

UE材质使用规范

1. 实验目的

熟悉UE4.27和UE5.2的材质使用，避免出现材质丢失问题。

2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链^[1]；Unreal Engine 4.27；Unreal Engine5.2。
- 硬件要求：笔记本/台式电脑1台^[2]。

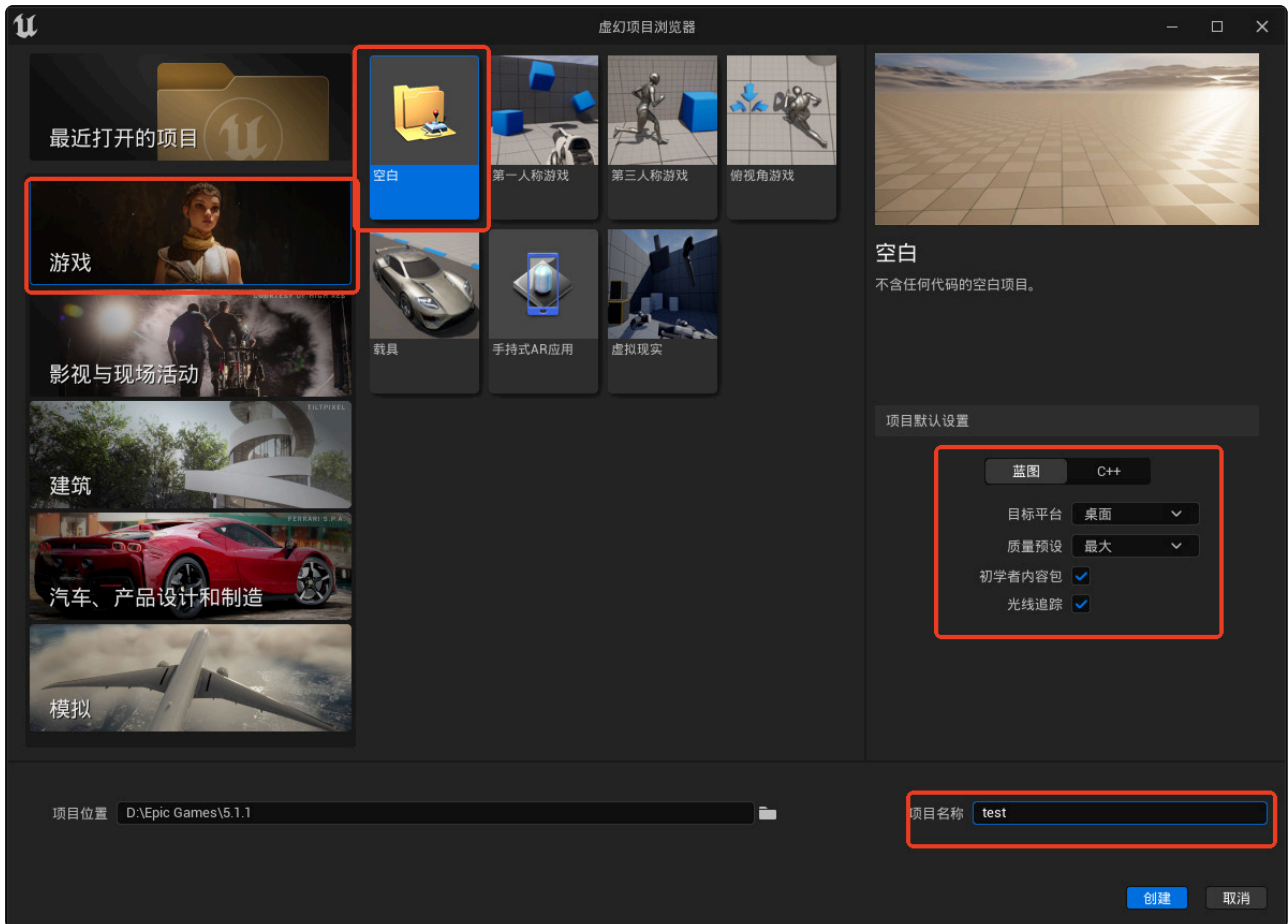
3. 实验地址

例程目录：

[\[安装目录\]\RflySimAPIs\3.RflySim3DUE\1.BasicExps\e0_StarterContent\3.MaterialContent](#)

4. 实验内容或步骤

首先以如下配置新建一个UE5.2项目：



本文档主要讲解UE（特别是UE5）编辑器中可见的材质导入RflySim3D后失效的问题。

4.1 步骤1：检查材质引用

步骤1.1 通过引用查看器检查材质引用路径是否正确

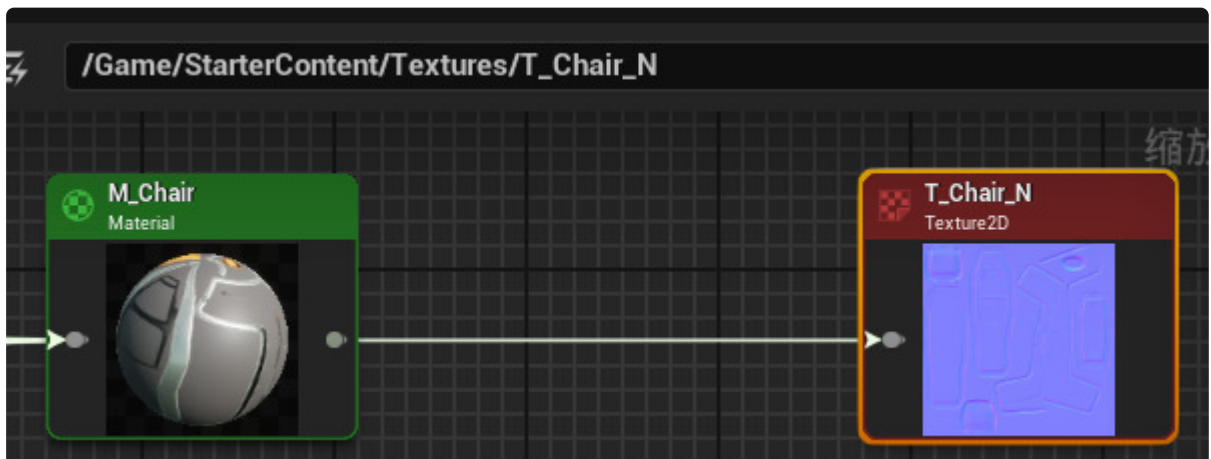
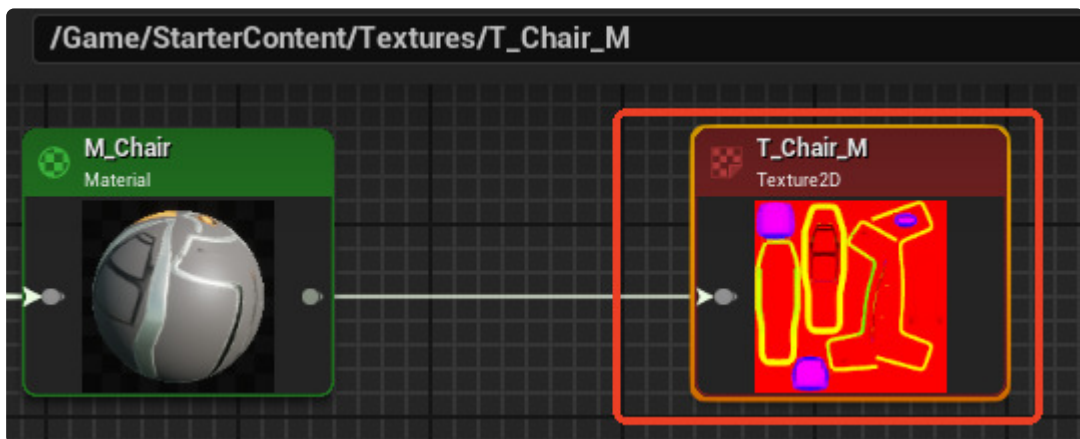
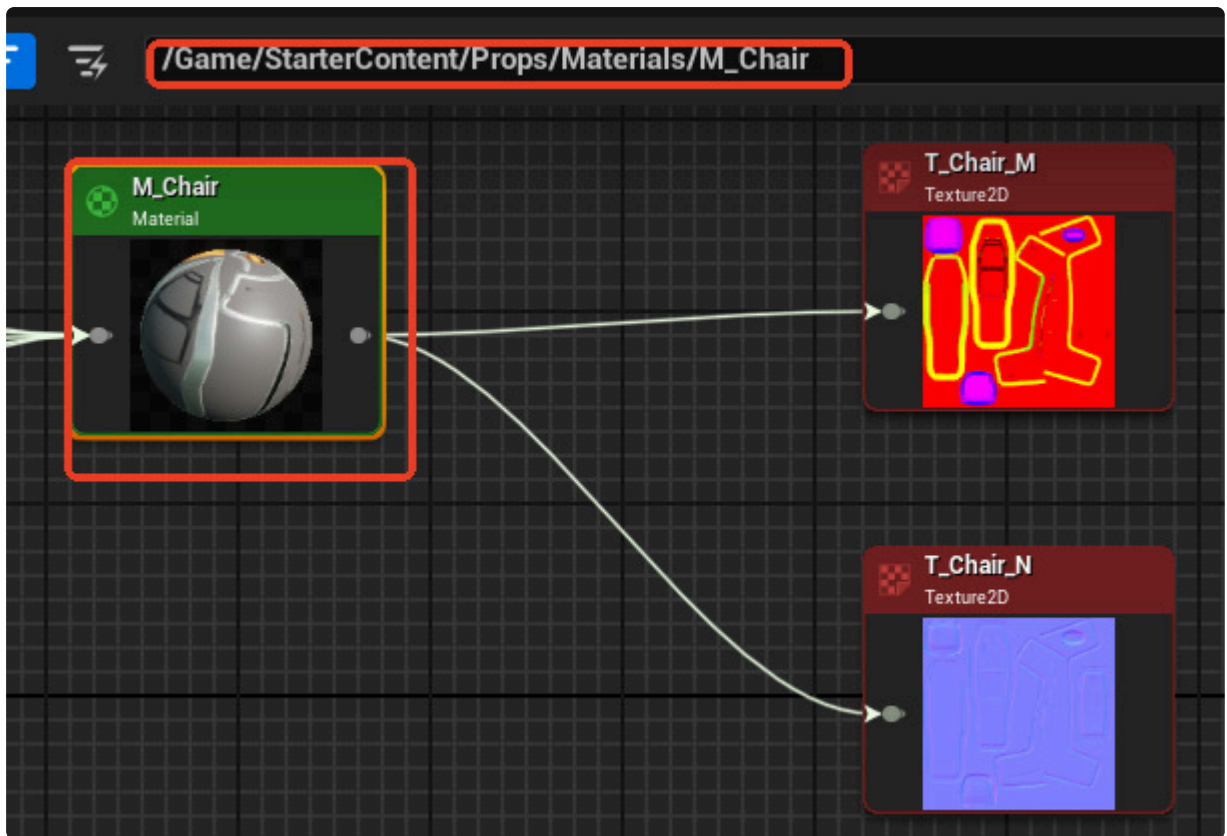
使用引用查看器可以帮助确认材质是否被正确引用，若引用了引擎默认材质，可参考如下步骤修改：[Checkmaterial.pdf](#)

操作步骤：

1. 在 Unreal Engine 编辑器中，右键点击材质资产。
2. 选择 `Reference Viewer`。



3. 在弹出的引用查看器中，查看材质的引用关系，确认是否有错误的引用。如下图所示，材质本身及其引用的纹理和法线贴图都处于项目内容目录下，整个项目内容烘焙后可以一起导入RflySim。



步骤1.2 通过输出日志查看是否有引用缺失

在项目内容烘焙完成后，查看输出日志窗口，其中显示了 Unreal Engine 编辑器运行时的所有日志信息，包括材质加载和引用的详细信息。

操作步骤：

1. 在 Unreal Engine 编辑器中，点击菜单栏中的 **Window** > **Output Log**。



2. 在输出日志窗口中，搜索关键字 **Material** 或 **材质** 查看与材质相关的日志信息，特别是包含 **Missing Material** 或 **Material Reference Error** 的条目。如下图可以看到所有材质引用信息。

```

LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Materials/M_Wood_Pine.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Materials/M_Wood_Pine.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Materials/M_Wood_Malrus.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Materials/M_Wood_Malrus.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Props/Materials/Sphere.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/Sphere.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Props/Materials/M_Bush.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_Bush.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Props/Materials/M_Chair.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_Chair.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Props/Materials/M_Door.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_Door.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Props/Materials/M_Frame.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_Frame.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Props/Materials/M_Lamp.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_Lamp.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Props/Materials/M_Rock.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_Rock.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Props/Materials/M_Shelf.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_Shelf.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Props/Materials/M_Statue.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_Statue.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Props/Materials/M_TableRound.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_TableRound.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Particles/Materials/M_Dust_Particle.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Particles/Materials/M_Dust_Particle.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Particles/Materials/M_Explosion_subUV.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Particles/Materials/M_Explosion_subUV.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Particles/Materials/M_Fire_SubUV.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Particles/Materials/M_Fire_SubUV.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Particles/Materials/M_Flare_01.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Particles/Materials/M_Flare_01.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Particles/Materials/M_Heat_Distortion.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Particles/Materials/M_Heat_Distortion.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Particles/Materials/M_Radial_Gradient.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Particles/Materials/M_Radial_Gradient.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Particles/Materials/M_Radial_Ramp.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Particles/Materials/M_Radial_Ramp.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Particles/Materials/M_Smoke_SubUV.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Particles/Materials/M_Smoke_SubUV.usasset"
LogPackFactory: Copied ".../Samples/StarterContent/Content/StarterContent/Particles/Materials/M_Spark.usasset" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Particles/Materials/M_Spark.usasset"
LogMaterial: Display: Material /Game/StarterContent/Props/Materials/M_TableRound needed to have new flag set bUsedWithNanite !
Cmd: OBJ_SAVEPACKAGE PACKAGE=/Game/StarterContent/Props/Materials/M_Chair FILE=D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_Chair.usasset SILENT=true
LogSavePackage: Moving output files for package: /Game/StarterContent/Props/Materials/M_Chair
LogSavePackage: Moving "D:/Epic Games/5.1.1/test/Save/M_Chair/FD0941704FECC0693E5462993B41B089.tmp" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_Chair.usasset"
Cmd: OBJ_SAVEPACKAGE PACKAGE=/Game/StarterContent/Props/Materials/M_TableRound FILE=D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_TableRound.usasset SILENT=true
LogSavePackage: Moving output files for package: /Game/StarterContent/Props/Materials/M_TableRound
LogSavePackage: Moving "D:/Epic Games/5.1.1/test/Save/M_TableRound/9E2FEC546C288AF60E329B70A099ED5.tmp" to "D:/Epic Games/5.1.1/test/Content/StarterContent/Props/Materials/M_TableRound.usasset"
LogContentValidation: Display: 正在验证 /Script/Engine.Material /Game/StarterContent/Props/Materials/M_Chair

```

4.2 步骤2：检查Nanite支持

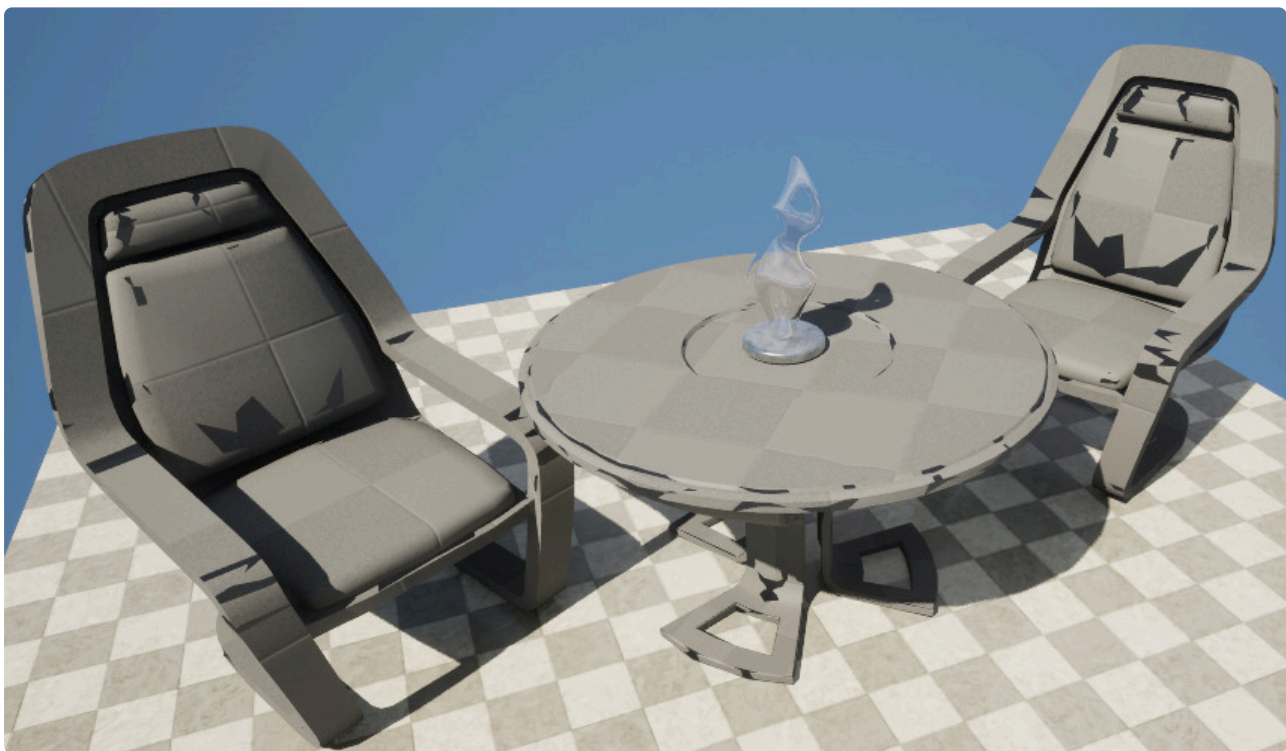
对于UE5.2，如果启用了Nanite的网格体使用的材质缺少**bUsedWithNanite** 标记，输出日志中通常会提出如下警告：

```

⚠ M_TableRound 材质 /Game/StarterContent/Props/Materials/M_TableRound/M_TableRound 缺失使用标记bUsedWithNanite。如果材质资产未重新保存，那么其在编辑器外运行时可能无法正确渲染。🔧 修复
⚠ M_Chair 材质 /Game/StarterContent/Props/Materials/M_Chair/M_Chair 缺失使用标记bUsedWithNanite。如果材质资产未重新保存，那么其在编辑器外运行时可能无法正确渲染。🔧 修复
• 地图检测完成：0个错误、2个警告、完成用时0.128毫秒。

```

且相应材质会被替换为默认材质，导致场景中"材质丢失"：



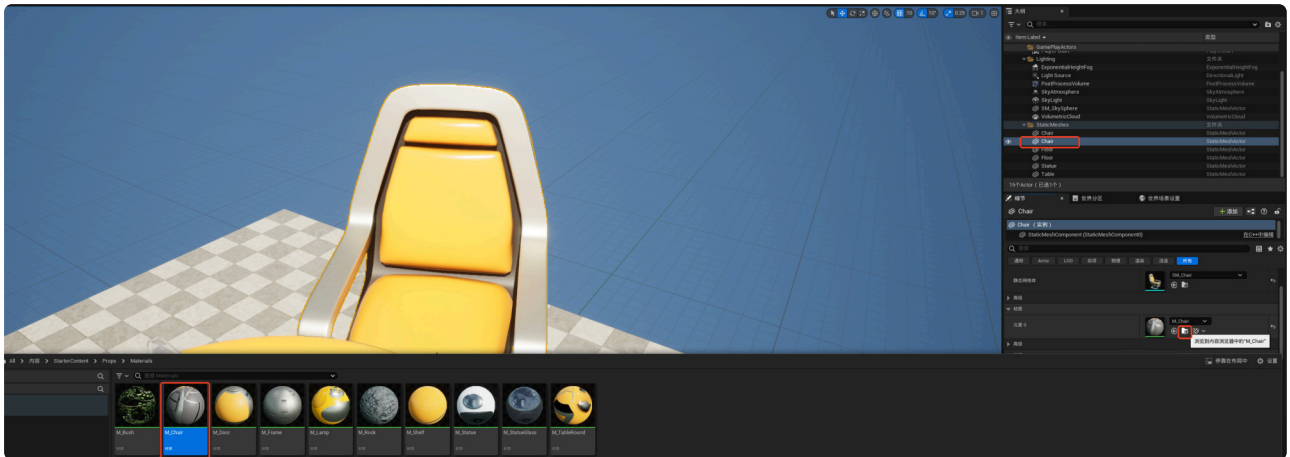
解决步骤

步骤1：打开有问题的材质

- 在 **Content Browser** 中找到

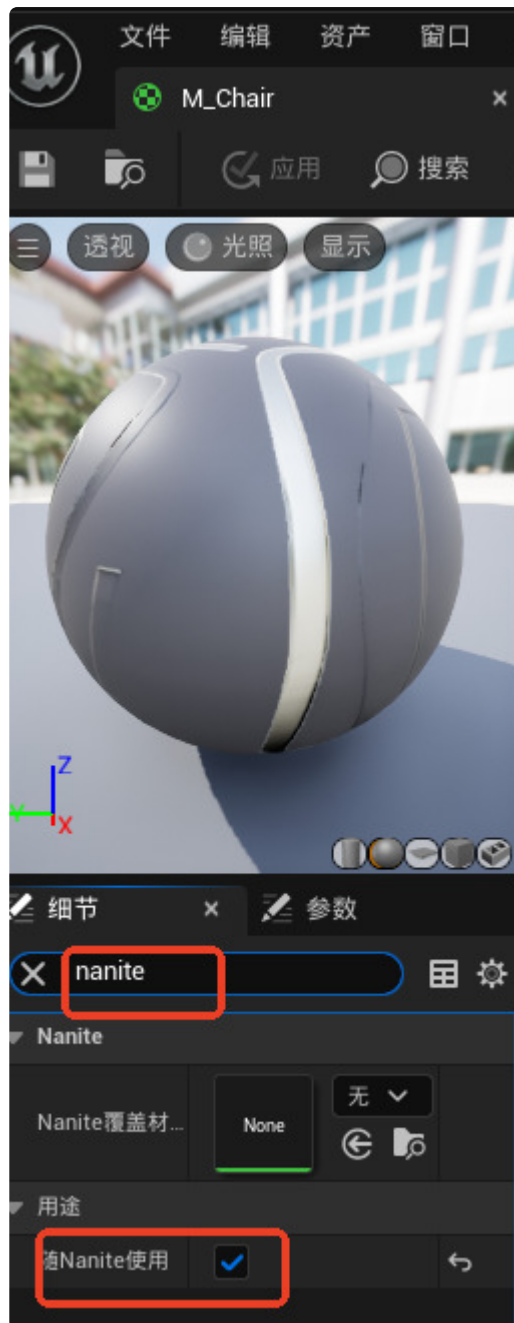
`/Game/StarterContent/Props/Materials/M_TableRound` 和 `M_Chair`，双击打开它

们。



步骤2：启用 "Used With Nanite" 标记

- 在材质编辑器中，细节面板勾选 **随Nanite使用** (bUsedWithNanite) 选项 ([Epic Developer Community Forums](#), [Reddit](#))。



步骤3：保存材质并重启

- 勾选后点击 **Save**。
- 重启编辑器以确保设置不被覆盖。

注意：如果材质保存在官方 StarterContent 中，每次启动可能会被引擎重置。

步骤4：清理 StartupActions 设置（可选但建议保留修改）

如果使用的是 Starter Content 模板，UE 可能会在 `DefaultGame.ini` 中自动重新导入这些内容，每次启动都会恢复原始状态并清除你的更改 ([GameDev.tv](https://www.gamedev.tv)):

操作路径： 项目根目录 → `Config/DefaultGame.ini` 删除或注释以下段落：

```
1 | [StartupActions]
2 | bAddPacks=True
3 | InsertPack=(PackSource="StarterContent.upack",PackName="StarterContent")
```

这样可确保材质改动真正生效，不会被重写。

经过上述步骤后重新烘焙并导入RflySim，可以看到模型部分细节可能失真：



这是因为Nanite支持光线追踪功能，但光线只与Nantie网格体的代理模型（回退网格体）相交，而不是与拥有完整细节的Nanite网格体相交。

此时可以在RflySim3D中通过控制台变量 `r.RayTracing.Nanite.Mode 1` 启用对Nanite网格体的原生光线追踪的初始支持。



r.RayTracing.Nanite.Mode 1



4.3 步骤3：UE材质处理完整 workflow

步骤3.1 创建材质

步骤3.1.1 新建材质

1. 在 **Content Browser**（内容浏览器）中，右键点击空白区域，选择 **Material**，并为其命名。
2. 双击打开材质编辑器，进入材质编辑界面。

步骤3.1.2 材质类型选择

- 在 **Material Editor**（材质编辑器）中，设置材质类型。材质的主要类型有：
 - **Surface**：最常用，适用于大多数对象。
 - **Volume**：适用于体积效果，如雾气。
 - **Post Process**：适用于后期处理材质。
 - **Light Function**：用于灯光的影响。

步骤3.1.3 设置材质属性

1. **Material Domain**：设置为 **Surface**（大多数情况）或者其他类型（如 **Volume**）。
2. **Shading Model**：设置为不同的阴影模型（如 **Opaque**，**Transparent**）。
3. **Blend Mode**：选择 **Opaque**（不透明）或 **Translucent**（透明）等。

步骤3.2 材质编辑与节点设置

步骤3.2.1 基本材质节点

- **Base Color**：决定颜色的主节点。
- **Metallic**：决定物体的金属感（0为非金属，1为金属）。
- **Roughness**：决定表面的粗糙度。
- **Specular**：反射的强度。

步骤3.2.2 高级节点与特效

- **Normal Map**：为表面添加法线纹理，用于模拟表面细节。
- **Emissive Color**：发光颜色。
- **Opacity**：透明度，适用于透明材质。
- **Fresnel Effect**：控制物体的光泽效果，增加视觉层次感。
- **Reflection**：通过反射模型来增加环境光的映射。

步骤3.2.3 引入纹理贴图

1. 将纹理（如 Normal Map、Roughness Map、Specular Map）拖拽到材质编辑器中。
2. 将纹理连接到对应的材质节点。

步骤3.3 应用材质

步骤3.3.1 将材质应用于模型

- 将材质从 `Content Browser` 拖拽到模型或对象上，直接应用材质。
- 或者，在对象的 **Details Panel**（细节面板）中，选择材质槽并应用相应材质。

步骤3.3.2 材质实例化

- 创建 `Material Instance`（材质实例）以便对材质进行修改而不影响原始材质。
 - 在材质实例中，你可以调整参数（如颜色、粗糙度、金属度等）而不需要重新编译整个材质。
-

步骤3.4 材质优化

步骤3.4.1 性能优化

1. **减少复杂度**：避免过多复杂的节点计算，减少 GPU 负担。
2. **合并纹理**：尽量合并多个小纹理为一张大纹理（Atlas）来减少 Draw Call 数量。
3. **使用简单的材质模型**：比如使用 `Unlit` 模型来简化计算。

步骤3.4.2 合并材质和纹理

- 使用 `Material Instances` 来复用相同的材质基础，并调整实例的参数。
- 通过合并纹理图层（例如，基础色、金属、粗糙度）来减少内存使用。

步骤3.4.3 动态材质优化

- **动态参数**：在 `Material Instance` 中，通过 `Parameter` 动态修改材质属性（如时间变化、光照变化等）。
-

步骤3.5 调试与问题排查

步骤3.5.1 使用材质调试工具

1. 打开 `Material Editor` 中的 **Preview** 面板，查看材质效果。
2. 在 `Content Browser` 中选择材质，点击右键并选择 **Check Shader Complexity**，检查材质的复杂度。

步骤3.5.2 日志检查

1. 查看 `Output Log` 中的日志，检查是否有材质加载错误或引用错误。
2. 如果有 `Missing Material` 或 `Material Reference Error`，说明该材质无法找到或丢失引用。

步骤3.5.3 检查材质实例化问题

- 如果使用了 `Material Instances`，确保实例中的所有参数都正确设置并且没有冲突。
 - 检查是否在 `Material Instance` 中引用了不存在的纹理或贴图。
-

步骤3.6 特效与进阶应用

步骤3.6.1 自定义着色器

- 通过 `Custom` 节点，编写 GLSL 或 HLSL 代码，实现自定义效果。

步骤3.6.2 PBR 材质

- 使用基于物理的渲染（PBR）材质，以获得更真实的光照和反射效果。通常包括 `Base Color`、`Metallic`、`Roughness`、`Normal Map` 等。

步骤3.6.3 虚拟纹理与 Nanite

- 使用 **Virtual Textures** 来处理大规模纹理的加载与渲染，提高性能。
 - 在使用 **Nanite** 时，确保材质的兼容性，以避免出现丢失材质或材质无法显示的问题。
-

步骤3.7 材质的动态与动画处理

步骤3.7.1 动画材质参数

- 使用 **Timeline** 或 **Material Parameter Collections** 来动态控制材质的属性（如时间变化、颜色变化等）。

步骤3.7.2 场景中动态材质的使用

- 动态控制材质的颜色、粗糙度、透明度等，例如，在游戏过程中通过脚本或蓝图修改材质效果。

步骤3.7.3 后期处理材质

- 使用 **Post Process Materials** 来影响整个场景的外观，如模糊、色彩调整、边缘检测等。

5. 关键知识点

关键知识点1：材质引用检查方法

使用引用查看器（Reference Viewer）可以帮助确认材质是否被正确引用，确保材质和纹理都处于项目内容目录下，以便烘焙后一同导入RflySim。

关键知识点2：Nanite材质兼容性

在UE5中，如果启用了Nanite的网格体使用的材质缺少**bUsedWithNanite**标记，会导致材质被替换为默认材质，造成"材质丢失"现象。

关键知识点3：材质烘焙和导入流程

材质在烘焙后导入RflySim时，需要注意材质路径和依赖关系，确保所有引用的纹理和资源都能正确打包和传输。

关键知识点4：性能优化策略

包括减少复杂节点计算、合并纹理、使用材质实例等方法来优化材质性能，减少GPU负担。

6. 参考资料

1. [RflySim官方文档](#)
2. [UE4学习笔记：材质系统使用入门 - U_N_Owen - 博客园](#)
3. [Epic Developer Community Forums](#)

7. 常见问题

Q1：为什么在RflySim3D中某些材质会变成默认材质（粉红色）？

A1：这通常是由于材质引用路径不正确或者Nanite材质未启用**bUsedWithNanite**标记导致的。需要检查材质引用关系，确认材质和相关纹理都在项目目录下，同时确保使用Nanite的网格体所关联的材质已勾选"随Nanite使用"选项。

Q2：如何修复UE5中Nanite网格体材质丢失的问题？

A2：打开有问题的材质，在材质编辑器的细节面板中勾选"随Nanite使用"（**bUsedWithNanite**）选项，然后保存材质并重启编辑器。如果使用的是StarterContent中的材质，还需要删除或注释Config/DefaultGame.ini中的StartupActions部分，防止每次启动时被引擎重置。

Q3：烘焙后的材质在RflySim3D中出现细节失真的问题怎么办？

A3：这是由于Nanite支持光线追踪功能，但光线只与Nanite网格体的代理模型（回退网格体）相交，而不是与拥有完整细节的Nanite网格体相交造成的。可以在RflySim3D中通过控制台变量 `r.RayTracing.Nanite.Mode 1` 启用对Nanite网格体的原生光线追踪的初始支持。

-
1. <https://rflysim.com/> ↩

2. 推荐配置请见: <https://rflysim.com/> ↩