



Pedro Henrique Silva
github.com/pedrohjsilva
pedrohjsilva@ufmg.br

1. Da codificação do programa

```
#define alert printf  
#define prompt scanf
```

Os comandos do tipo `#define` foram utilizados para permitir fácil entendimento tanto para usuários de Javascript quando de C. Apesar de `alert` e `prompt` substituírem `printf` e `scanf` no corpo do código, a sintaxe da linguagem C não foi alterada.

```
#include <stdio.h>  
#include <locale.h>  
#include <math.h>  
#include <string.h>  
#include <stdlib.h>  
#include <time.h>  
#include "batalha_naval_lib.h"  
#include <windows.h>  
#define L 10  
#define C 10
```

As bibliotecas `<stdio.h>`, `<locale.h>`, `<math.h>`, `<string.h>`, `<stdlib.h>`, `<time.h>`, `"batalha_naval_lib.h"` e `<windows.h>` foram adicionadas ao código, de maneira que, respectivamente, o código pudesse:

- Utilizar os comandos de entrada e saída padrão;
- Imprimir caracteres da do idioma brasileiro no prompt de comando;
- Operar utilizando atribuições e termos matemáticos;
- Manipular vetores de caracteres (strings);
- Gerar a randomização de um número;
- Utilizar a biblioteca de leitura de arquivo criada pelo Professor;
- Trabalhar no prompt de comando executando programas do Windows (foi utilizada para reproduzir a trilha sonora do jogo);

```
int randomico(int LimI, int LimS)  
{  
    int num;  
    num=(rand() % (LimS-LimI + 1)) + LimI;  
}
```

Essa função permitiu a criação de um número randômico, o qual foi utilizado para gerar as jogadas da máquina posteriormente.



```
int carregarTabuleiros(char arquivo1[], char arquivo2[],char
tabuleiro1[NUM_LINHAS][NUM_COLUNAS], char tabuleiro2[NUM_LINHAS]
[NUM_COLUNAS]);
```

Essa função, disponibilizada pelo Professor, permite carregar os mapas para executar as jogadas posteriormente.

```
void mostrarTabuleiros(char mapa_pc[10][11], char mapa_j[10][11], int jogadas[10][11]){
    alert("\n\nMAPA DO JOGADOR\t\t\t\t\tMAPA DA MÁQUINA\n\n");
    // na exibição dos mapas, foi decidido usar apenas números, e não caracteres, para tornar
    // a experiência do usuário mais simples. dessa maneira, ao ser solicitado uma coordenada
    // ele precisará inserir apenas os números correspondentes
    // além disso, nos simplifica de definirmos uma função de conversão
    // ente (A-J) e (1-10)
    alert("\n      1 2 3 4 5 6 7 8 9 10\t\t\t 1 2 3 4 5 6 7 8 9 10\n");
    alert("      _____\t\t\t_____ \n");
    for (int i = 0; i < 10; i++)
    {
        if(i<9)
        {
            alert("%d ", i+1);} else
            alert("%d", i+1);
            alert("| ");
            for(int j = 0; j < 11-1; j++)
            {
                alert("%c ", mapa_j[i][j]);
            }
            alert(" |\t\t| ");
            for(int j = 0; j < 11-1; j++)
            {
                if(jogadas[i][j]!=0)
                {
                    alert("%c ", mapa_pc[i][j]);
                }
                else
                {
                    alert("~ ");
                }
            }
            alert(" |\n");
        }
        alert(" |_____| |\t\t| _____|\n");
        alert("\n\n");
    }
}
```

As linhas do código descritas acima, foram utilizadas para realizar a exibição do tabuleiro no prompt de comando. Os comentários, foram mantidos, pois auxiliam na compreensão do código.

```
// atirar - essa função gera duas matrizes para manipular as jogadas.
// nós a inicializamos apenas com 0's. ao atiras, substituímos uma
posição por 2
// e, ao acertarmos uma embarcação, nós substituímos a posição por 1
// no fim, utilizamos isso para verificar o status da vitória
// essa, armazena os disparos do jogador
int atirar_j (char mapa_pc[10][11], int jogadas_pc[10][11] )
{
    int linha,coluna;
    // para o jogador, solicitamos as coordenadas e lemos ela do teclado
    alert("VEZ DO JOGADOR\nDigite a linha:\n");
    prompt("%d", &linha);
    alert("\nInsira a coluna:\n");
    prompt("%d", &coluna);
    jogadas_pc[linha-1][coluna-1] = 2;
    if (mapa_pc[linha-1][coluna-1] != '~')
    {
        jogadas_pc[linha-1][coluna-1] = 1;
    }
}

// já essa, armazena disparos da máquina
int atirar_pc (char mapa_j[10][11], int jogadas_j[10][11] )
{
    int linha,coluna;
    // para a máquina geramos os valores de maneira randomica
    Sleep(1000);
    alert("\n\nA MÁQUINA ESTÁ JOGANDO...");
    linha = randomico (1,10);
    Sleep(1000);
    alert("\n\nA MÁQUINA ESTÁ JOGANDO...");
    coluna = randomico(1, 10);
    jogadas_j[linha-1][coluna-1] = 2;
    if (mapa_j[linha-1][coluna-1] != '~')
    {
        jogadas_j[linha-1][coluna-1] = 1;
    }
}
```

As linhas de código acima mostram como foi implementado o sistema de jogadas. Note que as jogadas da máquina são geradas aleatoriamente, utilizando a função criada anteriormente.

```
// essa é a função que definimos para verificar a matriz com o status
da vitória da máquina
int verifica_pc (int jogadas_pc[10][11])
{
    int acertos = 0;
    for (int i=0; i<10; i++)
    {
        for (int j=0; j>10; j++)
        {
            if(jogadas_pc[i][j]==1){acertos++;}
        }
    }
}
```

```
}
}
if(acertos==21){system("cls"); alert("Vitória da máquina!");
}
return verifica_pc;
}
// e essa verifica o status da vitória para o jogador
int verifica_j (int jogadas_j[10][11])
{
    int acertos = 0;
    for (int i=0; i<10; i++)
    {
        for (int j=0; j>10; j++)
        {
            if(jogadas_j[i][j]==1){acertos++;
            }
        }
    }
    if(acertos==21){system("cls"); alert("Vitória do jogador!");
    }
    return verifica_j;
}
```

As linhas acima, mostram os critérios de verificação do status de jogo. Ao ser identificada a vitória de um jogador, a mensagem correspondente é impressa na tela.

```
//função principal, onde nosso jogo é executado
int main ( )
{
    srand(time(0));
    system("start contra_nes.mp3");
    setlocale (LC_ALL, "");
    setlocale (LC_CTYPE, "pt_BR.UTF-8" );
    char caminho_j[] = "arquivo1.txt";
    char mapa_j[10][11];
    gera_mapa(caminho_j, mapa_j);
    char caminho_pc[] = "arquivo2.txt";
    char mapa_pc[10][11];
    gera_mapa(caminho_pc, mapa_pc);
    int jogadas[10][11];
    for (int i = 0; i < 10; i++)
    {
        for(int j = 0; j < 11-1; j++)
        {
            jogadas[i][j]=0;
        }
    }
    // esse char foi criado para imprimir durante a tela de carregamento
    // caso não o "criássemos", a interface não imprimiria o símbolo de "%"
    // na interface do cmd
    char porcento = '%';

    // essas linhas do código tentam simular a API de um jogo de Nintendo
    Entertainment System
    // utilizamos um front-end bem simples, uma vez que não podemos
    utilizar
```

[illegible]

[illegible]

```
// jogadas - a partir daqui, o laço de repetição garante que as
jogadas sejam repetidas de maneira infinita
// o único critério para encerrar o jogo, é a vitória de um dos
jogadores
// não podemos fazer nada quanto ao velho "Alt + F4"
```

```
atirar_j(mapa_pc, jogadas);
verifica_j(jogadas);

while (verifica_pc != 21)
{
system("cls");
mostrarTabuleiros(mapa_pc, mapa_j, jogadas);
atirar_pc(mapa_j, jogadas);
verifica_j(jogadas);
while (verifica_j !=21)
{
system("cls");
mostrarTabuleiros(mapa_pc, mapa_j, jogadas);
atirar_j(mapa_pc, jogadas);
verifica_j(jogadas);
};
};
}
```

Esse é o corpo da função principal do programa. O jogo, bem como o laço de repetição que permite as jogadas serem intercaladas, estão dentro dela. Novamente, comentários do código foram mantidos, por serem pertinentes ao que foi proposto para esse documento.

2 Das capturas de tela

A seguir, algumas das capturas de telas do jogo.

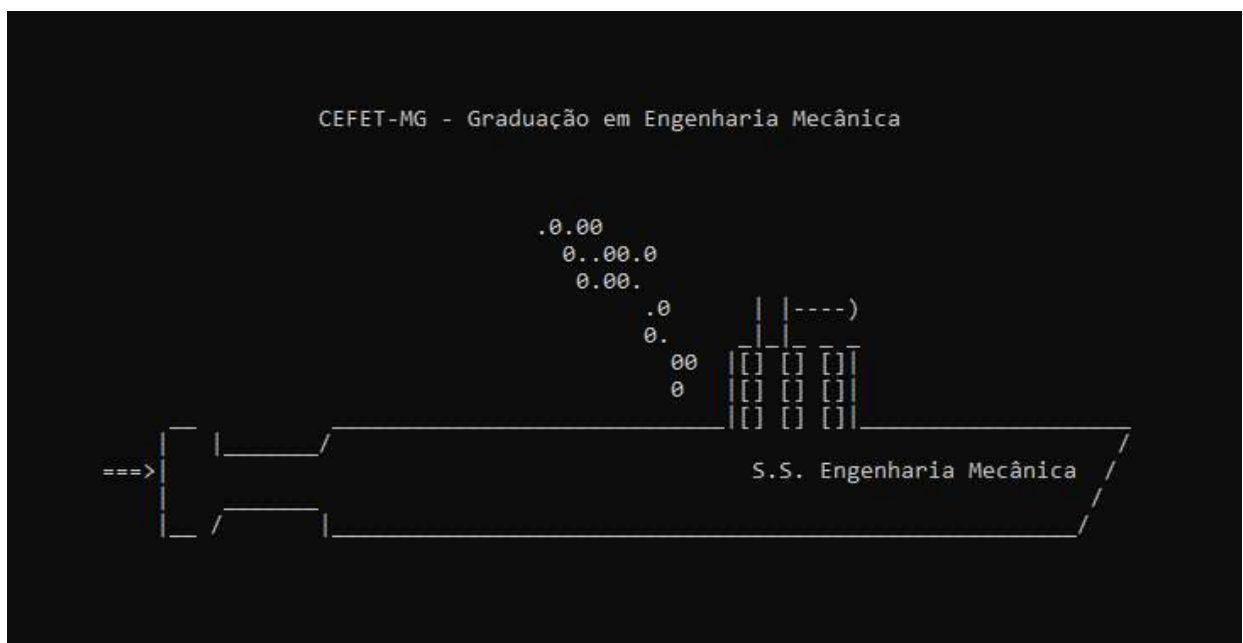


Figura 2.1: Tela inicial da API do Jogo

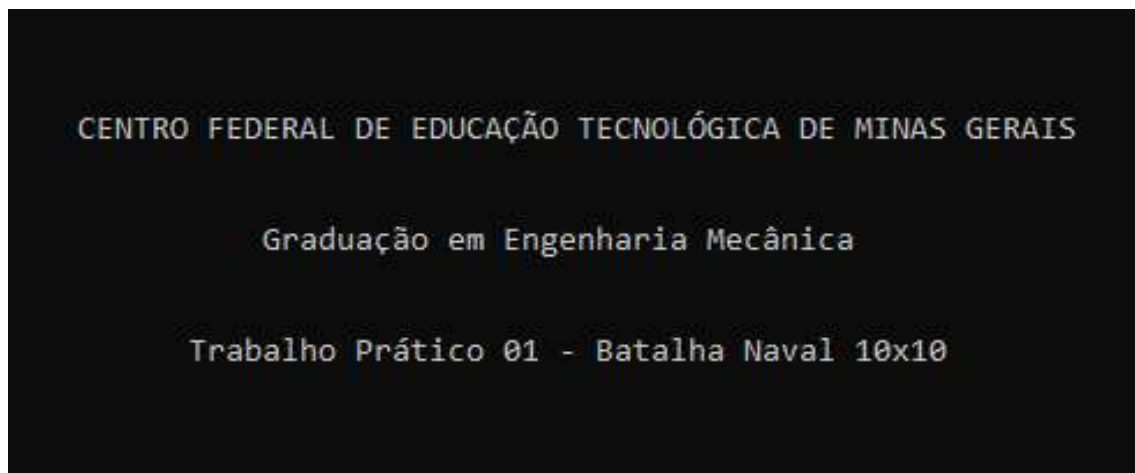


Figura 2.2: Créditos iniciais da API do jogo

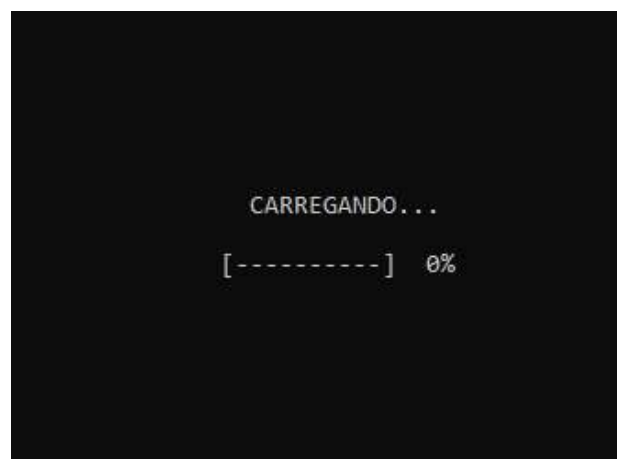


Figura 2.3.1: Tela de carregamento do jogo, em 0%



Figura 2.3.2: Tela de carregamento do jogo, em 50%

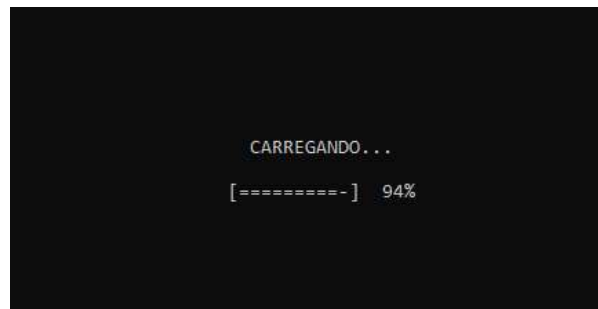


Figura 2.3.3: Tela de carregamento do jogo, em 94%



Figura 2.4.1: Mapas sendo exibidos durante a execução do jogo. Jogador inserindo coordenada de linha

MAPA DO JOGADOR										
	1	2	3	4	5	6	7	8	9	10
1	D	~	~	~	~	~	~	~	~	S
2	D	~	~	P	P	P	P	~	~	S
3	~	S	S	S	~	~	D	D	~	S
4	~	~	~	~	~	S	~	~	~	~
5	~	~	~	~	~	S	~	~	~	D
6	B	~	~	~	~	S	~	~	~	D
7	B	~	~	~	~	~	~	~	~	~
8	~	~	~	~	~	~	~	~	~	~
9	~	~	~	~	~	~	~	~	~	~
10	~	~	~	~	~	~	~	~	~	~

MAPA DA MÁQUINA										
	1	2	3	4	5	6	7	8	9	10
1	~	~	~	~	~	~	~	~	~	~
2	~	~	~	~	~	~	~	~	~	~
3	~	~	~	~	~	~	~	~	~	~
4	~	~	~	~	~	~	~	~	~	~
5	~	~	~	~	~	~	~	~	~	~
6	~	~	~	~	~	~	~	~	~	~
7	~	~	~	~	~	~	~	~	~	~
8	~	~	~	~	~	~	~	~	~	~
9	~	~	~	~	~	~	~	~	~	~
10	~	~	~	~	~	~	~	~	~	~

VEZ DO JOGADOR
Digite a linha:
1

Insira a coluna:
2_

Figura 2.4.2: Mapas sendo exibidos durante a execução do jogo. Jogador inserindo coordenada de coluna