

Miniprojeto 2 (Introdução)

Realização do miniprojeto em grupo:

O trabalho deve ser realizado em grupo. O grupo pode ter **no máximo 2 elementos** e não é aconselhável a realização do miniprojeto individualmente (embora seja possível). É permitido que os grupos sejam formados por alunos de turmas diferentes. No caso destas terem professores diferentes, terão que optar por um deles para a defesa e avaliação do trabalho.

Níveis de dificuldade do miniprojeto:

São propostos no enunciado vários níveis de dificuldade – níveis I, II e III, para 2, 3 e 4 valores, respetivamente –, cabendo aos autores do trabalho escolherem aquele que pretendem implementar.

Cotação do miniprojeto:

O miniprojeto é cotado para um máximo de 4 valores.

Material a submeter no InforEstudante:

Cada grupo deve submeter, até à data limite fixada para a entrega do miniprojeto, um ficheiro compactado (.zip ou .rar) contendo os seguintes elementos:

- Ficheiros de código fonte C++ do programa implementado (e.g. todos os ficheiros do projeto QtCreator);
- Ficheiro PDF com um pequeno relatório sobre a execução do miniprojeto (cerca de uma página A4), contendo uma descrição do programa implementado, nomeadamente a justificação das principais opções tomadas em termos de estruturas de dados e algoritmos utilizados.

Defesa do miniprojeto:

O miniprojeto terá de ser defendido oralmente por ambos os elementos do grupo, em data e hora a definir entre o grupo de alunos e o docente avaliador do trabalho (ver primeiro parágrafo) dentro os horários definidos para o efeito no InforEstudante. **A não comparência no ato da defesa do miniprojeto implica a anulação do mesmo (com a classificação de 0%) para o aluno que falte.**

Avaliação:

Na avaliação, será dada ênfase às estruturas de dados e aos algoritmos utilizados na realização do miniprojeto. **A implementação deve ser orientada a objetos, caso contrário haverá uma forte penalização.**

Fraude:

Qualquer tentativa de fraude, tais como a utilização dissimulada de código fonte disponível na Internet ou o plágio de código fonte escrito por alunos externos ao grupo, será punida com a anulação do miniprojeto. A utilização de código obtido na Internet ou de terceiros para algumas funções específicas e limitadas deve ser mencionada no relatório do trabalho.

Miniprojeto 2

Jogo 4 em Linha

Descrição Sumária:

Pretende-se que seja desenvolvido um *software* para permitir a um utilizador jogar o “Jogo 4 em Linha” com o computador.

Neste segundo trabalho vai ser continuado e melhorado o primeiro trabalho, acrescentando-se os itens a vermelho abaixo.



Este jogo é jogado por dois jogadores alternadamente, num tabuleiro vertical com 42 posições (6 linhas e 7 colunas). Vence o jogo aquele que primeiro conseguir colocar 4 peças seguidas, na horizontal, na vertical ou na diagonal. O jogador deverá efetuar a sua jogada escolhendo a coluna onde quer colocar a peça, que irá “cair” até encontrar a última linha disponível dessa coluna (em cada coluna as peças vão sendo colocadas uma em cima das outras, em linhas consecutivas).

O jogo termina quando um dos jogadores atingir o objetivo (4 em linha), ou quando o tabuleiro estiver completo (não existirem posições vazias); nesta última situação, o jogo termina empatado.

Com um jogo perfeito, o primeiro jogador pode forçar uma vitória, antes ou na 41ª jogada, começando na coluna do meio. O jogo é um empate teórico quando o primeiro jogador começa nas colunas adjacentes ao centro. Quando o primeiro jogador começa numa das quatro colunas externas (1, 2, 6 ou 7), o segundo jogador pode forçar uma vitória. Há 4.531.985.219.092 configurações para todos os tabuleiros de jogo preenchidos com 0 a 42 peças.

Descrição Detalhada:

O jogo apresenta vários modos de funcionamento:

- 1 – *elementar*, em que o computador joga aleatoriamente, provavelmente perdendo a maior parte das vezes;
- 2 – *básico*, em que o computador procura ganhar na jogada corrente, ou de uma forma aleatória caso isso não seja possível;
- 3 – *médio*, em que o computador procura ganhar na jogada corrente e, não sendo isso possível, evitando que o utilizador ganhe na jogada seguinte;
- 4 – *avançado*, em que o computador nunca perde (se for ele a iniciar), empatando ou podendo ganhar se o jogador for distraído ou não souber jogar.
- 5 – *2 jogadores*, em que o jogo é jogado por 2 jogadores, A e B.
- 6 – *computador contra computador*, em que o jogo só é jogado pelo computador, havendo 2 jogadores virtuais que jogam alternadamente, sempre que o utilizador manda jogar.

O jogo é iniciado alternadamente pelo jogador (utilizador) e pelo computador, i.e. pelo jogador A ou pelo jogador B, respetivamente.

O tabuleiro pode ser redesenhado a cada jogada, apagando-se o anterior, ou desenhando novo tabuleiro para cada jogada, abaixo do tabuleiro anterior (vendo-se assim as jogadas anteriores).

O computador mostra sempre no ecrã o nº da jogada e quem está a jogar (o jogador ou o computador).

O computador mostra sempre no ecrã o modo de funcionamento.

O computador mostra sempre no ecrã as estatísticas: nº de vitórias do jogador, nº de vitórias do computador e nº de empates.

Sugestões para implementação:

- Criar uma classe em que um dos atributos é a configuração atual do tabuleiro; outros atributos poderão ser quem iniciou, quem está a jogar, nº de jogadas, etc.

- Criar um método para verificar se, com uma determinada configuração do tabuleiro, um dos jogadores ganha (e qual). Basta analisar as 4 direções (horizontal, vertical e duas diagonais) a partir da última peça jogada. Para verificar se há 4 em linha, pode ser usada a função `strstr (<cstring>)` ou equivalente.
 - Criar um contador de jogadas. Verificar a cada jogada se o número de jogadas atingiu o limite e terminar o jogo em caso afirmativo.
 - Criar um método para determinar as (até 7) jogadas possíveis e/ou se uma jogada é válida.
 - Criar uma função/método para aplicar uma jogada, identificada por `Xc1`, em que `X` é 'U' ou 'C' (ou 'A' e 'B'), `c` é a identificação da coluna, que pode ser de 1 a 4, e `1` a identificação da linha.
 - O tabuleiro pode ter o aspeto seguinte no início e a meio do jogo, respetivamente:
- | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|--|--|---|---|---|---|---|---|---|--|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | C | | | |
| | | | | | | | | | | | | C | U | | | | |
| | | | | | | | | | | | C | U | U | U | C | | |
- No nível avançado, o computador ou possui uma base de dados de muitas jogadas possíveis, ou vai aprendendo com os jogos que vai fazendo (perdendo ganhando ou empatando), ou vai testando todas as hipóteses, construindo uma árvore de `n` jogadas futuras possíveis e escolhendo a jogada que tem maior probabilidade de vitória.

O programa deve apresentar um menu com as seguintes opções:

Especificações para nível I (2 valores)

Especificações para nível II (3 valores)

Implementar as opções 9 e 10 do menu.

Implementar os modos de funcionamento 5 e 6 (*2 jogadores e computador contra computador*).

Especificações para nível III (4 valores)

Implementar o modo de funcionamento 4 (*avançado*).

Extras:

- Desenhar novos tabuleiros ao lado (direito) dos anteriores, enquanto couberem, sendo desenhados depois por baixo dos anteriores. Deste modo pode ver-se a sequência de jogadas.