

Programação

Terceiro trabalho prático

Semestre de Verão de 2022/2023

Cada grupo terá que entregar até 5 de Junho, no respetivo site do moodle da turma, os ficheiros fonte (.kt) com o código do trabalho devidamente indentado e comentado.

Este trabalho consiste em fazer um programa em Kotlin que será a versão final do [jogo Frogger](#) que foi realizado ao longo dos três trabalhos deste semestre. O objetivo do jogo consiste em controlar os saltos de um sapo até cada uma das casas vazias do topo, atravessando uma estrada sem ser atropelado pelos automóveis e atravessando um rio saltando sobre troncos e tartarugas sem cair na água.

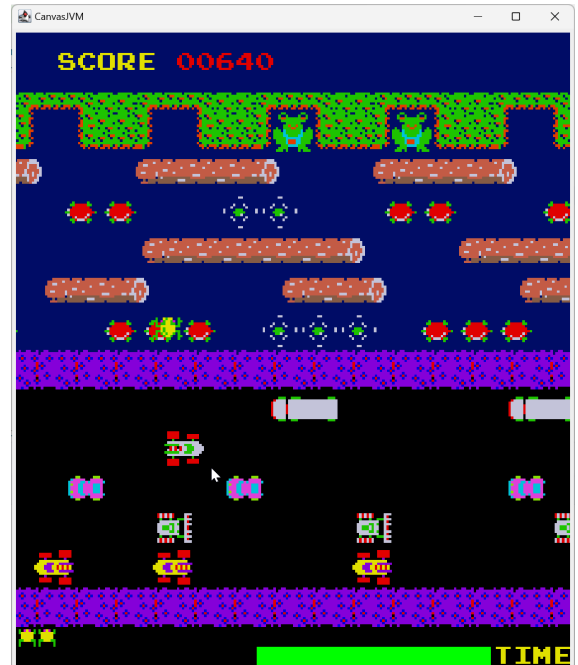
Está disponível online [o jogo original](#) desenvolvido em 1981.

Esta versão já tem as tartarugas e os troncos no rio e o sapo pode atravessar o rio saltando por cima das tartarugas e dos troncos até chegar às casas do topo.

A pontuação do jogo, o número de sapos disponíveis e o tempo de vida limitado de cada sapo, são para implementar opcionalmente.

Quando o sapo está em cima de uma tartaruga ou de um tronco, desloca-se na horizontal solidariamente com eles. O sapo morre afogado se cair na água ou estiver em cima de uma tartaruga que mergulhou, ou morre esmagado se não acertar numa casa vazia, sair pelas laterais em cima de tartarugas ou de troncos, ou terminar o tempo de vida.

O jogo termina com sucesso, aparecendo a mensagem YOU WIN, quando todas as casas ficam ocupadas com sapos, ou termina com insucesso, aparecendo a mensagem GAME OVER, quando o sapo morre e não há mais sapos disponíveis. O programa só termina quando for fechada a janela.



Uma implementação do programa pretendido está disponível no ficheiro [Trab3.jar](#).

Em anexo são fornecidos os ficheiros fonte [Log.kt](#), [Turtle.kt](#) e [Home.kt](#) com a definição dos tipos Log, Turtle e Home. O tipo Frogger, sem as partes opcionais, passa a ser:

```
data class Frogger(  
    val frog: Frog, val cars: List<Car>, val over: Boolean = false, // 2º trabalho  
    val logs: List<Log>,  
    val turtles: List<Turtle>,  
    val homes: List<Home>,  
    // Propriedades para as funcionalidades opcionais...  
)
```

Recomenda-se que desenvolvam o programa seguindo os seguintes passos:

- 1 - Apresentar todos os troncos a deslocarem-se no rio.
- 2 - Apresentar todos os grupos de tartarugas (mergulhadoras ou não) e nadar no rio.
- 3 - Permitir que o sapo salte para cima das tartarugas e dos troncos e se desloque solidariamente com eles.
- 4 - Permitir que o sapo salte na horizontal quando está em cima de troncos ou tartarugas.
- 5 - Permitir que o sapo salte para as casas do topo e detetar o fim de jogo com sucesso ou insucesso.

Seguindo-se as funcionalidades opcionais:

- 6 - Ter vários sapos disponíveis (mais 3 vidas) e o jogo só termina com insucesso quando todos morrem.
- 7 - Ter um tempo de vida limitado para cada sapo (30 segundos) até chegar a uma casa vazia.
- 8 - Ter pontuação. 10 pontos por cada linha de progressão e 10 pontos por cada segundo de vida restante quando chega a uma casa.

Bom trabalho,

ISEL, 18 de Maio de 2023