Multicopter Design and Control Practice Experiments

RflySim Advanced Courses
Lesson 05: UE4 3D Scene Development

Dr. Xunhua Dai, Associate Professor,
School of Computer Science and Engineering,
Central South University, China;
Email: dai.xh@csu.edu.cn ;
https://faculty.csu.edu.cn/daixunhua

北航可靠飞行控制研究组
BUAA Reliable Flight Control Group

Content

1. Setup Instruction
2. UE4 3D scene build
3. RflySim3D 3D scene import
4. Vehicle 3D model build and import
5. Control demo of objects in scene
6. Summary

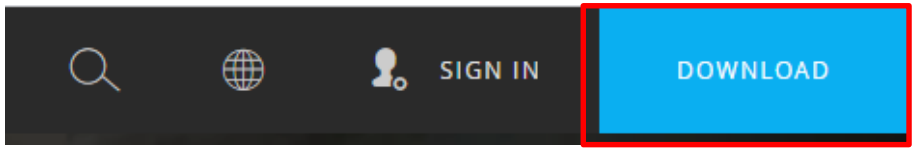
Path of source code in this Lesson:
"RflySimAPIs\UE4MapSceneAPI"

1. Setup Instruction

Note: If you failed to download the file, you can use the installer "RflySimAPIs\UE4MapSceneAPI\EpicInstaller-******.msi"

1.1 Components to be installed

- **3Ds Max 2020** (or other versions, please install by yourself)
- **Visual Studio 2017** (not required to develop UE4 scenes, but required for UE4 C++/plugin programming development)
- **Unreal Engine 4.22** (UE4.22 this version only)



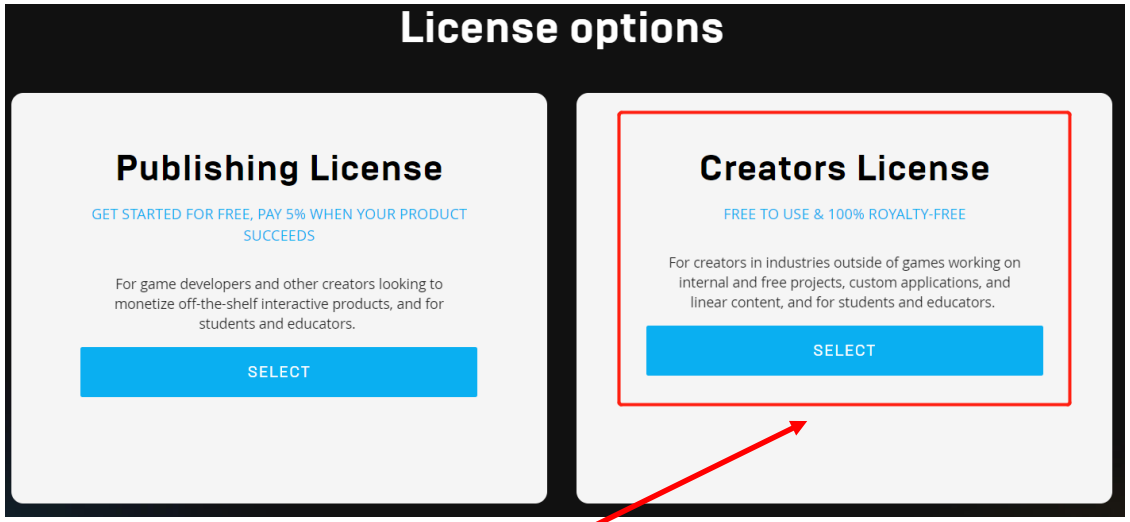
The following describes how to install **UE4.22** engine (internet connection required):

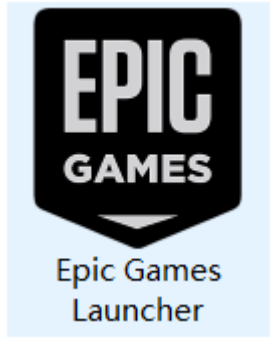
- Open the official website of the **EPIC Unreal Engine**:
- https: //www.unrealengine.com/en-US/?lang=en-US
- Register an EPIC account and log in
- Then, click the "**DOWNLOAD**" button on the upper right

1. Setup Instruction

1.2 How to install Epic Games

- After clicking the "**DOWNLOAD**" button, the following selection box will pop up



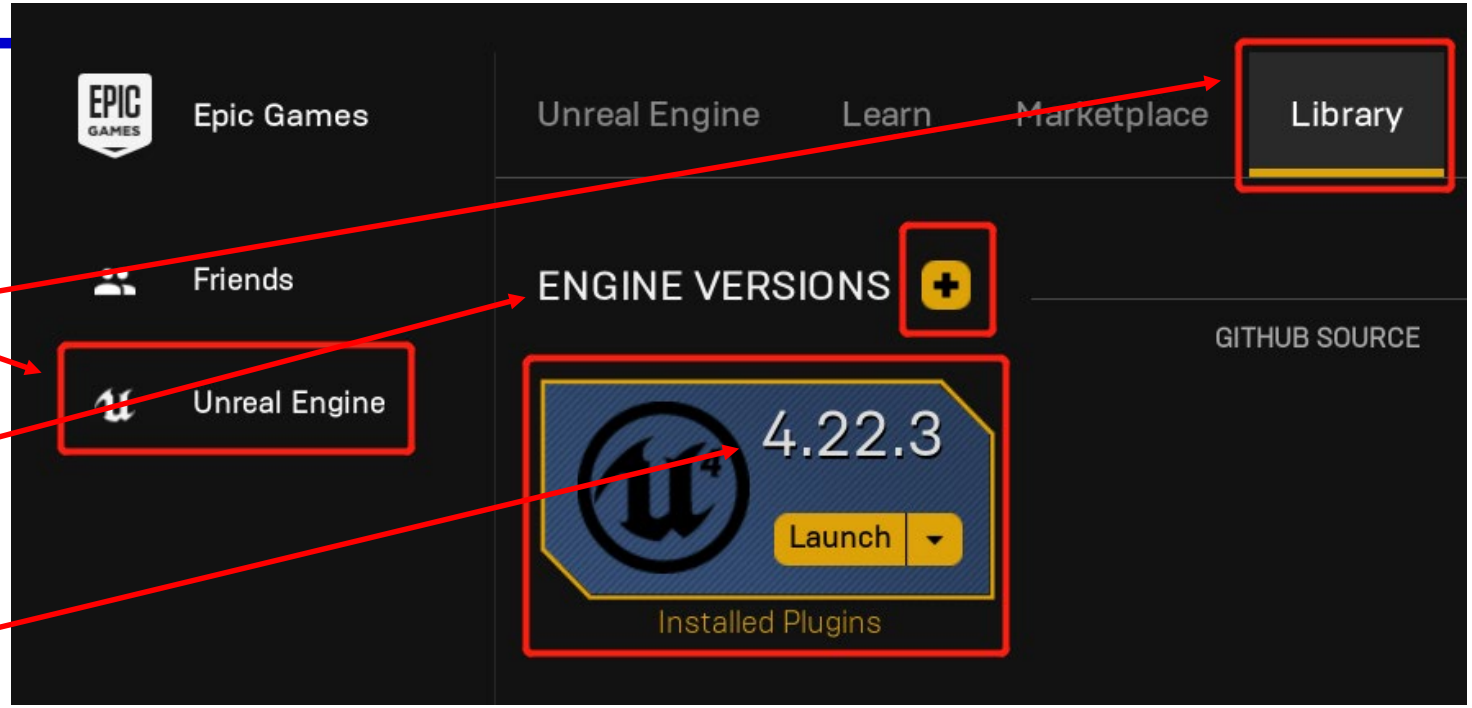


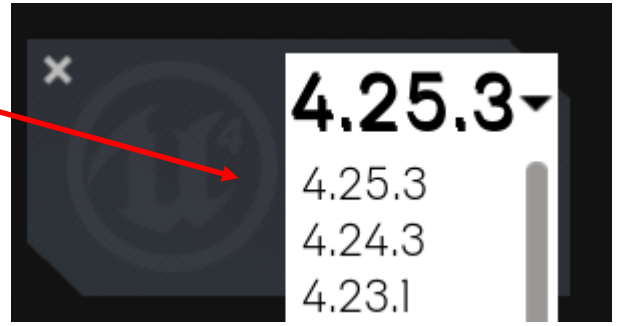
- Select the "**Creators License**" on the right to download the latest installation package.
- Click the installation package to install, you can get "**Epic Games Launcher**" shortcut
- Click this shortcut and log in to your Epic account, you can enter the Epic Games management page

1. Setup Instruction

1.2 How to install UE4.22

- In the Epic Games program, click "**Unreal Engine**" icon in the left column, then click the "**Library**" page on the right, the "**+**" icon after "**ENGINE VERSIONS**" tab.
- Select the UE4 version to be installed in the drop-down box (**Note**: Please install the **4.22.3** version engine). Wait patiently for the end of the installation (**Note**: these need to download more than 10G resources, the entire installation process takes about an hour)





Content

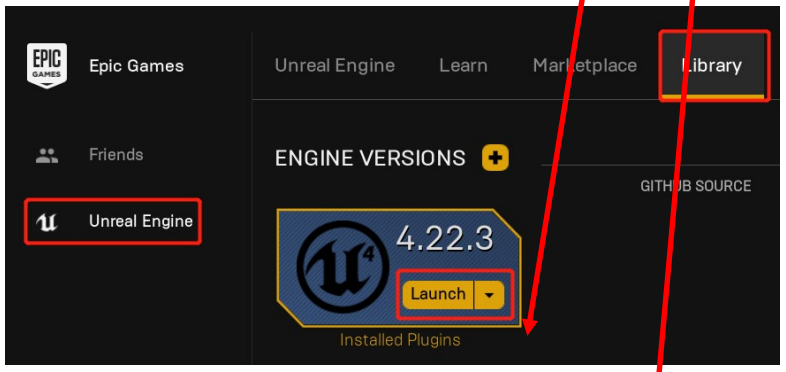
1. Setup Instruction
2. UE4 3D scene build
3. RflySim3D 3D scene import
4. Vehicle 3D model build and import
5. Control demo of objects in scene
6. Summary

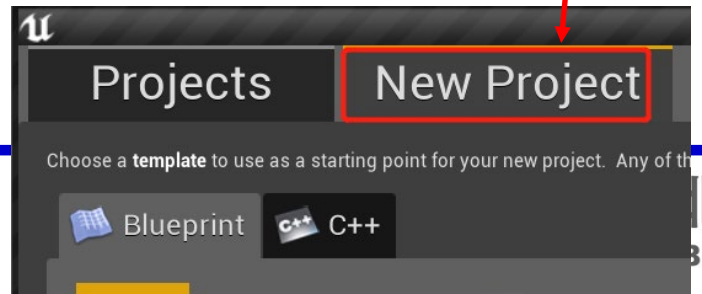
Path of source code in this section:
"RflySimAPIs\UE4MapSceneAPI\MapCreate"

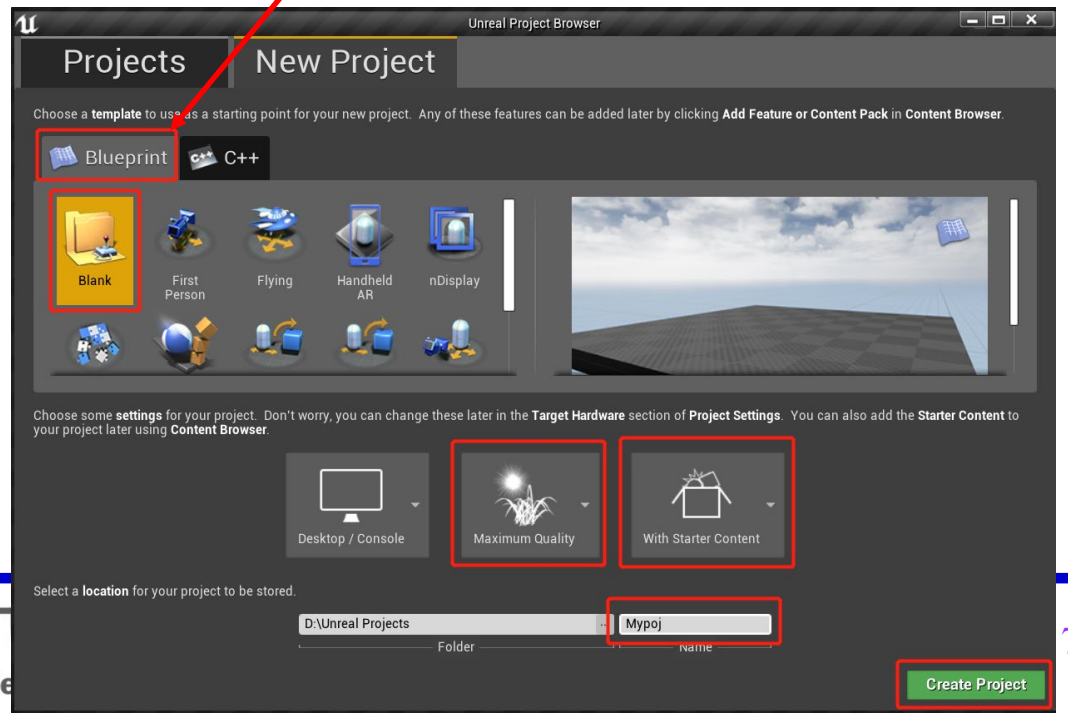
2. UE4 3D scene build

2.1 Creating a simple scene -- creating a project

- In the Epic Games, click "**Launch**" button below "**4.22.3**", and the "**Project Browser**" page shown in the lower left will pop up
- Click and enter the "**New Project**" page as shown in the lower left picture, configure it according to the lower right picture, check "**Blueprint**" -"**Blank**", "**Desktop/Console**", "**With Start Content**", "**Maximum Quality**", and name the project (For example, we use "**Mypoj**" here), click the "**Create Project**" button.



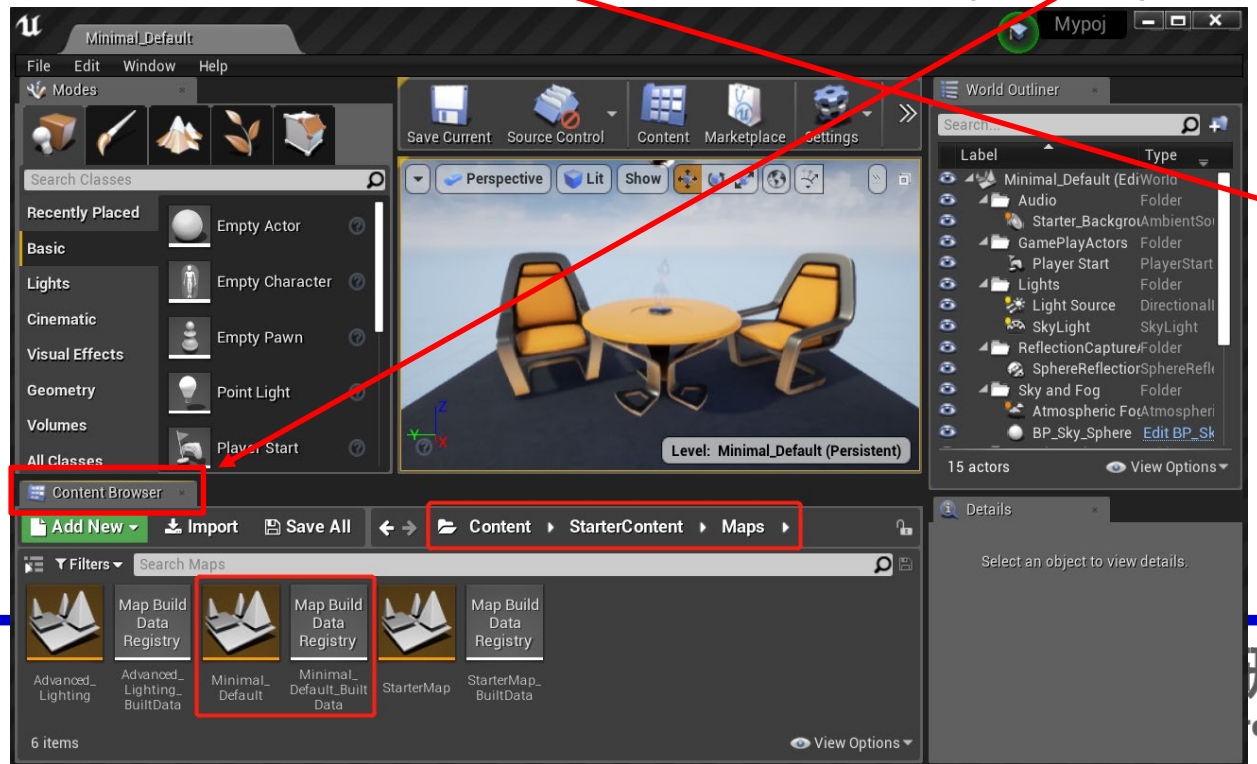


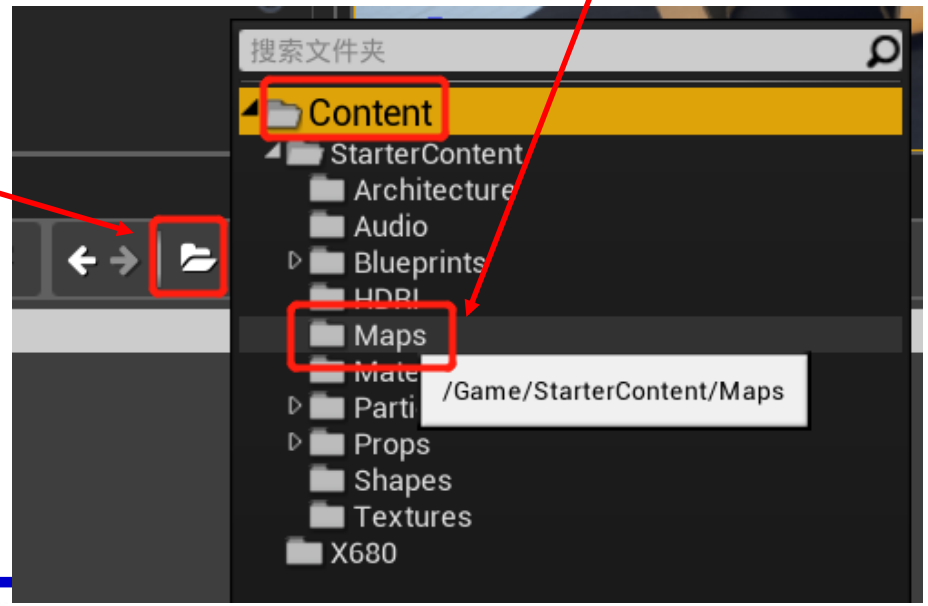


2. UE4 3D scene build

2.1 Creating a simple scene -- looking for map files

- You can get the UE4 editor as shown in the lower left picture. Here is an indoor scene for beginners.
- As shown on the right, the bottom of the page is "**Content Browser**", click the **folder icon** on the left of "**Content**" and locate the "**Content\StartContent\Maps**" directory, you can see several map (scene) files in this starters' scene .

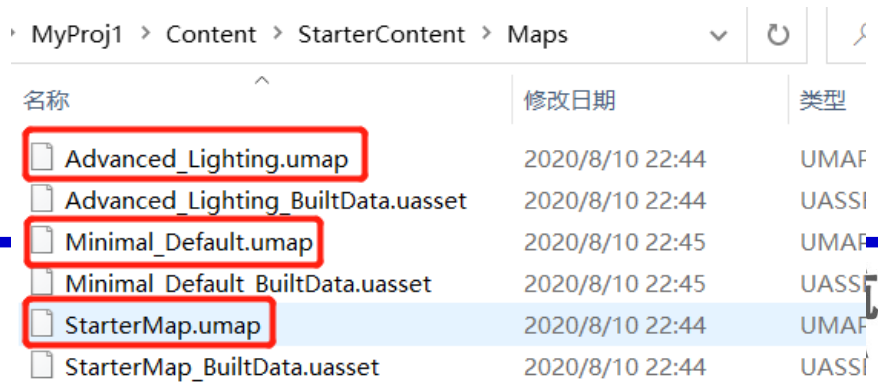


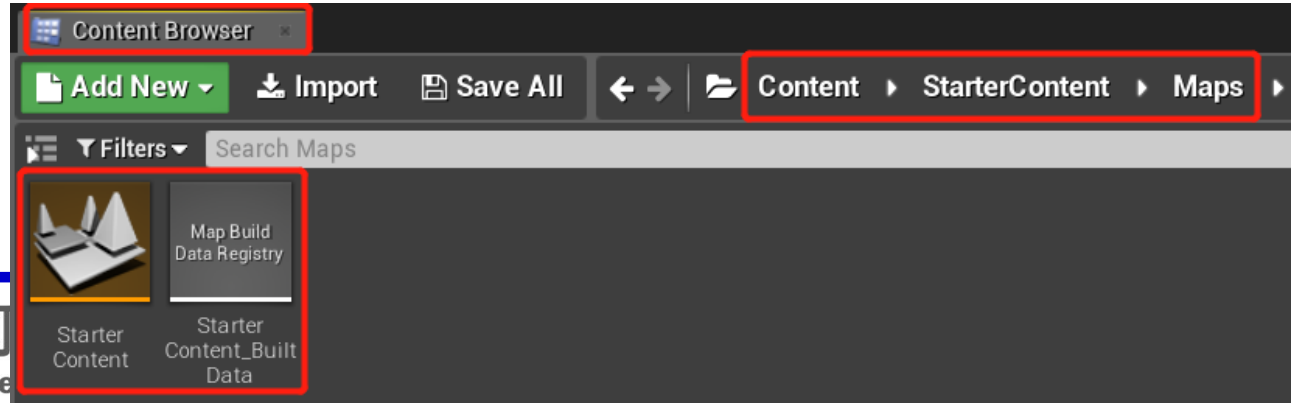


2. UE4 3D scene build

2.1 Creation of a simple scene -- delete redundant maps

- These map files actually correspond to the "***.umap**" files in the local directory "**[project folder]\Content\StarterContent\Maps**", and each **.umap** terrain map file corresponds to an independent 3D scene
- RflySim3D will automatically scan all the **.umap** files in the content directory and automatically import them, so you need to delete the redundant maps here and rename them to easy recognizable names.
- As shown in the picture on the right, rename the "**Minimal_Default**" file to "**StarterContent**" (this step is not necessary, the purpose is to make the map name is same as the folder name, which is convenient to use), and then delete "**Advanced_Lighting**", "**Advanced_Lighting_BuiltData**", "**StarterMap**" ", "**StarterMap_BuiltData**" four files.

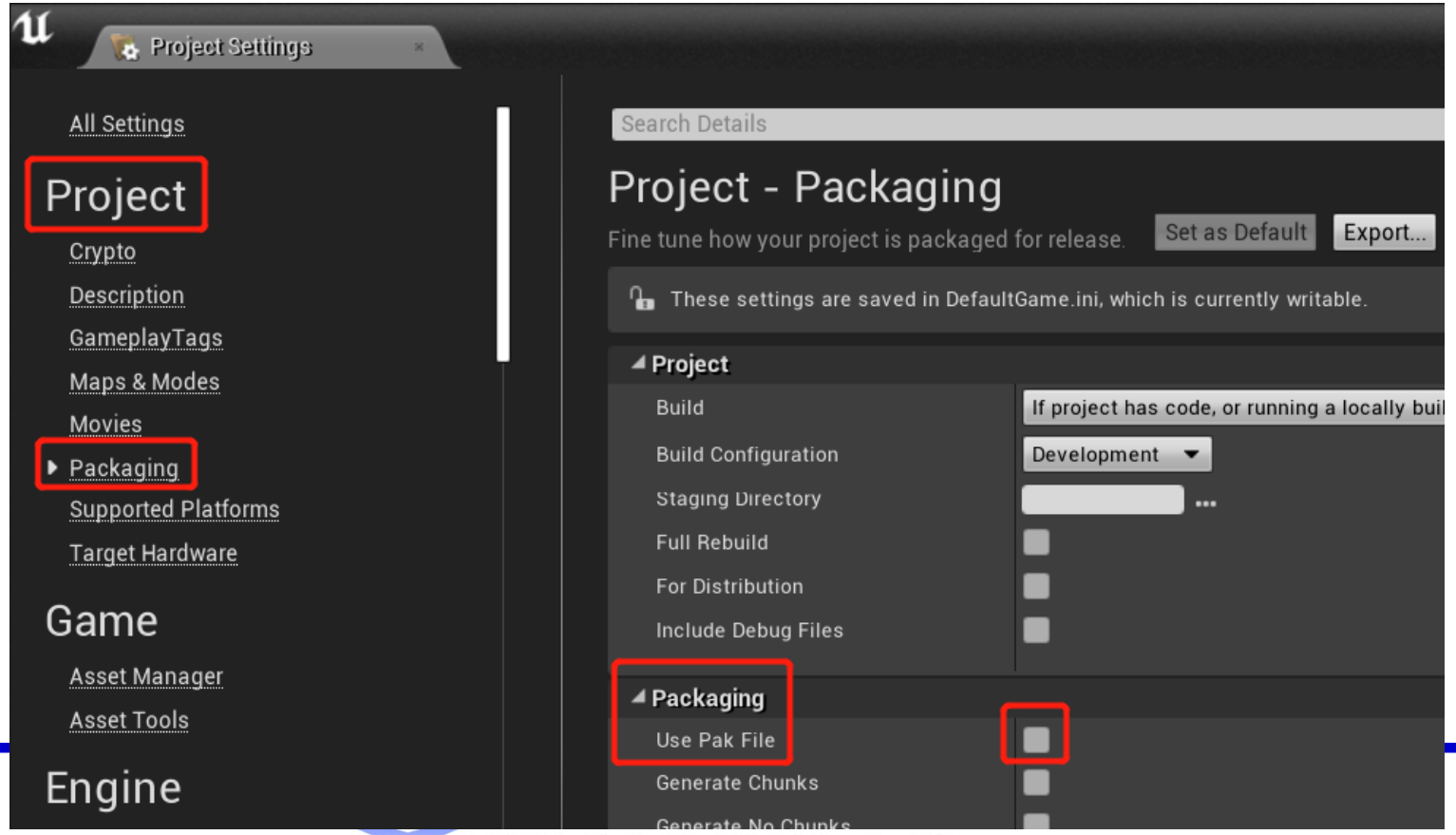




2. UE4 3D scene build

2.2 Export UE4 3D scene -- option configuration

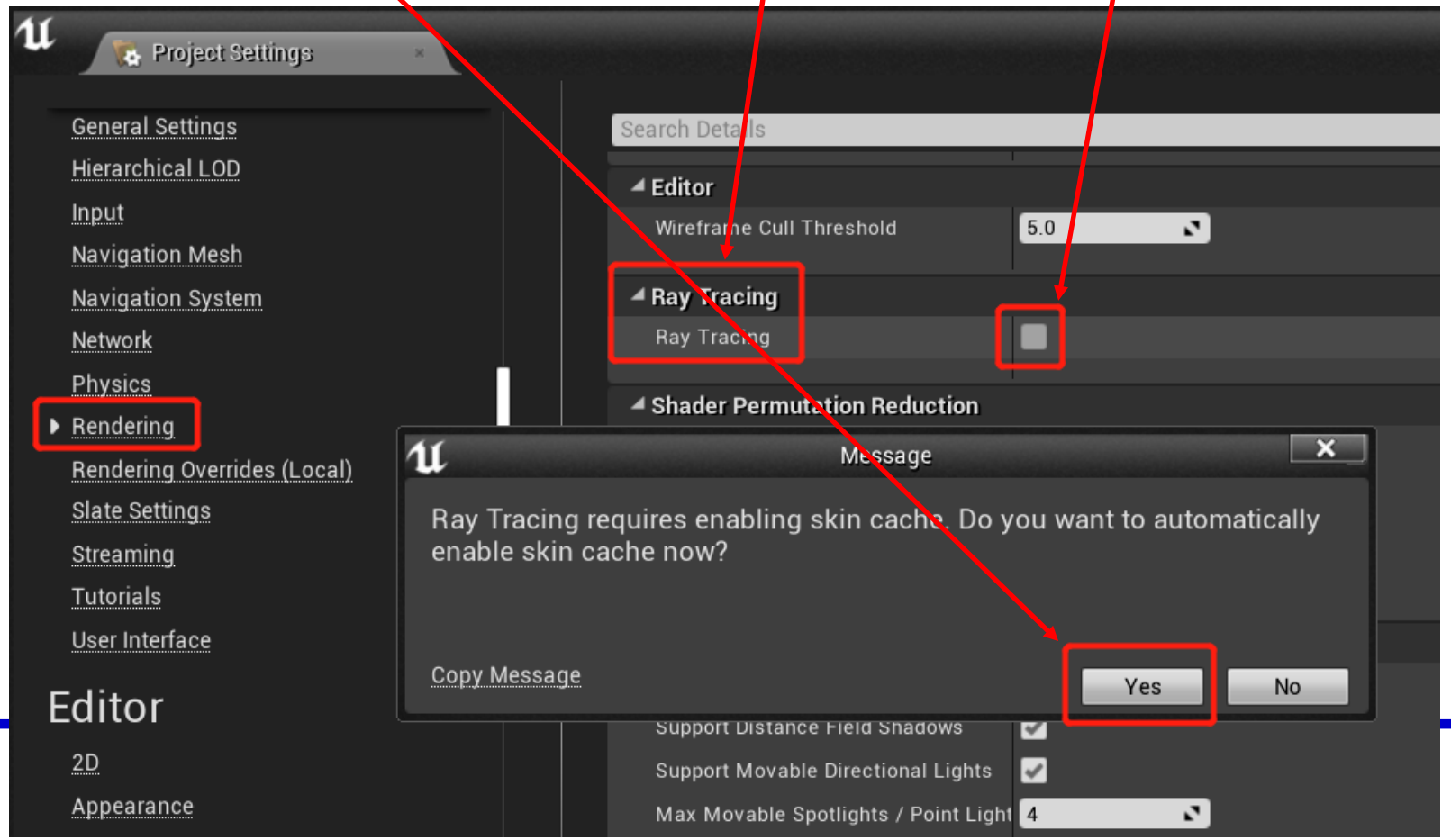
- Click the menu bar of the UE4 editor – "**Edit**" – "**Project Settings**" – "**Project**" – "**Packaging**" page, uncheck "**Use Pak File**".



2. UE4 3D scene build

2.2 Export UE4 3D scene -- option configuration

- Enter the "**Engine**"-"**Rendering**"-"**Ray Tracing**" tab, check "**Ray Tracing**" to turn on ray tracing. Click "**Yes**" in the pop up menu, and then select restart UE4.

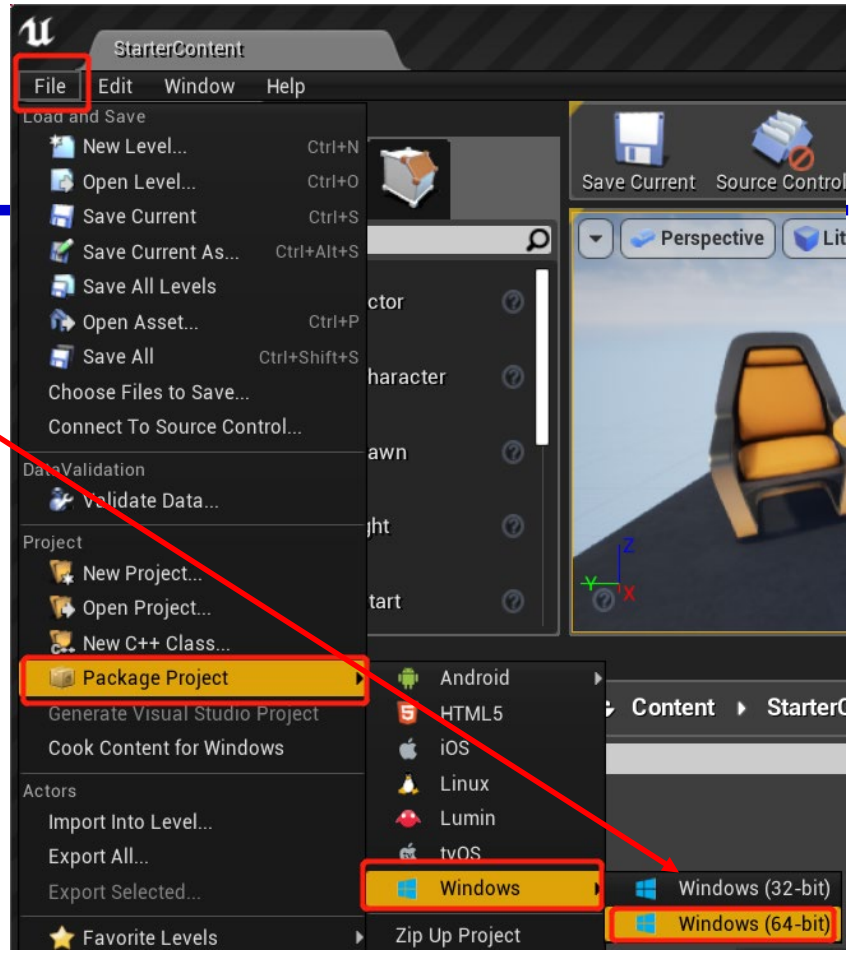


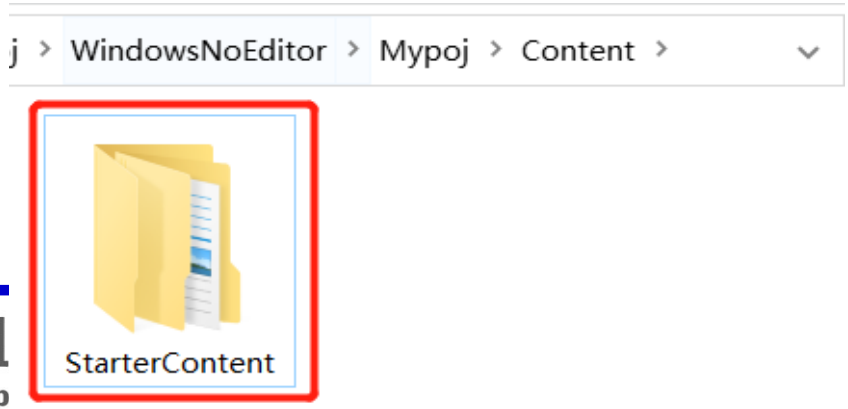
2. UE4 3D scene build

2.2 Export UE4 3D scene -- compile and package

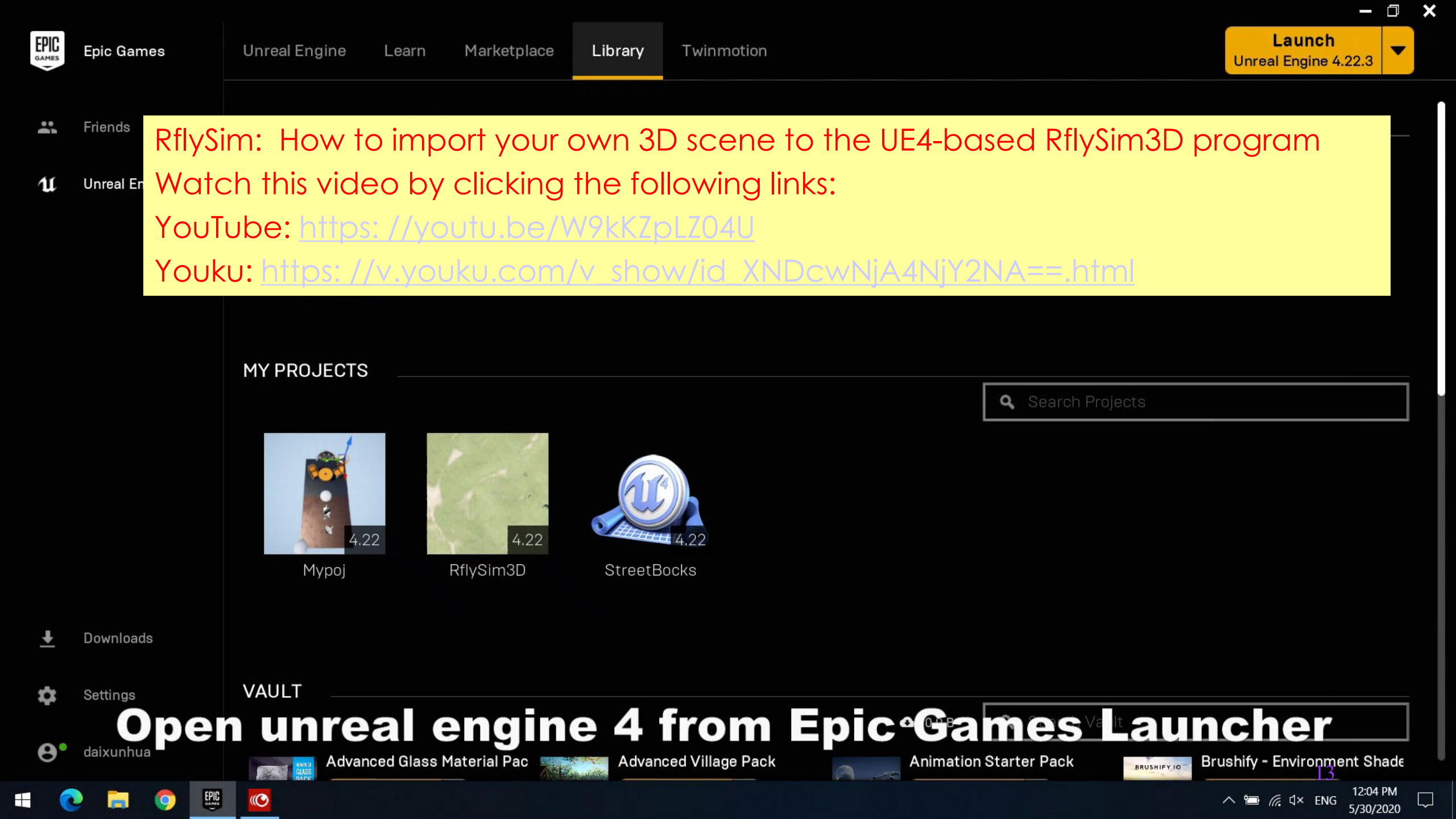
- Click on the menu bar-"**File**"-"**Package Project**"-"**Windows**"-"**Windows (64-bit)**", wait patiently for the packaging to be completed.
- Wait for the packaging to be completed, you can see the "**WindowsNoEditor**" folder in the Windows project folder "**Mypoj**" as shown in the lower right picture, and click into the "**Mypoj\Content**" folder as shown below

(**Note**: **Mypoj** here is my project name, which corresponds to your own project name for specific operations), copy the "**StarterContent**" folder for use. This folder contains a "**StarterContent\Maps\StarterContent.umap**" map file, and all required scene files, which can be directly copied to our RflySim3D program for use.





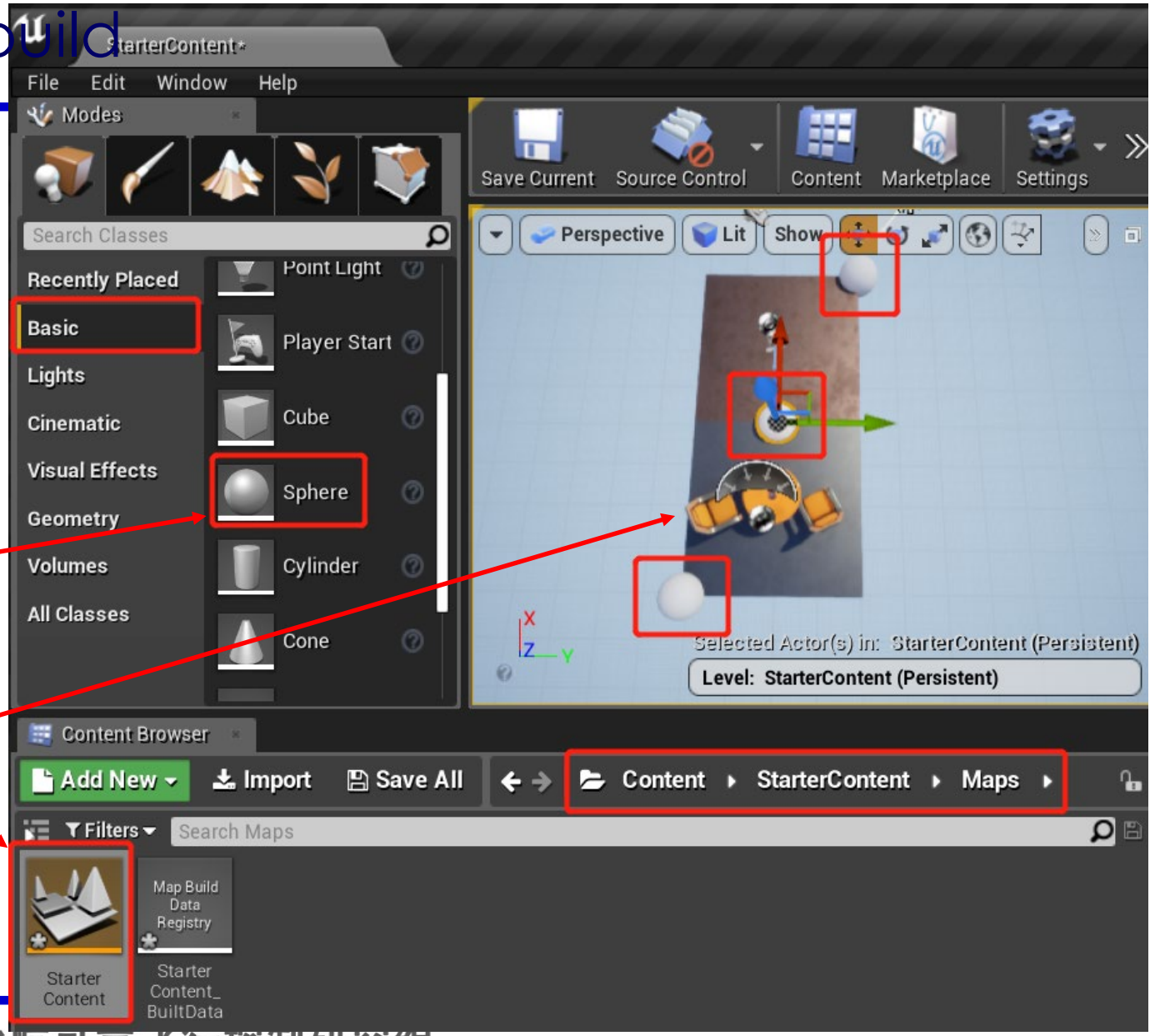
Note: Do not change the names of the obtained packed files and folders. You can only rename contents in the UE4 editor.


Epic Games
Unreal Engine
Learn
Marketplace
Library
Twinmotion
Launch
Unreal Engine 4.22.3
Friends
Unreal En
RflySim: How to import your own 3D scene to the UE4-based RflySim3D program
Watch this video by clicking the following links:
YouTube: https: //youtu.be/W9kKZpLZ04U
Youku: https: //v.youku.com/v_show/id_XNDcwNjA4NjY2NA==.html
MY PROJECTS
Search Projects
4.22
Mypoj
4.22
RflySim3D
4.22
StreetBocks
Downloads
Settings
daixunhua
VAULT
Open unreal engine 4 from Epic Games Launcher
Advanced Glass Material Pac
Advanced Village Pack
Animation Starter Pack
Brushify - Environment Shade
ENG
12:04 PM
5/30/2020

2. UE4 3D scene build

2.3 Generate terrain files needed by CopterSim

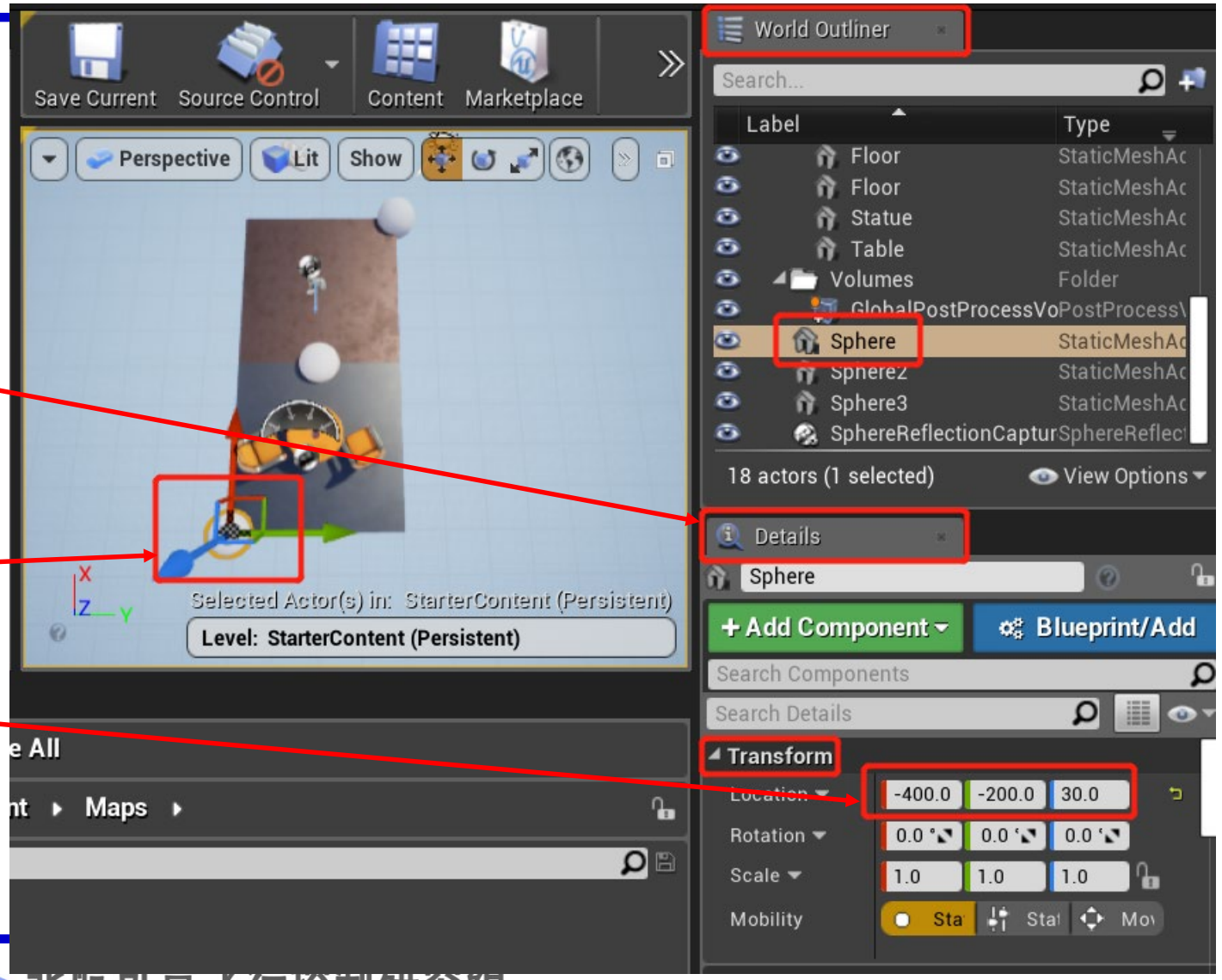
- Open the Mypoj project again, in the content browser on the lower side, double-click to open the "**StarterContent**" map file, and then drag three "**Sphere**" balls into the scene, and place them in the lower left, upper right corners and the middle of the map.



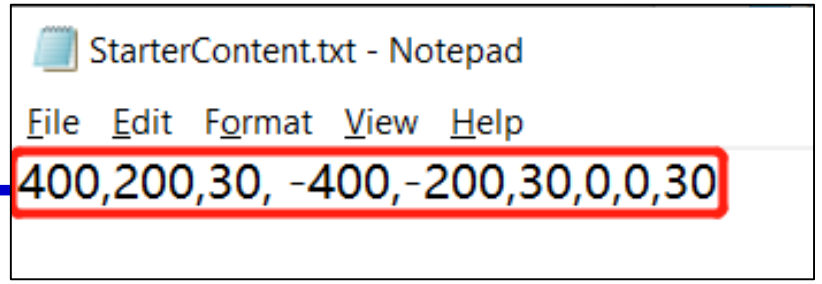
2. UE4 3D scene build

2.3 Generate terrain files needed by CopterSim

- Click on three balls in turn, under the **Transform** tab in the "**Details**" list on the lower right side of the UE4 interface, you can see the coordinate information of the ball. For example, in the figure below, click the small ball in the lower left corner, and read the coordinates as **-400,-200,30**
- Note: after getting the point coordinates, please delete these three balls from the scene



2. UE4 3D scene build



2.3 Generate terrain files needed by CopterSim

- Get the three-dimensional coordinates of the upper right ball (xy coordinates are all positive), the lower left ball (xy all positive) and the middle ball (xy all zero) in sequence, and write them in one line. Create a new **.txt** file and name it "**StarterContent.txt**", and write the three ball coordinates as follows.

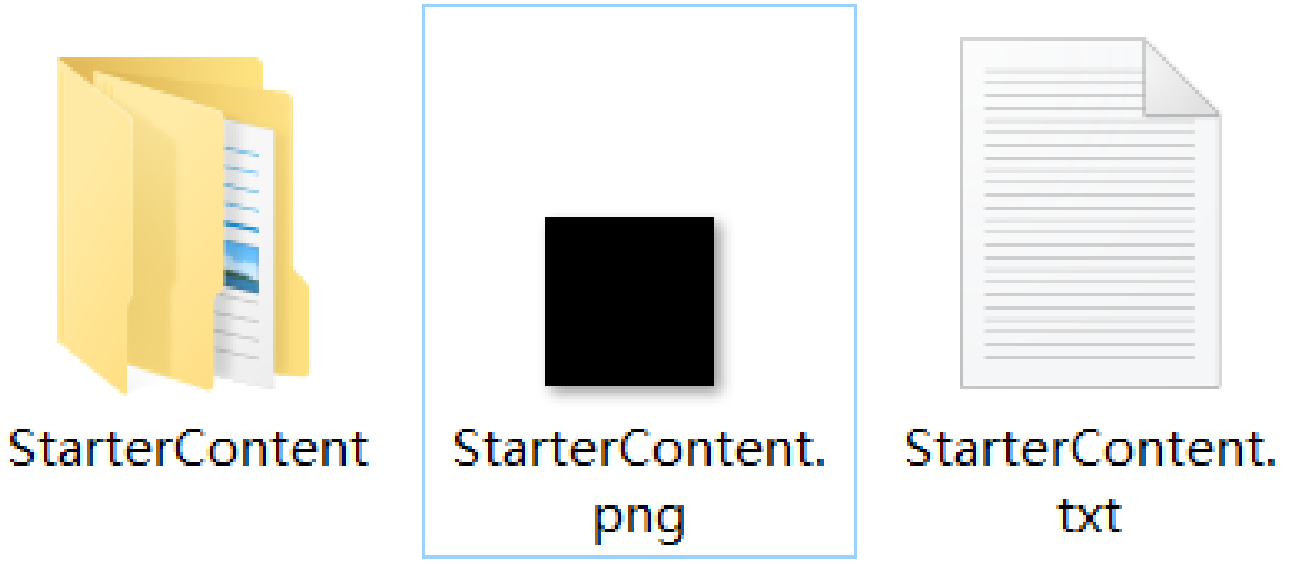
400,200,30, -400,-200,30,0,0,30

- The above txt file is a terrain calibration file, we also need a terrain grid file stored in **.png** format. Since this is a flat terrain, the grid file corresponds to a constant matrix (each pixel has the same value), and here is the file "**RflySimAPIs\UE4MapSceneAPI\MapCreate\FlatTerrainDemo.png**" (corresponding to a matrix file with all zero values) renamed to "**StarterContent.png**" is alright. For complex terrain, the png terrain file acquisition method can be found in **Section 2.5**.

2. UE4 3D scene build

2.3 Generate terrain files needed by CopterSim

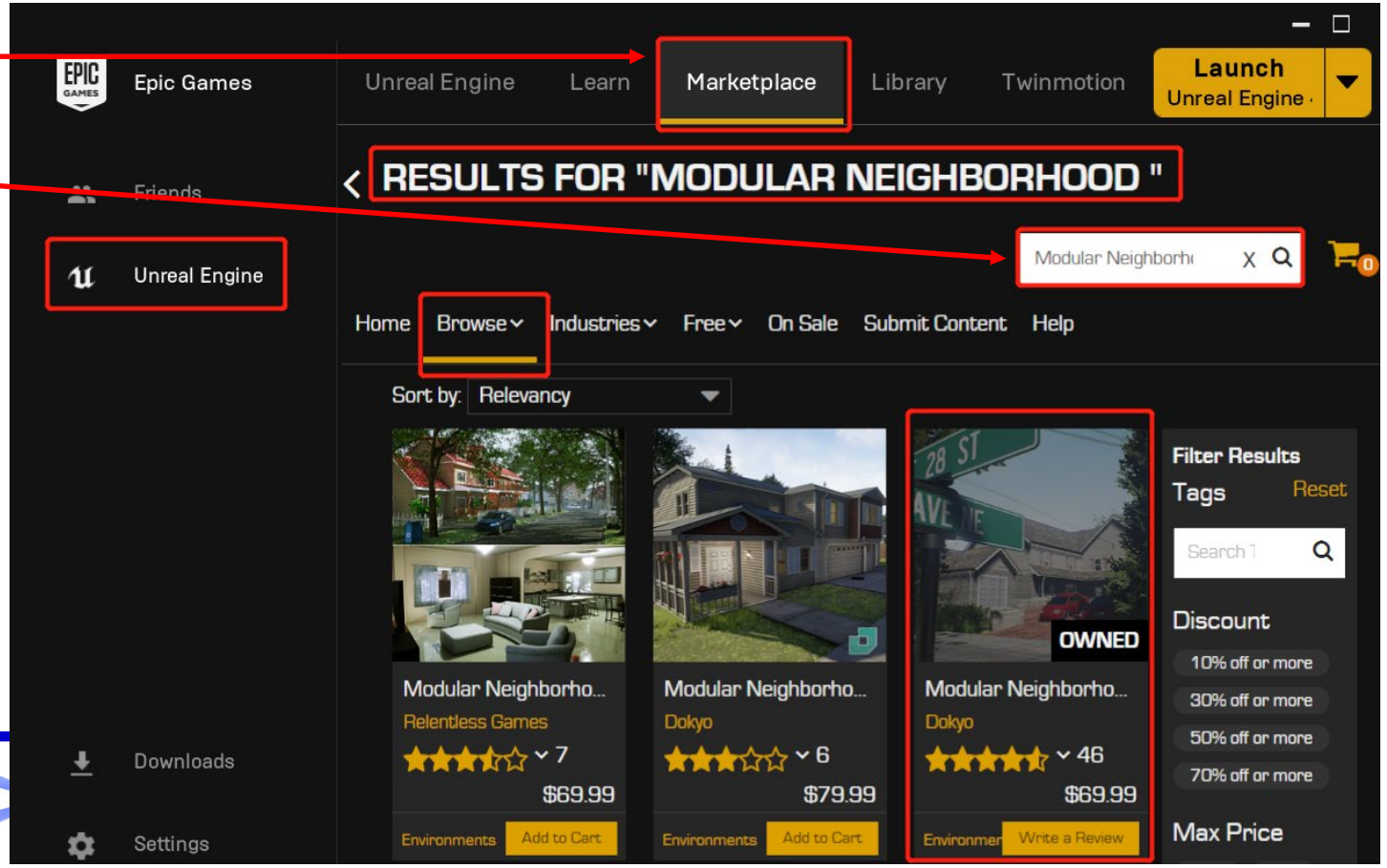
- After the above steps, we get a "**StarterContent**" folder, and two files "**StarterContent.png**" and "**StarterContent.txt**". For the application method of these three files, please refer to the content in the following **Section 3**.



2. UE4 3D scene build

2.4 Import ready-made scenes in Epic Marketplace

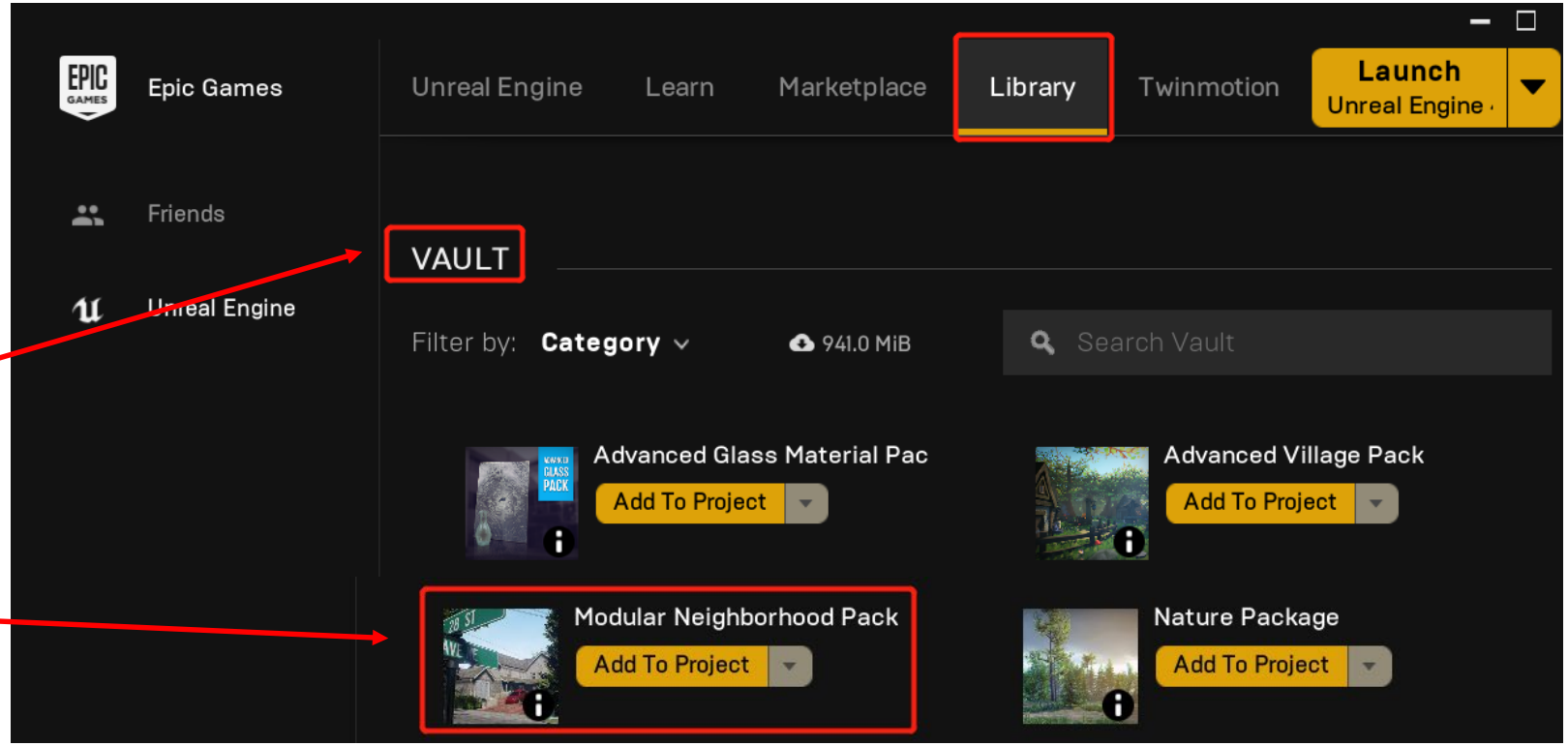
- There are a lot of charged and free realistic 3D scenes in Epic Marketplace, which can be used directly. The following uses the "Modular Neighborhood Pack" scene as an example to introduce how to import these scenes.
- Click on the "**Marketplace**" page of the Epic Games, and enter "**Modular Neighborhood** " in the search box to get the scene result page shown on the right. Add "**Modular Neighborhood Pack**" to the shopping cart and purchase.
- **Note**: You can also find many free scenes in the mall to reproduce the next steps



2. UE4 3D scene build

2.4 Import ready-made scenes in Epic Marketplace

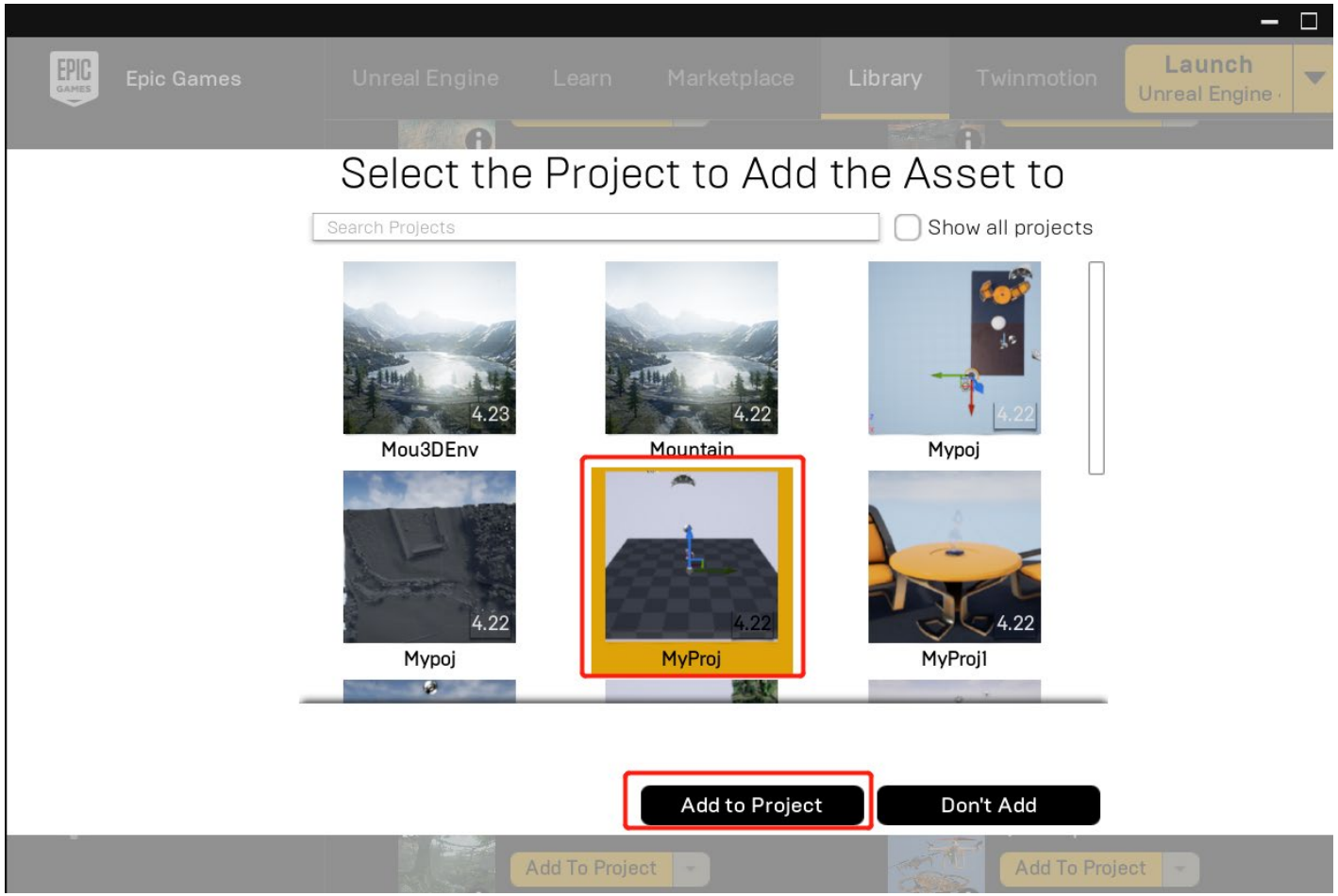
- After the scene purchase is completed, return to the Epic Games main interface again, as shown on the right, find the purchase scene in "**Library**"-"**VAULT**" (here is Modular Neighborhood Pack), and then click the "**Add to Project**" button
- Same import method for other scenes



2. UE4 3D scene build

2.4 Import ready-made scenes in Epic Marketplace

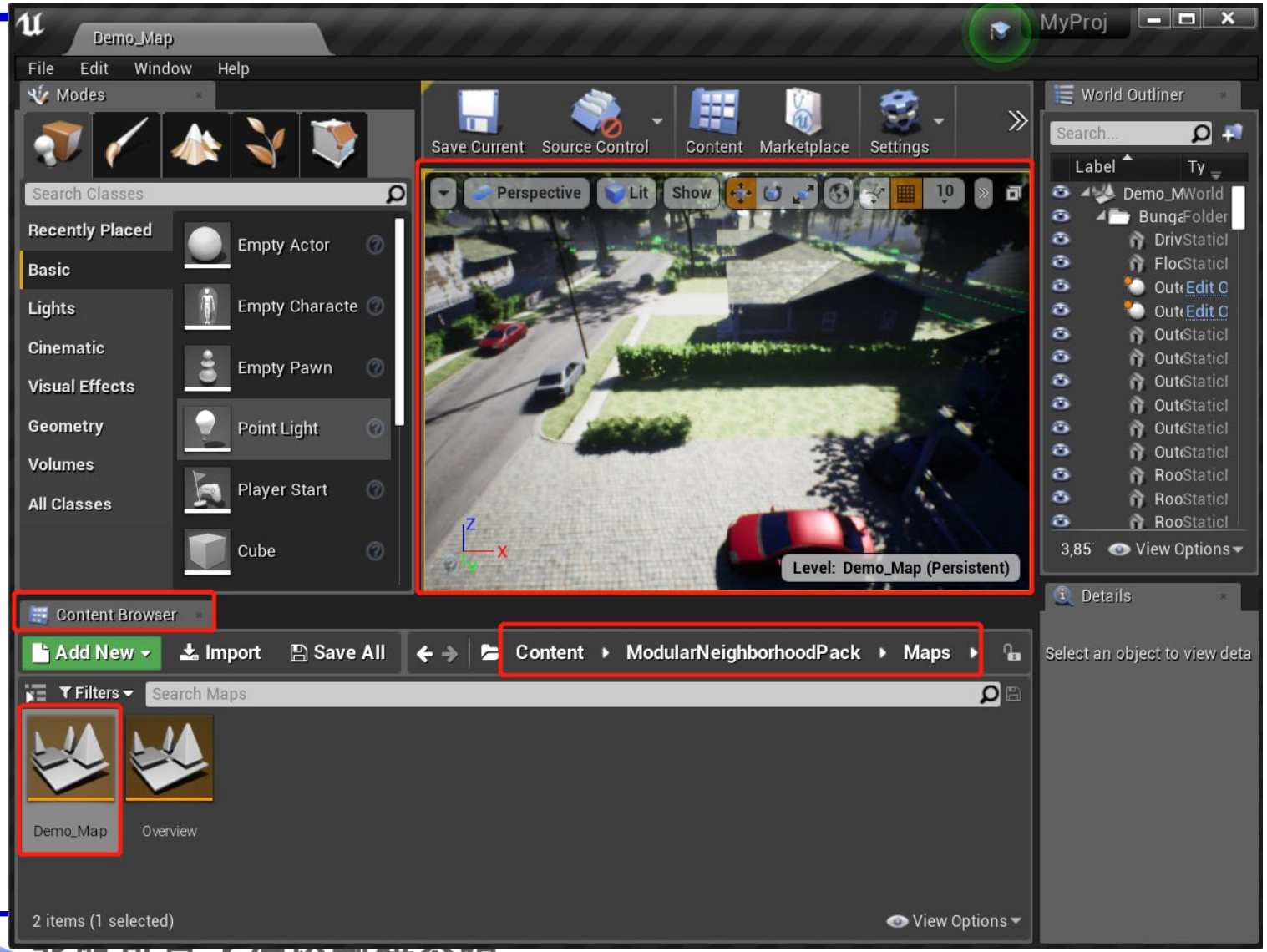
On the library page, the **VAULT** tab page displays the scene components purchased from the Epic Marketplace. Click "**Add to Project**", and then select the name of the project you want to add to as shown below (here, select the new MyProj project in **Section 2.1**), And click the "**Add to Project**" button, and then wait patiently for the download and deployment to be completed.



2. UE4 3D scene build

2.4 Use existing complex scenes

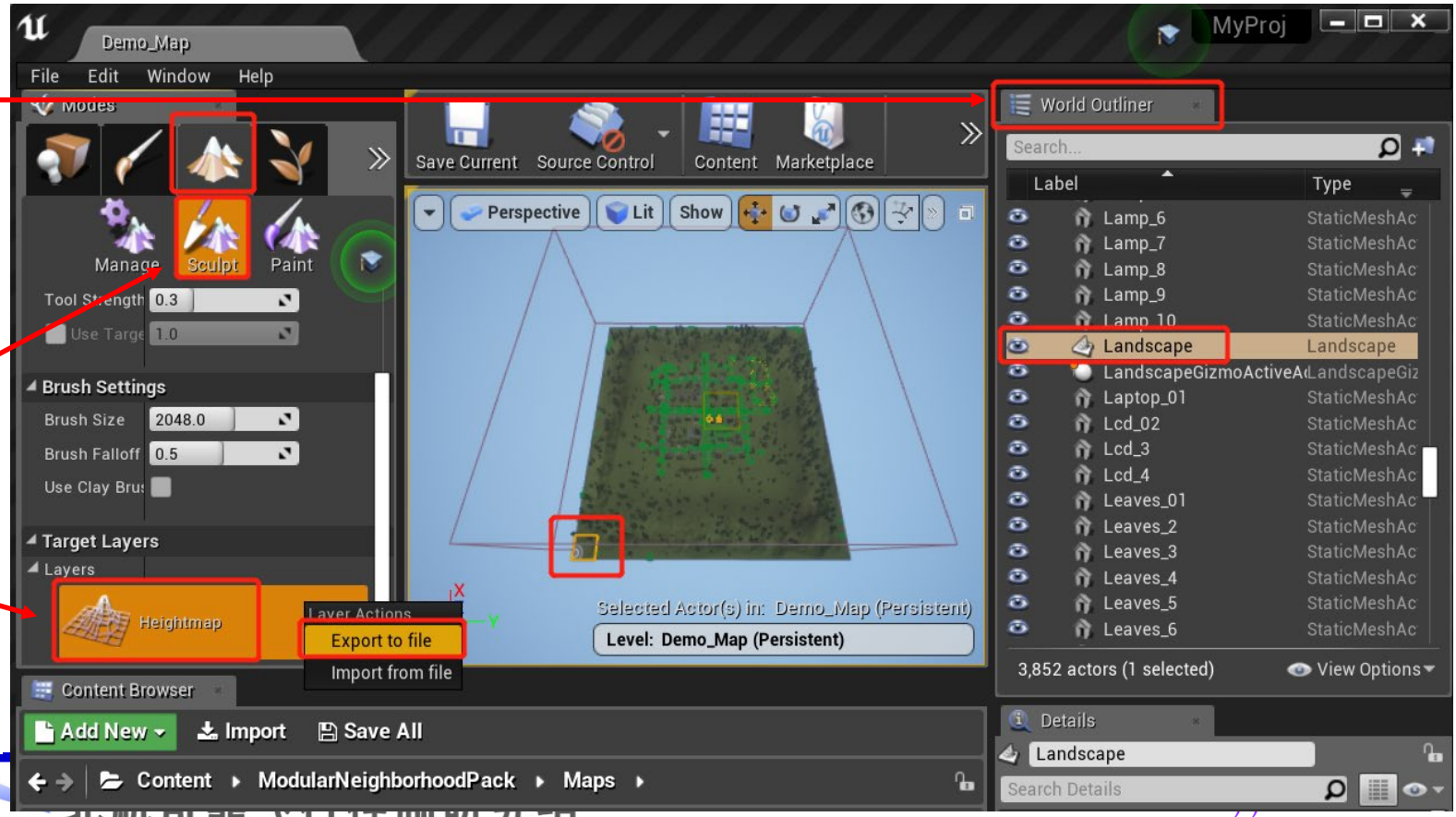
- After add in the folder, open the MyProj project again, and you can see the imported resource folder "**ModularNeighborhoodPack**" in the Content directory as shown on the right.
- Double-click the "**Maps\Demo_Map**" file to open a 3D map scene of a block. Then follow the steps in **Section 2.1**, rename (NeighborhoodPark in RflySim3D) and delete redundant scenes.
- The principle of importing other scenes is similar.



2. UE4 3D scene build

2.5 Export of terrain files for complex scenes

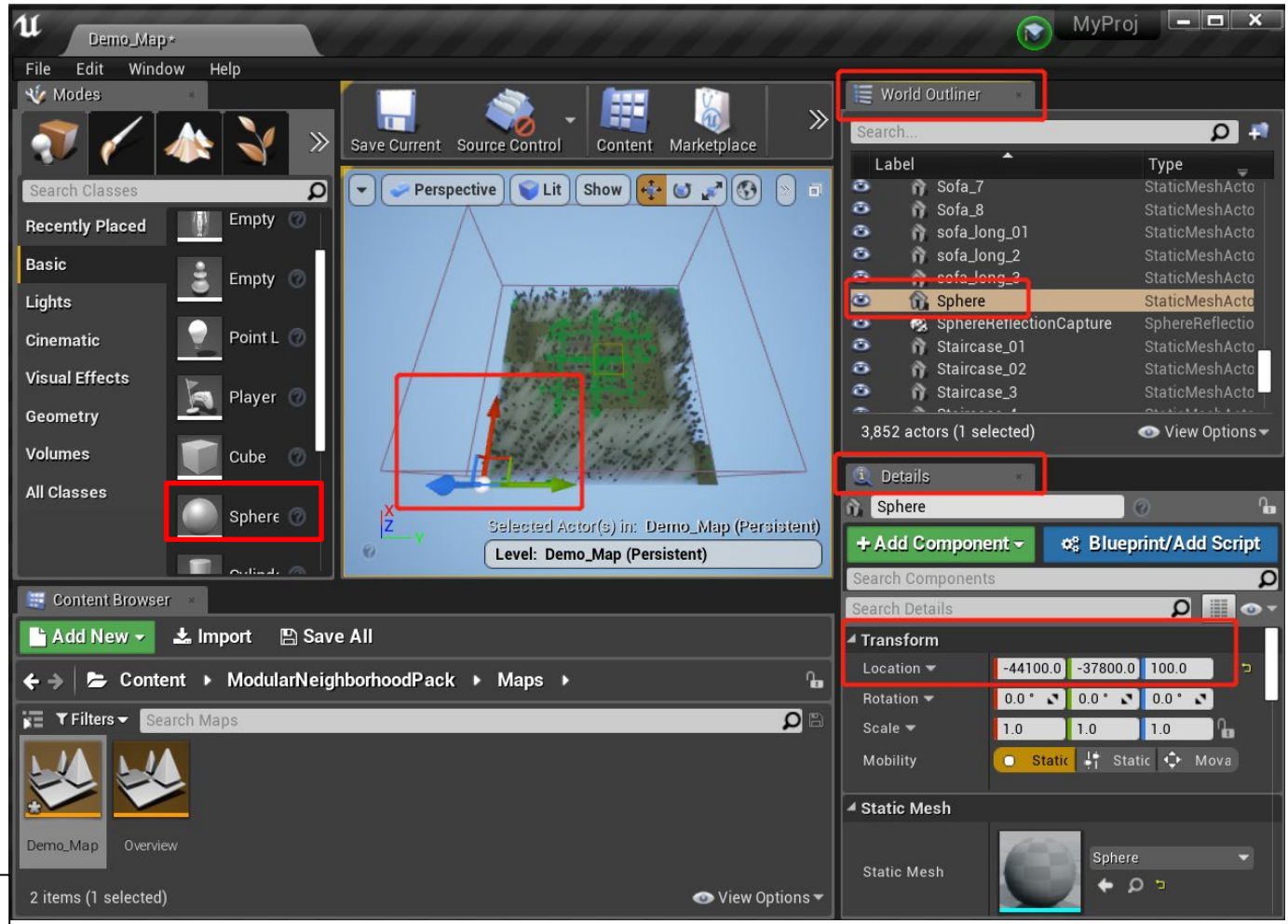
- Here is the "**Modular Neighborhood Pack**" scene as an example (corresponding to the **NeighborhoodPark** scene in RflySim3D)
- Open the **MyProj** project, in the "**World Outliner**" window on the right side of the UE4 editor, find and click to select the "**Landscape**" terrain object.
- On the left, "**Modes**"-"**Landscape**"-"**Sculpt**"-"**Layers**"-"**Heightmap**", right-click and click "**Export to File**" to save it as a ***.png** file (this example takes **NeighborhoodPark.png**) terrain grid file.

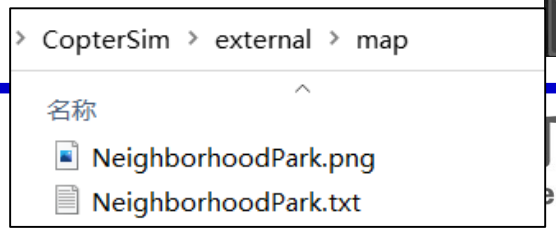


2. UE4 3D scene build

2.5 Export of terrain files for complex scenes

- Place three Sphere balls on the **upper right, lower left, and origin** (or the highest point) of the terrain surface (the center of the sphere just sticks to the terrain surface, delete the ball after getting the point coordinates) and record three sets of coordinates and write them into a **txt** file.
- **Note**: The xy coordinates of the **lower left** ball should all be **all negative**, please make sure that the direction is correct.
- See the "**CopterSim\external\map**" folder for examples





Content

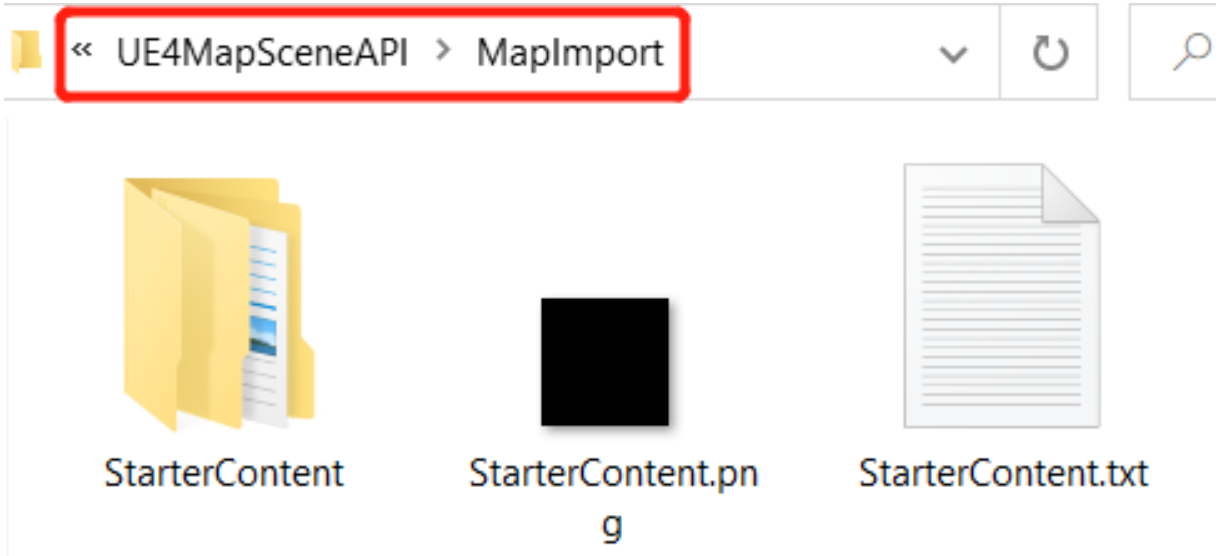
1. Setup Instruction
2. UE4 3D scene build
3. RflySim3D 3D scene import
4. Vehicle 3D model build and import
5. Control demo of objects in scene
6. Summary

Path of source code in this section:
"RflySimAPIs\UE4MapSceneAPI\MapImport"

3. RflySim3D 3D scene import

3.1 3D scene file preparation

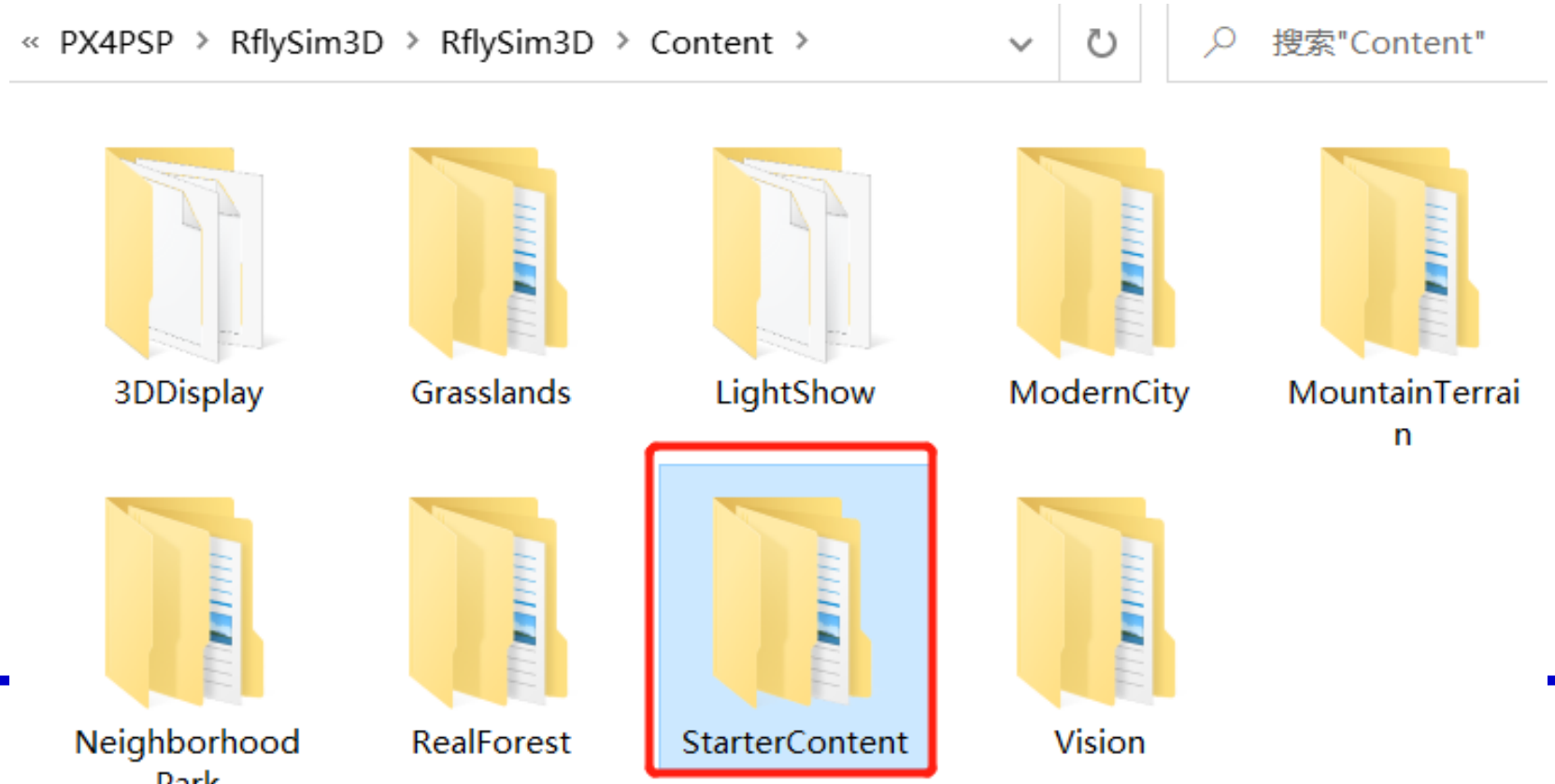
- According to **Section 2.3**, the 3D scene required by RflySim3D contains three file: **scene folder**, ****.png** terrain file, and ****.txt** terrain calibration file. **Note**: The map file name in **.umap** format in the scene folder must be exactly the same as the terrain file name in **.png** and **.txt**.
- Here is a designed example "**RflySimAPIs\UE4MapSceneAPI\MapImport**"



3. RflySim3D 3D scene import

3.2 RflySim3D scene file import

- The importing steps are simple. Completely copy the folder "**UE4MapSceneAPI\MapImport\StarterContent**" to the "**RflySim3D\RflySim3D\Content**" folder under the installation path (**C: \PX4PSP** by default) to complete all installation tasks



3. RflySim3D 3D scene import

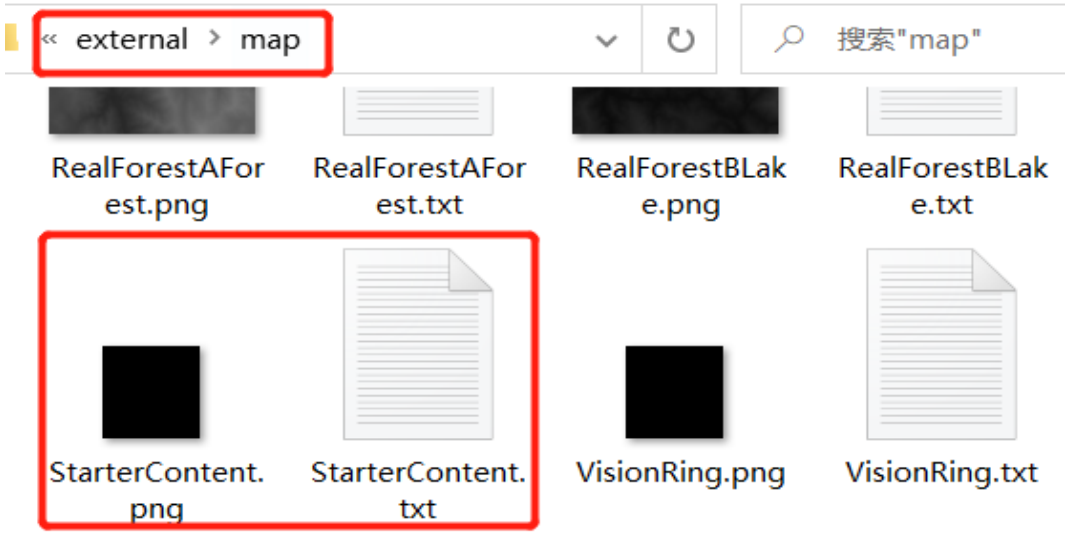
3.3 RflySim3D scene view

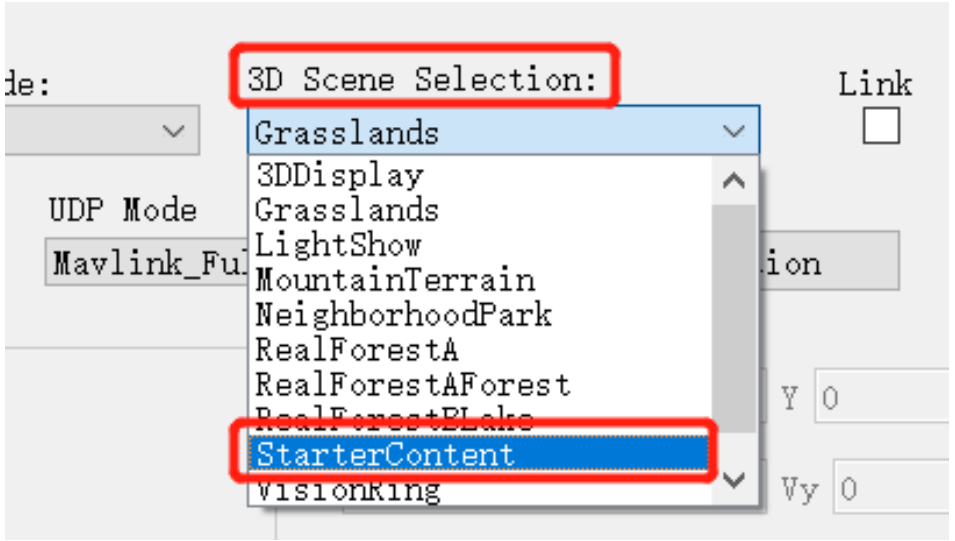
- Open the **RflySim3D.exe** program and press the "**M**" key on your keyboard, you can view the imported scenes as shown in the figure below (press the **M** key several times, or press the combination key "**M**" + **number**) and complete the scene switching.

3. RflySim3D 3D scene import

3.3 Import of terrain files in CopterSim

- Copy the two files "**StarterContent.png**" and "**StarterContent.txt**" under the path "**RflySimAPIs\UE4MapSceneAPI\MapImport**" to the "**CopterSim\external\map**" folder so that **CopterSim** can correctly identify the new terrain.
- After the copy is completed, open the **CopterSim** program, and you can see the new added "**StarterContent**" 3D scene in the "**3D Scene Selection**" tab.

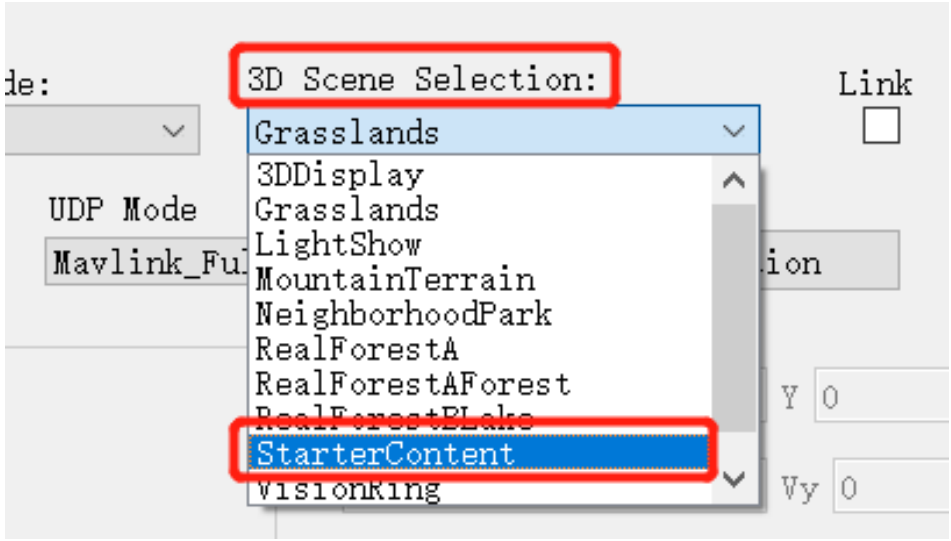




3. RflySim3D 3D scene import

3.4 Switching scene files

- Open **CopterSim** and **RflySim3D** software at the same time, you can see a quadcopter appear in the scene
- Directly select a specific scene in the "**3D Scene Selection**" tab of **CopterSim**, **RflySim3D** will automatically switch to the set scene and create a quadcopter.

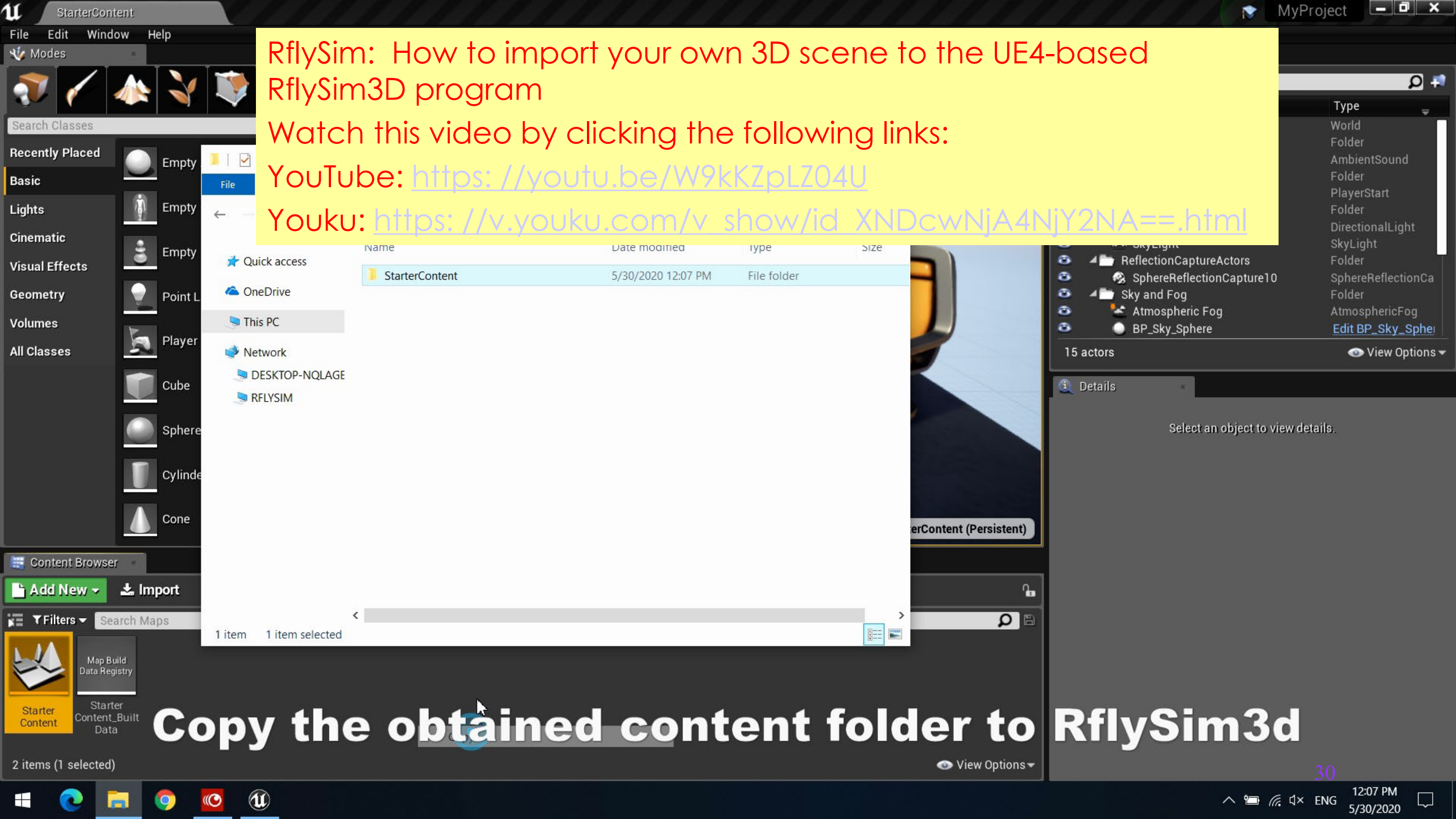


RflySim: How to import your own 3D scene to the UE4-based RflySim3D program

Watch this video by clicking the following links:

YouTube: https://youtu.be/W9kKZpLZ04U

Youku: https://v.youku.com/v_show/id_XNDcwNjA4NjY2NA==.html



Content

1. Setup Instruction
2. UE4 3D scene build
3. RflySim3D 3D scene import
4. Vehicle 3D model build and import
5. Control demo of objects in scene
6. Summary

Path of source code in this section:
"RflySimAPIs\UE4MapSceneAPI\VehicleModel"

4. Vehicle 3D model build and import

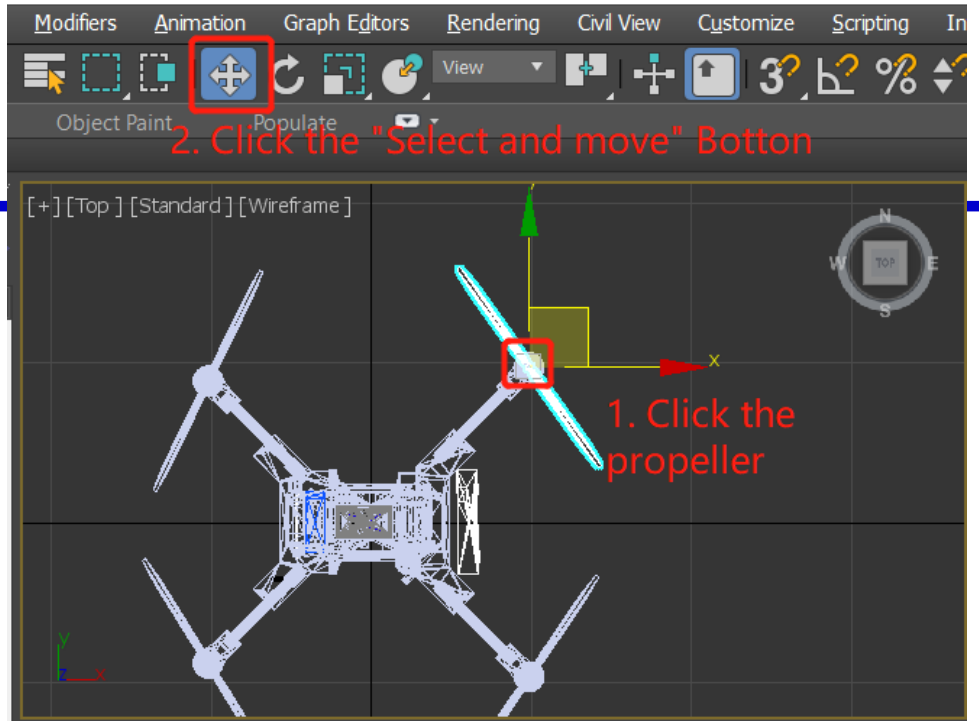
4.1 Obtaining 3D Model File of Vehicle

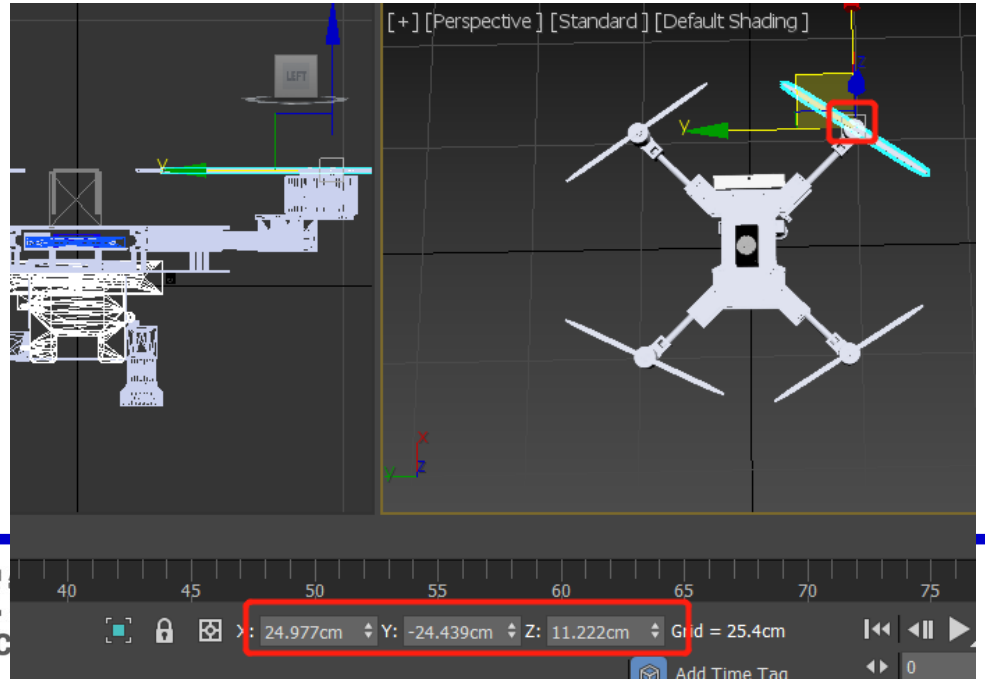
- First, you need to install 3Ds Max, here we are using 3Ds Max 2020. Open 3Ds Max in the start menu, and click "**Customize**"-"**Unit Settings**" in the menu bar to change the unit to cm.
- Open a 3D vehicle model file in .max format with 3Ds Max, or import a 3D model from software such as CAD to get a .max model file. Here you can go to "**RflySimAPIs\UE4MapScene API\VehicleModel**" to open our given demo "**DroneyeeX680.max**".
- Select all the components in 3Ds max, click the menu bar "**Group**"-"**Group**", you can make all the components into a whole, then you can adjust the position, attitude, axis position, axis direction of the vehicle. The following requirements need to be met:
  1. Ensure that the nose direction points to the positive x-axis direction of **3Ds Max**, and the top of the vehicle points to the positive z-axis direction (upward);
  2. Ensure that the center of mass of the vehicle is in the center of **3Ds Max** coordinates;
  3. Ensure that there is no invisible in the model components, delete it if necessary.
- Select the overall group of the vehicle in the previous step and click the menu bar-"**Group**"-"**Ungroup**" to unlock the combination. Then, combine the components of the body (except for the four propellers) into a group, and each propeller should form a single group. This leaves five groups, the body and four propellers.

4. Vehicle 3D model build

4.2 Get the coordinates of a component

- As shown on the right, click the "**Select and Move**" button on the toolbar, and then click to select the body or propeller object, you can read the three-dimensional coordinates of the object in the status bar below (input all coordinates as 0 to move the object to the center).
- Click on the body to see the three-dimensional coordinates of the body (x, y, z)=[-0.449,- 0.363,0]. The upper right propeller [24.977, -24.439, 11.222], the lower left propeller [-25.99, 22.528, 11.022], the upper left propeller [24.977, 24.693, 11.022], the lower right propeller [-25.99, -22.274, 11.022].

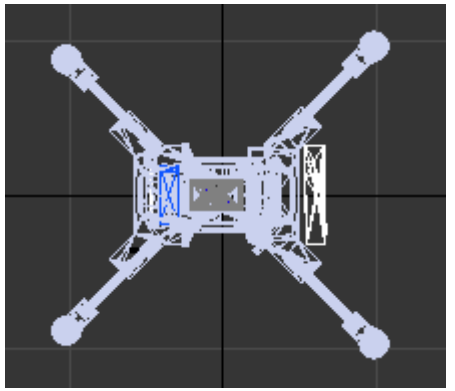


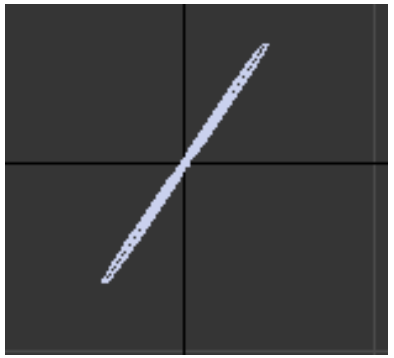


4. Vehicle 3D model build and import

4.3 Obtain the coordinates of the propeller relative to the body

- By subtracting the coordinates of each propeller from the coordinates of the body, the three-dimensional coordinates of each propeller relative to the center of the body can be obtained. Since 3Ds Max adopts the **right-handed coordinate system** of the front upper left and UE4 adopts the **left-handed coordinate system** of the front upper right, the y-axis needs to be reversed.
- Then we got the coordinates sequence of **right upper, left lower, right lower, left upper**: [25.4260,24.0760,11.2220], [-25.5410,-22.8910,11.0220], [25.4260, -25.0560,11.0220], [-25.5410,21.9110,11.0220], these few coordinates are kept for spare. Then, record the distance from mass center of the vehicle to ground(lowest position), about **8cm**.
- Copy **DroneyeeX680.max** twice and name them separately as **DroneyeeX680Body.max** and **DroneyeeX680Prop.max**. Remove four propellers in **DroneyeeX680Body.max** and move the body to 3Ds Max coordinate center. Remove the body component and three propellers in **DroneyeeX680Prop.max**, left one and move to 3Ds Mac coordinate center.

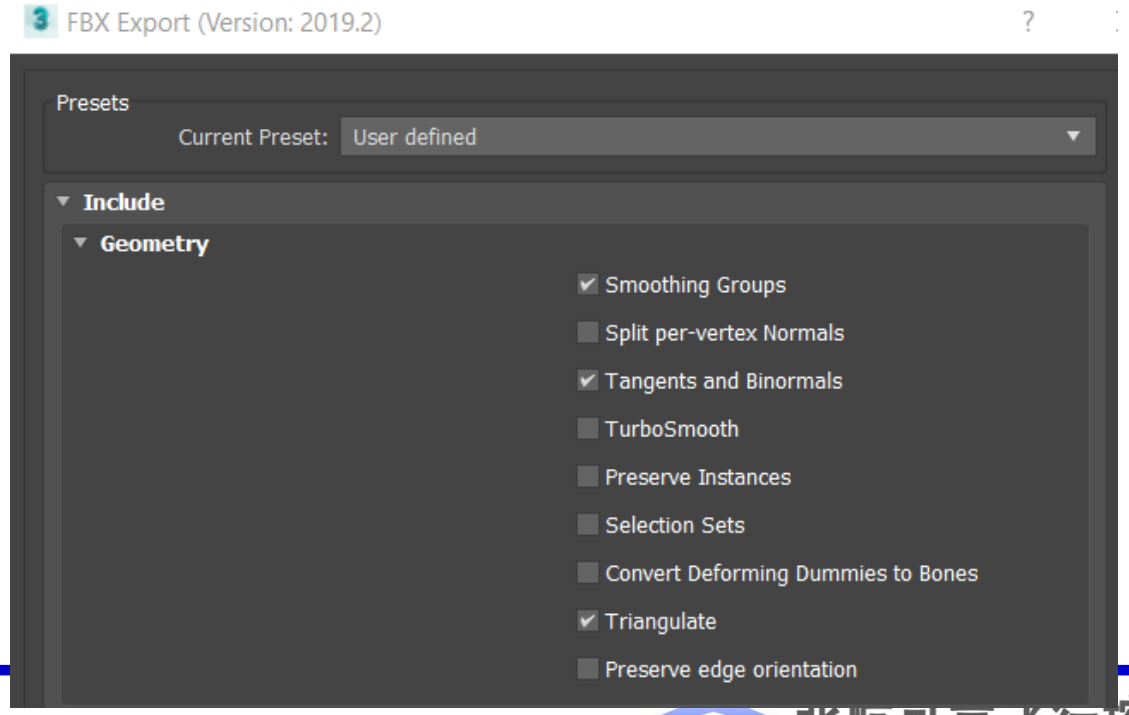


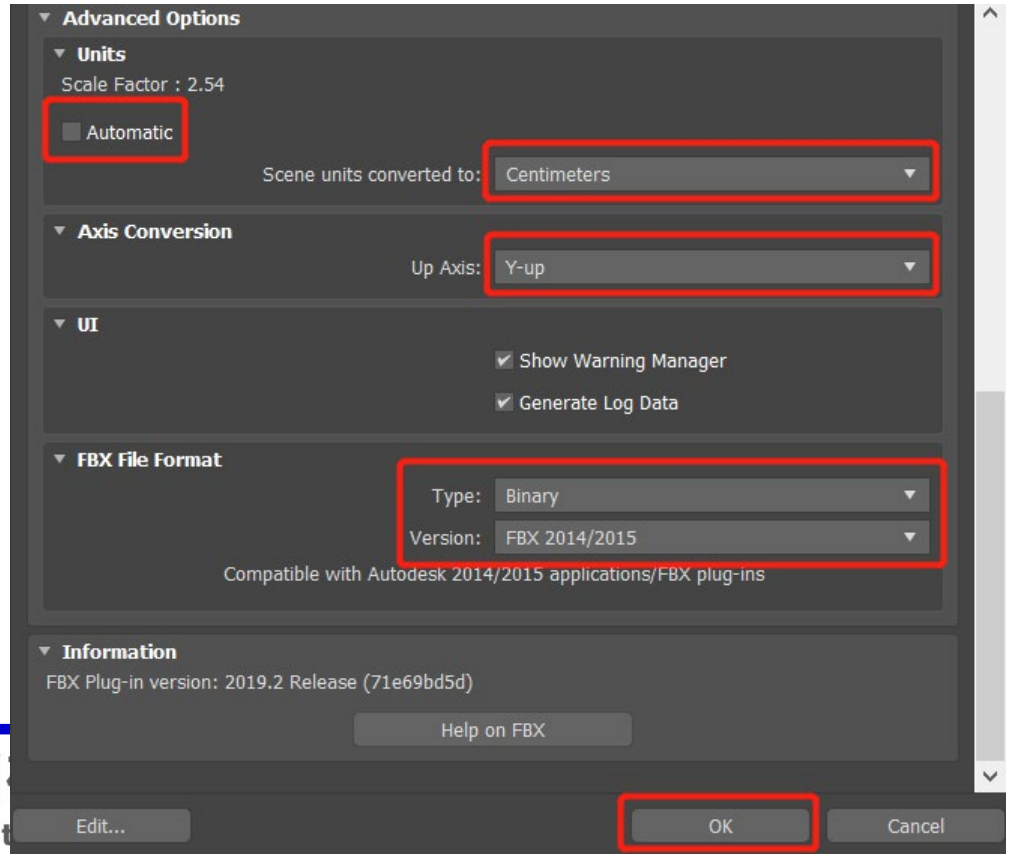


4. Vehicle 3D model build and import

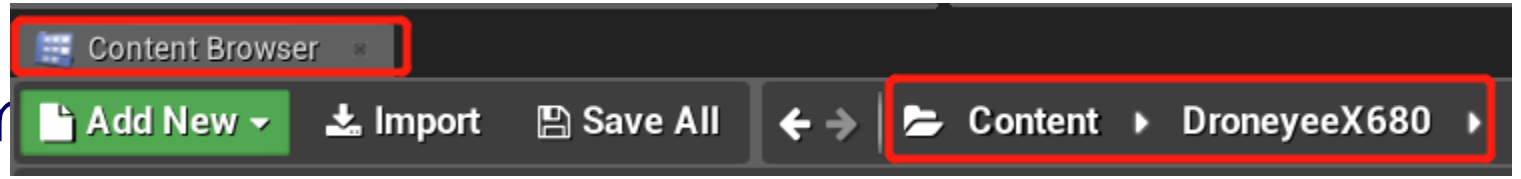
4.4 Export to get FBX model file

- Click 3Ds Max menu – **File** – **Export** - **Export**, configure as shown in the figure below, and export to get **DroneyeeX680Body.FBX** and **DroneyeeX680Prop.FBX** files
- The key export configuration is shown in the figure below. Note: Do not check cameras, animations, and lights options



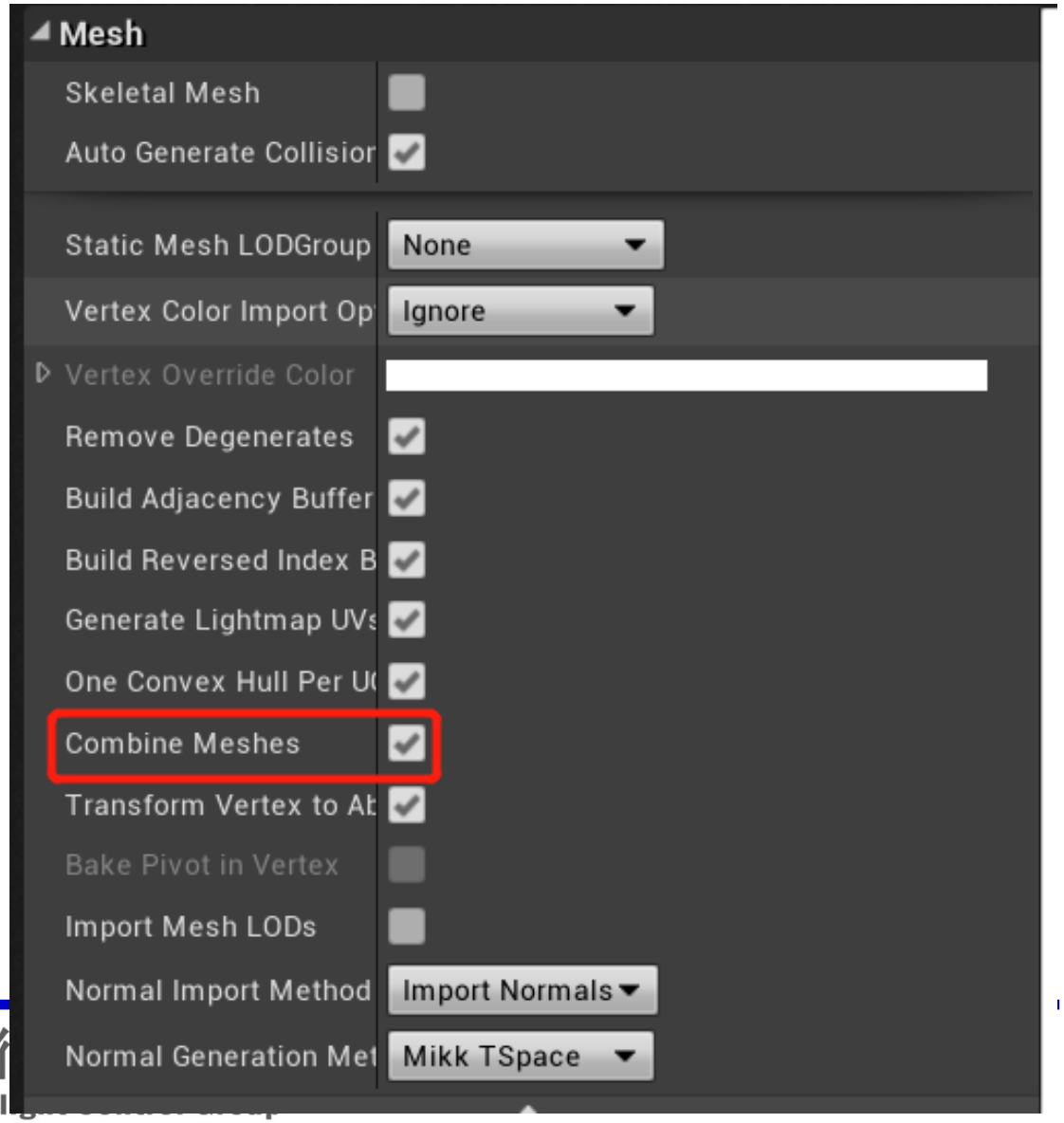


4. Vehicle 3D m



4.5 Import vehicle model in UE4

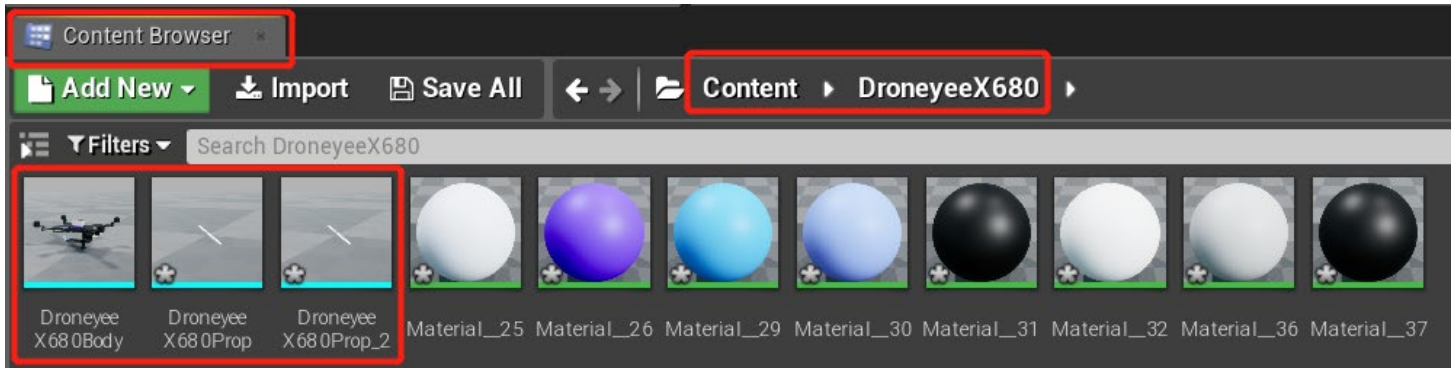
- Open a UE4 project at will, for example, open the **MyProj** project in the first two sections.
- Create a new folder in its Content (here named **DroneyeeX680**), enter the folder as shown below, click the Import button, and then import **DroneyeeX680Body.FBX** and **DroneyeeX680Prop.FBX** in turn.
- Note: Remember to check the "**Combine Meshes**" option when importing, so that all components form a whole import, otherwise the body will be divided into N parts.



4. Vehicle 3D model build and import

4.6 Generate and export models in UE4

- Since our model has no defined material, it is imported in white, which is not good looking. You need to double-click the **DroneyeeX680Body** component to add color materials to key components, such as carbon fiber black. At the same time, copy a **DroneyeeX680Prop** component named **DroneyeeX680Prop_2**, and set its material to a different color to distinguish the two front propellers and the back two propellers of the vehicle.



- Click "**Save All**" button in the above picture, and then click UE4 menu bar - **File-Package Project-Windows-Windows (64-bit).** Follow the tutorial method of generating maps, enable ray tracing and disable PAK packaging, you can generate identifiable 3D files then. Go to the generated file path "**WindowsNoEditor**"-**MyProj** (your project name)-"**Content**", copy the entire "**DroneyeeX680**" directory for use.

4. Vehicle 3D mode

4.7 Writing XML files

- After exporting the model, we also need to write an **XML** file to help **RflySim3D** to recognize the propeller position, rotation direction, material and etc.
- **XML** format file can refer to "**RflySimAPIs\UE4MapSceneAPI\VehicleModel\DroneyeeX680.xml**".
- The standard language format is a combination of various tags. Each tag is surrounded by two pairs of <> symbols **<***><*/***>**, and string or digital information is stored inside the tag. Sub-labels can be superimposed inside the label

DroneyeeX680.xml

E: > RflySimSource > RflySimAPIs > RflySimAPIsAdv3 > RflySimAPIs > UE4MapSceneAPI > VehicleMod

```xml
<?xml version="1.0"?>
<vehicle>
    <ClassID>3</ClassID>
    <DisplayOrder>1015</DisplayOrder>
    <Name>DroneyeeX680</Name>
    <Scale>
        <x>1</x>
        <y>1</y>
        <z>1</z>
    </Scale>
    <AngEulerDeg>
        <roll>0</roll>
        <pitch>0</pitch>
        <yaw>0</yaw>
    </AngEulerDeg>
    <body>
        <isAnimationMesh>0</isAnimationMesh>
        <MeshPath>/Game/DroneyeeX680/DroneyeeX680Body</MeshPath>
        <MaterialPath></MaterialPath>
        <AnimationPath></AnimationPath>
        <CenterHeightAboveGroundCm>8</CenterHeightAboveGroundCm>
        <NumberHeigthAboveCenterCm>20</NumberHeigthAboveCenterCm>
    </body>
    <ActuatorList>
        <Actuator>
            <MeshPath>/Game/DroneyeeX680/DroneyeeX680Prop</MeshPath>
            <MaterialPath></MaterialPath>
            <RelativePosToBodyCm>
                <x>25.4260</x>
                <y>24.0760</y>
                <z>11.2220</z>
            </RelativePosToBodyCm>
```

4. Vehicle 3D model build and im

Note: except the first few tags, not every tag needs to be included in the XML file, fill them in according to actual needs

4.7 XML language format

- **ClassID** label indicates the type of the vehicle, and **3** in RflySim3D means quadcopter.
- **DisplayOrder** indicates the order of priority in the list of available models of the quadcopter (the smaller the row is, the higher the priority). The priority of built-in models starts from 1000. If it is less than **1000**, it will replace the built-in vehicle and display the latest imported models first. **1015** in the template corresponds to the fourth-ranked vehicle.
- **Name** tag indicates the name of the vehicle displayed in UE4; the Scale tag indicates the size of the three-dimensional zoom of the whole machine.
- **AngEulerDeg** indicates that the vehicle is deflected by a certain angle (in degrees). For example, if you enter **yaw=180**, the initial attitude of the vehicle will turn its head back.
- **body** is the main label of the body.
- **isAnimationMesh** indicates whether it is a mesh with animation. It is generally necessary to select **0** here, except for some 3D models with animations (for example, walking people).
- **MeshPath** indicates the directory where the 3D files of the body are located, **where /Game/ indicates the game content directory, and DroneyeeX680/DroneyeeX680Body indicates the 3D model file of the body just copied**.
- **MaterialPath** indicates the path of the material. If the material has been set in UE4, it can be left blank. If it is filled in, it will be superimposed on the vehicle with this material.

4. Vehicle 3D model build and import

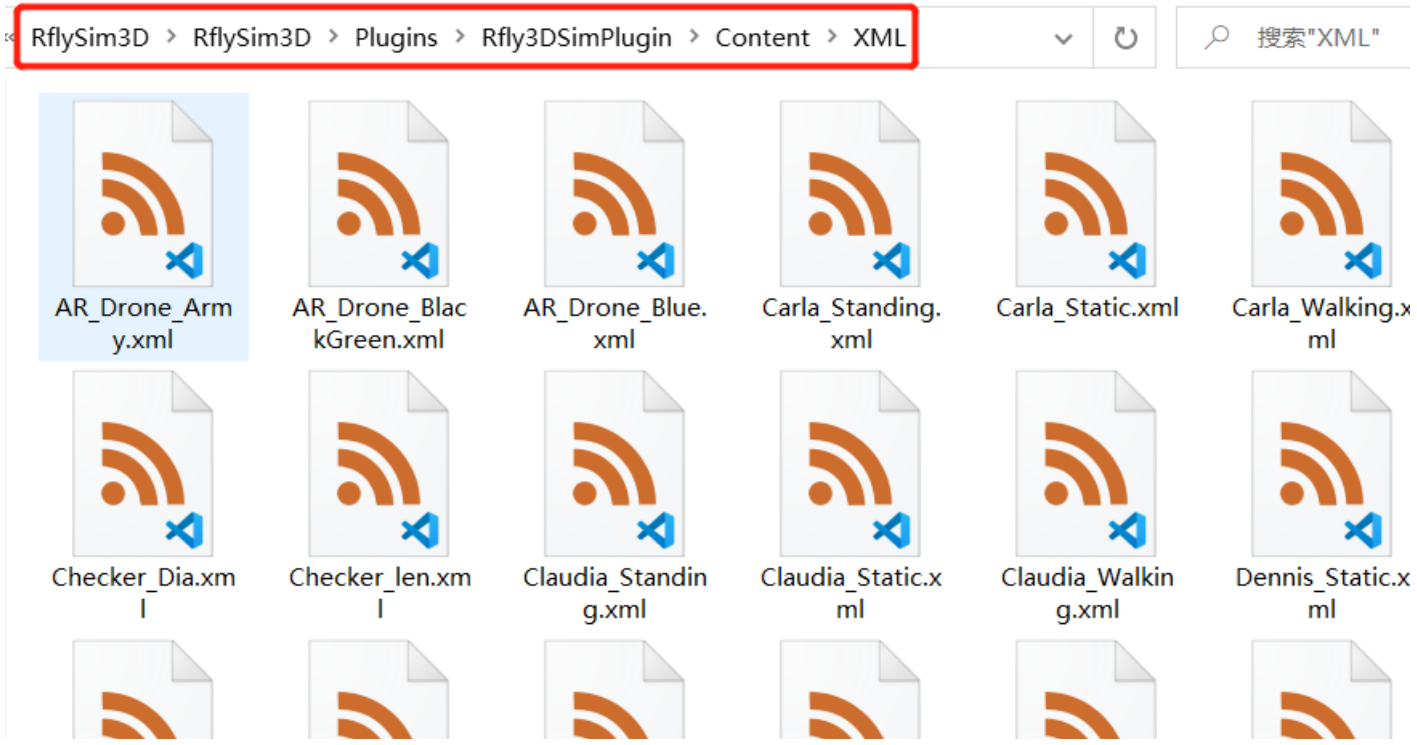
4.7 XML language format

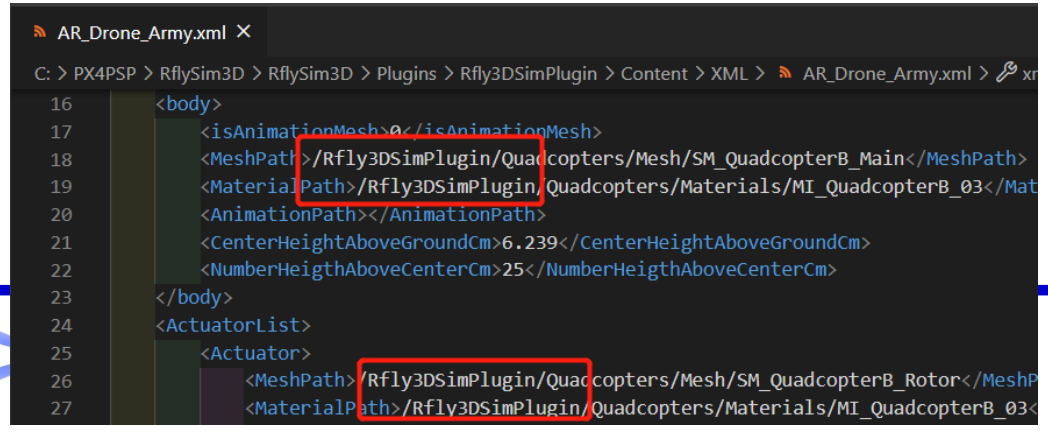
- **CenterHeightAboveGroundCm** indicates the height of the center of mass from the ground, here is 8 from 3Ds max
- **NumberHeigthAboveCenterCm** indicates the altitude displayed by the digital label on the vehicle
- **ActuatorList** is a list of external activity (definable) components such as actuators (propellers, steering gears, tires, hose rigid bodies, turntables)
- The Actuator tag contains the specific parameters of an actuator
- **MeshPath** 3D model path, defined with the body tag
- **MaterialPath** material path is defined with the body tag
- **RelativePosToBodyCm** represents the position of the actuator (propeller in this case) relative to the center of the body, and here is the value recorded in **Section 4.3**
- **RelativeAngEulerToBodyDeg** represents the installation angle of the actuator (in degrees), normally 0 is enough, which means that the posture is the same as when importing UE4, no need to rotate
- **RotationAxisVectorToBody** represents rotation axis, where the multicopter propeller rotates around the z axis, if it is a vehicle, it rotates around the y axis, if it is a fixed-wing's propeller, it rotates around the x axis.
- **RotationModeSpinOrDefect** represents motion mode of actuator inputs, 0: rotation speed (unit: RPM, around axis), 1: angular deflection (unit: degree, around axis), 2: static components, 3: translation (unit: cm, along axis)
- **OnboardCameras** tab defines some onboard cameras for observation. Starting from the second camera, they are all fixed on the body, so fill in the installation angle and position of the camera to define; the first camera is a follow-up observation viewing target and angle, it will not follow the vehicle's pitch and roll, but will follow the vehicle's yaw.

4. Vehicle 3D model build and import

4.8 More XML file demo examples

- All models or obstacle components of this platform are implemented through **XML** files
- You can visit the "**RflySim3D\RflySim3D\Plugins\Rfly3DSimPlugin\Content\XML**" folder under the installation directory to view the **.xml** suffix file and learn how to import other 3D entities.
- It contains examples of various objects such as cars, fixed-wing aircraft, small balls, light-emitting lights, stationary people, moving people, two-dimensional targets, etc. for learning



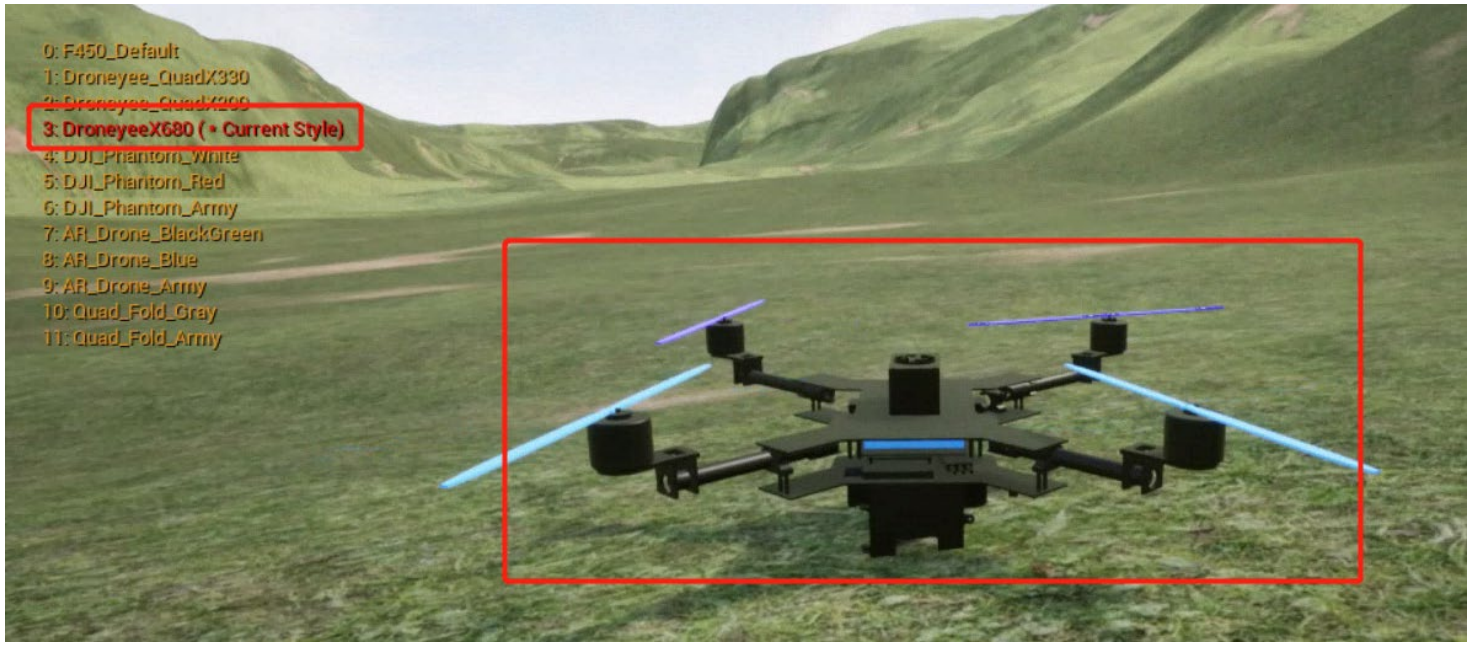


Note: XML demos in this folder use "/Rfly3DSimPlugin/" in MeshPath, which represents the Plugin Content folder. You should change it to "/Game/" for meshes in UE4 Content folder.

4. Vehicle 3D model build and import

4.9 Display imported model

- Copy the prepared XML file to the **DroneyeeX680** folder generated in **Section 4.6** (or directly use the **RflySimAPIs\UE4MapSceneAPI\VehicleModel\DroneyeeX680** folder), and then copy them together to the **RflySim3D\RflySim3D\Content** directory
- Open **CopterSim** and **RflySim3D**, press the key **C** to switch the vehicle style, and see if your vehicle model has been imported.
- **Note**: If the **DisplayOrder** number in XML is **less than 1000**, it will exceed the built-in model and be arranged to the forefront, becoming the **default display vehicle**




DroneyeeX680.max - Autodesk 3ds Max 2020
RflySim: How to import your own 3D vehicle models to the UE4-based RflySim3D program
Watch this video by clicking the following links:
YouTube: https: //youtu.be/mKUehJwqqsU
Youku: https: //v.youku.com/v_show/id_XNDcwNjA4NzIxMg==.html
[+] [Left] [Standard] [Wireframe]
[+] [Perspective] [Standard] [Default Shading]
Open a 3D model from 3Ds Max
1 Group Selected
Click or click-and-drag to select objects
Add Time Tag
ENG
21:29
2020/6/9

Content

1. Setup Instruction
2. UE4 3D scene build
3. RflySim3D 3D scene import
4. Vehicle 3D model build and import
5. Control demo of objects in scene
6. Summary

Path of source code in this section:
"RflySimAPIs\UE4MapSceneAPI\GetTerrainAPI"

5. Control demo of objects in scene

5.1 Obtaining the height map matrix in MATLAB

- Use MATLAB to enter the "**RflySimAPIs\UE4MapSceneAPI\GetTerrainAPI**" folder
- Enter and execute "**LoadPngData XXX**" in the command line window to import the desired map data, where XXX corresponds to a certain map file in the map folder. For example, use "**LoadPngData MountainTerrain**" to read the map information of high mountain terrain. After running the above script, a "**MapHeightData.mat**" file will be generated to store the height map matrix data.
- Note: This script will search for the two terrain files **XXX.png** (terrain grid) and **XXX.txt** (calibration data) needed. The search path is in order: the "**map**" folder under this path, "**..\.. \..\CopterSim\external\map**" folder, and "**[installation path]\CopterSim\external\map**" folder ([installation path] defaults to **C:\PX4PSP**, determined according to the platform installation configuration), please ensure that the XXX terrain file is in one of the above folders.
- **Note**: The core of this function is the "**LoadPngData.m**" file, you can view the algorithm by yourself, mainly importing **png** as a compressed matrix file, adding calibration data, and then saving it as a height map matrix

5. Control demo of objects in scene

5.2 Obtaining altitude terrain in MATLAB

- Run commands such as **getTerrainAltData(0,0)** and **getTerrainAltData(200,0)** directly in MATLAB to view the height of the desired coordinate (note that the north-east-down NED coordinate system is used here, and the downward is positive)
- The function of **getTerrainAltData** is to input the x and y coordinates of the map and output the current terrain height z
- Through this function, the height information of any position in the terrain can be obtained, so that a trajectory close to the ground can be created

```
>> clear all
>> LoadPngData MountainTerrain
>> getTerrainAltData(0, 0)

ans =

 -178.6947

>> getTerrainAltData(200, 0)

ans =

 -101.7023

fx >>
```

5. Control demo of objects in scene

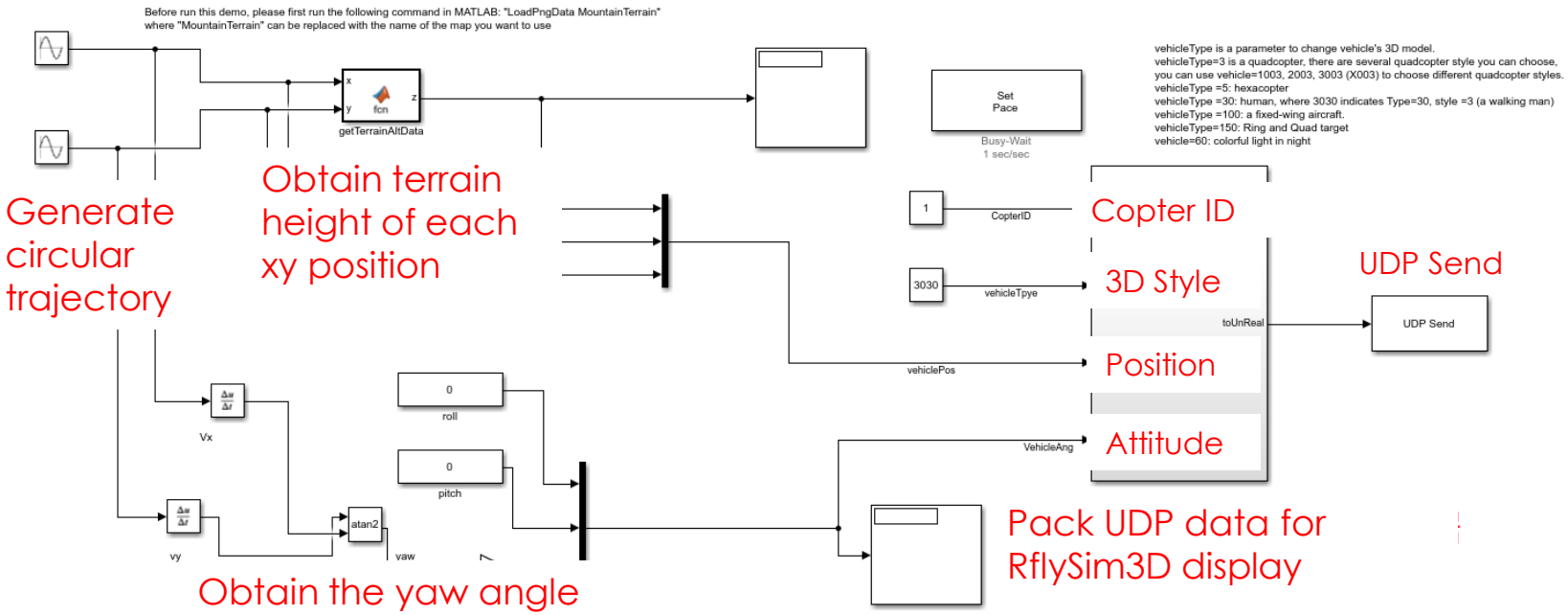
5.3 CopterSim height data verification

- Open CopterSim, select the 3D scene (for example, select "**MountainTerrain**"), enter the coordinates, select the "**PX4_SITL**" mode (or connect to Pixhawk to select the **PX4_HITL** mode), and then click "**Start Simulation**" button to view the height value of the current coordinates
- For example, input the XY coordinate values tested in MATLAB on the previous page and compare them with the results of CopterSim. (Note: The height value in CopterSim is reversed, and upward is positive)

5. Control demo of objects in scene

5.4 Generate altitude trajectory in Simulink

- Run the command "**LoadPngData XXX**" in MATLAB to import the desired map data, where XXX corresponds to a certain map file in the map folder. For example, use "**LoadPngData MountainTerrain**" to read height map information.
- Run RflySim3D program, keyboard input "**M 3**" to switch to "**MountainTerrain**" three-dimensional map
- Open "**TrajGen.slx**" and run. The **getTerrainAltData** module calls the MATLAB function in the previous section.



5. Control demo of objects in scene

5.4 Trajectory control effect in Simulink

- Run the "**TrajGen.slx**" file
- You can see the demo of people walking along the terrain surface in the scene as follows. Note: Through this interface, functions such as tracking objects and targets can also be realized for visual tracking or obstacles during hovering flight
- Note: **vehicleType** is a variable that controls 3D objects.
- **vehicleType=3** corresponds to a quadcopter. RflySim3D has multiple quadcopter styles to choose from. Enter **vehicleType =1003, 2003, 3003** etc. to select specific styles
- **vehicleType = 5** corresponds to a hexacopter
- **vehicleType = 30** corresponds to a person, where **3030** represents a person with a style of **3**, which corresponds to the person walking in the picture on the right
- **vehicleType = 100** corresponds to a fixed-wing aircraft.
- **vehicleType=150** corresponds to targets such as ring and square
- **vehicleType=60** corresponds to luminous, etc., used for light show display

5. Control demo of objects in scene

5.5 Switch view interface

- You can use the mouse and shortcut keys (**V** or **N**) **in RflySim3D** to switch between different camera perspectives at runtime. This method can only be qualitatively adjusted
- The **RflyCameraPosAng.m** file is a script to control the camera's perspective, directly input in the **MATLAB command** line window (only RflySim Vision version and Full version)

RflyCameraPosAng x y z roll pitch yaw

- Where x y z unit m represents the position of the camera relative to the center of the vehicle, roll, pitch, yaw unit degrees, represents the camera attitude angle, such as input

RflyCameraPosAng 0.2 0 0 0 0 0

- Equivalent to the angle of view of a front camera (0.2m forward from the center)

5. Control demo of objects in scene

5.6 Multi-vehicle trajectory generation example (same type of vehicle)

- Same as the above example, open the RflySim3D program and press "**M**" to switch to the "**MountainTerrain**" 3D map
- Run the "**TrajGenMulti.slx**" program, you can see that multiple vehicle take off from the ground, fly a rectangular trajectory and then land

5. Control demo of objects in scene

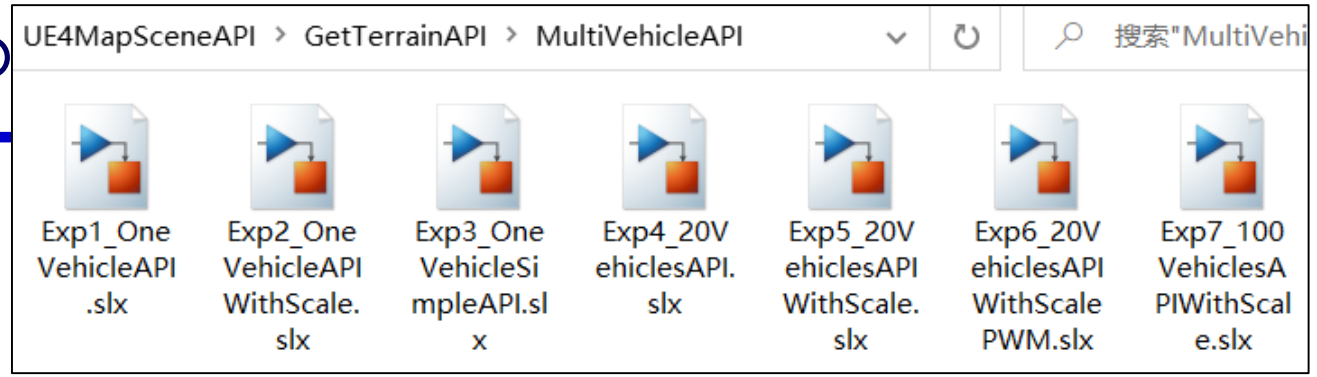
5.7 Multi-vehicle trajectory generation example (different vehicle styles)

- Close the above Simulink demo, and press key "**ESC**" in RflySim3D to clear all vehicle
- Then run the "**TrajGenMulti2.slx**" demo, you can see examples of various drones and cars running on ice.
- Press key "**S**" in RflySim3D to display the vehicle ID, press "**B**" to switch between different vehicles, and press "**N**" or "**V**" to switch different perspectives.

5. Control demo of o

5.8 Large scale swarm simulation

- In the following folder "**RflySimAPIs\UE4MapSceneAPI\GetTerrainAPI\MultiVehicleAPI**" , Simulink demos are presented to send data and control 3D objects in RflySIm3D with APIs for **1/20/100** vehicles.
- With these APIs, you can create multiple 3D objects (vehicle/ character/barrier models) through one UDP send module, which is easier to use with higher performance.
- You can use the UDP communication struct defined in these demos to write APIs in Python/C or other programming languages



Thanks