

## Sincronização

O recurso partilhado do projeto apresentado é a classe *Cell*. Este objeto representa cada posição individual de um tabuleiro de jogo representado pela classe *Board*, que funciona como um gestor dos recursos partilhados e por onde as cobras, representadas pela classe *Snake*, se irão movimentar. É criada uma instância da classe *Board* e é inicializada com um número parametrizável de células, o que irá criar um tabuleiro quadrado com um número igual de colunas e linhas.

Para se movimentar, a cobra terá que requisitar uma célula ao gestor de recursos partilhados, passando-lhe uma posição XY, que por sua vez irá chamar um método sincronizado da célula com essas coordenadas no tabuleiro. Esse método sincronizado verifica se o estado do *boolean isEmpty* da célula requisitada está a *true* ou *false*, caso esteja a *true*, o seu estado é alterado para *false* e a cobra passa a ocupar essa célula, caso esteja a *false*, a cobra fica em espera ( *wait()* ) nessa célula até que seja interrompida pelo utilizador para definir um novo destino. Este método sincronizado assegura que cada célula não tem o seu estado de *isEmpty* alterado de forma inconsistente por mais que uma *thread*.

O modo *TurboBoost* permite que uma cobra se movimente de três em três células e neste caso a cobra em questão terá que requisitar o cadeado do tabuleiro. Enquanto tem o cadeado do tabuleiro procede três vezes à sua movimentação e depois liberta-o para que a próxima cobra o possa requisitar. Este é um problema típico de escritores-leitores em que existem duas condições chave, a de não estar nenhum escritor ou leitor a aceder ao recurso partilhado e a de não estar nenhum escritor a aceder ao recurso. No caso do projeto, uma cobra no modo *TurboBoost* atua como um escritor e uma cobra normal atua como um leitor. O tabuleiro mantém uma contagem do número de cobras normais e um estado de se alguma cobra em modo *TurboBoost* lhe está a aceder. Caso haja uma cobra em *TurboBoost*, mais nenhuma cobra consegue aceder ao tabuleiro até que o cadeado deste esteja livre, caso não haja, todas as cobras em modo normal podem aceder ao tabuleiro. A cobra em modo *TurboBoost* tem que assegurar que requisita as três células sem que mais nenhuma cobra esteja a alterar o estado do tabuleiro, para assegurar a consistência dos dados dessas três células. Esta abordagem reduz o nível de concorrência do jogo quando há uma cobra em modo *TurboBoost*, no entanto, assegura a consistência dos recursos partilhados.

Na classe *Snake* existe um método sincronizado de acesso à lista de células ocupadas pela cobra. A sincronização neste método é necessária para que quando a lista de células é passada para o utilizador não esteja a ser obtida enquanto está a ser alterada.