

Program Design :

How It Works:

1. **Client-Server Communication:** The system consists of two main components: the client and the server. The server is responsible for managing the war simulation, the client interacts with the server to initiate the simulation and provide input parameters.
2. **gRPC Communication:** gRPC is used for communication between the client and server. It allows the client to send messages (requests) to the server, and the server responds with messages (responses).
3. **Client Input:** The client prompts the user to input various parameters related to the war scenario, including the number of soldiers (M), the size of the battlefield (N), the total duration of the war (T), and the time interval for missile attacks (t).
4. **Server Initialization:** The server initializes and sets up the gRPC server, which listens for incoming requests from the client.
5. **War Simulation:** The server contains the core logic for the war simulation. It maintains the state of the war, including the positions and status of soldiers, the current commander, and the progression of time.
6. **Missile Attacks:** At regular intervals (specified by t), the server simulates missile attacks on the battlefield. It calculates the impact of the missiles, determines which soldiers are hit, and updates the battlefield accordingly.
7. **Commander Selection:** The server selects a commander at the beginning of the war, and if the commander is killed during the simulation, a new commander is randomly chosen from the surviving soldiers.
8. **Result Evaluation:** After the specified duration of the war (T), the server evaluates the result of the war based on the number of surviving soldiers. If more than half of the soldiers are alive, the client is informed that the war was won; otherwise, it's considered lost.

Client Output: The client receives the war result from the server and displays it to the user.

Design Tradeoffs:

1. **Simulation Complexity:** The code aims to simulate a war scenario with certain simplifications and assumptions, such as the random selection of commanders, simplified missile impact calculations, and a fixed time interval for attacks. These simplifications were likely made to keep the code concise and focused on demonstrating gRPC usage.
2. **Realism vs. Performance:** The simulation prioritizes simplicity and performance over realism. For instance, the missile attacks are randomly generated within a time interval, and the impact calculation is based on simplified rules. In a more realistic simulation,

factors like soldier strategies, terrain, and advanced missile physics would be considered.

3. **Communication Protocol:** gRPC is chosen as the communication protocol between the client and server due to its efficiency and ease of use for defining service interfaces. However, this also ties the client and server implementations to gRPC, which may not be necessary for simpler simulations.
4. **Input and Configuration:** The code relies on user input for configuring the simulation parameters. In a real-world scenario, these parameters might come from external sources or configuration files, making the simulation more flexible.
5. **Server Resource Usage:** The server uses a thread pool executor to handle incoming requests. This design choice limits the number of concurrent requests the server can handle based on the thread pool size. For high scalability, alternative approaches like asynchronous handling or load balancing could be considered.
6. **Scalability:** The current implementation is designed for a single server-client interaction. For more complex simulations with distributed components or multiple clients, the design would need to be extended to accommodate such scenarios.

Overall, the code provides a basic example of using gRPC for a simulated war scenario, but it makes certain tradeoffs to keep the code simple and focused on the core concepts of gRPC and simulation. Depending on the requirements and goals of a real-world application, additional features, scalability improvements, and more realistic simulation logic may be needed.

Steps to Run Our Program :

Your system must have Python to run this program. Now,

1. Install gRPC and clone our GitHub Repository:

Install gRPC on your system using following commands:

```
python -m pip install grpcio
python -m pip install grpcio-tools
```

Now clone our GitHub repository on your systems.

2. Run the Server and Client :

Open a terminal and navigate to the directory containing greeter_server.py.

Run the server using the following command:

```
python3 greeter_server.py
```

Similarly open another terminal and navigate to the directory containing greeter_client.py.

Run the client using the following command:

```
python3 greeter_client.py
```

3. Give input and view results :

When you will run greeter_client.py , you have to follow these steps

Step 1: Enter the number of soldiers (M) : this is the total number of soldiers.

Step 2: Enter the size of matrix (warzone) (N) : this will create a N*N matrix.

Step 3: Enter the duration of war (T) : This will take the total duration of the war.

Step 4 : Enter the time interval of the missile attack (t) : this will take time intervals between each missile attack.

Initially the commander, initial matrix, speed and the positions of the soldiers will be chosen randomly. Now in every t second there will be a missile attack and it will print the matrix, each soldier's position and speed. Now it will print Type of missile fired , target coordinates, red zone coordinates, the following soldier died, soldiers who escaped. At the end of the war, it will print whether the war is won or lost.

4. Cleanup :

After running the program, you can stop both the client and server by closing their respective terminal windows.

Tests, inputs and outputs :

Test 1 :

Here war is began and server will broadcast the information to the soldiers...

Enter the details...

Enter the no. of soldiers (M):10

Enter the size of matrix (warzone) (N):10

Enter the duration of war (T):30

Enter the time interval for missile attack (t):5

Iteration with Commander: Soldier 8

Initial Positions of Soldiers:

```
.....
.....
... C8 .....
.. 4 6 3 .....
.. 1 .. 9 .....
..... 2 .
.....
7 .....
..... 5 . 10 ..
.....
```

Coordinates and Speed:

Soldier 1: (4, 2) (Speed 2)

Soldier 2: (5, 8) (Speed 0)

Soldier 3: (3, 4) (Speed 1)

Soldier 4: (3, 2) (Speed 4)
Soldier 5: (8, 5) (Speed 4)
Soldier 6: (3, 3) (Speed 4)
Soldier 7: (7, 0) (Speed 3)
Soldier 8: (2, 3) (Speed 0) [Commander]
Soldier 9: (4, 5) (Speed 1)
Soldier 10: (8, 7) (Speed 4)
Missile Type 4 targeted (3, 2)

Missile Fired:

Missile Type: 4

Target Coordinate: (3, 2)

RED Zone Coordinates: [(0, -1), (0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (1, -1), (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, -1), (2, 0), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, -1), (3, 0), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (4, -1), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, -1), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (6, -1), (6, 0), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5)]

The following soldiers died:

Soldier 1
Soldier 3
Soldier 8
Soldier 9

Soldiers who Escaped:

Soldier 2
Soldier 4
Soldier 5
Soldier 6
Soldier 7
Soldier 10

Commander was killed! Selecting a new commander.

Next missile-attack in 5 seconds...

Iteration with Commander: Soldier 7

Initial Positions of Soldiers:

.....
.....
.....
.. 4 6
.....
..... 2 .
.....
C7
..... 5 . 10 ..
.....

Coordinates and Speed:

Soldier 2: (5, 8) (Speed 0)
Soldier 4: (3, 2) (Speed 4)

Soldier 5: (8, 5) (Speed 4)
Soldier 6: (3, 3) (Speed 4)
Soldier 7: (7, 0) (Speed 3) [Commander]
Soldier 10: (8, 7) (Speed 4)
Missile Type 2 targeted (5, 8)

Missile Fired:
Missile Type: 2
Target Coordinate: (5, 8)
RED Zone Coordinates: [(4, 7), (4, 8), (4, 9), (5, 7), (5, 8), (5, 9), (6, 7), (6, 8), (6, 9)]

The following soldiers died:
Soldier 2

Soldiers who Escaped:
Soldier 4
Soldier 5
Soldier 6
Soldier 7
Soldier 10

Next missile-attack in 5 seconds...

Iteration with Commander: Soldier 7

Initial Positions of Soldiers:

.....
.....
.....
.. 4 6
.....
.....
.....
C7
..... 5 . 10 ..
.....

Coordinates and Speed:
Soldier 4: (3, 2) (Speed 4)
Soldier 5: (8, 5) (Speed 4)
Soldier 6: (3, 3) (Speed 4)
Soldier 7: (7, 0) (Speed 3) [Commander]
Soldier 10: (8, 7) (Speed 4)
Missile Type 2 targeted (8, 5)

Missile Fired:
Missile Type: 2
Target Coordinate: (8, 5)
RED Zone Coordinates: [(7, 4), (7, 5), (7, 6), (8, 4), (8, 5), (8, 6), (9, 4), (9, 5), (9, 6)]

No soldiers were killed.

Soldiers who Escaped:

Soldier 4
Soldier 5
Soldier 6
Soldier 7
Soldier 10

Next missile-attack in 5 seconds...

Iteration with Commander: Soldier 7

Initial Positions of Soldiers:

.....
.....
.....
.. 4 6
.....
.....
.....
C7
..... 5 . 10 ..
.....

Coordinates and Speed:

Soldier 4: (3, 2) (Speed 4)
Soldier 5: (8, 5) (Speed 4)
Soldier 6: (3, 3) (Speed 4)
Soldier 7: (7, 0) (Speed 3) [Commander]
Soldier 10: (8, 7) (Speed 4)
Missile Type 4 targeted (8, 7)

Missile Fired:

Missile Type: 4

Target Coordinate: (8, 7)

RED Zone Coordinates: [(5, 4), (5, 5), (5, 6), (5, 7), (5, 8), (5, 9), (5, 10), (6, 4), (6, 5), (6, 6), (6, 7), (6, 8), (6, 9), (6, 10), (7, 4), (7, 5), (7, 6), (7, 7), (7, 8), (7, 9), (7, 10), (8, 4), (8, 5), (8, 6), (8, 7), (8, 8), (8, 9), (8, 10), (9, 4), (9, 5), (9, 6), (9, 7), (9, 8), (9, 9), (9, 10), (10, 4), (10, 5), (10, 6), (10, 7), (10, 8), (10, 9), (10, 10), (11, 4), (11, 5), (11, 6), (11, 7), (11, 8), (11, 9), (11, 10)]

No soldiers were killed.

Soldiers who Escaped:

Soldier 4
Soldier 5
Soldier 6
Soldier 7
Soldier 10

Next missile-attack in 5 seconds...

Iteration with Commander: Soldier 7

Initial Positions of Soldiers:

.....
.....
.....
.. 4 6
.....
.....
.....
C7
..... 5 . 10 ..
.....

Coordinates and Speed:

Soldier 4: (3, 2) (Speed 4)
Soldier 5: (8, 5) (Speed 4)
Soldier 6: (3, 3) (Speed 4)
Soldier 7: (7, 0) (Speed 3) [Commander]
Soldier 10: (8, 7) (Speed 4)
Missile Type 3 targeted (8, 7)

Missile Fired:

Missile Type: 3

Target Coordinate: (8, 7)

RED Zone Coordinates: [(6, 5), (6, 6), (6, 7), (6, 8), (6, 9), (7, 5), (7, 6), (7, 7), (7, 8), (7, 9), (8, 5), (8, 6), (8, 7), (8, 8), (8, 9), (9, 5), (9, 6), (9, 7), (9, 8), (9, 9), (10, 5), (10, 6), (10, 7), (10, 8), (10, 9)]

No soldiers were killed.

Soldiers who Escaped:

Soldier 4
Soldier 5
Soldier 6
Soldier 7
Soldier 10

Next missile-attack in 5 seconds...

Iteration with Commander: Soldier 7

Initial Positions of Soldiers:

.....
.....
.....
.. 4 6
.....
.....
.....
C7
..... 5 . 10 ..
.....

Coordinates and Speed:
Soldier 4: (3, 2) (Speed 4)
Soldier 5: (8, 5) (Speed 4)
Soldier 6: (3, 3) (Speed 4)
Soldier 7: (7, 0) (Speed 3) [Commander]
Soldier 10: (8, 7) (Speed 4)
Missile Type 1 targeted (8, 7)

Missile Fired:
Missile Type: 1
Target Coordinate: (8, 7)
RED Zone Coordinates: [(8, 7)]

No soldiers were killed.

Soldiers who Escaped:
Soldier 4
Soldier 5
Soldier 6
Soldier 7
Soldier 10

no of soldires left 5
no of soldires were 10

We lose the war...

Conclusion : We lost the war because the number of the soldiers remaining is not greater than 50%.

Test 2 :

Here war is began and server will broadcast the information to the soldiers...
Enter the details...

Enter the no. of soldiers (M):10
Enter the size of matrix (warzone) (N):10
Enter the duration of war (T):30
Enter the time interval for missile attack (t):6
Iteration with Commander: Soldier 1

Initial Positions of Soldiers:

```
.....  
5.....7 3 10 .  
.....  
.....  
.....4.....  
..2.....C1..  
.....
```


.....8.....
...6.....
....9.....

Coordinates and Speed:

Soldier 1: (5, 7) (Speed 4) [Commander]
Soldier 2: (5, 2) (Speed 0)
Soldier 3: (1, 7) (Speed 1)
Soldier 4: (4, 5) (Speed 0)
Soldier 5: (1, 0) (Speed 4)
Soldier 6: (8, 3) (Speed 2)
Soldier 7: (1, 6) (Speed 4)
Soldier 8: (7, 5) (Speed 0)
Soldier 9: (9, 4) (Speed 1)
Soldier 10: (1, 8) (Speed 4)
Missile Type 2 targeted (1, 7)

Missile Fired:

Missile Type: 2

Target Coordinate: (1, 7)

RED Zone Coordinates: [(0, 6), (0, 7), (0, 8), (1, 6), (1, 7), (1, 8), (2, 6), (2, 7), (2, 8)]

The following soldiers died:

Soldier 3

Soldiers who Escaped:

Soldier 1
Soldier 2
Soldier 4
Soldier 5
Soldier 6
Soldier 7
Soldier 8
Soldier 9
Soldier 10

Next missile-attack in 6 seconds...

Iteration with Commander: Soldier 1

Initial Positions of Soldiers:

.....
5.....7.10.
.....
.....
.....4.....
..2....C1..
.....
.....8.....
...6.....
....9.....

Coordinates and Speed:

Soldier 1: (5, 7) (Speed 4) [Commander]

Soldier 2: (5, 2) (Speed 0)

Soldier 4: (4, 5) (Speed 0)

Soldier 5: (1, 0) (Speed 4)

Soldier 6: (8, 3) (Speed 2)

Soldier 7: (1, 6) (Speed 4)

Soldier 8: (7, 5) (Speed 0)

Soldier 9: (9, 4) (Speed 1)

Soldier 10: (1, 8) (Speed 4)

Missile Type 3 targeted (1, 0)

Missile Fired:

Missile Type: 3

Target Coordinate: (1, 0)

RED Zone Coordinates: [(-1, -2), (-1, -1), (-1, 0), (-1, 1), (-1, 2), (0, -2), (0, -1), (0, 0), (0, 1), (0, 2), (1, -2), (1, -1), (1, 0), (1, 1), (1, 2), (2, -2), (2, -1), (2, 0), (2, 1), (2, 2), (3, -2), (3, -1), (3, 0), (3, 1), (3, 2)]

No soldiers were killed.

Soldiers who Escaped:

Soldier 1

Soldier 2

Soldier 4

Soldier 5

Soldier 6

Soldier 7

Soldier 8

Soldier 9

Soldier 10

Next missile-attack in 6 seconds...

Iteration with Commander: Soldier 1

Initial Positions of Soldiers:

.....
5 7 . 10 .

.....
.....
..... 4
.. 2 C1 ..

.....
..... 8
... 6
.... 9

Coordinates and Speed:

Soldier 1: (5, 7) (Speed 4) [Commander]

Soldier 2: (5, 2) (Speed 0)

Soldier 4: (4, 5) (Speed 0)

Soldier 5: (1, 0) (Speed 4)
Soldier 6: (8, 3) (Speed 2)
Soldier 7: (1, 6) (Speed 4)
Soldier 8: (7, 5) (Speed 0)
Soldier 9: (9, 4) (Speed 1)
Soldier 10: (1, 8) (Speed 4)
Missile Type 1 targeted (5, 7)

Missile Fired:
Missile Type: 1
Target Coordinate: (5, 7)
RED Zone Coordinates: [(5, 7)]

No soldiers were killed.

Soldiers who Escaped:

Soldier 1
Soldier 2
Soldier 4
Soldier 5
Soldier 6
Soldier 7
Soldier 8
Soldier 9
Soldier 10

Next missile-attack in 6 seconds...

Iteration with Commander: Soldier 1

Initial Positions of Soldiers:

.....
5 7 . 10 .
.....
.....
.....4.....
..2.....C1..
.....
.....8.....
...6.....
....9.....

Coordinates and Speed:

Soldier 1: (5, 7) (Speed 4) [Commander]
Soldier 2: (5, 2) (Speed 0)
Soldier 4: (4, 5) (Speed 0)
Soldier 5: (1, 0) (Speed 4)
Soldier 6: (8, 3) (Speed 2)
Soldier 7: (1, 6) (Speed 4)
Soldier 8: (7, 5) (Speed 0)
Soldier 9: (9, 4) (Speed 1)
Soldier 10: (1, 8) (Speed 4)

Missile Type 1 targeted (5, 7)

Missile Fired:

Missile Type: 1

Target Coordinate: (5, 7)

RED Zone Coordinates: [(5, 7)]

No soldiers were killed.

Soldiers who Escaped:

Soldier 1

Soldier 2

Soldier 4

Soldier 5

Soldier 6

Soldier 7

Soldier 8

Soldier 9

Soldier 10

Next missile-attack in 6 seconds...

Iteration with Commander: Soldier 1

Initial Positions of Soldiers:

```
.....
5 ..... 7 . 10 .
.....
.....
..... 4 .....
.. 2 ..... C1 ..
.....
..... 8 .....
... 6 .....
..... 9 .....
```

Coordinates and Speed:

Soldier 1: (5, 7) (Speed 4) [Commander]

Soldier 2: (5, 2) (Speed 0)

Soldier 4: (4, 5) (Speed 0)

Soldier 5: (1, 0) (Speed 4)

Soldier 6: (8, 3) (Speed 2)

Soldier 7: (1, 6) (Speed 4)

Soldier 8: (7, 5) (Speed 0)

Soldier 9: (9, 4) (Speed 1)

Soldier 10: (1, 8) (Speed 4)

Missile Type 3 targeted (5, 2)

Missile Fired:

Missile Type: 3

Target Coordinate: (5, 2)

RED Zone Coordinates: [(3, 0), (3, 1), (3, 2), (3, 3), (3, 4), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (6, 0), (6, 1), (6, 2), (6, 3), (6, 4), (7, 0), (7, 1), (7, 2), (7, 3), (7, 4)]

The following soldiers died:

Soldier 2

Soldiers who Escaped:

Soldier 1

Soldier 4

Soldier 5

Soldier 6

Soldier 7

Soldier 8

Soldier 9

Soldier 10

no of soldires left 8

no of soldires were 10

we won the war...

Conclusion : Here more than 50% soldiers are remaining that is why the war is won.