**Getting Started - Chapter 5 - Sprite Trees**

**@@启程-第五章-精灵树@@**

**Sprite Trees**

**@@精灵树@@**

We are going to plant a couple of woods in our world each containing 500 trees. To maintain rendering speed we are going to use sprites. These are two dimensional images that will always face the camera.

@@我们将在我们的世界中种植一些树林，每片树林包含500棵树。为了保证渲染速度，我们将使用精灵。它们是2D的图片，并且将永远朝向相机。@@

We are using this image

@@我们使用这张图片@@



 for out tree sprite and we set up a sprite manager for it.

@@我们为它建立一个精灵管理器。@@

const spriteManagerTrees = new BABYLON.SpriteManager("treesManager", "url to tree image", 2000, {width: 512, height: 1024}, scene);

The parameters are a name for the manager, the url of the image, the maximum number of sprites, an object specifying the width and height of the sprite, in this case it is the width and height of the image.

@@参数分别是精灵管理器的名字，图片的url，精灵的最大数量，一个指定精灵的宽度和高度的对象，在这个例子中它们就是图片本身的宽高。@@

for (let i = 0; i < 500; i++) {

const tree = new BABYLON.Sprite("tree", spriteManagerTrees);

tree.position.x = Math.random() \* (-30);

tree.position.z = Math.random() \* 20 + 8;

tree.position.y = 0.5;

}

for (let i = 0; i < 500; i++) {

const tree = new BABYLON.Sprite("tree", spriteManagerTrees);

tree.position.x = Math.random() \* (25) + 7;

tree.position.z = Math.random() \* -35 + 8;

tree.position.y = 0.5;

}

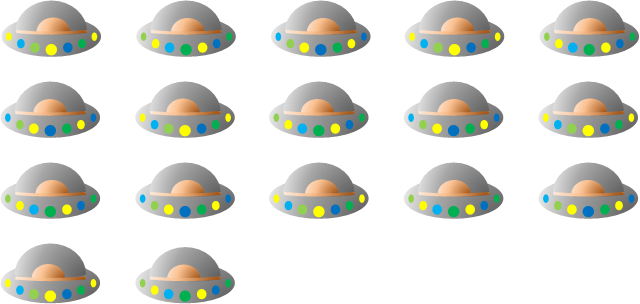
Adding Sprites

添加精灵

https://playground.babylonjs.com/#KBS9I5#89

You can also use a collection of images in a sprite map to produce an animation.

@@你也可以在一张精灵图片里使用一系列小图，来产生动画效果。@@



The above map consists of cell frames of the same size, 5 across and 4 down. This time the width and height given in the manager is the width and height of one cell.

@@上面的图由同样大小的单元块组成，5列4行。这时精灵管理器中设置的宽度和高度是一个单元块的宽高。@@

const spriteManagerUFO = new BABYLON.SpriteManager("UFOManager","url to ufo image", 1, {width: 128, height: 76});

The animation for a sprite is set by giving the first and last cell to be used, whether it loops (true) or not and time between cell frames

@@通过设定动画中出现的第一个单元块和最后一个单元块，来设置精灵的动画的帧范围，下一个参数是动画是（true）否循环，最后一个参数是每两个单元块之间的时间。@@

const ufo = new BABYLON.Sprite("ufo", spriteManagerUFO);

ufo.playAnimation(0, 16, true, 125);

Animating Sprites

精灵动画

https://playground.babylonjs.com/#KBS9I5#90

Now we are going to use some more features to produce a working fountain.

@@现在我们将使用Babylon.js引擎的更多功能来建立一个喷水的喷泉。@@

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));

    const spriteManagerUFO = new BABYLON.SpriteManager("UFOManager","https://assets.babylonjs.com/environments/ufo.png", 1, {width: 128, height: 76});

    const ufo = new BABYLON.Sprite("ufo", spriteManagerUFO);

    ufo.playAnimation(0, 16, true, 125);

    ufo.position.y = 5;

    ufo.position.z = 0;

    ufo.width = 2;

    ufo.height = 1;

    //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");

    return scene;

};