



**BOSCH**

Invented for life

## Docupedia Export

Author: Goncalves Donathan (CtP/ETS)  
Date: 11-May-2023 13:32

## Table of Contents

<b>1 Desafio 1: Simulação na Partida de War com Monte Carlo</b>	<b>3</b>
---	----------

# 1 Desafio 1: Simulação na Partida de War com Monte Carlo

Monte Carlo é um método estatístico onde para obter uma probabilidade, esperança matemática ou outra informação repetimos um experimento computacionalmente (ou não) quantas vezes for necessário. Por exemplo, para descobrir se a probabilidade de um dado obter 6 é realmente ~16.7%, basta simular um dado com a classe Random milhões de vezes e ver quantas delas se obteve um 6:

```
1  using System;
2
3  Random rand = new Random();
4  int N = 1_000_000;
5
6  monteCarlo();
7
8  int roll()
9      => rand.Next(6) + 1;
10
11 void monteCarlo()
12 {
13     int count = 0;
14     for (int i = 0; i < N; i++)
15     {
16         if (roll() == 6)
17             count++;
18     }
19     Console.WriteLine(count / (float)N); // ~0,167
20 }
```

Seu desafio é criar um programa de computador que para grandes exercitos, determine o vencedor no jogo War. Relembrando as regras:

1. A guerra consiste em várias batalhas de até 3 contra 3. Os exércitos sempre mandam a maior quantidade possível, contudo, o atacante não pode mandar todos os seus soldados, tendo que ficar com ao menos 1 no seu território. Assim, com menos de 4 soldados, os atacantes atacam com menos. Os defensores também mandam o máximo possível para defesa, contudo, eles estão lutando na sua casa e não precisam deixar um soldado ao menos para trás.
2. Para cada soldado na batalha joga-se um dado e ordena-se, assim, por exemplo, os atacantes terão 3 dados e os defensores terão 3 dados também, a não ser que tenham menos que 3 soldados na batalha.
3. Comparando os dados ordenados 1 a 1, o maior da defesa contra o maior do ataque e assim por diante, determina-se um vencedor que é o que possui o maior valor no dado.
4. A defesa ganha todos os empates.
5. Para cada derrota, um time perde um soldado, podendo perder de 0 a 3 de uma vez.
6. A guerra acaba quando ou a defesa for totalmente dizimada, ou reste apenas um atacante.

Simule para grandes exércitos usando um único objeto da classe Random para toda aplicação. Simule 10 mil vezes a luta entre 1000 atacantes e 500 defensores e meça quanto tempo durou para realizar o calculo. Apresente o resultado e compare com o esperado de aproximadamente 50%.

Dica 1: Faça o código não paralelo primeiro.

Dica 2: Este exercício não precisa de Async e Await.

Dica 3: Usar ConcurrentQueue pode ajudar a resolver o problema de forma inteligente.

Dica 4: Na solução do professor, foram usadas menos de 150 linhas, incluindo a implementação sequência, paralela e o teste de velocidade e resultado.