

# O Jogo do Galo

## Trabalho Prático 1

### Programação

Departamento de Informática  
Universidade da Beira Interior

**Data limite para entrega:** Terça-feira, 24 de Novembro de 2020, 23h59m

**Grupos:** Este trabalho é para ser realizado em grupos de dois elementos. Os grupos podem ser formados por elementos de turnos práticos distintos mas apenas entre grupos que são lecionados pelo mesmo docente, i.e. alunos dos turnos práticos PL1, PL2, PL3 e PL4 podem formar grupo entre si e os alunos do PL5 e PL6 também. Alunos dos turnos PL5 e PL6 não podem formar grupo com alunos dos turnos PL1, PL2, PL3 e PL4.

Alunos sem inscrição em turno prático devem inscrever-se com a maior brevidade e antes da data limite para entrega do trabalho, caso contrário **o trabalho poderá não ser considerado**.

**Código Base:** Encontra-se disponível no Moodle o ficheiro `JogoGalo.c` que contém uma estrutura base para a implementação deste trabalho. As indicações presentes no ficheiro `JogoGalo.c` são para ser consideradas em conjunção com este enunciado. Note que não pode alterar a assinatura das funções nem as pode remover. Se usar outras funções auxiliares, estas terão de ser utilizadas dentro das funções existentes no ficheiro para serem consideradas.

Para que a sua submissão seja considerada **terá que usar como nome do ficheiro submetido os números de aluno dos elementos do grupos separados pelo carácter ‘\_’ (underscore)**. Por exemplo, se os alunos com números a12345 e Z123 formam grupo, o ficheiro deverá chamar-se `a12345_Z123.c`

**Trabalhos que não compilem não serão considerados, sendo nesse caso atribuída a nota de zero valores.**

# 1 Trabalho a Realizar

O objectivo deste trabalho é escrever um programa em C que permita jogar o Jogo do Galo<sup>1</sup> e que satisfaça os requisitos especificados nesta secção.

No Jogo do Galo dois jogadores, o X e o O, jogam à vez, marcando células num tabuleiro 3×3. Cada jogador usa uma marca diferente (normalmente, o X usa a marca ‘X’ e O usa a marca ‘O’). O vencedor é o primeiro jogador que conseguir colocar três das suas marcas numa fila horizontal, vertical ou diagonal. Por exemplo, a seguinte grelha mostra uma situação onde O foi o primeiro a jogar e venceu o jogo com três marcas consecutivas na diagonal.

O	X	
	O	X
		O

## 1.1 Representação da Grelha

Considere que a grelha é representada por um array multidimensional 3×3 de **chars**:

```
#define N 3
char grid[N][N]
```

Cada célula deverá conter o caracter ‘`␣`’ (espaço branco)<sup>2</sup>, ou o caracter ‘X’ ou o caracter ‘O’ (maiúsculas). As coordenadas da grelha são identificadas por  $(x, y)$  onde  $0 \leq x < 3$  e  $0 \leq y < 3$ .

## 1.2 Funções a Implementar

As funções a escrever durante este projecto são as seguintes:

### 1.2.1 Inicialização da Grelha (1 valor)

**Assinatura:** `void inicializaGrelha(char grid[N][N])`

**Descrição:** Esta função recebe uma grelha como argumento e inicializa a grelha, colocando em todas as células o caracter ‘`␣`’ (espaço branco). **A inicialização tem que ser feita utilizando ciclos. Inicialização por atribuição manual resulta em penalização.**

<sup>1</sup>Também conhecido como Jogo da Velha ou Três em Linha: [https://pt.wikipedia.org/wiki/Jogo\\_da\\_velha](https://pt.wikipedia.org/wiki/Jogo_da_velha)

<sup>2</sup>O símbolo `␣` indica a existência de um espaço branco.

### 1.2.2 Visualização da Grelha (2.5 valores)

**Assinatura:** `void imprimeGrelha(char grid[N][N])`

**Descrição:** Esta função recebe uma grelha como argumento e imprime-a no ecrã. Por exemplo, logo após inicialização da grelha através da função `inicializaGrelha`, a função `imprimeGrelha` deverá imprimir a grelha seguinte:

```

      |   |
      ---
      |   |
      ---
      |   |

```

Por outro lado, se a grelha recebida como argumento for a que é mostrada como exemplo na página 2, então esta função deverá imprimir o seguinte:

```

      0 | X |
      ---
      | 0 | X
      ---
      |   | 0

```

**IMPORTANTE:** O output da função tem que corresponder exactamente ao output mostrado acima, caso contrário será rejeitado pelo sistema Mooshak (ver secções 2 e 3 para detalhes sobre o sistema). Note que:

- As linhas de símbolos contêm 11 caracteres seguidos do carácter ‘\n’. Por exemplo, na grelha acima, a primeira linha termina com os seguintes caracteres: `|_ _ _\n`
- As linhas de divisão (i.e. as compostas por traços) contêm 12 caracteres seguidos do carácter ‘\n’. Por exemplo, na grelha acima, a segunda linha termina com os seguintes caracteres `- _ _\n`

### 1.2.3 Execução de Jogada (2.5 valores)

**Assinatura:** `int jogada(char grid[N][N], int x, int y, int jogador)`

**Descrição:** Esta função recebe como argumentos uma grelha, dois inteiros `x` e `y`, correspondentes a uma coordenada e um inteiro `jogador` que indica o jogador que realizou a jogada que está a ser executada. Se a jogada for realizada pelo jogador 0, a variável `jogador` é 0 (zero); se for realizada pelo jogador X, é 1 (um). O comportamento desta função corresponde ao jogador identificado pela variável `jogador` colocar a sua marca na posição  $(x, y)$ .

Por exemplo, se no estado inicial (grelha vazia), o jogador 0 colocar a sua marca na posição  $(0, 1)$ , o programa deverá executar a instrução

```
jogada(grid, 0, 1, 0)
```

A representação da grelha após esta jogada é:

```

      | 0 |
      ---
      |  |
      ---
      |  |

```

**Valor de retorno:** A função `jogada` devolve um inteiro com a seguinte semântica:

- Se a jogada é possível e foi executada, a função devolve o valor 0 (zero)
- Se as coordenadas  $(x, y)$  são inválidas (i.e., estiverem fora da grelha), a função devolve o valor 1 (um)
- Se a célula  $(x, y)$  já está marcada (i.e., não está vazia), a função devolve o valor 2 (dois)

#### 1.2.4 Verificação de Vencedor (4 valores)

**Assinatura:** `int tresEmLinha(char grid[N][N], int x, int y)`

**Descrição:** Esta função recebe como argumentos uma grelha e dois inteiros `x` e `y`, correspondentes a uma coordenada. O objectivo da função é verificar se o jogador que colocou a sua marca na posição  $(x, y)$  tem três marcas em linha que incluem a marca na posição  $(x, y)$ .

**Valor de retorno:** A função `tresEmLinha` devolve um inteiro com a seguinte semântica:

- Se a posição  $(x, y)$  estiver vazia, a função devolve -1 (valor negativo 1)
- Se a posição  $(x, y)$  faz parte de uma linha de três marcas iguais consecutivas, a função devolve 1 (um)
- Caso contrário, a função devolve 0 (zero)

Exemplo: para a seguinte grelha, a função `tresEmLinha(grid, 2, 2)` devolve 1.

```

      0 | X |
      ---
      | 0 | X
      ---
      |  | 0

```

### 1.2.5 Função *main* (6 valores)

**Assinatura:** `int main(void)`

**Descrição:** Esta função deve interagir com o utilizador e implementar um jogo completo entre dois jogadores. Todas as funções indicadas anteriormente terão de ser utilizadas de forma apropriada dentro desta função. Os requisitos incluem os seguintes pontos (mas não estão limitados a estes):

- O estado da grelha deverá ser mostrado no início do jogo e após cada jogada (incluindo a última)
- Todos os inputs deverão ser validados
- Deverá imprimir mensagens que permitem uma interação com o jogo que seja simples e intuitiva
- A cada passo deverá ser indicado qual o jogador que deve proceder com uma jogada
- No final do jogo deve imprimir uma mensagem que indica qual o jogador vencedor ou que o jogo terminou em empate

## 2 Entrega do Trabalho

A entrega do trabalho deve ser feita via Mooshak e Moodle. No início do ficheiro de código tem de identificar claramente o nome, número de aluno, e turno prático dos elementos do grupo (num comentário). Todas as funções devem estar devidamente documentadas com comentários no código.

**Entrega no Mooshak.** O primeiro passo da entrega consiste no registo de uma conta no sistema Mooshak. Para obter a sua senha, terá de inserir o seu número de aluno (e.g., a12345 ou Z123) no formulário disponível em

`https://penhas.di.ubi.pt/~amendes/cgi-bin/getpass`

A password será gerada e enviada para o seu email registado na plataforma Moodle (não se esqueça de verificar a pasta de “E-mail de Lixo” caso não receba a password). A partir desse momento, poderá aceder ao sistema Mooshak onde deverá entregar o seu trabalho:

`https://penhas.di.ubi.pt/~amendes`

A submissão do trabalho deverá ser feita por apenas um elemento do grupo. Para submeter o trabalho, deverá:

- Utilizar o botão “Choose file”, seleccionando o ficheiro com extensão .c contendo todo o código do seu projeto. O seu ficheiro .c deve conter a implementação das funções pedidas no enunciado. Relembre que para que a sua submissão seja considerada **terá que usar como nome do ficheiro submetido os números de aluno dos elementos do grupos separados pelo caracter ‘\_’ (underscore)**. Por exemplo, se os alunos com números a12345 e Z123 formam grupo, o ficheiro deverá chamar-se a12345\_Z123.c
- De seguida clique no botão “Submit” para efetuar a submissão. **Aguarde (10-30 seg) para que o sistema processe a sua submissão.**
- Quando a submissão tiver sido processada, poderá visualizar na tabela o resultado correspondente. Se o resultado for *Wrong answer*, poderá clicar no resultado para obter dicas.
- Para sair do sistema utilize o botão “Logout”.

Mais detalhes sobre o sistema Mooshak serão mostrados na Secção 3.

**Entrega no Moodle.** Após o envio da submissão final no Mooshak terá de enviar o mesmo ficheiro .c no Moodle, utilizando o link de submissão do TP1. A submissão tem de ser feita através da conta do aluno que submeteu o trabalho no Mooshak. Garanta que fez todos os passos necessários para a submissão. Terá de aceitar a declaração de submissão e finalizar a submissão carregando no botão “Submeter o trabalho”. Note que trabalhos não submetidos como finais não serão considerados e que caso o ficheiro submetido no Moodle seja diferente do ficheiro final submetido no Mooshak, a sua submissão poderá não ser considerada.

### 3 Sistema Mooshak

Submeta o seu projeto atempadamente, dado que as restrições seguintes podem não lhe permitir fazê-lo no último momento:

- Só poderá efetuar uma nova submissão pelo menos 15 minutos depois da submissão anterior.
- Só são permitidas 10 submissões em simultâneo no sistema, pelo que uma submissão poderá ser recusada se este limite for excedido<sup>3</sup>.
- Não pode ter submissões duplicadas, ou seja, não pode ter uma submissão igual a outra anterior.

---

<sup>3</sup>Note que o limite de 10 submissões simultâneas no sistema Mooshak implica que, caso haja um número elevado de tentativas de submissão sobre o prazo de entrega, alguns grupos poderão não conseguir submeter nessa altura e verem-se, por isso, impossibilitados de submeter o código dentro do prazo.

- Será considerada para avaliação a última submissão (mesmo que tenha pontuação inferior a submissões anteriores). Deverá, portanto, verificar cuidadosamente que a última entrega realizada corresponde à versão do projeto que pretende que seja avaliada. **Não serão abertas exceções.**

### 3.1 Testes Automáticos

A avaliação da execução será feita através do sistema Mooshak, que executará o seu programa num conjunto de testes. O tempo de execução de cada teste está limitado, bem como a memória utilizada. O sistema não deverá ser utilizado para *debug* e como tal, relembra-se que o tempo entre submissões e o número de submissões é limitado.

**Cada aluno tem direito a 15 submissões sem penalização no Mooshak. Por cada submissão adicional serão descontados 0,2 valores na componente de avaliação automática.**

Não será disponibilizado qualquer tipo de informação sobre os casos de teste utilizados pelo sistema de avaliação automática.

**Pontuação:** Como a avaliação automática vale 100 pontos (equivalente a 10 valores) da nota, uma submissão no Mooshak obtém a nota máxima de 100 pontos. Veja a próxima secção para mais informações sobre a pontuação final.

## 4 Classificação e Critérios de Correção

A nota do trabalho será baseada nos seguintes aspectos:

1. Execução correcta das funções requeridas, excepto a função *main* (até 10 valores). Esta parte da avaliação é feita recorrendo ao sistema Mooshak que sugere uma nota face aos vários aspectos considerados. Esta pontuação não é garantida e poderão existir penalizações posteriores caso o código não esteja implementado de acordo com os requisitos apresentados ou não siga boas práticas de programação.
2. Implementação do jogo completo e qualidade da interacção com o utilizador (até 6 valores).
3. Estilo de programação e facilidade de leitura do código, nomes de variáveis e funções bem escolhidos, qualidade (e não quantidade) dos comentários, tamanho das funções e utilização dos elementos discutidos nas aulas (até 4 valores).

**Conduta:** Chama-se a atenção para o Código de Integridade e para o Regulamento Disciplinar dos Estudantes da Universidade da Beira Interior. **Submissões com código semelhante serão investigadas.**

## 5 Recomendações

Esta secção lista recomendações importantes que deverá seguir e ter em consideração:

- Leia todo o enunciado, procurando perceber o objectivo das várias funções pedidas. Em caso de dúvida de interpretação, utilize o horário de dúvidas para esclarecer as suas questões ou o canal `#avaliacoes` no Slack da UC.
- Tenha o cuidado de reproduzir *fielmente* os outputs ilustrados nos vários exemplos.
- O seu código será compilado no servidor do Mooshak utilizando o `gcc` com as opções `-Wall` e `-std=c99`. A opção `-Wall` mostra avisos sobre práticas que se devem evitar. Caso algum aviso seja mostrado na compilação do seu trabalho, este não será aceite pelo Mooshak. Assim, é recomendado que só submeta o seu trabalho depois de a sua compilação com a opção `-Wall` não mostrar nenhum aviso. A opção `-std=c99` compila o código de acordo com o Standard C99.

Portanto, um ficheiro com o nome `JogoGalo.c` deverá ser compilado localmente utilizando o comando

```
gcc -o JogoGalo JogoGalo.c -Wall -std=c99
```

e só deverá ser submetido no Mooshak após compilação com sucesso e sem avisos.

- Inicie a resolução deste trabalho assim que possível. Deixar a realização do projecto para os últimos dias é desaconselhado, pois poderá ficar sem tempo para resolver problemas que surjam.
- Casos de fraude académica serão levados muito a sério. Plágio resultará em reprovação na disciplina e será tratado de acordo com o regulamento disciplinar em vigor. O corpo docente da cadeira será o único juiz do que se considera ou não plágio. Não vale a pena copiar código e apenas alterar nomes de variáveis, funções, etc. O corpo docente utilizará software especializado na detecção de plágio em trabalhos de programação.