**Getting Started - Chapter 7 - Light the Night**

**@@启程-第七章-照亮夜晚@@**

**Street Lights**

**@@路灯@@**

So far we have just used the hemispheric light for all the scenes. There are a range of lights but for the moment the only new one we will introduce is the spot light. This can be positioned anywhere and given a direction in which to shine. The spread of the light is given by an angle in radians, the larger the angle the wider the spread. The final parameter indicates how fast the light will fall away, the larger the number the less distance the light will shine. The spot light can be given a color.

@@目前我们所有的场景用的都是半球形光源。Babylon.js支持很多种光源，但是现在我们只介绍一种新的光源——锥形光源。这种光源可以被放置在任意位置，并且设置一个发光方向。光的发散效果以弧度角表示，这个属性越大则光发散的越宽。最后一个参数表示光的衰减速度，这个数字越大则光所能照到的距离越短。锥形光源还可以设置一个颜色属性。（译者注：这里是漫反射光颜色，它与网格的漫反射颜色或漫反射纹理相互作用。）@@

const lampLight = new BABYLON.SpotLight("name", position, direction, angle of spread, speed of disipation);

lampLight.diffuse = BABYLON.Color3.Yellow();

We will add a spot light to a street lamp. In order to create a lamp post we introduce another way of creating a mesh by extruding a shape along a path.

@@我们将把点光源添加到一个街灯网格上。为了建立灯柱我们将介绍另一种建立网格的方式——沿着曲线路径挤压形状。（译者注：之前的小车挤压网格，可以看作是沿着直线挤压）@@

We create the shape outline to extrude with a sequence of vector3s using points in the x, y plane only.

@@我们使用一个以三维向量为元素的队列，来建立挤压形状的轮廓，队列里的三维向量只有x、y分量起作用。@@

const lampShape = [];

for(let i = 0; i < 20; i++) {

lampShape.push(new BABYLON.Vector3(Math.cos(i \* Math.PI / 10), Math.sin(i \* Math.PI / 10), 0));

}

lampShape.push(lampShape[0]); //close shape闭合形状

We then set a path for the extrusion, again using vector3s. The path does not have to be restricted to the x, y plane it can be described using the full 3D space.

@@然后我们设置一个挤压路径，还是使用三维向量数组。这个路径并不需要限制在x、y平面内，可以用整个3维空间描述它。@@

const lampPath = [];

lampPath.push(new BABYLON.Vector3(0, 0, 0));

lampPath.push(new BABYLON.Vector3(0, 10, 0));

for(let i = 0; i < 20; i++) {

lampPath.push(new BABYLON.Vector3(1 + Math.cos(Math.PI - i \* Math.PI / 40), 10 + Math.sin(Math.PI - i \* Math.PI / 40), 0));

}

lampPath.push(new BABYLON.Vector3(3, 11, 0));

We then form the extrusion.

@@接着我们形成挤压效果@@

const lamp = BABYLON.MeshBuilder.ExtrudeShape("lamp", {cap: BABYLON.Mesh.CAP\_END, shape: lampShape, path: lampPath, scale: 0.5});

To make the lamp light more visible we turn down the intensity of the hemispheric light,

@@为了让路灯更显眼，我们调低了半球形光源的强度，@@

light.intensity = 0.5;

Create a Street Light

建立街灯

[https://playground.babylonjs.com/#4G38H4#6](https://playground.babylonjs.com/#4G38H4)

（代码：）

var createScene = function () {

    const scene = new BABYLON.Scene(engine);

    const camera = new BABYLON.ArcRotateCamera("Camera", 3 \* Math.PI / 2, Math.PI / 2.2, 50, BABYLON.Vector3.Zero(), scene);

    camera.attachControl(canvas, true);

    const light = new BABYLON.HemisphericLight("hemi", new BABYLON.Vector3(0, 50, 0));

    light.intensity = 0.5

    //add a spotlight and later after a mesh lamp post and a bulb have been created

    //then will make the post a parent to the bulb and

    //the bulb to the parent，让点光源作为灯泡的子元素、灯泡作为灯柱的子元素

    const lampLight = new BABYLON.SpotLight("lampLight", BABYLON.Vector3.Zero(), new BABYLON.Vector3(0, -1, 0), Math.PI, 1, scene);

    lampLight.diffuse = BABYLON.Color3.Yellow();

    //shape to extrude 挤压形状

    const lampShape = [];

    for(let i = 0; i < 20; i++) {

        lampShape.push(new BABYLON.Vector3(Math.cos(i \* Math.PI / 10), Math.sin(i \* Math.PI / 10), 0));

    }

    lampShape.push(lampShape[0]); //close shape

    //extrusion path 挤压路径

    const lampPath = [];

    lampPath.push(new BABYLON.Vector3(0, 0, 0));

    lampPath.push(new BABYLON.Vector3(0, 10, 0));

    for(let i = 0; i < 20; i++) {

        lampPath.push(new BABYLON.Vector3(1 + Math.cos(Math.PI - i \* Math.PI / 40), 10 + Math.sin(Math.PI - i \* Math.PI / 40), 0));

    }

    lampPath.push(new BABYLON.Vector3(3, 11, 0));

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

    yellowMat.emissiveColor = BABYLON.Color3.Yellow();//所谓发射颜色，只是使得材质能在没有光照时呈现颜色，它并不能照亮其他网格！

    //extrude lamp 挤压路灯网格

    const lamp = BABYLON.MeshBuilder.ExtrudeShape("lamp", {cap: BABYLON.Mesh.CAP\_END, shape: lampShape, path: lampPath, scale: 0.5});

    //add bulb 添加灯泡

    const bulb = BABYLON.MeshBuilder.CreateSphere("bulb", {diameterX: 1.5, diameterZ: 0.8});

    bulb.material = yellowMat;

    bulb.parent = lamp;

    bulb.position.x = 2;

    bulb.position.y = 10.5;

    lampLight.parent = bulb;

    const ground = BABYLON.MeshBuilder.CreateGround("ground", {width:50, height: 50})

    return scene;

}

We export the lamp, of appropriate size, to use it in the village. As we need more than one street light, once the lamp is loaded we will clone it several times. For each street light we will add the lamp light. Lights are normally restricted to four in any scene. Any more and only the last 4 created are active. To extend the number of lights used we set the number needed on any material to be lit.

@@为了在村庄场景里使用，我们以合适的尺寸导出路灯模型。因为我们要用到不只一个路灯，在加载路灯模型后，我们将多次克隆它。对于每个路灯，我们将添加一个锥形光源。Babylon.js场景中的光源数量通常被限制为4个，如果添加更多的光源则只有最新添加的4个生效。我们可以修改需要承受光照的材质的属性，来扩展可用光源数量。@@

material.maxSimultaneousLights = 5;

Add Street Lights

添加街灯

https://playground.babylonjs.com/#KBS9I5#94

It would be good if we could slide daylight into nightlight and vice-versa. We can when we add the Babylon.js GUI.

@@如果我们能平滑地将白天变成黑夜将会很妙，反之亦然。添加Babylon.js GUI则我们能够做到这一点。@@

const createScene = function () {

    const scene = new BABYLON.Scene(engine);

    const camera = new BABYLON.ArcRotateCamera("camera", -Math.PI / 2.2, Math.PI / 2.2, 15, new BABYLON.Vector3(0, 0, 0));

    camera.upperBetaLimit = Math.PI / 2.2;

    camera.attachControl(canvas, true);

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

    light.intensity = 0.1;

    BABYLON.SceneLoader.ImportMeshAsync("", "https://assets.babylonjs.com/meshes/", "lamp.babylon").then(() =>{

        const lampLight = new BABYLON.SpotLight("lampLight", BABYLON.Vector3.Zero(), new BABYLON.Vector3(0, -1, 0), 0.8 \* Math.PI, 0.01, scene);

        lampLight.diffuse = BABYLON.Color3.Yellow();

        lampLight.parent = scene.getMeshByName("bulb")

        const lamp = scene.getMeshByName("lamp");

        lamp.position = new BABYLON.Vector3(2, 0, 2);

        lamp.rotation = BABYLON.Vector3.Zero();

        lamp.rotation.y = -Math.PI / 4;

        lamp3 = lamp.clone("lamp3");

        lamp3.position.z = -8;

        lamp1 = lamp.clone("lamp1");

        lamp1.position.x = -8;

        lamp1.position.z = 1.2;

        lamp1.rotation.y = Math.PI / 2;

        lamp2 = lamp1.clone("lamp2");

        lamp2.position.x = -2.7;

        lamp2.position.z = 0.8;

        lamp2.rotation.y = -Math.PI / 2;

    });

    //Skybox

    const skybox = BABYLON.MeshBuilder.CreateBox("skyBox", {size:150}, scene);

      const skyboxMaterial = new BABYLON.StandardMaterial("skyBox", scene);

      skyboxMaterial.backFaceCulling = false;

      skyboxMaterial.reflectionTexture = new BABYLON.CubeTexture("textures/skybox", scene);

      skyboxMaterial.reflectionTexture.coordinatesMode = BABYLON.Texture.SKYBOX\_MODE;

      skyboxMaterial.diffuseColor = new BABYLON.Color3(0, 0, 0);

      skyboxMaterial.specularColor = new BABYLON.Color3(0, 0, 0);

      skybox.material = skyboxMaterial;

    BABYLON.SceneLoader.ImportMeshAsync("", "https://assets.babylonjs.com/meshes/", "valleyvillage.glb").then(() => {

        scene.getMeshByName("ground").material.maxSimultaneousLights = 5;

    });

    return scene;

};