User Manual
For
UAV Swarm

Team Members:

- Samantha Balistreri
- Tyler Wise
- Joseph Moran
- Michael Fornito
- William Edwards
- Daniela Regueira

Version/Author	Date
1.0 Samantha Balistreri	12/06/2020
1.0 Michael Fornito	12/08/2020

Table of Contents

1. Software Description

1.1. Purpose

The UAV Swarm Simulator is a research tool designed to simulate a swarm of up to 10 drones. The swarm is formed through image processing which utilizes the onboard cameras.

1.2. System Recommendations

- Windows 10 platform, 64-bit
- Python 3.6, Anaconda 3-5.2.0, 64-bit
- RAM: 16.0 GB
- Multi-core processor
- Gaming-grade graphic card
- Big boi memory

1.3. Acronyms and Definitions

- UAV - Unmanned Aerial Vehicle
- UAS - Unmanned Aircraft Systems
- UE4 - Unreal Engine 4
- VS - Visual Studio
- SBF - Swarm Behavior Framework
- AirSim - An open-source, cross platform simulator for drones, ground vehicles such as cars and various other objects, built on Epic Games' Unreal Engine 4 as a platform for AI research.

2. Getting Started

2.1. Required Software

- Epic Games Launcher
- Unreal Engine
- Visual Studio 2019

2.2. Installation

1. Create an Epic Games account
2. Visit https://www.epicgames.com/store/en-US/download and download the Epic Games Launcher
3. Once the launcher opens, log into your account
4. Navigate to the Unreal Engine tab and install Unreal Engine 4.24.3 or newer
5. Visit https://visualstudio.microsoft.com/downloads/ and install Visual Studio 2019
    a. Make sure to select Desktop Development with C++ and Windows 10 SDK 10.0.18362 (should be selected by default)
6. Start Developer Command Prompt for VS 2019

7. Clone the repo: git clone https://github.com/Microsoft/AirSim.git, and go the AirSim directory by cd AirSim
8. Run build.cmd from the command line
   a. This will create ready to use plugin bits in the Unreal\Plugins folder that can be dropped into any Unreal project
9. In the Epic Games Launcher, navigate to the Learn tab and find a project called Landscape Mountains
10. Click create project and download the content
11. Open LandscapeMountains.uproject
12. From the File menu select New C++ class, leave default None on the type of class, click Next, leave default name MyClass, and click Create Class.
    a. This should trigger a compile and open up Visual Studio solution LandscapeMountains.sln.
13. Go to your folder for the AirSim repo and copy Unreal\Plugins folder into your LandscapeMountains folder
    a. This adds the AirSim plugin to your project
14. Edit the LandscapeMountains.uproject so that it looks like this:

```
{
  "FileVersion": 3,
  "EngineAssociation": "4.24",
  "Category": "Samples",
  "Description": "",
  "Modules": [
    {
      "Name": "LandscapeMountains",
      "Type": "Runtime",
      "LoadingPhase": "Default",
      "AdditionalDependencies": [
        "AirSim"
      ]
    }
  ],
  "TargetPlatforms": [
    "MacNoEditor",
    "WindowsNoEditor"
  ],
  "Plugins": [
    {
      "Name": "AirSim",
      "Enabled": true
    }
  ]
}
```

15. Close Visual Studio and the Unreal Editor and right click the LandscapeMountains.uproject in Windows Explorer and select Generate Visual Studio Project Files
16. Reopen LandscapeMountains.sln in Visual Studio, and make sure "DebugGame Editor" and "Win64" build configuration is the active build configuration

17. Press F5 to runIn Window/World Settings as shown below, set the GameMode Override to AirSimGameMode
18. Go to 'Edit->Editor Preferences' in Unreal Editor, in the 'Search' box type 'CPU' and ensure that the 'Use Less CPU when in Background' is unchecked
19. Be sure to Save these edits, then hit the Play button in the Unreal Editor
20. The UAV Swarm Simulator files can be found downloaded from: https://github.com/samb519/UAVswarm
21. To change the view of the World during the simulation Left Click + M and use the arrow keys to move around.

2.3. AirSim Json Settings Setup

AirSim settings are defined in a file called settings.json which can be found in your C:/~/Documents/AirSim directory. This file can be modified manually to alter the settings of the simulation. Additionally, by implementing new settings within this file, custom configuration for new algorithms can be supported.

2.4. Python Plugins Into Unreal Engine

Install the python plugin within Unreal Engine so that the UI terminal can work within the engine.

1. Go to Edit and click Plugins
    a. Verify AirSim is installed
    b. Go to scripts and install all 3 python editor plugins
2. Go to Edit and click Project Settings
    a. Go to plugins then python
    b. Add additional paths
        i. Go to the python folder in GitHub and get its directory
        ii. Add that directory to the additional paths, make sure the array count is 0
        iii. Check mark the developer mode
        iv. Restart Unreal Engine

3. Using the System

3.1. Using Provided Algorithm

How to run Unreal Engine

1. Load the desired world whether it's the default or landscape
2. Determine **where the “settings.json”** file is being modified. This can be determined by **opening** the **ModifySettings.py**, and modifying the directory path to where the settings.json file is being used on your local machine. Most likely the file can be found in Documents>AirSim
3. Ensure there are drones in the settings file
    a. Go to the terminal and switch it Python mod
    b. Import ModifySettings as mod
    c. Type in the terminal mod.displayDrones()

   d. Verify if drones exists
   e. If not then type in mod.addDrone(XLocation, YLocation, ZLocation) -coordinate values as an int or float
   f. Re-verify that the drone exists
4. Ensure there is force in the settings file
   a. Type in the terminal mod.displayDistribution()
   b. Verify that there is a value
   c. If not type in mod.setDistribution(a number) -coordinate values as an int or float
5. Click play on the Unreal Engine
6. Execute the StartUp.py and ensure that it is connected to Unreal successfully through the users choice of IDE

3.2. Configuring the Simulation

To configure the simulation, the terminal in Unreal will need to be used. The available terminal commands are as follows:

- addDrone() = is a method that takes 3 (int/float) input parameters (X,Y,Z) and will add the drone to the last position of the list. User cannot add more than 10 to the list and the UAV ID is randomly generated
- removeDrone() = is a method that removes a drone based on the given index (must be an integer). The user can not remove all the drones, must have at least 1 drone in the list
- displayDrones() = will display information (Drone Name, Location, Drone Type) for all the drones in the settings file
- displayDistribution() = will display the force/distribution size for the swarm algorithm
- setDistribution() = is a method that allows the user to set the force/distribution size (int/float) for the swarm algorithm

3.3. Implementing Custom Algorithms

To implement a custom algorithm, the interface outlined in AbstractDroneControlAlgorithm should be followed. Implementing this interface will make a new object of the new data type compliant with the constructor for the Drone object. Then, through modifying the parameters in the settings.json file and augmenting it with additional parameters to suit the new algorithm, a new swarming algorithm can be realized.

3.4. Collecting Data

After the simulation has ended and disconnected from the StartUp.py, go to the python folder to locate the newly created log files

1. A json log file will be created that contains all the drones with their positions across the entire simulation time

2. CSV files are created for each individual drone with their positions across the entire simulation time
3. Scatter plot images are created for each individual drone as a .png
4. Scatter plot images are created for the all the drones together as a .png

4. Troubleshooting and Support

4.1. Installation Support

4.1.1. Installation Problems

4.1.1.1. Installing AirSim with Unreal

1. Open up Microsoft VS 2019 console command and go change your directory to the path of the GitHub repo
2. Once the path has been changed to the repo type: build.cmd
3. CMake may be asked to be installed on your computer, accept the download

4.1.2. Settings File Location

The python code's have a directory path that reads the settings.json file. The directory path must change to the local settings.json file that Unreal uses within the AirSim folder or replace the setting.json file in the local AirSim folder every time there is a change to it in GitHub repo

4.1.3. Additional Info for installation

https://github.com/samb519/UAVswarm/wiki/How-to-install-AirSim-and-Unreal-Engine-on-Windows-OS

5. References

Building AirSim on Windows: https://microsoft.github.io/AirSim/build_windows/
Unreal Environment Setup: https://microsoft.github.io/AirSim/unreal_custenv/
AirSim Settings: https://microsoft.github.io/AirSim/settings/
AirSim APIs: https://microsoft.github.io/AirSim/apis/