

Programação

Segundo trabalho prático

Semestre de Inverno de 2020/2021

Cada grupo terá que entregar até 14 de dezembro, no respetivo *site* da turma, os ficheiros fonte (.kt) com o código do trabalho devidamente indentado e comentado.

O trabalho consiste em fazer um programa em *Kotlin* onde várias bolas se deslocam na área do *canvas*, com dimensão 400 x 600, e onde o jogador controla a posição horizontal de uma raquete com deslocamentos do rato. As bolas refletem nas laterais e no topo. A raquete impede que as bolas desapareçam pela base. A figura 1 mostra um possível estado deste jogo onde já estão 7 bolas na área.

Quando o jogo começa não existem bolas, mas uma nova bola aparece pela base a cada 5 segundos. O jogo termina quando é fechada a janela ou quando desaparece a última bola que estava na área.

Cada bola tem um raio de 7 pixels e move-se a cada 10 milissegundos segundo um vetor de deslocamento. O deslocamento na horizontal (dx) está compreendido no intervalo $[-6..6]$ e o deslocamento na vertical (dy) é -4 quando a bola se move para cima, ou 4 quando se move para baixo. A figura 2 representa estas propriedades.

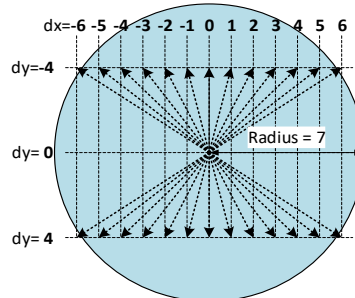


Figura 2: Deslocamento da bola

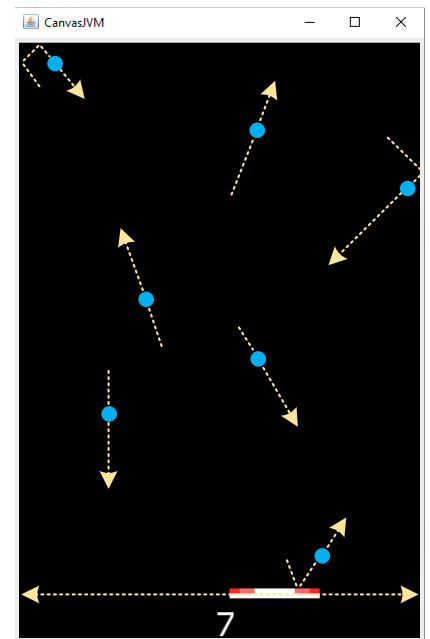


Figura 1: Janela do programa

O topo e as laterais provocam reflexões simétricas nas bolas, sendo invertido o sinal do deslocamento respetivo. O topo inverte o deslocamento vertical (dy) e as laterais invertem o deslocamento horizontal (dx), tal como mostra a figura 3.

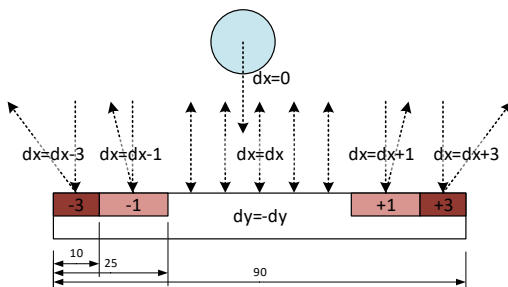


Figura 4: Reflexão na raquete

A largura total da raquete está dividida em três tipos de zonas, tal como mostra a figura 4. A zona central tem 40 pixels de largura e provoca reflexões simétricas. As zonas dos extremos têm 10 pixels e adicionam ou subtraem 3 pixels ao deslocamento horizontal resultante. As zonas intermédias têm 15 pixels e adicionam ou subtraem 1 pixel ao deslocamento horizontal resultante. Em qualquer dos casos, o deslocamento horizontal (dx) resultante deve estar sempre limitado ao intervalo $[-6..6]$.

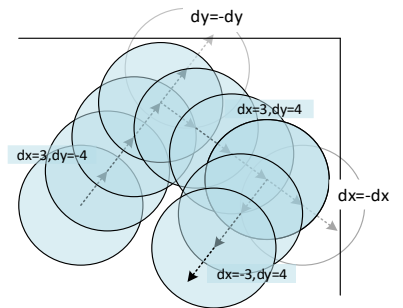


Figura 3: Reflexão no topo e nas laterais

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 (movimento da raquete ou das bolas; aparecimento e desaparecimento de bolas).

O tipo **Game** deve ser o seguinte tipo agregado, em que **Area**, **Ball** e **Racket** são outros tipos necessários neste trabalho:

```
data class Game(val area:Area, val balls:List<Ball>, val racket:Racket)
```

Na realização deste trabalho devem ser respeitadas as regras já enunciadas no primeiro trabalho: Evitar mutabilidade; Não repetir código; Não fazer funções demasiado extensas; Não repetir valores com o mesmo significado nem usar “valores mágicos”.

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, deve existir um ficheiro para cada tipo relevante do programa: *Game.kt*, *Ball.kt* e *Racket.kt*.

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

Bom trabalho.