

# Concorrência e Paralelismo 2016-17 - HW 1

## *Game of Life* sequencial

**Aluno:** Gonçalo Sousa Mendes, n.º 42082, Turno p.3

Prof: João Lourenço

Dept. de Informática

FCT/NOVA

24 de Dezembro de 2016

## 1 Introdução

O objectivo deste trabalho de casa era implementar em C o *game of life* de Conway, de forma sequencial [1].

## 2 Implementação

Para iniciar o jogo é necessário indicar o tabuleiro a usar, o número de iterações (que por *default* são 20) e indicar se queremos imprimir todas as iterações, nenhuma ou só a primeira e a última. O primeiro passo é ler de um ficheiro o tabuleiro e alocar memória para este. De seguida inicia-se o jogo e calcula-se que células ficam vivas e que células ficam mortas, calculando-se o número de vizinhos que estão vivos. O essencial neste passo é garantirmos que lemos do tabuleiro certo e escrevermos num novo, evitando que se esteja a escrever e a ler num mesmo tabuleiro, o que traduzia num tabuleiro errado. Para satisfazer esta necessidade, foi criado um outro tabuleiro, *old\_broad*, de onde se lê os valores das células sem as alterações da nova iteração. No final, copia-se o tabuleiro novo para esta estrutura, que está declarada como variável global. Para cada célula, percorre-se todos os seus vizinhos, usando dois ciclos de -1 até 1 (p.e. o valor -1,-1, representa o vizinho em cima a esquerda), conseguindo-se assim aceder diretamente aos vizinhos através da posição da célula corrente, e verifica-se o estado de cada um. Caso se tenha mais do que três vizinhos vivos, pode-se interromper estes ciclos, pois já se sabe o destino da célula.

Existem células que requerem um cuidado especial, pois estão na fronteira e os vizinhos encontram-se no lado oposto do tabuleiro. Para estes casos, verifica-se primeiro qual a posição da célula e depois de que vizinho se deseja saber o estado, acendendo-se assim a posição correta no tabuleiro.

## 3 Avaliação e conclusão

A implementação usada, embora correta, apresenta um código difícil de interpretar, dado que para célula na fronteira se define um comportamento específico, resultando em várias condições *if-then-else* aninhadas. Como alternativa poderia-se construir um tabuleiro com uma 'camada' extra, a volta do tabuleiro original para guardar estes valores, embora esta opção seja mais simples no acesso às células vizinhas, pode implicar um maior tempo de construção do *old\_broad*, pois é preciso, para cada célula da fronteira copiar o valor duas vezes, uma para a sua posição correta, outra para a sua posição na fronteira. Como os dados se encontram num vetor contínuo, esta operação também não é trivial de se fazer. Para validar os tabuleiros finais foi usado um simulador online do jogo [2] e, mais tarde, confirmado com os colegas da cadeira através do fórum da cadeira.

## Referências

[1] [https://en.wikipedia.org/wiki/Conway's\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway's_Game_of_Life)

[2] <http://pmav.eu/stuff/javascript-game-of-life-v3.1.1/>