Programação

Terceiro trabalho prático

Semestre de Verão de 2020/2021

Cada grupo terá que entregar até 21 de junho, no respetivo *site* da turma, os ficheiros fonte (.kt) com o código do trabalho devidamente indentado e comentado.

O trabalho consiste em continuar o desenvolvimento do programa realizado no segundo trabalho para fazer uma versão jogável do tradicional jogo *Snake*.

Nesta versão, a cobra já come maçãs. Por cada maçã comida a cobra aumenta o seu comprimento em 5 elementos e a pontuação é incrementada. Os novos elementos aparecem durante os próximos movimentos na posição anterior da cabeça da cobra.

No início do jogo a cobra já tem 5 elementos, já existem alguns blocos de tijolos nos cantos da arena e aparece uma maçã numa posição livre aleatória. Quando a cobra come a maçã aparece logo outra noutra posição livre aleatória, enquanto existirem posições livres.

Continuam a aparecer blocos de tijolos novos a cada 5 segundos enquanto existirem posições livres. A cobra deixa de se movimentar quando vai contra um bloco de tijolos ou contra o próprio corpo.

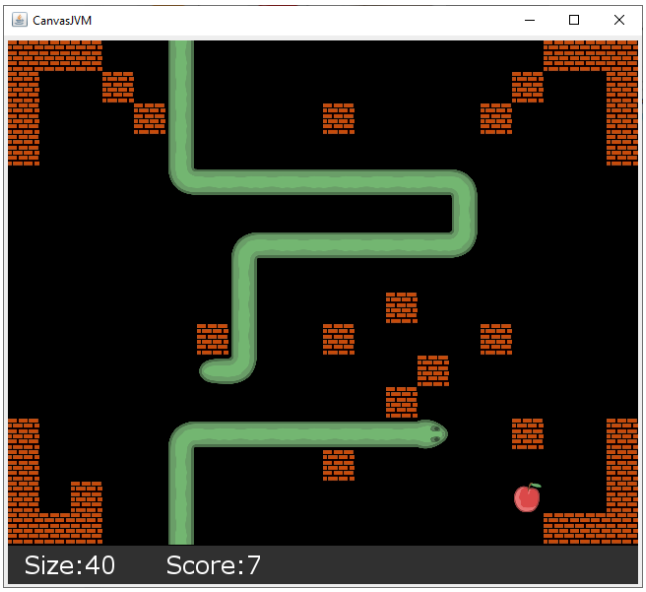

Figura 1: Janela do programa

O programa apresenta uma barra de estado em rodapé, onde apresenta o comprimento atual da cobra, a pontuação e outras informações.

O jogo termina quando a cobra já não se puder movimentar em nenhum sentido, aparecendo a mensagem “You Win” na barra de estado caso a cobra já tenha atingido o comprimento mínimo de 60 elementos, ou “You lose” caso contrário.

A única variável da função main() é do tipo **Game** que agora deve ser o seguinte tipo agregado:

```
data class Game(val snake:Snake, val wall:List<Position>,
                val apple:Position?, val score:Int ... )
```

em que as reticências representam outras propriedades que sejam necessárias.

O tipo **Snake** será agora o seguinte tipo:

```
data class Snake(val body:List<Position>, val dir:Direction,
                 val stopped:Boolean, val toGrow:Int ... )
```

em que a posição da cabeça é o primeiro elemento da lista; a posição da cauda é o último elemento da lista e a propriedade `toGrow` é o número de elementos que ainda faltam adicionar à cobra nos próximos movimentos.

Na realização deste trabalho devem continuar a ser respeitadas as regras já enunciadas nos trabalhos anteriores: Evitar mutabilidade; Não repetir código; Não fazer funções demasiado extensas; Não repetir valores com o mesmo significado nem usar “valores mágicos”; 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.

Opcionalmente, podem ser acrescentadas mais características ao jogo, como por exemplo:
- Reprodução de um som quando a cobra come uma maçã.
- Suportar de vários níveis de jogo com diferentes blocos iniciais e comprimento mínimo da cobra.
- Deslocar progressivamente a cabeça da cobra, em vez dos “saltos” de célula em célula.

Uma implementação do programa pretendido está disponível no ficheiro trab3.jar.

Bom trabalho.

ISEL, 1 de junho de 2021