Trabalho Prático
Segmentos

Melhorias de Nota (Época Especial)
Paradigmas de Programação I / Programação Funcional

LMCC/LCC + LESI/LEI 1º ano (2006/2007)

O trabalho é individual e deverá ser apresentado na semana da Época Especial de Exames (em data a combinar). O relatório do trabalho deve ser entegue 3 dias antes do dia da apresentação.

Relembre o conhecido jogo de 4-em-linha, onde 2 jogadores, com um tabuleiro $n \times m$ vertical, e jogando alternadamente, tentam obter uma linha de 4 peças iguais. Em cada jogada, o jogador pode colocar uma das suas peças numa coluna não cheia. A posição dessa coluna que é ocupada é a seguinte à última que foi ocupada. Uma linha pode ser obtida horizontalmente, verticalmente ou na diagonal.

O que propomos é que implemente em Haskell uma variante deste jogo a que vamos chamar **"Segmentos"**.
O objectivo do jogo é acumular o maior número de pontos possível. Os pontos são conseguidos quando se obtêm linhas de n peças iguais. Essas linhas podem ser verticais, horizontais ou na diagonal. A pontuação será diferente, consoante o número de peças em linha. A pontuação sugerida é a seguinte:

Segmentos	Pontos
2 peças	1
3 peças	3
4 peças	6
5 peças	10
6 peças	15
7 peças	21
...	...

Pode alterar estas cotações, se achar conveniente (ou mesmo, dar aos jogadores a possibilidade de estabelecerem a sua tabela de cotações para um dado jogo).
De seguida ilustra-se uma situação de jogo com um tabuleiro de 5×6, de que resulta a seguinte pontuação: Brancas = 4*1 + 1*3 = 7 pontos; Pretas = 3*1 + 2*3 = 9 pontos.

...	...	...	...	...	...
...	...	○	...	○	...
...	○	●	...	●	...
○	○	●	●	○	●
○	●	●	○	○	●

Na implementação este jogo, sugerimos que execute as seguintes tarefas.

Tarefas

1. Comece por definir tipos apropriados para representar tabuleiros e jogadas. Defina o tipo dos tabuleiros como instância de classes que julgue apropriadas (por exemplo, `Show`).
2. Defina funções que testem a validade de um tabuleiro e de uma jogada.
3. Defina uma função para efectuar uma jogada (que retornará a próxima configuração do tabuleiro.
4. Defina uma função que calcule o resultado de um determinado tabuleiro (o número de pontos de cada jogador).
5. Defina o programa que gere um jogo, aceitando alternadamente as jogadas de dois jogadores e terminando quando o tabuleiro fica cheio.
6. Defina um programa que implemente o comportamento de um jogador (que poderá se mais ou menos inteligente, dependendo do nível escolhido). Integre esse programa no definido atrás, permitindo desta forma a um jogador jogar com o seu programa.

Extras

Considere ainda algumas extensões ao problema proposto.

1. Estender o jogo para dar aos jogadores a possibilidade de escolher, previamente, a dimensão do tabuleiro e/ou a tabela de cotações.
2. Gerir uma tabela de resultados (histórico).
3. Dar a possibilidade de gravar e recuperar de ficheiro um jogo/tabela de resultados.
4. Usar uma biblioteca gráfica do Haskell para melhorar a interface com o(s) utilizadores.
5. Pode ainda implementar as variantes deste deste jogo que lhe pareçam interessantes. Por exemplo, terminar o jogo assim que um jogador complete uma linha de x peças (que deve ter uma cotação bastante alta).

Relatório

Deverá elaborar um relatório onde constem:

- Descrição do problema.
- Definição das estruturas de dados de suporte à solução implementada.
- Apresentação das principais funções e dos algoritmos usados.
- Listagem do programa (em anexo).