**Project Proposal & Team Information**

Team 6

**Introduction**

Distributed Multi-Agent System for Autonomous Drones/Robots A system where autonomous drones/robots communicate peer-to-peer to accomplish complex missions through emergent collective intelligence, inspired by bee colonies. Agents make group decisions, share information, and adapt to changing conditions while executing tasks such as coordinated search patterns, perimeter security, or formation flying.

**Objectives**

1. **Collective Decision Making**
   * Peer-to-peer messaging with fault tolerance
   * Range-limited communication simulating real-world constraints.
   * Message prioritization for critical vs. routine information
2. **Distributed Communication System**
   * Consensus algorithms for group divisions (voting systems)
   * Distributed task assignment without central coordination
   * Priority conflict resolution when agents have conflicting objectives
3. **Emergent Behavior Patterns**
   * Flocking algorithms
   * Formation flying capabilities
4. **Additional Features/Ideas for Implementation**
   * Mission Management Interface
   * Advanced Intelligence Features

**Actors/Users**

* **Users:** Government Contractors
* **Team Members:** Sanidhaya Sharma, Anthony Kwasi, John Ademola, Lauren Fries

**Functional Requirements**

* **Agent Management** - How individual agents are created, managed, and controlled
* **Communication** - How agents talk to each other
* **Swarm Behavior** - Flocking, formations, collective decision-making
* **Mission Management** - Task assignment and execution
* **User Interface** - Ground control station and visualization
* **Fault Tolerance** - Handling failures and recovery
* **Hardware Integration** - **Optional** RC car support

**Non-Functional Requirements**

* **Performance** - Speed, scalability, resource usage
* **Reliability** - Uptime, fault tolerance, data consistency
* **Usability** - Interface design, user experience
* **Security** - Communication protection, access control
* **Compatibility** - Platform support, integration capabilities
* **Maintainability** - Code quality, modularity
* **Documentation—Technical** docs, user manuals

**Project Constraints**

* **Time Constraints—**The semester is a short amount of time to complete a project of this complexity, so the scope will have to be limited in a way that allows for completion of the project on time.
* **Resource Constraints—**We don’t have the money or time to implement real-life drones, so we will need to simulate an environment for the system to operate. There are also laws that affect the operation of drones within city limits that would cause an issue as well.

**Languages/Tools Used**

* JAVA and JAVAFX

**Team Information & Skill Sets**

* Sanidhaya Sharma (Team Leader)—JAVA and Python
* Anthony Kwasi—Python and a bit of Java
* John Ademola—Python, C#, Java
* Lauren Fries—Java, Python, C#, .NET

**Team Policies**

* **Meetings:** Saturday at 5 PM online.
* **Voting Scheme:** Voting will be done on a majority rules basis, except in the event of a tie, where the outcome will be chosen by the team leader.
* **Absent Policy:** If any team member(s) is absent, they must contact the team and let them know through the group chat. The absent member must devise a plan to make up whatever they missed, along with taking the responsibility to ask the other team members to fill them in on what they missed so they can catch up. If a team member is unreachable/unresponsive for 3 days, then Dr. V will be notified, and further action will be taken as needed.
* **Resolving Conflicts:** Team members will communicate with one another openly. If there are issues that cannot be resolved through inter-team communication, then Dr. V will be contacted to help resolve the disagreement(s).
* **Task Assignments:** Tasks will be assigned in a way that makes each person contribute as evenly as possible or to work with their current knowledge, skill sets, or interests. Exact assignments are unknown at this stage.