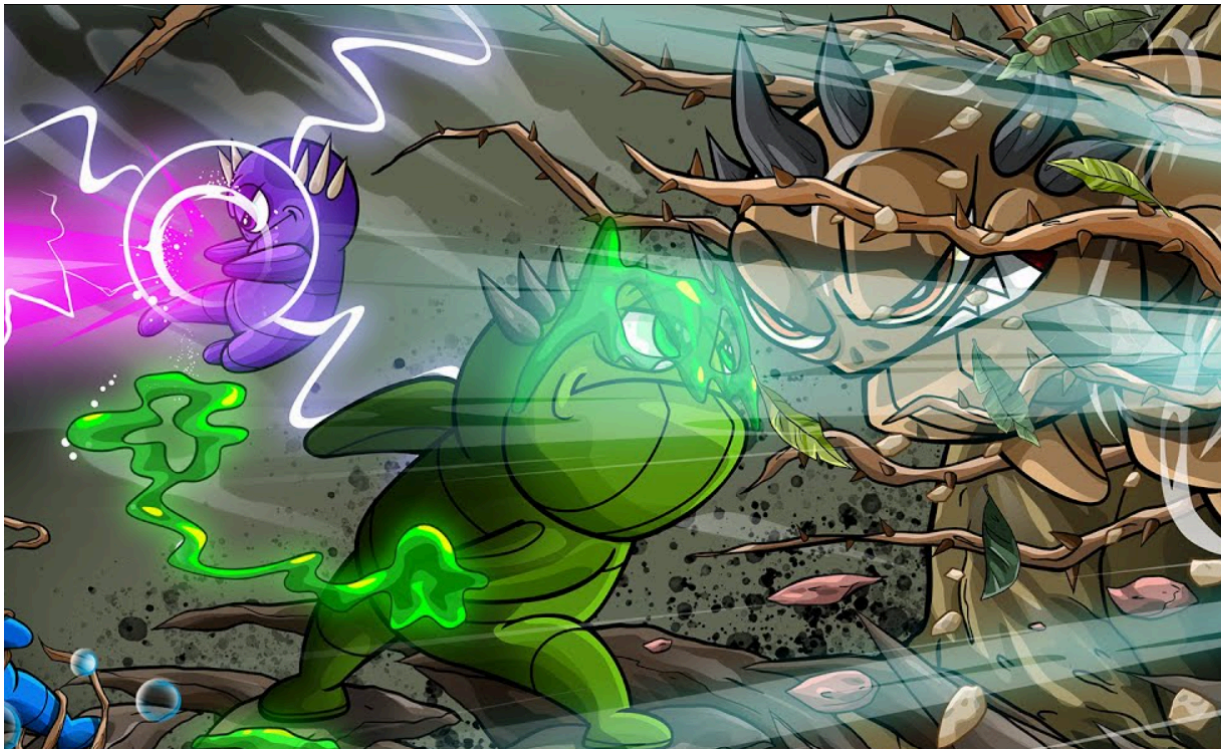
KRSSG SOFTWARE TASK ROUND

Task 4



Titanic Tussle: Clash of the Titans

In a realm where ancient titans reign supreme, two legendary warriors prepare for an epic 3v3 showdown.Commander Alpha, a master of futuristic warfare, leads a team of fire, water, and earth titans. On the other side stands Commander Beta, a brilliant inventor and engineer, commanding a team of rock, thunder, and wind titans. As these mighty commanders step onto the battlefield, their chosen elements converge, sparking an atmosphere charged with anticipation and raw elemental power. Through the mastery of advanced technology, Commander Alpha and Commander Beta communicate their battle plans and commands with the **Main Server**, the nerve center of their operation, using **ROS topics**, orchestrating devastating attacks on each other's titans. With the aid of cutting-edge algorithms and a magical **finite state machine**, they unleash a breathtaking clash where earth shakes, flames rage, and storms rage on with relentless fury.

"In a world of chaos and conquest, rise as the fearless guardian who dares to conquer the unconquerable."

Game Plan: The task is to simulate a 3v3 Monsters battle using ros topic to communicate between players and server.

Player A (Terminal 1)	
Monster name	**Max Hitpoints**
Fire 🔥	300
Water 🌊	400
Earth 🌲	500

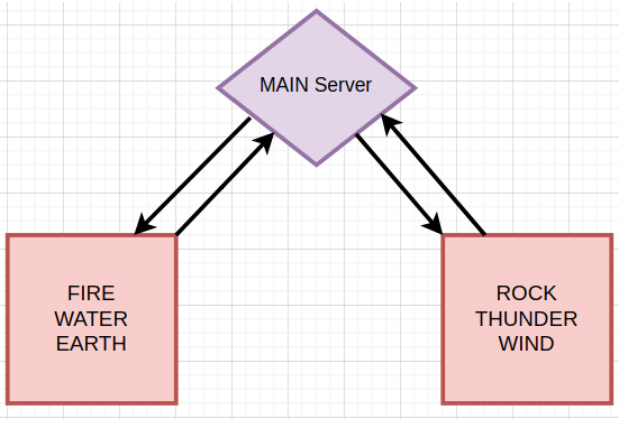
Player B (Terminal 2)	
Monster name	**Max Hitpoints**
Rock 🪨	300
Thunder ⚡	400
Wind 🍃	500

- Each monster has two types of attacks:
    a. **Attack all** : deals 10 percent of the attacking monster's max hitpoints to each of the opponent's monsters that are alive.
    b. **Attack one** : deals 20 percent of damage to the selected Monster (20 percent of max hitpoints)

- In each round, the attack order goes like this:
    - ❖ First, all of Player A's monsters will get to attack in the order (fire, water earth) for which A will communicate with the main server and ask for the hitpoints of all the monsters via ros topic and depending on the hitpoints of the opponent, strategize his attack (manually) and send his move to the main server via another ros topic.
    - ❖ Then, all of Player B's monsters will get to attack in the order (Rock, Thunder, Wind) for which B will communicate with the main server and ask for the hitpoints of all the monsters via ros topic and depending on the hitpoints of the opponent, strategize his attack (manually) and send his move to the main server via another ros topic.
    - ❖ The main server will receive the attacks from Players and will also send the updated monster hit points before each round begins. It will also end the game once one of the Players wins.

    - ❖ The attack of each monster should be provided as input through the terminal For "ATTACK ONE", input 1.
      For "ATTACK ALL", input 2 and the opponent monster name.

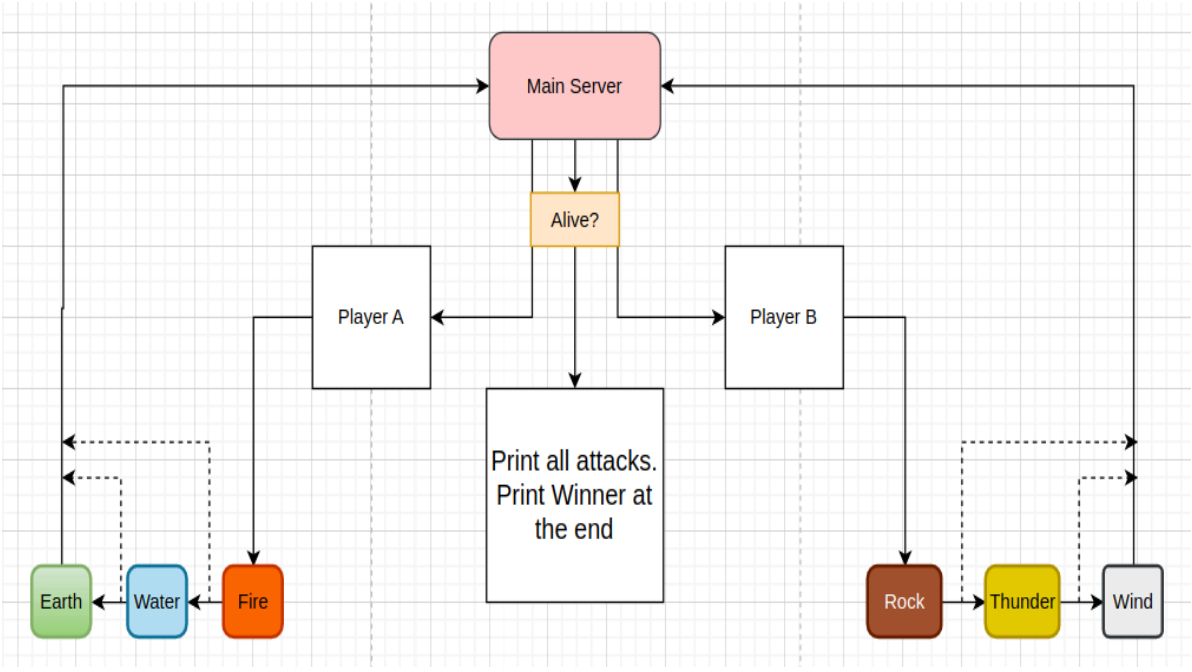
❖ Print the attacks that have taken place after receiving the ROS message from one of the players in the main server terminal. Also print “Fire’s turn”, “Water’s



turn” etc. In the respective terminal when it is that monster’s turn to attack.

- You have to make a communication architecture that looks like the above diagram.

  The lines indicate the ROS communication setup to be made.



- Understand the above flow diagram that shows how the sequence of commands will look like and build the FSM architecture for the same.

Program Files:

You have to make Three scripts in Python/C++.

1. **Game Moderator (Server):** The server will perform the following tasks during gameplay:
    a. Store the hitpoints of all monsters for each player.
    b. Send all the hitpoints **(PRINT)** to the players using ROS topics before the beginning of each round to let the players strategize their moves.
    c. Receive the attack from each player via ROS topics. I.e. Get the inputs after a player has provided info about all his monsters' moves (Alive monsters) and then make the necessary subtractions in the opponent's hitpoints.
    d. After updating the hitpoints of the opponent's monsters, query the opponent to give inputs for his moves (Alive monsters).
    e. Display the attack moves played by the players in the terminal as well.
    f. The chance to attack will alternate between players, starting with player A.
    g. After all the monsters of any of the players are dead, print the winner and end the game.

2. **Player A(B) :** Make two scripts, one for each player.

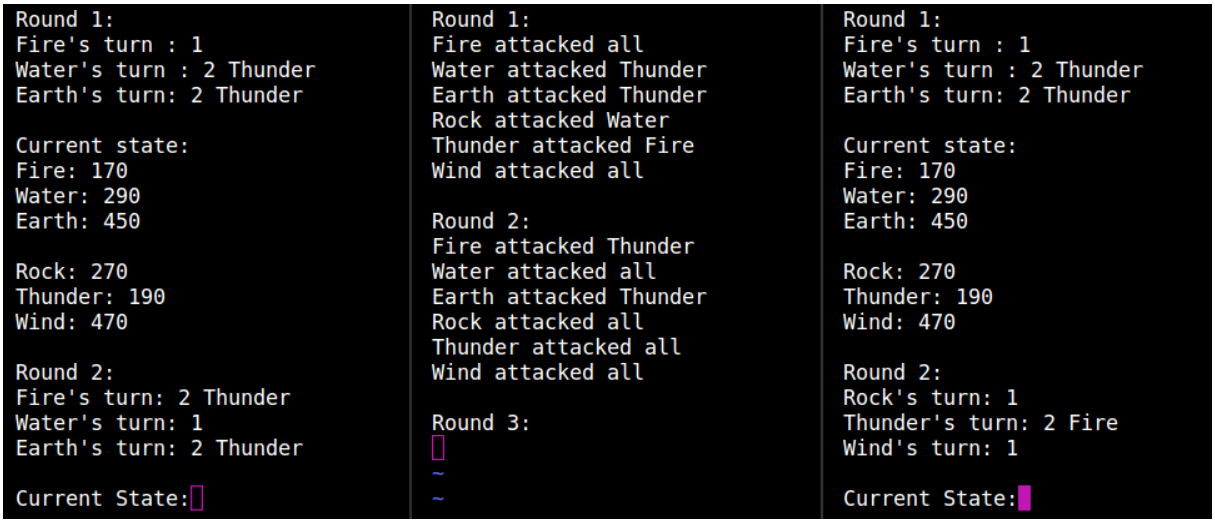
    The Player A(B) will perform the following tasks during gameplay:

    a. Receive the hitpoints of the Player B(A)'s monster from the main server.
    b. Print the hitpoints of all the monsters received from the main server. (**Note** - You should only print hit points that are received from the main server and not calculate inside Player A(B)'s script.)
    c. Take the moves as input from the user for Player A(B)'s monsters.
    d. Send the attack moves taken as input to the main server via ROS topics.
    e. The moves for Player A(B)'s monster (fire, water, and earth for A) will be compiled into a single message and sent via ROS topic to the server.

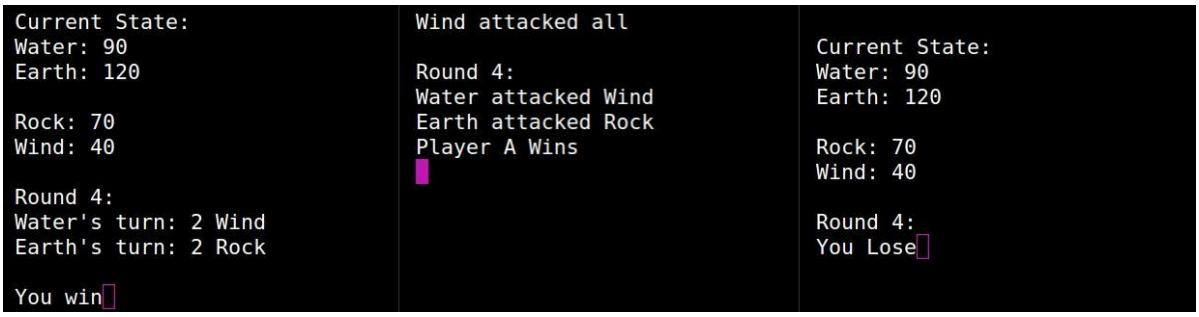
Example: The input should **start** like this :

Player A Terminal	Server terminal	Player B Terminal

Progress like this :



And **end** somewhat like this :



In round 1, The state of the FSM can be seen to change as:

B|Name	Start	Fire attack	Water attack	Earth attack
Rock	300	270	270	270
Thunder	400	370	290	190
Wind	500	470	470	470

A|Name	Start	Rock attack	Thunder attack	Wind attack
Fire	300	300	220	170
Water	400	340	340	290
Earth	500	500	500	450

Language: C++ / Python

Suggested Reading : FSM (Finite State machines), ROS communication