

Trabalho Prático II

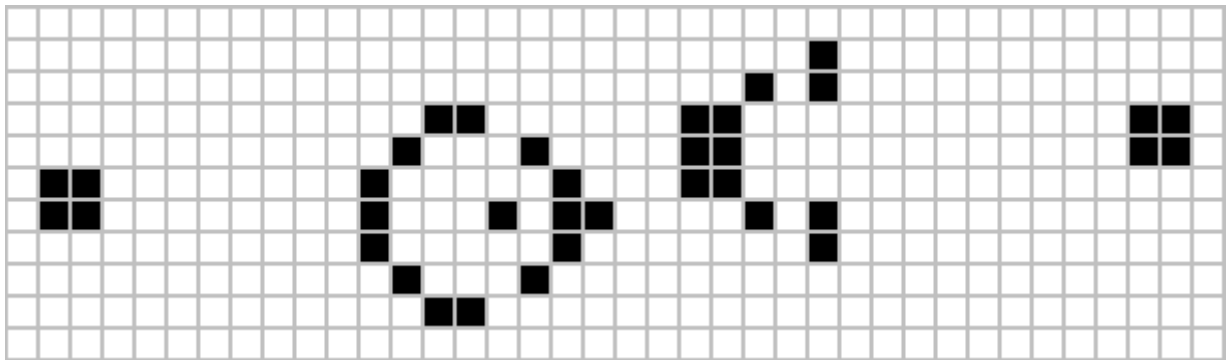
Programação Concorrente e Distribuída

Valor: 5 pts

Apesar do Game of Life também ser uma simulação de vida artificial, o principal objetivo de John Conway durante sua criação foi para simplificar um modelo matemático complexo de uma máquina hipotética criada por John von Neumann (breve entrevista com o criador Conway no YouTube [2]). Apenas utilizando as regras do Game of Life é possível criar uma Máquina Universal de Turing [1].

O Game of Life é composto por uma matriz $N \times M$ onde cada célula da matriz está no estado vivo ou morto. A cada geração (iteração) um conjunto de regras é aplicado em cada célula da matriz, definindo o estado das células para a geração seguinte [3].

- Qualquer célula viva com menos de dois vizinhos vivos morre de solidão.
- Qualquer célula viva com mais de três vizinhos vivos morre de superpopulação.
- Qualquer célula morta com exatamente três vizinhos vivos se torna uma célula viva.
- Qualquer célula viva com dois ou três vizinhos vivos continua no mesmo estado para a próxima geração.



Implementar em C++ e também em CUDA a aplicação de automato celular chamada **Game of Life** com as características a seguir:

- Deve ser possível variar o tamanho da matriz $N \times M$.
- A matriz deve ser inicializada aleatoriamente, dado uma probabilidade P de cada célula estar viva.
- Deve ser possível variar o número de gerações (iterações).
- Apenas a matriz final deve ser exibida na saída padrão.
- **Avaliar o speedup da implementação em CUDA em relação à implementação em C++.**

Referências:

[1] Rendell, P., "A Universal Turing Machine in Conway's Game of Life," in High Performance Computing and Simulation (HPCS), 2011 International Conference on , vol., no., pp.764-772, 4-8 July 2011

[2] <https://www.youtube.com/watch?v=R9Plq-D1gEk>

[3] https://pt.wikipedia.org/wiki/Jogo_da_vida