

## Relatório Técnico: Implementação do Jogo Batalha Naval em Assembly x86

### 2. Estrutura Geral do Código

O código foi dividido em seções principais:

#### 2.1 Segmento de Dados (.DATA):

##### 1. Tabuleiros Pré-configurados:

- Três tabuleiros diferentes (TABULEIRO1, TABULEIRO2 e TABULEIRO3), cada um com posições fixas de embarcações.

- Permite variabilidade no jogo por meio da seleção aleatória de tabuleiros.

##### 2. Matriz de Tiros:

- Matriz inicializada com zeros ("0"), que armazena os resultados dos tiros do jogador.

##### 3. Mensagens:

- Mensagens de interação com o jogador, como solicitações de linha e coluna, mensagens de vitória ou desistência.

##### 4. Contador de Acertos:

- Variável CONT\_ACERTOS para acompanhar o progresso do jogador.

#### 2.2 Segmento de Código (.CODE):

##### 1. Configuração Inicial:

- Configura os segmentos de dados e pilha com MOV AX, @DATA, MOV DS, AX e MOV ES, AX.

- Seleciona dinamicamente um tabuleiro com base em um número aleatório gerado pelo procedimento RANDOM.

##### 2. Procedimentos Principais:

- RANDOM: Gera um número aleatório entre 1 e 3 para selecionar o tabuleiro.

- Procedimentos específicos para cada tabuleiro (JOGO1, JOGO2 e JOGO3):

- Gerenciam o loop principal do jogo.

- Executam as rotinas de entrada, verificação e atualização do status do jogo.

### 3. Rotinas e Funções Auxiliares:

- Impressão da Matriz de Tiros:
  - Rotina IMPRIMIR\_TIROS exibe o estado atualizado do tabuleiro de tiros.
- Entrada do Usuário:
  - Recebe e converte as entradas de linha e coluna.
  - Adiciona suporte para desistência ao pressionar "F" ou "f".

### 4. Lógica de Jogo:

- Verificação de acertos e erros:
  - A matriz TABULEIRO[SI+BX] é acessada diretamente com registros.
  - Acertos são marcados com "\*", e erros com "x" na matriz TIROS[SI+BX].
- Condição de vitória:
  - O jogo termina ao atingir 19 acertos, com uma mensagem de vitória.
- Condição de desistência:
  - Permite ao jogador encerrar o jogo antes de completá-lo.

### 3. Detalhamento das Funções e Rotinas Atualizadas

#### 3.1 Seleção de Tabuleiro com Função Aleatória

O procedimento RANDOM utiliza a interrupção INT 21h para obter os centésimos de segundo, garantindo um número pseudo aleatório confiável. O número gerado (entre 1 e 3) determina qual tabuleiro será usado na rodada atual.

#### 3.2 Procedimentos de Jogo (JOGO1, JOGO2, JOGO3)

- Cada procedimento é associado a um tabuleiro específico.
- Gerencia o loop principal do jogo (LOOP\_JOGO) com as seguintes etapas:
  1. Imprime o tabuleiro de tiros com a rotina IMPRIMIR\_TIROS.
  2. Solicita entradas do usuário para linha e coluna.
  3. Calcula o índice com SI e BX para acessar diretamente os elementos da matriz.
  4. Atualiza o tabuleiro com os resultados (acerto ou erro).

### 3.3 Rotina de Impressão da Matriz de Tiros

- A rotina IMPRIMIR\_TIROS exibe o estado atualizado da matriz de tiros no terminal.
- Utiliza INT 21h para imprimir cada célula e realiza quebras de linha ao final de cada linha do tabuleiro.

### 3.4 Entrada do Jogador

- As entradas de linha e coluna são capturadas com INT 21h.
- Adiciona suporte para caracteres ASCII de desistência ("F" ou "f").

### Referências Bibliográficas:

YU, Ytha; MARUT, Charles. Assembly Language Programming and Organization of the IBM PC. New York: McGraw-Hill, 1992.

PANNAIN, Ricardo. Slides Aula. Disponível em: arquivos de aula da disciplina de Organização de Computadores. Acesso em: 13 nov. 2024.