Trabalho 01

Descrição

Projetar e implementar um programa em C/C++ para jogar uma versão modo texto do jogo de tabuleiro **Mastermind**. Você não deve utilizar outras bibliotecas, além daquelas fornecidas pela própria linguagem. Além disso, sua solução deve ser estruturada (não utilizar classes).

Informações sobre o jogo podem ser obtidas em:

- http://en.wikipedia.org/wiki/Mastermind_(board_game)
- http://www.gomes-mota.nome.pt/joao/www/jgcores/regras_mm.html

A versão do jogo que você irá implementar deve ter as seguintes propriedades:

- O computador irá gerar o código secreto aleatoriamente.
- O jogador tentará adivinhar o código secreto.
- O jogador tem 12 tentativas para adivinhar o código.
- Se o jogador não adivinhar o código corretamente em 12 ou menos tentativas, ele perde o jogo.
- O jogo consiste em 4 pinos coloridos.
- As cores válidas para os pinos são azul, verde, laranja, amarelo, roxo e vermelho.
- Os resultados de uma tentativa são apresentados com pinos pretos e brancos (fornecem feedback para o jogador).
- Um pino preto indica que um dos pinos colocados pelo jogador tem a cor correta e está na posição correta.
- Um pino branco indica que um dos pinos colocados pelo jogador tem a cor correta, mas está na posição errada.
- Cada pino da tentativa irá gerar somente 1 pino preto, 1 pino branco ou nenhum pino. Ou seja, para cada pino colocado, haverá somente 1 pino (ou nenhum) de feedback.
- A ordem do feedback não dá informações sobre quais pinos da tentativa geraram os pinos de feedback.
- Letras em caixa alta devem ser usadas para indicar as cores (B-azul, G-verde, O-laranja, Y-amarelo, P-roxo e R-vermelho). Foi utilizada a primeira letra da cor na língua inglesa para garantir a diferenciação.
- As tentativas dos usuários (entradas) devem ser verificadas para assegurar que a quantidade de pinos está correta e que as cores informadas são válidas.

Trabalho 01

Parte da avaliação será determinada pelas suas decisões de design, incluindo como você quebrou o problema em funções e arquivos, qualidade da abstração e alta coesão. Programas que não estiverem funcionando serão descontados inicialmente de 2 pontos. Os demais pontos serão observados em relação ao grau de atendimento dos requisitos, qualidade do código, qualidade dos comentários de documentação e organização.

Observe que a função main() é apenas a porta de entrada para sua implementação e não deve conter muita lógica de programa. Você deve fornecer uma descrição dos testes que você executou. Os testes implementados devem ser entregues na sua submissão. Separe os arquivos de teste (fonte e header) dos arquivos do programa.

Dicas

1. Este trabalho não é apenas de implementação. Você deve utilizar algum tempo para pensar na solução e como o programa será organizado em funções e arquivos. Não tente escrever o código diretamente.
2. Depois de decidir quais funções você precisa, implemente uma de cada vez (considerando as mais simples primeiro). Realize testes para assegurar que elas estão funcionando antes de prosseguir. Eventualmente, você perceberá a necessidade de ajustes no seu design e não será necessário modicar todo o código. Siga a ideia: projete um pouco, codifique um pouco, teste um pouco. Repita.
3. Escreva um código legível e utilize comentários (conforme orientações nas aulas) para facilitar o entendimento. Lembre-se que o código deve ser de todos.
4. Fique livre para usar estruturas (struct).
5. Funções não devem ser muito longas e devem ter um objetivo claro.
6. Não utilize variáveis globais.