Study on the Virtual Natural Landscape Walkthrough by Using Unity 3D

Jiang Jie
Institute of Education
Jiang Xi Science and Technology
Normal University
NanChang, China
email: janyjiang@163.com

Kuang Yang
Institute of Education
Jiang Xi Science and Technology
Normal University
NanChang,China
email: ky_super@163.com

Shen Haihui
Institute of Art Design
Jiang Xi Science and Technology
Normal University
NanChang, China
email: Shenhh@163.com

***Abstract*—This paper describes the use of this 3D virtual reality engine of making natural landscape walkthrough. It explains the methods of making the sky, topography, trees and flowers, water of nature common things in Unity 3D, and expound Unity 3D software has quick and convenient advantage in making of virtual roaming, at last, it emphaise the pursuit for realistic scene is more important than the scene optimization under the condition of current level of computer.**

Keywords-virtual natural landscape; unity 3D; walkthrough

I. Introduction

Virtual walkthrough technology is a perfect tool used in architecture designing and showing. It integrates computer graphics, multimedia, artificial intelligence, multi-sensor technology, network and so on, it provides strong support for us to create a virtual world. It uses the computer to generate a simulation environment, realize the users have natural interaction with environment through multiple sensor equipment.

Unity 3D has a concept of resources store which is a good material management mode, it also has more than 250,000 customers in global world. The customers can easily use other users' Unity 3D resources to existing programs during developing. When we completed a project, also can release the project resources into the shop, sharing or paid for other developers using. The web browsing take up the space is less than 3.2 M in Unity 3D, can better adapt to the present network environment. Virtual natural landscape walkthrough making is different from construction walkthrough. Making water, smoke, trees, flowers and plants in natural environment need complicated processing and time-consuming, and easy to produce more model plane number. So it has certain difficulty to achieve good performance and running speed, but now, we can better create virtual natural landscape walkthrough based on Unity 3D.

II. Scene production

In the natural environment, such as ocean, mountains, plain and so on. These models have the features: far distance from the virtual view, no detail requirement, emphasize expression. In Unity 3D, through the edge of the terrain, closed structure composed of several polygon, surround main scenes to realize this direction vision of simulation. The sky often need the sky presents sunny, cloudy, Yin, foggy morning or evening etc. Simulating the sky through making a scene of hexahedron surrounded by "skybox"， on the surface, the texture mapping corresponding weather effects, As shown in figure 1. So, when viewpoint moves, the uses can feel strong vision. In addition, in the Unity 3D, also use "skybox" to acquire the effect of performance of the material and the water surface.

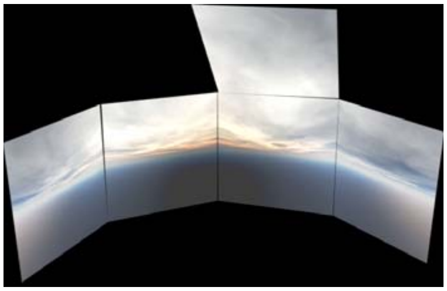

Figure 1. Skybox

III. Topography making

Early developers adopt VRML as the virtual reality software to develop virtual environment, it need to consider how to save system resources. The models' faces cannot have too much, or will influent the real-time rendering speed. Now, with the software and hardware performance improving, developers can focus on truly represent the scene when making landscape walkthrough by using Unity 3D.

Method one: using contour to make terrain

Contours refer to the closed lines curve on the terrain map have equal elevation. Vertical closed curve which linked by the same altitude points onto a standard surface, and paint on the drawing by using reduced proportion and then get contours. Contour also can be regarded as a line of different height level and practical ground, so contour is a closed curve. As shown in the left illustration of figure 2.

978-0-7695-5004-6/13 $26.00 © 2013 IEEE
DOI 10.1109/ICCIS.2013.9

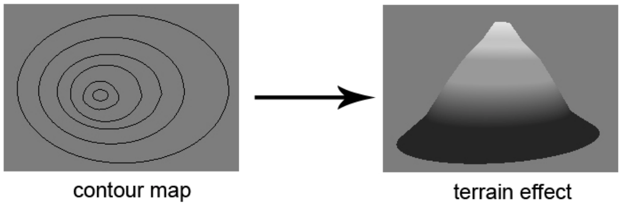

Figure 2. Contour and terrain

We can use the principle of contours to draw a different elevation contours to generate changeable terrain of natural environment when produce a rolling terrain. If designers need depict a few terrain and landforms can adopt this method, and then carry on the map making to produce a real terrain effect. As shown in the right illustration of figure 2.

Method 2: using gray facture to make terrain

For terrain production of large areas, designers can use terrain brush tool in Unity 3D. The more efficient way is import a real terrain of gray image (or a gray image in image software) into the software, then the software automatically generate large area of landform effect, as show in figure3. In the gray image, bright parts eventually generated high position, dark parts generate low-lying position, through the color of black and white as showing terrain ups and downs. In Unity 3D the gray image format is 8 bits multi-channel RAW image, so wants to change greyscale into the effect of figure 2, the image format must be convert to RAW, common image processing software can complete greyscale image format conversion.

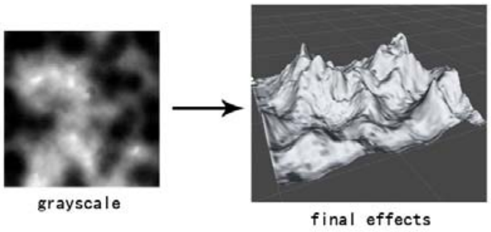

Figure3. Gray image

IV. Vegetation production:

It is quickly to decorate area vegetation. First, establish the model for the trees in 3D software, and make a good texture, then, import into Unity 3D, use the brush tool "draw" large-area vegetation on the terrain map. Additional, designers can set stroke sway the vegetation effect.

Unity 3D have special editing tools to make high precision for the plants, it can customize attributes of trunk, branches and leaves etc, editing a tree accurates to each branch and every leaf. As shown in figure 4.

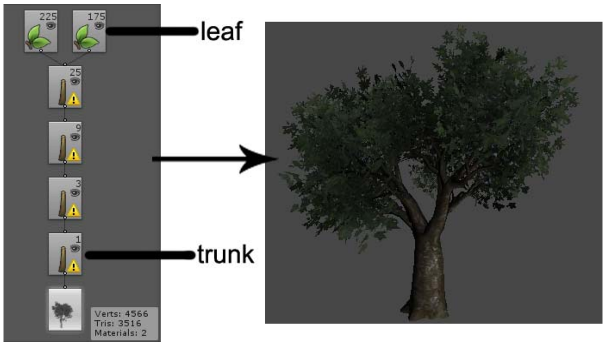

Figure 4. Editing a tree

V. Water production

A. *The water production*

Water has the characteristics of reflection, refraction and fluid. Designers combine c # program to produce realistic surface effect in Unity 3D. First, use Unity engine can produce water prototypes, then define the effect of surface is reflected or refractive, or no effect; set the texture size of water surface, set the reflection display effect of objects in the water; selected which object produce reflection or refraction effect in a scene. The code is as follows:

```
using UnityEngine;
using System.Collections;
[ExecuteInEditMode]
public class Water : MonoBehaviour
{
public enum WaterMode {
Simple = 0,
Reflective = 1,
Refractive = 2,
};
public WaterMode m_WaterMode = WaterMode.Refractive;
public int m_TextureSize = 256;
public float m_ClipPlaneOffset = 0.07f;
public LayerMask m_ReflectLayers = -1;
public LayerMask m_RefractLayers = -1;
```

The second tep is add texture mapping to water surface, the purpose is making the water surface looks more truely. Texture mapping technology has been obtained the extensive research and application. since Catmull first adopted in the mid-seventies. According to the texture of the manifestations, texture can be divided into color texture, geometric texture and process texture. Color texture refers to a variety of patterns and characters, such as the marble wall etc on surfaces of objects. Geometry texture refers to the geometry of the surface texture based on micro such as uneven texture details on the rock surface. Process texture refers to various regular or irregular dynamic changes of natural sights, such as water, cloud, etc[2].

In Unity 3D, the designers define the Shader for water and the size of the water ripple, reflection and refraction level of distortion, wave interference effect; waves bump texture, wave velocity parameters. The effect such as shown in figure 5.

```
Shader "FX/Water" {
Properties {
_WaveScale ("Wave scale", Range (0.02,0.15)) = 0.063
_ReflDistort ("Reflection distort", Range (0,1.5)) = 0.44
_RefrDistort ("Refraction distort", Range (0,1.5)) = 0.40
_RefrColor ("Refraction color", COLOR) = ( .34, .85, .92, 1)
_Fresnel ("Fresnel (A) ", 2D) = "gray" {}
_BumpMap ("Bumpmap (RGB) ", 2D) = "bump" {}
WaveSpeed ("Wave speed (map1 x,y; map2 x,y)", Vector) = (19,9,-16,-7)
_ReflectiveColor ("Reflective color (RGB) fresnel (A) ", 2D) = "" {}
_ReflectiveColorCube ("Reflective color cube (RGB) fresnel (A)", Cube) = "" { TexGen CubeReflect }
_HorizonColor ("Simple water horizon color", COLOR) = ( .172, .463, .435, 1)
_MainTex ("Fallback texture", 2D) = "" {}
_ReflectionTex ("Internal Reflection", 2D) = "" {}
_RefractionTex ("Internal Refraction", 2D) = "" {}
}
```

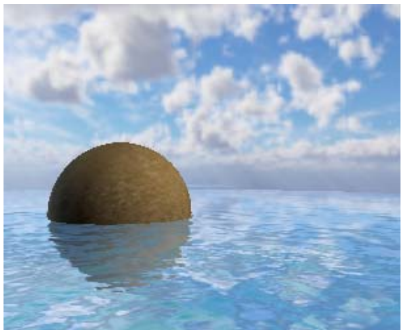

Figure 5. Water effect

B. The waterfall production

The particle system is the most suitable if we make the flame, waterfalls and other natural landscape effect. Particle system has been considered as the most successful graphics generation algorithm simulate irregular fuzzy object since 1983 [3]. The particle system was proposed by Reeves. The basic principle is describe the irregular objects through adopt many tiny particles. In the particle system, each particle has attributes concluded the shape, size, colour, transparency, speed, direction and life cycle. All of these attributes are functions of time. Each particle will experience production, development and extinct stages with time. As shown in figure 6.

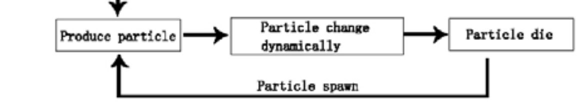
Setting the initial attribute of particle
Produce particle
Particle change dynamically
Particle die
Particle spawn

Figure 6. The circle of the particle

VI. WALKTHROUGH EFFECT PRODUCTION

It is very convenient of making walkthrough in Unity 3D. First make a cylinder avatars (probably an adult's height, also can set higher), then bind camera for body, set camera position in the position of eye, then book walkthrough perspective, finally binding statement for body:

```
var speed = 6.0;
var jumpSpeed = 8.0;
var gravity = 0;
private var moveDirection = Vector3.zero;
private var grounded : boolean = false;
function FixedUpdate() {
my=moveDirection.y;
moveDirection = new Vector3(Input.GetAxis("Horizontal"),
0, Input.GetAxis("Vertical"));
moveDirection =
transform.TransformDirection(moveDirection);
if (grounded) {
moveDirection *= speed;
} else {
moveDirection *= speed+transform.position.y/5;
moveDirection.y=my;
}
if(Input.GetKey ("e")) {
transform.Translate(0, -1, 0);
}
if (Input.GetKey ("q") ) {
transform.Translate(0, 1, 0);
}
if(Input.GetKey("left shift")) {
speed=6*5;
} else {
speed=6;
}
moveDirection.y -= gravity * Time.deltaTime;
var controller : CharacterController =
GetComponent(CharacterController);
var flags = controller.Move(moveDirection *
Time.deltaTime);
grounded = (flags & CollisionFlags.CollidedBelow) != 0;
}
@script RequireComponent(CharacterController)
private var myWalker: FPSWalker=null;
var maxHeight: float=250;
function Start () {
myWalker = gameObject.GetComponent(FPSWalker);
}
@script RequireComponent(FPSWalker)
```

When running the procedure, the user can use the the up, down, left and right key on the keyboard to control avatars how to move in the scene, use "E", "Q" key to control avatars flight in scene, use the mouse to control avatars perspective. also can pass t device control, Unity 3D has hardware control interface, it can realize the interactive walkthrough by data glovet.

VII. CONCLUSION

Now, people pay more attention on the subject of virtual reality technology, our life is changed slowly by virtual reality. This paper researchs on the virtual natural landscape walkthrough technology, realizes natural landscape walkthrough by using Unity 3D, describes some common methods of making sky, landform, trees, flowers and water in the virtual walkthrough, and put forward making virtual reality should focus on how to performant and interactive, and optimize model is second location when the computer has higher software and hardware configuration.

[1]. ChenYouPing ZhouZuDe, virtual reality and virtual manufacturing. Hubei science and technology press, 2005.

[2]. GaoZhiQing 3DSMAX large scene browse animation classic example. China water conservancy and hydropower press, 2005.

[3] ZhaoJingMi, zhang hui has, ZhengGuoQin. Based on particle system of fountain simulation. The computer application, 2006. (l) : 244 ~ 245,249