**Getting Started - Chapter 2 - Combining Meshes**

**@@启程-第二章-合并网格@@**

**Combining Meshes Using Merge Meshes**

**@@使用“合并网格”方法合并网格@@**

This is a straight forwarded way of combining two or more meshes.

@@这是一个直截了当的合并两个或更多网格的方法@@

const combined = BABYLON.Mesh.MergeMeshes(Array\_of\_Meshes\_to\_Combine)

In our case this would be

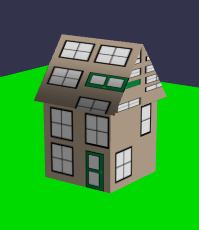
@@在我们当前的例子中将会是@@

const house = BABYLON.Mesh.MergeMeshes([box, roof])

Combining Meshes In Your Scene

在你的场景中合并网格

https://playground.babylonjs.com/#KBS9I5#75



The first thing we note is that the whole house is covered in only one of the materials used. Fortunately this can be corrected using the multiMultiMaterial parameter of *MergeMeshes*, unfortunately this is the final parameter of a long list. The code now looks like

@@我们首先发现的问题是这个房子都被唯一使用的一个材质覆盖了。幸运的是这个问题可以通过使用“*MergeMeshes*”方法的“多重材质”属性修正。新的代码是这样的： @@

const house = BABYLON.Mesh.MergeMeshes([box, roof], true, false, null, false, true);

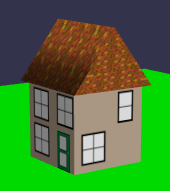
At this stage it is important to note that the second parameter being true disposes of the original meshes and the last parameter being true allows the original material to be applied separately to the parts matching the original meshes.

@@目前需要知道的是，第二个参数为true表示在合并网格后，回收组成合并网格的源网格（房体和屋顶网格），最后一个参数为true表示合并网格中属于源网格的部分分别使用对应的源材质@@

Combining Meshes And Preserving Material Assignments

合并网格并且保留材质指定

https://playground.babylonjs.com/#KBS9I5#76



Before considering how to make multiple copies of our house we will first: find out the basics of [exporting](https://doc.babylonjs.com/extensions/glTFExporter) our models; how to [import](https://doc.babylonjs.com/divingDeeper/importers) models made with Babylon.js or other software; and how to display your scene or models on your own website.

@@在考虑如何为我们的房子建立大量复制体之前，我们先要：掌握导出我们的模型的基础方法；以及了解如何导入Babylon.js或其他软件生成的模型；并且还要明确如何在你自己的网站上显示你的场景和模型。@@

（译者注：关于Babylon.js训练场导出还有一些难点，首先训练场导出的.babylon文件只是文本文件，并不包含纹理图片，在实际使用时还是要根据url去assets.babylonjs.com网站获取图片，为了本地化，可以在训练场中填写本地的相对地址，然后运行、导出，当然，这样将无法在训练场中预览到纹理（.glb格式则在导出文件中包含了纹理内容，不需要额外加载！）。另外训练场中的代码并没有为导出的每个网格分配不同的id，这样当一个场景加载由训练场导出的多个模型时，可能会出现网格id重复的情况（还有材质id重复），这并不会导致Babylon.js出错，因为Babylon.js使用一个数字类型的“唯一id”而非id区分场景中的对象，但会影响调试和网格查找，在训练场代码中明确指定不同的id可以避免这一问题。）

**Further reading 延伸阅读**

Merging Meshes 合并网格

Learn how to merge meshes in Babylon.js. 学习如何在Babylon.js中合并网格

https://doc.babylonjs.com/divingDeeper/mesh/mergeMeshes