

# Programação Orientada por Objetos

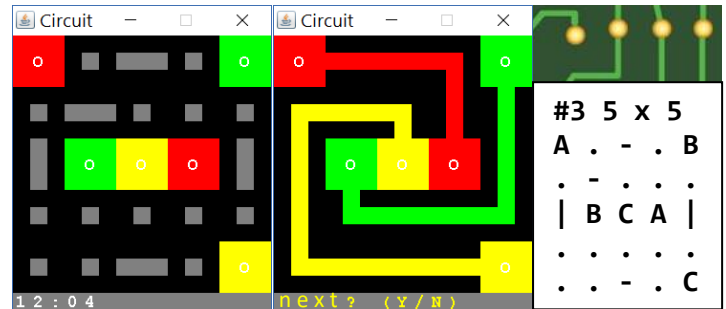
Primeiro trabalho prático  
Semestre de Verão 2016/2017

**Objetivos:** Neste trabalho pretende-se que os alunos adquiram prática de utilização de herança e polimorfismo usando a abordagem MVC na implementação de um jogo em consola de texto.

**Entrega:** Cada grupo entregará no *site* da sua turma, até dia 24 de abril, os ficheiros fonte (java) do projeto realizado e o documento que descreve a implementação do trabalho contendo o diagrama de classes.

**Enunciado:** Implemente o jogo designado por **Circuit** que permite desenhar as pistas que fazem as ligações entre os terminais (da mesma cor) de um circuito impresso.

A figura 1 apresenta o aspeto inicial de um nível deste jogo. A área do circuito é dividida numa grelha em que cada célula ocupa 3x3 caracteres na janela da Consola. Os terminais são representados por quadrados com um 'O' no centro. As restantes células terão que ser ocupadas pelas pistas que fazem a ligação dos terminais, sujeitando ou não que a pista passe na horizontal ou vertical. O utilizador desenha as pistas arrastando o cursor do rato desde um terminal ao outro da mesma cor. A figura 2 mostra o nível já concluído, em que as pistas ligam os terminais da mesma cor, passando por todas as células disponíveis.



Os níveis do jogo são descritos num ficheiro de texto, em que a primeira linha indica o número do nível e as dimensões (linhas x colunas) e as linhas seguintes descrevem as características de cada célula. As letras representam terminais, cada símbolo '|' indica que a pista só pode passar na vertical, cada símbolo '-' força a que a pista só possa passar na horizontal e cada '.' permite que a pista passe de qualquer forma. A figura 3 apresenta um troço desse ficheiro que descreve o nível mostrado nas figuras anteriores.

O ficheiro anexo T1POO.zip inclui uma versão do jogo pretendido. Este *zip* contém um *jar* com o programa, o ficheiro de descrição dos níveis, o *jar* da biblioteca ConsolePG e os ficheiros fonte das classes *Game*, *Loader* e *CellView* representadas na figura 4. Destas classes também é possível depreender alguns dos métodos públicos das restantes classes.

Implemente o jogo descrito seguindo a abordagem MVC.

1. Defina uma hierarquia de classes que represente os vários tipos de células (classe *Cell* e suas derivadas) e a classe principal do **modelo** (classe *Circuit*) que implementa a lógica do jogo (as regras). As instâncias destas classes representarão o estado corrente de um nível do jogo e não devem resolver aspetos de visualização. O código destas classes será reutilizado na implementação deste jogo em *Android* sem necessidade de alterações. *Position* e *Dir* são tipos auxiliares que devem também fazer parte do package do modelo.
2. A componente de **controlo** que interage com o utilizador (classe *Game*), processa as teclas e os eventos do rato. O **controlo** carrega cada nível, usando a classe *Loader*, chama métodos do **modelo** para evoluir no estado do jogo e métodos da componente de **visualização** para redesenhar as células.
3. Realize a componente de **visualização** (classe *Panel*). Use a classe *Console*, da biblioteca ConsolePG incluída no *zip*, para representar cada célula na respetiva posição e cores pretendidas. A classe *CellView* e suas derivadas implementam a visualização de cada tipo de célula.

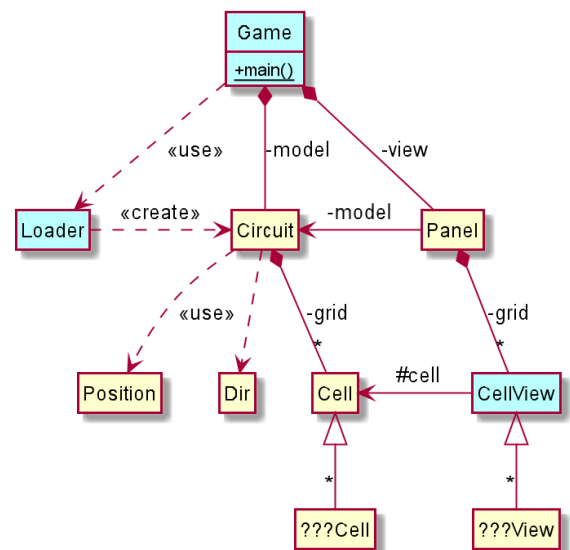


Figura 4: Diagrama de classes