

Aluno: Guilherme Pedrilho Soares

nºUSP: 11800862

Aluno: Leonardo Zaniboni Silva

nºUSP: 11801049

SSC0304 - Introdução à Programação para Engenharias **Prof:** Jo Ueyama

Trabalho Final - 16/06/2020

BATALHA NAVAL

Este documento explicativo se refere ao trabalho final da disciplina de Introdução à Programação para Engenharias ministrada pelo professor orientador Jo Ueyama.

Breve introdução ao jogo

A princípio, a batalha naval é um jogo de dois jogadores cujo objetivo de cada jogador é adivinhar a posição das embarcações inimigas e destruí-las. Desse modo, é declarado vencedor aquele que consegue destruir e afundar todas as embarcações inimigas.

Organização do jogo

O código do jogo foi organizado em funções específicas para cada comando. Ao iniciar o jogo, é chamada a função `layout()` que age como um menu do jogo. A partir do número colocado na entrada, o jogador é direcionado para a respectiva função que corresponde ao número que ele colocou.

Para facilitar o entendimento, as funções são definidas em um arquivo header.

A partir disso, o jogo de fato é iniciado quando o caractere 1 é digitado.

Implementação do jogo na linguagem C e lógica presente

O jogo consiste na implementação de 2 matrizes que simulam o tabuleiro. Cada uma dessa matriz simula a visão de cada um dos jogadores. Ademais, cada matriz possui sua matriz auxiliar onde as armadas são setadas. Para cada uma dessas matrizes, tornou-se necessário o uso de alocação dinâmica em sua criação.

No código, as funções `setArmada1` e `setArmada2` se encarregam de setar as embarcações de forma aleatória com o uso da função `rand` presente na biblioteca `<stdlib.h>`. O mapa de cada jogador consiste na presença de embarcações do seguinte tipo, sendo vista por cada jogador:

- 1 Porta-Aviões
- 2 Couraçados
- 2 Torpedeiros
- 4 Hidroaviões

Um exemplo da disposição da armada no tabuleiro auxiliar de um dos jogadores pode ser encontrada abaixo :

*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	C2	C2	C2	C2	*	*	*	T2	*	T2	*	*	*	*	*	*	*	*
*	*	*	C2	*	*	*	T2	*	T2	T2	T2	*	*	*	*	*	*	*	*
*	*	*	C2	*	*	T2	T2	T2	T2	*	T2	*	*	*	*	*	*	*	*
*	*	C2	C2	C2	C2	T2	T2	T2	*	*	C2	C2	C2	C2	*	*	*	*	*
*	*	*	H2	H2	*	*	*	*	T2	*	*	C2	*	*	*	*	*	*	*
*	*	*	H2	H2	*	*	*	*	T2	T2	T2	*	C2	*	*	*	*	*	*
*	*	*	H2	H2	*	*	*	*	T2	T2	T2	C2	C2	C2	C2	*	*	*	*
*	*	*	H2	H2	*	*	*	*	*	*	*	H2	H2	H2	H2	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	H2	H2	H2	H2	*	*	*	*
H2	H2	*	*	*	*	H2	H2	H2	H2	*	*	*	*	*	*	*	*	*	*
H2	H2	*	*	*	*	H2	H2	H2	H2	P2	P2	P2	P2	*	*	*	*	*	*
H2	H2	*	*	*	*	*	*	*	*	P2	P2	P2	P2	P2	*	*	*	*	*
H2	H2	*	*	*	*	*	*	*	*	*	P2	P2	P2	P2	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

Após o jogo ser iniciado pelo MENU, as matrizes de cada jogador são printadas intercaladamente de acordo com a vez dos mesmos.

Andamento do jogo

Grande parte da lógica da construção do jogo ocorre nas funções `acaoP1` e `acaoP2`. Em ambas as funções, são passadas como parâmetro as matrizes de cada jogador e as suas auxiliares.

A chamada dessas duas funções estão em um loop definido na main, portanto, elas são chamadas intercaladamente de acordo com a vez de cada jogador.

Nessas funções, é requerida uma entrada (ação) do jogador. A partir disso, torna-se possível executar os comandos listados abaixo através da comparação de caracteres.

Comandos

Durante o decorrer do jogo, os jogadores podem utilizar os seguintes comandos :

- **POW:** Comando utilizado para atirar no tabuleiro inimigo.
- **RESET :** inicia uma nova partida e volta ao estado inicial, reiniciando a pontuação e o tempo no relógio
- **SAIR :** Abandona o jogo.
- **AJUDA :** Exibe os comandos que podem ser usados durante o jogo
- **ACASO :** Inicia um novo tabuleiro aleatório, conserva seus pontos e tempo no relógio
- **“GRAVAR {NOME DO ARQUIVO}” :** Salva o jogo em um arquivo com o nome tabuleiro-{timestamp}.txt (timestamp é um número inteiro que indica a data, hora, minuto e segundo de um sistema computacional).
- **“CARREGAR {NOME DO ARQUIVO} “ :** carrega o estado do tabuleiro, pontuação e o tempo no relógio de quando o jogo foi salvo.

Definindo um vencedor

Através de um contador, que é acrescentado 1 toda vez que uma embarcação é atingida, é possível identificar o vencedor quando acertar todas as embarcações inimigas. Ou seja, quando o contador chega a 64 (quantidade total de ‘partes’ de embarcações), a função ‘playervencedor’ é chamada e o loop é encerrado.

Observações

Ao rodar o código em windows, torna-se necessário incluir ‘functions.c’ no início do código para conseguir ler as funções definidas em ‘functions.h’

Contato dos integrantes da dupla

Guilherme Soares Pedrilho : guilherme_soares@usp.br

Leonardo Zaniboni Silva : leonardozaniboni@usp.br