

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ENGENHARIA DA COMPUTAÇÃO

ARNALD SOUZA

PROJETO CAMPO MINADO

FUNDAMENTOS DE PROGRAMAÇÃO

APUCARANA

2021

INTRODUÇÃO

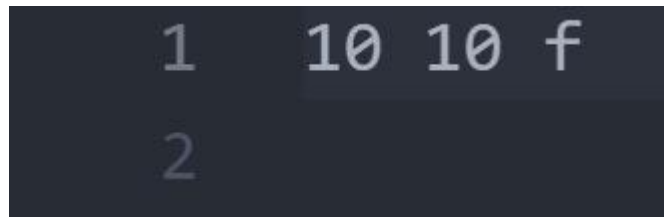
O presente relatório tem como objetivo deixar claro a estruturação do jogo campo minado desenvolvido na disciplina de Fundamentos de Programação na linguagem C, seguindo as devidas determinações passadas pelo professor Luiz Fernando Carvalho.

DESENVOLVIMENTO

Utilizando o editor de texto Atom, inicialmente foram declaradas as bibliotecas usadas que auxiliam na hora de usar diferentes funções. Logo depois, criou-se uma “struct” chamada ‘Estados_bombas’, que contém os tipos de bombas que podem ser usadas.

Na função main, primeiramente usou-se a função setlocale, para que possam ser usados caracteres especiais como acentuação. A declaração das variáveis usadas vem logo abaixo. A função srand(time(NULL)) é para que as bombas não fiquem sempre no mesmo local.

Definiu-se uma variável “definicao” do tipo arquivo, que será aberto com o nome de modo_de_jogo, no modo de leitura, a qual vai ler as definições de tamanho de linhas e colunas e a dificuldade do jogo com a função fscanf. Caso ele não for encontrado no diretório em que se encontra, exibirá a mensagem Arquivo não encontrado.



Logo após foi criado uma matriz, do tamanho especificados pelo jogador no arquivo. Também uma variável do tipo arquivo para a saída.

Um menu de entrada foi criado para o usuário entrar com o que ele desejar:

```
*****
*----- BEM-VINDO AO CAMPO MINADO DO ARNALD -----*
*-----
*----- 1 - Para iniciar o jogo -----*
*----- 2 - Para saber as regras do jogo -----*
*----- 3 - Sair -----*
*-----
*****
```

Se o usuário digitar 1, o jogo se iniciará, limpando a tela com a função `system("cls")` e mostrando a matriz.

[illegible]

Quando digitado 2, as regras serão mostradas.

```
1º - Evitar onde há bombas;  
2º - Os espaços adjacentes ao quadro selecionado dirá quantas bombas existem ao redor do mesmo;  
3º - Para ganhar, deve-se preencher todos campos que não há bombas;  
4º - TENHA UM BOM JOGO.  
Pressione qualquer tecla para continuar. . .
```

Em 3 , sairá do jogo aparecendo a mensagem de finalizando.

Finalizando...

No início do jogo, quando o usuário digitar 1, terá uma condição para saber a quantidade certa de bombas a serem colocadas de acordo com a dificuldade. Foram adicionados mais três bombas às variáveis, por conta da margem de erro do número de bombas que aparecem.

Um FOR foi criado para colocar as bombas em lugares aleatórios usando a função rand. Para pedir ao usuário a coordenada das linhas em que se deseja “abrir” criou-se um laço de repetição do while, que enquanto o que o usuário inserir for maior ou igual ao tamanho da linha ou da coluna ele se repetirá. Se as coordenadas estiverem erradas ele as pedirá novamente e exibirá a mensagem : “Uma das coordenadas Inválidas!”.

00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00

Digite a coordenada da linha: 0

Digite a coordenada da coluna: 0_

Teste para ver as coordenadas que tem bomba:

LINHA: 09 COLUNA: 06
LINHA: 01 COLUNA: 09
LINHA: 04 COLUNA: 07
LINHA: 08 COLUNA: 04
LINHA: 02 COLUNA: 09
LINHA: 09 COLUNA: 07
LINHA: 10 COLUNA: 02
LINHA: 07 COLUNA: 02
LINHA: 09 COLUNA: 02
LINHA: 04 COLUNA: 06
LINHA: 07 COLUNA: 08
LINHA: 05 COLUNA: 02
LINHA: 05 COLUNA: 01

Resultado esperado para vitória:

VITÓRIA											
								0			
								0			
0			0								
0			0					0			
										0	
					0	0					
		0					0				

Resultado esperado para derrota:

1	DERROTA													
2											o			
3														
4											o			
5	o				o									
6														
7	o			o							o			
8													x	
9														
10							o		o					
11			o								o			