TEB1113/TFB2023: ALGORITHM & DATA STRUCTURE

Performance Report on Drone Swarm Simulation

Prepared By:

Num.	Full Name	Student ID
1.	Manas Ismail Abdylas	22008600
2.	Syahir Amri bin Mohd Azha	22007728
3.	Adam Marwan bin Husin Albasri	22009203
4.	Safiqur Rahman bin Rowther Neine	22008929
5.	Danish Harith bin Ahmad Nizam	22009489

TABLE OF CONTENTS

1.0 DEVICE SPECIFICATIONS

Model: Optiplex 3020
RAM: 16GB
Storage: 1TB SSD
Processor: Intel Core i5-4590 CPU @3.30GHz
GPU: Geforce GTX 1060 6GB
Operating System: Windows 10 x64

2.0 APPLICATION DOMAIN

2.1 Introduction

The purpose of this project is to create a fire-extinguishing drone swarm simulation in Unity, designed to mimic real-world scenarios where autonomous drones are deployed to fight fires in large, complex environments. Each drone in the swarm is equipped with a fire extinguisher, and the simulation focuses on monitoring and dynamically adjusting the swarm's behavior based on each drone's fire extinguisher capacity. This allows the simulation to showcase how these drones can effectively distribute resources during emergency situations.

One key aspect of the project is the implementation of an efficient O(N) algorithm to partition the drone swarm into two subgroups based on their fire extinguisher capacities. This partitioning ensures that drones with higher capacities are grouped separately from those with lower capacities, enabling the system to prioritize resources when needed. Drones with higher capacities are assigned blue sprites, while those with lower capacities are assigned red sprites for easy visualization of their roles within the swarm.

The simulation also integrates real-time performance monitoring, capturing key metrics such as partitioning time and frames per second (FPS) to assess the efficiency of the drone swarm in handling firefighting tasks. Additionally, visual enhancements have been made to the drones and the environment to improve the overall realism of the simulation, making it a more accurate representation of a fire emergency response system. The project not only demonstrates the potential of autonomous drone swarms in firefighting but also highlights the importance of optimizing resource allocation and maintaining performance in real-time operations.

2.2 Algorithm Design

The algorithm used in the drone swarm simulation is an efficient partitioning algorithm with an O(N) time complexity, designed to divide the drones based on a key attribute: their fire extinguisher capacity. This attribute plays a crucial role in determining each drone's effectiveness in firefighting tasks. The partitioning begins by selecting the fire extinguisher capacity of the first drone as a pivot value. This pivot serves as the threshold to separate the drones into two distinct groups: those with fire extinguisher capacities greater than or equal to the pivot (high capacity) and those with lower capacities (low capacity).

Once the pivot is selected, the algorithm iterates through the list of all drones, comparing each drone’s fire extinguisher capacity against the pivot. During this iteration, the drones are sorted into one of two groups. Drones that have a capacity greater than or equal to the pivot are placed in the "high capacity" group, while those with lesser capacities are sorted into the "low capacity" group. This approach ensures that the algorithm only needs one complete pass through the list of drones, maintaining an O(N) time complexity, which makes it highly efficient for real-time operations.

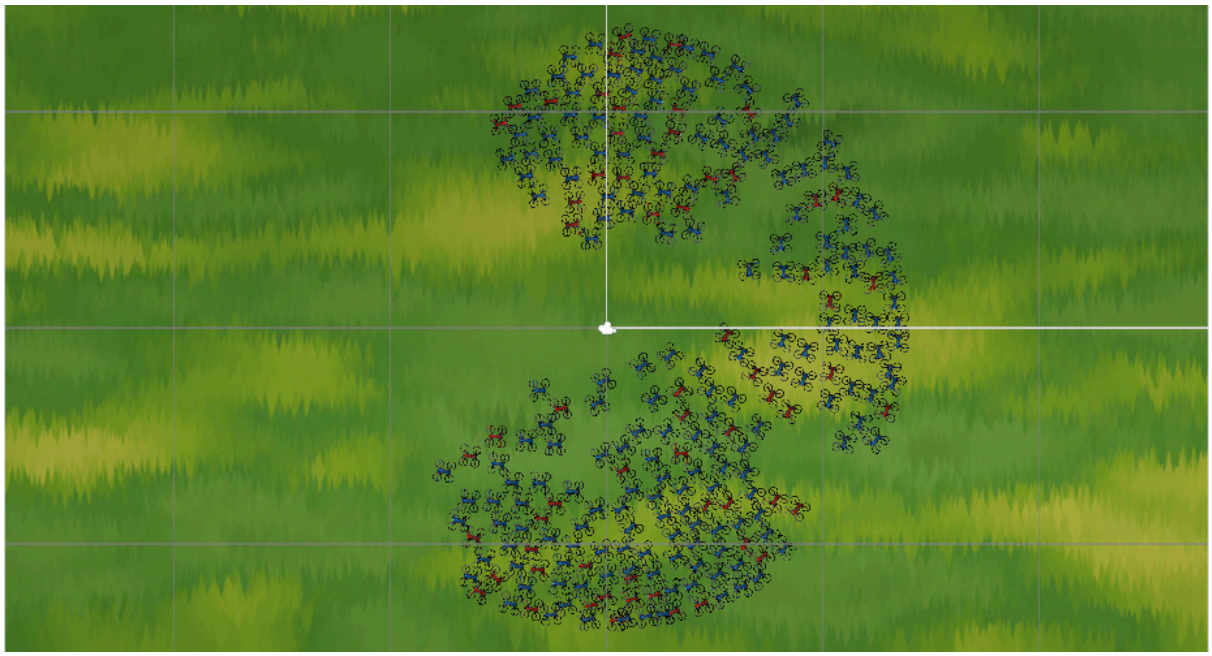
To enhance visual clarity, the algorithm assigns a unique sprite to each drone based on the group it belongs to. Drones in the high capacity group are assigned a blue sprite, while drones in the low capacity group are given a red sprite. This visual partitioning allows the user to quickly distinguish between drones with different firefighting capacities during the simulation, providing clear feedback on the resource distribution within the swarm.

The partitioning process is integrated into the Unity simulation’s Update function, which is executed every frame. This ensures that the partitioning occurs in real-time, updating dynamically as the simulation progresses. Additionally, the time taken for the partitioning process is measured in each frame using the Stopwatch class, allowing the simulation to track performance metrics like partitioning time and frame rates (FPS). By executing this algorithm in real-time, the simulation maintains both high efficiency and visual clarity, making it a practical tool for simulating resource allocation in fire-extinguishing drone swarms.

2.3 Performance Analysis

The machine that was used to conduct this performance analysis was the Optiplex 3020 which consisted of the following parts: 16GB of RAM, 1TB SSD of storage, an Intel Core i5-4590 and a Geforce GTX 1060 6GB. When performing the analysis, there were rarely any signs of stress on the machine with an average frame per second of around 100. On some occasions however, there was evidence of FPS drops as low as around 20 frames per second. The reasons seemed to be unknown as it happens at random, however one of the reasons may come from the machine itself which could hint at a sign of underperforming hardware parts. There were minor frame drops when there was a large quantity of drones shifting from one color to another with it being on average a 20 frame drop. In conclusion, the simulation was running effectively with minimal issues happening on the machine that was being run on.

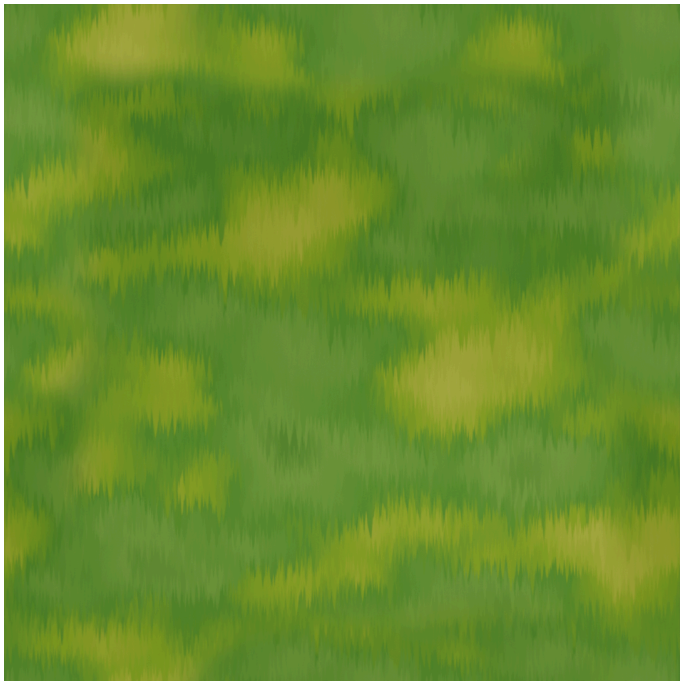
2.4 Screenshot(s)



1	PartitionT	FPS
2	1	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	177.1197
24	0	177.1197
25	0	177.1197
26	0	177.1197
27	0	177.1197
28	0	177.1197
29	0	177.1197
30	0	177.1197
31	0	177.1197
32	0	177.1197
33	0	177.1197
34	0	177.1197
35	0	177.1197
36	0	177.1197
37	0	177.1197

2.5 Drone Model and Background

Drone Models (Isometric and Top View)



Background (Seamless Stylised Grass Texture)