**Getting Started - Chapter 5 - Skys Above**

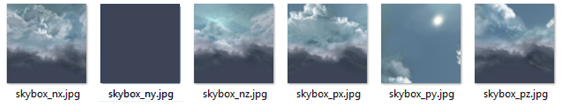
**@@启程-第五章-天空之上@@**

**Sky**

**@@天空@@**

We can simulate the appearance of a sky by applying six suitable images to the insides of a large box. Each image should have a common name followed by one of \_px, \_nx, \_py, \_ny, \_pz or \_nz, these indicate which face the image is for; the positive (p) or negative (n) x, y or z axis. For example skybox\_px.jpg, skybox\_nx.jpg, skybox\_py.jpg, skybox\_ny.jpg, skybox\_pz.jpg, skybox\_nz.jpg. These are applied to the box as a *cubeTexture*. The first parameter of a *cubeTexture* is the url to the skybox and only the common name is added.

@@我们可以通过在一个大盒子的内部应用六张合适的图片，来模拟天空的效果。每张图片都需要有一个相同的名字主体，后面跟着“\_px, \_nx, \_py, \_ny, \_pz or \_nz”中的一个后缀，这些后缀表示图片对应天空的哪个面；方向分为正向（p）、逆向两种，坐标轴分为x、y、z三种。比如你可把它们命名为“skybox\_px.jpg, skybox\_nx.jpg, skybox\_py.jpg, skybox\_ny.jpg, skybox\_pz.jpg, skybox\_nz.jpg”。它们将作为一个“立方体纹理”被应用在盒子网格上（注意与之前“为立方体的六个面设置不同纹理”的区别）。“*cubeTexture* ”方法的第一个参数是天空盒图片的url，只需要写相同的名字主体就可以了。@@



Copy the code below and link to you own skybox.

复制下面的代码，然后连接你自己的天空盒。

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

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

skyboxMaterial.backFaceCulling = false;

skyboxMaterial.reflectionTexture = new BABYLON.CubeTexture("url path/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;

In our developing world we will also add a restriction to the camera so that it cannot display anything below ground level.

@@在我们开发的世界里，我们将为相机添加一个角度限制，这样相机就不会显示地面网格以下的内容。@@

camera.upperBetaLimit = Math.PI / 2.2;

Adding a Skybox

添加一个天空盒

https://playground.babylonjs.com/#KBS9I5#88

Next environmental improvement is to grow some trees.

@@下一个环境改进是种树。@@

const createScene = function () {

    const scene = new BABYLON.Scene(engine);

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

    //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");

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

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

        car.rotation = new BABYLON.Vector3(Math.PI / 2, 0, -Math.PI / 2);

        car.position.y = 0.16;

        car.position.x = -3;

        car.position.z = 8;

        const animCar = new BABYLON.Animation("carAnimation", "position.z", 30, BABYLON.Animation.ANIMATIONTYPE\_FLOAT, BABYLON.Animation.ANIMATIONLOOPMODE\_CYCLE);

      const carKeys = [];

      carKeys.push({

        frame: 0,

        value: 10

      });

      carKeys.push({

        frame: 200,

        value: -15

      });

      animCar.setKeys(carKeys);

      car.animations = [];

      car.animations.push(animCar);

      scene.beginAnimation(car, 0, 200, true);

      //wheel animation

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

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

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

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

      scene.beginAnimation(wheelRB, 0, 30, true);

      scene.beginAnimation(wheelRF, 0, 30, true);

      scene.beginAnimation(wheelLB, 0, 30, true);

      scene.beginAnimation(wheelLF, 0, 30, true);

    });

    return scene;

};