

## JOGO BATALHA NAVAL

### ATENÇÃO PARA COMPILAR E RODAR

```
gcc -std=c99 -O2 batalha_naval.c -o batalha_naval  
./batalha_naval
```

### CÓDIGO DO JOGO

(basta colar em [batalha\\_naval.c](#)), porém se for para aprender digite o arquivo passo a passo entendendo toda a sequência.

```
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <time.h>  
#include <ctype.h>  
#include <stdbool.h>  
  
#define SIZE 10  
static const int SHIP_SIZES[] = {5,4,3,3,2};  
#define SHIP_COUNT (sizeof(SHIP_SIZES)/sizeof(SHIP_SIZES[0]))  
  
typedef struct {  
    char ships[SIZE][SIZE]; // 'S' = navio, 0 = vazio  
    char shots[SIZE][SIZE]; // 'H' = hit, 'M' = miss, 0 = desconhecido  
} Board;  
  
int total_ship_cells(void){  
    int sum = 0;  
    for (int i=0;i<SHIP_COUNT;i++) sum += SHIP_SIZES[i];  
    return sum;  
}  
  
void clearBoard(Board *b){  
    memset(b->ships, 0, sizeof(b->ships));  
    memset(b->shots, 0, sizeof(b->shots));  
}  
  
bool canPlace(const Board *b, int r, int c, int len, bool vertical){  
    if (vertical){  
        if (r + len > SIZE) return false;  
        for (int i=0;i<len;i++) if (b->ships[r+i][c] == 'S') return false;  
    } else {
```

```

        if (c + len > SIZE) return false;
        for (int i=0;i<len;i++) if (b->ships[r][c+i] == 'S') return false;
    }
    return true;
}

```

```

void placeShip(Board *b, int len){
    while (1){
        bool vertical = rand()%2;
        int r = rand()%SIZE;
        int c = rand()%SIZE;
        if (!canPlace(b, r, c, len, vertical)) continue;
        if (vertical){
            for (int i=0;i<len;i++) b->ships[r+i][c] = 'S';
        } else {
            for (int i=0;i<len;i++) b->ships[r][c+i] = 'S';
        }
        return;
    }
}

```

```

void placeAllShips(Board *b){
    for (int i=0;i<SHIP_COUNT;i++) placeShip(b, SHIP_SIZES[i]);
}

```

```

void printHeader(void){
    printf("  ");
    for (int c=0;c<SIZE;c++) printf("%c ", 'A'+c);
    printf("\n");
}

```

```

void printPlayerBoard(const Board *player){
    printf("==== SEU TABULEIRO ==== \n");
    printHeader();
    for (int r=0;r<SIZE;r++){
        printf("%2d | ", r+1);
        for (int c=0;c<SIZE;c++){
            char ch = '.';
            if (player->ships[r][c]=='S'){
                if (player->shots[r][c]=='H') ch='X'; // seu navio atingido
                else ch='S'; // seu navio intacto
            } else {
                if (player->shots[r][c]=='M') ch='O'; // água que o PC já atirou
                else if (player->shots[r][c]=='H') ch='X'; // (raro) acerto em vazio? não ocorre
                else ch='.';
            }
        }
    }
}

```

```

    }
    printf("%c ", ch);
}
printf("\n");
}
}

```

```

void printCpuFogBoard(const Board *cpu){
    printf("\n==== TABULEIRO DO COMPUTADOR (o que você enxerga) ==== \n");
    printHeader();
    for (int r=0;r<SIZE;r++){
        printf("%2d | ", r+1);
        for (int c=0;c<SIZE;c++){
            char ch='.';
            if (cpu->shots[r][c]=='M') ch='O';
            else if (cpu->shots[r][c]=='H') ch='X';
            else ch='.';
            printf("%c ", ch);
        }
        printf("\n");
    }
}

```

```

int handleShot(Board *target, int r, int c){
    // -1 = repetido/ inválido, 0 = água, 1 = acerto
    if (r<0 || r>=SIZE || c<0 || c>=SIZE) return -1;
    if (target->shots[r][c] != 0) return -1; // já tentado
    if (target->ships[r][c]=='S'){
        target->shots[r][c]='H';
        return 1;
    } else {
        target->shots[r][c]='M';
        return 0;
    }
}

```

```

bool parseMove(const char *line, int *outR, int *outC){
    // Aceita: "A5", "5A", "a10", "5 3", "5,3"
    int r=-1, c=-1;
    char A=0, B=0;
    int num1=-1, num2=-1;

    // Remove espaços iniciais/finais
    while (*line && isspace((unsigned char)*line)) line++;
}

```

```

// Caso 1: letra + número (A5 / a10)
if (isalpha((unsigned char)line[0])){
    A = toupper((unsigned char)line[0]);
    if (A<'A' || A>'A'+SIZE-1) return false;
    c = A - 'A';
    // pular letra e espaços
    line++;
    while (*line && isspace((unsigned char)*line)) line++;
    if (!isdigit((unsigned char)*line)) return false;
    num1 = (int)strtol(line, NULL, 10);
    r = num1 - 1;
}
// Caso 2: número + letra (10C)
else if (isdigit((unsigned char)line[0])){
    num1 = (int)strtol(line, (char**)&line, 10);
    while (*line && isspace((unsigned char)*line)) line++;
    if (*line==',' || *line==' '){
        // Caso 3: dois números "5,3" ou "5 3"
        line++;
        while (*line && isspace((unsigned char)*line)) line++;
        if (!isdigit((unsigned char)*line)) return false;
        num2 = (int)strtol(line, NULL, 10);
        r = num1 - 1;
        c = num2 - 1;
    } else if (isalpha((unsigned char)*line)){
        B = toupper((unsigned char)*line);
        r = num1 - 1;
        c = B - 'A';
    } else {
        // pode ser que não tenha nada após (inválido)
        return false;
    }
} else {
    return false;
}

if (r<0 || r>=SIZE || c<0 || c>=SIZE) return false;
*outR = r; *outC = c;
return true;
}

void readMoveInteractive(int *r, int *c){
    char buf[128];
    while (1){
        printf("\nSua vez! Informe a posição (ex.: A5, 5A ou 5 3): ");
    }
}

```

```

        if (!fgets(buf, sizeof(buf), stdin)) continue;
        // remover \n
        buf[strcspn(buf, "\r\n")] = 0;
        if (parseMove(buf, r, c)) return;
        printf("Entrada inválida. Tente novamente.\n");
    }
}

void cpuPickTarget(const Board *player, int *r, int *c){
    // simples: aleatório sem repetir
    do {
        *r = rand()%SIZE;
        *c = rand()%SIZE;
    } while (player->shots[*r][*c] != 0);
}

int main(void){
    srand((unsigned)time(NULL));

    Board player, cpu;
    clearBoard(&player);
    clearBoard(&cpu);

    placeAllShips(&player);
    placeAllShips(&cpu);

    const int TARGET = total_ship_cells();
    int playerHits = 0;
    int cpuHits = 0;

    printf("=== BATALHA NAVAL (C) ===\n");
    printf("Regras: X = acerto, O = água. No seu tabuleiro, 'S' mostra onde estão seus navios.\n");

    while (playerHits < TARGET && cpuHits < TARGET){
        printPlayerBoard(&player);
        printCpuFogBoard(&cpu);

        // Jogada do usuário
        int r, c, res;
        readMoveInteractive(&r, &c);
        res = handleShot(&cpu, r, c);
        if (res == -1){
            printf("Você já atirou nessa posição! Perdeu a vez? Não, tente outra.\n");
            continue; // não consome a vez
        } else if (res == 1){

```

```

        playerHits++;
        printf("-> ACERTOU em %c%d! (%d/%d)\n", 'A'+c, r+1, playerHits, TARGET);
    } else {
        printf("-> ÁGUA em %c%d.\n", 'A'+c, r+1);
    }

    if (playerHits >= TARGET) break; // venceu antes do PC jogar

    // Jogada do computador
    int cr, cc;
    cpuPickTarget(&player, &cr, &cc);
    int cres = handleShot(&player, cr, cc);
    if (cres == 1){
        cpuHits++;
        printf("PC atirou em %c%d: ACERTOU! (%d/%d)\n", 'A'+cc, cr+1, cpuHits, TARGET);
    } else {
        printf("PC atirou em %c%d: água.\n", 'A'+cc, cr+1);
    }
}

printPlayerBoard(&player);
printCpuFogBoard(&cpu);

if (playerHits >= TARGET) {
    printf("\nPARABÉNS! Você afundou toda a frota inimiga!\n");
} else {
    printf("\nQue pena... o computador afundou sua frota.\n");
}

return 0;
}

```

Copie o código para um arquivo chamado **batalha\_naval.c**.  
Configure o Git No terminal digite o seguinte código

```

git config --global user.name "Seu Nome"
git config --global user.email "seuemail@example.com"

```

Logo após crie o repositório local, no terminal, dentro da pasta onde está o **batalha\_naval.c**:

```

git init
git add batalha_naval.c

```

```
git commit -m "Versão inicial do jogo Batalha Naval em C"
```

Crie o repositório no GitHub seguindo os seguintes passos:

- Acesse o [GitHub](https://github.com), faça login.
- Clique em New repository.
- Dê um nome, por exemplo: `batalha-naval-c`.
- Escolha público ou privado.
- Crie o repositório (sem README, para não dar conflito).

Para finalizar conecte o repositório local ao GitHub digite no terminal:

```
git remote add origin https://github.com/SEU_USUARIO/batalha-naval-c.git  
git branch -M main  
git push -u origin main
```

Boa sorte!