



**UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO**

MC404 – ORG. BÁSICA DE COMPUTADORES E LING. DE MONTAGEM
1º Semestre de 2007
Prof. Ricardo Pannain

RELATÓRIO DO TRABALHO PRÁTICO

Alex Oliveira Cremonezi (058589)
Haraldo Sérgio Albergaria Pereira Filho (003059)
Jônatas Botelho da Silva (033546)

1 – INTRODUÇÃO

1.1 - Trabalho

O trabalho foi desenvolvido pelos alunos durante o primeiro semestre de 2007, levando o prazo de 20 dias para ser concluído.

Para montagem/linkagem do programa foram utilizados os executáveis disponibilizados em: <http://www.ic.unicamp.br/~pannain/mc404/aulas/montador/>.

O trabalho foi testado nas seguintes condições:

- Na residência dos alunos, em computadores rodando Windows XP Home Edition utilizando o programa com0com, disponível no seguinte endereço: <http://sourceforge.net/projects/com0com/>
- Nos laboratórios do IC-3, em computadores rodando Windows 2000 Professional conectados pela porta serial.

Nas duas condições o programa funcionou perfeitamente, não apresentando nenhum problema.

1.2 - Jogo

O jogo escolhido para desenvolvimento foi o jogo da velha, conhecido também como "Jogo do Galo" e "Tic Tac Toe". Seu nome teria se originado na Inglaterra, quando nos finais de tarde, mulheres se reuniam para conversar e bordar. As mulheres idosas, por não terem mais condições de bordar em razão da fraqueza de suas vistas, jogavam este jogo simples, que passou a ser conhecido como o da "velha". Mas sua origem seria ainda mais antiga. Fala-se em tabuleiros escavados na rocha de templos do antigo Egito que teriam sido feitos por escravos há 3.500 anos.

Regras:

- O [tabuleiro](#) é uma [matriz](#) de três linhas por três colunas.
- Dois [jogadores](#) escolhem uma marcação cada, um [círculo](#) (O) e um [xis](#) (X).
- Os jogadores jogam alternadamente, uma marcação por vez, numa lacuna que esteja vazia.
- O objetivo é conseguir três círculos ou três xis em linha, quer horizontal, vertical ou diagonal e ao mesmo tempo, quando possível, impedir o adversário de ganhar na próxima jogada.

2 – ESPECIFICAÇÃO DO JOGO

Este é um Jogo da Velha para ser jogado entre dois computadores conectados pela porta serial. O funcionamento do jogo é muito simples e intuitivo, usando-se o mouse para efetuar as jogadas.

3 – MANUAL

3.1 – Instalação

- Extraia para o Computador 1 o arquivo "Jogo da Velha.zip", que criará a pasta "Jogo da Velha".
- Entre na pasta e execute CONFIGURA, escolhendo as opções de configuração do programa.
- Repita o mesmo procedimento para o Computador 2.

IMPORTANTE! : Escolha opções diferentes no Computador 1 e no Computador 2 para os parâmetros (X / O) e (C / E), caso contrário o jogo não funcionará corretamente.

OBS: Execute CONFIGURA novamente se desejar alterar as configurações.

3.2 – Como Jogar

Após instalar o jogo nos dois computadores, dê dois cliques no ícone "Jogo da Velha" em ambos. Caso tenha sido escolhido exibir a abertura, espere a mesma terminar. Quando for mostrado o tabuleiro é só começar a jogar, seguindo as instruções dadas na tela. Para sair do jogo a qualquer momento, pressione ESC.

4 – DESENVOLVIMENTO

4.1 – Animação de Abertura

Para a abertura do jogo foi utilizada a técnica de animação que consiste em se escrever algo na tela para logo em seguida apagá-la e escrevê-la rapidamente em outra posição. Foram desenvolvidos para isso procedimentos nos arquivos "ABERT.ASM" e "TIMER.ASM" , além de macros no arquivo "MACROS.ASM".

Primeiramente, no arquivo "ABERT.ASM" são empilhados os registradores e é selecionado o modo de vídeo 12h. São desenhadas as duas linhas horizontais, com 45 pixels cada, e as duas linhas verticais com 46 pixels, utilizando um ciclo da interrupção 8h (aproximadamente 0,08 seg) como intervalo entre cada pixel. Para calcular esse intervalo é utilizada a macro "ESPERA" que utiliza o procedimento "TIMER" com a quantidade de ciclos desejada.

Em seguida, o cursor é movido com a macro MOV_CURSOR, para várias posições do tabuleiro, onde são escritos os caracteres X, em vermelho, e O, em verde, com intervalo de 2 segundos (26 ciclos da int8h).

Em seguida é apresentado em tela cada letra da palavra jogo caindo, até chegar a altura do meio do tabuleiro. Para isso é utilizada a seguinte lógica:

A cada 2 ciclos da int 8h (aproximadamente 0,16 segs) é escrito o caracter e é feita a verificação se o mesmo está na linha 10, caso não esteja, imprime o caracter na próxima linha, caso esteja, pula uma coluna, vai para a próxima letra da palavra e verifica se chegou ao fim da palavra, se não, imprime nova letra. Após isso, fazemos uma letra O, cor verde, cair até chegar ao fim da tela.

Depois, a letra D é movida da esquerda para a direita, e a letra A é movida da direita para esquerda, até que as duas se choquem no centro da tela. Para isso é alternada a movimentação de "D" para a direita e "A" para a esquerda, sempre verificando se já chegaram ao centro da tela.

Por fim, é mostrado em tela a palavra "VELHA" subindo até chegar a linha 14, logo embaixo da palavra "DA". Para isso é utilizada a mesma lógica utilizada para a palavra "JOGO", porém com a palavra inteira de uma vez. É aguardado o período de 5 segundos (64 ciclos da int 08h) para iniciar o jogo.

4.2 – Bitmaps para “X”, “O” e “Velha”

Para a exibição dos bitmaps de “X” e “O” foram desenvolvidos procedimentos no arquivo "BITMAPS.ASM" e para o do bitmap da imagem da “Velha” no arquivo "VELHA.ASM".

Para os bitmaps de “X” e “O” criamos os arquivos "X.BMP" e "O.BMP", declarados no arquivo "DATA_SEG.ASM". Para os dois arquivos é utilizada a mesma lógica no procedimento "carrega_X_O": Primeiramente utiliza-se a função 3Dh da interrupção 21h para abrirmos os arquivos existentes e salvamos os handles, ou o código de erro, contido em AX, em uma variável na memória. Em seguida utilizamos a função 3Dh para lermos os bytes do arquivo, movendo o handle para BX, a quantidade de bytes a serem lidos para CX, e carregamos o desenho em memória na variável X_BUF ou O_BUF. Por fim, utilizamos a função 3Eh da int 21h para fecharmos o arquivo.

Para desenhar os bitmaps na tela foram desenvolvidos "desenha_X" e "desenha_O". Esses procedimentos funcionam da seguinte maneira: Primeiro é desabilitado o mouse, são salvos os registradores, zera-se SI e são impressos os valores de X_BUF ou O_BUF, previamente carregados, linha por linha, utilizando para isso a função 0Ch da int 10h. Dependendo do jogador, ele envia ou não a posição para o outro computador através da porta serial.

Para o bitmap da imagem da velha, criamos o arquivo "VELHA.BMP", declarado no arquivo "DATA_SEG.ASM". Como o arquivo da velha é muito grande e não cabia em um segmento de memória, a lógica para a exibição desse arquivo, através do procedimento "exibe_velha", é um pouco diferente: cada linha da imagem é lida do arquivo, armazenada na memória na variável "BUFFER" e é escrita logo em seguida na tela, ou seja, é lida e impressa uma linha de imagem por vez.

Todos os arquivos de bitmap foram criados da seguinte maneira: Desenhou-se o bitmap utilizando o aplicativo "Paintbrush", abriu-se o arquivo gerado em um editor hexadecimal e apagou-se os primeiros bytes referentes ao cabeçalho gerado pelo aplicativo.

4.3 – Comunicação Serial

Para a comunicação serial foram desenvolvidos os procedimentos "RECEBER" e "ENVIAR" no arquivo "SERIAL.ASM". Para a inicialização da serial, foi utilizado o mesmo procedimento passado pelo professor (no arquivo

“TALK.ASM”), colocado no procedimento “INICIALIZAR_SERIAL” do mesmo arquivo.

Esses procedimentos funcionam da seguinte maneira:

RECEBER: Primeiramente verifica-se o estado do registrador para saber se há um carácter a receber. Depois é lido o carácter do anel, para verificar onde o outro jogador realizou sua jogada. O registrador AL recebe um valor para cada casa onde foi efetuada a jogada. Portanto, dependendo do valor de AL, as variáveis “POS_COLUNA”, “POS_COLUNA_FIM”, “POS_LINHA” e “POS_LINHA_FIM” recebem os valores correspondentes do campo da jogada. Após isso, dependendo do jogador é chamado o procedimento correspondente para desenhar o bitmap da jogada efetuada (“desenha_X” ou “desenha_O”).

ENVIAR: Envia o carácter AL (com o valor da casa correspondente), para o outro computador e faz as verificações de término de jogo. Caso o jogo tenha terminado, monitora o teclado, verificando se foi pressionada a tecla ESC, que encerra e sai do jogo ou a tecla F1, que reinicia o jogo.

4.4 – Mouse

Para a utilização do mouse foram desenvolvidos os procedimentos “LER_MOUSE” e “ANALISA_CLIQUE” no arquivo “MOUSE.ASM”, além dos macros “HABILITA_MOUSE”, “DESABILITA_MOUSE” e “MONITORA_MOUSE” no arquivo “MACROS.ASM”.

As macros “HABILITA_MOUSE” e “DESABILITA_MOUSE” utilizam respectivamente as funções 01h e 02h da int 33h.

A macro “MONITORA_MOUSE” utiliza a função 06 h da int 33h, que retorna em CX a coordenada horizontal do clique e em DX a coordenada vertical.

4.5 – Jogo

Para a lógica do jogo foram desenvolvidos vários procedimentos no arquivo “JOGO.ASM”, além de macros no arquivo “MACROS.ASM”.

Basicamente, o jogo segue a seguinte lógica:

Inicializa placar: Escreve 0 em todas as posições na memória das variáveis PLACAR, GANHADOR e DEU_VELHA.

Seleciona o modo de vídeo: É utilizada a função 0h da int 10h para selecionar o modo de vídeo 12h: 640x480 pixels e 16 cores VGA. Em seguida o cursor é movido para a linha 0, coluna 0 da tela, com a função 02h da int 10h, e é chamado o procedimento para desenhar o tabuleiro de jogo.

Carrega os bitmaps de X e O: ver seção 4.2

Inicializa a porta serial: ver seção 4.3.

Desenha tabuleiro: Para isso é utilizada a função 0ch, da int 10h, para desenhar pixels, são utilizados 300 pixels para cada linha ou coluna do tabuleiro, chamando em seguida o procedimento para início da leitura com o mouse.

Lê o mouse: Primeiramente, verifica se é a vez do jogador utilizando a variável "VEZ_DO_JOGADOR", tendo valor 0 quando é a sua vez e 1 quando é a vez do outro. Em seguida o procedimento para lidar com o mouse é dividido em várias partes, são elas:

LER: Monitora se uma tecla de comando é acionada, por exemplo "ESC". Se for ESC, encerra o programa através da macro "ENCERRA_PROGRAMA" e se não for, verifica novamente se é vez do jogador ou não. Se não for aguarda, se for, habilita o mouse, através da macro "HABILITA_MOUSE" e monitora quantos cliques houve e em qual local da tela através da macro "MONITORA_MOUSE".

ESPERAR_OUTRO_JOGAR: Recebe a jogada via porta serial do outro jogador. Para isso chama o procedimento "RECEBER" do arquivo "SERIAL.ASM". Em seguida são feitas as verificações para saber se algum jogador ganhou ou se deu velha, através dos procedimentos "VERIFICA_PERDEDOR" e "VERIFICA_VELHA" de "JOGO.ASM". O primeiro verifica todas as linhas, colunas e diagonais, verificando se existem três caracteres iguais juntos. Caso encontre, exibe mensagem de ganhador ou perdedor, dependendo do jogador, também riscando a jogada ganhadora. O segundo verifica se todas as casas estão cheias e se nenhum jogador ganhou. Caso isso seja verdadeiro, chama o procedimento "exibe_velha" do arquivo "VELHA.ASM".

ANALISA: Chama o procedimento ANALISA_CLIQUE, do arquivo "MOUSE.ASM", que verifica se a jogada foi feita fora do tabuleiro ou se a casa clicada está ocupada ou não, marcando a casa como ocupada caso o clique seja válido, também chamando o procedimento para desenhar o bitmap referente ao caractere clicado. Caso seja o jogador ativo, envia as coordenadas da jogada realizada para o outro jogador através do procedimento "ENVIAR" do arquivo "SERIAL.ASM" e faz as verificações de término de jogo.

TESTA_TECLA: Lê caractere digitado sem apresentar na tela, caso seja ESC, encerra o programa.

SAIR: Encerra o programa.

5 – REFERÊNCIAS

- Material disponibilizado pelo professor em seu website:
<http://www.ic.unicamp.br/~pannain/mc404/mc404.html>
- DOS INT 21h - DOS Function Codes:
<http://spike.scu.edu.au/~barry/interrupts.html>
- Tabela ASCII:
<http://equipe.nce.ufrj.br/adriano/c/apostila/tabascii.htm>