

Programação Orientada por Objetos
Licenciatura em Engenharia Informática e de Computadores
Segundo Trabalho prático
Semestre de Inverno de 2017/2018

Este trabalho prático tem o propósito de exercitar os mecanismos da linguagem Java e o desenvolvimento de aplicações em *Android* usando componentes **Activity** e a hierarquia de elementos **View**.

Cada grupo deve entregar até dia 4 de dezembro os ficheiros fonte (java e xml) do projeto *Android* realizado, comprimidos num ficheiro, não sendo necessário relatório.

O trabalho consiste em desenvolver um jogo para *Android*, com uma apresentação aproximada à da figura. Onde consta a melhor pontuação e o respetivo nome do jogador, a pontuação atual e a área de jogo.

A área de jogo é composta por uma grelha de retângulos em que é apenas apresentado mais um retângulo verde por cada progressão, até a área ficar toda verde. Para progredir sem perder, o jogador tem que tocar apenas no último retângulo apresentado, ganhando mais um ponto.

O jogo termina quando o jogador tocar noutro retângulo verde qualquer, ficando com os pontos até esse momento e recomeçando novo jogo. Se forem obtidos mais pontos que o melhor, é perguntado o nome do jogador e essa informação (pontos e nome) é guardada de forma persistente.

A área de jogo deve esconder todos os retângulos durante um segundo (aproximadamente) antes de apresentar os retângulos da próxima fase da progressão.

Quando for tocado o retângulo que preenche totalmente a área, o jogo progride para uma grelha de maior dimensão, passando pelas dimensões 3x2, 4x2, 4x3, 5x3, 5x4, 6x4, etc.

O jogo deve manter o seu estado quando a atividade é interrompida temporariamente, como é o caso do que acontece quando é rodado o ecrã do dispositivo.

A área de jogo é constituída por um objeto de uma classe derivada de **View**, que redefine o método **onTouchEvent()** para capturar os toques no retângulos e redefine o método **onDraw()** para apresentar os retângulos visíveis.

Na implementação deste trabalho, deve usar o modelo MVC, em que o controlo é a **Activity** que mantém uma referência para o visual (área de jogo) e outra para o modelo (o estado do jogo).

Na manutenção de estado, considere guardar no **Bundle** um *array* de bytes e implementar no modelo os métodos **save(OutputStream)** e **load(InputStream)** utilizando as classes **ByteArrayOutputStream** e **ByteArrayInputStream**.

Deve ser usada uma instância de **AlertDialog** para perguntar o nome do jogador.

Para implementar a temporização sem retângulos visíveis use o método **postInvalidateDelayed()** ou o método **postDelayed()** da classe **View**.

Uma possível implementação deste trabalho pode ser experimentada instalando a aplicação em [anexo](#). Para instalar esta aplicação no emulador de *Android* basta arrastar o ficheiro *apk* para a área de aplicações.

