

**Algorithm and Data Structure  
Guided Project: Report for Drone Swarm Program in Unity**

|  |  |
| --- | --- |
| NAME | ID |
| Ahmad MuqriI Bin Md. Yazid | 22006824 |
| Syazwan Amri Bin Jaafar | 22006712 |
| Mohamad Farid Ikhwan B. Mohd Nazri | 22007823 |
| Damien Leju Idris | 22007158 |

**Objective**

Do research about Drone Swarm and develop a program based on the research.

**Research Findings**

**Swarm drones** are drones especially designed to operate together as robotic swarms. Military swarms are capable of surveillance and attack missions.

**Introduction**

This report outlines the plan on how to design and develop a drone swarm simulation using unity and GitHub. The goal is to design a drone swarm base and platform, and which can be used for different purposes, for instance, simulations, games or research. These heads in brief include the objectives and the scope of the project, the technical details and requirements, resources and the timeline of the project.

**Project Goals**

Simulation Accuracy: Create a realistic simulation of drone behaviour in a swarm formation.

Scalability: Design the system to handle varying numbers of drones efficiently.

User Interactivity: Provide an interface for users to control and observe the drone swarm.

Extendability: Ensure that the system is modular and can be extended with new features or adapted to different scenarios.

**Scope**

Drone Behavior: Implement basic drone movement algorithms, including flocking behavior, collision avoidance, and formation maintenance.

Simulation Environment: Develop a 3D environment where the drones can interact. This will include obstacles, boundaries, and possibly dynamic elements.

User Interface: Create controls for users to manage and monitor the drone swarm.

Data Collection: Enable logging of drone behavior and performance metrics for analysis.

**Technical Specifications**

Unity:

Version: Unity 2023.1 or later

Features: Utilize Unity’s physics engine for realistic movement and collision handling, and Unity’s UI tools for creating user controls.

GitHub:

Repository: Host the project repository on GitHub to manage version control, collaborate with team members, and track issues.

Branching Strategy: Implement a branching strategy such as Git Flow to manage development, testing, and production phases.

Programming Languages:

C#: Primary language for scripting in Unity.

Shader Language (HLSL/GLSL): Optional, for advanced visual effects.

Dependencies:

Unity Asset Store: Use assets such as models and textures if needed.

External Libraries: Evaluate and include any necessary third-party libraries for enhanced functionality.

**Resource Requirements**

Development Team:

Unity Developer(s): To handle coding, design, and implementation.

3D Artist: For creating or sourcing drone and environment models.

UI/UX Designer: To design user interfaces and interactions.

Project Manager: To oversee development, coordinate tasks, and ensure milestones are met.

Software Tools:

Unity Editor: For development and testing.

Visual Studio: For C# programming.

GitHub Desktop or Command Line: For version control and collaboration.

Hardware:

Development PCs: Capable of running Unity and handling simulations.

Testing Devices: Optional, for testing performance on different hardware setups.

**Milestones and Timeline**

1. Initial Planning (Week 1-2):

- Define detailed project requirements.

- Set up the GitHub repository.

- Create initial design documents and project plan.

2. Prototype Development (Week 3-5):

- Implement basic drone behavior and swarm algorithms.

- Develop a simple test environment.

3. Environment and UI Development (Week 6-8):

- Create and integrate the 3D environment.

- Develop user interface components.

4. Testing and Refinement (Week 9-11):

- Perform unit testing and integration testing.

- Optimize performance and address any bugs.

5. Finalization (Week 12):

- Prepare documentation and user guides.

- Conduct final reviews and make necessary adjustments.

6. Deployment (Week 13):

- Finalize the GitHub repository with all code, documentation, and assets.

- Release the project or prepare for the next phase of development.

**Risk Management**

Technical Risks: Address potential challenges related to Unity’s physics and rendering performance by using efficient algorithms and optimizing assets.

Team Coordination: Mitigate risks of miscommunication by holding regular meetings and using project management tools.

Timeline Delays: Plan for buffer periods and prioritize tasks to manage time effectively.

**Conclusion**

This report lays the groundwork that will be used to create a drone swarm simulation in Unity. Accordingly, by achieving the outlined goals, developing the technical specifications and providing the desired timeline as presented in this report, the project seeks to enhance development and make an effective, functional, and scalable drone swarm system. That is why frequent updates and compliance with the development plan will become critical factors influencing the results of the work. It can also be used in as a starting point in the actual development process so that all the members of the development team will be aware of the objectives and expectation of the project.