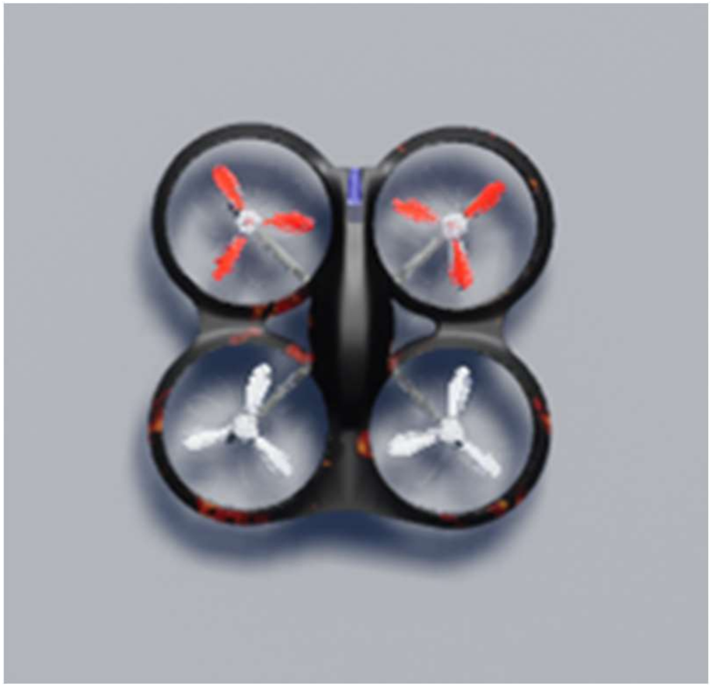
# Modeling Unmanned Aerial Swarms Using Unreal Game Engine and AirSim Simulator

Presented By: Naimah-Joy Chapman, Elijah Keck, Dillion Mead, and John Mueller

# Outline

I. Introduction
II. Objectives
III. FAA Right-of-Way Rules
IV. Design Considerations
V. System Design
VI. Demonstration Videos
VII. Challenges and Lessons Learned



# Introduction

Unmanned Aerial Vehicles in Practice:

- Controlled within Visual Line of Sight (VLOS) by human pilot
- Operating environment is dangerous
- Flight restricted by distance and time

Marketability of Autonomy:

- Does not need human pilot
- Beyond VLOS
- Can fly within any environment with reduced risk
- Flights can be less restricted

# Introduction

Subset of Autonomy - Collision Avoidance

- Mid-air collisions are undesirable
- Autonomous systems don’t have same faculty as humans
- Truly autonomous flight must have guaranteed operation

Why a simulation?

- Project budget of $0!
- Easy to execute multiple tests
- Nearly no risk involved

# Objectives

## Implementation of Autonomous Collision Avoidance System

- Object Detection
  - Lidar sensor primary
  - Camera secondary
  - Transform data for evaluation
- Collision Detection
  - Ascertain distance of obstacles
  - Evaluate obstacle distance
  - React within avoidance radius
  - Initiate collision avoidance
  - Restart collision detection
- Avoidance Behavior
  - FAA Right-of-Way Rules
  - Make right turn maneuver
- Mission Pathing
  - Save list of waypoints
  - Identify next unvisited waypoint
  - Orient UAV to Path heading
  - Call move commands
  - Mark waypoints as visited

# FAA Right-of-Way Rules

- ➢ “When a rule of  this section gives another aircraft the right-of-way, the pilot shall give way to that aircraft and may not pass over, under, or ahead of it unless well clear”
- ➢ Small UAV always yield to other aircraft types

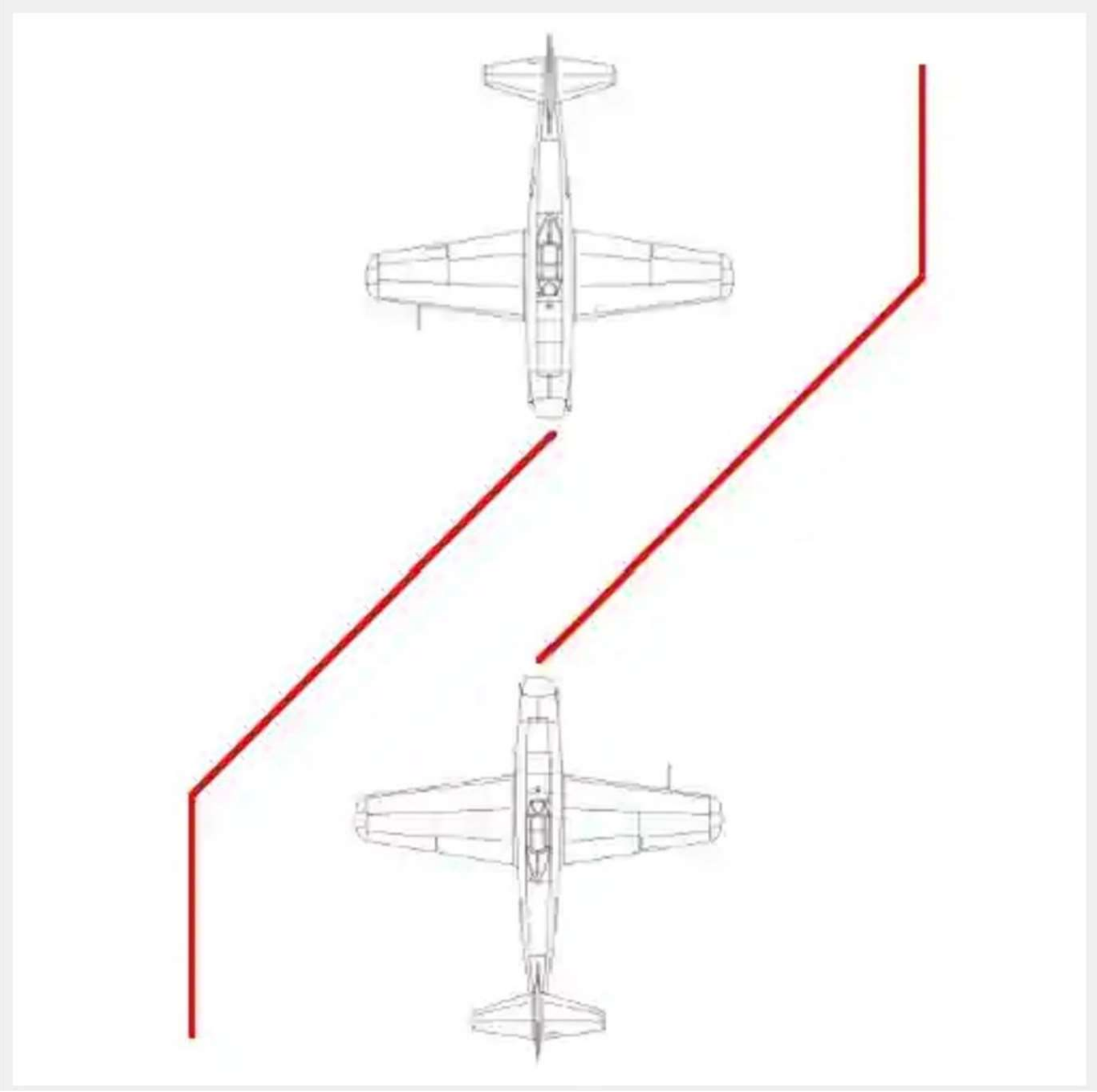
- Head-On Approach
- Converging
- Overtaking
- Formations of Aircraft

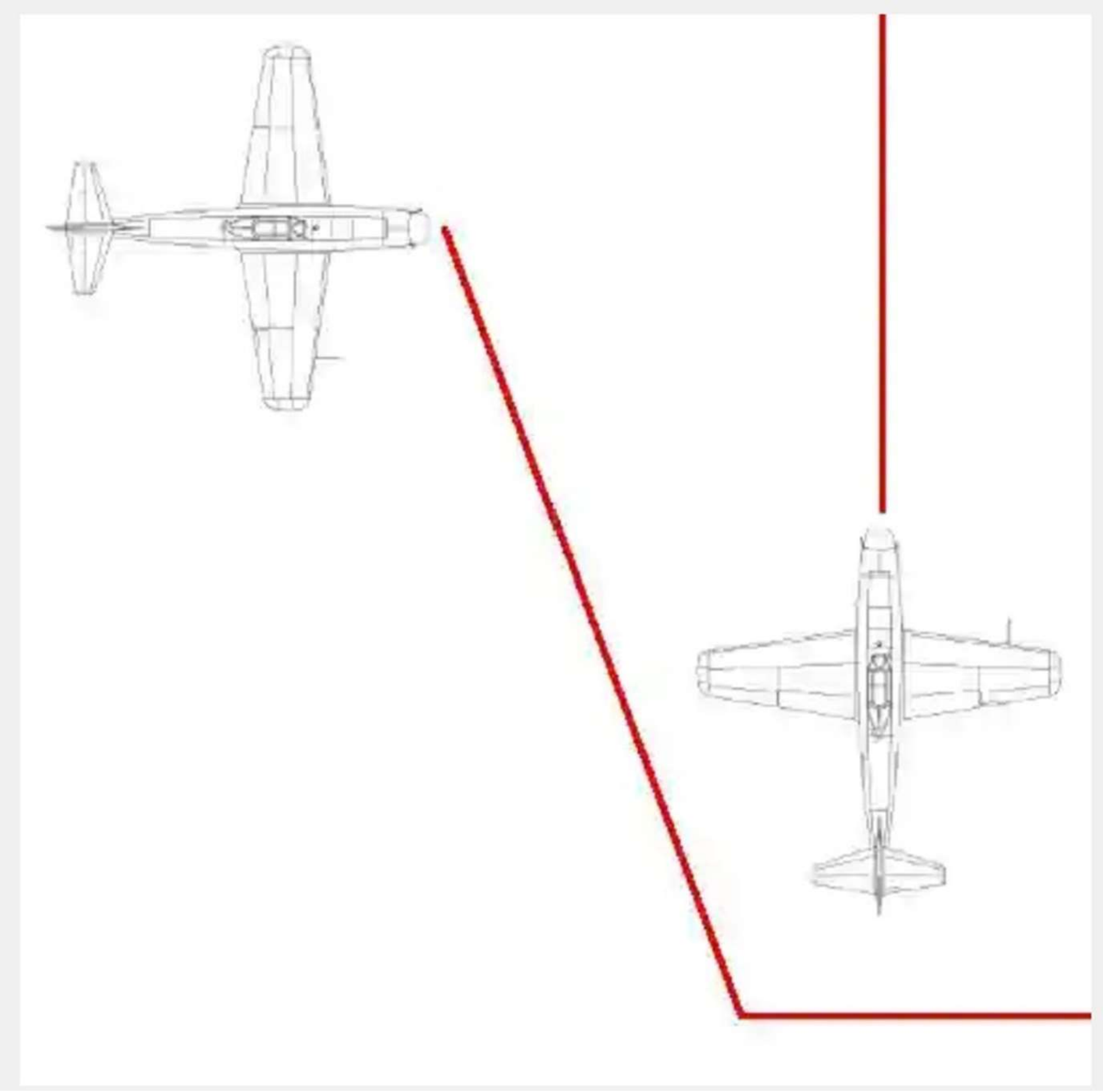
“Rules and Regulations/ Right-Of-Way.” CFI Notebook. https://www.cfinotebook.net/notebook/rules-and-regulations/right-of-way (accessed Apr 06, 2022).

# Head-On Approach

Neither aircraft have right of way

- Both must avoid to right





# Converging

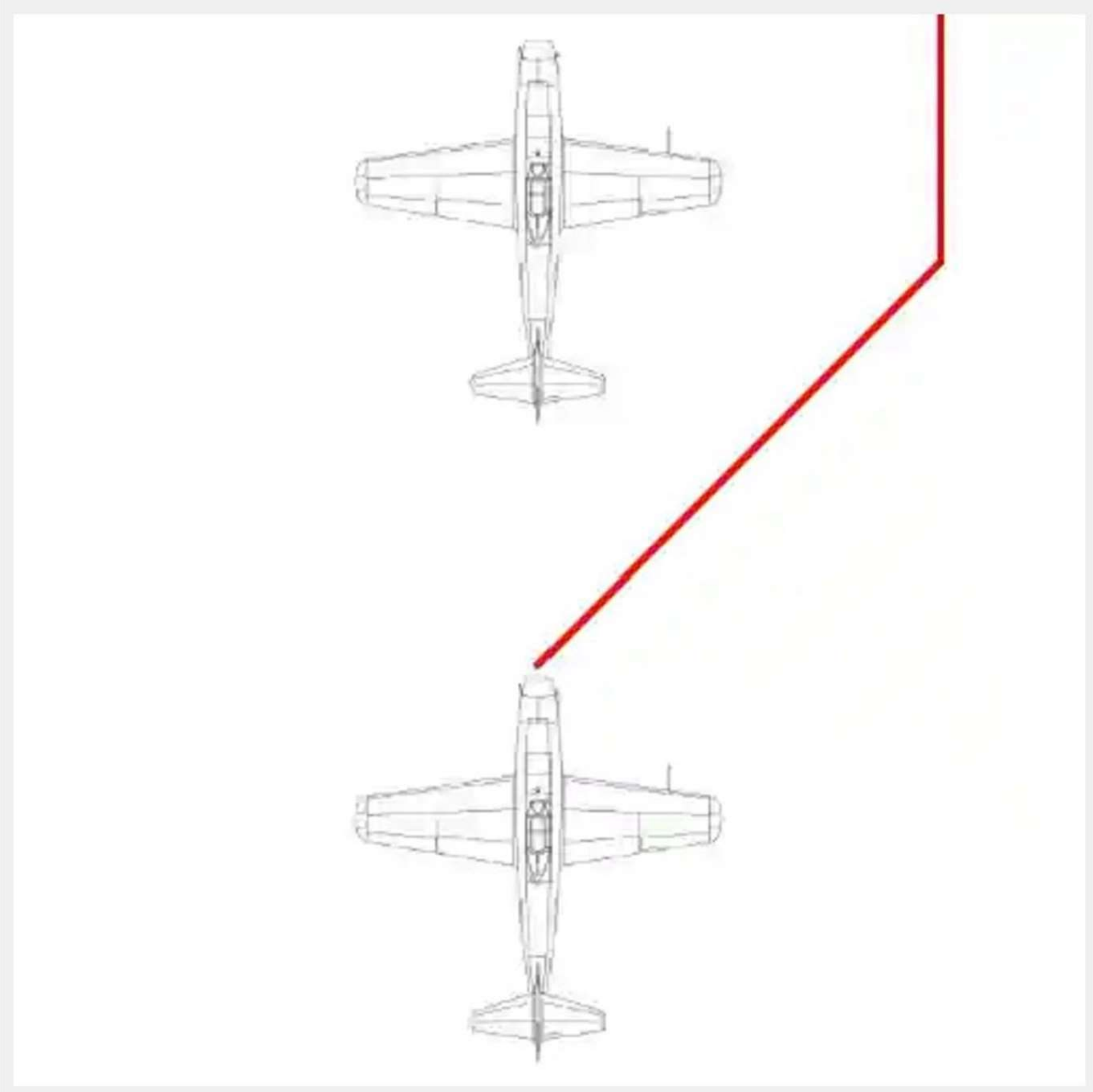
Aircraft to “Right” has right of way

- “Left” aircraft must avoid to right and pass behind other aircraft.

# Overtaking

Aircraft in front has right of way

- Rear aircraft must pass on right



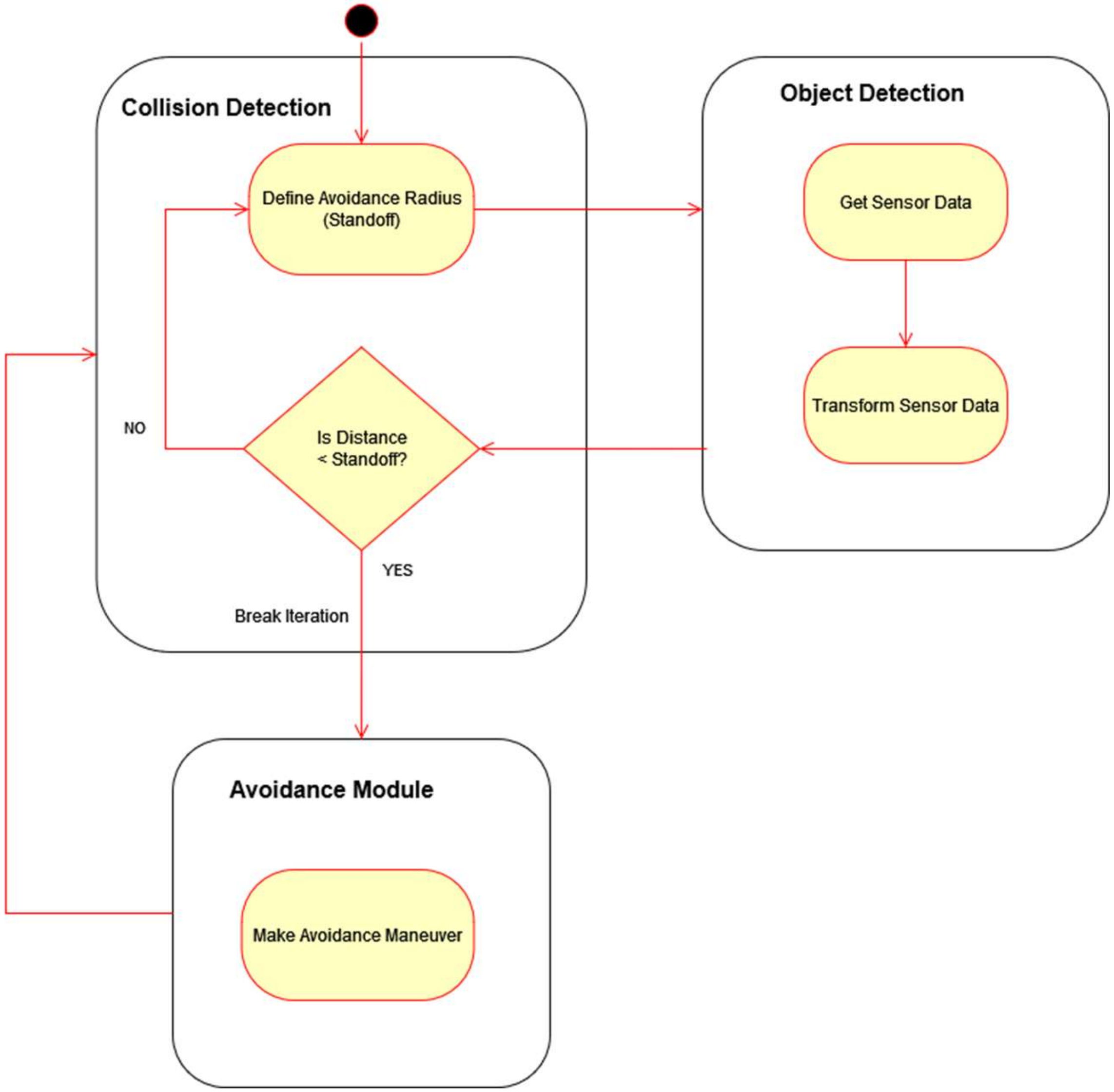
# Design Considerations

- Assumptions
  - Sensors for detection are oriented in the direction of movement
  - There is no interference with the sensors
  - UAV will only move in a forward direction
  - There are no adversarial actors
  - Any non-autonomous UAV are controlled with known parameters
  - Only small UAV aircraft in environment
  - No landing or water scenarios

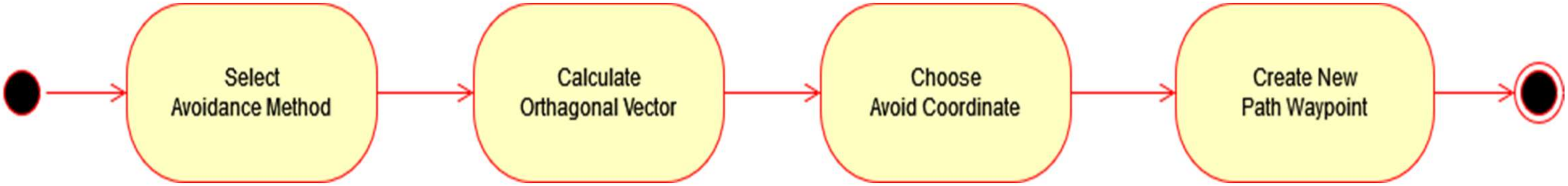
- Dependencies
  - Microsoft AirSim Simulator
  - Unreal Engine 4
  - Visual Studio 2019
  - Utilizes Python Environment
- Design Constraints
  - Limited to Unreal Engine 4 & AirSim environments

# System Design

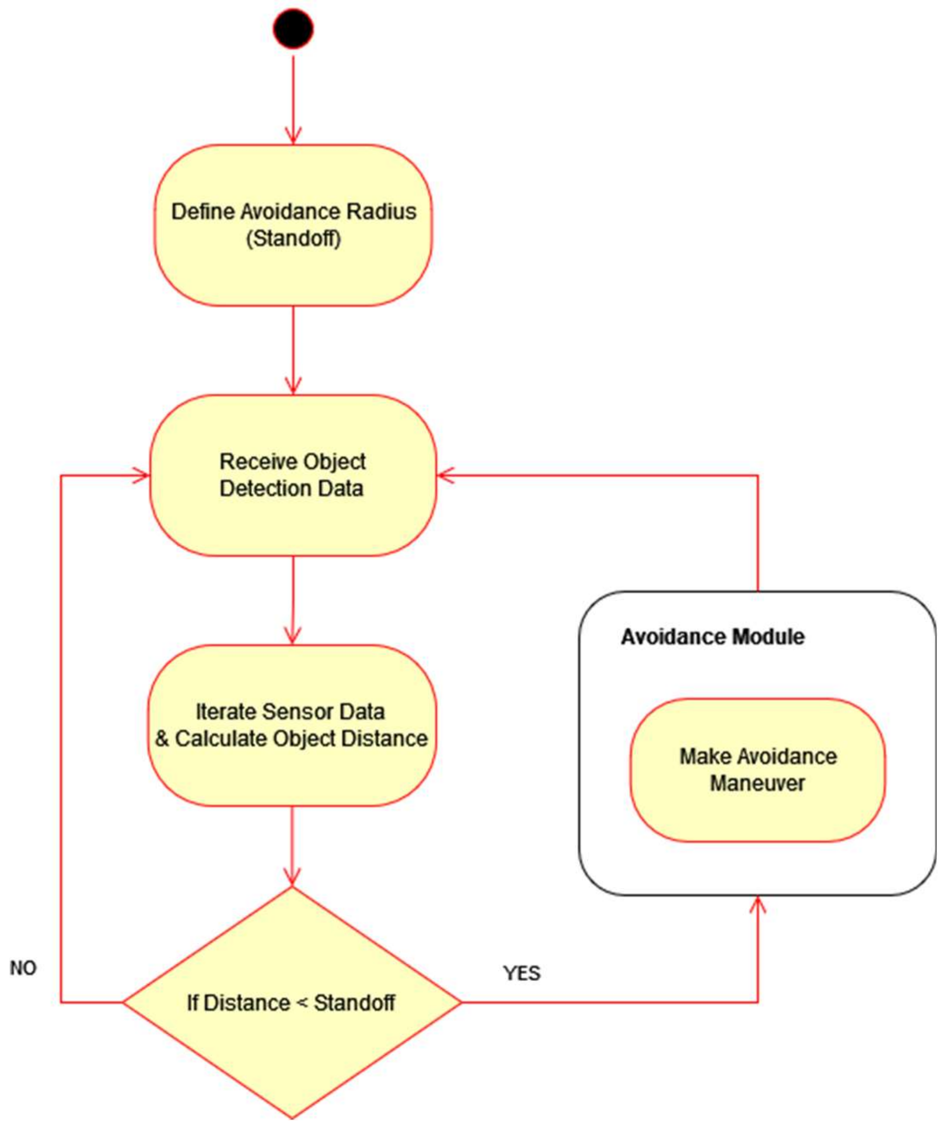
# Collision Avoidance System



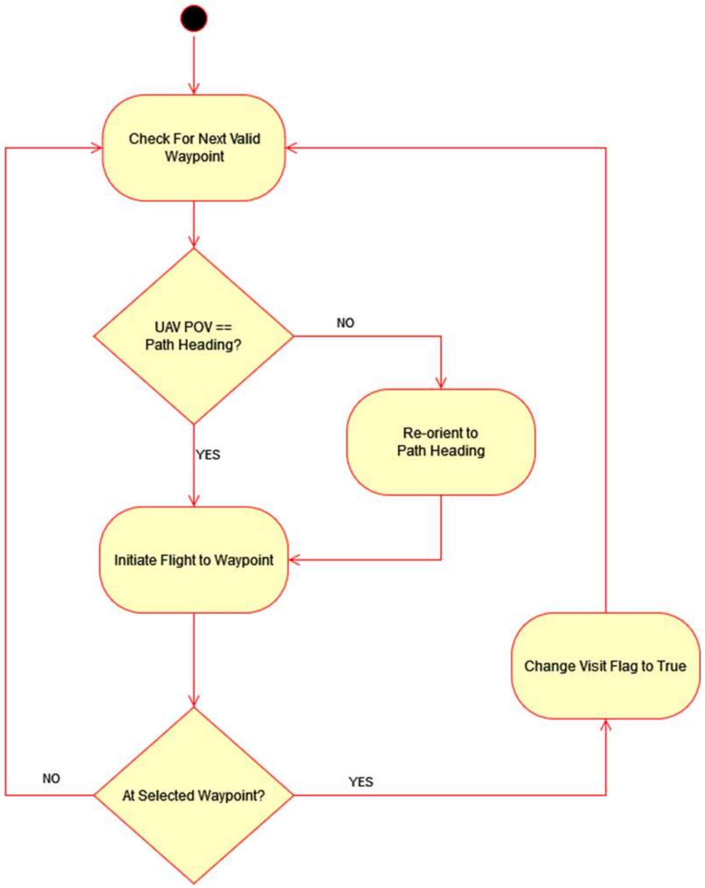
# Avoidance Behavior



# Collision Detection



# Mission Pathing


Check For Next Valid Waypoint
UAV POV == Path Heading?
NO
Re-orient to Path Heading
YES
Initiate Flight to Waypoint
Change Visit Flag to True
NO
At Selected Waypoint?
YES

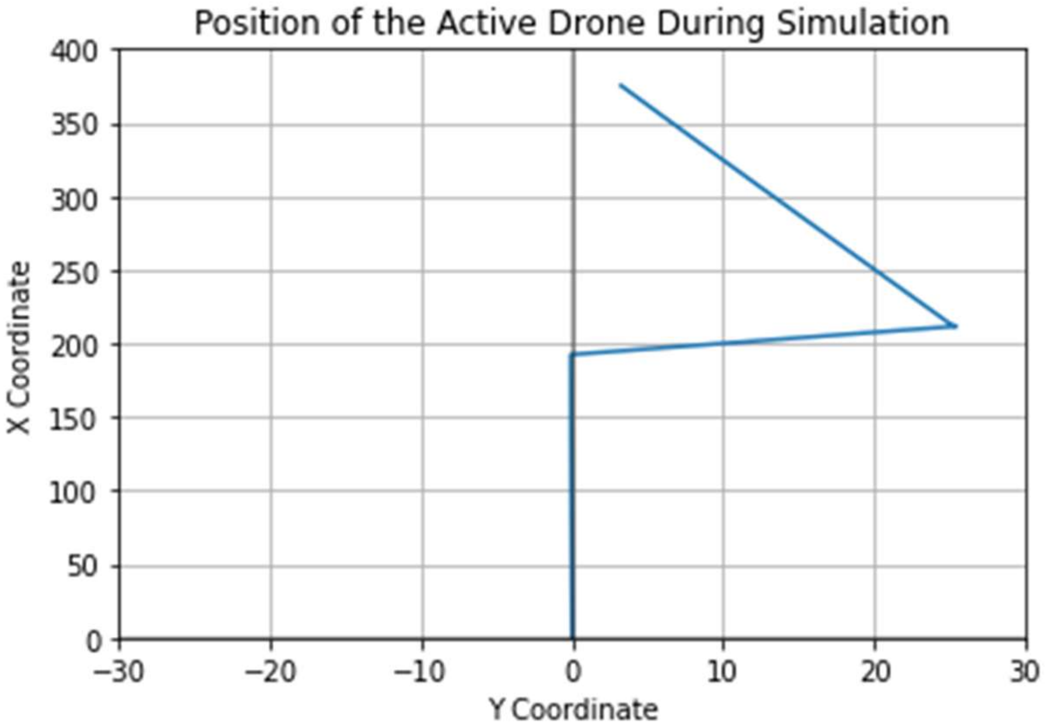
# Demonstration

# Demonstration

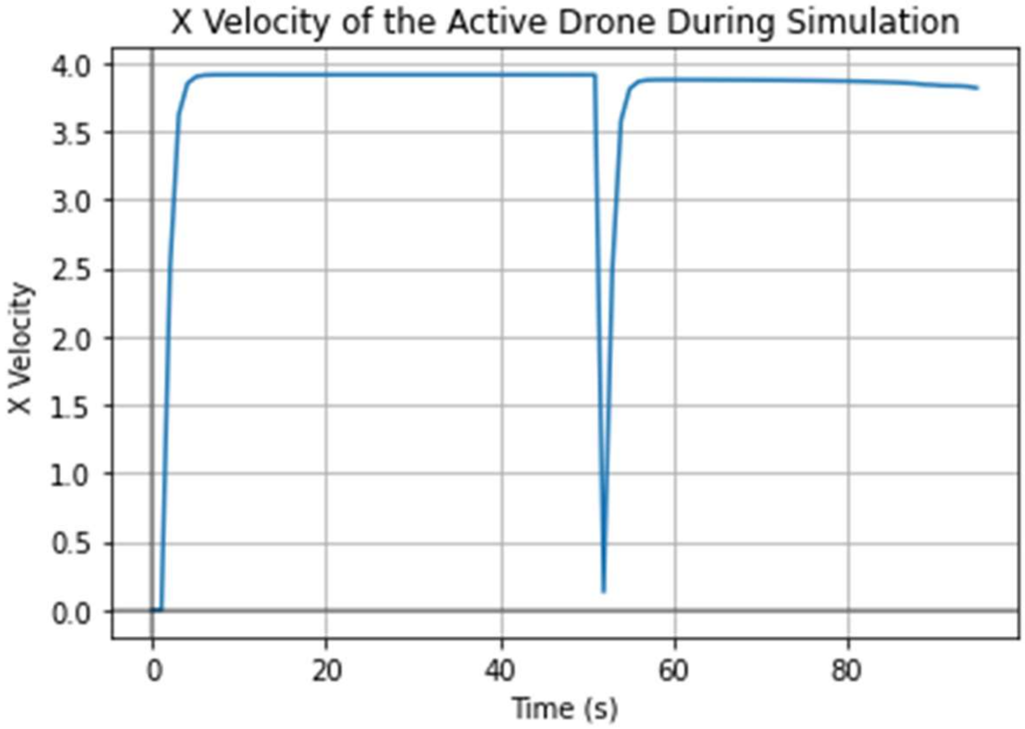
# Demonstration

# Position Plot

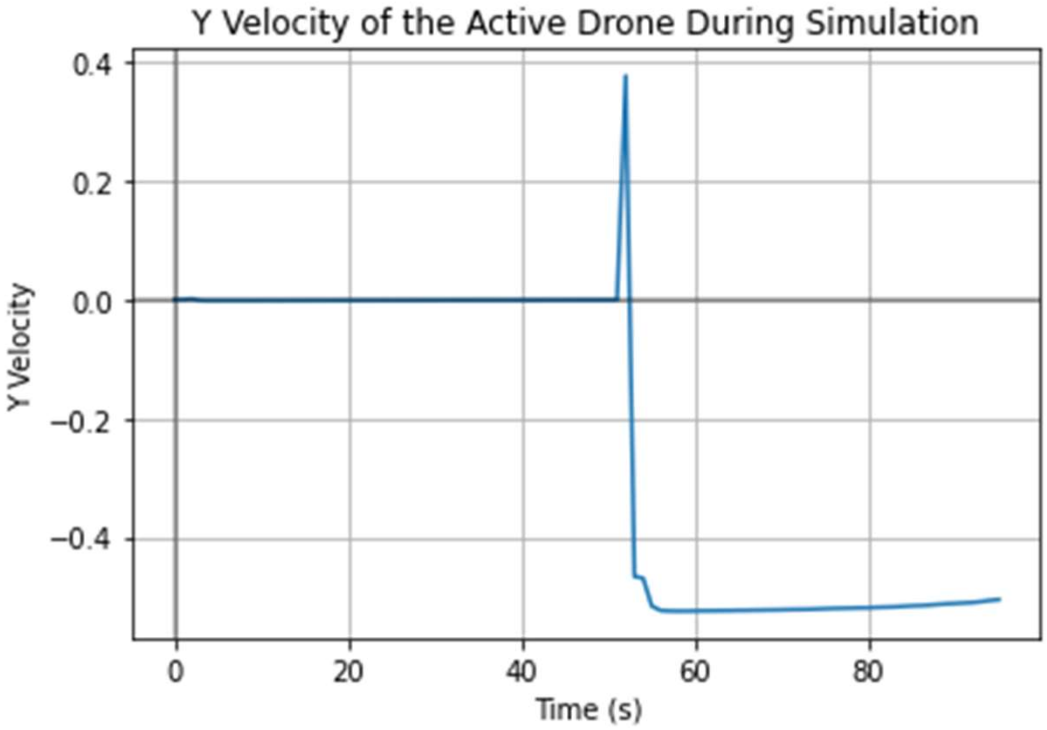
# Position Plot



# X Velocity Plot



# Y Velocity Plot



# Challenges and Lessons Learned

- Python classes passing objects through multiple classes is different than we initially thought
- Data structure for coordinate data differs between Lidar and agent kinematics
- Multi-agent behavior is difficult to solve
- AirSim appears at times to be non-deterministic
- NED sucks (lul)

# Thank You

Questions?