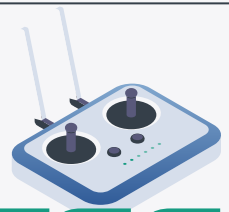
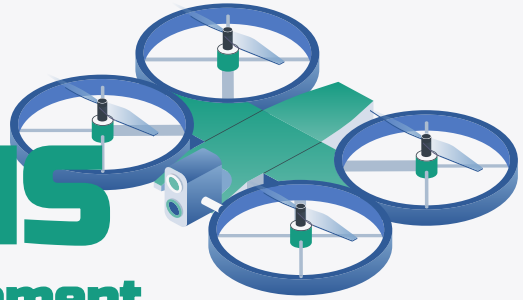




SKY MISSIONS

Machine Learning & Game Development Project

Made By:

Hadi Rida

AHMAD AYOUB

Supervised By:

Dr. Ali Ballout

Fall 2025-2026

Overview

- Train a drone to autonomously navigate a 3D Unity environment
- Use Reinforcement Learning (PPO, ML-Agents)
- Drone can reach targets, avoid obstacles, and return home
- Integrated into a playable game for observation or manual control

Problem Formulation & Environment

Agent: Drone with Rigidbody physics

Environment

3D Unity scene with:

- Targets (static/moving/randomized)
- Obstacles
- Home base (randomized)

Observations

- Drone position, velocity, and altitude
- Target and home base information
- Relative distance and direction to target
- Mission state (deliver / return)
- Obstacle direction, distance, and danger level

Actions

- continuous 3D movement (X/Y/Z forces)

Rewards

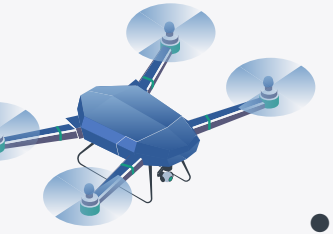
- positive for reaching target considering speed, or getting closer

- penalty for collisions, small timestep penalty

RL Algorithm & Training

Algorithm

Proximal Policy Optimization (PPO) via Unity ML-Agents



Training Settings

- Batch size: 2048
- Learning rate: 0.00025
- Gamma: 0.99
- 3 hidden layers × 256 units each

Training Process

Millions of steps until cumulative reward stabilizes

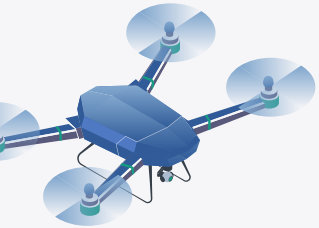
Obstacle Avoidance

Sphere-based detection for all-around awareness

Randomization

Target and home base positions randomized each episode to improve generalization

Obstacle Avoidance Approaches



Sphere-based sensing (Implemented)

Detects obstacles using a spherical overlap region
Efficient, simple, no blind spots

Ray-based sensing (Alternative)

Simulates directional sensors (ultrasonic system)
More realistic sensing, helps path planning
May have blind spots depending on ray configuration

Hybrid approach (Recommended)

Combines sphere + ray sensing
Sphere ensures full coverage, rays improve path planning
Results in safer, smoother navigation and more stable learning

Training Challenges

Observed Issue: Training instability after introducing obstacle avoidance

Catastrophic Forgetting:

- The drone **forgot the primary objective (reaching the target)**
- It learned to **avoid obstacles and "dance" near them**
- The agent received rewards **without making progress toward the target**

Cause:

- Conflicting reward signals between obstacle avoidance and target-reaching

Training Signal (Bad Sign):

- Mean reward showed high variance and frequent collapse
- Indicates unstable learning and conflicting objectives
- **Training logs with negative and highly variable rewards** are included in the report

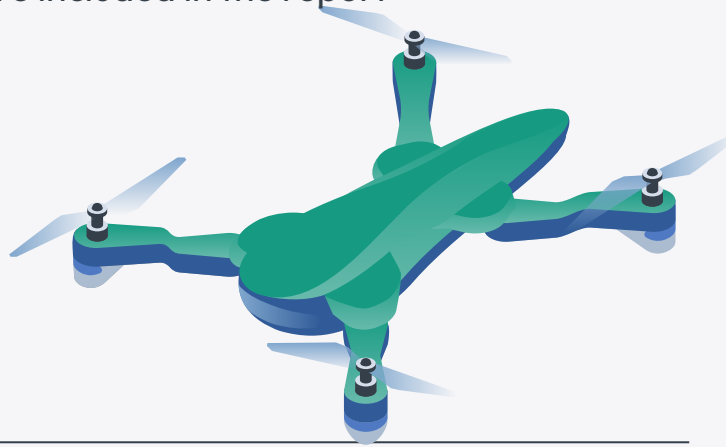
Solutions

Solution – Reward Shaping:

- Temporarily disable or reduce target-distance rewards when obstacles are detected
- Prioritize obstacle avoidance until the area is cleared
- Restore target-reaching rewards afterward

Training Signal (Good Sign):

- Mean reward becomes stable with lower variance
- Indicates consistent policy improvement
- Stabilized training logs with positive mean rewards are included in the report



Results & Conclusion

Results:

- Drone successfully learned to navigate toward targets
- Effective obstacle avoidance with stable behavior
- Able to return home when required
- Training stabilized after reward shaping

Conclusion:

- Reinforcement Learning is effective for autonomous drone navigation
- Proper reward design is critical for stable learning
- Able to return home when required
- Training stabilized after reward shaping

Drone Simulation Overview

This project presents a Unity-based drone simulation designed to model autonomous behavior in a controlled virtual environment.

The simulation is organized into **four mission-based scenarios**, each increasing in complexity and designed to evaluate navigation, decision-making, and mission execution.

User input defines mission parameters, while the drone operates autonomously using Reinforcement Learning techniques.

Missions :

01

Building Strike - Manual Control

02

Building Strike - Ai Assist

04

Vehicle Hunter - Ai Assist

03

Control Vehicle - Avoid Drone

Mission 01 – Building Strike (Manual Control)

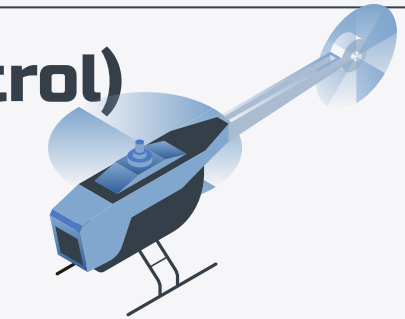
Purpose: Compare AI-assisted behavior with direct human control.

Mission Flow

- Drone starts from a predefined position
- User manually controls movement
- Drone enters selected building
- Payload is deployed

User Interaction

- WASD controls for movement
- User manually navigates the drone
- No AI-assisted path selection



Mission 02 – Building Strike (AI Assist)

Purpose: Simulate an autonomous drone strike with AI-assisted navigation.

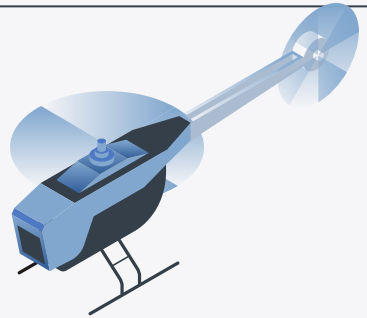
Mission Flow

- Drone starts camouflaged between trees
- User selects target building
- Drone navigates autonomously
- Payload is deployed inside the building
- Drone returns to its initial position

User interaction

- Press **1** → Target House 1
- Press **2** → Target House 2
- Press **H** → Return to start position

Mission 03 – Vehicle Hunter (Ai Assist)



Purpose: Simulate AI-assisted target tracking and engagement of a moving vehicle.

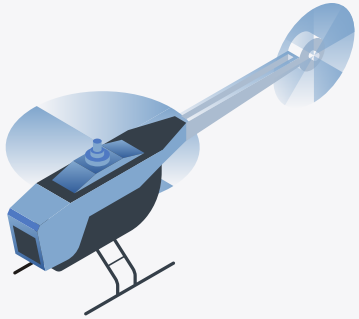
Mission Flow

- Drone starts in standby mode
- User initiates target pursuit
- Drone autonomously tracks the vehicle
- Engagement sequence is triggered
- Mission concludes after execution

User Interaction

- Press **C** → Command drone to chase the vehicle
- Press **A** → Trigger attack sequence

Mission 04 – Control Vehicle (Avoid Drone)



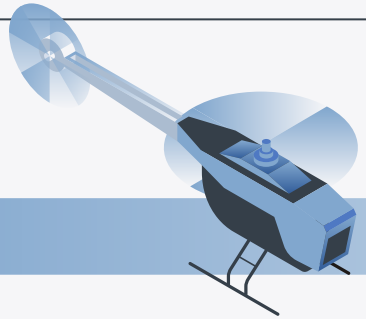
Purpose: Simulate an evasion scenario involving a moving vehicle.

Mission Flow

- User controls a ground vehicle
- An autonomous drone tracks the vehicle
- Vehicle must avoid drone detection

User Interaction

- User controls vehicle movement
- WASD controls for movement
- Drone behavior is AI-driven



LET'S FLY