

doi: 10.14098/j.cn35-1288/z.2015.06.007

基于 Unity3D 和 Kinect 的体感跑酷游戏开发 关键技术设计与实现

张 帅,周恒杰,张琳涛

(三明学院 信息工程学院,福建 三明,365004)

摘要:针对游戏中虚拟现实游戏的趣味性,交互性,以及虚拟现实游戏的操纵性等问题,设计基于 Kinect 体感设备和 Unity3D 游戏,对体感设备捕获到的数据进行分析,使游戏中的人物的行动符合我们的预期,研究场景的实时动态加载、场景优化和 NGUI 等关键技术,以满足游戏的交互性、趣味性和游戏真实性,对今后体感游戏的开发有一定的参考价值。

关键词:Unity3D 引擎;Kinect 体感设备;体感识别;NGUI 插件

中图分类号:TP311.52

文献标志码:A

文章编号:1673-4343(2015)06-0032-05

The Research and Implementation of the Key Technology of the Somatosensory of Parkour Games Based on Unity3D and Kinect

ZHANG Shuai, ZHOU Heng-jie, ZHANG Lin-tao

(Institute of Information Engineering, Sanming University, Sanming 365004, China)

Abstract: According to the problems of the virtual reality game in the funny, interactivity and the manipulation of the virtual reality games, design game based on Kinect somatosensory equipment and Unity3D game engine, analyzed the captured data from the Kinect somatosensory equipment, making the character in the game consisted with our expectations. To meet the game's interactivity, funny and game authenticity, researched the key technologies such as the Real-time and dynamics loaded the scene, the scene optimization and NGUI and so on to meet the game's interactivity, funny and game authenticity, for the certain reference value of the future development of somatosensory game.

Key words: Unity3D; Kinect; somatosensory; recognition; NGUI

随着输入设备的发展越来越先进,体感捕捉设备渐渐映入人们的眼帘。据调查,目前以 XBox360 为平台的体感游戏(如 Kinect 大冒险和运动会、舞蹈大师、基于体感技术的切水果等)越来越丰富,玩法越来越新颖。与传统的采用输入设备(如键盘、鼠标或触屏)来操作游戏,体感游戏采用 Kinect 体感捕捉设备,玩家能够通过自己的肢体动作来控制游戏中角色的行动,感觉如置身游戏之中,增强了游戏的趣味性,且游戏中的操作易于掌控,通过一些简单上手的肢体动作来操控。在游戏设计与实现方面,有着关卡简单,目标明确,趣味性强等特点。

本文基于 Unity3D 引擎,利用 Maya 三维建模软件完成游戏场景搭建,在 Unity3D 进行脚本开发,使用 Kinect 体感设备来进行人物控制。重点研究 Kinect 设备的体感数据处理以及 Unity 中模型

收稿日期:2015-05-19

基金项目:福建省大学生创新创业训练项目(201411311061);福建省教育厅科技项目(JA13295))

作者简介:张帅,女,山西交城人,讲师。主要研究方向:虚拟现实与游戏开发。

简化、脚本驱动等关键技术,最终完成 PC 端体感游戏的发布,为今后 Kinect 体感游戏开发提供参考。

1 Kincet 设备的体感数据处理

1.1 Kinect 传感器

Kinect 骨架追踪处理流程的核心是一个无论周围环境的光照条件如何,都可以让 Kinect 感知世界的 CMOS 红外传感器。该传感器通过黑白光谱的方式来感知环境:纯黑代表无穷远,纯白代表无穷近。黑白间的灰色地带对应物体到传感器的物理距离^[1]。Kinect 负责遍历图像中较可能是人体的移动物体,类似于人眼下意识地聚焦在移动物体上。然后 Kinect 会对景深图像进行像素级评估,来辨别人体的不同部位。同时,这一过程必须以优化的预处理来缩短响应时间。Kinect 采用分割策略来将人体从背景环境中区分出来,即从噪音中提取出有用信号。Kinect 可以主动追踪最多两个玩家的全身骨架,或者被动追踪最多四名玩家的形体和位置。在这一阶段,我们为每个被追踪的玩家在景深图像中创建了所谓的分割遮罩,这是一种将背景物体(比如椅子和宠物等)剔除后的景深图像。在后面的处理流程中仅仅传送分割遮罩的部分,以减轻体感计算量。

1.2 体感设备的数据处理

在 Unity3D 中,接入 Kinect 显示的是一个三维的骨架。Kinect 能识别全身 21 个点,包括头部,腰部,左手,右手、左脚、右脚并对其进行细分,例如,左手的部分就包括左肩部,左手肘部,左手腕处等^[2]。如图 1。

在 Unity3D 中给每一个断点创建一个 Sphere,并重命名成部位的名称,并给每个部位打上标签(Tag),方便根据标签(Tag)进行场景中物体的查找。

```
Head = GameObject.FindWithTag("HEAD");
```

```
//获取断点信息,通过 Tag
```

在系统中还应处理三维端点信息来判定人物的动作。例如,需要判断人物是否进行跳跃,可以假设如果头部骨架的坐标超过了一个事先设定好的值的时候,就执行跳跃操作。

```
float headHigh = 1.6f;//设置一个最高点的值
```

```
if (Head.transform.localPosition.y > headHigh && isJump){
```

```
....
```

```
}//满足条件执行跳跃功能
```

如果要识别一个身体向左或者向右扭动的动作时,就需要计算腰部断点和肩部端点的夹角。将腰部端点和肩部端点的正切值计算出来,在进行反三角,即可得到其弧度,在进行弧度的判断,就可以得知身体是否扭动。

```
KL=(Shoulder_left.transform.position.y-Hip_center.transform.position.y)/(Shoulder_left.transform.position.x - Hip_center.transform.position.x);
```

```
KL = Mathf.Atan(KL);
```

```
if (KL >= 0.8f && KL <) { //判断夹角是否大于 0.8
```

```
.....
```

```
}
```

在 Unity3D 中只需要具体研究身体部位上每个端点是如何移动的,从中分析其数学关系,通过简单的数学计算,便可得到我们需要的结果。

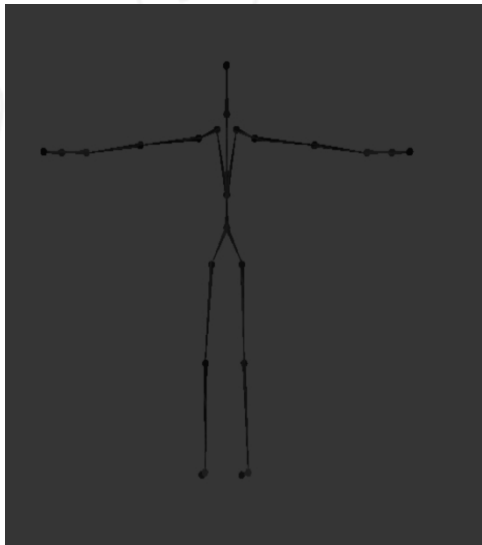


图1 Kinect 骨架

1.3 手掌模拟鼠标技术

在体感游戏中,一切的操作都应该用身体动作来实现。因此,在开始界面的控制上,通过手掌模拟鼠标来进行悬空操作,进行按钮的选择。

```
rightHand = GameObject.FindGameObjectWithTag("HAND_RIGHT");
```

初始化右手物体,这样,就能获取到其相关的位置信息。

```
tempX = XValue * (rightHand.transform.localPosition.x+0.3f);
```

```
tempY = YValue * ((rightHand.transform.localPosition.y-1.5f)*(-1.0f));
```

自定义一个手势识别区间,通过计算,得到手掌在屏幕上显示的位置。

```
GUI.DrawTexture (new Rect (tempX, tempY, 50, 50), texture);
```

通过自带的 GUI 相关 API 把模拟出的鼠标显示在计算机显示屏上, texture 为显示鼠标的 UI。当模拟出的鼠标在按钮上悬停的时候触发一个定时器 (time -= 0.04f; time 是事先定义好的值,当 time 等于 0 且鼠标悬停在物体上,则触发按钮,执行按钮的功能)。

2 Unity3D 跑酷游戏设计关键技术

2.1 游戏场景设计

在模型设计上,采用三维建模软件(3DMAX,MAYA 等)设计完整的 3D 模型并且导入。并且制作好规格相同可以拼接的模型。在 Unity3D 中,简单的地形和简单的基本对象(如树木,花草等)可在 Unity3D 中直接进行创建,需要特定的模型则需要通过 Maya 这类三维软件来进行创建。默认情况下 Unity3D 中的单位为 1cm,而 Maya 转 Unity3D 的比例为 0.01:1,因此,在创建模型之前,需要做好模型大小的规划操作。对于模型的面片数,控制在 32500 个之内为佳。当一个三维模型创建完毕之后,需要对其进行命名规范。

2.2 游戏场景优化

Unity3D 中,并不支持所有的材质球,只有标准材质球如 Blinn, Lambert 等材质会被 Unity3D 支持。并且贴图的分辨率必须为 2 的 N 次幂才能被 Unity3D 所接受。在游戏场景中,不断的实例化灯光和场景必然会消耗更多的 CPU 和 GPU,通过灯光烘焙技术,将灯光的效果烘焙到贴图上,就不必每次都实例化灯光消耗 GPU 的渲染计算,从而使游戏变得流畅。遮挡剔除技术在游戏优中必不可缺的一种技术^[3]。遮挡剔除的运行将通过在场景中使用一个虚拟的摄像机来创建一个物体潜在可视性状态(set)的层级。这些数据可以让每个运行时间内的摄像机来确定什么能看见什么看不见。通过这些数据,Unity 将确定只把可以看见的物体送去渲染。这将降低 Drawcalls 的数量并提供游戏的运行效率。

2.3 地图控制

在 Unity3D 中,开辟存放需要实例化模型块的数组,在这个数组中,存放的都是拼接地图的零件。拼接地图需要的技巧就是使用队列进行拼接。首先初始化一个队列,并将模型实例化,存入队列中,同时定义一个 TempPos 的变量来存下当前的位置坐标,将当前的位置坐标加上模型宽度的一半,在这个坐标下实例化下一个模型块,然后存下当前坐标,无限上述步骤。一般我们会把队列定义在有 50 个元素左右,当存满之后,只需要从队尾出队一个元素,做坐标的变换,然后再让物体入队,这样,就能重复使用这些物体,减少内存和 CPU 消耗。

```
private Queue<GameObject> Terrian;
```

通过 C# 自身的泛型容器来生成一个队列类。

```
for (int i = 0; i < TerrianNum; i++){
```

```
    StartRun();
```

```
}
```

通过一个 For 循环来进行道路的初始化。

```
temp = Instantiate(NoZhangAi, terrianPosition, thisRotation) as GameObject;
temp.transform.parent = father.transform;
temp.transform.localPosition = Vector3.zero;
temp.transform.localPosition = terrianPosition;
temp.transform.localEulerAngles = Vector3.zero;
Terrian.Enqueue(temp);
terrianPosition.z += 12.0f;
count++;
```

在道路初始化方面,通过 Instantiate 方法实例化出道路模型,同时把道路加到一个父节点上,方便进行场景的管理,设置改模型的方向和旋转为默认值,把模型压入队列中,同时改变实例化坐标为下一个点。到达一定的节点,就生成弯道的模型,同时改变从此以后所有直道模型的旋转属性。向左转时,角度加上 90° ,向右转时,角度加上 180° ,角度变化的规律取决于道路模型是创建的规格。

```
terrianR.y += 180.0f;
terrianRotation.eulerAngles = terrianR;
temp = (GameObject)Instantiate(Terrian_Turn, terrianPosition, terrianRotation);
```

道路的更新依据队列的性质,先进后出,在需要更新道路时,只需将队尾的物体调用到队首即可。代码如下:

Destroy(Terrian.Dequeue());做相应的变换之后,再将物体入队:

Terrian.Enqueue(temp);无限场景的处理就是基于队列这个结构,不断的出队入队,不断的拼接地图来达到这个效果。

2.4 NGUI 插件的使用

2.4.1 NGUI 介绍及其创建 UI

在 Unity3D 中,创建并且管理 UI 的解决法案大部分是使用 NGUI 插件。NGUI 是严格遵循 KISS 原则并用 C# 编写的 Unity(适用于专业版和免费版)插件,提供强大的 UI 系统和事件通知框架^[4]。其代码简洁,这意味着程序员可以很容易地扩展 NGUI 的功能或调节已有功能。对所有其他用户而言,则是更高的性能、更低的学习难度和更加有趣。在 NGUI 中,通过 Widget Tool 来创建所需要的 UI。

2.4.2 NGUI 图集工具

在游戏开发的过程中,往往需要很多的图片素材。随着图片素材的增多,图片渐渐的变得难以管理,而且占的空间越来越多,会产生很多的冗余。这时,使用图集工具是最好的选择。将图片打包成图集,方面管理,减少图片所占空间冗余^[9]。选择 NGUI>OPEN>ALTAS MAKER 选项,选择图集打包工具,选择相应的图片进行打包,NGUI 会自动生成相关的材质球和贴图文件,方便快捷,见图 2~3。



图 2 创建的图集材质球

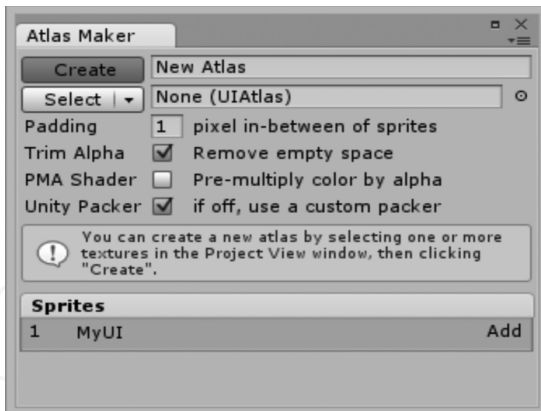


图 3 NGUI 图集打包工具

2.4.3 NGUI 事件监听机制

NGUI 的监听事件我们通过 C# 委托机制来实现监听。

```
GameObject button = GameObject.Find ("UIRoot (2D)/Camera/Anchor/Panel/LoadUI/Main-Common/Button");
```

```
UIEventListener.Get(button).onClick = ButtonClick;
```

获取需要监听的对像,通过 UIEventListener 中的代理来监听事件。上述代码中把 ButtonClick 这个方法赋予 onClick 这个事件。当 UIEventListener 接收到信息时,则会自动调用该事件。

3 游戏实现

在 Unity3D 中创建一个新的工程并导入 KinectForUnity 插件包,完成 Kinect 对 Unity3D 的支持。同时导入 NGUI 插件包,以便基于 NGUI 进行 UI 的开发。在菜单栏中选择 BuildSettings,将保存好的场景文件拖拽到对话框里,完成游戏的导出,见图 4~5。

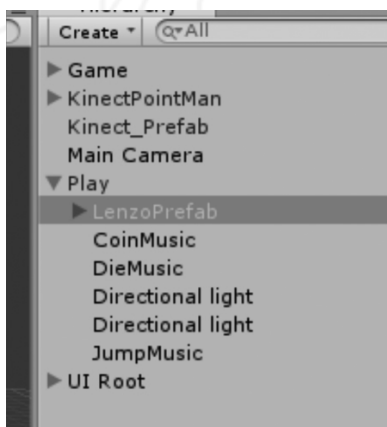


图 4 资源列表

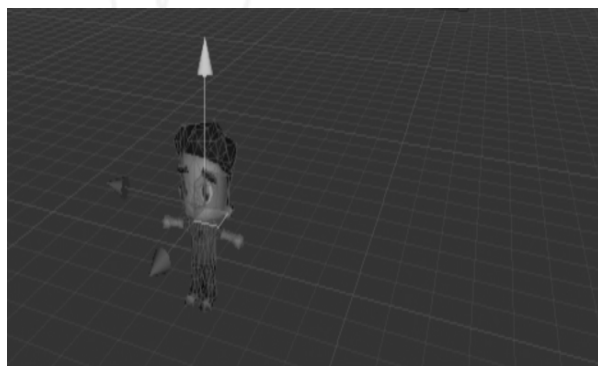


图 5 游戏编辑视窗

4 结束语

本文主要研究基于 Unity3D 的 Kinect 体感游戏开发,深入剖析体感识别的基础数学思想,巧妙利用场景自由变换技术完成跑酷游戏的基本设计思路和 NGUI 的监听事件及其用法,对体感游戏的开发具有一定参考价值。

参考文献:

- [1] MICHELLE MENARD.Game development with Unity[M].史晓明,李强,译.北京:机械工业出版社,2012.
- [2] 张帆.Unity3D 游戏开发基础[M].杭州:浙江工商大学出版社,2013.
- [3] 宣雨松.Unity3D 游戏开发[M].北京:人民邮电出版社,2012.
- [4] 刘明昆.三维游戏设计师宝典[M].成都:四川电子音像出版中心,2005.
- [5] WILLIAM R SHERMAN,ALAN B CRAIG.Understanding virtual reality: interface,application and design [M].San Francisco:Morgan Kaufmann,2002.

(责任编辑:朱联九)