DCC019 - Linguagens de Programação
Departamento de Ciência da Computação DCC - UFJF
Professor: Leonardo Vieira dos Santos Reis e-mail:lvsreis@ice.ufjf.br
Trabalho Prático

1 Instruções Importantes

Nessa seção são apresentadas diversas informações relevantes referentes a entrega do trabalho e orientações a serem seguidas durante a implementação do mesmo. Leia atentamente antes de começá-lo.

1.1 Equipe de Desenvolvimento

O trabalho será desenvolvido individualmente.

1.2 Linguagem de Programação

O trabalho deverá ser desenvolvido na linguagem funcional Haskell.

1.3 Artefatos a Serem Entregues

Os artefatos a serem entregues são:

- código fonte do programa;
- documentação do trabalho em formato pdf.

Antes de enviar seu trabalho para avaliação, assegure-se que:

1. seu código executa no *ghci*. Programas com erros de sintaxe receberão nota zero;
2. todos os fontes a serem enviados têm, em comentário no início do arquivo, nome e matrícula do autor do trabalho;
3. arquivo de documentação tenha a identificação do autor do trabalho;
4. arquivo compactado com os artefatos estão devidamente identificados com nome e matrícula.

1.4 Critérios de Avaliação

A avaliação será feita mediante análise do código fonte, documentação e apresentação do trabalho (entrevista). Os seguintes fatores serão observados na avaliação do código fonte: corretude do programa, estrutura do código, redigibilidade e legibilidade. A corretude se refere à implementação correta de todas as funcionalidades especificadas, i.e., se o programa desenvolvido está funcionando corretamente e não apresenta erros. Os demais fatores avaliados no código fonte são referentes a organização e escrita do trabalho.

A documentação do código deve conter informações relevantes para auxiliar no entendimento do código fonte. Ressalta-se que a documentação não deve conter cópias do fonte – afinal o seu fonte é um dos artefatos entregue, mas deve apresentar as decisões de projetos tomadas, em especial as estruturas de dados para modelar e solucionar o problema.

O trabalho deverá ser apresentado ao professor da disciplina e, só serão avaliados após a realização da entrevista, i.e., trabalhos que não forem apresentado não terão nota. Na entrevista, o discente deverá elucidar, ao menos, como modelou e resolveu o problema. A entrevista também tem a finalidade de avaliar a confiabilidade e segurança do autor do código em explicar pontos relevantes do trabalho desenvolvido.

Assim, a entrevista influenciará na avaliação dos artefatos entregues. Portanto, a nota final será dada a partir da avaliação do conjunto do código fonte, documentação e entrevista. **É de responsabilidade do discente solicitar a marcação do dia e horário da entrevista com o professor da disciplina**.

Dias de Atraso	Nota
1	n*0.98
2	n*0.96
3	n*0.92
4	n*0.84
5	n*0.68
6	n*0.36
7	0

Atrasos serão penalizados por uma função exponencial de dias de atraso, i.e., será reduzido da nota um percentual referente a exponencial na base 2 dos dias de atraso. A tabela a seguir mostra a nota em função dos dias de atraso:

Observe que a partir do 7º dia de atraso seu trabalho não será mais avaliado.

2 Especificação Técnica do Trabalho

O objetivo desse trabalho é implementar uma versão do jogo campo minado. Campo minado é um jogo para um jogador cujo objetivo é revelar um campo de minas sem que nenhuma seja detonada. O jogo consistem em um tabuleiro com posições em que cada posição pode ou não conter uma mina. O jogador tem a opção de abrir uma posição ou marcá-la como uma mina. Caso o jogador abra uma posição com uma mina, ele perde e o jogo encerra (*game over!*). Caso contrário, a posição aberta mostrará a quantidade de minas na vizinhança. Por exemplo, suponha um jogo em um tabuleiro 4×4 com 5 minas. Inicialmente, todas as posições do tabuleiro estão fechadas (sem marcação de mina ou posições sem mina abertas). Na figura abaixo, o tabuleiro é representado usando o símbolo * para indicar as posições fechadas e índices nas linhas e letras nas colunas para designar cada posição. Desde modo, A1 se refere a posição do canto esquerdo inferior.

4	*	*	*	*
3	*	*	*	*
2	*	*	*	*
1	*	*	*	*
	A	B	C	D

Suponha que o jogo sorteou as minas nas posições A1, B3, C2, C4 e D2, as quais o jogador não tem conhecimento. Na sequência, o jogador executa uma ação de abrir a posição C1. Como essa posição não contém uma mina, o jogo irá mostrar a quantidade de minas na vizinhança, que nesse exemplo em particular será 1 correspondente a mina vizinha na posição C2. Como resultado, o tabuleiro ficará da seguinte forma:

4	*	*	*	*
3	*	*	*	*
2	*	*	*	*
1	*	*	1	*
	A	B	C	D

Em seguida, o jogador decide abrir a posição D1, tendo como resultado o tabuleiro:

4	*	*	*	*
3	*	*	*	*
2	*	*	*	*
1	*	*	1	1
	A	B	C	D

Após perceber que na vizinha de D1 só há uma mina, o jogador decide marcar a posição D2 como uma mina. No tabuleiro, a posição marcado como uma mina é exibida com a letra **B** e, essa posição não será mais válida para ser aberta. Após essa última ação, o tabuleiro ficará da seguinte forma:

4	*	*	*	*
3	*	*	*	*
2	*	*	*	B
1	*	*	1	1
	A	B	C	D

Nesse trabalho deverá ser implementado um jogo que tenha o funcionamento como descrito anteriormente. O usuário deverá iniciar o programa passando como parâmetros o tamanho do tabuleiro e o número de bombas. O programa irá sortear as posições para cada uma das bombas e irá exibir o tabuleiro com todas as posições "fechadas" (com asterisco). Nesse momento, o programa entrará em modo de espera, aguardando um comando do usuário. Os possíveis comandos serão:

Comando	**Descrição**	**Exemplos**
posição	posição a ser aberta	A1, D4, B3
+ *posição*	indicação da posição a ser marcada como mina	+D2, + C4
- *posição*	indicação para desmarcar uma posição que está marcada como mina	-D2, -C4, -A1

Após cada comando do usuário, o programa irá exibir na tela o novo estado do tabuleiro. Caso seja informado algum comando inválido, a ação do programa será exibir o estado atual do tabuleiro sem modificações. As possíveis situações de informações errôneas digitadas pelo usuário são:

- digitar um comando que não siga um das três opções estabelecidas
- informar uma posição inválida em qualquer uma das ações
- tentar abrir uma posição que já foi aberta ou marcada
- tentar marcar uma posição que já está aberta ou marcada
- tentar desmarcar uma posição que não está marcada
- tentar marcar mais posições que o total de bombas

Na contagem de bombas da vizinhança, deverá ser considerado como vizinho somente as células que estão imediatamente acima, abaixo e aos lados. As células nas diagonais não serão consideradas vizinhas. Ou seja, uma célula terá, no máximo, 4 vizinhos.

O jogo e o programa será encerrado somente se uma das duas situações ocorrer: o jogador tentar abrir uma posição que contém uma bomba. Nesse caso, o jogo irá mostrar o tabuleiro com todas as células abertas e exibir as posições das bombas e emitirá a mensagem "Game Over! Você foi explodido!". Ou, o jogador conseguirá abrir todas as posições e marcar todas as bombas do tabuleiro. Nessa hipótese, o jogo emitirá a mensagem "Parabéns! Você venceu!".

Adicionalmente, assuma que o número máximo que o usuário poderá escolher de bombas será a metade da quantidade de posições do campo. Assim, no exemplo ilustrado de um campo 4×4, o usuário poderá escolher entre 1 e 8 minas. Caso seja inserido um valor fora desse intervalo, o programa ajustará para o extremo mais próximo.

3 Entrega do Trabalho

A data da entrega do trabalho será até o dia **07 de julho de 2019**. A entrevista deverá ser realizada juntamente com o professor da disciplina até o dia **15 de julho de 2019.**

A entrega será realizada via e-mail para o endereço *lvsreis@ice.ufjf.br*.

4 Bônus

Os trabalhos que implementarem também uma versão gráfica do jogo campo minado será bonificados com a pontuação extra de 8 pontos. Ressalta-se que a versão modo tempo também deve existir e uma opção, ou função, deverá ser fornecida para que o usuário possa escolher em qual dos modos irá jogar. Aos interessados em implementar um modo gráfico, recomendo a biblioteca *gtk2hs*.