I liked the game of life lab primarily because of what could be done with the starting patterns and seeing which patterns would cause continuous motion (for example, patterns like gliders and pulsars). I think it’s really cool and interesting, and I’m curious to know what the math that goes behind this is. As for what I liked and disliked, I enjoyed creating the nextgen algorithm and using my understanding of for loops to make the code as efficient as possible. I disliked making the test class—it was really tedious to make each individual point in the test class by entering a y and x value for each location. Since I coded my program to take user input, every time I wanted to test my code, I would have to enter the locations too (but I got pretty good at this). This lab, though still challenging, was not quite as difficult as I originally thought it was going to be after I browsed the documentation of GridWorld. The methods that were already defined were really convenient in helping me find information that I needed, such as “getOccupiedAdjacentLocations()”. In this lab, I learned how to implement classes that I had not defined myself using documentation. For example, I didn’t know what an ArrayList was (the type returned by getOccupiedAdjacentLocations()” but when I looked at the java API I could find the method that I needed to invoke for my purposes, which was size(). And even though I initially didn’t understand how the GridWorld framework was put together (the relationships between Grids, Actors, and Worlds), I quickly became familiar with their useful methods through looking at documentation. My advice to the students next year is to thoroughly study the documentation before they begin to implement any code. It gives you an idea of the “tools” you have in your “toolkit”, and you will be able to find efficient ways to write code when you have useful methods at hand.