TEB1113/TFB2023: ALGORITHM & DATA STRUCTURE

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

CONTENTS	PAGE
1.0 Introduction	3
2.0 Objective	4
3.0 Simulation Design and Architecture	5
4.0 Algorithms	6
5.0 Unity	7
6.0 Conclusion	8
References	9

1.0 Introduction

Drones are flying vehicles that can be controlled remotely or be flown autonomously by using systems that are embedded into itself which hugely relies on a number of sensors such as air pressure sensors, gyroscopes and accelerometers and also simultaneously using the Global Positioning System or GPS for short to help aid the drone in the action they would be performing at that given times. Drones are used in a numerous amount of activities such as search and rescue, surveillance, traffic monitoring or even agriculture.

A number of given drones being used at the same time can help exponentially perform these tasks faster. The given term for this is called a drone swarm. In a drone swarm, they would typically communicate with each other to synchronize movement and will integrate advanced computer algorithms with the local sensors and communication technologies to synchronize multiple drones to achieve a goal. Drone swarms use various systems of control and communication, predetermined missions with specific flight paths, a centralized control by a ground system or by one single given drone or even a fully decentralized system where drones communicate and collaborate based on given information. These systems use certain given algorithms to help aid the flow of these networks.

2.0 Objective

The primary objective of the drone swarm simulation in Unity is to study the performance of drone swarms in various dynamic environments. The simulation allows for testing of collision avoidance systems, optimized flight paths, and environmental adaptability, which are crucial for real-world applications. Additionally, it serves as a testing ground for evaluating how well different control algorithms and communication protocols perform when scaled to large numbers of drones. By analysing these aspects, the project aims to identify strengths and weaknesses in swarm behavior that can be refined for practical use in fields such as defense, logistics, and environmental monitoring.

The secondary objective is to replicate the behaviour and coordination of numerous autonomous drones in a single area. Their task may range from area surveillance to resource gathering. By simulating real-world scenarios, the project aims to explore the effectiveness of autonomous drones in situations where human involvement is limited or impractical. The simulation seeks to model the communication, movement, and decision-making processes that allow drones to operate as a cohesive unit, demonstrating swarm intelligence principles like flocking, pathfinding, and task allocation.

3.0 Simulation Design and Architecture

The simulation design and architecture of the drone swarm system in Unity are centered around modularity and scalability, which ensures that the simulation can be easily extended and customized for different scenarios. The simulation environment is built using Unity’s 3D capabilities, which allows for the creation of dynamic terrains with obstacles, varying weather conditions, and other environmental factors that influence the drones' behavior. Drones are controlled using physics-based flight systems, leveraging Unity's Rigidbody components to simulate realistic movement and interaction with the environment.

The architecture includes several key components such as drone control systems, swarm behavior algorithms, and communication protocols. Each drone is equipped with flight control mechanisms that allow it to move in three dimensions, avoiding obstacles and maintaining formation with other drones. The control system integrates various sensors, including simulated proximity detectors and raycasting techniques, which help the drones perceive their surroundings and make real-time decisions.

To achieve swarm behavior, algorithms like the Boids model are implemented, allowing drones to follow simple rules such as separation, alignment, and cohesion. These rules govern how each drone maintains distance from its neighbors, aligns with the group’s direction, and stays together as a collective. In addition to these basic rules, a collision avoidance system is implemented to ensure that drones can navigate safely even in complex environments. Communication between drones is modeled either through centralized control, where a single entity dictates swarm behavior, or decentralized control, where each drone makes autonomous decisions based on its local environment and interactions with neighboring drones.

Pathfinding is handled using Unity's NavMesh system, which enables the drones to navigate around obstacles while following predefined waypoints. This system allows for efficient planning of flight paths, ensuring that drones can complete their tasks in a coordinated manner without collisions or deviations from their objectives.

4.0 Algorithms

A drone swarm simulation involves coordinating the behavior of multiple drones to achieve a collective goal. Key factors to consider include formation, avoidance and navigation. Formation is when objects are maintained in a specific way or shape such as a circle, square or the letter ‘W’. Avoidance can be separated into two which are obstacle avoidance, detecting and avoiding objects in the environment and collision avoidance, preventing the drones from colliding with each other. As for navigation, the drones are controlled to a desired destination without damaging itself.

There are two main approaches for implementing the algorithms which are writing from scratch and using libraries or packages. Two examples will be given to each approach accordingly to elaborate more on the approach. Algorithms written from scratch include Boids and A* Search. Algorithms that involve using packages or libraries are DOTS (Data-Oriented Technology Stack) and Unity NavMesh.

Boids is an algorithm that realistically models a flock of birds which can be translated to a swarm of drones. Boid is an abbreviation of “bird-oid object” and it follows a very simple set of rules which include separation, alignment and cohesion. These rules make sure the drones avoid crowding, collision and ensure each drone aligns its velocity with its neighbors. Besides Boids, A* Search is the other instance, an efficient algorithm for pathfinding and graph traversals in real life situations. Many games and web-based maps use this algorithm to find the shortest path very efficiently. It uses a heuristic approach to estimate the shortest path.

DOTS is a powerful framework that can efficiently simulate a large number of drones by managing memory and CPU resources better. At its core, DOTS implements the Entity Component System (ECS) architectural pattern which is built upon three fundamental elements including entities, components and systems. It helps manage thousands of drones efficiently by separating data from behavior logic. Unity Navmesh is a pathfinding AI navigation system that has many built in features, with the main goal of the system being AI pathfinding. It is suitable for scenarios with static environments and simple pathfinding requirements.

5.0 Unity | Why Unity?

Since the plan is to simulate drone swarms, Unity is a great option for it since there are several key reasons that make it ideal for the simulation. These reasons include a flexible development environment, its integration capabilities and large community. The elaboration for each reason will be explained accordingly.

A comfortable development environment for users is important as they will spend countless hours developing on it. Unity designed its interface to provide an incredible experience to the users, prioritizing easy navigation for them, especially to those who are new to game development or simulation. For instance, it provides a clear overview of the scene, objects and properties. Unity also offers a high degree of customization, allowing users to tailor the environment to their preferences. Custom tool sets, shortcuts, and plugins can be created to streamline the development process.

Integration with other tools is another key advantage of Unity. It allows users to leverage the capabilities of other software and programming languages to enhance the drone swarm simulation. For example, Python can be used for data analysis and machine learning, C# for complex logic, and MATLAB for mathematical and scientific computing. By integrating these tools with Unity, more sophisticated and realistic simulations can be created, and perform tasks that would be difficult or time-consuming to do within Unity alone.

Unity has a large and active community of developers who share resources, tutorials, and support for other developers. The Unity Asset Store offers a vast collection of pre-made assets, including models, scripts, and plugins, that can save time and effort in development. Free tutorials are available online and cover almost anything a developer might search for to improve on their work.

6.0 Conclusion

In a nutshell, drone swarms are a game-changer in autonomous systems, many drones can work together to get the job done much better than one-offs. For applications including search and rescue, drone swarms offer advantages thanks to their consolidated movement capability, management communication methods and real-time decision-making capabilities. This extreme scaling and adaptability to changing environments allows them to provide the efficiency seen in the tasks for which AI is beneficial.

Unity offers a flexible development environment, seamless integration with other tools and also strong community support that make it an ideal choice for such simulations. This project aim also is to study some critical aspects which are key for real world applications across industries from defense to agriculture or environmental monitoring. Since drone swarms can offer high performance in terms of task capability, it makes them an increasingly valuable tool in industries where work efficiency is really crucial to be done in a time. To explore all these capabilities, unity can offer an ideal platform for simulating all behavior. It can allow developers to design a 3D terrain or even weather conditions to test how the drones perform. By replicating and testing drone swarms, Unity enables developers to push the boundaries of drone technology to make sure that it can be helped in future usage.

References

https://www.gao.gov/products/gao-23-106930#:~:text=Drone%20swarms%20can%20use%20various,collaborate%20based%20on%20shared%20information.

https://lumasky.show/journal/what-is-a-drone-swarm/

https://www.techtarget.com/iotagenda/definition/drone

https://dronewars.net/aboutdrone/

https://medium.com/@xceltecweb/top-advantages-of-using-unity-for-game-development-bd64b6a3004

https://ventionteams.com/blog/unity-pros-and-cons

https://docs.unity3d.com/Manual/Navmeshes.html

https://github.com/albertpatterson/unity-drone-swarm-simulation

https://vanhunteradams.com/Pico/Animal_Movement/Boids-algorithm.html

https://www.geeksforgeeks.org/a-search-algorithm/

https://medium.com/my-games-company/exploring-unity-dots-and-ecs-is-it-a-game-changer-b21b5d18674b

https://blog.devgenius.io/game-dev-unity-c-what-is-navmesh-68a63d63360b