Fundamentos de Programação

Trabalho 1 - Jogo da Velha

Trabalho individual
Entrega: até 23:59:59 de 31/05/2018

Jogo da velha

O objetivo deste trabalho é programar um jogo popular e muito simples: o jogo da velha!

Neste jogo, temos um grid (ou tabuleiro) 3×3 onde uma pessoa joga contra outra, uma utilizando 'X' (xis) e outra utilizando 'O' (círculo, bola, ou simplesmente a letra 'O'). O objetivo dos jogadores é colocar três símbolos iguais em sequência no tabuleiro. Esta sequência pode ser feita em uma linha, em uma coluna, ou em uma das diagonais. Caso um jogador monte esta sequência, este jogador é declarado vencedor. Caso todas as posições sejam preenchidas e não haja vencedor, dizemos que "deu velha".

Exemplos de caso para o jogo da velha:

o		
	o	
x	x	o

vitória do 'O'

o	o	x
o	x	
x	o	x

vitória do 'X'

o	x	o
x	o	o
x	o	x

deu velha

Portanto, você deve implementar um jogo da velha que permita a disputa entre dois jogadores, mostrando o tabuleiro a cada jogada. Os jogadores devem inserir uma posição onde querem colocar sua peça, e o jogo deve efetuar a jogada. Se um jogador ganhar, o jogo deve imprimir uma mensagem de vitória identificando o jogador vencedor, e o jogo se encerra. Se o tabuleiro foi totalmente preenchido e nenhum jogador venceu, o jogo deve imprimir uma mensagem de empate (dizendo que "deu velha"), e encerrar o jogo.

Pontos que serão avaliados

(a) (3.0 pontos) Imprimir a configuração atual do tabuleiro

1. (1.0) a impressão deve ser feita ao início do jogo (tabuleiro sem nada preenchido) e a cada nova jogada (apresentando o novo tabuleiro), inclusive quando o jogo terminar (apresentando o tabuleiro final, seja por vitória ou empate)
2. (1.0) esta impressão deve separar caracteres de linhas e colunas diferentes (por exemplo, em vez de imprimir uma linha do tabuleiro como "xox", teríamos algo como "x|o|x"
   - dica: olhe o fim deste arquivo para alguns exemplos
3. (1.0) posições ainda não preenchidas do tabuleiro devem ser representados por um espaço na hora da impressão, não podem ser representados por uma letra (como 'V')

(b) (3.0 pontos) Detectar casos de vitória e empate, e imprimir a mensagem correspondente

1. (1.0) se um jogador é vitorioso, deve-se imprimir uma mensagem de vitória identificando o jogador que ganhou
   - a identificação do jogador fica a seu critério, mas deve ser algo intuitivo: por exemplo, pode-se identificar o jogador pelo seu símbolo, ou por 1 e 2
   - obs: em caso de vitória, não é necessário identificar/imprimir qual foi o caso de vitória (ou seja, qual linha/coluna/diagonal foi preenchida); basta identificar o jogador vitorioso
2. (1.0) se houve empate, deve-se imprimir uma mensagem de empate
3. (1.0) se foi detectado um caso de vitória ou empate, o jogo acabou, logo não se deve imprimir mensagem solicitando uma nova jogada

(c) (3.0 pontos) Receber a posição na qual o jogador quer jogar, e efetuar a jogada

1. (1.0) a posição deve ser algo intuitivo
   - exemplos: dois valores numéricos (representando linha e coluna do tabuleiro), ou um valor de 0 a 8, ou um valor numérico de 1 a 9 (onde as posições são de acordo com o teclado numérico de um teclado padrão)
2. (1.0) deve-se evitar jogadas inválidas
   - exemplo 1: tentar jogar em uma posição já preenchida
   - exemplo 2: tentar jogar em uma posição inexistente (por exemplo, linha 0, coluna 20)
3. (1.0) no caso de uma jogada inválida, deve-se imprimir uma mensagem de jogada inválida e deve-se solicitar novamente a jogada para o mesmo jogador

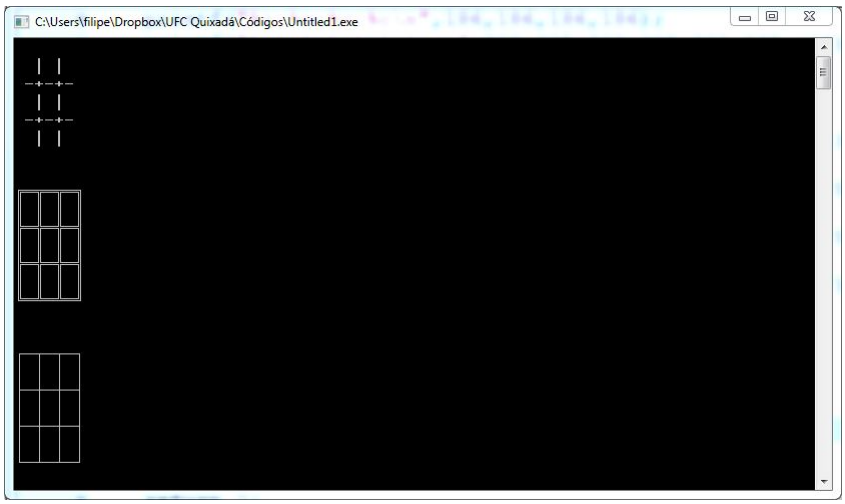
(d) (1.0 pontos) A cada turno, imprimir uma mensagem solicitando a jogada de um jogador diferente, revezando os jogadores a cada turno

- esta mensagem deve identificar qual jogador jogará neste turno
- sugestão: indique também como a jogada deve ser feita
  - apresentamos alguns exemplos no item (c2)
  - exemplo: "Vez do jogador 1. Insira um numero entre 0 e 8"

Comentários adicionais

- a nota será dada avaliando o cumprimento ou não de cada tópico, tratando cada um individualmente como feito ou não feito
  - ou seja, sem casos intermediários
  - exemplo: no item (c3) se o jogo evita que você jogue em uma posição inexistente (exemplo 2), mas permite que uma pessoa jogue em uma posição já preenchida (exemplo 1), a nota para o ponto (c3) será zero
- ao fim do jogo, não é necessário perguntar se a pessoa deseja jogar novamente
- também não é necessário limpar a tela para mostrar a nova configuração do tabuleiro
  - apesar disto, se você desejar, pode utilizar o comando system("@cls||clear") da biblioteca "stdlib.h"
  - obs: tenho dúvidas sobre o funcionamento deste comando no Linux, mas creio que funciona em ambos Windows e Linux
- relembrando: a entrega vai até meia-noite de quinta-feira, 31/05/2018
  - a forma de entrega ainda está em aberto e será divulgada por mensagem, mas provavelmente será via Moodle (ou e-mail, caso o Moodle esteja fora do ar)
  - você deve submeter o seu código como se fosse um exercício normal

Dicas de como imprimir o tabuleiro utilizando caracteres simples e especiais (caracteres da tabela ASCII):




```
//maneira 1
printf("   | | \n");
printf(" -+-+-\n");
printf("   | | \n");
printf(" -+-+-\n");
printf("   | | \n");

//maneira 2
printf("%c%c%c%c%c%c%c\n",201,205,203,205,203,205,187);
printf("%c %c %c %c\n",186,186,186,186);
printf("%c%c%c%c%c%c%c\n",204,205,206,205,206,205,185);
printf("%c %c %c %c\n",186,186,186,186);
printf("%c%c%c%c%c%c%c\n",204,205,206,205,206,205,185);
printf("%c %c %c %c\n",186,186,186,186);
printf("%c%c%c%c%c%c%c\n",200,205,202,205,202,205,188);

//maneira 3
printf("%c%c%c%c%c%c%c\n",218,196,194,196,194,196,191);
printf("%c %c %c %c\n",179,179,179,179);
printf("%c%c%c%c%c%c%c\n",195,196,197,196,197,196,180);
printf("%c %c %c %c\n",179,179,179,179);
printf("%c%c%c%c%c%c%c\n",195,196,197,196,197,196,180);
printf("%c %c %c %c\n",179,179,179,179);
printf("%c%c%c%c%c%c%c\n",192,196,193,196,193,196,217);
```