Concurrency Lab 3

Cody Bloemhard 6231888
Egor Dmitriev 6100120
LaTeX

February 2, 2019

1 Questions

1. We have ported the framework to do all the calculations relating to game of life on the GPU.
2. Game is stored in on **bit per cell** at the simulation.
3. Our code support both **dead edges and edge wrapping**. One can toggle between them in the Init function of GameOfLife.cs. There at the creation of the GameProcessorCL on can toggle the last argument to toggle between the edge simulation handling. True means the edge will wrap around, false means that everything beyond the edge is dead;
4. The whole board is uploaded once to the GPU. After that all the data is read or written from the GPU memory.
   If one has a GPU which support OpenGL interop, then data will never leave the GPU. The rendered area is selected on the GPU and copied into OpenGL texture.
   If GPU doesnt support OpenGL interop, then the frame is reendered on the GPU. Pulled to the CPU and then pushed into an OpenGL texture.
5. Our program supports both, zooming and panning. One can zoom by using mouse scroll wheel and move by clicking the mouse anywhere and draging it across the screen. The moving is made smooth (pixel by pixel instead of tile by tile), so the user wont be disoriented.
6. N/A
7. N/A
8. We have sped up the kernel by calculating for every 32 tiles their next state. This is stored in a private uint. This uint is then written once to the global memory. This way we avoid 32 separate writes.