**Implement Vehicle into AirSim**

Disclaimer: This is by no means guaranteed to work, however since there is currently no guide I figured I would create one.

1. Find a model of what you want to use and ensure it is in .FBX format. Once in .FBX format import into Unreal Engine and keep it as a separate model somewhere that can be easily accessed (I usually create a separate folder in /AirSim/Content/Models). Once imported save the project to ensure it keeps the files imported.
2. Depending on the kind of vehicle you may want to make new vehicle initializer classes within MultiRotorParams.hpp. For example I have been attempting to implement a ground vehicle, so I created a new class called initializeGroundRover and based it around the initializeRotorQuadX class. In this new class you can change how many rotors there are/ will be generated (reference the Hexacopter initialize class to see an example of 6) so you can add or subtract from the number of rotors this will however mean that you must later add/ subtract from the model in Unreal Engine (Reference <https://github.com/Microsoft/AirSim/wiki/hexacopter> for a semi viable tutorial/ explanation). The Vector3r unit\_z variable is what is in charge of thrust direction so if you need to change rotor/ thrust directions, modify that variable. I use <https://academo.org/demos/3d-vector-plotter/> in order to visualize the direction of thrust (default is (0, 0, -1) meaning that there is only downwards thrust). You can also modify the physics of the rotor by going into rotorParams.hpp and modifying and relevant variables (such as propeller\_diameter, max\_rpm, etc.). In PX4MultiRotor.hpp you must in the private block setup a “void setupFrameMODELNAME”, simply base it around the already existing setupFrameGenericQuad and have it call whatever you named the previous initialize class. Add to the if block by adding an else if (connection\_info\_.model == “MODEL NAME HERE”)

for example I have

if (connection\_info\_.model == “Rover”){

setupFrameRover(Params);

}

and

void setupFrameRover(Params&params){

…

initializeGroundRover(params.rotor\_poses, params.rotor\_count…);

…

}

1. You must insure in the settings.json file to set MODEL = “NAME”. Once this is complete you may reopen the Unreal project and go to the BP\_FlyingPawn and switch the model to whichever model you like. You will most likely also have to modify the number of rotors, their position, and orientation depending on what you are attempting to make.
2. This new model should now work assuming everything was done correctly. The next step would be to find and build the correct PX4 version for your model. For a rover it is simple for a SITL simulation as you can simply replace the last part in the build command for PX4 “iris” with “rover” and if I remember correctly this will build and run the relevant PX4 config for a rover.
3. It should now run in UnrealEngine successfully and your custom vehicle should be implemented, however it may work as expected (in terms of physics, or animation) so tweaks will be required.