**Getting Started - Chapter 3 - Car Materials**

**@@启程-第三章-车的材质@@**

**Car Material**

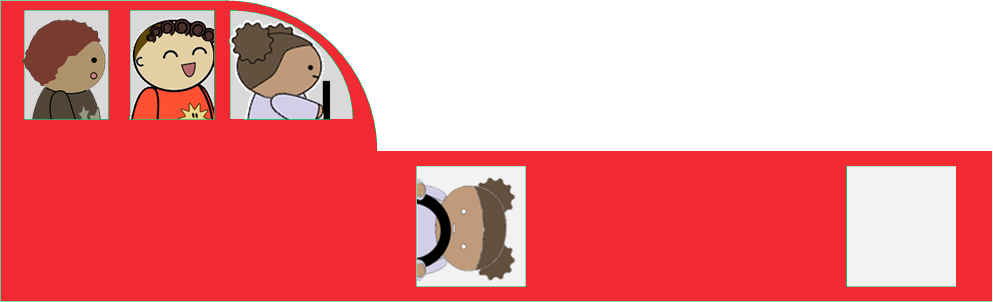
**@@车的材质@@**

Just as we used different images on different faces of a box something similar is available for extruded polygons and cylinders.

@@正如我们为盒子网格的不同面应用不同的图片，我们也可以对挤压多边形和圆柱体做类似的事。@@

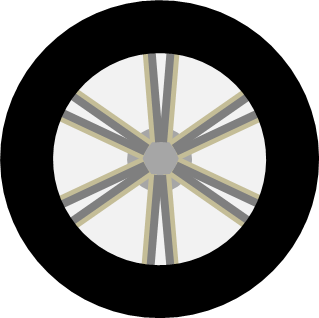
For the car body we use this image

@@对于小车我们使用这张图片@@



and this one for the wheels

@@对于轮子使用这张@@



For both the extruded polygon and the cylinder face 0, is the bottom, face 2 the top and face 1 the edge joining the bottom and top. Remember that currently the car body and its wheels are built lying down

@@对于挤压多边形和圆柱体网格，索引0对应的面是底面，2对应顶面，1对应连接底面和顶面的边缘面。请注意现在车体和它的轮子们都是躺着建造的。@@

The top and bottom of the car body use the image in the top left (almost) quarter. The edge part goes across the bottom, round the front, across the top and down the back of the body uses the lower half of the image.

@@车体的顶面和底面使用大图的左上四分之一（大概），边缘面占用大图的整个底部，使用图片下半部分绕过车头，经过车顶、车尾、车底。 @@

The bottom of the car body is as you would expect and is given by the bottom left co-ordinates (0, 0.5) to the top right ones (0.38, 1);

@@如你所料车体的底面通过左下坐标(0, 0.5) 到右上坐标(0.38, 1)给出@@

faceUV[0] = new BABYLON.Vector4(0, 0.5, 0.38, 1);

The top of the car body uses the same image but it needs flipping over to fit the other side of the car.

@@车体的顶面使用相同的图片，但它需要翻转过来，以适应另一侧的车体形状@@

faceUV[2] = new BABYLON.Vector4(0.38, 1, 0, 0.5);

The edge runs from (0, 0) to (1, 0.5)

@@边缘面从(0, 0) 到(1, 0.5)@@

faceUV[1] = new BABYLON.Vector4(0, 0, 1, 0.5);

Adding Materials To The Car

@@为小车添加材质@@

[https://playground.babylonjs.com/#KDPAQ9#12](https://playground.babylonjs.com/#12)

A wheel is more straight forward because of its symmetry and it uses the whole image for the top and bottom and just picks up a black pixel for the edges.

@@轮子网格就更直接了，因为它是对称的并且它的顶面和底面都是用整张图片，同时只需要为边缘面取一个黑色像素。@@

wheelUV[0] = new BABYLON.Vector4(0, 0, 1, 1);

wheelUV[1] = new BABYLON.Vector4(0, 0.5, 0, 0.5);

wheelUV[2] = new BABYLON.Vector4(0, 0, 1, 1);

Putting these together and rotating the finished car upright gives

@@把它们合在一起，然后将完成的小车扶正就获得了：@@



Adding Materials To The Wheels

为轮子添加材质

https://playground.babylonjs.com/#KDPAQ9#13

At last we are ready to think about animating the wheels.

@@最后我们已经准备好研究如何为轮子添加动画。@@

const createScene =  () => {

    const scene = new BABYLON.Scene(engine);

    const camera = new BABYLON.ArcRotateCamera("camera", -Math.PI / 2, Math.PI / 2.5, 3, new BABYLON.Vector3(0, 0, 0));

    camera.attachControl(canvas, true);

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

    const car = buildCar();

    car.rotation.x = -Math.PI / 2;

    return scene;

}

const buildCar = () => {

    //base

    const outline = [

        new BABYLON.Vector3(-0.3, 0, -0.1),

        new BABYLON.Vector3(0.2, 0, -0.1),

    ]

    //curved front

    for (let i = 0; i < 20; i++) {

        outline.push(new BABYLON.Vector3(0.2 \* Math.cos(i \* Math.PI / 40), 0, 0.2 \* Math.sin(i \* Math.PI / 40) - 0.1));

    }

    //top

    outline.push(new BABYLON.Vector3(0, 0, 0.1));

    outline.push(new BABYLON.Vector3(-0.3, 0, 0.1));

    //back formed automatically

    //car face UVs

    const faceUV = [];

    faceUV[0] = new BABYLON.Vector4(0, 0.5, 0.38, 1);

    faceUV[1] = new BABYLON.Vector4(0, 0, 1, 0.5);

    faceUV[2] = new BABYLON.Vector4(0.38, 1, 0, 0.5);

    //car material

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

    carMat.diffuseTexture = new BABYLON.Texture("https://assets.babylonjs.com/environments/car.png");

    const car = BABYLON.MeshBuilder.ExtrudePolygon("car", {shape: outline, depth: 0.2, faceUV: faceUV, wrap: true});

    car.material = carMat;

    //wheel face UVs

    const wheelUV = [];

    wheelUV[0] = new BABYLON.Vector4(0, 0, 1, 1);

    wheelUV[1] = new BABYLON.Vector4(0, 0.5, 0, 0.5);

    wheelUV[2] = new BABYLON.Vector4(0, 0, 1, 1);

    //car material

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

    wheelMat.diffuseTexture = new BABYLON.Texture("https://assets.babylonjs.com/environments/wheel.png");

    const wheelRB = BABYLON.MeshBuilder.CreateCylinder("wheelRB", {diameter: 0.125, height: 0.05, faceUV: wheelUV})

    wheelRB.material = wheelMat;

    wheelRB.parent = car;

    wheelRB.position.z = -0.1;

    wheelRB.position.x = -0.2;

    wheelRB.position.y = 0.035;

    wheelRF = wheelRB.clone("wheelRF");

    wheelRF.position.x = 0.1;

    wheelLB = wheelRB.clone("wheelLB");

    wheelLB.position.y = -0.2 - 0.035;

    wheelLF = wheelRF.clone("wheelLF");

    wheelLF.position.y = -0.2 - 0.035;

    return car;

}