

Jogo da memória em C

—

Karine Bezerra Furtado
(Trabalho individual)

```
55 void esconder_easy() {
56     int i, j;
57     for (i=0; i<LINHAS_EASY; i++)
58         for (j=0; j<COLUNAS_EASY; j++)
59             TAB_EASY[i][j]='*';
60 }
61 void esconder_medium() {
62     int i, j;
63     for (i=0; i<LINHAS_MEDIUM; i++)
64         for (j=0; j<COLUNAS_MEDIUM; j++)
65             TAB_MEDIUM[i][j]='*';
66 }
67 void esconder_hard() {
68     int i, j;
69     for (i=0; i<LINHAS_HARD; i++)
70         for (j=0; j<COLUNAS_HARD; j++)
71             TAB_HARD[i][j]='*';
72 }
73 void esconder_insane() {
74     int i, j;
75     for (i=0; i<LINHAS_INSANE; i++)
76         for (j=0; j<COLUNAS_INSANE; j++)
77             TAB_INSANE[i][j]='*';
78 }
```

Na primeira parte do código foram feitas quatro funções (uma para cada nível) para ocultar o conteúdo de cada carta no início do jogo e enquanto seu par não for acertado.

```
void embaralhar_easy() {  
    int i, j, x, y;  
    char aux;  
    srand(time(NULL));  
    for (i=0; i<LINHAS_EASY; i++)  
        for (j=0; j<COLUNAS_EASY; j++){  
            x=rand()%LINHAS_EASY;  
            y=rand()%COLUNAS_EASY;  
            aux=GAB_EASY[i][j];  
            GAB_EASY[i][j]=GAB_EASY[x][y];  
            GAB_EASY[x][y]=aux;  
        }  
}
```

Nessa parte do código foi utilizado a função rand dentro da função 'embaralhar' para embaralhar as letras do jogo, cada nível teve uma função para o embaralhamento como a mostrada ao lado.

```

void jogar_easy (int acao) {
    int i, j, x, y, virar=0, checar=1;
    while (score<3 && acao==1){
        do {
            printf ("\n> Informe as coordenadas (Linha e Coluna) da primeira carta: ");
            scanf ("%d %d", &i, &j);
            if (i<0 || j<0){
                acao=-1;
                break;
            }
            if (i>=LINHAS_EASY || j>=COLUNAS_EASY){
                printf ("Coordenada Invalida!");
                checar=0;
                atualizartelaeasy(acao);
            }
            else
                checar=1;
            if (TAB_EASY[i][j]!='*' && checar==1){
                TAB_EASY[i][j]=GAB_EASY[i][j];
                system ("cls");
                tela_jogo_easy(acao);
                virar++;
            }
            else if (checar==1){
                printf ("Esta carta já foi escolhida!");
                atualizartelaeasy(acao);
            }
        }
    }
}

```

Na função 'jogar' é checado se as coordenadas informadas pelo jogador são válidas.

Caso as coordenadas não estejam dentro dos limites estabelecidos pelo nível escolhido do jogo é mostrada a mensagem 'coordenada inválida'.

Também é checado se as coordenadas informadas pelo jogador estão disponíveis, caso não estejam é mostrada a mensagem 'Esta carta já foi escolhida'.

```

if (TAB_EASY[i][j]==TAB_EASY[x][y]){
    virar=0;
    score++;
    printf ("\n\t\t\t\t\t ACERTOU!");
}
else {
    TAB_EASY[i][j]='*';
    TAB_EASY[x][y]='*';
    virar=0;
    erros++;

    printf ("\n\t\t\t\t\t ERROU!");

}
atualizartelaeasy(acao);
}
}

```

Ainda na função 'jogar', é contabilizado o número de acertos (se as coordenadas corresponderem às mesmas letras) e de erros (caso não correspondam). Cada nível tem a sua função 'jogar'.

Metodologia da pontuação

O jogador vai receber uma quantidade de pontos logo no início da partida (a quantidade vai depender do nível escolhido):

Nível **easy**: começa com 300 pontos

Nível **medium**: começa com 400 pontos

Nível **hard**: começa com 500 pontos

Nível **insane**: começa com 2000 pontos

Metodologia da pontuação

A cada erro, uma certa quantidade de pontos é retirada da pontuação do jogador

Nível **easy**: perde 30 pontos a cada erro

Nível **medium**: perde 40 pontos a cada erro

Nível **hard**: perde 50 pontos a cada erro

Nível **insane**: perde 20 pontos a cada erro

```

void saida_easy() {
    int resultado_easy;
    if(score==0)
        printf("\n\n\tJogo encerrado.\n\n");
    else {
        if(score==3)
            printf("\n\tPARABENS, VOCE ACHOU TODOS OS PARES!");
        resultado_easy=300-(erros*30);
        printf("\n\n\tScore Final: %d\n", resultado_easy);

        if(resultado_easy==300){
            printf("\t ----- \n\t
            printf("\t|    Perfeito! Voce obteve todos os pontos!    | \n\t
            printf("\t ----- \n\t
        }

        if(resultado_easy>=240&&resultado_easy<299){
            printf("\t ----- \n\t
            printf("\t|    Exelente! Voce obteve mais de 80%% dos pontos!    | \n\t
            printf("\t ----- \n\t
        }

        if(resultado_easy<239&&resultado_easy>=150){
            printf("\t ----- \n\t
            printf("\t| Bom! Voce obteve mais de 50%% dos pontos, mas menos que 80%%! | \n\t
            printf("\t ----- \n\t
        }

        if(resultado_easy<149&&resultado_easy>0){
            printf("\t ----- \n\t
            printf("\t|    Regular! Voce obteve menos de 50%% dos pontos!    | \n\t
            printf("\t ----- \n\t

```

Na função ‘saida’, é feito o cálculo da pontuação do jogador, o jogador vai receber uma das quatro mensagens, dependendo da sua performance:

Perfeito: se obter 100% dos pontos

Excelente: se obter 80% ou mais dos pontos

Bom: se obter mais de 50% e menos que 80% dos pontos

Regular: se obter menos de 50% mas não zerar

Péssimo: se perder todos os pontos