Giovannie Alteri, Dylan Do, Yara Khoury
Chem 274B: Introduction to Software Engineering
Final Project - Design and Implementation of A General Purpose Library for Basic Cellular Automata Modeling

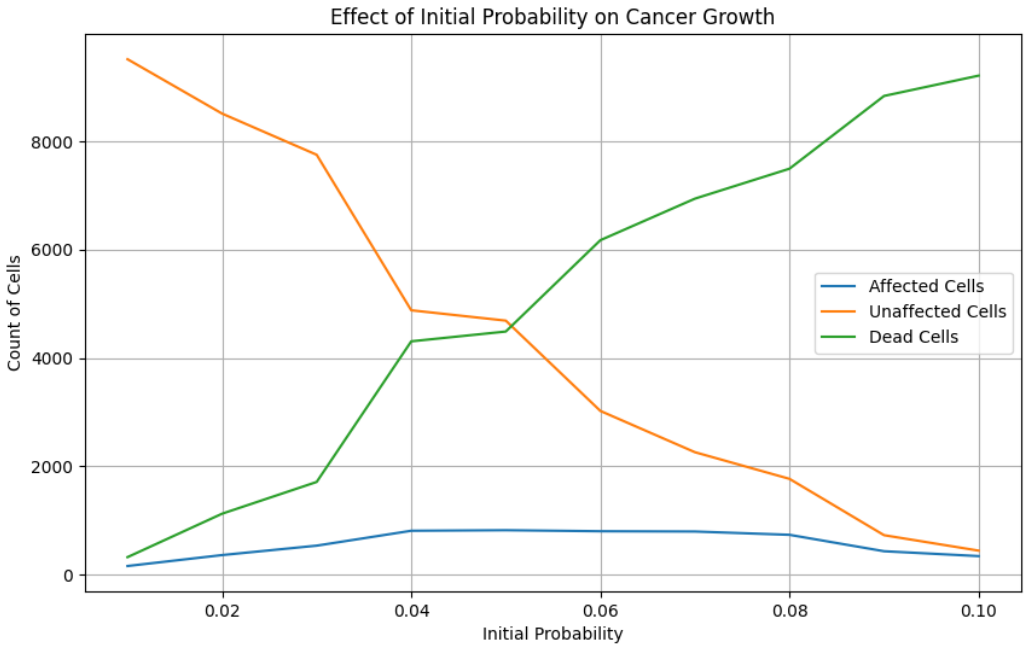
Brief Summary of the CA Application, CA Modeling and Simulation
Cancer Growth CA Application

The application we decided to create is how cancer cells will act/spread in a certain area(i.e body area). There is no 'set' rate at which all cancer grows, some tend to grow slowly, fast, or not much growth at all. This is because there are many types of cancer cells (benign and malignant) and varying environmental factors may cause cancer cells to grow or not grow at all. Our application will explore this phenomena with our Cellular Automata application. In this model we will use our general purpose library to simulate the cancer spread. We will study the effects of the cell states and the cancer spread. The cancer tumor will be initialized in the cellular automata grid which will have 3 states: healthy cells, affected cells from the tumor, and the cells that will die from the spread. To test our application we will create different simulations under different probabilities to compare. We will then plot these simulations, including the number of healthy/affected/dead cells after a certain amount of steps.

This application will use our general purpose library found in Include/myCellularAutomata.h. The actual implementation is found in Source and our Application will use these in our Application directory. The plotting for the CA can be found in the Utils directory. Our application will study the relationship between the probability of a cell to develop cancer in a certain system vs the size of the system itself. Given a certain starting probability P that a cell can become cancerous, and given a certain area of the system (dimensions of a matrix of rows * columns), what is the relative final number of cancer cells (cancer cells/total number of cells) that we get after a number of steps? We will also observe after how many steps the entire system is considered 'dead'.

For our simulation, it allows us to explore with different probabilities (ranging from 0.001 to 0.99) and differing matrix sizes (from 10 rows/columns to 100 rows/columns). We decided in our CA to use the von Neumann neighborhood, majority rule, and infinite boundaries in order to simulate a real life cancer growth.

(Pictures of simulations here)
(Example Output services)



From these observations, we can see that as the probability rises, the spread of the tumor has drastically increased, which was to be expected. The number of affected cells turning into dead cells increases as the initial probability rises. In scientific terms, if the cancer growth was malignant and were to spread drastically, it would simulate the higher probability of the growth we implemented.

PART III. Problem 3 [LO5, LO6 and LO7] - 15 pts [Individual work] Individual student software engineering reflection on their final project. For this part, every student must submit an individual PDF and place this file in the root of the student's individual repository. The student reflection must cover the following points:

3.1. A description of their role in the project.

I was in charge of implementing the test code and application. Yara had the responsibility of outputting/displaying the current CA configuration by writing the relevant code and python script for it. Dylan helped both for the library and the application. In addition, all of the team members shared the responsibility of updating the repository documentation, fixing code, and debugging for any errors. We all had the responsibility of updating the documentation and libraries.

3.2. How much do they contribute to the successful project completion (e.g. lead work, help others, coordinated meetings, etc),
I was able to help my team members when they needed it. Everyone had an equal part to contribute to the project. I led the weekly zoom meetings held to account for project progression. Yara and Dylan were able to help debug and fix any code.

3.3. Describe any challenges/problems you and/or your team dealt with and how you solved them.
My greatest challenge was the organization of the time for the meetings and the weekly goals we were pursuing as a team.

3.4. Comment on the algorithmic and performance analysis of your library (you can base this on the data structures and functions you included). You do not need to provide a detail of every function that you implemented but rather provide a high level description of the design considerations that make you choose the data structures and algorithmic implementations that you used in this project. In particular, the parts of the project that you lead.

Our Class is based on a matrix of integers, composed of 0, 1, and 2 to codify the different cellular states. The setup_cell_states(), straight conditional rule(), setup_dimension(), setup_neighborhood(), change_state_cell(), and conditional transition rule functions have a time complexity of O(1), because they simply have to read the matrix. The function for setting up bound types and the initial conditions does have a time complexity of $O(n^2)$ because it iterates through the matrix grid. The majority rule has a time complexity of O(S), where S is the size of the neighborhood.

3.5. Comment on how effective your showcase CA application from Problem 2 worked (i.e., how realistic was the model, any suggestions on how to improve its accuracy, etc)
We were able to show the cancer growth with different probabilities of the spread and different areas of the system. The model was realistic and coherent. showed realistically in real time how the affected cells were able to spread out or not. We could improve its accuracy by introducing a third dimension.

3.6. What can you have done differently:
3.6.1. Library development:
I am satisfied with the work done, I don't think we could have done better.

3.6.2. Software project management

Having enough time, it could be interesting to try to add parallelization to the library, and include other libraries that can help include other biochemical mechanisms in the system simulated.

3.6.3. Product improvements

We could add more languages to make it more inclusive.