

UNIVERSIDADE FEDERAL DO PARANÁ

## BATALHA NAVAL

Documento com os requisitos do trabalho para a disciplina CI067 - Oficina de Computação do Departamento de Informática da Universidade Federal do Paraná.

Professora: Silvia Regina Vergilio.

Área de concentração: *Ciência da Computação*.

CURITIBA PR

2017

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Componentes do Jogo</b>	<b>2</b>
2.1	Matriz de células . . . . .	2
2.2	Embarcações . . . . .	2
2.3	Mecânica do Jogo . . . . .	3
2.4	Jogadas . . . . .	4
2.5	Resultados . . . . .	4
2.6	Posicionamento das embarcações . . . . .	4
2.6.1	Quem começa? . . . . .	5
2.6.2	Comportamentos do jogo . . . . .	5
<b>3</b>	<b>Condições de entrega e avaliação</b>	<b>7</b>
3.1	Condições de entrega . . . . .	7
3.2	Avaliação . . . . .	7
<b>A</b>	<b>Dicas</b>	<b>9</b>
A.1	Random . . . . .	9

# Capítulo 1

## Introdução

O bem conhecido jogo da Batalha Naval é um jogo para 2 jogadores, jogado com papel e lápis na sua forma original. A história deste jogo antes da sua comercialização não é fácil de coligir, dadas as múltiplas versões existentes na Web. Ao que parece foi inventado por Clifford Von Wickler nos primeiros anos do século XX (ainda antes da 1<sup>a</sup> Guerra Mundial), o qual nunca o patenteou. No entanto, outras versões apresentam o jogo já em uso por militares franceses no mesmo período. O jogo foi comercializado em 1931 pela empresa Starex Novelty Co. sob o nome *Salvo*. Em 1943 a empresa Milton Bradley Company voltou a comercializá-lo ainda como jogo de papel e lápis, e em 1967 como jogo de tabuleiro.

No jogo Batalha Naval, cada jogador dispõe no início a sua **frota** numa matriz de células com dimensão 10x10. As células livres representam **água**.

Os jogadores jogam alternadamente, sendo o objectivo de cada jogador adivinhar onde se encontram as embarcações do adversário, i.e., onde se encontram todas as células ocupadas do adversário. No contexto do jogo, isto é referido como afundar as embarcações do adversário. Na versão mais simples, quando chega a sua vez de jogar, o jogador identifica a célula para onde quer disparar fornecendo as respectivas coordenadas (i.e., B7). O adversário, por sua vez, diz-lhe qual foi o resultado do disparo: tiro ou água (caso a célula pedida se encontre ocupada ou desocupada, respectivamente). No caso de tiro, o adversário também indica o tipo de embarcação atingida. Quando a última célula de uma embarcação é atingida, o respectivo jogador avisa tiro, embarcação ao fundo. O jogador que consegue atingir todas as posições ocupadas do adversário em primeiro lugar, ganha o jogo.

No projeto pretende-se implementar um jogo de Batalha Naval em linguagem C. O jogador utiliza o programa para jogar contra outro jogador ou contra a máquina. A máquina apenas escolhe aleatoriamente uma coordenada que ainda não tenha escolhido, tentando adivinhar onde voc colocou a frota do seu jogo. Neste caso, o objetivo do jogador é minimizar o número de jogadas com que adivinha as posições das embarcações. As 10 melhores pontuações (número mínimos de disparos) são disponibilizadas em um Hall da Fama.

## Capítulo 2

### Componentes do Jogo

Os componentes do jogo consistem numa matriz de células e num conjunto de gráficos que representam embarcações a ser dispostos na matriz.

#### 2.1 Matriz de células

A matriz tem dimensão 10x10, em que as linhas são identificadas por letras (A..J) e as colunas por números no intervalo 0..9 (ver matriz abaixo).

	0	1	2	3	4	5	6	7	8	9
A										
B										
C										
D										
E										
F										
G										
H										
I										
J										

No jogo tradicional, cada jogador utiliza um par de matrizes, uma para dispôr a sua frota e outra em que marca os resultados dos disparos que efectua, assinalando também a posição das embarcações adversários já **afundadas** e células necessariamente desocupadas em seu redor - torna-se inútil efetuar disparos para estas posições que se sabem desocupadas à partida.

#### 2.2 Embarcações

Embora existam diversas variantes da Batalha Naval, será utilizada como referência uma das versões mais populares, a qual define as embarcações apresentadas na Tabela 2.1. Nessa

Tabela 2.1: Descrição das embarcações

Embarcação	Sigla	Tamanho	Quantidade
Porta-aviões	P	5	1
Embarcação de Guerra	N	4	1
Cruzador	C	3	1
Submarino	S	3	1
Destruidor	D	2	1

tabela são apresentados: o tipo de embarcação, a sigla da embarcação, o tamanho de células que ocupará na matriz e a quantidade desse tipo de embarcação que estará presente na matriz.

Ao serem colocados na matriz, o jogador poderá definir qual a orientação que cada embarcação poderá assumir apenas em múltiplos de  $90^\circ$ . Assim, serão 4 direções: acima, abaixo, direita ou esquerda. As embarcações podem ser colocadas em qualquer local dentro da matriz (linhas A a J, colunas 0 a 9), desde que não se sobreponham.

## 2.3 Mecânica do Jogo

Ao iniciar o jogo o programa deverá mostrar as seguintes opções de menu:

- 1) Jogar
- 2) Hall da Fama
- 3) Sobre o jogo
- 4) Sair

A opção de menu **2) Hall da Fama** mostrará um ranking das 10 melhores pontuações realizadas até o momento. O jogo **não** precisa salvar nenhum ranking em arquivo para carregar a cada nova execução do programa, assim, o ranking permanece apenas para aquela execução do programa e sendo reiniciada a cada nova execução. A opção de menu **3) Sobre o jogo** conterá informações sobre como jogar o jogo, autor, etc. A opção de menu **4) Sair** fechará o jogo.

Ao acessar a opção de menu **1) Jogar** o programa deverá mostrar as seguintes opções de menu:

- 1) PvP
- 2) PvCOM
- 3) Voltar

Sendo que a opção de menu **1) PvP** iniciará um jogo entre duas pessoas (*players*) e a opção de menu **2) PvCOM** iniciará um jogo contra o computador. A opção de menu **3) Voltar** voltará para as opções de menu anterior (1 ao 4).

Ao iniciar uma batalha naval, opções de menu **1) PvP** ou **2) PvCOM**, cada jogador deverá:

1. Informar seu nome. No caso do computador deverá ser estabelecido como **COM**.

2. Posicionar todos os seus embarcações. No caso do computador os embarcações deverão ser posicionados aleatoriamente.

Lembrando que a batalha ocorrerá numa área fixa, como mencionado anteriormente, utilizando uma matrix 10x10.

## 2.4 Jogadas

Os jogadores jogam um de cada vez. Cada jogada corresponde ao envio de apenas um tiro (disparo). Não são permitidos disparos repetidos, i.e., atingindo uma célula já antes atingida. Quando o jogador atingir todas as embarcações acaba o jogo.

## 2.5 Resultados

O programa deve mostrar as seguintes informações de cada jogador:

1. Número total de disparos realizados;
2. Número de disparos na água;
3. Número de disparos em embarcações, por tipo de embarcação (na ordem inversa apresentada na Tabela 2.1, ou seja, começando pelo Destruído), e omitindo os tipos para os quais não foi atingido qualquer barco;
4. Número de embarcações afundadas, por tipo de embarcação (na ordem inversa apresentada na Tabela 2.1, ou seja, começando pelo Destruído), e omitindo os tipos para os quais não foi afundado qualquer embarcação.

## 2.6 Posicionamento das embarcações

Quando o jogo começa, o programa deverá mostrar a matriz vazia e solicitar ao jogador para colocar as embarcações. A colocação só pode terminar depois de colocados todas as embarcações da frota. Sendo que, toda embarcação deve estar completamente na matriz, nenhum embarcação pode estar com alguma parte do lado de fora.

Durante a colocação o jogador deve informar para a embarcação que está colocando uma coordenada inicial, por exemplo G1, e solicitar a direção que deve ser colocado a embarcação. Além disso, o usuário deve confirmar se deseja posicionar a embarcação naquela direção para aquela coordenada. As direções possíveis são:

- W: Para Cima

- **S:** Para Baixo
- **D:** Para Direita
- **A:** Para Esquerda

Abaixo é mostrado um exemplo:

```
Informe uma coordenada para posicionar o "Porta-aviões": G1
Informe a direção que o "Porta-aviões" deve ser posicionado: W
Deseja confirmar a coordenada "G1" com direção "Para Cima"
para a embarcação "Porta-aviões"? Sim
```

E a saída será:

	0	1	2	3	4	5	6	7	8	9
A										
B										
C			P							
D			P							
E			P							
F			P							
G			P							
H										
I										
J										

### 2.6.1 Quem começa?

Depois de colocados as embarcações para todos os jogadores o programa deve determinar quem é o primeiro a jogar. O jogador 1 deve escolher cara ou coroa, então, o programa irá "lançar" uma moeda. Quem vencer decide se deseja começar ou não.

### 2.6.2 Comportamentos do jogo

Inicialmente, a matriz a ser mostrada ao jogador deve estar vazia, pode-se utilizar um ponto "." para auxiliar na visualização. Se um jogador erra um tiro, mostrar na matriz como um "O" o local que errou. Se um jogador acertar um tiro, deve mostrar um "X" no local.

Quando um jogador atingir uma coordenada que não contenha nenhuma embarcação o programa deve mostrar a mensagem "**Água!**". Se atingir uma coordenada com uma embarcação, o programa deve mostrar a mensagem "**Fogo!**" e também deve anunciar "qual" embarcação foi atingido (por exemplo, o porta-aviões).

Se cada célula de uma embarcação levar um tiro, a embarcação será afundada. O programa deve então informar dizendo "**Afundou meu \_\_\_\_!**" indicando qual tipo de embarcação

foi afundada. Além disso, o programa deve mostrar em cada posição a inicial do que derrubou, por exemplo, N de embarcação de guerra.

O exemplo abaixo ilustra uma situação em que foi afundado um Porta-aviões, alguns tiros acertaram algumas embarcações e outras atingiram a água:

	0	1	2	3	4	5	6	7	8	9
A	.   .   .   .   .   .   .   .   .   .									
B	.   .   .   .   .   .   .   X   X   .   .									
C	O   P   O   .   .   .   .   .   .   .									
D	.   P   .   .   .   .   .   .   .   .									
E	.   P   .   .   .   .   .   .   .   .									
F	.   P   .   .   .   .   .   O   .   .   .									
G	.   P   .   .   .   .   .   .   X   .   .									
H	.   .   .   .   .   O   .   .   .   .   .									
I	.   .   .   .   .   .   .   .   .   .   .									
J	.   .   .   .   .   .   .   .   .   .   O									

Ao final de cada jogo o programa deve informar o vencedor e perguntar se o usuário deseja começar um novo jogo ou voltar ao menu inicial.



## Capítulo 3

# Condições de entrega e avaliação

A seguir são descritas as condições de entrega do trabalho e procedimento de avaliação.

### 3.1 Condições de entrega

A resolução deste trabalho deve ser entregue até as 23:59h de domingo, 24 de Setembro de 2017 (horário de Brasília). A resolução deve ser entregue via e-mail para [silvia@inf.ufpr.br](mailto:silvia@inf.ufpr.br), contendo no assunto [**Trabalho-OficinaC**] e o nome do aluno, exemplo, [**Trabalho-OficinaC**] **Silvia Regina Vergilio**. O nome do aluno também deve estar no corpo do e-mail tendo em anexo um arquivo zip contendo apenas os arquivos fonte onde se encontra o código C (extensão .c). Deverá ser entregue também um relatório em pdf, utilizando o modelo SBC de publicação de artigos (<http://www.sbc.org.br/documentos-da-sbc/category/169-templates-para-artigos-e-capitulos-de-livros>), que conterá obrigatoriamente a listagem do código C e um pequeno texto explicando as opções tomadas para a resolução.

A entrega fora do prazo implica a penalização de um ponto por cada dia útil de atraso (i.e., 25 de Setembro -1 ponto, 26 de Setembro -2 pontos, 27 de Setembro -3 pontos, 28 de Setembro -4 pontos e 29 de Setembro -5 pontos) não se aceitando trabalhos após quinta-feira, 29 de Setembro de 2017, até às 23:59h (horário de Brasília).

As datas das defesas dos trabalhos serão anunciadas logo que possível.

### 3.2 Avaliação

A avaliação do trabalho terá em conta não só a correcta execução do programa, mas também, e principalmente, a correcta estruturação dos métodos, funções e procedimentos que compõem o programa e ainda a sua legibilidade. Valorizar-se-á mais a simplicidade que a eficiência. Valorizar-se-á também código C correctamente comentado (mas não excessivamente, não vale a pena comentar o óbvio).

A nota deste trabalho será dada apenas após uma discussão, individual, com cada aluno. Nesta discussão o aluno terá de demonstrar um total conhecimento do programa e ser capaz de operar as alterações que forem pedidas. Nessa apresentação serão também feitas perguntas sobre a matéria em geral. A nota final dependerá não só da qualidade do trabalho, mas também, e principalmente, do conhecimento do mesmo programa e da matéria em geral e da capacidade de resolver problemas em C demonstrados nessa discussão.

Quaisquer funcionalidades extra que não tenham sido pedidas no enunciado, tais como menus mais sofisticados, utilização de arquivos, etc., não serão forçosamente avaliadas.

# Apêndice A

## Dicas

### A.1 Random

A função *rand()* da biblioteca *<stdlib.h>* retorna um pseudo-random inteiro entre 0 e *RAND\_MAX*. Voc pode utilizar *srand(sem seed informado)* para determinar um *seed* (um padrão de números aleatórios).

É uma prática comum utilizar o operador *%* em conjunto com a função *rand()* para pegar uma faixa diferente. Por exemplo:

```
1 #include <time.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 srand(time(NULL)); // deve ser chamado apenas uma vez
6 int r = rand() % 10; // retorna um pseudo-random inteiro entre 0 e 9
```

Outro exemplo:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(){
6     time_t t;
7
8     // Inicializa o gerado de número aleatório
9     srand((unsigned) time(&t));
10
11     /* Mostra 5 números elatórios entre 0 e 49 */
12     for(int i = 0; i < 5; i++){
13         printf("%d\n", rand() % 50);
14     }
15     return(0);
16 }
```