

# Relatório Batalha Minada

Maria Eduarda Rosa da Silva\* e Mateus Wagner dos Santos†

\*Universidade Federal de Santa Catarina (UFSC)

**Resumo**—Relatório de implementação de um jogo batalha naval modificado em C para Linux.

**Palavras-chave**—Codificação, Linux, Batalha, Barco, Bomba, Água, Debug, Tabuleiro, Aprendizado, Jogabilidade, Posições.

**Abstract**—Implementation report of a C modified naval battle game for Linux.

**Index Terms**—Coding, Linux, Battle, Boat, Bomb, Water, Debug, Board, Learning, Gameplay, Positions.

## I. INTRODUÇÃO

ESTE relatório tem o intuito de introduzir a implementação de um jogo batalha naval modificado em C para Linux. O jogo batalha naval conceitualmente é um jogo em que o jogador ou usuário posiciona navios em um tabuleiro de modo aleatório e de escolha própria. Com propósito do oponente adivinhar onde estes estão posicionados e vice-versa. Desse modo, o objetivo é derrubar os barcos do adversário, ganhando quem derruba todos os navios primeiro.

O Jogo proposto nesse trabalho possui algumas modificações com relação ao jogo original. Neste, o computador introduz os navios aleatoriamente para o usuário. Além disso, o projeto apresenta o diferencial de possuir bombas espalhadas por todo o tabuleiro. Sendo assim, uma junção de dois jogos já existentes e conhecidos: Batalha naval e Campo minado. Em vista disso, o jogador deve tentar acertar os barcos e desviar das bombas.

O trabalho apresentado pelo professor para a turma de Programação I tem como finalidade utilizar o conteúdo aprendido em sala, isto é, a linguagem C e lógica de programação, em uma aplicação real. Dessa maneira, desenvolver e avançar no conhecimento pela implementação de um código em um projeto de jogo simples.

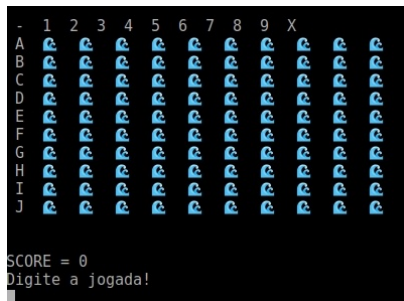


Figura 1. Tabuleiro do jogo.

O artigo é estruturado em desenvolvimento para apresentação da lógica de programação, código e jogabilidade do jogo e conclusão para exposição de resultados.

## II. DESENVOLVIMENTO

Esta seção descreve as regras do jogo e procedimentos aplicados na codificação.

### A. Regras do jogo

O objetivo principal do jogo batalha minada é atingir os barcos no tabuleiro. Em vista disso, o jogo é constituído de um tabuleiro com marcações de linhas com letras e colunas com números, bombas e barcos colocados aleatoriamente no campo. O projeto possui três fases, sendo cada fase tem -se um aumento do número de linhas, colunas, bombas e barcos com o propósito de ampliar a dificuldade.

No início do jogo, é apresentado ao jogador as instruções ou modo de jogar. A seguir, exibe-se o menu inicial com cinco opções de escolhas. As opções são, sucessivamente, iniciar jogo, reiniciar jogo, apresentar score, mostrar instruções e sair.

O usuário a fim de acertar um barco precisa escolher uma posição do campo. Para isso deve escolher uma letra e número do tabuleiro e tentar a sorte. É necessário digitar *LetraNúmero* respectivamente e pressionar a tecla *Enter* do teclado. Para passar de nível, o jogador precisa acertar todos os barcos. Porém ao acertar três bombas é dado *Game over*, o jogo é reiniciado e o score zerado. O jogador tem a possibilidade de voltar ao menu ou reiniciar a qualquer momento o jogo.

No score total, a cada acerto de barco contabiliza-se dez pontos, de bomba são menos cinco pontos e ao derrubar todos os barcos e passar de nível acrescenta-se cinquenta pontos.

### B. Procedimentos aplicados

Inicialmente para o funcionamento e realização dos procedimentos do jogo foram utilizadas diversas bibliotecas, sendo elas: *stdio.h*, *unistd.h*, *stdlib.h*, *time.h*, *string.h*, *wchar.h*, *locale.h*. Após a printar as mensagens iniciais, apresentação e modo de jogo, exibe-se o menu inicial no qual emprega-se uma função de modo a facilitar a jogabilidade onde o usuário entra opção desejada para prosseguir o programa.

O jogo é executado com uma matriz tabuleiro de  $i$  linhas e  $j$  colunas que variam nos diferentes três níveis do jogo através da aplicação da estrutura de repetição *for*. Ademais, para posicionar barcos e bombas em locais totalmente aleatórios a cada nova partida jogada, independente da fase, a função *rand()* foi utilizada. Além de, sempre ocorrer uma comparação entre tabelas e locais de armazenamento de objetos para que não acontecer sobreposição de elementos. Porém, no começo

o usuário apenas tem acesso a um campo completo com água e ao fazer sua jogada, se acertar outro objeto diferente disso, a matriz apresenta o novo elemento.

Tendo em vista que o tabuleiro foi criado em vetor de caracteres do tipo *char*, foram necessárias diversas comparações para leitura do código e facilitar a jogabilidade, sempre com intuito de proporcionar uma melhor experiência ao jogador com o Batalha Minada. Em razão da necessidade do usuário de inserir uma entrada a fim de fazer sua jogada, foi aplicado *fgets* considerando evitar conflitos com o *buffer* e melhor conversão de valores de letras e números, uma vez que, armazena e compara *strings*.

Com o propósito de deixar mais limpo e descontraído o Batalha Minada, ocorreu a aplicação da função *system*, própria do sistema operacional do LINUX, para limpar tela e colocar áudio a cada elemento selecionado no tabuleiro.

### III. CONCLUSÃO

Ao final do processo de criação do jogo, conclui-se que a partir do objetivo proposto no encabeçamento do projeto houve cem por cento de aproveitamento e resultado esperado. Além do evidente crescimento na compreensão da linguagem e sua lógica, uma vez que, o esforço demandado para a realização deste envolveu muitos estudos e discussões.

Maiores dificuldades encontradas no desenvolvimento do projeto foram, principalmente, a necessidade de pesquisar conteúdos essenciais e/ou mais avançados que ainda não tinham sido apresentados em aula para realização de instruções específicas da jogabilidade. Outra questão que demandou tempo e atenção foram erros mínimos no código que influenciaram na saída e compilação, assim, surgindo a necessidade de buscas sucessivas de erros no programa.

#### APÊNDICE A CABEÇALHO.H

Esta seção apresenta o cabeçalho do jogo.

```
#ifndef arquivo_h
#define arquivo_h
#include <stdio.h>
int menu(int a){
[...]
```

#### APÊNDICE B FUNÇÕES E PROCEDIMENTOS.C

Esta seção retrata o código do terceiro nível do jogo.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

#include <wchar.h>
```

```
#include <locale.h>
```

```
int menu(int a){
//menu
wprintf(L"_____MENU\n");
wprintf(L"Aperte:_\n");
wprintf(L"\n[1] JOGAR");
wprintf(L"\n[2] REINICIAR");
wprintf(L"\n[3] SCORE");
wprintf(L"\n[4] INSTRU ES");
wprintf(L"\n[5] SAIR\n");
scanf("%d", &a);
switch (a){
case 1:
wprintf(L"o_jogo_vai_come ar!\n");
return(1);
case 2:
wprintf(L"O_jogo_ser _reiniciado!");
return(2);
case 3:
//wprintf(L"Pressione [S] para som");
return(3);
case 4:
wprintf(L"O_seu_score_atual _de:");
return(4);
case 5:
wprintf(L"Obrigado_por_jogar!:_)\n");
break;
return(5);
default:
if (a < 0 || a > 5){
wprintf(L"Por_favor,_digite_um");

return(0);
}
}
return(0);
}
```

```
int main (void){
int i=0,j=0,d=0, barco=0, bomba=0, score=0;
int li,c,p,q, retorno=0, a=0;
char lin=0, col=0;
char tabuleiro1[16][16];
wchar_t tabuleironovo[16][16];
time_t t;
wchar_t n = 0x1F4A5;
srand(time(NULL));
setlocale(LC_CTYPE, "");
```

```
system("clear");
retorno = menu (a);
while (retorno == 0){
```

```

    retorno = menu (a);

    }

    if (retorno == 1 || retorno == 3){
        for (i=0; i<16; i++){
            for ( j=0; j<16; j++){
                tabuleirol[i][j] = '~';

            }
            // wprintf(L"\n");
        }
        for(i=1; i<=5; i++){
            int b=0;
            do{
                b=0;
                li = 1 + rand()%15;
                c = 1 + rand()%15;
                for(j=li; j<li+i; j++){
                    if(tabuleirol[j][c]!='~'){
                        b=1;
                        break;
                    }
                }
            }
            while(b==1);
            // wprintf(L"%d", li);
            // wprintf(L"-%d\n", c);
            if((i!=0) || (j!=0)){
                for(j=li; j<li+i; j++){
                    tabuleirol[j][c]='@';
                }
            }
        }
        int cont = 0;
        while (cont < 12){

            li = 1 + rand()%15;
            c = 1 + rand()%15;
            if((li!=0) || (c!=0)){
                if(tabuleirol[li][c] == '~' || tabuleirol[li][c] != '@') tabuleirol[li][c]='*';
                cont++;
            }
        }
        //sleep(1);
        system("clear");

        for(i=0; i<16; i++){
            for(j=0; j<16; j++){
                tabuleironovo[i][j]= '_';

            }

            tabuleironovo[0][0] = L'-' ;
            tabuleironovo[1][0] = L'A' ;
            tabuleironovo[2][0] = L'B' ;
            tabuleironovo[3][0] = L'C' ;
            tabuleironovo[4][0] = L'D' ;
            tabuleironovo[5][0] = L'E' ;
            tabuleironovo[6][0] = L'F' ;
            tabuleironovo[7][0] = L'G' ;
            tabuleironovo[8][0] = L'H' ;
            tabuleironovo[9][0] = L'I' ;
            tabuleironovo[10][0] = L'J' ;
            tabuleironovo[11][0] = L'K' ;
            tabuleironovo[12][0] = L'L' ;
            tabuleironovo[13][0] = L'M' ;
            tabuleironovo[14][0] = L'N' ;
            tabuleironovo[0][1] = L'1' ;
            tabuleironovo[0][2] = L'2' ;
            tabuleironovo[0][3] = L'3' ;
            tabuleironovo[0][4] = L'4' ;
            tabuleironovo[0][5] = L'5' ;
            tabuleironovo[0][6] = L'6' ;
            tabuleironovo[0][7] = L'7' ;
            tabuleironovo[0][8] = L'8' ;
            tabuleironovo[0][9] = L'9' ;
            tabuleironovo[0][10] = L'X' ;
            tabuleironovo[0][11] = L'Y' ;
            tabuleironovo[0][12] = L'Z' ;
            tabuleironovo[0][13] = L'R' ;
            tabuleironovo[0][14] = L'S' ;

            for ( i=0; i<15; i++){
                for ( j=0; j<15; j++){
                    if (j !=0 && i!=0){
                        tabuleironovo[i][j] = 0x1F30A;
                    }
                }
            }

            for ( i=0; i<16; i++){
                for ( j=0; j<16; j++){
                    wprintf(L"%21c", tabuleironovo[i][j]);
                }
            }

            wprintf(L"\n");
        }

        //-----Entrada do usuario-----
        wprintf(L"\nSCORE_=%d\n", score);
        char entrada[256];
        i=0;
        j=0;
        int controle = 0;

```

```

while(bomba < 3 && barco < 15){
    wprintf(L"Para_sair_digite_[sair]\n");
    wprintf(L"Digite_a_jogada!\n");
    wprintf(L"\n\nPara_sair_digite_[sair].\n");
    wprintf(L"Para_reiniciar_digite_[reiniciar].\n");
    setbuf(stdin, NULL);
    fgets(entrada, 256, stdin);
    system("clear");
    if (strcmp(entrada, "sair\n")==0) break;
    if (strcmp(entrada, "reiniciar\n")==0){
        score = 0;
        for ( i=0; i<15; i++){
            for ( j=0; j<15; j++){
                if (j !=0 && i!=0){
                    tabuleironovo[i][j] = 0x1F30A5;
                }
                wprintf(L"%2lc", tabuleironovo[i][j]);
            }
            wprintf(L"\n");
        }
        wprintf(L"\nSCORE_=%d\n", score);
    }
    if (strcmp(entrada, "sair\n")==0) break;

    lin = entrada[0];
    if(strlen(entrada) == 4){
        col = 'x';
    }else if(strlen(entrada) == 3){
        col = entrada[1];
    }else{
        i = 20;
    }

    if (lin=='A' || lin=='a') i=1;
    else if (lin=='B' || lin=='b') i=2;
    else if (lin=='C' || lin=='c') i=3;
    else if (lin=='D' || lin=='d') i=4;
    else if (lin=='E' || lin=='e') i=5;
    else if (lin=='F' || lin=='f') i=6;
    else if (lin=='G' || lin=='g') i=7;
    else if (lin=='H' || lin=='h') i=8;
    else if (lin=='I' || lin=='i') i=9;
    else if (lin=='J' || lin=='j') i=10;
    else if (lin=='K' || lin=='k') i=11;
    else if (lin=='L' || lin=='l') i=12;
    else if (lin=='M' || lin=='m') i=13;
    else if (lin=='N' || lin=='n') i=14;
    else{
        i = 20;
    }

    if (col=='1')
        {j=1; printf("_");}
    else if (col=='2')
        {j=2; printf("_");}
    else if (col=='3')
        {j=3; printf("_");}
    else if (col=='4')
        {j=4; printf("_");}
    else if (col=='5')
        {j=5; printf("_");}
    else if (col=='6')
        {j=6; printf("_");}
    else if (col=='7')
        {j=7; printf("_");}
    else if (col=='8')
        {j=8; printf("_");}
    else if (col=='9')
        {j=9; printf("_");}
    else if ((col=='x') || (col=='X'))
        j=10; //printf(" ");
    else if ((col=='y') || (col=='Y'))
        j=11; //printf(" ");
    else if ((col=='z') || (col=='Z'))
        j=12; //printf(" ");
    else if ((col=='r') || (col=='R'))
        j=13; //printf(" ");
    else if ((col=='s') || (col=='S'))
        j=14; //printf(" ");
    else{
        j = 20;
    }
    controle = 0;
    if( (i < 1) || (i > 14) || (j<1) || (j>14) )
    {
        wprintf(L"ENTRADA_INV LIDA, _DIGITE_NOVAMENTE\n");
        controle = 10;
    }

    if(controle !=10){
        wprintf(L"%c\n", tabuleiro1[i][j]);

        if(tabuleiro1[i][j]=='*'){
            //if (retorno = 3){
            system("play_bomba.mp3_>_/dev/null_2>&");

            //}
            bomba += 1;
            score = score-5;
            wprintf(L"\nOLHA_A_BOMBA!\n\n");

            // novo tabuleiro
            tabuleironovo[i][j]= 0x1F4A5;

            wprintf(L"\nSCORE_=%d\n\n", score);
        }

        if(bomba==3){
            wprintf(L"\n__GAME_OVER!__\n");
            wprintf(L"SCORE_=%d", score);
            break;
        }
    }
}

```

```

        }
        wprintf(L"Obrigado_por_jogar!:_:\n");
    }
    return(0);
}

}

if(tabuleiro1[i][j]=='@'){
//if (retorno = 3){
system("play_buzina.mp3>_/dev/null_2>&1_&");

//}
score = (score + 10);
wprintf(L"\nFRAGMENTO_ENCONTRADO!\n");

// novo tabuleiro
tabuleironovo[i][j]= 0x26F5;

wprintf(L"\nSCORE=_%d\n", score);

barco += 1;
if(barco == 15){
    wprintf(L"PARAB NS ,_PASSOU_DE_N VEL!\n");
    score = score + 50;
    wprintf(L"\nSCORE=_%d\n", score);

}

}

if(tabuleiro1[i][j]=='~'){
//if (retorno = 3){
system("play_agua.mp3>_/dev/null_2>&1_&");

//}
wprintf(L"\nACERTOU_ GUA !\n");

// novo tabuleiro

wprintf(L"\nSCORE=_%d\n", score);

}

}

for ( i=0; i<16; i++){
for ( j=0; j<16; j++){
if (tabuleironovo[i][j] == 0x26F5)
wprintf(L"%lc_", tabuleironovo[i][j]);
else if (tabuleironovo[i][j] == 0x1F4A5 || tabuleironovo[i][j] == 0x1F30A)
wprintf(L"%lc", tabuleironovo[i][j]);
else
wprintf(L"%lc_", tabuleironovo[i][j]);
}
wprintf(L"\n");
}
}
}

```