed my patterns to evolve. It was very rewarding to see my code work and my pattern change right in front of me. The only thing I *disliked* about the lab was how tedious it was to create all the different coordinates, actors, and test statements for each cell that is initially alive in the first generation. After creating the coordinates and actors for 26 different starting cells, I was a bit tired with the lab, but thankfully there wasn’t much left to do. I was initially surprised that all we were doing was essentially creating the *createNextGeneration* method. That was also unexpected in itself because I underestimated how challenging it would be to successfully create a working method to figure out the next generation.

I learned a lot in this lab, especially many things about gridworld, but the most important thing I learned was the necessity of pseudo code. Pseudo coding made creating the actual method MUCH easier, as I just had to translate what I was explaining and writing down in everyday language into Java syntax. Also, drawing the grids by hand really helped cement the rules (for which cells survive and die) into my head, making it much easier to create the pseudo code. However, there are still some things I was never able to grasp. I wanted to give the user a choice for what image they would like to be used for the cell, but couldn’t design the code in time. My main concern was that I could only designate one image per actor class, so based off my initial theories I would have had to create multiple classes (one for each image) or possible cycle through an array of images, similar to what I did in my Cityscape Lab. I also would have liked to figure out how to make the grid expand *with* my cells so that they would never be cut off.

To next year’s students, I strongly advise them to design your grids by hand and pseudo code your method BEFORE even considering attempting to program it. Thinking about how the method should work before having to worry about how to say it correctly in Java makes the job *incredibly* easier, both mentally and time-wise. Once you figure it out using pseudo code, you simply have to translate it to the proper syntax and *boom* – you’re done.