Modelling Unmanned Aerial Swarms Using Unreal Game Engine and AirSim Simulator

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

Introduction

The aerial swarm will be autonomous with a unified, cohesive behavior. The aerial swarm will employ a collision avoidance algorithm that will allow the swarm to maneuver around objects while collect data from a three dimensional environment.

Benefits:

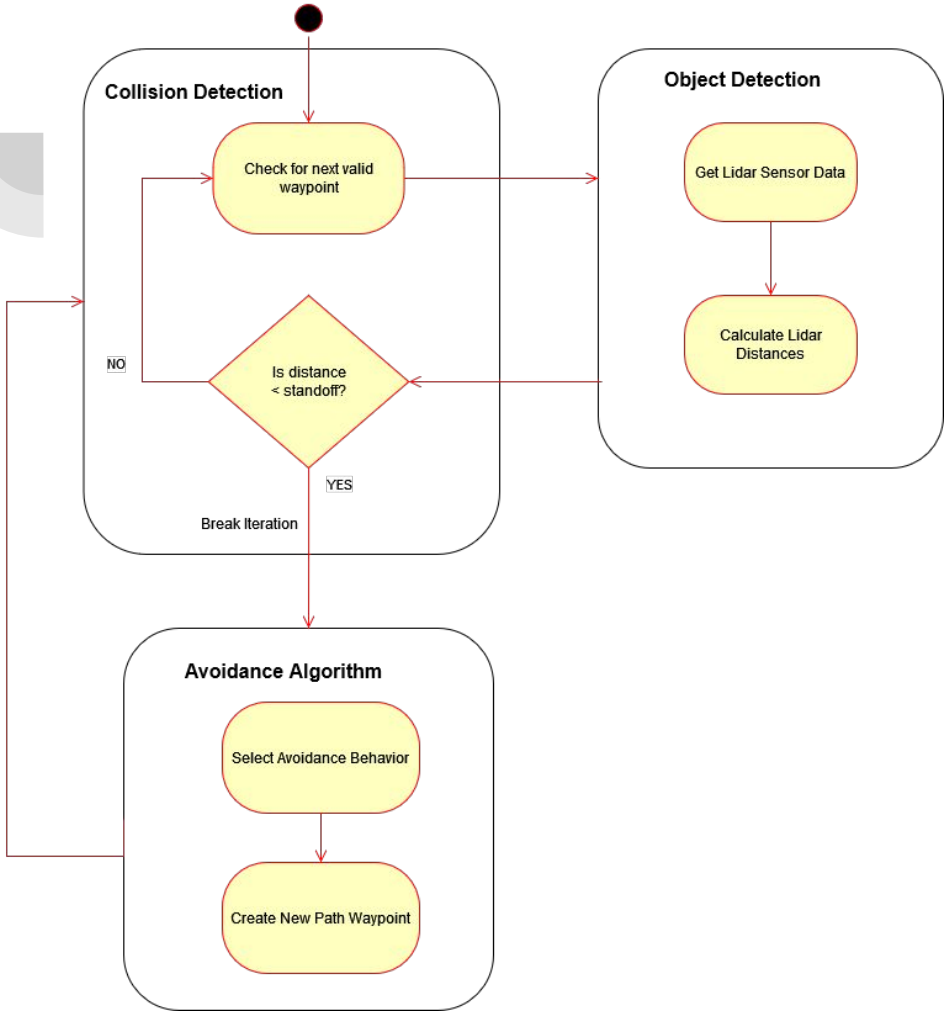
- Easier to manage swarm with autonomy
- Users allowed to focus on aspects other than piloting
- Safer for use in dangerous situations
- Easier for use in time constrained missions

Previous Work

- Created Swarm of UAV as singular entity
- Formed Swarm into formations based on No. of UAV using VSEPR geometries
- Implemented movement method for Swarm in formation
- Implemented 3-Dimensional object measurement

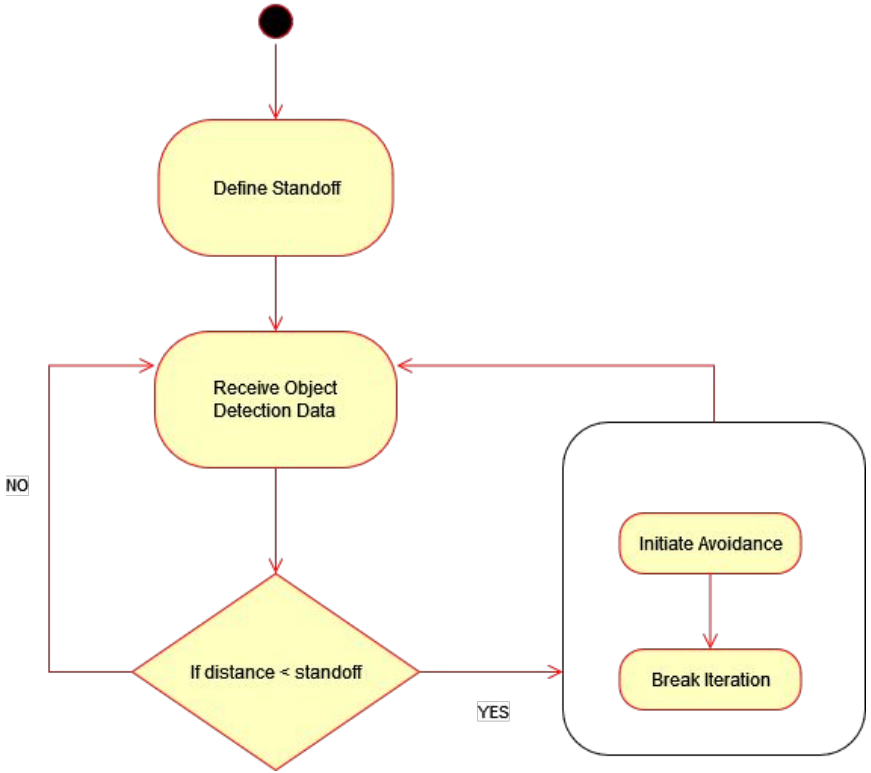
Design Considerations

- Assumptions
  - Minimum of 3 drones in swarm to measure and detect an object
  - Measure objects are solid and uniform
  - Lidar Sensor is oriented in the direction of movement
  - UAV will only move in forward direction
  - There are no adversarial actors
- Dependencies
  - Microsoft AirSim Simulator
  - Unreal Engine 4
  - Visual Studio 2019
  - Utilizes Python Environment
- Design Constraints
  - Limited to Unreal Engine 4/ AirSim environments



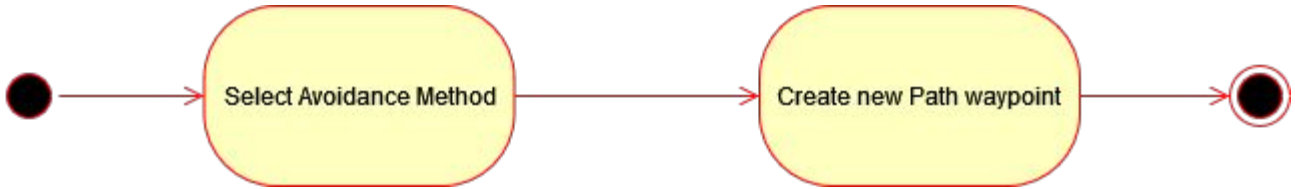
Collision Avoidance System

Collision Detection Module

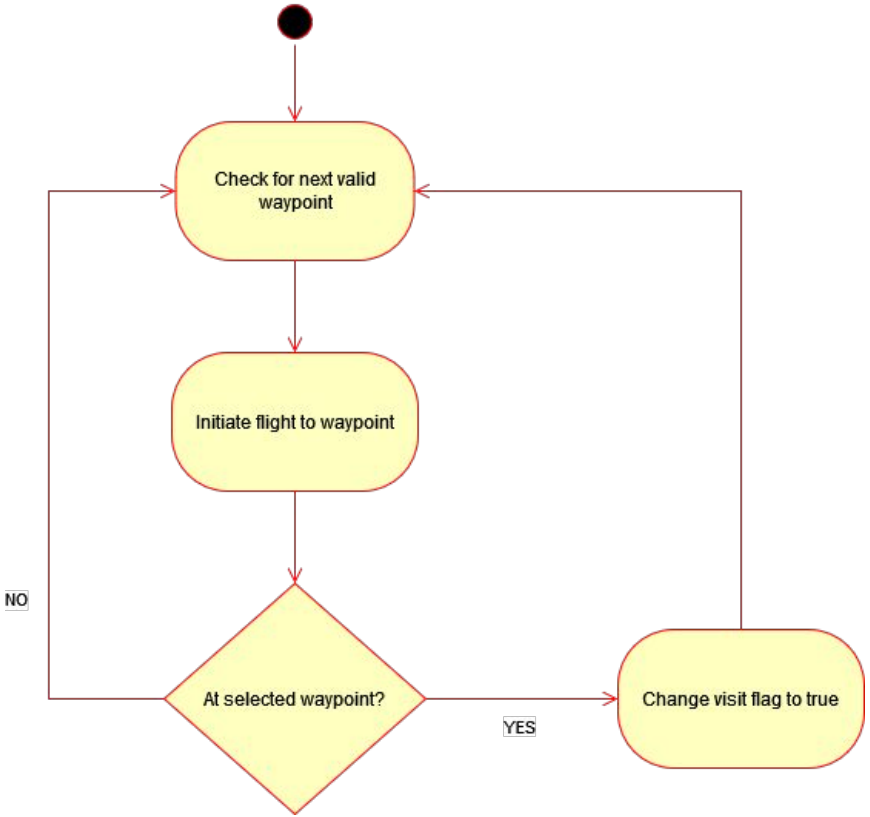


Object Detection Module

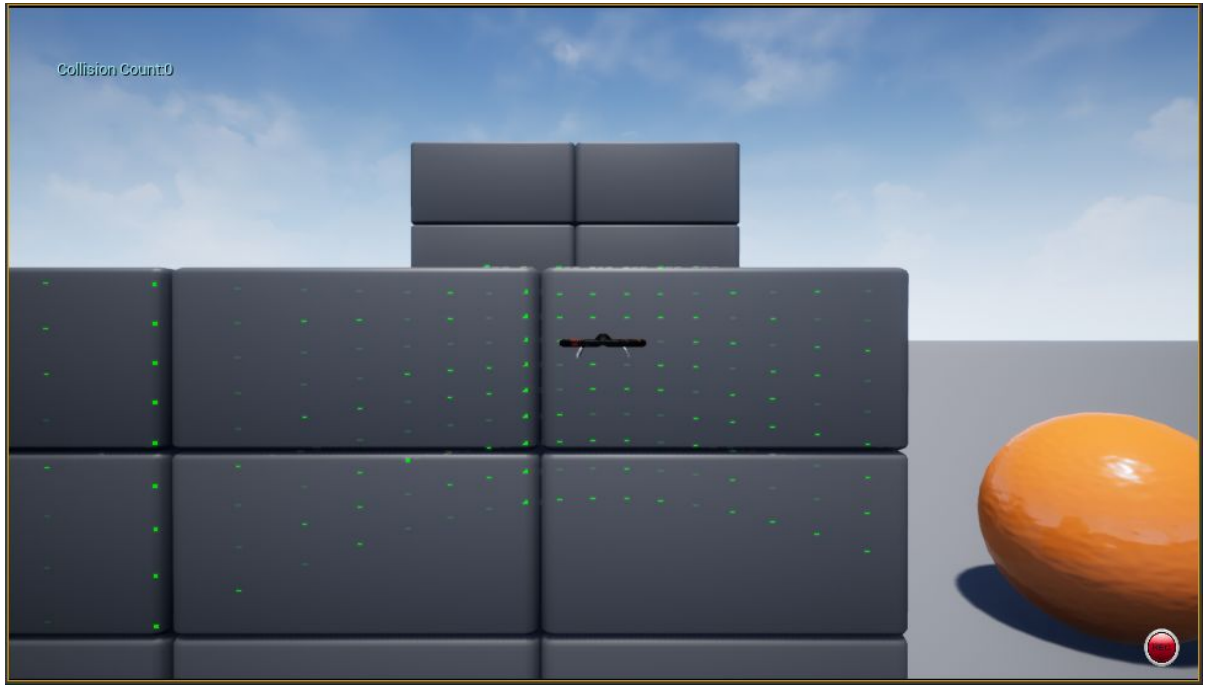
Avoidance Algorithm



Pathing Module



Lidar Detecting an Object



Lidar Readings PointCloud Data

```
Press any key to get Lidar readings
       Danger, possible collsion detected
\Time_stamp: 1644512539937357312 number_of_points: 120
            lidar position: <Vector3r> {   'x_val': 10.40916633605957,
    'y_val': 10.419997215270996,
    'z_val': -11.299046516418457}
            lidar orientation: <Quaternionr> {   'w_val': 0.7218220233917236,
    'x_val': -0.22452542185783386,
    'y_val': 0.014270582236349583,
    'z_val': -0.6544903516769409}
             PointCloud Data: array([[12.886221  ,  7.8301115 , -1.5848281 ],
      [14.874401  ,  7.5843983 , -1.7548647 ],
      [18.521404  ,  6.671424  , -0.68745995],
      [12.410933  ,  7.5413017 , -0.5071368 ],
      [ 8.859289  ,  8.192183  , -0.42136884],
      [14.26931   ,  7.27586   , -0.5593333 ],
      [ 9.931175  ,  7.8973846 , -0.4430895 ],
      [17.64355   ,  6.355218  ,  0.6548767 ],
      [11.971525  ,  7.27431   ,  0.48918223],
      [ 8.473789  ,  7.835709  ,  0.40303516],
      [13.714135  ,  6.992777  ,  0.5375705 ],
      [ 9.622591  ,  7.6519938 ,  0.42932224],
      [ 6.80538   ,  8.100973  ,  0.36946702],
      [17.008373  ,  6.12642   ,  1.9000845 ],
      [11.560235  ,  7.0243955 ,  1.4217505 ],
      [ 8.232473  ,  7.612566  ,  1.1785033 ],
      [ 5.7850785 ,  8.0314865 ,  1.04033   ],
      [13.441706  ,  6.8538694 ,  1.5858381 ],
      [ 9.33123   ,  7.420299  ,  1.253047  ],
      [ 6.627298  ,  7.888987  ,  1.0829167 ],
      [ 4.5037184 ,  8.275042  ,  0.99021244],
      [16.09867   ,  5.7987504 ,  3.0171654 ],
      [11.170773  ,  6.7877455 ,  2.3048275 ],
      [ 8.001242  ,  7.3987455 ,  1.9215709 ],
      [ 5.641165  ,  7.8316917 ,  1.7018803 ],
      [ 3.6871228 ,  8.169878  ,  1.5804827 ],
      [18.834785  ,  5.2380857 ,  3.4471216 ],
      [12.712706  ,  6.4821534 ,  2.5161748 ],
      [ 9.053053  ,  7.1990886 ,  2.0394914 ],
      [ 6.455858  ,  7.684911  ,  1.7697468 ],
      [ 4.3823166 ,  8.051984  ,  1.6164408 ],
      [ 2.6223822 ,  8.533637  ,  1.5741546 ],
      [15.402428  ,  5.547963  ,  4.0817895 ],
      [10.79791   ,  6.5611806 ,  3.1502666 ],
      [ 7.7773404 ,  7.1917033 ,  2.6410847 ],
```

Demonstration

Live demo on lab computer

Lessons Learned

- Data structure for coordinate data differs between Lidar and agent kinematics
- Python classes passing objects through multiple classes is different than we initially thought
- AirSim message packs need to be handled for errors
- Start Integration testing earlier

Project Timeline

Sprint 1 Promised

- Implement Lidar Sensor
- Object Detection
- Collision Detection
  - Executes continuously
  - Initiates Avoidance Module
- Swarm Pathing
  - Executes continuously
  - Moves swarm to next unvisited waypoint
  - Mark waypoints as visited

Sprint 1 Completed

- Lidar implemented on individual UAV
- Objects within range detected
- Collision Detection
  - Executes continuously
  - Initiates Avoidance Module
- Swarm Pathing
  - Executes continuously
  - Marks waypoint as visited

Project Timeline

Sprint 2

- Fix AirSim msg_pack bug
- Fix UAV avoidance movement response
- More complex collision avoidance behavior
- Increase precision of Object Detection

Sprint 3

- Increase size of swarm
  - implement collision avoidance on each UAV
- Run multitude of simulation scenarios
- Use scenario data for statistical analysis

Thank You

Any Questions?