Assignment 1

Adam Luqman Hakim Bin Mohamad 40021709

COMP 426
Concordia University
9/10/2020

Techniques Used & Architecture

In my task, the 2D region of 1024 x 768 cells is represented by a global array. The reason I used an array is that the problem size was already known. This implies that it would never be appropriate to resize. All I will have to do is change the values of particular cells, which is simple to do with an array.

Each time the update function is called, four threads are generated to handle the status changes of the cells. One quadrant of the total cell area (as seen in figure 1) is handled by each of the four threads. In order to ensure that all threads are able to complete their computations before proceeding, the threads are then joined.

Thread 1	Thread 2
Thread 3	Thread 4

Figure 1

Finally, in the "MedicineCellSurroundingHeal" feature, recursion is used. When a cancer cell is healed, this purpose is used to heal all the surrounding drug cells. The surrounding cells are all tested to see if they are medicine cells, starting from the cancer cell that is being healed. If they are, then they are converted into healthy cells and studied by their surrounding cells. I can easily guarantee that all the surrounding medicine cells can be healed by recursion when a cancer cell is healed.

Most Important Part of Assignment

From my point of view, the use of multiple threads on a common core to try to maximise the productivity of the programme. It can be discovered after completing the task that using several threads on a single core does not always produce an performance improvement. To provide an example, My lenovo computer multi core threads has the ability to run up to two threads per core in parallel. Thus, using multiple threads on a single core on these machines can achieve some degree of performance increase. However, the threads would have to be executed sequentially while operating on a single-core or multicore computer that does not have the capacity to run concurrent threads on a single core. Thus, having multiple threads running on a single core on these machines would not result in any performance improvement. To conclude, it is not possible to have four threads running in parallel on the machines that I have readily available for testing while using only a single core.

Most Difficult Part of Assignment

For me, using OpenGL was the most challenging aspect of the task. I had absolutely no experience with OpenGL. So, I had to learn everything from scratch. However, OpenGL is a very well known Application Programming Interface (API), so finding resources online to solve all the problems I faced was very easy. I was glad, in the end, to have had this chance to play with OpenGL. It is an API that is very strong and has quite a few interesting features.

Test Scenarios

Test Scenario #1: Directly healing a single cancer cell

User Input: Click on a red cancer cell.

Expected Result: Medicine is injected into the cancer cell and is instantly absorbed. The cancer cell is healed and turns green.

Test Scenario #2:Indirectly healing a single cancer cell

User Input: Click on the cells around a single cancer cell. Be sure to click at a distance of two (2) cells away due to the radial expansion of the medicine. It should take three (3) clicks in order for the cancer cell to be sufficiently surrounded by medicine.

Expected Result: The cancer cell is cured and turns into a green healthy cell due to the high number of surrounding yellow medicine cells. The surrounding yellow medicine cells also become green healthy cells.

Test Scenario #3: Injecting medicine into a healthy cell

User Input: Click on a green healthy cell.

Expected Result: Medicine is injected into the healthy cell and the cell becomes yellow. The medicine is not absorbed and moves radially outwards by one cell position in each direction. Note that all the surrounding cells turn into yellow medicine cells regardless of their previous state.

Test Scenario #4: Injecting medicine into a healthy cell at the edge of the area

User Input: Click on a green healthy cell that is at the edge of the area (either on the left, right, top, or bottom edges, or in a corner).

Expected Result: Medicine is injected into the healthy cell and the cell becomes yellow. The medicine is not absorbed and moves radially outwards by one cell position in each possible direction, being sure to not expand outside of the boundaries of the area. Note that all the affected surrounding cells turn into yellow medicine cells regardless of their previous state.

Test Scenario #5: A healthy cell turns into a cancer cell

User Input: Click on a red cancer cell that is surrounded by more than the majority (≥ 6) red cancer cells.

Expected Result: Medicine is injected into the cancer cell and is instantly absorbed. The cancer cell is temporarily healed and turns green. However, the cell quickly turns back into a red cancer cell due to the high number of surrounding red cancer cells.

Reference

[1]. Home. (2017, June 07). Retrieved October 10, 2020, from

http://www.opengl-tutorial.org/

[2].(n.d.). Retrieved October 10, 2020, from

http://www.opengl-tutorial.org/beginners-tutorials/