Advanced Operating System(CS G623)

Team

Manisha K Johnson (2022H1120281P)

Jyotsana Chandrakar (2022H1120286P)

## Program Design

There are 5 files for program execution

1. commander\_server.py  
 Initializes the battlefield and manages its state using a commander .

2. missile.py  
Represents missiles with varying impact areas and how they should act

3. soldier\_client.py  
Simulates the actions of soldiers on the battlefield, including movement and response to missile alerts.

4. soldier.py  
Represents individual soldiers, including their position, speed, and status (alive/dead, commander/regular).

5. variables.py  
Defines global constants, such as grid size, number of soldiers, and missile intervals.

## Description

We initially run the Server file in command line of 1 system, and later the client file in command line of another system then a connection is made to server via rpyc.connect**. After connection is established,** then we start a simulation loop at intervals of t until a simulation time T is reached. Random missile is launched and battlefield server is alerted about the missile. Soldiers who are alive in missiles path are alerted and moved to safer positions. In output, we print out the layout of the grid with impacted areas by missiles marked as R, and soldiers in S. A status report is printed, indicating which soldiers are alive and which are dead. The battlefield is cleared of missile impacts in preparation for the next round. g. If more than half of the soldiers are dead or the commander is dead, accordingly battle status is marked. Remote procedure calls to call subroutine on another machine without knowing the content of the subroutine. to enable synchr communication between microservices. Method of using existing communications features in a transparent way. This pattern improves the system's ability to build distributed systems.

## Design Tradeoffs

1. Instead of random movements, soldiers can be optimized to choose a direction that maximizes their survival chances.
2. For real-time simulations, each soldier's behavior can be handled concurrently, making the simulation more dynamic

## TestCases

Test cases tested to validate the program correctness

1. Check if Soldiers is assigned random position and speed in battlefield layout

Result: Soldier position is dynamic

A screenshot of a computer

Description automatically generated

1. Check the impact zones of missiles are random in Range between 0-4, and visible as R on the battlefield layout

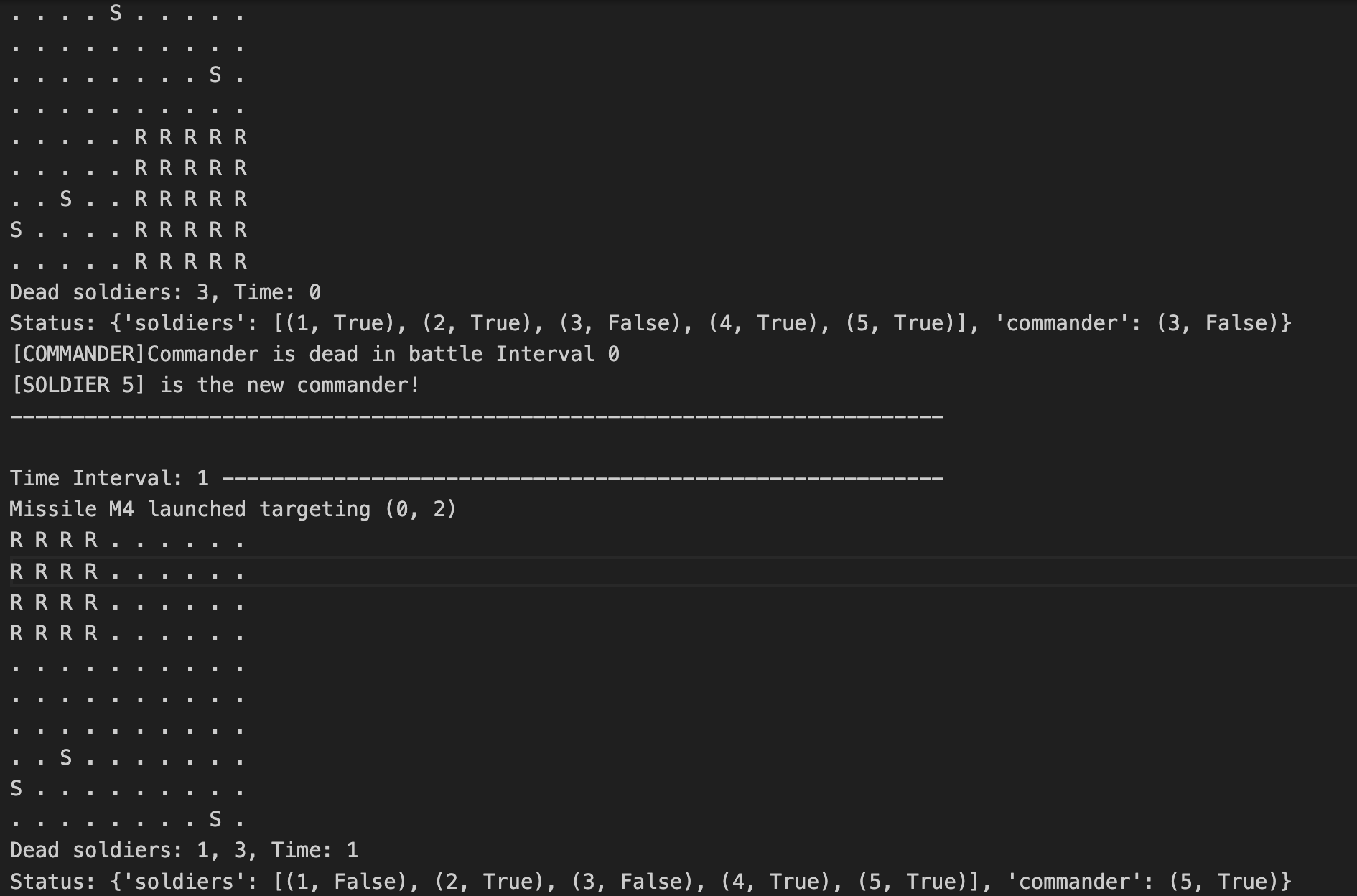
Result:Given 2 iterations below, Missile launched are of different impact sizes

A screenshot of a computer

Description automatically generated

1. Check a new Commander is elected from soldiers if the current commander dies.

Result: Status message displays new elected members



1. Check soldiers attempt to move out of the impact zone based on their speed and in random directions when notification of the missile comes.

Result: Soldier movement is dynamic

A screenshot of a computer

Description automatically generated

1. Check soldiers report their status after each missile impact.

Result: Status about soldiers alive = true, and dead = false

A screenshot of a computer

Description automatically generated

1. Check battle is won when if >50% soldiers are alive after end of deadline

Result: Battle is won

A screen shot of a computer

Description automatically generated

1. Check Soldier in the RED zone dies where missile hits in particular battlefield zone

Result: Soldier dies when hit by missile

A screenshot of a computer

Description automatically generated

1. Check battle is lost when soldiers <50% are alive anytime during missile

Result: Battle is lost

A screenshot of a computer

Description automatically generated

1. Check if the commander is only communicating with soldiers that are alive at the time interval instance

Result: Communication is only with soldiers who are alive and this information is stored in battlefield in a set on the server side

A screen shot of a computer

Description automatically generated

1. Check when missile attacks are infrequent, will battle be won

Result: Battle is won

A screen shot of a computer

Description automatically generated