

Relatório Técnico – Batalha Naval

1. Objetivo do projeto

Desenvolver um jogo de Batalha Naval em linguagem C, utilizando conceitos de programação estruturada, modularização, manipulação de estruturas e entrada/saída de dados. O jogo permite dois jogadores, com posicionamento manual ou automático das frotas.

2. Estrutura do código

- `main.c`: Menu principal, seleção de jogadores, tamanho do tabuleiro e tipo de posicionamento da frota.
- `game.c` / `game.h`: Inicialização do jogo, turnos dos jogadores, checagem de vitória, funções de tiro.
- `board.c` / `board.h`: Criação e manipulação do tabuleiro, verificação de limites, colocação de navios.
- `fleet.c` / `fleet.h`: Gerenciamento da frota e navios, contagem de acertos, verificação de afundamentos.
- `io.c` / `io.h`: Funções de entrada de dados do usuário, parsing de coordenadas.
- `rnd.c` / `rnd.h`: Funções de geração de números aleatórios para posicionamento automático.
- `Makefile`: Compilação automatizada do projeto.

3. Decisões de design

- Cada módulo tem responsabilidade única (coisas separadas para facilitar manutenção e leitura do código).
- Estruturas `Board` e `Fleet` encapsulam dados importantes, como células e navios.

- Utilização de funções para checagem de limites (`board_in_bounds`) e verificação de tiros (`already_shot`).
- Mensagens amigáveis ao usuário para interações de tiro, erros de entrada e afundamento de navios.
- Permite posicionamento manual e automático para aumentar a flexibilidade do jogo.

4. Fluxo do jogo

1. O usuário seleciona o modo de posicionamento e informa os nomes dos jogadores.
2. As frotas são posicionadas automaticamente ou manualmente.
3. Jogadores alternam turnos, digitando coordenadas para disparar.
4. O jogo informa acertos, erros, afundamentos e impede disparos repetidos.
5. Quando todos os navios de um jogador são afundados, o outro jogador vence.

5. Dificuldades e soluções

- **Verificação de limites e sobreposição de navios:** resolvido com a função `can_place`.
- **Evitar tiros repetidos:** implementado com `already_shot`.
- **Entrada de dados do usuário:** funções do módulo `io` garantem parsing correto de coordenadas.

6. Como compilar e executar

- Abrir terminal na pasta do projeto.
- Compilar com `make`.
- Executar o jogo com `./batalha_naval` (Windows: `batalha_naval.exe`).

- Limpar arquivos compilados com `make clean`.