**Getting Started - Chapter 2 - Face Materials**

**@@启程-第二章-面材质@@**

**Table Of Contents 内容列表**

[Material For Each House Side 为房子的每个面设置不同的材质](https://doc.babylonjs.com/start/chap2/face_material" \l "material-for-each-house-side)

[Detached House Example 正方形房子的例子](https://doc.babylonjs.com/start/chap2/face_material" \l "detached-house-example)

[Semi Detached House Example 长方形房子例子](https://doc.babylonjs.com/start/chap2/face_material" \l "semi-detached-house-example)

[Moving On 更进一步](https://doc.babylonjs.com/start/chap2/face_material" \l "moving-on)

**Material For Each House Side**

**@@房子的每个面的材质@@**

In the options properties for a box one is *faceUV* and array of Vector4s. We can use this to obtain a part of the area of an image to apply to one face of the box.

@@盒子网格的选项参数有一个“*faceUV* ”属性，这是一个由四维向量作为元素的数组。我们可以使用这个属性来安排图片中的一个区域对应盒子网格的一个面。@@

In the *faceUV* array faces are numbered 0 for back, 1, front, 2 left, 3 right, 4 top and 5 bottom.

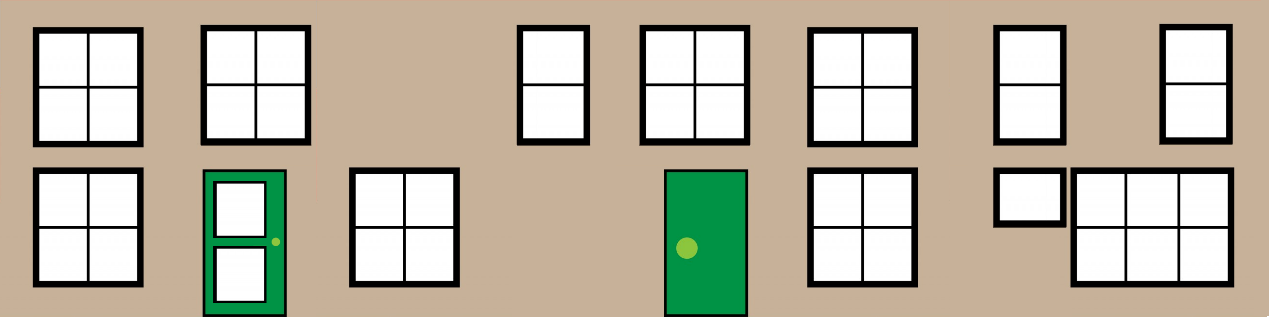
@@在“*faceUV* ”数组中，盒子的六个面被编号，0对应背面、1对应前面、2对应左面、3对应右面、4对应顶面、5对应底面@@

**Detached House Example**

**@@单体房子示例@@**

We will start with this image

@@我们将从这张图片开始@@



which contains, in order, images of the same size for the front, right, back and left sides of the house.

@@这张图按顺序包含了四幅相同尺寸的图片，分别对应房子的前面、右面、背面、左面@@

The width of each image is 0.25 of the whole image width. To specify the part of the image to use we give two co-ordinates one for the lower left corner and one for the upper right corner. For the whole image we would use (0, 0) and (1, 1), for part images the co-ordinate values will be a fraction between 0 and 1.

@@每张小图片的宽度为整个图片宽度的四分之一。为了区分要使用的图片区域，我们要提供两个坐标值，分别表示区域左下角和右上角的坐标。对于整个大图来讲这两个坐标值是(0, 0) 和 (1, 1)，对于大图的一部分，这些坐标值将在0到1之间变化@@

Rather then using two sets of co-ordinates we use a 4 dimensional  
vector (lower left x, lower left y, upper right x, upper right y)

@@我们用一个4维向量表示这两个坐标值（左下角的x值，左下角的y值，右上角的x值，右上角的y值）@@

Matching sides to part images gives

@@将盒子网格的面与图像的部分对应起来@@  
front, 1, (0.0, 0.0, 0.25, 1.0)  
left, 2, (0.25, 0, 0.5, 1.0)  
back, 0, (0.5, 0.0, 0.75, 1.0)  
right, 3, (0.75, 0, 1.0, 1.0)  
as the top and bottom are not seen we will just use the defaults.

@@因为顶部和底部不可见，我们直接使用默认值即可（(0,0,1,1)）@@

We set these using

@@用以下方法设置它们：@@

faceUV = [];

faceUV[0] = new BABYLON.Vector4(0.5, 0.0, 0.75, 1.0); //rear face 后面

faceUV[1] = new BABYLON.Vector4(0.0, 0.0, 0.25, 1.0); //front face 前面

faceUV[2] = new BABYLON.Vector4(0.25, 0, 0.5, 1.0); //right side 右面

faceUV[3] = new BABYLON.Vector4(0.75, 0, 1.0, 1.0); //left side 左面

Unless we set another option property, *wrap = true*, some of these partial images will still be rotated. We create the box like this

@@除非我们设置另一个选项属性——wrap为true，一些图块还是会被旋转。我们需要这样建立盒子网格：@@

const box = BABYLON.MeshBuilder.CreateBox("box", {faceUV: faceUV, wrap: true});

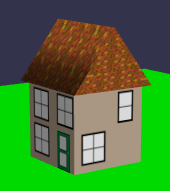
Of course we also need to change the image used for the diffuse texture of the box material.

@@当然，我们也要修改盒子的材质的漫反射纹理所使用的图片的url@@

Adding Materials To Individual Object Faces

为物体的单个面添加材质

https://playground.babylonjs.com/#KBS9I5#72

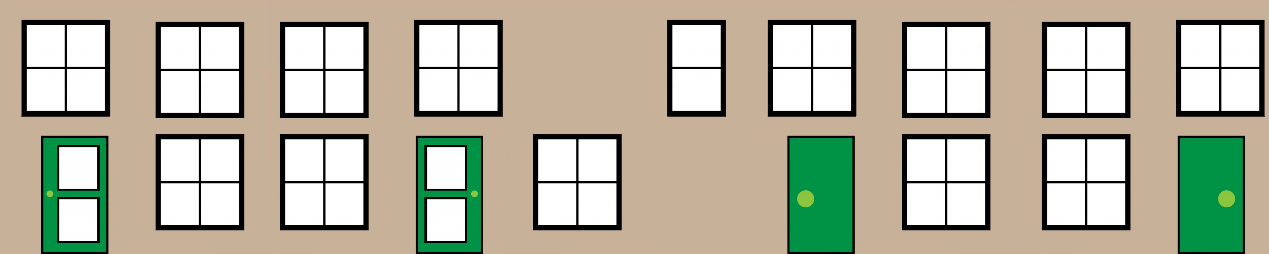


**Semi Detached House Example**

**@@双体式房子的例子@@**

In this case the house is twice as wide and so are parts of the image

@@在这个例子中，房子的宽度是“厚度”的两倍，在图片的分块上也是如此。@@



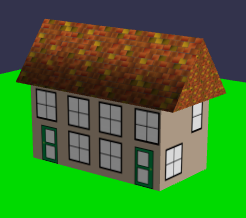
The front and back of the house (far left and right of the image) is twice the width of a side (middle image) which we can use twice.

@@房子的前面和后面（对应图片的左端和右端）宽度是中间块的两倍，并且中间块使用了两次。@@

Reusing Materials On Individual Object Faces

在物体的指定面上重用材质

https://playground.babylonjs.com/#KBS9I5#73



**Moving On**

**@@更进一步@@**

Having created our detached and semi-detached houses we would like many copies of them to form our world. We could make, separately, copies of the boxes and the roofs but it would be easier if we could combine a box and roof into one mesh, a house. Before we do this let's tidy up the code so that we can concentrate on the new code we are adding. To this end we place the existing code into functions.

@@在完成我们的单体和双体房子后，我们可能要大量复制它们来组成我们的世界。我们确实可以分别的复制盒子网格与屋顶网格，但是把盒子网格和屋顶网格合并为一个网格——房子网格，然后再复制显然会更容易。在我们做这件事之前，让我们先整理一下代码，以便我们能集中精力在之后新添加的代码上。在结尾我们将现有的代码封装为一个方法：@@

Wrapping House Building Into Functions

将建造房屋封装为一个方法

[https://playground.babylonjs.com/#KBS9I5#74](https://playground.babylonjs.com/#KBS9I5)

const createScene =  () => {

    const scene = new BABYLON.Scene(engine);

    /\*\*\*\* Set camera and light \*\*\*\*\*/

    const camera = new BABYLON.ArcRotateCamera("camera", -Math.PI / 2, Math.PI / 2.5, 10, new BABYLON.Vector3(0, 0, 0));

    camera.attachControl(canvas, true);

    const light = new BABYLON.HemisphericLight("light", new BABYLON.Vector3(1, 1, 0));

    const ground = buildGround();

    const box = buildBox();

    const roof = buildRoof();

    return scene;

}

/\*\*\*\*\*\*Build Functions\*\*\*\*\*\*\*\*\*\*\*/

const buildGround = () => {

    //color

    const groundMat = new BABYLON.StandardMaterial("groundMat");

    groundMat.diffuseColor = new BABYLON.Color3(0, 1, 0);

    const ground = BABYLON.MeshBuilder.CreateGround("ground", {width:10, height:10});

    ground.material = groundMat;

}

const buildBox = () => {

    //texture

    const boxMat = new BABYLON.StandardMaterial("boxMat");

    boxMat.diffuseTexture = new BABYLON.Texture("https://assets.babylonjs.com/environments/cubehouse.png")

    //options parameter to set different images on each side

    const faceUV = [];

    faceUV[0] = new BABYLON.Vector4(0.5, 0.0, 0.75, 1.0); //rear face

    faceUV[1] = new BABYLON.Vector4(0.0, 0.0, 0.25, 1.0); //front face

    faceUV[2] = new BABYLON.Vector4(0.25, 0, 0.5, 1.0); //right side

    faceUV[3] = new BABYLON.Vector4(0.75, 0, 1.0, 1.0); //left side

    // top 4 and bottom 5 not seen so not set

    /\*\*\*\* World Objects \*\*\*\*\*/

    const box = BABYLON.MeshBuilder.CreateBox("box", {faceUV: faceUV, wrap: true});

    box.material = boxMat;

    box.position.y = 0.5;

    return box;

}

const buildRoof = () => {

    //texture

    const roofMat = new BABYLON.StandardMaterial("roofMat");

    roofMat.diffuseTexture = new BABYLON.Texture("https://assets.babylonjs.com/environments/roof.jpg");

    const roof = BABYLON.MeshBuilder.CreateCylinder("roof", {diameter: 1.3, height: 1.2, tessellation: 3});

    roof.material = roofMat;

    roof.scaling.x = 0.75;

    roof.rotation.z = Math.PI / 2;

    roof.position.y = 1.22;

    return roof;

}

**Further reading 延伸阅读**

Map Materials to Individual Mesh Faces 对于单个网格面的材质图

Learn how to apply different materials to different faces of a mesh. 学习如何将不同的材质应用在一个网格的不同面上

https://doc.babylonjs.com/divingDeeper/materials/using/texturePerBoxFace