

## Snake - Versão preliminar

Cada grupo terá que entregar até 18 de novembro, no respectivo *site* da turma, os ficheiros fonte (.kt) com o código do trabalho devidamente indentado e comentado.

O trabalho consiste em desenvolver um programa em *Kotlin* que usa a biblioteca *CanvasLib* para fazer uma versão preliminar do tradicional jogo [Snake](#).

Nesta versão, a cobra ainda não come maçãs e por isso não aumenta o seu comprimento. A cobra é apenas constituída por duas partes (cabeça e cauda) e atravessa os limites da área de jogo passando para o lado oposto. A área de jogo tem 20 x 16 células de 32 pixels.

A cobra avança de quadrícula a cada 250 milissegundos, podendo o utilizador mudar o sentido de deslocamento usando as quatro teclas de cursor. A cada 5 segundos aparece um bloco de tijolos numa célula livre aleatória. A cobra não pode atravessar os blocos e pára quando choca contra um deles, podendo o utilizador escolher outro sentido de deslocamento. Não é possível mudar a direção para o sentido oposto ao do deslocamento nem contra um bloco.

A figura 1 mostra um possível estado deste jogo quando já decorreram 25 segundos. Nesta versão não existe pontuação nem o jogo termina, mas a cobra pode ficar impedida de se deslocar e pode não haver células livres para mais blocos.



Figura 1: Janela do programa

Para a realização do jogo são fornecidos os [dois ficheiros de imagens](#) apresentados na figura 2. Algumas partes do ficheiro `snake.png` só serão usadas no 3º trabalho.

O programa só deve ter um ponto de mutabilidade, ou seja, só deve ter uma variável (`var`) do tipo `Game` e cujo valor é alterado por cada modificação, ou seja, quando a cobra altera a direção, quando a cobra se movimenta e quando aparece um bloco de tijolos. No entanto, as funções podem ter variáveis locais, que desaparecem quando a função retorna.

O tipo `Game` deve ser o seguinte tipo agregado, em que `Snake` e `Position` são outros tipos necessários que têm que ser definidos neste trabalho:

```
data class Game(val snake: Snake, val wall: List<Position>)
```

Na realização deste trabalho devem ser respeitadas as regras: Evitar mutabilidade; Não repetir código; Não fazer funções demasiado extensas; Não repetir valores com o mesmo significado nem usar “valores mágicos” (constantes sem um nome atribuído).

Além destas regras, as declarações dos tipos e das funções do programa devem estar distribuídas em vários ficheiros fonte (.kt) cuja responsabilidade deve ser descrita no comentário inicial de cada um deles, usando apenas um parágrafo. Por exemplo, devem existir ficheiros por cada tipo relevante do programa: `Game.kt`, `Snake.kt` e `Position.kt`.

Uma implementação do programa pretendido está disponível no ficheiro [trab2.jar](#). Para executar este programa deve bastar executá-lo “normalmente” no seu sistema operativo ou, caso não resulte, abra uma janela de comandos localizada na pasta onde reside o ficheiro `trab2.jar` e execute o comando: `kotlin trab2.jar` ou então o comando: `java -jar trab2.jar`

Bom trabalho,  
Pedro Pereira