Comp 426 Assignment 2

ZIsen Ling, 40020293

Implement the 2D cell growth simulation from Assignment 1 using Intel Threading Building Blocks (TBB)

Check List

1. **Set Up the Development Environment**
    - ☑ ~~Install and configure Intel Threading Building Blocks (TBB).~~
2. **Implement 2D Cell Growth Simulation**
    - ☑ ~~Implement parallel computation of cell growth using TBB's parallel algorithms.~~
    - ☑ ~~Use TBB's task parallelism constructs to simulate multiple simultaneous medicine injections.~~
3. *_OpenGL_
   ? Embed the necessary OpenGL calls within the TBB program. ?*
    - I seperate the render code with logic code in to different class and different loop, and opengl use graphic card, there is no need of tbb in my point of view.
4. **Optimize the Program**
    - ☑ ~~Analyze and reduce unnecessary data synchronization and locks.~~
    - ☑ ~~Optimize data structures and access patterns to improve cache coherency.~~
    - ☑ ~~Further optimize the code based on results from performance profiling.~~

Thoughts

1. should I use concurrent vector or other thread safe data container
    - No, in assignment 1, I encountered some thread safety problem, my solution was by using two vector instead of one vector. One to store current data and it is read only, and another one is to store the data in next frame and it is write only.
    - I am pretty sure it is thread safe, and we knew concurrent data container performance less, thus no need for it.

TBB method that I used:

- parallel_for

    - I used tbb::parallel_for + tbb::blocked_range + tbb::auto_partitioner() to implement parallel_for
    - In assigment one, I partition data by myself, this time I want it to be done automatically
- task_group
    - I used tbb::task_group to implement task parallel multiple simultaneous medicine injections.
    - we know tbb is based on Task Scheduler. The jobs that parallel_for created will be recognized as task.
    - so my job is simple, just adding each medicine injection task into task_group