ELECTRICAL TEAM TRAINING

TASK 9

Task9: Group Task

Batman's Ninja Turtles Battle Royale

About

Batman has invited his Justice League friends to play the new ninja turtles battle royale game over the weekend. In this game, each player controls a turtle, competing against each other in an exciting and strategic battle. The game involves movement, attacks, and managing health, all while aiming to be the last turtle standing with the highest health.

Rules:

1. **Movement:**
    - Each turtle can be controlled using the following predefined keys:
    - Example:
        - W to move up

- A to move left
- S to move down
- D to move right

2. **Attack:**
    - Each turtle can attack using the Q key.
        - Damages other turtles within a specified radius (radius to be chosen by you) during the attack duration.
        - Reduces the attacking turtle's number of available attacks by 1.
        - Each turtle has 10 attacks, with each attack inflicting 50 damage.
3. **Health:**
    - Each turtle starts with 100 health points.
    - If a turtle's health drops to 0 or below, it is removed from the game.
4. **Game Over:**
    - The game ends when all turtles have exhausted their attacks.
    - The winner is the turtle with the highest remaining health.

Requirements:

Players will engage in a multiplayer turtle battle game using ROS `turtlesim`, where each turtle can be controlled using predefined keys for movement and attacks. The system includes components to manage the game state, control turtle actions, and handle combat.

1. **Game Engine Node:**
    - Manages the overall game state, including tracking the health and attacks of each turtle.
    - Detects collisions between turtles and handles damage calculations during the attack duration.
    - Logs the scores and determines the winner when the game ends.
    - The game ends when all turtles have exhausted their attacks, and the winner is the turtle with the highest health.
2. **Turtle Controller Node:**

- Controls individual turtle movements and actions based on predefined keys:
  - Movement: `W`, `A`, `S`, `D`
  - Attack: `Q`
- Each turtle can move in any direction using the predefined movement keys.
- Each turtle can attack other turtles using the `Q` key, causing damage within a specified radius during the 1-second attack duration.
- Each turtle has a limited number of attacks (10) and can only attack a certain number of times

***Note**:* You can choose to design the system with these components as separate nodes or integrate them into fewer nodes based on your preference and requirements.

Task 9: Individual Tasks

Task 9.1: let's start docker

About

Batman has set a task for the Justice League to delve into Docker technology. Your mission is to create a Docker image that runs a Python script of your choice. The script should simply print a message of your choosing. Once the Docker image is created, you will push it to Docker Hub so that other League members can review and use it.

Requirements:

1. **Python Script:**

   Create a Python script named `script.py` that prints any message of your choice.

2. **Dockerfile:**

Develop a Dockerfile that defines an image to run you python script

Build Docker Image:

Build a Docker image named `justice-league-python-app` from the Dockerfile.

3. **Test Docker Image:**

   Run a container from the built image to verify it prints the message from your script.

4. **Push Docker Image to Docker Hub**

Task 9.2: let's start ROS

About:

Create a ROS system with two nodes: one that publishes messages and one that subscribes. Package the system into a Docker image and run both nodes within the container.

Requirements:

1. **ROS Nodes:**
   - **Publisher Node:**
     - Publishes messages to the `/chat` topic at 1 Hz.
   - **Subscriber Node:**
     - Subscribes to the `/chat` topic and prints messages.
2. **ROS Package:**
   - Create a ROS package with the publisher and subscriber nodes.
3. **Bonus: Docker Integration:**
   - **Dockerfile:**
     - Install ROS, copy the ROS package, and run the nodes.
   - **Build Image:**
     - Build an image named `ros-pub-sub-app`.
   - **Run Container:**

- Run a container from the image, ensuring both nodes are operational.

Submission

- You will upload the package files to a single GitHub repository for your group and **submit the repository link for Group Task 9**.
- You will submit your **docker image** and the **DockerHub link for individual Task 9.1**.
- You will submit a google drive link for a **video that explains your ROS system for individual Task 9.2**.
- The Task's deadline is 28/8 11:59 PM.
- Q&A Sheet (if you have any question regarding the sessions or the task) : Q&A Sheet
- Submission form: https://forms.gle/i7vEUgE4jFUvG2Qg7
- **Cheating is severely penalized**