PERFORMANCE MEASUREMENT REPORT

TEB1113 – Algorithm and Data Structure

Sept. 2024

Submitted by

24000182	NURUL HANIIZATI BINTI HAZLI
24000574	SARA EUDORA BINTI SAID
22006373	DANIA ADRIANA BINTI MOHD FAIZAL
24000918	ZULAIKHA BINTI MOHD AZHAR
24000227	NUR FATIHAH BINTI MOHD NOOR

Submitted to

Dr M Nordin B Zakaria

CIS Department
Universiti Teknologi PETRONAS, Malaysia

Objective

This report evaluates the performance of the Unity-based Drone Application by measuring the impact of partitioned communication networks on **frame rate (FPS)** and **operation times** for search and self-destruct functions. The goal is to assess how the drone count and linked list traversal impact the overall system's responsiveness.

Introduction

This report presents and analyses a drone swarm simulation developed in Unity, focusing on managing communication and drone behaviour through a linked list structure. The drones are divided into two groups—Red and Blue—based on predefined tags, with their visual appearance reflecting their assigned group. Each drone moves autonomously toward random targets while avoiding overcrowding within a defined radius.

The system supports real-time addition and removal of drones, as well as communication between them through linked nodes. Performance metrics such as partition time and FPS, are tracked and displayed to ensure smooth operation during the simulation.

System Setup

- **Hardware**: CPU = Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz
  RAM = 16GB
  GPU = NVIDIA GeForce RTX 3060 Laptop GPU.

- **Unity Version**: Unity 2022.3.47f1
- **Application Configuration**: Two partitions, each represented as a linked list of drones.
- **Video Link :** https://youtu.be/PhZpGtpoHD4

User Interface(UI)

The system's main UI interface consists of a main menu and scene where users can easily interact with drones, visually identifying them and performing actions through buttons like "Play".

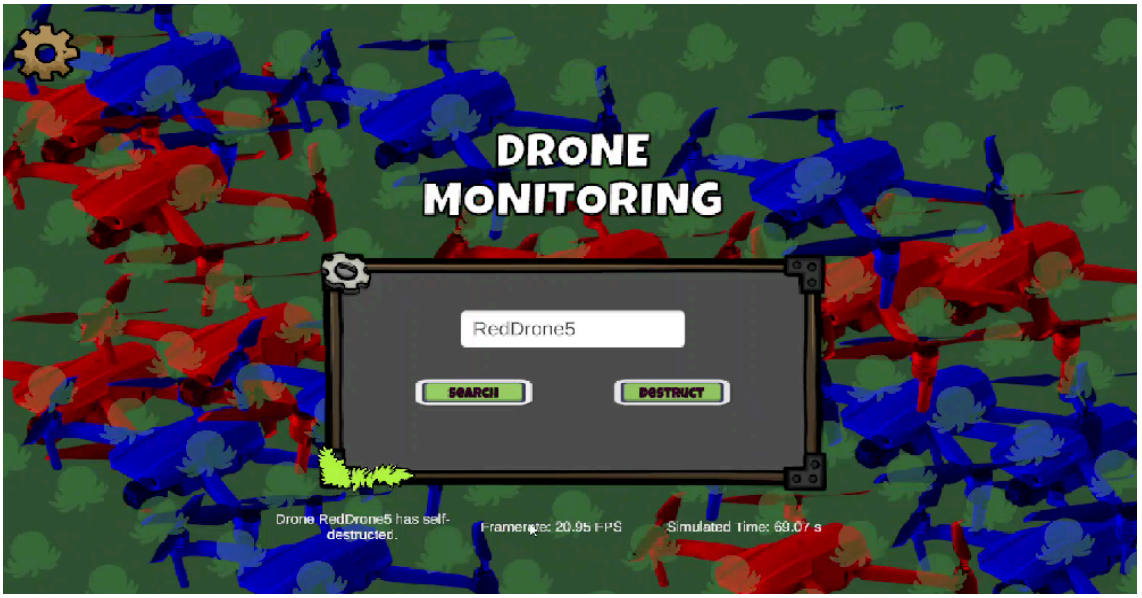
By selecting "Play" from the Main menu, the player is sent to the game level where drone generation begins which produces a collection of drones that are subsequently allocated to the scene. Every drone has unique characteristics, like colour and colours are awarded based on this characteristic. This colour-coding makes it easier for users to easily identify the various drone kinds in the surrounding area.

After clicking the “Play”, it will move to the scene where we can search and destruct the drone.

Within this panel, there is a **text input field** where users can type a drone’s identifier (e.g., "RedDrone5") to interact with a specific drone. When the “SEARCH” button is clicked, at the bottom left of the screen, there are several pieces of technical information:

- **Drone Position**: Shows the coordinates of the currently selected drone (e.g., "RedDrone5") as (4.95, 2.93, 0.0), indicating its 3D location within the simulation.
- **Framerate**: Displays the current frames per second (FPS) rate, indicating the performance of the simulation (e.g., 21.36 FPS).

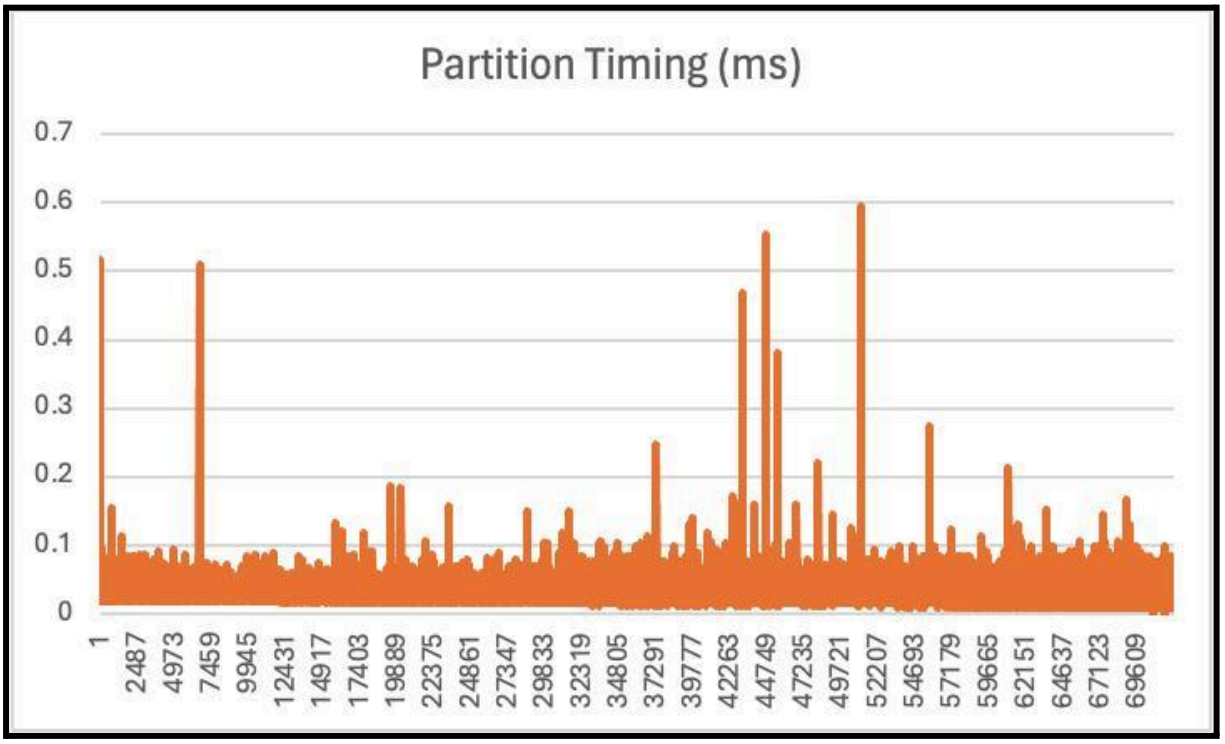
- **Simulated Time**: Shows the total time that has elapsed in the simulation, with a timestamp of 59.67 seconds, which helps in tracking the simulation's progress.



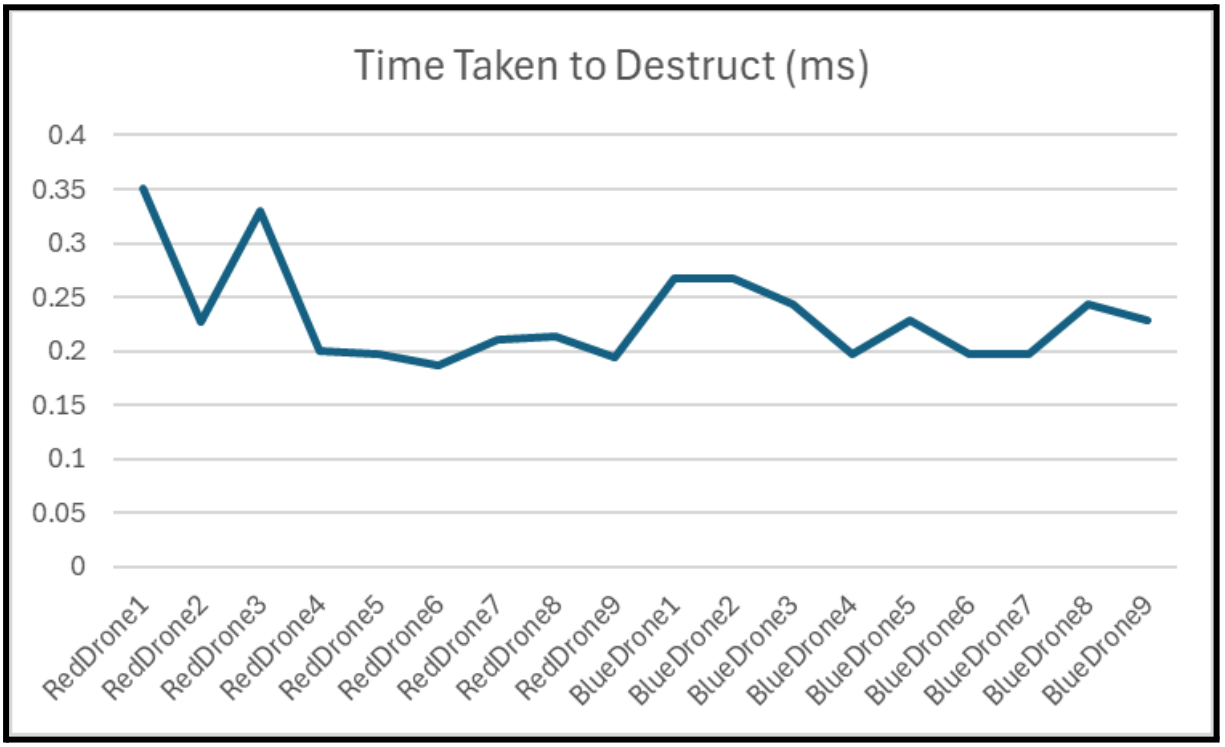
When the "DESTRUCT" button is clicked, the information at the bottom left of the screen changes as the specific drone entered in the text input field has been deleted from the UI.

- **Drone Position**: Shows the selected drone as has been self-destructed from the simulation.
- **Framerate**: Displays the current frames per second (FPS) rate, indicating the performance of the simulation (e.g., 20.95 FPS).
- **Simulated Time**: Shows the total time that has elapsed in the simulation, with a timestamp of 69.05 seconds, which helps in tracking the simulation's progress.

Analysis



Two charted results from a study were used to better comprehend the system's efficiency. The time spent on the partitioning and destruction operations was the main focus of this analysis. Partitioning Time Analysis revealed that while the overall trend indicates an increase in time with greater drone counts, the amount of time needed for partitioning varies as more drones are added. There were spikes seen in the direction of the higher indices, suggesting that the environment gets more complicated and variable.



This graph shows the "Time Taken to Destruct" (in milliseconds) for different drones, labelled as "RedDrone" and "BlueDrone," numbered from 1 to 8 for each type. The y-axis represents the time in milliseconds, while the x-axis lists each drone sequentially. From the data, we can observe: Red Drones generally start with a higher destruction time, with some fluctuations, and gradually stabilise around 0.2 ms. Blue Drones seem to have more consistent destruction times, hovering just below 0.2 ms with minor variations. Overall, the Blue Drones show slightly less variation in destruct times than the Red Drones, potentially indicating more stability or consistency in their destruct mechanisms.

Conclusion

The Unity Drone Application performs efficiently with a smaller number of drones, maintaining stable FPS.

- **FPS Impact:** While search and self-destruct operations have minimal impact on performance with fewer drones, their effect becomes more pronounced as the swarm size grows.
- **Operation Time:** The use of a **linked list structure** introduces **linear time complexity** for operations such as search and communication. As the number of drones increases, the traversal time grows, resulting in longer operation times.

Optimising the system for larger swarms may require more efficient data structures or multi-threading to ensure smooth performance.