

# 基于 U3D 和 kinect1.0 月光下舞蹈互动游戏的设计与实现

王 涛<sup>1</sup>, 高贤强<sup>2</sup>

(1. 西安航空学院 教务处, 陕西 西安 710077; 2. 塔里木大学 信息工程学院, 新疆 阿拉尔 843300)

**摘 要:**传统的游戏设备缺乏躯体活动, 体感设备 kinect 的出现解决了这一问题。通过 kinect1.0 制作体感识别模块 KinectSuperMario; 通过 unity3D 中 U2D 制作“月光下舞蹈的游戏”模块 MoonLight; 通过体感识别模块发送系统键盘和鼠标消息对游戏进行控制, 阐释了“月光下舞蹈的游戏”模块功能、环境需求、游戏的开发流程、测试结果等。

**关键词:**体感设备; kinect; unity3D; 月光下舞蹈的游戏

**中图分类号:**TP311.52 **文献标识码:**A **文章编号:**1008-9233(2017)01-0068-05

## 0 引言

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<sup>[1]</sup>, 是一个全面整合的专业游戏引擎, 开发游戏简单快捷, 效果逼真。微软推出的体感设备 kinect 则打破了游戏中人和电脑交互的传统模式, 仅通过肢体语言和声音就可以实现用户的输入。

本文将两者结合开发的游戏“月光下的舞蹈”介绍给读者, 并从游戏的原理、功能、流程等方面做了阐释。

## 1 月光下的舞蹈体感游戏的设计

### 1.1 游戏的原理

该系统将海豚、鲸鱼、鲨鱼、企鹅、海豹这五种海洋动物的捕食、嬉戏、冲浪、卖萌等高清视频画面投影到墙上, 通过 kinect 1.0 传感器采集人物的各种动作, 以人的控制动作作为指令控制游戏, 互动程序采用 VS2012 C# 开发<sup>[2-3]</sup>。原理见图 1。

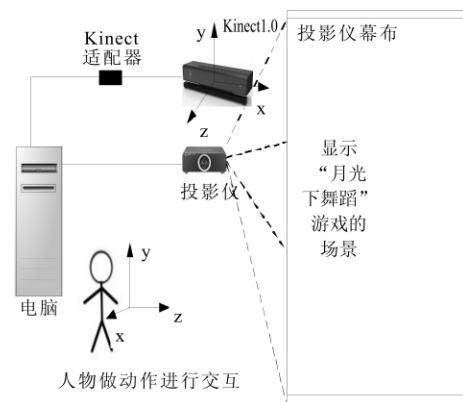


图 1 系统的布置结构图

### 1.2 游戏的功能需求

从海豚、鲸鱼、鲨鱼、企鹅、海豹五种海洋动物相关高清视频中提取它们捕食、嬉戏、冲浪、卖萌等精彩画面, 制作体感互动操作界面, 体感者可根据操作界面提示选择调取相关精彩画面。如果 30 秒内无人物被 kinect 检测到, 返回到游戏主界面, 开始随机播放各种动物动画。游戏界面设置: 月光下舞蹈之窗制作成动态界面作为游戏主界面, 二级界面分海豚、鲸鱼、鲨鱼、企鹅、海豹五个动态界面, 互动过程中由二级界面进入相关视频动态(动画帧调取)界面, 并匹配合适背景音乐。游戏交互动作的

收稿日期: 2016-12-21

作者简介: 王涛(1984—), 男, 陕西西安人, 讲师, 主要从事图形图像处理研究和 U3D 游戏开发。

设定:通过程序识别人体简单动作,如摆左手或摆右手等,实现一级、二级到视频帧序列的切换。

## 2 游戏流程的开发

启动游戏后,进入游戏主界面并开始播放动画。当 kinect 检测到有人时,游戏停止播放动画,返回游戏主界面。当人物做动作一:胸前挥动右手,游戏界

面向“左方向”切换游戏主题;当人物做动作二:胸前挥动左手,游戏界面向“右方向”切换游戏主题;当人物做动作三:抬起右手超过头顶时,游戏根据当前选择的主题随机播放相关的视频;当人物做动作四:抬起左手超过头顶时,由当前的游戏主题视频返回游戏主界面,整个游戏流程如图2所示。

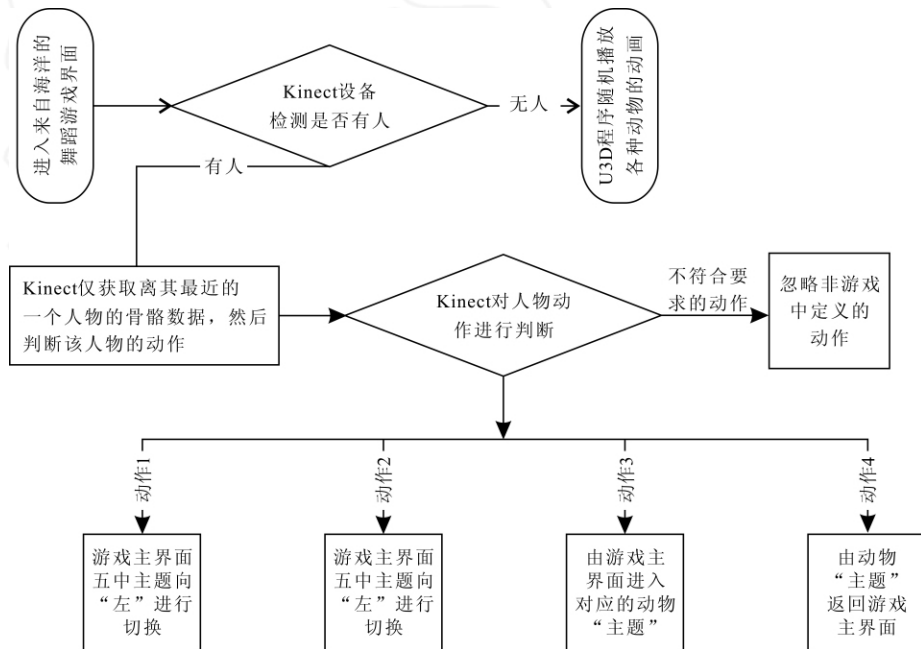


图2 游戏的流程图

## 3 游戏开发中关键技术介绍

本游戏开发分为两个部分:基于 WPF 的 kinect1.0 体感识别程序 KinectSuperMario 的开发和基于 U3D 游戏 MoonLight 的开发,两个模块通过在 KinectSuperMario 中增加键盘模拟消息进行交互控制。本文所写的 KinectSuperMario 是一个万能的模块,只需要增加新的动作识别函数和模拟键盘消息,就可以体感控制 windows 上的所有应用程序。

### 3.1 基于 WPF 的 kinect1.0 体感识别程序 KinectSuperMario

#### 3.1.1 该模块整体开发的流程

该程序模块使用 WPF 框架,整个交互系统的流程为:先完成 Kinect 的初始化工作并启动设备,然后获取骨骼和语音数据<sup>[4-5]</sup>;其次,在空间位置中持续追踪游戏玩家的动作,并根据产生的数据来判断玩家是否满足四种动作中的一

种。如果满足则模拟键盘发送对应的键盘消息,该消息会被 U3D 程序的接口捕获,并进行相应处理。

#### 3.1.2 程序之间消息接口

此处以“胸前摆左手”消息交互为例,其代码如下:

```

if(mygest == Gestures.WaveLeftHand){
    if(! isBackGestureActive && ! isForwardGestureActive){isBackGestureActive=true;
        System.Threading.Thread.Sleep(300); // 毫秒调节动作识别的频率单位毫秒
    }
}
  
```

Counttime3=0; Counttime4=0; System.Windows.Forms.SendKeys.SendWait("{A}");}}。其中函数 System.Windows.Forms.SendKeys.SendWait("{A}")为模拟人按下键盘中的“A”键,用于和 U3D 游戏模块预留“接口”进行通信来完成游戏功能。该“体感识别程序”的效果见图3。

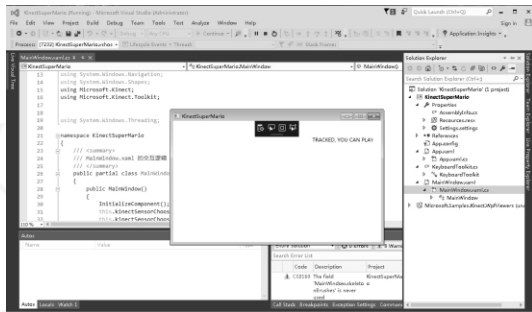


图 3 kinect 程序模块

### 3.1.3 体感识别 KinectSuperMario 中的体感识别代码

在 KinectGestures 中实现四个动作的识别,在程序中定义了四个枚举变量(isWaveRightHand, isWaveLeftHand, isRaiseRightHand 和 isRaiseLeftHand)来表示四个动作,此处以游戏中“胸前挥动右手”为例,其代码如下:

```
if(s.Joints[JointType.HandRight].Position.Y > s.Joints[JointType.ElbowRight].Position.Y){
    if(s.Joints[JointType.HandRight].Position.X > s.Joints[JointType.ElbowRight].Position.X){
        GestureStepRight = 1;
    }
    if(GestureStepRight == 1){
        if(s.Joints[JointType.HandRight].Position.Y > s.Joints[JointType.ElbowRight].Position.Y){
            if(s.Joints[JointType.HandRight].Position.X < s.Joints[JointType.ElbowRight].Position.X){
                GestureStepRight = 0;
                mygest = Gestures.WaveRightHand;
            }
        }
    }
}
```

第一个 if 嵌套的代码表示对“胸前挥动右手”的检测,第二个 if(GestureStepRight == 1) 的嵌套表示对该动作检测的确认和完成。

## 3.2 基于 U3D 的游戏程序

### 3.2.1 游戏逻辑的处理

以往游戏引擎是配合使用 C++、C# 或 JAVA 来进行开发,其在逻辑、声音、视频播放、UI 界面的处理过于复杂,且效果不好,U3D 的出现很好地解决了这个问题。本文为了简化逻辑,将游戏分为 6 个子模块(即 6 个场景),分别为:游戏主界面、海豚主题模块、鲸鱼主题模块、鲨鱼主题模块、企鹅主题模块和海豹主题模块,让各个主题模块和游戏主界面调用交互,避免了在一个 C 文件中复杂的完成整个游戏逻辑功能<sup>[6]</sup>,U3D 提供了场景切换函数为

Application.LoadLevel(“three”),该函数的意思是加载一个新的场景“three”,并销毁当前运行的场景。

### 3.2.2 游戏中视频的播放

视频播放的主函数为 void OnGUI(){GetComponent<AudioSource>().pitch = 1f; //得到音效组件,设置声音播放的频率 GUI.DrawTexture(new Rect(0, 0, Screen.width, Screen.height), movTexture[videoRand1], ScaleMode.StretchToFill);} //在纹理上绘制电影

### 3.2.3 播放动画帧

游戏主界面效果是采用播放动画帧的方式来处理,为了灵活播放图片帧,本程序中写了一个播放动画帧的类<sup>[7]</sup>:

```
class FrameAnimation {private float fps = 15f;
private Rect drawPos; private float time = 0;
public void DrawFrameAnimation(Texture[] frameTex){ //默认方式播放图片
    public void DrawFrameAnimation1(Texture[] frameTex, int start, int end) //播放指定数目的图片
    public FrameAnimation(Rect drawPos, float fps) //加参数的构造函数}
```

### 3.2.4 体感识别程序 KinectSuperMario 和 U3D 游戏程序 MoonLight 接口

为了和 kinect 体感识别程序 KinectSuperMario 进行交互,在该程序中预留了 6 个键盘接收接口,分别为:按 a 键往前播放;按 b 键往后播放;按 c 键进入某个场景;按 d 键退出某个场景;按 e 键表示有人进入;按 f 表示人物离开。接收消息函数为 Input.GetKeyDown(“a”),它的意思是接收用户的键盘输入值‘a’<sup>[8]</sup>。而 kinect 体感识别程序发送消息函数为: System.Windows.Forms.SendKeys.SendWait。

### 3.2.5 MoonLight 中每个主题模块的处理效果

对于每个主题,都精选了 5 个以上的 720P 视频片段/2 分钟,并保证主界面进入每个主题时都不会播放相同的视频(即随机播放视频),且每个视频片段都配置不同的音效,采用声音切片 AudioClip 类,播放函数为 PlayClipAtPoint<sup>[9]</sup>,该“U3D 游戏程序 MoonLight”的效果见图 4。



图 4 unity3D 程序模块

4 搭建环境所需设备和游戏测试结果

4.1 系统主要设备配置和技术参数

该系统主要设备的配置和参数:

- (1)计算机:win7 旗舰版 32 位;处理器:i5 2.5-GHz 或更快的处理器,4GB 内存以上,DX11 图形适配器,160G 硬盘或以上。
- (2)投影机:4500 流明,分辨率 1024 \* 768,视现场环境亮度而定。
- (3)游戏开发软件为 unity3D 4.6.3 及以上版本。
- (4)显卡的显存必须要 1G 以上。
- (5)第一代 Kinect for Windows v1 和 Kinect SDK 1.8。
- (6)辅材:包括 USB 2.0 信号放大器,视频信号放大器,室外屏蔽网线等。

4.2 游戏的测试结果

表 1 测试结果统计

测试项目	测试次数	成功数	成功率
动作一	100	96	97%
动作二	100	100	100%
动作三	100	99	99%
动作四	100	97	97%

\*注:对该游戏进行 24 小时测试,功能正常无宕机

4.3 游戏的相关截图



图 5 游戏主界面“海豹”主题



图 6 游戏主界面“企鹅”主题



图 7 游戏中企鹅相关视频

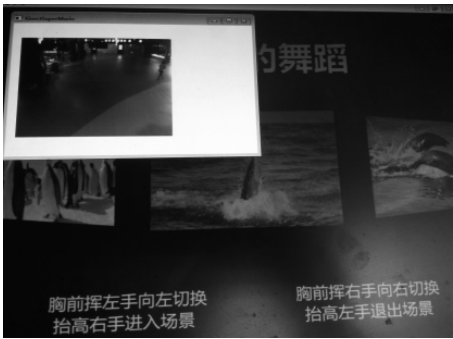


图 8 游戏现场效果图

5 游戏的部署

对于一个崭新的环境,部署如下:

- (1)首先确保 WIN7 系统所有的驱动都安装成功<sup>[10]</sup>,并安装 netframework4.5 以上版本和 KinectSDK-v1.8-Setup.exe;
- (2)安装 QuickTime\_7.79.exe 以上版本,用于 unity3d 视频播放解码;
- (3)安装 startup-delayer-v3.0b363.exe,之后把版本 moonlight.exe 和 KinectSuperMario.exe 制作为启动快捷键方式,并将 moonlight.exe 和 KinectSuperMario.exe 的快捷键方式加入到 startup-delayer-v3.0b363 中,设置 KinectSuperMario.exe 延时启动时间为 1 分 30 秒,moonlight.exe 为 2 分 10 秒;

(4)安装 Silverlight. exe 包,目的是 kinect 设备驱动自会发现。

至此,整个游戏的运行环境搭建完毕,现场的效果见图 8。

### 参 考 文 献

- [1] 王涛. 人机交互多媒体教学系统浅析[J]. 西安航空学院学报,2015,33(1):43-44.
- [2] 邓见光,袁华强. 基于游戏引擎的三维虚拟漫游系统实现[J]. 计算机应用与软件,2012,29(03):122-124.
- [3] 胡兴中,吴彬,郭雪峰,等. Unity 4. x 从入门到精通[M]. 北京:中国铁道出版社,2013:32-35.
- [4] 杜钊君. 基于体感传感器的手势识别及人机交互系统研究[D]. 武汉:武汉科技大学,2013:21-24.
- [5] 晏浩. 基于 kinect 的三维多手指跟踪算法及应用[D]. 杭州:浙江大学,2013:31-36.
- [6] 翁颖明. 卓越游戏的诞生:游戏设计内幕[M]. 北京:清华大学出版社,2010:55-60.
- [7] CREIGHTON R H. Unity3D Game Development by Example Beginner's Guide[M]. Birmingham:Packt Publishing,2010:12-43.
- [8] 宣雨松. Unity3D 游戏开发[M]. 北京:人民邮电出版社,2012:51-70.
- [9] 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,2012:38-56.
- [10] 朱惠娟. 基于 Unity3D 的虚拟漫游系统[J]. 计算机系统应用,2012,21(10):41-52.

[责任编辑、校对:东 艳]

## Design and Implementation of Interactive Moonlight Dance Game Based on the U3D and Kinect1.0

WANG Tao<sup>1</sup>, GAO Xian-qiang<sup>2</sup>

(1. Office of Academic Affairs, Xi'an Aeronautical University, Xi'an 710077, China;

2. College of Information Engineering, Tarim University, Alar 843300, China)

**Abstract:** Traditional game devices lack body movements, and the appearance kinect solves the problem. Somatosensory identification module KinectSuperMario is produced through kinect1.0, "Moonlight" produced through U2D of unity3D, and game is controlled through somatosensory identification module by sending system keyboard and mouse messages. The paper expounds the design and development flow, module function, environment demand, and test result of the interactive moonlight dance software.

**Key words:** somatosensory devices; kinect; unity3D; interactive moonlight dance