

SME0230 - Introdução à Programação de Computadores

Primeiro semestre de 2011

Professora: Marina Andretta (andretta@icmc.usp.br)

Estagiária PAE: Pâmela Michele Cândida Cortez (pamela@icmc.usp.br)

Monitor: Matheus Doretto Compri (matheusdc@grad.icmc.usp.br)

Trabalho: Jogo Hex - segunda parte

Data: 10/05/2011.

Data máxima de entrega: 21/06/2011, até às 23h59min. A cada dia de atraso, será descontado 20% da nota recebida.

Grupos: Os trabalhos poderão ser feitos em grupos de até 2 pessoas (devem ser mantidos os mesmos grupos da primeira parte do trabalho).

Forma de entrega: O trabalho deverá ser entregue por e-mail para andretta@icmc.usp.br, com cópia para pamela@icmc.usp.br e matheusdc@grad.icmc.usp.br, e o título do e-mail deverá ser IPC2011_T2. Ele deverá estar escrito em linguagem C e o nome do arquivo deverá ser

<número usp 1>-<número usp 2>.c,

com *<número usp i>* o número usp de cada componente do grupo. Apenas um componente do grupo deverá enviar o trabalho.

Enunciado: Jogo Hex

O Jogo Hex possui um tabuleiro com casas hexagonais (ou seja, cada casa possui outras 6 casas adjacentes) com n linhas e n colunas. Neste trabalho usaremos um tabuleiro com 8 linhas e 8 colunas. Dois jogadores com pedras de cores distintas (neste trabalho, símbolos x e o) jogam alternadamente e cada jogada consiste em um jogador colocar uma de suas pedras em uma posição vazia do tabuleiro. O objetivo do jogo é que as pedras formem um caminho ligando um lado do tabuleiro ao lado oposto. Antes do início do jogo devem ser definidos os lados a serem conectados por cada um dos dois jogadores.

O objetivo deste trabalho é implementar em linguagem C duas versões

do Jogo Hex. Na primeira versão, dois usuários jogarão um contra o outro. Na segunda versão, um usuário jogará contra o computador.

Primeira versão do Jogo Hex

Na primeira versão do Jogo Hex, dois usuários irão jogar um contra o outro.

Um usuário (**Jogador 1**) irá digitar a posição na qual quer que seu caractere (**x**) seja inserido no tabuleiro. Quando ela é digitada, é necessário verificar se a posição escolhida é válida (ou seja, se não há outro caractere no mesmo lugar) e se o jogo não acabou. Se a posição escolhida pelo **Jogador 1** não for válida, ele deve digitar outra.

Caso o jogo não tenha acabado na jogada do **Jogador 1**, o **Jogador 2** deve digitar a posição na qual deseja inserir seu caractere (**o**). As mesmas verificações feitas para o **Jogador 1** devem ser feitas para o **Jogador 2**.

Sempre que alguma jogada válida for feita, deve-se imprimir na tela o tabuleiro atualizado.

O processo é repetido até que o jogo acabe com a vitória de algum dos jogadores. Uma mensagem deve ser impressa na tela informando quem ganhou o jogo.

Segunda versão do Jogo Hex

Na segunda versão do Jogo Hex, um usuário **Jogador** jogará contra o **Computador**.

Do mesmo modo como feito na primeira versão do jogo, o **Computador** deverá escolher a posição na qual quer que seu caractere (**x**) seja inserido no tabuleiro. Quando esta posição é escolhida, é necessário verificar se ela é válida (ou seja, se não há outro caractere no mesmo lugar) e se o jogo não acabou. Se a posição escolhida pelo **Computador** não for válida, ele deve escolher outra.

Caso o jogo não tenha acabado na jogada do **Computador**, o **Jogador** deverá digitar uma posição na qual deseja inserir seu caractere (**o**). A escolha deve ser feita de modo que a posição seja válida. Digitada a posição, deve-se verificar se o jogo não acabou.

Sempre que alguma jogada válida for feita, deve-se imprimir na tela o tabuleiro atualizado.

O processo é repetido até que o jogo acabe com a vitória de algum dos jogadores. Uma mensagem deve ser impressa na tela informando quem ganhou o jogo.

Especificações

Neste trabalho, deve-se implementar, em linguagem C, um algoritmo para descrever a primeira versão do Jogo Hex e um algoritmo para descrever a segunda versão. Note que, no caso da segunda versão, o modo como é feita a escolha de cada jogada do **Computador** também deve ser implementada.

O tabuleiro deverá ser representado por uma matriz de caracteres 8x8. Cada posição desta matriz deverá ser preenchida com **x**, **o** ou **v** (caso a posição esteja vazia). Os índices das linhas e colunas deverão estar entre 0 e 7. Dada uma casa na posição (i, j) , seus seis vizinhos estão nas posições $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$, $(i, j + 1)$, $(i - 1, j + 1)$ e $(i + 1, j - 1)$.

O jogador que joga com o caractere **x** deve ligar a primeira e a última linha. O jogador que joga com o caractere **o** deve ligar a primeira e a última coluna.

Haverá duas funções fornecidas a todos os alunos, que deverão ser usadas na implementação deste trabalho. Uma função deverá receber como parâmetro o tabuleiro **tab** e o jogador **jog** (1 ou 2) e retornará o valor 1, se o jogador ganhou, e 0, caso contrário. O cabeçalho desta função será o seguinte:

```
int ganhou(char tab[8][8], int jog);
```

A outra função deverá receber a matriz de caracteres que representa o tabuleiro e imprimirá o tabuleiro de Hex correspondente. O cabeçalho desta função será o seguinte:

```
int desenhaTab(char tab[8][8]);
```