

# 基于 Unity3d 的虚拟现实家装系统的设计与实现

● 黄 玮, 余叶兰

(闽江学院 软件学院, 福建 福州 350108)

**摘 要:**在家庭装修管理系统中使用虚拟现实技术,可以使系统具有更强的用户体验并能降低客户的决策成本和装修成本。对家庭装修管理系统做了总体框架设计,通过引入射线实现物体的选取,并且对装修结果的存储进行了研究,最后给出代码的实现细节与系统的仿真结果。

**关键词:**Unity3d;虚拟现实;家装系统

**中图分类号:**TP391.9 **文献标识码:**A **文章编号:**2095-3054(2017)03-0019-04

虚拟现实技术具有高仿真性与互动性,是当前IT行业的热点之一<sup>[1]</sup>。目前虚拟现实已经在教育、工程、培训、医疗等诸多的领域得到广泛的应用<sup>[2]</sup>。

将虚拟现实应用于家装系统,可以通过家装系统高仿真地进行家庭模拟装修。这种模拟装修对于客户而言意义重大:一是客户不需要实际装修中漫长的等待,客户只要简单地选择各种装修材料并布置就可以看到装修效果;二是经济性好,相比新建样板房,提供可选择的家装模板的成本与样板房的成本相比,可以忽略不计;三是易于计算家装成本,在使用家装系统时,可自动计算相关的材料、工时等数据,便于给客户提供参考。

Unity3D 是 Unity Technologies 推出的开发引擎,在该引擎中提供了如射线、物理系统、粒子系统等在游戏领域常用的基础工具,由于其强大的功能与易上手性,目前已经广泛应用于如建筑、教育、工业设计、游戏等多个领域。本文采用 Unity3D 技术来实现家装系统的仿真设计。

## 1 家装系统的功能与整体框架

客户对家装系统有如下的功能要求:

(1) 用户管理功能。提供最基本的注册、登录等功能。

(2) 房间管理功能。提供对房型的添加、删除、查询及修改功能。

(3) 室内物品的管理功能。这些功能有如物体的移动、旋转、放置、张贴、悬挂、解绑等功能。另外,还需要提供更换物体材质与颜色的功能。

(4) 保存与加载的功能。当用户在系统上进行模拟家庭装修时,可以随时保存与加载效果,并进行

对比。

此外,用户还希望实现如留言、消息、评论等社交功能。在分析成本后,将这一部分功能移至其他应用程序中实现,此处不进行描述。

系统的整体框架采用经典的三层架构模式,即UI层、业务逻辑层与数据访问层。系统整体架构中所涉及的类较多,图1是对架构进行精简后的设计结果。

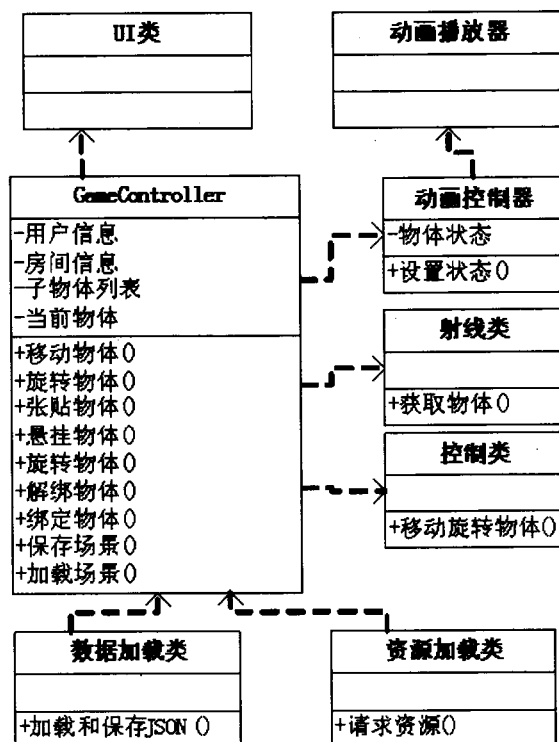


图1 系统的总体框架

收稿日期: 2016-11-19

作者简介:黄玮(1978-),男,福建福州人,讲师,硕士。

在图1中,最底层的是数据访问层,数据加载类用于获取和保存JSON格式的数据,而对于一些模型、图片等资源则使用资源加载类进行交互。

图1的中间是业务逻辑层,该类中GameController是最为核心的类,由该类控制对某个当前物体的操作,如移动旋转放置等。该类从MonoBehavior类中继承,并提供对整个场景的保存和加载。GameController中的每个子物体均将控制类作为其组件。控制类中记录了物体如位置、旋转、材质等各种状态,并对这些状态进行操作。另外还提供射线类用于选取场景中的物体。动画控制器类则用于将当前物体状态的变更与UI上的动画结合。同时GameController中的OnGUI、Update等内置方法也可用于界面交互。

## 2 系统关键问题及解决

### 2.1 物体的选取

当用户通过计算机或其他设备访问家装系统,并用输入主设备如鼠标、手柄进行物体时,后面的逻辑与在2D平面上的操作存在较大的不同。鼠标手柄在操作时得到的是在屏幕上的二维坐标。而场景中的物体用三维坐标表示,因此需要对二维坐标进行转换。另外,可以使用Unity3d中提供的射线类Ray,该类可用于从模拟人眼的Camera向场景发射一条射线。为了表示射线所遇到的物体,场景中的对象需要加上Collide组件以达到与射线碰撞的效果,这样就可以模拟出人眼“看”的效果。如果对于射线碰撞到的物体添加改变外观的效果,即可模拟出物体的获取。

### 2.2 预制件

在场景中,为了展示一个具体的物体,有许多的细节需要处理,例如材质、贴图、脚本处理逻辑、动画等。以场景中的人物为例,首先需要从3D Max中导出fbx格式的人物模型并加入场景中,并在场景中对其设置材质与贴图。此处,还需要设置动画控制器与动画。另外,还需要对应的控制代码作为组件加入到人物模型中。可见,一个人物模型是由诸多元素组成的。但对于人物的使用者而言,只需关注在场景中人物的位移速度等,无需关注人物的实现细节。为了便于开发,该人物及相关元素可被打包为一个整体,供后续使用者直接使用。Unity3d中提供了预制件的制作方法。

### 2.3 坐标变换

在使用鼠标等输入设备对物体进行操作时,需要进行物体坐标系的变换。在Unity3d中有多种坐标系,如世界坐标、本地坐标、屏幕坐标、视图坐标等。世界坐标是全局的坐标系,对应现实世界的东西南北,而本地坐标则以物体本身为基准,对应现实

世界中的前后左右。屏幕坐标则是2D的坐标,在进行鼠标、射线操作时常需要对屏幕坐标的计算。视图坐标则是场景中摄像机输出的Viewport中的坐标。

由于坐标系的不同,在对场景中物体进行如位移、旋转时需要进行不同坐标系的换算。

### 2.4 结果存储

虚拟装修后,需要保存结果。如果保存的是整个场景,由于场景中存在大量物体,且每个物体所包含的材质、模型等细节数据较多,造成存储量较大。在早期开发版本中,所要保存的模型的文件大小至少在30M以上,大的达到500M,以场景为出发点进行保存不利于保存与还原,因此拟采用Json进行关键信息的存储。

JSON<sup>[3]</sup>是一个轻量级的数据组织格式,与XML相比,具有更强的灵活性。存储的数据长度也较XML少,有利于减少数据传送与处理的系统开销。

json对数组和对象采用两种不同的形式<sup>[4]</sup>,数组采用括号“[]”括起来的内容,格式如["java","javascript","vb",...],取值时采用索引进行读取。而对象则采用{}进行内容的组织,格式采用键值对的形式,如{key:value,key:value,...}

Unity3d提供了对JSON的支持。

## 3 系统实现

### 3.1 物体选取功能的实现

场景中物体的选取需要用到射线和碰撞检测。在Unity3D中可以使用RaycastHit类来实现。当使用Camera向屏幕发出一条射线后,如果遇到装有碰撞器组件的物体,则可从RaycastHit的属性中获得所碰撞的物体。示例代码如下:

```
RaycastHit hit;
Vector3 pos = Input.GetTouch(0).position
Ray ray = Camera.main.ScreenPointToRay
(pos);
if (Physics.Raycast(ray, out hit)) {
    return hit.transform;
}
else return null;
```

在上面的代码中,如果射线在手指触碰的屏幕位置发出一条射线,遇到碰撞体后返回该物体对象,否则返回null,从而完成了物体的选取。

### 3.2 预制组件的实现

预制组件体现了软件工程中逐步求精和封装的思想。大量采用预制组件的方式设计可以大幅提高软件的质量和开发效率。由于家装系统中涉及的电器、家具、书籍、墙纸、地板等各元素,无法全部事先制作,在实际中也需要不断补充,因此可以将这些物品以预制件的形式提供。以下以坐标系预制件的创

建为例,介绍预制组件的一般逻辑。

在选中物体后往往需要能拖动对应的物体。比较直观的是当选中物体后出现如图 2 所示的坐标轴。选中坐标轴后可沿相应方向拖动物体。

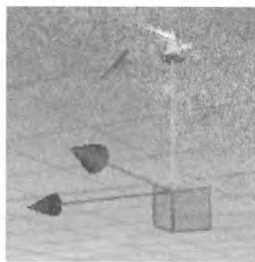


图 2 坐标轴组件

为了实现坐标轴的效果,需要完成以下的功作:

Step 1: 从 3dMax 中导出 flx 格式的圆锥体。

Step 2: 在场景中添加 Cylinder 组件,调整相关属性,使其成为细长外圆柱。

Step 3: 组合圆锥和圆柱,使之成为单个坐标轴。

Step 4: 为锥体添加脚本 Cone.cs,该脚本有两功能,当鼠标在锥体上时,将某材质改为黄色材料,鼠标移开后恢复。另外能用鼠标拖动坐标轴。

Step 5: 复制并旋转坐标轴,使之组成如图 2 所示的 X、Y、Z 三个方向。并更改其他坐标轴的材质。使得不同坐标轴呈现不同颜色。

Step 6: 提供 ConeManager 类,当选中某物体时,新创建坐标轴的副本,并将该副本作为子物体加到所选中的目标上。该脚本还提供一个功能,即当移动坐标轴时,也移动所选中的目标。

当完成上述工作后,可直接在 Unity3d 开发场景中将坐标轴拖到 Assert 中,Assert 中就出现了打包好的预制体,该预体可以被拖放到其他场景中共用。除了三维坐标外,系统中的其他元素,如各种家具、电器等,均可以被事先制作成预制件,然后根据需要再在场景中加载。

### 3.3 坐标变换的实现

客户在场景中进行家装活动时,需要进行各种家具、物品的摆放。这时需要对物品进行坐标变换。以下以移动物体为例,进行坐标变换的介绍。

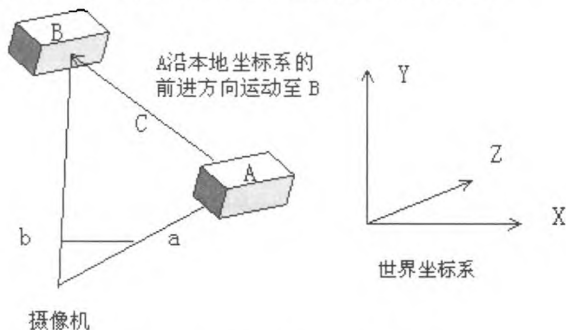


图 3 物体移动时的坐标系

图 3 描述了移动物体的过程。在图 3 的右边,标出场景中世界坐标系的各方向,而左边则是使用输入设备移动物体的实现过程。摄像机用于代表人的眼睛,而 a、b 两点则为鼠标拖物体时点击与放开的位置,线段 ab,则为鼠标拖动的距离。现在要将物体从 A 移到 B 位置,则需要求出 B 在世界坐标系中的位置。要完成上述工作,需要进行如下步骤:

Step 1: 根据屏幕坐标 a、b,求出它们在世界坐标系中的位置。两者相减,求出世界坐标系中的向量 ab。

Step 2: 将物体本地坐标系的前进方向向量转换为世界坐标系中的单位向量 m。

Step 3: 求出 ab 在 m 上的投影,得向量 p。

Step 4: A 物体的世界坐标位置加上向量 p,即得到 B 的位置。

以上过程代码实现如下:

```
Vector3 screenSpace = mCamera.WorldToScreenPoint (transform.position);
float x = Input.mousePosition.x;
float y = Input.mousePosition.y;
Vector3 vector = new Vector3 (x, y, screenSpace.z);
Vector3 startPos = mCamera.ScreenToWorldPoint (vector);
while (Input.GetMouseButton (0))
{
    Vector3 curScreenSpace = new Vector3 (Input.mousePosition.x, Input.mousePosition.y, screenSpace.z);
    //求出偏移量
    var offset = mCamera.ScreenToWorldPoint (curScreenSpace) - startPos;
    //将两点间的向量投影到物体的 UP 方向上。
    offset = Vector3.Project (offset, GetMoveDirection ());
    //对目标位置加上一个偏移量,得到新位置
    transform.position += offset;
    //重置开始点,为下一帧做准备
    startPos = transform.position;
    //等待下一帧
    yield return new WaitForFixedUpdate ();
}
```

### 3.4 结果存储的实现

首先需要进行 JSON 数据格式的设计。由于家装系统需要与远程服务器进行交互以便下载各种模

型,因此在家装系统中只需要记录模型的 ID,而不用保存所有数据。在记录 ID 的情况下,则无需保存模型的细节。

在场景中可以在模型中进行坐标系的变换,因此需要记录物体所在的三维坐标。另外,物体还可能被旋转,所以也要记录其绕各方向轴旋转的角度。

最关键的是,物体需要与其他物体保持一定的关系,如图画贴在墙上、书放置在桌上等,这种关系也要随之保存。系统中设计“悬挂”、“张贴”、“放置”、“绞合”、“倚靠”等多种方法来表示不同物体之间的关系。

以下给出一个书放置在桌上的 JSON 片段。

```
{ "ID": "S0132", "Name": "Book1", "X": "200", "Y": "300", "Z": "165", "RotateX": "0", "RotateY": "0", "RotateZ": "0", "Action": "Put", "Target": "Desk1" }
```

在该片段的最后部分,Action 与 Target 代表与何对象产生何种关系。

最终,场景中所有物体的存储可以采用以下的伪码流程实现存储。

```
for (object obj in 场景中的所有物体)
{
    if (obj 已经生成过 json) continue;
    if (obj 有关联对象)
        生成关联对象的 json;
    生成当前对象的 json;
}
```

如果要根据生成的 JSON 还原场景,则可根据物品的 ID 从本地缓存中加载。如果本地缓存中不存在,则从服务器加载,从而减少了加载时的数据量。

#### 4 仿真效果

本系统采用 Unity3d + C# + MonoDevelop 开发环境进行开发。系统中不仅包含诸如桌、床、椅子等家具,甚至包含如书、报纸、电线、开关、室内植物等细

节。通过光源组件模拟出灯光系统。在使用 VR 眼镜的情况下可通过双视角模拟出三维的场景,场景视角可跟随人物头部的旋转而旋转。在使用摇杆的情况下,可控制室内物体的摆放。VR 场景的实现采用了 Google 提供的 Cardboard SDK,在 Unity 中导入该 SDK 包后,仅使用少量的配置就能运行出双镜头的效果。实验结果表明,该系统完成了客户所需功能,达到预期的用户体验的效果。图 4 为在开发工具中呈现的仿真效果。

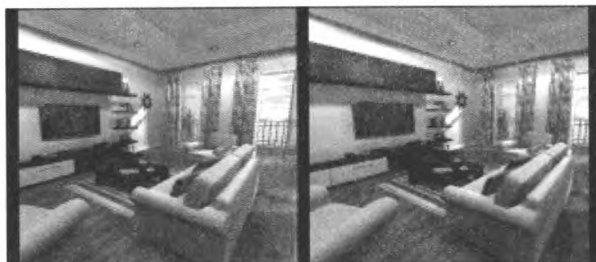


图 4 室内家装效果预览

#### 5 结语

使用 Unity3d 技术,可以高效地开发出针对家庭装修的实时仿真系统。这种仿真系统增加了用户的体验,这种体验能为用户带来装修时间与成本上的节约,并能更好地表达用户的需求。

#### 参考文献:

- [1] 黄涛. 基于 Unity3D 的虚拟校园漫游系统的研究和实现 [D]. 桂林: 广西师范大学, 2014.
- [2] 王守尊, 陈虎, 郭聪. 基于 Unity3D 的舰载机虚拟训练系统设计 [J]. 海军工程大学学报, 2016, 28(4): 92-96.
- [3] 高静, 段会川. JSON 数据传输效率研究 [J]. 计算机工程与设计, 2011, 32(7): 2267-2270.
- [4] 韩敏, 冯浩. 基于 JSON 的地理信息数据交换方法研究 [J]. 测绘科学, 2010, 35(1): 159-161.

(责任编辑:任 华)

## Design and implementation of virtual reality home improvement system based on Unity3d

HUANG Wei, YU Ye-lan

(Minjiang University, Fuzhou 350108 China)

**Abstract:** The use of virtual reality technology in the home decoration management system can make the system have a stronger user experience and reduce the customer's decision-making costs and decoration costs. In this paper, the overall framework of the family decoration management system is designed, and through the introduction of ray the selection of objects is implemented, and the storage of decoration result is studied. The implementation details of the code and the simulation results are given in the end.

**Key words:** Unity3d; virtual reality; home decoration management system