

**TEB1113/TFB2023:Algorithm and Data Structure**

**September 2024**

# REPORT

Group:

## 1. See Tho Soon Yinn (24000187)

## 2. Avinash Kumar A/L Jayaseelan (24000113)

## 3. Ashwin A/L Ravichandran (22012188)

## 4. Syukri Fadhli bin Ahmad (24000074)

## 5. Hamzah Muhsin Bin Hafiz Al-Asadi (22001057)

## 6. Aiman Naim bin Shaiful Zahrin (22005653)

Introduction

This report shows and analyses the current ‘snHm’ project, which added a linked list data structure for managing drone communication networks and evaluates the system's performance in real-time by measuring partitioning and deletion times. The drones in the system are assigned to two networks, categorized by their visual colour—red or blue—based on a calculated "coolness" factor. The main interface includes key functionalities such as partitioning drones into networks, real-time deletion of drones from the simulation, and recording performance metrics.

A screenshot of a video game

Description automatically generated

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 "Start" and "Delete." "Options" currently has no function and "Quit" requires the game to be downloaded to be utilised as no application can be closed in the Unity editor.

In the Main menu, pressing "Start" takes the user to the game level where the creation of drones is initiated, generating a set of drone objects that are then assigned to the scene. Each drone has specific attributes, such as color and a "coolness factor," with colors assigned accordingly. This color-coding aids users in quickly distinguishing between different types of drones within the environment.

The "Delete" button offers the functionality to remove drones from the system. When this button is clicked, the system randomly selects a drone from both lists and removes it from both the visual display and the networked environment, ensuring that the drone is no longer part of the active system. The interface then updates dynamically to reflect this change, immediately adjusting the visual representation in the Unity environment in the “Time Taken: ”field.

A screenshot of a video game

Description automatically generated

The above diagram is the background which appears when the start button is clicked. In the system, drones are organized into distinct communication networks based on their colour, utilizing a linked list structure to manage these networks. The drones are classified into two primary networks: the Red Communication Network and the Blue Communication Network. These networks are overseen by a custom class, `DroneCommunication`, which maintains linked lists for both red and blue communication heads. When drones are assigned to a network, they are appended to the respective list through the `AddToCommunication` method defined in `DroneCommunication.cs`.

The process for drone management is threefold: initialization, assigning, and deletion. During initialization, each drone is created as a new node containing relevant drone data. Depending on the drone’s colour, it is then added to either the red or blue linked list within the network, also handled in `DroneCommunication.cs`. Assigning drones to a network involves appending each initialized drone to the tail of its respective linked list, allowing the list to dynamically grow as additional drones join the system. Deletion of a drone requires the system to traverse the appropriate linked list—either red or blue—to locate and remove the designated drone node. The time taken for each deletion action is recorded and saved.

ANALYISIS

A graph showing a blue line

Description automatically generated

To better understand the system’s efficiency, an analysis was conducted using two charted results. This analysis focused on the time taken for both partitioning and deletion tasks. Partitioning Time Analysis showed that as more drones are added, the time required for partitioning fluctuates, though the overall trend shows an increase in time with higher drone counts. Spikes were observed towards the higher indices, indicating variability as the environment becomes more complex.

A green line graph with numbers

Description automatically generated

In contrast, Deletion Time Analysis revealed that the time taken to delete drones decreased across entries, indicating a more efficient deletion process as drones are sequentially removed. This trend suggests improvements in deletion time with repeated operations.

The findings from these analyses are illustrated through two graphs. The Partitioning Time Graph displays time variations during drone assignment, showing an upward trend as the drone count increases. Meanwhile, the Deletion Time Graph depicts the efficiency of the deletion process, with decreasing times as more drones are removed from the network.

**Conclusion**

In conclusion, the drone simulation system partitions and manages drones using a linked list structure for communication networks. Through real-time measurement and analysis, it is evident that the partitioning process exhibits fluctuations as more drones are added, whereas the deletion process becomes increasingly efficient as drones are removed. The integration of performance logging and the ability to visualize these metrics in real time offers valuable insights into system performance, contributing to a better understanding of network management in dynamic environments. Overall, the project demonstrates the effectiveness of using linked lists for drone communication, providing a scalable and manageable solution for large numbers of entities.