**1. Airsim Build/Demo steps**

AirSim is a simulator for drones, cars and more, built on [Unreal Engine](https://www.unrealengine.com/) . It is open-source, cross platform, and supports hardware-in-loop with popular flight controllers such as PX4 for physically and visually realistic simulations. It is developed as an Unreal plugin that can simply be dropped into any Unreal environment.

* Latest release v1.3.1 for [Windows](https://github.com/microsoft/AirSim/releases/tag/v1.3.1-windows) and [Linux](https://github.com/microsoft/AirSim/releases/tag/v1.3.1-linux)
* Upgraded to Unreal Engine 4.24, Visual Studio 2019, Clang 8, C++ 17 standard.
* Support for [ArduPilot](https://ardupilot.org/ardupilot/) - [Copter, Rover vehicles](https://ardupilot.org/dev/docs/sitl-with-airsim.html)
* Updated [airsim](https://pypi.org/project/airsim/) Python package, with lots of new APIs
* ROS wrapper for multirotors is available
* [Plotting APIs for Debugging](https://github.com/microsoft/AirSim/pull/2304)
  1. **How to Get It**

Users can either use pre-compiled binaries or build the Unreal Engine and Airsim framework locally.

* D[ownload binaries](https://microsoft.github.io/AirSim/use_precompiled) :  You can simply download precompiled binaries and run to get started immediately.
* [Build it](https://microsoft.github.io/AirSim/build_windows) locally

**1.1.1 Precompiled Binaries:** Precompiled binaries are provided for both linux and windows. Users can just download the zip files and run the application.

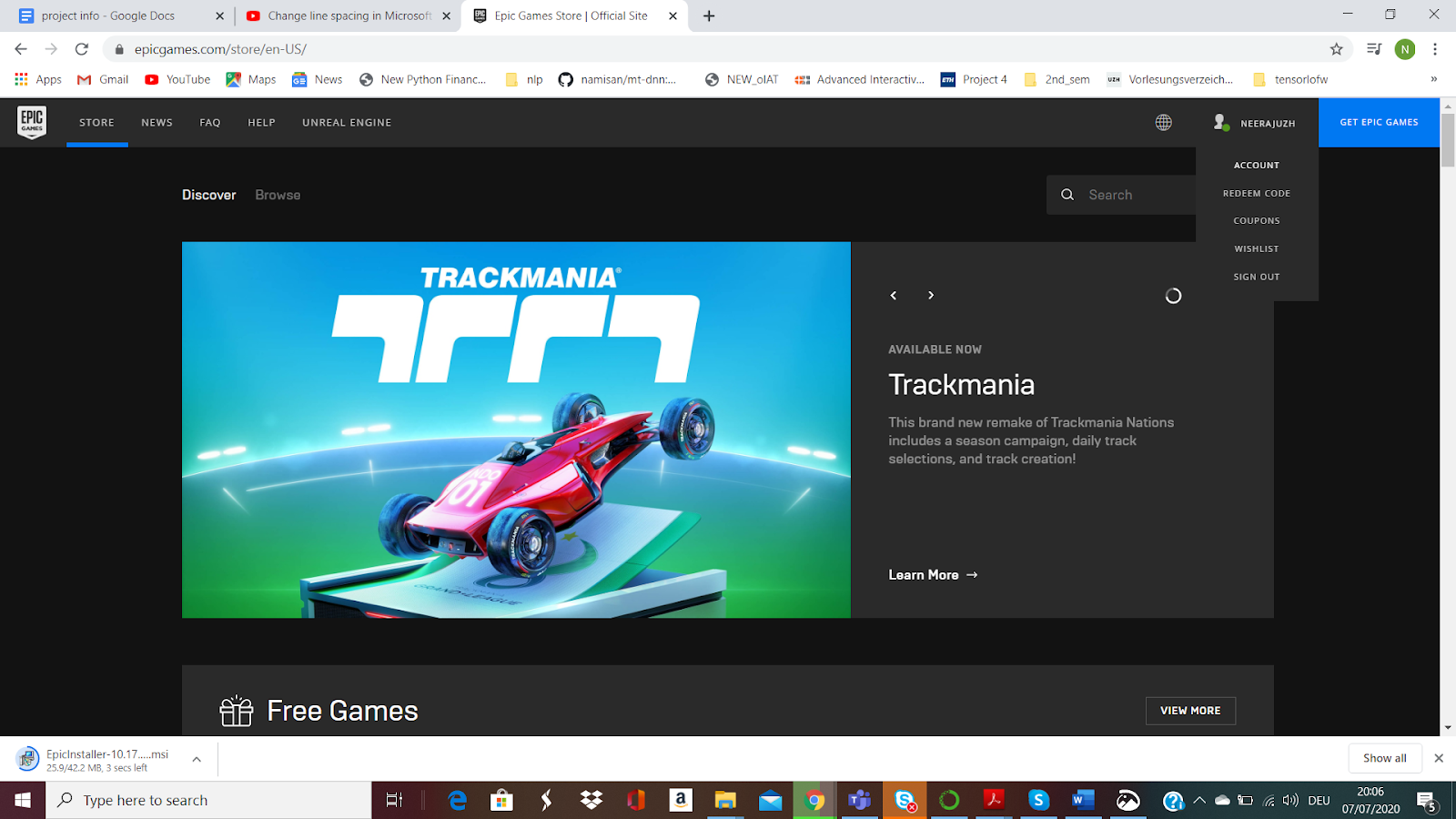
<https://github.com/microsoft/AirSim/releases/tag/v1.3.1-windows><https://github.com/microsoft/AirSim/releases/tag/v1.3.1-linux>

**1.1.2 Build Airsim Framework on Linux**

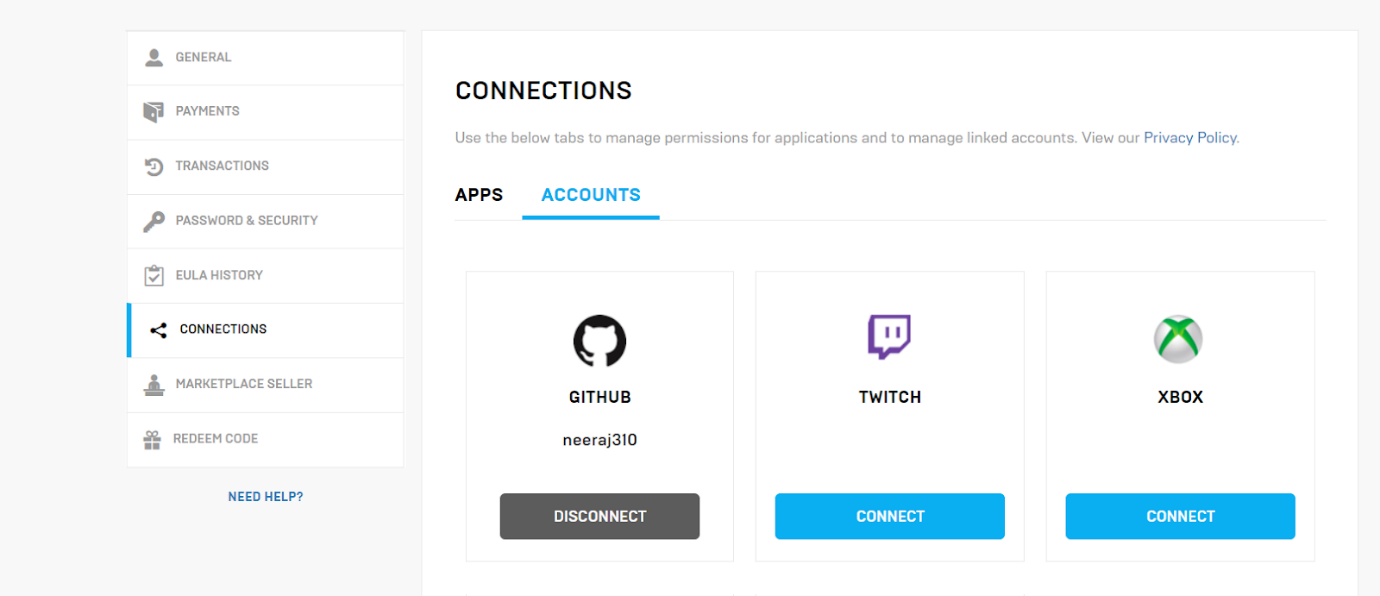
The current recommended and tested environment is **Ubuntu 18.04 LTS**. Theoretically, you can build on other distros as well, but we haven't tested it.

Following steps needs to be followed.

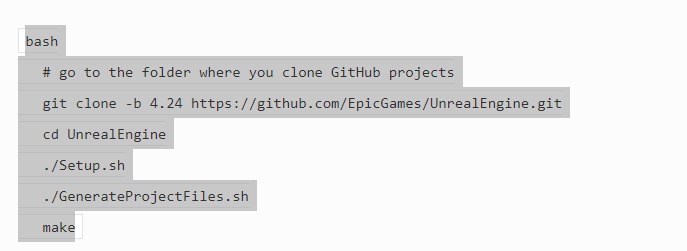
* **Build Unreal Engine**
* Make sure you are [registered with Epic Games](https://docs.unrealengine.com/latest/INT/Platforms/Linux/BeginnerLinuxDeveloper/SettingUpAnUnrealWorkflow/1/index.html). This is required to get source code access for Unreal Engine.
* Go to account settings



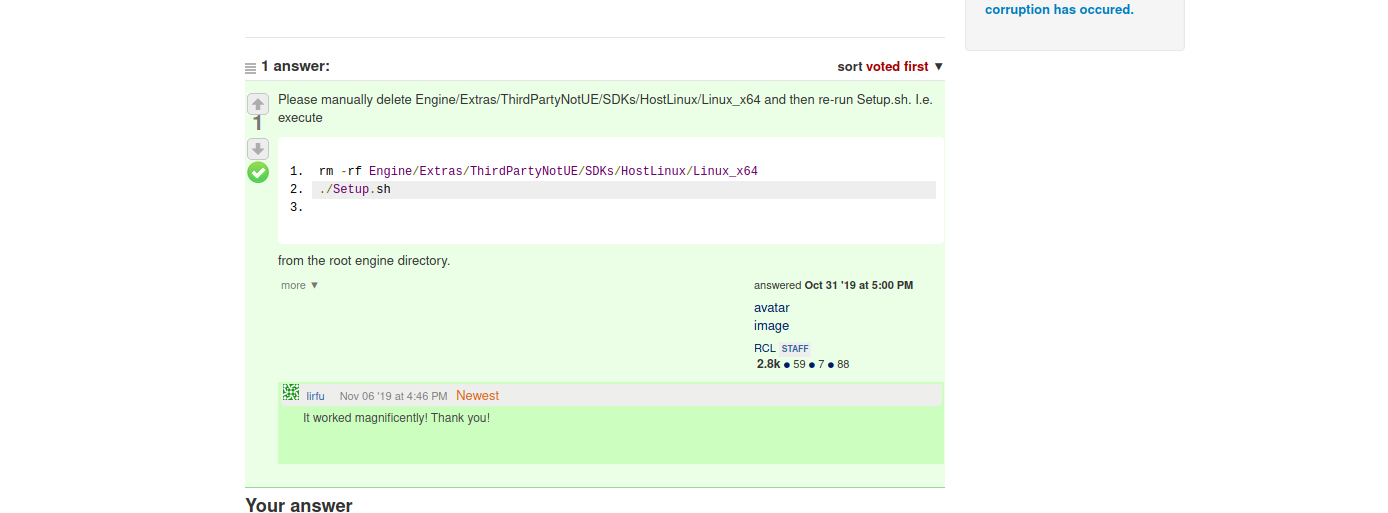
* Create a connection with user github repository



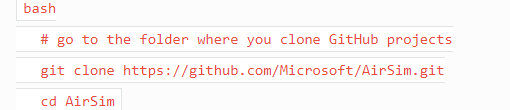
* Clone Unreal in your favorite folder and build it (this may take a while!).

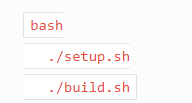


If user is building Unreal Engine on Ubuntu 20.04  LTS , use following trick to resolve the make error

****

* **Build Airsim**
  + Make sure you are [registered with Epic Games](https://docs.unrealengine.com/latest/INT/Platforms/Linux/BeginnerLinuxDeveloper/SettingUpAnUnrealWorkflow/1/index.html). This is required to get source code access for Unreal Engine.





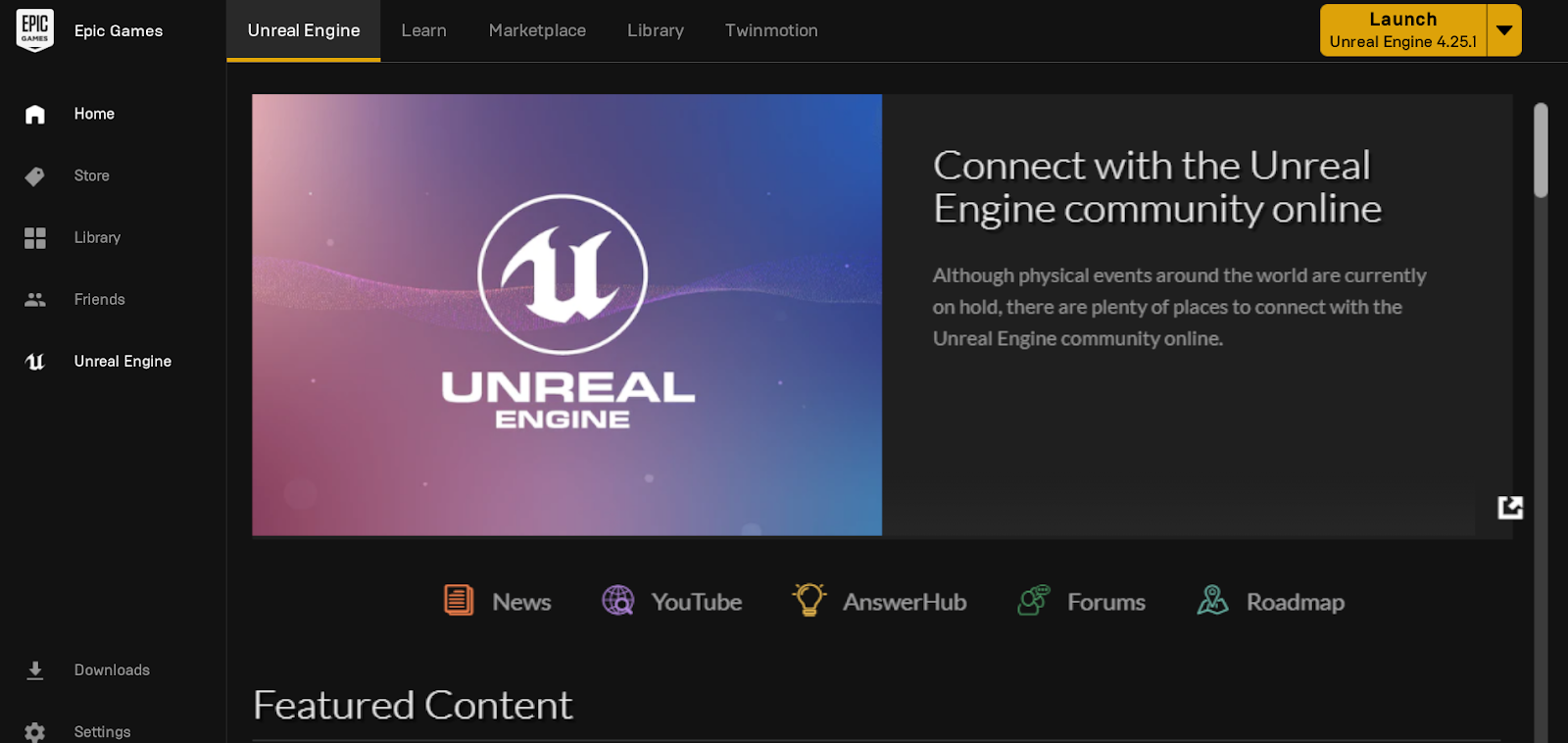
By default AirSim recommends using clang 8 to build the binaries as those will be compatible with UE 4.24. The setup script will install the right version of cmake, llvm, and eigen.

**1.1.3 Build Airsim Framework on Windows**

The current recommended and tested environment is **Ubuntu 18.04 LTS**. Theoretically, you can build on other distros as well, but we haven't tested it.

Following steps needs to be followed.

* **Install Unreal Engine**
* [Download](https://www.unrealengine.com/download) the Epic Games Launcher. While the Unreal Engine is open source and free to download, registration is still required.
* Run the Epic Games Launcher, open the Library tab on the left pane. Click on the Add Versions which should show the option to download **Unreal 4.24** as shown below.



* **Build Airsim**
* Install Visual Studio 2019. Make sure to select Desktop Development with C++ and Windows 10 SDK 10.0.18362 (should be selected by default) while installing VS 2019.
* Start Developer Command Prompt for VS 2019.
* Clone the repo: git clone https://github.com/Microsoft/AirSim.git, and go the AirSim directory by cd AirSim
* Run build.cmd from the command line. This will create ready to use plugin bits in the Unreal\Plugins folder that can be dropped into any Unreal project.

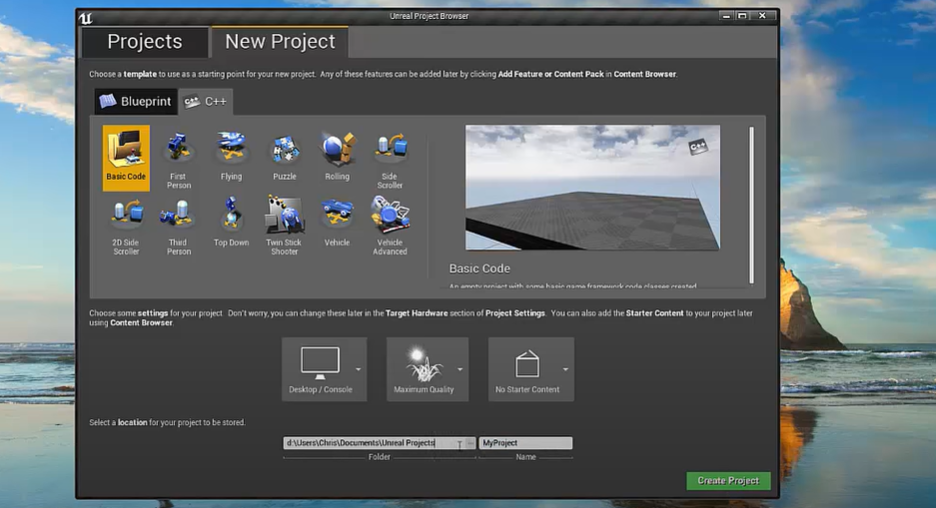
**1.2 Build Unreal Project**

You will need an Unreal project that hosts the environment for your vehicles. AirSim comes with a built-in "Blocks Environment" which you can use, or you can create your own. The Unreal Marketplace has [several environments](https://www.unrealengine.com/marketplace/content-cat/assets/environments) available that you can start using in just few minutes. It is also possible to use environments available on websites such as [turbosquid.com](https://www.turbosquid.com/) or [cgitrader.com](https://www.cgtrader.com/) with bit more effort (here's [tutorial video](https://www.youtube.com/watch?v=y09VbdQWvQY&feature)). In addition there also several [free](https://github.com/Microsoft/AirSim/issues/424).

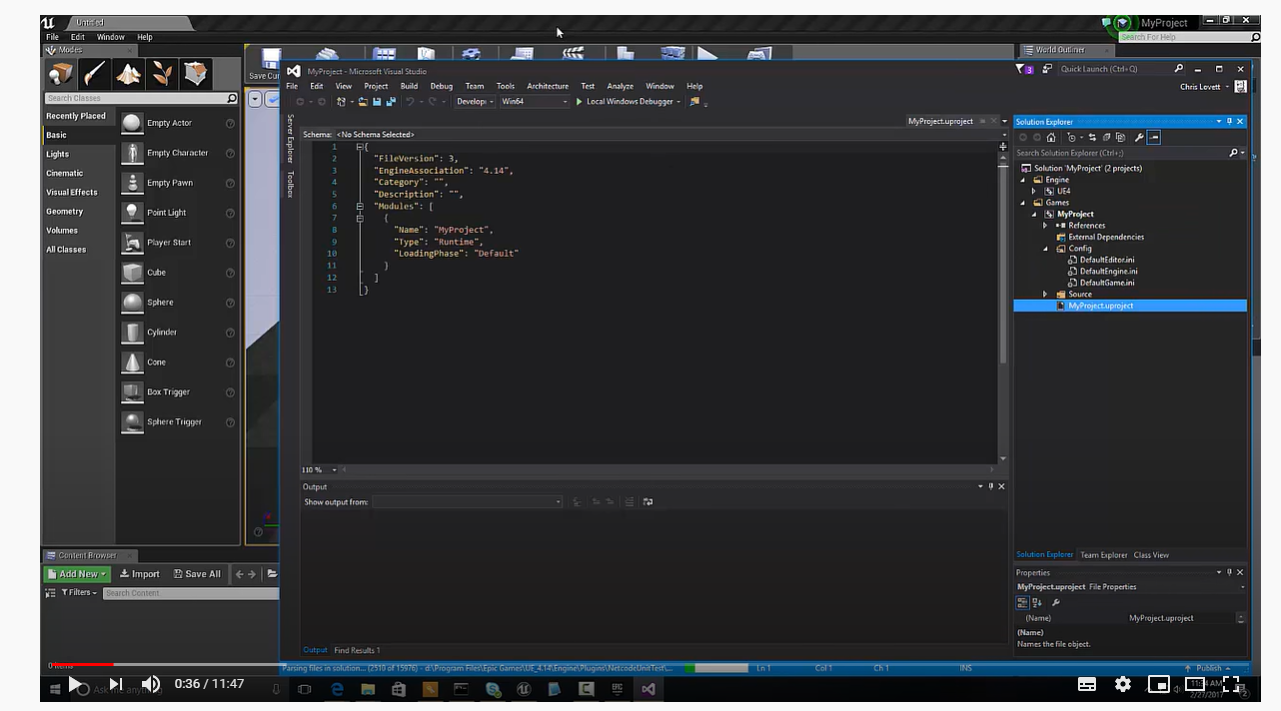
Below we will use a freely downloadable environment from Unreal Marketplace called Landscape Mountain but the steps are same for any other environments. You can also view these steps performed in [Unreal AirSim Setup Video](https://youtu.be/1oY8Qu5maQQ).

**Step by step instructions:**

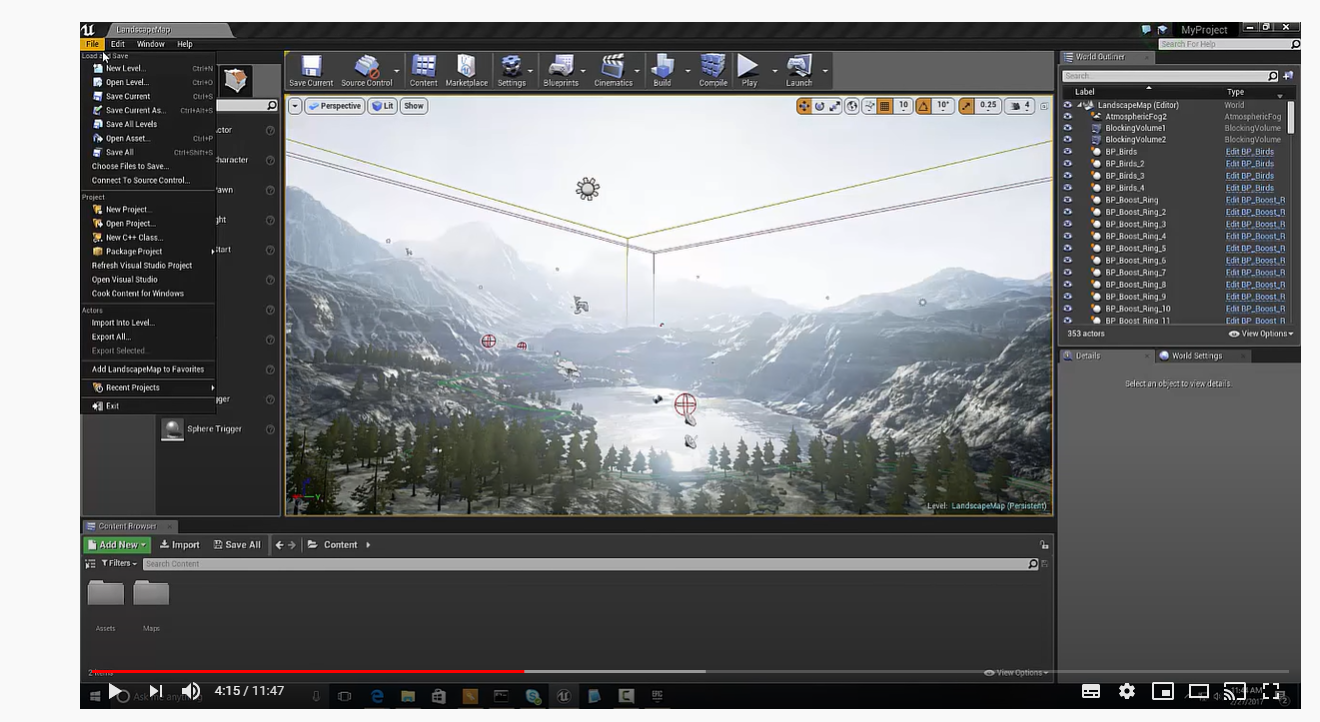
1. Make sure AirSim is built and Unreal 4.24 or above is installed as described in [build instructions](https://microsoft.github.io/AirSim/build_windows/).
2. Create a new C++ project in unreal editor.

****

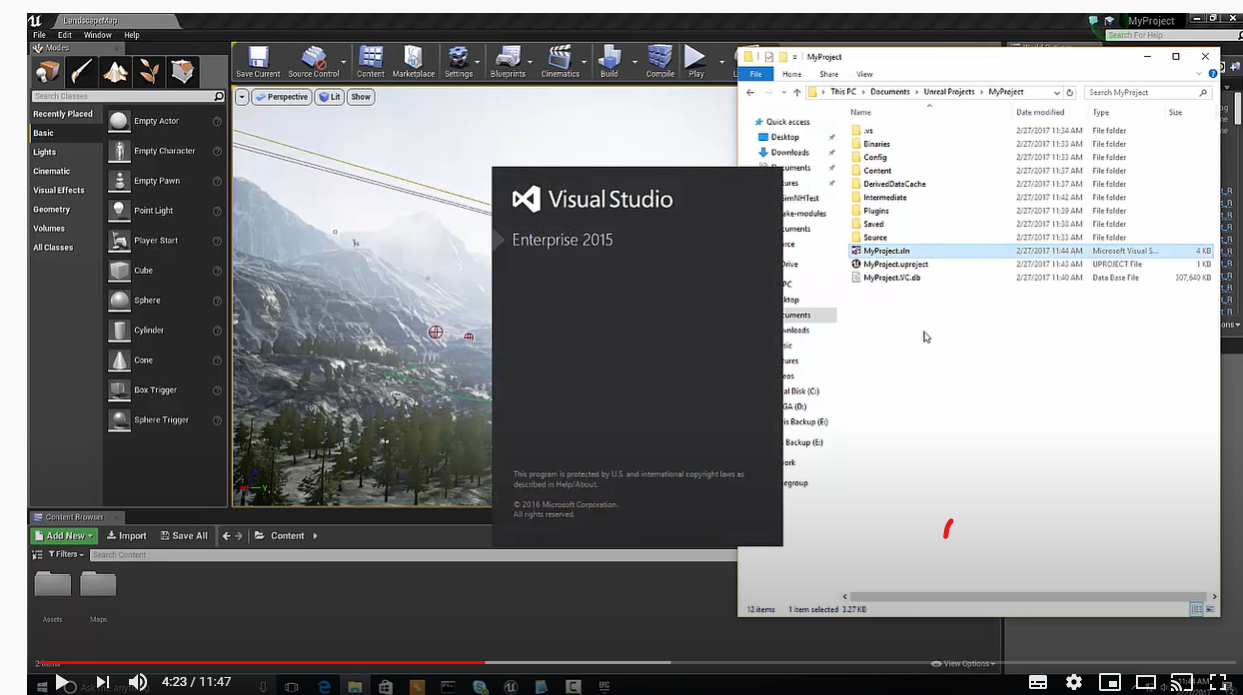
1. This will launch visual studio with new c++ project.

****

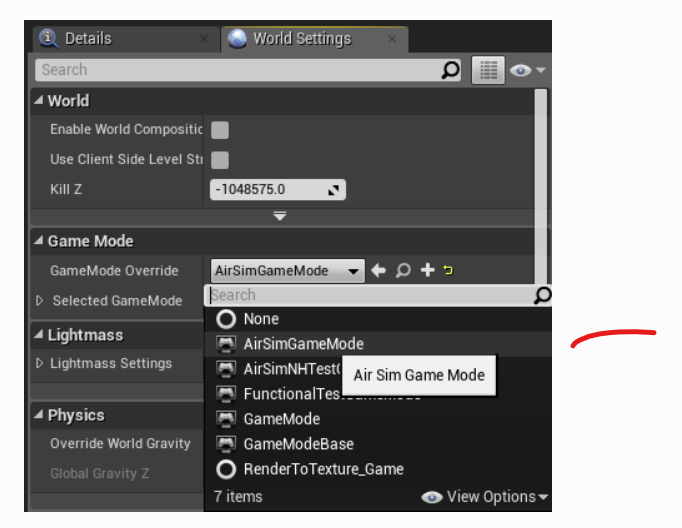
1. In Epic Games Launcher click the Learn tab then scroll down and find Landscape Mountains( User is free to choose any other environment). Click the Create Project and download this content (~2GB download)
2. Merge the downloaded content with the my project content. Follow the video link instructions to know the steps to follow for merging. We will add both the original config files and changes files on github, so users can cross check whether they have merged all the required changes.
3. Go to your folder for AirSim repo and copy Unreal\Plugins folder in to your my\_project folder. This way now your own Unreal project has AirSim plugin.
4. Edit the my\_project.uproject so that it looks like this
5. Launch unreal editor, go to file menu to refresh visual studio



1. Load visual studio. My project environment is setup now.



1. In Window/World Settings as shown below, set the GameMode Override to AirSimGameMode:



* 1. **Control Unreal Environment using python**

Airsim exposes APIs so you can interact with vehicle in the simulation programmatically. You can use these APIs to retrieve images, get state, control the vehicle and so on. If you want to use Python to call AirSim APIs, we recommend using Anaconda with Python 3.5 or later versions however some code may also work with Python 2.7 ([help us](https://microsoft.github.io/AirSim/CONTRIBUTING/) improve compatibility!).

First install this package:

pip install Airsim

Here's how to use AirSim APIs using Python to control simulated quadrotor

*# ready to run example: PythonClient/multirotor/hello\_drone.py*

**import** airsim

*# connect to the AirSim simulator*

client = airsim.MultirotorClient()

client.confirmConnection()

client.enableApiControl(**True**)

client.armDisarm(**True**)

*# Async methods returns Future. Call join() to wait for task to complete.*

client.takeoffAsync().join()

client.moveToPositionAsync(-10, 10, -10, 5).join()

*# take images*

responses = client.simGetImages([

airsim.ImageRequest("0", airsim.ImageType.DepthVis),

airsim.ImageRequest("1", airsim.ImageType.DepthPlanner, **True**)])

print('Retrieved images: %d', len(responses))

*# do something with the images*

**for** response **in** responses:

**if** response.pixels\_as\_float:

print("Type %d, size %d" % (response.image\_type, len(response.image\_data\_float)))

airsim.write\_pfm(os.path.normpath('/temp/py1.pfm'), airsim.getPfmArray(response))

**else**:

print("Type %d, size %d" % (response.image\_type, len(response.image\_data\_uint8)))

airsim.write\_file(os.path.normpath('/temp/py1.png'), response.image\_data\_uint8)