”Pair of L’s” Project Update

November 15, 2021

Zacree Carroll, Troy Clendenen, Raul Patel

1 Overall Status of LL

While our team is not as far as we would have liked to have been by this point we do have a working version of Cellular Automata using the Game of Life rule set. At this point with game of life nearly functioning properly (there is one small bug that needs finding/fixing) we now have a framework which will allow us to run some rule sets which have slightly more complicated rules than Game of Life. We would like to have at the very least one more rule set implemented but are hoping for more because the more rule sets we have the more we can compare data between them and run different types of tests using them. We will be setting our sights next on the next rule set as well as parallelizing what we already have. Once those two things are finished we will begin the testing phase where we analyze our results and find ways to optimize our existing code base. If that all runs according to plan then we should have time for one more rule set.

2 Streamling API

The API and architecture of our overall software is in a decent state. Without spending too much on it in fears of not getting enough parallel programming practice in, we’ve defined quite a few universal functions that are useful in general cellular automata simulations and not just for the game of life. The biggest roadblock however, is being able to define generic types for the Grid class and the Cell class. We’d like to define multiple cell classes to be used by a universal Grid class so it would be easier to design future simulations, however we’re still figuring out how to utilize C++ template classes and polymorphism properly in order to do this.

Regarding the state of how to run our program, we’ve delegated the instantiation of a simulation grid to reading from a file that has a list of x y coordinates in order to determine what the starting state of the simulation will be. That’s all the starting info we’ve needed so far, however as we expand this to a more broad group of simulation spaces, we may need to rethink this. However, we don’t expect to get to that point during this class.

3 Graphics Rendering

As far as the graphics rendering for this project goes, we currently are saving the state of the grid to a ”.out” file for access by a separate file that will use

OpenGL to create a window with a nicer visualization of the data. This will be done by translating the hyphens (representing dead cells) and zeroes (representing living cells) to the more traditional representations of the cells in simulations. This method will read from the file and update the display until the window is closed or it receives a signal from main that the simulation has ended.

The output file is currently updating correctly to the current state of the simulation and the graphical OpenGL implementation is underway and will be added as soon as it is working correctly.