

Campo Minado

Instruções

O trabalho é em dupla (não pode ser individual) e pode ser submetido via SIGAA (arquivo compactado com todo o código-fonte) ou via GitHub Classroom.

O Jogo

O jogo consiste em uma área matricial de n linhas por m colunas com um total de $n*m$ células e k minas. Inicialmente, as células estão ocultas, ou seja, não é possível determinar se ela possui uma mina ou não. O objetivo do jogo é revelar todas as células sem revelar uma das minas. A ação de revelar uma célula pode ser definida conforme as regras abaixo:

1. Uma mina é revelada: nesse caso, jogo encerra com a derrota do usuário;
2. Um número é revelado: o valor indica a quantidade de minas adjacentes considerando as 8 células ao redor de uma posição (vertical, horizontal e diagonais);
3. Uma posição vazia é revelada: nesse caso, o jogo aplica a ação revelar sobre todas as células adjacentes a essa posição, visto que ela não possui minas adjacentes.

O usuário vence o jogo ao revelar todas as células que não contêm minas. Por fim, a dificuldade do jogo irá determinar as dimensões da área do jogo e a quantidade de minas.

Interface do Jogo

A interface do jogo deve apresentar as seguintes ações para o usuário:

1. Ação de revelar uma célula;
2. Ação de verificar o tempo decorrido desde a primeira ação de revelar;
3. Ação de solicitar ajuda ao jogo.

Além disso, usuário deve conseguir visualizar toda a área do jogo identificando quais células estão ocultas, são vazias ou números, como também a quantidade de minas presentes. Também é necessário a apresentação de um feedback adequado ao usuário ao fim da partida indicando a vitória ou derrota.

A interface será textual (executada no terminal) e você é livre para determinar como usuário deve executar essas ações e como a área do jogo será apresentada. Porém, esse item também será avaliado de modo que a interface deve ser amigável ao usuário. Ainda neste contexto, soluções que utilizem uma interface gráfica serão bonificadas com 1 ponto extra considerando o limite da nota da unidade (não é possível obter uma nota maior que 10). Eu sugiro a ferramenta IUP¹ para os iniciantes, no entanto, vocês são livres para escolher outra ferramenta para o desenvolvimento de interfaces.

Por fim, a última ação "solicitar ajuda ao jogo" deve indicar uma posição do jogo que provavelmente não contém uma mina. Isso será feito através da implementação de uma função que receba o estado atual do jogo (sem as informações das minas) e retorna uma posição que possui baixa chances de ser uma mina. A próxima seção explica alguns detalhes de implementação incluindo a especificação dessa função de inteligência do jogo.

¹ <https://www.tecgraf.puc-rio.br/iup/>

Detalhes de Implementação

Além das regras descritas, o programa deve registrar o tempo das partidas, isto é, tempo decorrido da primeira ação de revelar até o fim do jogo. Esse tempo deve ser exibido ao fim da partida ou quando solicitado pelo usuário durante a partida. Para isso, utilize a biblioteca `time.h`² da linguagem C.

O programa terá um único nível de dificuldade possuindo uma área de 10 linhas por 20 colunas e com 40 minas. Neste quesito, são aceitas algumas variações, por exemplo, a primeira ação de revelar sempre exibe uma posição vazia ou então sempre exibe um número. Caso essas variações sejam implementadas de forma adequada, isto é, sem causar impactos no funcionamento do jogo, será atribuída uma bonificação de meio ponto por cada variação.

O programa ainda deve ter um modo de jogo autônomo, isto é, um modo em que o jogo execute sozinho suas ações revelando as posições do campo até o fim do jogo. Esse modo deve usar uma função que receba o estado atual do jogo (sem as informações da mina) e retorne uma posição que possui baixas chances de ser uma mina. Assim, ao selecionar essa opção, o jogo deve escolher uma posição e revelá-la iterativamente até que o final, ou seja, vitória ou derrota.

O modo autônomo não precisa vencer todas as vezes. No entanto, vocês devem propor uma estratégia que faça algum sentido para as regras do jogo e então implementá-la. Por exemplo, uma estratégia bastante simples é a escolha aleatória de uma posição a cada passo, porém, é esperada proposta de uma estratégia mais complexa. Este item será avaliado verificando se a implementação corresponde a estratégia proposta.

Essas ações de execução do jogo em modo autônomo ou a modificação das configurações das variações podem ser acessadas pelo usuário através de um menu antes do início da partida.

² Site com uma breve explicação de como usar a biblioteca `time.h` para registrar a passagem de tempo: <https://www.techiedelight.com/pt/find-execution-time-c-program/>

Por fim, será necessário o uso de geração de valores aleatórios para geração das minas e possivelmente para o desenvolvimento do modo autônomo. Para isso, utilize a função *rand* da biblioteca *stdlib.c*³.

Avaliação

O projeto será avaliado através de uma entrevista quanto a aplicação adequada das técnicas de programação vistas em aula, quanto à organização do código-fonte e quanto à implementação das funcionalidades do jogo.

Quanto à aplicação das técnicas de programação, vocês devem empregar ao menos 5 dos tópicos abaixo de modo adequado ao problema. Este item será avaliado considerando a relevância do tópico para solução do problema e em como ele foi aplicado no desenvolvimento do código:

- Estruturas, enumerações e typedef;
- Alocação dinâmica de memória;
- Passagem por referência;
- Recursão;
- Manipulação de arquivos;
- Modularização do código.

³ Links com uma breve descrição da função *rand* para geração de valores aleatórios: <https://www.cprogressivo.net/2013/03/Como-gerar-numeros-aleatorios-em-C-com-a-rand-srand-e-seed.html>