

Algoritmos
Jogo de Caça ao Submarino
Trabalho Final

JOGO DE CAÇA AO SUBMARINO

No jogo de caça ao submarino, o algoritmo controla um submarino e o jogador um navio de guerra chamado couraçado. O objetivo do jogo é uma embarcação destruir a outra. No entanto, as embarcações possuem capacidade de radar limitada. A única maneira de uma embarcação encontrar a outra é estar na mesma posição: o couraçado sobre o submarino e o submarino sob o couraçado. O vencedor é aquele que encontrar o outro primeiro, com o couraçado e o submarino alternando movimentos aleatórios no “oceano”.

O “oceano” é um campo de jogo pequeno com nove posições, que estão dispostas como uma matriz de três linhas e três colunas. O couraçado e o submarino tratam esse campo de jogo como um sistema de coordenadas cartesianas, com o par ordenado indicado por (0, 0) sendo considerado como o canto superior esquerdo.

Elabore um **algoritmo** para o jogo de caça ao submarino, considerando os seguintes itens:

- a) Utilize uma matriz de caracteres 3 x 3 para representar o “oceano” (campo de jogo). Cada elemento da matriz (indicado pela intersecção da respectiva linha e coluna) corresponde a uma parte do campo de jogo onde podem estar as embarcações.
- b) Apresente o campo de jogo antes da primeira jogada com “.” para representar cada parte do campo, conforme o seguinte exemplo:

```
  0  1  2
0  .  .  .
1  .  .  .
2  .  .  .
```

- c) Depois de apresentar o campo pela primeira vez, determine a posição do submarino por meio da função predefinida **rand()**, que é um gerador de números inteiros aleatórios. Como sugestão, para obter um número aleatório entre 0 e 2 para a linha e coluna da posição do submarino pode-se utilizar os seguintes comandos:

```
lin = rand() % 3;
col = rand() % 3;
```

Observação: Para produzir números inteiros aleatórios diferentes a cada execução do jogo, antes de começar é necessário inicializar a função **rand()** por meio da função **srand()**, utilizando um valor diferente como argumento, o que pode ser feito com o seguinte comando:

```
srand(time(NULL));
```

Para utilizar as funções **rand()** e **srand()** é necessário o arquivo de cabeçalho **stdlib.h** e para utilizar a função **time()** é necessário o arquivo **time.h**.

- d) A cada jogada, a linha e a coluna correspondente à posição do couraçado devem ser solicitadas ao jogador.
- e) Depois de cada jogada, apresente novamente o campo e indique a posição do submarino pela letra "S" e o couraçado pela letra "C". Por exemplo, o seguinte campo seria apresentado se a linha e coluna informadas pelo jogador fossem 2 (linha) e 1 (coluna) e o submarino estivesse na posição 0 (linha) e 1 (coluna).

	0	1	2
0	*	S	*
1	*	*	*
2	*	C	*

- f) O jogador não pode informar números de linha e coluna fora do intervalo de 0 a 2. Se o jogador digitar um número não permitido de linha ou coluna, uma mensagem deve ser mostrada e o número deve ser solicitado novamente (técnica de crítica de dados). Por exemplo, para o campo anterior, não poderiam ser informadas os seguintes pares de linha e coluna: 3 e 1 (linha fora do intervalo); 0 e 5 (coluna fora do intervalo); 3 e 3 (linha e coluna fora do intervalo).
- g) Depois de cada jogada, é necessário verificar se o couraçado conseguiu destruir o submarino, isto é, se a linha e coluna informada pelo jogador coincide com a linha e coluna do submarino. Caso positivo, uma mensagem deve ser mostrada.
- h) Depois de cada jogada, se o couraçado não conseguiu destruir o submarino, determine a nova posição do submarino por meio da função predefinida **rand()**. Em seguida, é necessário verificar se o submarino conseguiu destruir o couraçado, isto é, se a linha e coluna determinada para o submarino coincide com a linha e coluna do couraçado. Caso positivo, uma mensagem deve ser mostrada.
- i) Ao término do jogo, perguntar ao usuário se deseja jogar novamente. Caso positivo, repetir o jogo desde o início.