

文章编号 :1003-5850(2014)11-0007-05

Kinect 与 Unity3D 数据整合技术在体感游戏中的应用研究

刘晋钢¹,刘卫斌¹,刘晋霞²

(1.太原工业学院计算机工程系,太原 030008 2.太原科技大学经济与管理学院,太原 030024)

摘 要:通过分析 Kinect 与 Unity3D 数据整合关键技术,从 WPF 与 Unity 3D 内部调用方式展开系统设计。设计分为 Unity3D 场景展示模块、Unity3D 的接口模块和 Kinect 的数据获取三模块。其中 Unity3D 接口模块实现了的场景设置、骨骼绑定、镜像运动、近景模式、平滑处理功能,Kinect 数据获取模块通过代码实现设备控制、骨骼绑定算法、设备图像获取。测试证明,通过 C# 对非托管的 dll 的管理方式,导入 Kinect 硬件的驱动程序,调用自定义的数据结构和算法,实现在 unity 3D 场景中,使用 Kinect 体感镜头控制场景中的人物模型运动,提高了体感游戏的开发效率,在体感游戏的开发和应用中有一定的社会推广价值。

关键词 Kinect,Unity3D,体感游戏

中图分类号:TP391.41

文献标识码:A

Application Research of Somatosensory Game Based on Kinect and Unity3D Data Integration Technology

LIU Jin-gang¹, LIU Wei-bin¹, LIU Jin-xia²

(1.Department of Computer Engineering, Taiyuan Institute of Technology, Taiyuan 030008, China)

(2.School of Economics and Management School of Taiyuan University of Science and Technology, Taiyuan 030024, China)

Abstract: This paper analyzes data integration technology about Kinect and Unity3D. Authors design scheme based on WPF and Unity 3D internal calling mode. System includes screen show module, Unity3D and Kinect interface module and data acquisition module. There are scene settings, rigging, mirroring sports, close-range model, smoothing processing and other functions in Unity3D; and codes implement the device control, rigging algorithm, equipment image acquisition in Kinect. Tested by C# on unmanaged dll's management, it is a good scheme to import Kinect hardware driver program and calling a custom data structures and algorithms to achieve the unity 3D scene. In the unity 3D scene, Kinect somatosensory camera control motion of models to improve the development efficiency of somatosensory game, which has certain social value in development and application of somatosensory game.

Key words: Kinect, Unity3D, somatosensory game

引 言

近年来,游戏在大众娱乐中所占的比例越来越大,市场调查发现,手机游戏占有很大部分,其次是电脑中的网络游戏和单机游戏。然而,这些游戏在创造了很多经济价值和社会价值的同时也出现很多不利

的一面,在这些游戏中,多数的设计需要游戏者沉溺其中,并投入很多时间和金钱,这在很大程度上影响了游戏者的正常生活和身体健康,所以利弊掺半的手机游戏设计和开发是很多人所反对的。随着科技进步,人机交互的发展向着多元化多功能的方向,比如语音识别、手势识别等更加接近人的自然行为会成为

* 收稿日期 2014-09-11,修回日期 2014-10-14

** 基金项目:国家自然科学基金资助项目:基于视觉认知的三维模型重建(61379080)

*** 作者简介:刘晋钢,女,1975年生,副教授,硕士,研究方向:人工智能。

未来游戏发展的方向。在微软发布了基于 Xbox 的 Kinect2.0 以后,体感也开始悄悄的成熟起来,Kinect 更多的被人们所关注,也更多的被人们所认识。Kinect 彻底颠覆了游戏的单一操作,使人机互动的理念更加彻底的展现出来^[1]。微软的 Kinect 不需要使用任何控制器,它依靠相机捕捉三维空间中玩家的运动,操作简单吸引大众,还能辨识人脸,让玩家自动连上游戏,辨认声音、接受命令。

但是,作为游戏机的 Xbox 至今无法在中国以合法的途径进入市场,这使得很多开发者放弃对于微软 Kinect for Xbox 的开发。然而,在微软 2011 年宣布, Kinect SDK For Windows,并同步销售基于 Window 的硬件时,游戏开发者已感知到一种新型的人机交互的方式将要兴起。从发布到现今,在很多展会上都出现了基于 Kinect 的体感游戏,对于消费者视觉上产生了很大的冲击。目前,在体感应用软件的开发上,主流开发工具可分为两类。第一类是建立在二维平面上的,它主要是基于微软的 WPF 和 WinForm;第二类是建立在三维立体上的,开发者最为熟悉的是 Unity3D 游戏开发工具,它的编辑器可运行在 Windows 和 Mac OS X 平台下,其开发的游戏可发布至 Windows、Mac、iPhone 和 Android 平台,也可利用 Unity web player 插件发布网页及手机游戏,支持网页浏览^[2]。

在 Kinect 应用和体感游戏的开发过程中,存在着开发效率低下的问题,故此本项目的主要工作是让 Kinect SDK 与 Unity 3D 相结合,利用体感镜头,控制人物模型运动,大幅提高游戏开发效率,降低开发成本。

1 关键技术

1.1 Kinect 与 Unity3D 集成原理

Kinect 的设计思想是将人视为控制器。在 Kinect For Windows 面世后,基于它的游戏和应用在发布会上出现,已引起人们广泛关注。人们体会到自然交互语言的强大震撼力。随着游戏的升级,高效的的游戏开发工具也成为开发者的期盼,而 Unity3D 高效率的开发模块、强大的渲染效果和可扩展的能力,使 Unity3D 赢得了开发者的亲睐,在 2012 年成为最受欢迎的游戏开发引擎,尤其在处理三维场景的编辑和交互方面,使用基于脚本的开发方式简单又方便。但是 Unity3D 这种高效开发和体感硬件 Kinect 并不兼容。因为 Kinect SDK 对开发者开放的接口调用方

式有两种,一种是基于 C++ 的头文件和静态链接库的方式,另一种是程序集的方式^[3],这两种方式在 Unity3D 中均无法直接调用。幸好,在 C# Mono 平台中提供与非托管程序的使用方式,导入 Dll 文件,用户可通过自定义数据结构和算法,能够实现对 Kinect 中获得的数据重新组织,就可在 Unity 3D 中使用。

1.2 WPF 开发流程

Kinect 是微软在 2010 年 6 月 14 日对 XBOX360 体感周边外设正式发布的名字,它是一种 3D 体感摄影机,同时导入了即时动态捕捉、影像辨识、麦克风输入、语音辨识、社群互动等功能。

Windows Presentation Foundation (WPF) 是 Microsoft 在 .NET 3.0 中推出的一个重要新特性,为 Windows 下应用程序开发提供一套全新的显示系统,旨在为用户提供方便的用户操作和震撼视觉体验的界面。WPF 支持一套完整的应用程序开发功能,包括应用程序开发模型、资源、控件、图形、动画、布局、数据绑定、文档读写、本地化、安全性等。它非常适合 Windows 平台的软件应用的开发和图形交互的展现。WPF 的核心是一个与分辨率无关的基于向量的呈现引擎,可充分发挥现代图形设备的优势,WPF 开发和 C# 集成可以通过 Visual Studio 2008 实现应用程序的开发。由于 WPF 和公共语言运行环境(CLR)的完全集成,充分利用了 CLR 提供的类型安全、跨平台等特性。WPF 的运行机制中, PresentationFramework 和 PresentationCore 都是基于 CLR 之上,以托管代码的方式公开应用程序开发接口(API)^[4]。Milcore 位于 CLR 之下,是以非托管代码方式,直接和 DirectX 紧密集成,可以利用 DirectX 在图形处理上的技术及性能优势,从而为用户提供良好的用户体验。Milcore 是 WPF 隐藏于 CLR 之下的核心驱动组件,在实际应用程序开发中不允许直接访问。

1.3 Unity3D 的内部调用方式

Unity3D 是由 Unity Technologies 开发的一个让玩家轻松创建诸如三维视频游戏、建筑可视化、实时三维动画等类型互动内容的多平台的综合型游戏开发工具,其内部调用结构如图 1 所示。

Unity 项目中的 MONO 还能够实现在 Windows 下开发应用程序,尽管无法直接调用微软的托管的 SDK,但是,它提供了调用非托管的动态链接库的接口方式,获得数据,实现调用。

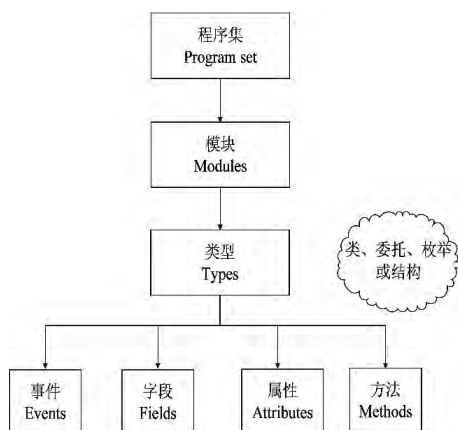


图 1 Unity 3D 内部调用结构

2 系统设计及实现

系统设计分为 Unity3D 场景展示模块、Unity3D 的接口模块及 Kinect 的数据获取 3 个模块。

2.1 场景模块

场景模块功能可实现人物模型随着体感镜头捕捉的人体来变化。模型搭建效果如图 2 所示。在场景中布置一人物,在 3DMax 中制作人物,并附着人物骨骼,在 Unity3D 中进行绑定,调节灯光和天空。



图 2 人物模型搭建效果

2.2 Unity3D 的接口模块设计

Unity3D 接口的模块为后期添脚本与 Unity3D 前台的属性相对应,实现骨骼绑定、镜像运动、近景模式、平滑处理等功能。

2.2.1 骨骼绑定

在 Kinect 中,站立模式可以跟踪 20 个关节点,其 NUI 骨骼跟踪分为主动和被动两种模式,提供最多两副完整的骨骼跟踪数据。主动模式下,调用相关帧,读取函数获得用户骨骼数据,而被动模式下,最多支持四个人的骨骼跟踪,但是在该模式下仅包含用户的位置信息,不包括详细的骨骼数据。对于所有获取

的骨骼数据,至少包含 3 类信息。第 1 类信息:相关骨骼的跟踪状态,被动模式时仅包括用户位置数据,主动模式包括完整的骨骼数据;第 2 类信息:唯一的骨骼跟踪 ID,用于分配给视野中的用户;第 3 类信息:用户质心位置,该值仅在被动模式下可用。

身体部位绑定,便于在 Unity3D 场景模型的骨骼获取数据,用数组 bones 表示场景中模型的各个骨骼节点,完成骨骼的映射功能。在骨骼变换时,采用四元素方式。四元素为简单的超复数。四元素都是由实数与 i 、 j 、 k 构成的线性组合,公式如下:

$$i^2=j^2=k^2=-1 \quad (1)$$

$$a+bi+cj+dk=0 \quad (2)$$

unity3D 中使用 Quaternion 来表达物体的旋转。基本的旋转用脚本内置旋转函数 transform.Rotate() 来实现。其功能函数及参数如下:

function Rotate (eulerAngles:Vector3, relativeTo: Space=Space.Self)

骨骼绑定部位如图 3 所示^[5],绑定后,头部代码定义方法 public Transform Head; 其他骨骼定义方式与之相同。

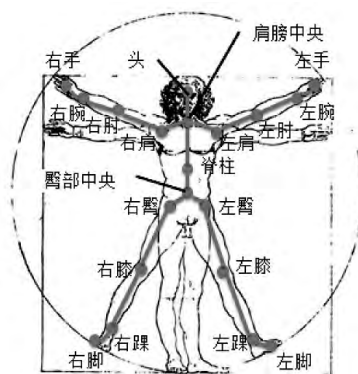


图 3 骨骼绑定 20 个关节点

在对旋转角度进行一些计算时,值得注意的是,用到四元素 Quaternion。Quaternion 的变量比较少,让 x, y, z, w 在 $[-1, 1]$ 上取值。物体旋转二周其所有数值回归初始值,初始值 $(0, 0, 0, 1)$,若沿着 y 轴旋转 180° 、 360° 、 540° 、 720° 时,分别对应的角度取值为 $(0, 1, 0, 0)$ 、 $(0, 0, 0, -1)$ 、 $(0, -1, 0, 0)$ 、 $(0, 0, 0, 1)$;若沿 x 轴旋转 180° 、 360° 、 540° 、 720° 时,分别对应的角度取值为 $(-1, 0, 0, 0)$ 、 $(0, 0, 0, -1)$ 、 $(1, 0, 0, 0)$ 、 $(0, 0, 0, 1)$ 。无旋转的写法是 Quaternion.identity。这样,就实现骨骼的绑定,自身映射和骨骼旋转。

2.2.2 镜像运动

镜像运动指一侧肢体作随意运动时,另侧肢体几

乎同时出现复制的不随意运动,在绑定人物之后模型的运动是否与实际的镜像的运动方式相符,如果不相符,则向相反方向运动。

在创建骨骼关节置换,需要指定关节索引号用来编辑不同的关节。

在骨骼映射的时候,对应的骨骼索引实现动态的变化,需要变量来判断,设置 bool 型变量 Mirrored-Movement,采用三目运算返回索引号。(Mirrored-Movement ? 12 : 1) 不同的索引号,对应不同的部位。

2.2.3 近景模式

在 Kinect for Windows SDK 1.0 以上版本中,近景模式下 Kinect 可以“坐”(seated)或“10 个关节”(10-joints)模式骨骼跟踪,只跟踪上半身的头部、颈部、肩膀、手臂等 10 个关节点。在人体骨骼绑定时,站与坐绑定关节数不同,如图 4 所示。

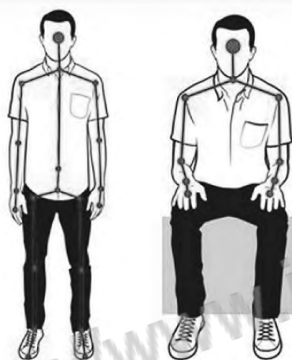


图 4 人体骨骼站与坐绑定关节数

定义“数据结构”如下:

```
public enum NuiImageStreamFlags
{
    None = 0x00000000, SuppressNoFrameData = 0x0001000,
    EnableNearMode = 0x00020000, TooFarIsNonZero = 0x00040000;
}
```

在 API 中显示调用语句如下:

```
if (isNearMode)
{
    KinectWrapper.NuiImageStreamSetImageFrameFlags (depth-
    StreamHandle, NuiImageStreamFlags.EnableNearMode);
}
```

其中 NuiImageStreamSetImageFrameFlags() 完成对于获得数据帧的类型选择,当选择近景模式的时候,设备启动近景模式,只识别近景模式下的对应骨骼,从而使得测试者可以在镜头前坐着实现绑定。

2.2.4 平滑处理

平滑处理可使得骨骼的旋转自如,防止出现跳跃和瞬移的效果,在骨骼旋转算法中实现。

首先,声明 TransformSmoothParameters 变量,为平滑算法的参数赋值,方法如下:

```
TransformSmoothParameters smoothParameters;
smoothParameters.Smoothing = 0.5f; 设置骨骼数据
帧时的平滑量,接受一个 0~1 的浮点值,值越大,平
滑的越多。0 表示不进行平滑。
```

接下来,用平滑处理,方法如下:

```
boneTransform.rotation = Quaternion.Slerp
(boneTransform.rotation, newRotation, Time.deltaTime * Smooth
Factor);
```

2.3 Kinect 数据读取模块

Kinect 数据的读取主要分为设备控制、骨骼绑定、设备图像获取 3 个部分。

2.3.1 设备控制

对设备的控制主要体现在:启动设备及初始化、相机角度调节、关闭设备。

因为在 Unity3D 内部不支持微软的 SDK,导入 Kinect 硬件的驱动程序,故对设备进行控制的代码如下:

```
[DllImportAttribute(@"Kinect10.dll", EntryPoint = "NuiIni-
tialize")]
public static extern int NuiInitialize (NuiInitializeFlags
dwFlags);
[DllImportAttribute(@"Kinect10.dll", EntryPoint = "NuiShut-
down")]
public static extern void NuiShutdown();
[DllImportAttribute(@"Kinect10.dll", EntryPoint = "NuiCam-
eraElevationSetAngle")]
public static extern int NuiCameraSetAngle (long angle);
```

2.3.2 骨骼绑定

对骨骼绑定的方法有:骨骼绑定控制、骨骼下一帧获取、骨骼平滑映射、骨骼旋转计算。对骨骼绑定代码如下:

```
[DllImportAttribute (@"Kinect10.dll", EntryPoint =
"NuiSkeletonTrackingEnable")]
public static extern int NuiSkeletonTrackingEnable (IntPtr
hNextFrameEvent, uint dwFlags);
[DllImportAttribute (@"Kinect10.dll", EntryPoint =
"NuiSkeletonGetNextFrame")]
public static extern int NuiSkeletonGetNextFrame (uint
dwMillisecondsToWait, ref NuiSkeletonFrame pSk-
eletonFrame);
[DllImportAttribute (@"Kinect10.dll", EntryPoint = "Nu-
iTransformSmooth")]
public static extern int NuiTransformSmooth (ref NuiSkeleton-
Frame pSkeletonFrame, ref NuiTransformSm-
```

oothParameters pSmoothingParams);

[DllImport(@"Kinect10.dll", EntryPoint = "NuiSkeletonCalculateBoneOrientations")]

public static extern int NuiSkeletonCalculateBoneOrientations (ref NuiSkeletonData pSkeletonData, NuiSkeletonBoneOrientation pBoneOrientations);

2.3.3 设备的图像获取

导入 Kinect 硬件的驱动程序,设备获取图像代码如下:

[DllImportAttribute(@"Kinect10.dll", EntryPoint = "NuiImageStreamOpen")]

public static extern int NuiImageStreamOpen(NuiImageType eImageType, NuiImageResolution eResolution, uint dwImageFrameFlags_NotUsed, uint dwFrameLimit, IntPtr hNextFrameEvent, ref IntPtr phStreamHandle);

[DllImportAttribute(@"Kinect10.dll", EntryPoint = "NuiImageStreamGetNextFrame")]

public static extern int NuiImageStreamGetNextFrame (IntPtr phStreamHandle, uint dwMillisecondsToWait, ref IntPtr ppclImageFrame);

[DllImportAttribute(@"Kinect10.dll", EntryPoint = "NuiImageStreamReleaseFrame")]

public static extern int NuiImageStreamReleaseFrame (IntPtr phStreamHandle, IntPtr ppclImageFrame);

[DllImportAttribute(@"Kinect10.dll", EntryPoint = "NuiImageResolutionToSize")]

public static extern int NuiImageResolutionToSize (NuiImageResolution eResolution, out uint frameWidth, out uint frameHeight);

[DllImportAttribute(@"Kinect10.dll", EntryPoint = "NuiImageStreamSetImageFrameFlags")]

public static extern int NuiImageStreamSetImageFrameFlags (IntPtr phStreamHandle, NuiImageStreamFlags dwImageFrameFlags);

上述代码中,通过 C# 对非托管的 Dll 的管理方式,获取了调用设备并获取数据的权限后,调用用户自定义数据结构和算法对数据进行处理。

3 测试

3.1 模块测试

对场景展示、接口模块、数据获取模块进行测试,从数据的一致性、Debug、误差限 3 方面给出测试结果,如表 1 所示。

3.2 手势识别测试

手势识别是通过数学算法来识别人类手势。手势识别可以来自人的身体各部位的运动,但通常情况

表 1 模块测试结果

模块	数据的一致性	Debug 显示	误差限
场景展示	数据传输	无	0%
接口模块	骨骼绑定	正常	0.1%
数据获取	彩色图像	MapBones()	1%

指脸部和手的动作。在鼠标、键盘和触控的时代,用这种手势识别技术可以让你摆脱键盘和鼠标的空间限制^[6]。Kinect 采集到的图像深度信息保存在一个二维数组中。这个数组中的第一个元素表示手势位置的深度,像素点对应的实际物体的点到 Kinect 的距离,以米为单位^[7]。可以利用 Kinect 采集的深度图像做分割,限定深度范围,将该范围内的图像提取出来^[8-9]。使用此方法实现手势识别,可识别挥手和点击两种操作。挥手时,是通过 Kinect 摄像头捕捉,点击通过调用指令完成,手势识别参数值如表 2 所示。

表 2 手势识别参数值

手势类型	测试次数	正确识别次数	识别率
挥手	1 500	1 490	99.2 %
点击	1 500	1 500	100 %

3.3 图像获取测试

彩色图像获取时,使用 Unity3D 内部纹理创建,事先设定显示大小、控制显示方式,防止图像错位,通过定义 Kinect 设备捕获的图片格式,作为传输的参数,达到手动调整和控制的目的。彩色图像获取效果如图 5 所示。



图 5 彩色图像获取效果

4 结 论

在设计场景中实现设备的启动、数据读取、骨骼绑定和彩色图像获取操作,模块间调用正常、数据读取正确、传输稳定。在骨骼绑定前,通过彩色图像确定用户的位置信息,在场景中的人物模型与检测骨骼点的方块完全对应。

(下转第 14 页)

务业务的SIM卡,才能完成数据传输的任务。SIM卡也叫用户识别卡,其中存储着用户的数据、鉴权方法及密钥,供GSM系统对用户身份进行鉴别。同时,用户通过它完成与系统的连接和信息的交换。

4 结束语

本系统完成了电路的设计并结合硬件设计原理对激光报警过程中的一些问题给出合理的实现方法。设计均采用普通元器件,若选用抗干扰能力强、稳定性好、精确度高的元器件其传输特性将会有很大改善。激光报警系统借助SIM300 GSM/GPRS模块,通过成熟可靠的GSM网络来实现远程报警功能,使用方便、经济可靠,具有批量生产及推广的现实意义,尤其适合于家居环境报警系统,具有良好的应用前景。

参考文献:

- [1] 李桂平,黄有全.基于STC12C5A的多路检测智能防盗报警器研究[J].Electronic Design Engineering 2013 21(3): 89-91.
- [2] 邱文静.基于GSM短信息的家居设施遥控监测系统设计[D].南京:南京理工大学,2012.
- [3] 邹曙光.基于GSM网络的无线显示系统设计[J].Modern Electronics Technique, 2011 34(17):34-36.
- [4] 赵冲,代茗枢.基于GSM的防盗报警系统[J].西安:西安科技大学学报 2006(增刊) 22-25.
- [5] 聂琼,张陈陈,宋博,等.一种家用防盗光电报警器的设计[J].Science & Technology Information 2013 (13):23-26.
- [6] 黄世轩,郑艺,李萍.基于GSM模块的智能防盗报警系统的设计[J].China New Telecommunications 2013(15): 8-11.
- [7] Meehan Joanne, Muir Lindsey. SCM in Merseyside SMEs: Benefits and barriers[J].TQM Journal. 2008.
- [8] Liu Lianda, Zhang You-tong, Liu Yong-Feng. ESC Testing System Based on Virtual Instrument and Distributed Network[C]// 哈尔滨 2010 IEEE 信息与自动化国际会议 (ICIA 2010), 2010.
- [9] 孙再信,迟崇巍,乔毅.基于CAN总线通信的远距离激光报警器的设计[J].ECDN 2010(11):56~58.
- [10] 李娜,牛晓飞.基于STM32的智能家居无线激光报警系统[D].宿州:宿州学院,2013.
- [11] 李琼琼,张仁杰,袁敏.基于嵌入式的无线智能家居控制装置[J].无线电研究与开发 2009 28(10):41-43.
- [12] 许凯军,林国锡,侯北平.基于GPRS的智能家居安防系统的设计[J].浙江科技学院学报 2010 22(1):27-30.
- [13] 长涛,韩中华,张楠.基于GPRS的无线智能家居控制系统方案[J].电子产品世界 2008(4):113-115.
- [14] 许志飞,姚正林.基于ARM的远程视频监控系统的的设计[J].微计算机信息 2010 26(9):105-106.

(上接第11页)

Kinect SDK存在内置的手势识别功能,设计中,根据绑定的骨骼运动规则自定义手势识别,实现挥手、点击两个动作。模块及手势识别扩展功能测试显示时,数据传输和数据类型转换均在误差允许的范围,有较高的可用性和可扩展性。使用Unity3D专业游戏引擎与Kinect数据对接准确,人机交互界面良好,在用于体感游戏开发,可降低开发成本,具有一定的社会推广价值。

参考文献:

- [1] Ross A, Clark, Yong-Hao Pua. Validity of the Microsoft Kinect for Assessment of Postural Control [J]. Gait & Posture 2012 36 (3) 372-377.
- [2] Tilak Dutta. Evaluation of the Kinect sensor for 3-D kinematic measurement in the workplace[J]. Applied Ergonomics 2011 43 (4) 645-649.
- [3] 吴国斌,李斌,阎骥洲. Kinect 人机交互开发实践[M]. 北京:人民邮电出版社 2012:45-160.
- [4] 刘铁猛.深入浅出WPF[M].北京:中国水利水电出版社, 2010:48-52.
- [5] 蔡彦. 3D游戏场景设计与底层技术研究[D].上海:上海交通大学,2008:200-310.
- [6] 希拉里 (Gastón C. Hillar). C# 并行编程高级教程:精通 NET 4 Parallel Extensions[M].北京:清华大学出版社 2012: 150-300.
- [7] 吴志达. 一个基于Unity3d游戏引擎的体感游戏研究与实现[D].广州:中山大学,2012.
- [8] 林海波,梅为林,张毅,等.基于Kinect骨骼信息的机械臂体感交互系统的设计与实现[J].计算机应用与软件, 2013(2),157-160.
- [9] 张毅,张烁,罗元,等.基于Kinect深度图像信息的手势轨迹识别及应用[J].计算机应用研究 2012(9): 3547-3550.



论文写作，论文降重，
论文格式排版，论文发表，
专业硕博团队，十年论文服务经验



SCI期刊发表，论文润色，
英文翻译，提供全流程发表支持
全程美籍资深编辑顾问贴心服务

免费论文查重：<http://free.paperyy.com>

3亿免费文献下载：<http://www.ixueshu.com>

超值论文自动降重：http://www.paperyy.com/reduce_repetition

PPT免费模版下载：<http://ppt.ixueshu.com>

阅读此文的还阅读了：

- [1. 基于Kinect体感的仿真对抗游戏系统设计](#)
- [2. 企业信用信息系统建设方案的研究](#)
- [3. 基于Kinect的体感控制技术在科普展项创新设计中的应用研究](#)
- [4. 神经影像信息数据库构建与数据处理策略探讨](#)
- [5. 浅谈邮政速递物流信息化建设](#)
- [6. 基于Kinect与3D技术的计算机编舞研究](#)
- [7. 矩阵式信号采集电路设计原理](#)
- [8. Kinect体感游戏机](#)
- [9. Unity3D与HTML交互机理的研究](#)
- [10. Kinect运动捕获技术在健康医疗中的应用研究](#)
- [11. 微学习推动互联网时代企业培训转型](#)
- [12. 基于Kinect体感交互技术的虚拟装配实验系统开发](#)
- [13. 跨平台游戏引擎Unity3D初体验](#)
- [14. Kinect体感技术在动物外科实验教学中的应用及展望](#)
- [15. Kinect体感系统 游戏迷的福音](#)
- [16. 基于Unity3D的虚拟漫游技术的研究](#)

- [17. 新一代游戏革命——小议Kinect体感游戏](#)
- [18. 基于Unity3D在基本游戏框架下客户端研究与实现](#)
- [19. 基于Unity3D的虚拟现实技术在科普活动中的应用](#)
- [20. Unity3D在教育游戏中的应用研究](#)
- [21. 信息系统集成的探索与实践](#)
- [22. 浅议房产测绘及地理信息系统的应用](#)
- [23. 稠油油田污水生物处理技术试验研究](#)
- [24. 浅谈软PLC控制技术及应用](#)
- [25. Unity3D与Flash交互方法研究](#)
- [26. 无需布线 WD Livewire Powerline影音网络套件](#)
- [27. Unity3D重要性与游戏发展趋势](#)
- [28. 宁夏有线广播电视综合网络设计与实现](#)
- [29. Web 3.0数据整合的挑战与对策](#)
- [30. 监控量测技术在长逢沟隧道施工中的应用与数据分析](#)
- [31. 改进VFP性能的几项实用技术](#)
- [32. 基于Unity3D的FPS游戏设计与开发](#)
- [33. 基于Android平台Unity3D游戏设计与实现](#)
- [34. 关于常见计算机网民加密技术的研究](#)
- [35. 耳目一新——明基精品BU201、BU202 U盘](#)
- [36. 数据恢复技术](#)
- [37. Unity3d产品虚拟展示技术的研究与应用](#)
- [38. 浅析计算机网络数字数据通信技术](#)
- [39. 基于Unity3D的多平台三维空战游戏的开发](#)
- [40. 基于Unity3d技术的三维数字校园系统研究](#)
- [41. 数据解读网坛“世纪大战”](#)
- [42. GIS在矿山中的应用探讨](#)
- [43. 浅析Unity3d开发游戏流程及常用技术](#)
- [44. 浅析计算机网络数字数据通信技术](#)
- [45. Unity3D与HTML交互机理的研究](#)
- [46. 基于Unity3d的室内漫游的关键技术研究](#)
- [47. 流动地震观测台阵数据快速整合研究](#)
- [48. 数据解读网坛 “世纪大战”](#)
- [49. 基于Kinect体感识别技术的研究与实现](#)
- [50. 体感技术何时闯入新教育?](#)