

Projeto – Tabuleiro de Batalha Naval no Terminal

Autoria do programa: William Carrara Orlato – N° USP 11800991

Vitor Ferreira Paschoal – N° USP 11215532

Contato: williamorlato@usp.br

vitor-pas7@usp.br

● Introdução:

O trabalho foi realizado durante o primeiro semestre de 2020, tendo por base o curso de Introdução à Programação, ministrado por Jó Ueyama junto aos auxiliares Erickson Aguiar e Ana Luisa Costa.

Foi proposto desenvolver um jogo de batalha naval em códigos e o tabuleiro aparecer no terminal.

● Objetivos:

Os objetivos desse projeto são desenvolver e aplicar habilidades em programação usando linguagem C.

● Tabuleiro:

A imagem de um tabuleiro 16x16 aparecerá no terminal e nela serão colocados os barcos. Sendo 1 porta-aviões (representado por P) que ocupa 11 blocos, 2 couraçados (representados por C) de 10 blocos cada, 3 torpedeiros (representados por T) ocupando 7 blocos cada e 4 hidroaviões (representados por H) que ocupam 8 blocos cada.

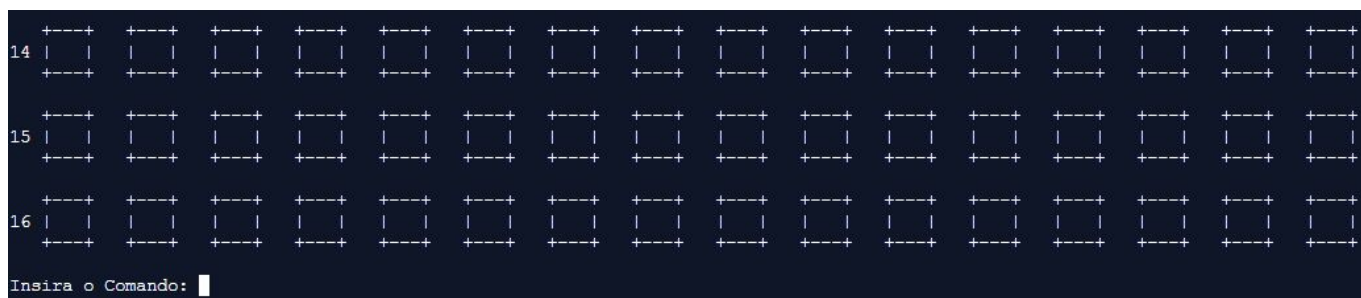
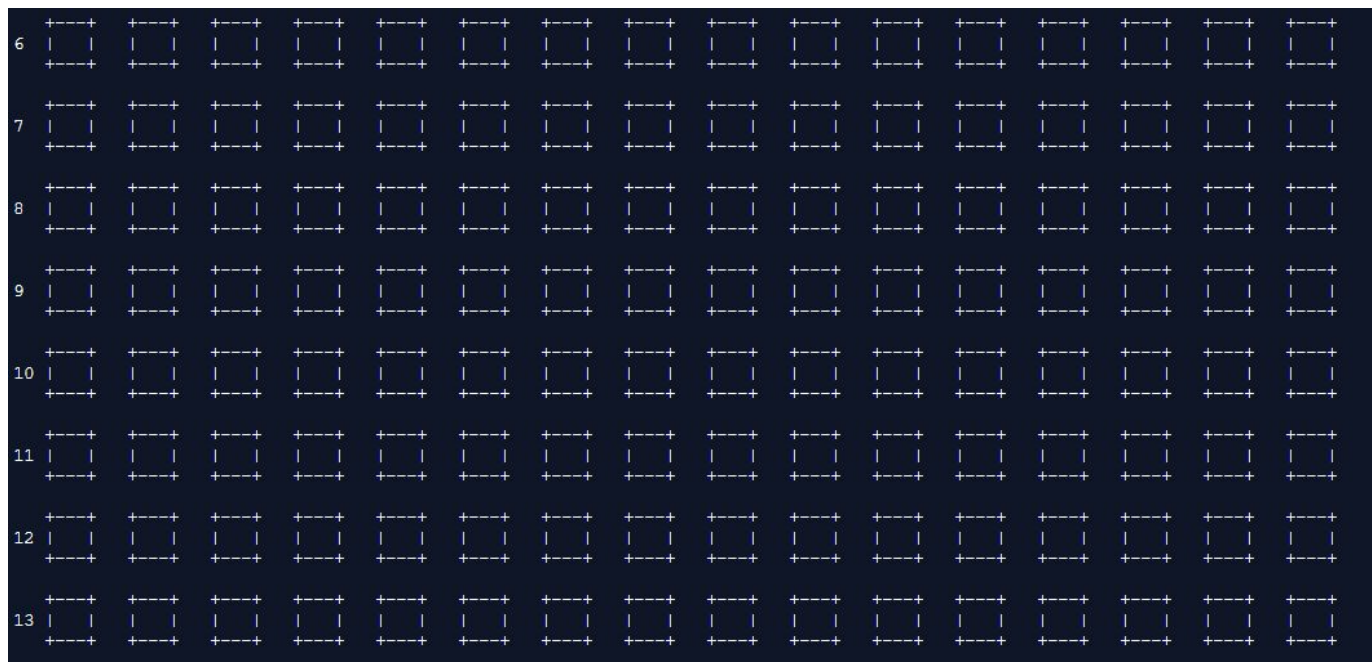
- **Funcionamento:**

Ao colocar o programa para rodar uma imagem como essa deverá aparecer:

```
Bem Vindo a Batalha Naval!  
Precione:  
1-Iniciar  
2-Ajuda  
3-Sair  
Selecione: █
```

Ao digitar 1, aparecerá o tabuleiro:

[illegible]



Onde está indicado por “Insira o comando” é o lugar no qual o usuário deverá indicar o comando desejado.

Ao digitar 2, aparecerá:

```
Pow (coluna) (linha) - Selecciona um lugar para atirar
Reset - Nova partida
Sair - Abandonar o Jogo
Ajuda-Retona a este Menu
Acaso - Inicia um novo tabuleiro mantendo os pontos
Gravar (nome do arquivo) - Salva o jogo
Carregar (nome do arquivo) - Carrega o jogo
Presseccione qualquer tecla para sair:
```

Após pressionar qualquer tecla, o jogador será encaminhado para o menu inicial do jogo.

Para que o jogo ocorra da maneira desejada, o programa se baseia em alguns comandos que direcionam o jogo:

Pow: ao digitar esse comando no terminal, seguido da coluna e da linha desejada, é efetuado um disparo naquela área indicada pela linha e coluna.

Caso o jogador acerte um barco, o espaço vazio será substituído por sua letra representante:

	A	B	C
1	+---+ T +---+	+---+ +---+	+---+ +---+
2	+---+ +---+	+---+ +---+	+---+ +---+

Belo disparo capitao
Tiro no Torpedeiro

Caso o jogador erre, o espaço vazio será marcado com um X:

+---+	+---+
+---+	+---+
+---+	+---+
	X
+---+	+---+

Tiro n'agua

Reset: ao digitar esse comando no terminal é iniciada uma nova partida, é zerada a pontuação e o tempo.

Sair: ao digitar esse comando no terminal o usuário abandona a partida.

Ajuda: ao digitar esse comando no terminal aparecerá nela o menu de ajuda, com os comandos que podem ser digitados inseridos.

Acaso: ao digitar esse comando no terminal as peças do tabuleiro serão reorganizadas de modo aleatório, porém manterá os pontos e o tempo decorrido.

Gravar: ao digitar esse comando no terminal o jogo será salvo na pasta onde o código é executado.

Carregar: ao digitar esse comando no terminal o jogo, que está salvo na pasta do programa, aparecerá no terminal.

Mode 123 (Developer): esse comando é um “especial” dedicado somente aos desenvolvedores. Os usuários não devem ter conhecimento dessa rotina, porque ao digitá-la no terminal os barcos irão aparecer no tabuleiro mesmo sem ser acertados.

Tela do Jogador #1 Pontos:0 Time:19.175955																
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	T	T	T	T	T	T	T									
2		T		H	H	H	H	H	H	H	H					
3		T	P	P	P	P	P	P	P	P	P	P	P			
4		T														
5	T	T										H				
6	T	T										H				
7	T	T										H		H		
8	T	T										H		H		

● Complementares:

Além dos comandos principais, esses complementares aparecem apenas para mostrar quem está ganhando (com uma pontuação) e o tempo decorrido.

```
Tela do Jogador #1 Pontos:1592 Time:25.640425
```

```
Tela do Jogador #2 Pontos:1574 Time:25.641381
```

Os pontos são calculados de acordo com três fórmulas:

PONTOS = ACERTOS - ERROS
ACERTOS = ($ba \cdot qa$) . 1000/N
ERROS = ($N - pt$) . 5/N

N é o número total de posições no tabuleiro, **ba** é o número total de blocos que a peça que foi atingida ocupa, **qa** é a quantidade de peças disponíveis no jogo, para aquela que foi atingida e **pt** indica a quantidade de blocos total que as peças da armada ocupam.

● O programa:

O programa apresenta 10 funções mais a main (rotina de inicialização). Cada rotina pode ser encontrada no programa por uma divisão visual:

```
//=====
//                                     <nome>
//=====
```

Onde no espaço indicado consta o nome de cada uma das funções apresentadas no programa que são:

1. main
2. Menu de Início
3. Comandos
4. Ajuda
5. Iniciar Matrizes
6. Matriz Imagem
7. Escrita_na_matriz
8. Trans
9. Tiro
10. Gravar
11. Carrega

Além disso, o programa apresenta uma função Struct que possui todos os valores de *qa* .(quantidade de peças disponíveis de cada barco).

1.main:

A main possui a função de definir todas as variáveis que serão utilizadas no programa, sendo que, é dela que serão chamadas todas as outras funções que compõem o programa. Além disso, é a partir desta que será realizado as trocas entre os jogadores 1 e 2.

2.Menu de Início:

Tipo	Nome	Parâmetros
int	Menu_de_Inicio	(void)

Essa função inicia o primeiro menu visível ao jogador, onde lá, este poderá ter três escolhas:

1. Iniciar
2. Ajuda
3. Sair

Onde na primeira opção retornará à main, na segunda irá para outra função chamada Ajuda, e na última irá finalizar o programa.

3.Comandos:

Tipo	Nome	Parâmetros
int	Comandos	(char* gravar, int* lc)

Essa função apresenta a leitura de todos os comandos digitados, também fazendo a conversão de dados do tipo char para int, quando necessário. Assim, possibilitando a leitura direta do comando e enviando para outras funções, como os enviados para a função que exercerá o tiro do barco.

4.Ajuda:

Essa função imprimirá na tela todos os comandos necessários para que o usuário consiga inserir o comando desejado.

5.Iniciar Matrizes:

Tipo	Nome	Parâmetros
void	Iniciar_matrizes	(int campo1[16][16],int campo2[16][16])

Essa função coloca as identificações dos barcos nas posições geradas pela função Escrita_na_matriz.

6.Matriz imagem:

Tipo	Nome	Parâmetros
void	Matriz_imagem	(int campo[16][16], int jogador,char imgcampo[16][16], int pontos, double tempo)

Essa função expõe os tabuleiros vazios no terminal.

7.Escrita nas Matrizes:

Tipo	Nome	Parâmetros
void	Escrita_na_matriz	(int mat[16][16], int N, int iden)

Essa função tem por objetivo gerar valores aleatórios de começo eles irão definir se o barco será colocado na horizontal ou na vertical, depois se será escrito para esquerda, para direita, para cima ou para baixo.

Além disso, irá verificar se o barco designado pela função anterior caberá no espaço gerado, caso não caiba, uma nova posição será gerada até ter espaço suficiente.

8.Trans:

Tipo	Nome	Parâmetros
void	Trans	(int campo[16][16],char imgcampo[16][16])

Essa função transforma os números de indicação dos barcos em letras, ou seja, caso seja ativada, o tabuleiro irá mostrar as posições de todos os barcos.

9.Tiro:

Tipo	Nome	Parâmetros
int	Tiro	(int dados[], int campo[16][16], char imgcampo[16][16], int* pontos, Barco* qa, int* pt)

Essa é a função a qual define quase todo o jogo, nela são separados 5 casos:

0 -> Erro: caso o tiro não acerte nenhum barco, será contabilizado o erro, aparecerá 'X' no espaço do tiro e a mensagem "tiro n'água" aparecerá para o jogador.

1 -> Acerto no Porta-aviões: caso o tiro acerte o porta-aviões, será contabilizado o ponto, aparecerá 'P' no espaço do tiro e as mensagens "Belo disparo capitão" e "tiro no porta-aviões" aparecerão para o jogador.

2 -> Acerto no Couraçado: caso o tiro acerte o couraçado, será contabilizado o ponto, aparecerá 'C' no espaço do tiro e as mensagens "Belo disparo capitão" e "tiro no couraçado" aparecerão para o jogador.

3 -> Acerto no Torpedeiro: caso o tiro acerte o torpedeiro, será contabilizado o ponto, aparecerá 'T' no espaço do tiro e as mensagens "Belo disparo capitão" e "tiro no torpedeiro" aparecerão para o jogador.

4 -> Acerto no Hidroavião: caso o tiro acerte o hidroavião, será contabilizado o ponto, aparecerá 'H' no espaço do tiro e as mensagens "Belo disparo capitão" e "tiro no hidroavião" aparecerão para o jogador.

10.Gravar:

Tipo	Nome	Parâmetros
void	Gravar	(char* gravar, int* pontos1, int* pontos2, double* tempo, int* aux, int campo1[16][16], int campo2[16][16], char imgcampo1[16][16], char imgcampo2[16][16])

Essa função grava as informações do programa:

Pontos do Jogador 1
Pontos do Jogador 2
Tempo
Vez do jogador
Matriz de Dados do Jogador 1
Matriz de Dados do Jogador 2
Matriz Imagem do Jogador 1
Matriz Imagem do Jogador 2

Salvando todas essas informações em binário em um arquivo .dat que não poderá ser lida por nenhum jogador. Além disso, salvará a vez de quem irá jogar.

11.Carregar:

Tipo	Nome	Parâmetros
void	Carrega	(char* gravar, int* pontos1, int* pontos2, double* tempo, int* aux, int campo1[16][16], int campo2[16][16], char imgcampo1[16][16], char imgcampo2[16][16])

Essa função carrega as informações de um arquivo .dat salvo anteriormente pelo programa carregando o:

Pontos do Jogador 1
Pontos do Jogador 2
Tempo
Vez do jogador
Matriz de Dados do Jogador 1
Matriz de Dados do Jogador 2
Matriz Imagem do Jogador 1
Matriz Imagem do Jogador 2

Após carregado o jogo retorna da onde foi gravado pela última vez.