

# (Introdução à) Programação Trabalho Prático

Vitor Beires Nogueira

2018–2019

## 1 Enunciado

O jogo Dots and Boxes é um conhecido jogo de "lápis e papel" para dois (ou mais) jogadores. Pretende-se com este trabalho a implementação de tal jogo (e algumas variantes) em Python. Sugere-se que tal implementação seja faseada e para tal propõe-se as seguintes fases:

### 1.1 Player vs Player

Nesta fase pretende-se que seja possível jogar um humano contra outro em modo texto. O jogo deverá seguir o seguinte fluxo:

1. Escrever no standard output

```
0 - Exit
1 - Player vs Player
```

2. Se o jogador introduzir 0 o jogo termina. Se o jogador inserir 1, o programa deverá escrever:

```
Grid dimension? (lines, columns)
Um exemplo possível é (3,3)
```

3. O programa deve desenhar o tabuleiro (em modo de texto) de acordo com as dimensões da alínea anterior

```
  0  1  2
0  .  .  .

1  .  .  .

2  .  .  .
```

e escrever:

```
Player 1: coordinates for the line endpoints? (x1,y1)(x2,y2)
```

e ficar à espera de uma jogada. Cada jogada deverá ser da forma  $(x_1, y_1)(x_2, y_2)$ , em que  $(x, y)$  é uma coordenada de um ponto e considerando o canto superior esquerdo com a coordenada  $(0, 0)$ . Por exemplo, para o tabuleiro de  $(3, 3)$  uma jogada pode ser  $(0, 2)(1, 2)$ .

4. Perante o "input" de uma jogada (válida), o programa deve escrever o tabuleiro resultante

```
    0  1  2
0  .  .  .
   |
1  .  .  .

2  .  .  .
```

e de seguida solicitar uma nova jogada ao outro jogador ou indicar que o jogo terminou:

Game ended. Score is

## 1.2 Resume/Save Game

Nesta fase, o menu inicial deverá conter mais uma opção:

```
0 - Exit
1 - Player vs Player
2 - Resume Game
```

O que se pretende é a possibilidade de regressar a um jogo previamente guardado. Para tal, será igualmente necessário adicionar a possibilidade de guardar um jogo ("Save Game") como possibilidade de jogada. Por isso, antes de um jogador inserir uma jogada, deverá ter o seguinte menu:

```
0 - Save Game
1 - Insert play
```

## 1.3 Player vs Computer

Nesta fase, o menu inicial deverá conter mais uma opção:

```
0 - Exit
1 - Player vs Player
2 - Resume Game
3 - Player vs Computer
```

Caso o utilizador escolha a opção 3, deverá passar ao seguinte menu:

```
1 - First Player
2 - Second Player
```

ou seja, se o jogador é o primeiro ou é o computador, respectivamente. Quando for a vez do jogador, deverá solicitar um input semelhante ao proposto na alínea anterior:

Player 1: coordinates for the line endpoints? (x1,y1)(x2,y2)

Nesta fase, cada grupo deve implementar algum estratégia de jogo que não seja simplesmente jogar aleatoriamente <sup>1</sup>.

## 2 Condições Gerais

- O trabalho deverá ser efectuado por grupos de 2 elementos. Em casos excepcionais, e acordados com o docente, poderão ser feitos por apenas 1 elemento.
- Será aplicado o código de conduta do Departamento de Informática. Em caso de fraude, para além reprovação à disciplina serão notificadas as comissões de curso respectivas.
- A entrega será via Moodle e até ao final do dia 8 de Janeiro de 2019 (para mais detalhes consultar a secção 3).
- Será discutido no dia 10 de Janeiro de 2019 em horário a anunciar.
- O relatório deverá conter a identificação dos elementos do grupo assim como uma descrição sucinta do funcionamento do trabalho para cada um dos itens do enunciado. Preferencialmente não deve ter código.
- O trabalho será classificado entre 0 e 20 valores e serão factores de avaliação:
  - a clareza dos algoritmos implementados;
  - a correcção do código;
  - a legibilidade do código (a qualidade dos identificadores, os comentários que o acompanham, etc);
  - a qualidade do relatório (lendo-o, deve-se ficar com uma ideia precisa do funcionamento do trabalho).
- Todo o código deverá funcionar no Python 3.\*

## 3 Entrega

- O upload do trabalho tem de ser efectuado até ao dia 8 de Janeiro de 2019 através do Moodle e só deverá ser efectuado pelo elemento do grupo que tiver o número menor.
- Deverá ser um ficheiro .tar.gz ou .zip que ao descomprimir crie uma pasta `num1_..._numi` em que `num1` e `numi` são os números dos alunos que compõe o grupo por ordem crescente, por exemplo `123_456_789`. Nessa pasta deverá existir uma subpasta `src` com todo o código necessário para testar o programa, incluindo um ficheiro chamado `main.py` (`$ python principal.py` deverá iniciar o programa). Deverá também existir na directoria principal um ficheiro (`relatorio`) em formato texto, HTML ou PDF com o relatório.

---

<sup>1</sup>Tal algoritmo ou heurística deve ser devidamente explicada no relatório.