

Pedro Inácio Rodrigues Pontes

Prática 7: Batalha Naval em Rede

Belo Horizonte, Brasil

2025

1 Introdução

Objetivo: Desenvolver dois programas de Console Application em C que se comuniquem via TCP para simular um jogo de Batalha Naval em um tabuleiro 10×10, com 10 navios de tamanho 1 posicionados pelo Servidor (Player 1) e ataques enviados pelo Cliente (Player 2).

2 Desenvolvimento

2.1 Infraestrutura

Como base para o jogo, foram criadas classes na pasta BatalhaNaval.Core:

- **Coordinate**

Representa as coordenadas. Pode receber uma tupla os inteiros linhas e coluna ou uma string no formato letra + número em seu construtor. Possui as variáveis *Row*, *Column* e *Value* (tupla com as duas variáveis anteriores. Possui o método *MapValue* que converte a string dada no construtor para o valor suportado em *Value*

- **TableBase**

Classe abstrata que será herdada pelas classes que representam as mesas do player um e dois. Possui os métodos *Initialilze*, *Show* (mais seus complementares) e *UpdateCell*

- **IConsole**

Interface do Console com os métodos *textitWrite*, *textitRead* e *textitReadLine*. Feita como injeção de dependência para facilitar os testes unitários, que testei uso nesse exercício.

- **IMessageStream**

Interface do *NetworkStream*. Feita para permitir injeção de dependência que vem a facilitar testes unitários. Possui os métodos *Send*, *Receive* e *Close*

- **SystemConsole**

Implementação do *IConsole*

- **NetwrokMessageStream**

Implementação do *NetworkStream*

2.2 Player1

Representa a logística do Player 1, o qual posiciona os navios no tabuleiro, de forma automática ou manual, além de ser o servidor TCP no qual o Player 2 irá entrar.

- Menu

Representa o menu inicial, no qual o player recebe as boas-vindas e pode escolher entre o posicionamento manual e automático. Possui os métodos *Show*, *ShowInitialMessage*, *ProcessOption* e *ReadOption*

- Ship

Representa um navio no jogo. Tem as propriedades *Coords* (do tipo *Coordinate*) e *Sink* (bool). Possui o método *IsHit*, o qual é autodescritivo e booleano.

- Server

Cria o servidor TCP. Métodos *Send*, *Receive*, *Close* e *AcceptConnection* (utilizado para iniciar o servidor fora do ambiente de testes)

- Table

Representa o tabuleiro do jogo. Tem métodos para criar a mesa com posições aleatórias ou manuais e receber o ataque do player2 e retornar a mensagem de feedback quanto ao acerto, erro ou vitória, além dos métodos complementares para que as funções funcionem corretamente.

- Program

Implementa a solução. Mostra o menu, inicia o server e roda uma sequência de receber a mensagem do player2, processá-la e retornar o feedback a ele.

2.2.1 Demonstrações

Implementação de Table (apenas os métodos públicos):

```
public class Table : TableBase
{
    private const int MaxShips = 10;
    private readonly Random _rand;
    int _shipsNumber = 0;
    Ship[] _ships;

    public Table(IConsole console, Random? random = null) :
        base(console)
    {
        _rand = random ?? new Random();
        _ships = new Ship[MaxShips];
    }
}
```

```
public void CreateTableWithRandomShipsPositions()
{
    AddShipsRandomly();
    Show();
}

public void CreateTableWithManualShipPositions()
{
    AddShipsManually();
    Show();
}

public string ReceiveAttack(string attack)
{
    Coordinate attackCoordinates = new
        Coordinate(attack);
    foreach (var ship in _ships)
    {
        if (ship.IsHit(attackCoordinates))
        {
            _table[ship.Coords.Row, ship.Coords.Column]
                = 'X';
            Show();
            return IsGameWin() ? "WIN" : "HIT";
        }
    }
    _table[attackCoordinates.Row,
        attackCoordinates.Column] = '0';
    Show();
    return "MISS";
}

public bool IsGameWin()
{
    return _ships.Take(_shipsNumber).All(ship =>
        ship.Sink);
}
}
```

```
var console = new SystemConsole();
var table = new Table(console);
var menu = new Menu(console, table);

menu.Show();

var server = new Server(15000);

while (!table.IsGameWin())
{
    var message = server.Receive();

    var attackStatus = table.ReceiveAttack(message);
    server.Send(attackStatus);
}
```

2.3 Player2

Implementa a dinâmica de ataque e estabelece conexão de cliente com o server do player1.

Seguem suas classes:

- Client

Estabelece a conexão cliente com o server player1.

- AttackTable

Implementa a lógica de ataque e exibição do tabuleiro.

- Program

Implementa a solução. Cria o cliente TCP e utiliza o método Play de AttackTable

2.3.1 Demonstrações

Implementação de AttackTable (apenas os métodos públicos):

```
public class AttackTable : TableBase
{
    List<Coordinate> attackedCells = new();
    Client _client;
    string lastAttackStatus = "";
}
```

```
public AttackTable(IConsole console, Client client) :
    base(console)
{
    _client = client;
}

public void Play()
{
    Show();
    while (lastAttackStatus != "WIN")
    {
        var coordinate = ReadAttackCoordinate();
        if (IsValidCoordinate(coordinate))
        {
            SendCoordinateToServer(coordinate);
            VerifyAttackStatus(coordinate);
        }
    }
}
}
```

Implementação de Program

```
IConsole console = new SystemConsole();
console.WriteLine("Bem vindo ao jogo Batalha Naval!");
Client client = new("127.0.0.1", 15000);
AttackTable table = new(console, client);
table.Play();
client.Close();
```

3 Resultados

```
Bem-vindo à Batalha Naval!  
Escolha uma das opções para o posicionamento dos navios  
1. Posicionamento Aleatório  
2. Posicionamento Manual  
1  
  A   B   C   D   E   F   G   H   I   J  
1  ~   ~   ~   ~   ~   ~   ~   ~   ~   ~  
2  ~   ~   ~   ~   ~   ~   ~   ~   *   ~  
3  ~   ~   ~   ~   *   ~   ~   ~   ~   ~  
4  ~   ~   ~   *   ~   ~   ~   ~   ~   ~  
5  ~   ~   ~   ~   ~   ~   ~   ~   ~   ~  
6  ~   ~   ~   ~   ~   ~   ~   ~   ~   ~  
7  ~   ~   ~   ~   ~   ~   *   ~   ~   ~  
8  ~   ~   ~   ~   ~   *   *   *   ~   ~  
9  ~   ~   ~   ~   ~   ~   ~   ~   ~   ~  
10 *   ~   ~   ~   ~   ~   ~   ~   *   *   ~  
Aguardando Player2 na porta 15000...
```

Figura 1 – layer1-posicionamento-aleatorio.png

```
Bem-vindo à Batalha Naval!
Escolha uma das opções para o posicionamento dos navios
1. Posicionamento Aleatório
2. Posicionamento Manual
2
Escreva as coordenadas dos 10 navios
a1
a2
a3
a4
a5
a6
a6
Erro: Coordenada repetida
asdtfgae
Erro: Coordenada fora do intervalo 1-10
a7
00
Erro: Coordenada fora do intervalo A-J
a8
a9
a10
```

	A	B	C	D	E	F	G	H	I	J
1	*	~	~	~	~	~	~	~	~	~
2	*	~	~	~	~	~	~	~	~	~
3	*	~	~	~	~	~	~	~	~	~
4	*	~	~	~	~	~	~	~	~	~
5	*	~	~	~	~	~	~	~	~	~
6	*	~	~	~	~	~	~	~	~	~
7	*	~	~	~	~	~	~	~	~	~
8	*	~	~	~	~	~	~	~	~	~
9	*	~	~	~	~	~	~	~	~	~
10	*	~	~	~	~	~	~	~	~	~

```
Aguardando Player2 na porta 15000...
Player2 conectado!
```

Figura 2 – player1-posicionamento-manual.png


```

Bem vindo ao jogo Batalha Naval!
Conectado ao servidor!

      A      B      C      D      E      F      G      H      I      J
1      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
2      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
3      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
4      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
5      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
6      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
7      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
8      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
9      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
10     ~      ~      ~      ~      ~      ~      ~      ~      ~      ~

Escreva a coordenada de ataque
j1
      A      B      C      D      E      F      G      H      I      J
1      ~      ~      ~      ~      ~      ~      ~      ~      ~      0
2      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
3      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
4      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
5      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
6      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
7      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
8      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
9      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
10     ~      ~      ~      ~      ~      ~      ~      ~      ~      ~

Escreva a coordenada de ataque
a1
      A      B      C      D      E      F      G      H      I      J
1      X      ~      ~      ~      ~      ~      ~      ~      0
2      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
3      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
4      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
5      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
6      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
7      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
8      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
9      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
10     ~      ~      ~      ~      ~      ~      ~      ~      ~      ~

Escreva a coordenada de ataque

```

Figura 3 – player2-inicio-jogo

Player2 conectado!

	A	B	C	D	E	F	G	H	I	J
1	*	~	~	~	~	~	~	~	~	0
2	*	~	~	~	~	~	~	~	~	~
3	*	~	~	~	~	~	~	~	~	~
4	*	~	~	~	~	~	~	~	~	~
5	*	~	~	~	~	~	~	~	~	~
6	*	~	~	~	~	~	~	~	~	~
7	*	~	~	~	~	~	~	~	~	~
8	*	~	~	~	~	~	~	~	~	~
9	*	~	~	~	~	~	~	~	~	~
10	*	~	~	~	~	~	~	~	~	~
	A	B	C	D	E	F	G	H	I	J
1	X	~	~	~	~	~	~	~	~	0
2	*	~	~	~	~	~	~	~	~	~
3	*	~	~	~	~	~	~	~	~	~
4	*	~	~	~	~	~	~	~	~	~
5	*	~	~	~	~	~	~	~	~	~
6	*	~	~	~	~	~	~	~	~	~
7	*	~	~	~	~	~	~	~	~	~
8	*	~	~	~	~	~	~	~	~	~
9	*	~	~	~	~	~	~	~	~	~
10	*	~	~	~	~	~	~	~	~	~
	A	B	C	D	E	F	G	H	I	J
1	X	~	~	~	~	~	~	~	~	0
2	X	~	~	~	~	~	~	~	~	~
3	*	~	~	~	~	~	~	~	~	~
4	*	~	~	~	~	~	~	~	~	~
5	*	~	~	~	~	~	~	~	~	~
6	*	~	~	~	~	~	~	~	~	~
7	*	~	~	~	~	~	~	~	~	~
8	*	~	~	~	~	~	~	~	~	~
9	*	~	~	~	~	~	~	~	~	~
10	*	~	~	~	~	~	~	~	~	~
	A	B	C	D	E	F	G	H	I	J

Figura 4 – player1-inicio-jogo

```

Escreva a coordenada de ataque
a8
      A      B      C      D      E      F      G      H      I      J
1      X      ~      ~      ~      ~      ~      ~      ~      ~      O
2      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
3      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
4      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
5      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
6      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
7      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
8      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
9      ~      ~      ~      ~      ~      ~      ~      ~      ~      ~
10     ~      ~      ~      ~      ~      ~      ~      ~      ~      ~

Escreva a coordenada de ataque
a9
      A      B      C      D      E      F      G      H      I      J
1      X      ~      ~      ~      ~      ~      ~      ~      ~      O
2      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
3      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
4      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
5      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
6      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
7      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
8      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
9      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
10     ~      ~      ~      ~      ~      ~      ~      ~      ~      ~

Escreva a coordenada de ataque
a10
Jogo Ganho!
      A      B      C      D      E      F      G      H      I      J
1      X      ~      ~      ~      ~      ~      ~      ~      ~      O
2      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
3      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
4      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
5      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
6      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
7      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
8      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
9      X      ~      ~      ~      ~      ~      ~      ~      ~      ~
10     ~      ~      ~      ~      ~      ~      ~      ~      ~      ~

```

Figura 5 – player2-fim-jogo

	A	B	C	D	E	F	G	H	I	J
1	X	~	~	~	~	~	~	~	~	0
2	X	~	~	~	~	~	~	~	~	~
3	X	~	~	~	~	~	~	~	~	~
4	X	~	~	~	~	~	~	~	~	~
5	X	~	~	~	~	~	~	~	~	~
6	X	~	~	~	~	~	~	~	~	~
7	X	~	~	~	~	~	~	~	~	~
8	X	~	~	~	~	~	~	~	~	~
9	X	~	~	~	~	~	~	~	~	~
10	*	~	~	~	~	~	~	~	~	~
	A	B	C	D	E	F	G	H	I	J
1	X	~	~	~	~	~	~	~	~	0
2	X	~	~	~	~	~	~	~	~	~
3	X	~	~	~	~	~	~	~	~	~
4	X	~	~	~	~	~	~	~	~	~
5	X	~	~	~	~	~	~	~	~	~
6	X	~	~	~	~	~	~	~	~	~
7	X	~	~	~	~	~	~	~	~	~
8	X	~	~	~	~	~	~	~	~	~
9	X	~	~	~	~	~	~	~	~	~
10	X	~	~	~	~	~	~	~	~	~

Figura 6 – player1-fim-jogo

4 Conclusão

Todos os resultados foram alcançados. A implementação de testes unitários fez a criação da solução se estender por horas a mais do que deveria, pois estes forçam o código a estar bem feito. Tal coisa gerou muito trabalho a mais.