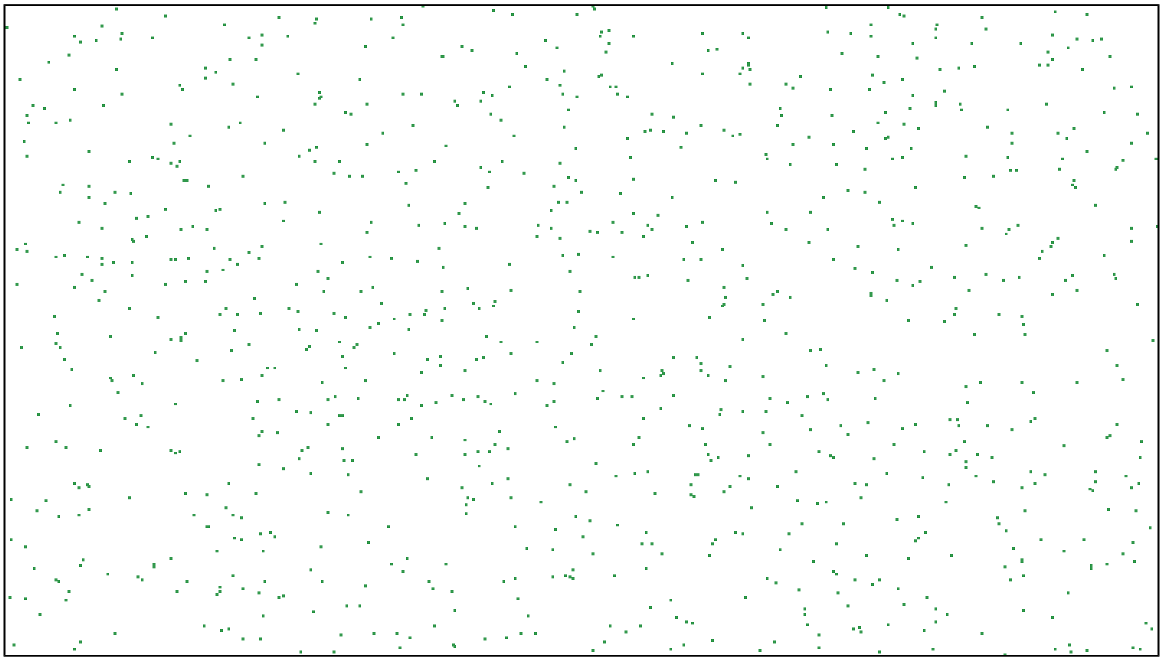
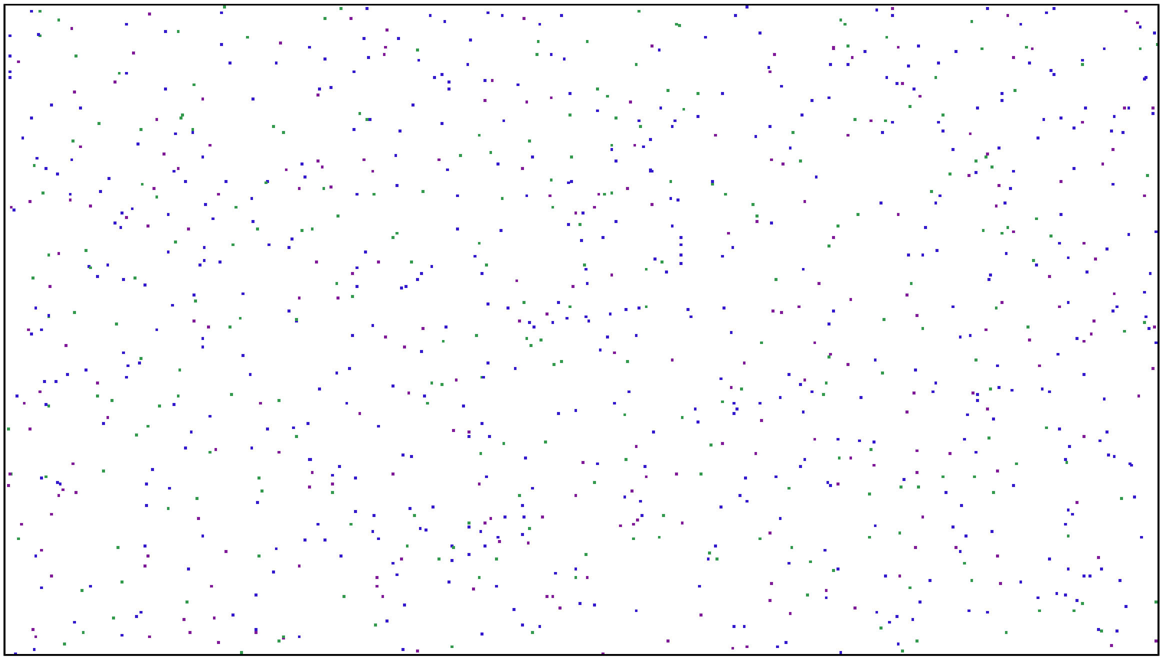
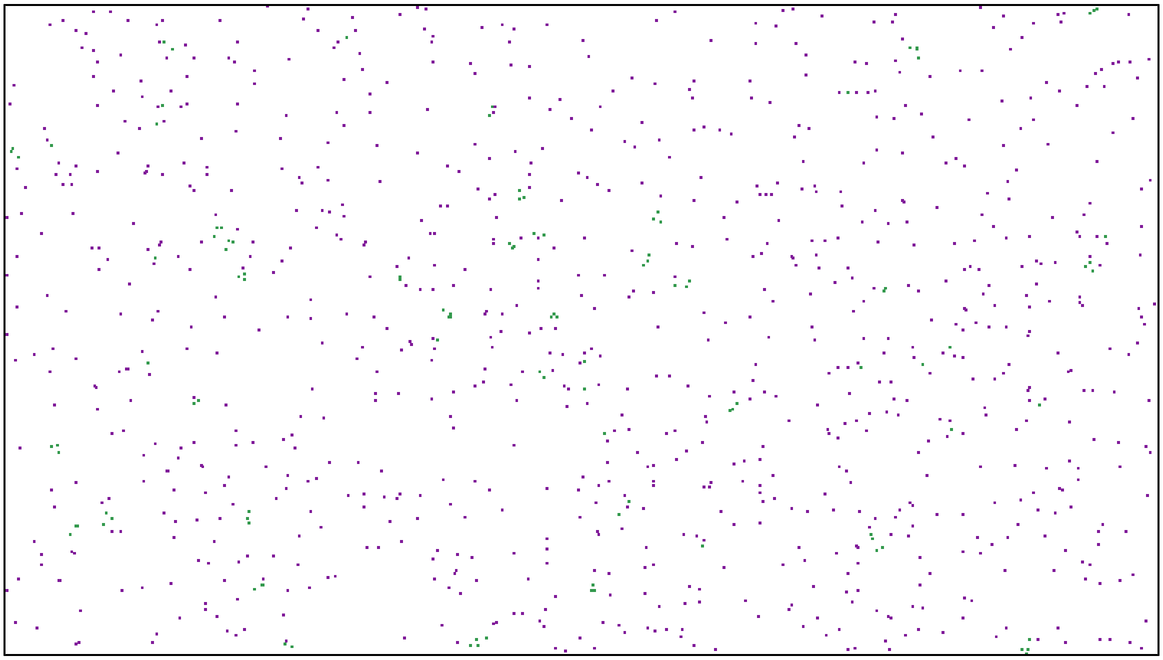
project 4

the purpose of this lab, as the title states, was to learn about abstract classes. An abstract class is a helpful thing. It allows you to make a few similar classes, that might share the same methods, by having each one inherit the shared meathods from a parent class rather than writing them anew each time. Thus for this project we were able to make different types of cells, ie points on a grid which interact with eachother, without too much effort.

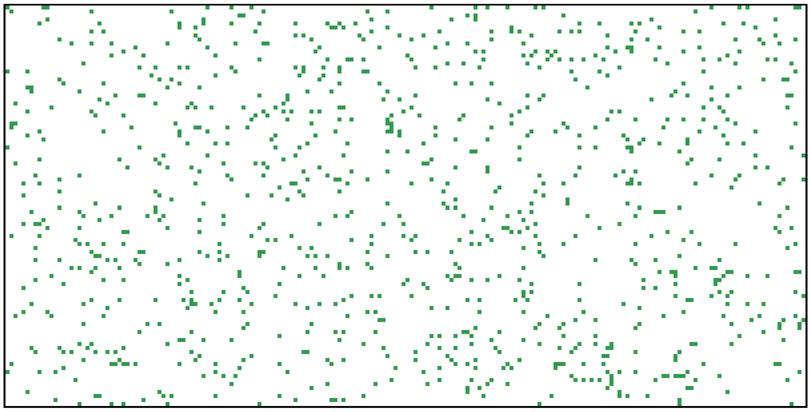
We made cells that are attracted to each other, ones that generate in 3 different colors and are only attracted to their own kind, and a cool one called lifecell which regenerates under certain conditions, creating a simulation that looks like growing mold or somesuch. It was all pretty algorithmic and alright.







for an extension I made a conway cell. I had to tell this cell to update based on the rules of the game of life, but the part of adding new cells based on the arrangement was tricky. I had to make a new landscape, loop thru the spaces create a temporary new cell at each, check its neighbors off the old landscape, decide if the new cell is a true offspring, then add the cell to the new landscape for real. Then I applied the rules of the game of life, reading the original landscape but only writing to the new. As usual, the algorithm was relatively simple but the debugging took a lot of time. I had a lot of trouble with the fact that a conway game takes integers for the cell's location, so I had to eventually overload the abstract cell constructor with another one for ints.



This project taught me a lot about how to use some helpful java tools, like inheritance, constructor overloading , typecasting, and general object stuff. I'm not sure if I could have done my conway cell better, as I had to make a different update method in landscape to handle the cloning of the landscape and other parts of the conway update. I'm pretty sure I couldnt put it all the a cell class though, because each cell can't make its own landscape or things would get pretty hectic.

Olivia Lang helped me with some debugging, and importantly showed me that if one class doesn't compile, then the ones following it alphabetically won't either! that was a frustrating one.