

# 数据中心三维可视化场景编辑器的设计<sup>①</sup>

王 磊<sup>1,2</sup>, 廉东本<sup>2</sup>, 王俊霖<sup>3</sup>

<sup>1</sup>(中国科学院大学, 北京 100049)

<sup>2</sup>(中国科学院沈阳计算技术研究所, 沈阳 100168)

<sup>3</sup>(大连理工大学, 大连 116024)

**摘 要:** 针对当前数据中心的三维可视化场景重复开发造成复用性较差的问题, 提出了数据中心三维可视化场景编辑器. 借助三维引擎技术, 以仿真、立体、互动实时的方式为管理者提供一个完整的、可视化、网络化的虚拟环境管理设计平台. 可以高效灵活的创建数据中心 3D 场景, 用于场景建模和数据中心的辅助设计. 可视化编辑器基于 Unity3D 引擎和 Web 技术进行搭建, 提供直观的浏览器操作界面和环境实时监控系统, 解决了三维场景远程实时查看数据中心监控等问题. 用户可以借此快速搭建三维可视化场景, 减少重复开发. 在提供其他 3D 模型基础上, 此编辑器也能够广泛用于其他非数据中心可视化场景的搭建和编辑.

**关键词:** 数据中心; 三维可视化; 场景编辑; Unity3D; 远程监控

## Design of Scene Editor for Three Dimension Visualization of Data Center

WANG Lei<sup>1,2</sup>, LIAN Dong-Ben<sup>2</sup>, WANG Jun-Lin<sup>3</sup>

<sup>1</sup>(University of Chinese Academy of Sciences, Beijing 100049, China)

<sup>2</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

<sup>3</sup>(Dalian University of Technology, Dalian 116024, China)

**Abstract:** Aiming at the problem of poor reusability of the 3D visual scene's repeated development in data center, a visualized editor system for three dimension visualization of data center is proposed. With the aid of 3D engine technology, the system provides a complete, visual and networked environment design platform for managers in the way of simulation, three-dimension and interaction. With high efficiency and flexibility, it can create 3D scene for data center for scene modeling and auxiliary design of the data center. The visual editor is structured on Unity3D engine and web technologies, which not only provides intuitive browser interface and real time environment monitoring system for users, but also solves the problem of remote real-time monitoring of data center monitoring in the way of three dimensional scenes. Users can use it to build a three-dimensional visual scene quickly, reducing duplication of development. Based on other 3D models, this system can be also widely used to build and edit scene of three dimension visualization for other non data center.

**Key words:** data center; three dimension visualization; scene editing; Unity3D; remote monitoring

随着社会的数据信息量不断的增多, 我们需要更大的数据中心处理这些数据, 这也导致数据中心具有向大规模集中式发展的趋势<sup>[1]</sup>. 数据中心计算机系统数量的扩大往往也伴随着相关的环境设备的增加, 这给数据中心的维护和管理带来了极大的不便. 在大型数据中心中往往分散着多种管理系统, 它们之间相互存在, 如何高效、统一地管理这些设备就成为企业的难题. 因此人们提出了开发三维可视化场景管理平台

为了方便对数据中心进行展示、管理、监控. 数据中心的三维可视化场景具有实时提供设备参数、提供动力环境的实时表现等特点, 满足了管理者对数据中心“集中监控, 统一管理”的需求. 而其中所用到的虚拟化技术以其交互性、想象性和沉浸式的技术特征<sup>[2]</sup>实现了将数据中心设备从现实到虚拟, 从物理到逻辑的全面对应三维可视化以及从人工被动管理到统一虚拟化管理的转变. 但目前的数据中心三维可视化场景往

① 收稿时间:2016-08-27;收到修改稿时间:2016-09-29 [doi: 10.15888/j.cnki.csa.005753]

往是针对特定数据中心场景的开发, 这造成了复用性不强、代码重复、开发难度高等问题. 因此本文设计了一个具有通用性的数据中心三维可视化场景编辑系统来快速搭建不同数据中心的三维场景.

### 1 系统概述

#### 1.1 编辑器系统结构

本编辑器采用了更加轻便的 B/S 架构, 减少了传统 C/S 模式给用户安装和升级带来的麻烦. 采用 Spring+Struts2+Hibernate 框架进行开发, 该框架可以让开发者在较短的时间内有效率地搭建结构清晰, 复用性较强的 Web 程序. 此系统分为三个层次: 浏览器端、服务器端、数据库. 如图 1 所示.

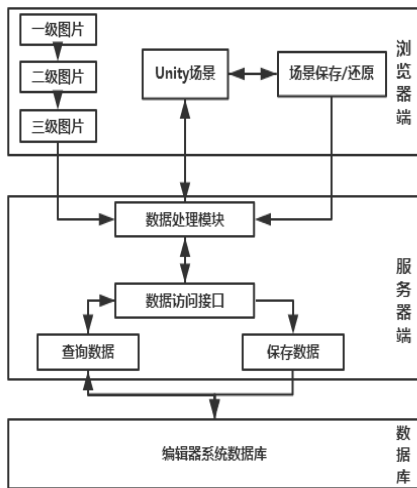


图 1 系统结构图

#### 1.2 三维可视化场景编辑器的功能介绍

该编辑器主要分四个功能部分: 系统管理, 场景管理, 基础管理, 功能管理. 如图 2 所示.

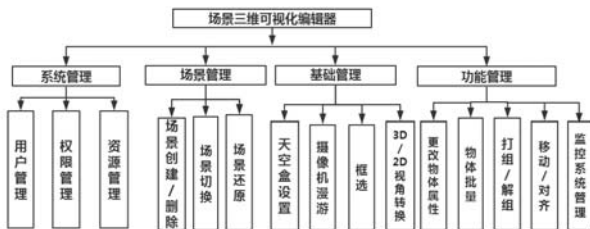


图 2 编辑器功能图

系统管理: 包含用户管理, 权限管理, 资源管理. 用户管理是分为用户和游客管理, 验证登录的合法性. 权限管理是指用户是否有权能对场景编辑操作. 资源管理是指具有权限的管理员可以对三维模型和物品

栏下的图片 UI 进行更新.

场景管理: 可以对场景的创建, 删除和切换. 同时还要对场景保存和还原.

基础管理: 该模块包含了场景基本操作. 天空盒的设置模拟真实室外场景, 摄像机漫游(非自动)可以不同角度环绕观察物体, 或者观测某一区域的场景变化. 框选则使用 OpenGL 技术能同时选中大量物体, 并对他们进行相同的操作. 3D/2D 的切换是在三维场景和二维场景相互切换, 方便用户查看整体场景布局.

功能管理: 该功能部分包括对三维数据中心物体的一些操作. 主要是 GUI 更改物体 Transform 属性, 物体批量生成, 物体打组/解组, 物体移动, 框选物体后物体按照某一方向对齐, 监控系统管理等操作.

#### 1.3 技术路线

图 3 详细的展示了 Unity3D 引擎和 JavaScript 引擎的交互、JavaScript 引擎和服务器的交互、服务器和数据库的交互等流程.

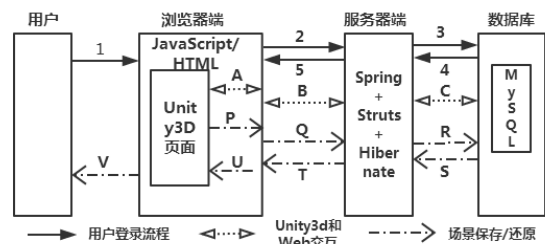


图 3 技术路线图

主要过程:

1) 1->2->3->4->5(用户登录过程): 用户在浏览器端输入用户名信息, 服务端和数据库协作验证用户登录的合法性和权限, 确认合法, 返回主页面.

2) A<->B<->C(Unity3D 和 Web 端的交互): 这个过程往往是多次相互通信完成的, 服务发起者可以是 Web 端, 也可以是 Unity3D 端. 该交互是 Unity3D 和 JavaScript 引擎的通信, 再由 Web 端 Struts2+Spring 解析通信消息, Hibernate 读取数据库, 完成交互服务.

3) U->P->Q->R(场景保存): Ajax 发出保存请求, Unity3D 引擎接收请求, 并广播消息让场景每个物体以 Json 数据格式向 JavaScript 发送自身信息, JavaScript 引擎收到消息后发送给服务器端解析 JSON 数据, 并存入 MySQL 数据库.

4) Q->R->S->T->U(场景恢复): JavaScript 向服务器端发起请求, 经 WebService 解析后, 根据场景名称读取相关场景数据, 将数据处理后发送给 Unity3D 工

具, Unity3D 使用 Resources.Load 实例化预设生成物体. 完成场景恢复.

5) Q->R->S->T->V(下载场景数据): 用户通过 Webservice 下载场景, 下载的数据格式为 Json 格式.

## 2 关键技术

数据中心三维可视化场景编辑器借助 Unity3D 进行三维可视化模型呈现. 有四个关键技术点和算法改进: 场景物体中批量算法改进的实现、场景的保存和恢复、环境实时监控系统、三维场景实时显示数据中心监控主系统的实现.

### 2.1 场景物体批量算法改进

本系统对原有物体批量算法进行了改进, 原算法只能在 X、Y、Z 中某单一方向进行批量复制, 而无法有效地在任意方向上批量生成物体. 因此本系统在原有算法基础上增加了向量来得到物体与鼠标之间的真实方向, 从而使物体能够在任意方向上批量生成. 解决了机柜不能按一定角度斜向批量摆放问题.

改进算法部分代码如下:

```
Vector3 transform_Screen = camera.WorldToScreenPoint(object.transform.position); //将物体世界坐标形成屏幕坐标
```

```
Vector2 obj = new Vector2(transform_Screen.x, transform_Screen.y);
```

```
Vector2 mouse = new Vector2(Input.mousePosition.x, Input.mousePosition.y); //获取鼠标位置
```

```
Vector2 v = (obj - mouse).normalized; //将物体坐标和鼠标坐标之间方向进行向量化
```

```
Vector3 vec = new Vector3(v.x, 0, v.y); //忽略 y 轴  
came = Camera.main.transform.rotation.eulerAngles.y;  
vec = Quaternion.Euler(0f, came - 180, 0f) * vec; //得到真实移动的方向.
```

### 2.2 场景的保存和恢复

用户在三维可视化场景编辑系统完成场景编辑后, 编辑器需要将场景数据永久保存. 在此过程中, 通过最高级别父物体向所有子物体广播信息, 将每个子物体的位置 position 参数、缩放 localScale 参数、旋转 eulerAngles 参数、名称、所属预设的名称、父体的名称(以上是三维场景保存和恢复的必要数据)均以 Json 的数据形式发送给 Web 端, 数据通过数据服务接口进行验证处理后保存到相应的数据库中.

场景保存流程如图 4 所示.

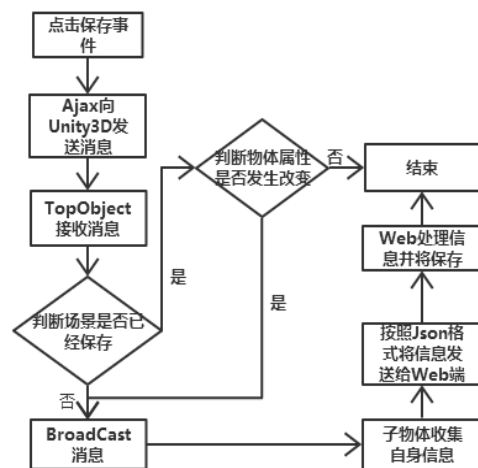


图 4 场景保存流程

保存场景后, 可以恢复场景, 并重新编辑. 恢复场景需要获得预设名字以实例化每个物体.

### 2.3 环境实时监控系统

环境监控一般是通过数据驱动实时自动生成和更新 3D 实景, 快速反映出管理对象的位置、烟感、温感等变更情况, 通过可视化的方式实时反映系统的运行状态, 提供最贴近真实情况的虚拟现实环境.

#### 2.3.1 位置信息数据驱动

对现实数据中心进行三维可视化场景的设计与编辑之后, 需要将数据中心中的机柜、摄像头等物体的现实编号、机型等信息和对应的三维场景的物体名称(必须为唯一)并存入数据库, 形成现实场景物体和虚拟场景物体一一对应的关系, 这样能够提供一个动态修改接口保证三维场景物体信息实时变化. 如当对现实中数据中心的机柜等物体进行移动时, 只需在编辑器中输入该机柜移动的水平横向距离、水平竖向距离、垂直移动距离, 根据位置转化算法将对三维场景里对应物体位置也做出相应的改变.

#### 2.3.2 烟感、温感信息数据驱动

对于烟感, 温感等实时显示, 编辑器不做任何数据采集工作, 它会通过标准接口 Webservice 服务远程读取烟感器和温感器的实时数据和其所所在位置信息, 并在通过预设数据中心上方生成一个 Plane, 添加 shader 脚本, 形成一个云图. 将传感器位置信息通过位置转化算法获取在三维可视化场景中的云图位置坐标, 并依据温度高低、烟感信息动态调整 shader 脚本里颜色 RGB 的百分比来实时支持云图的可视化渲染表现.

温感感应图如图 5 所示.

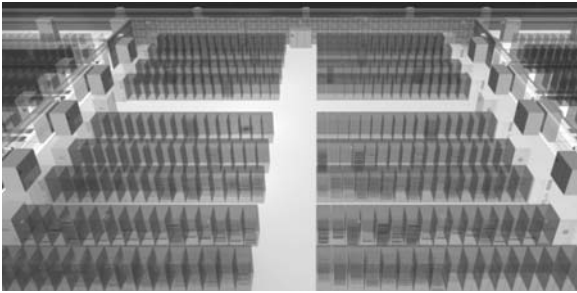


图 5 三维场景温感感应图

### 2.4 三维场景实时显示数据中心监控主系统的实现

由于 Unity3D 已经提供调用本地授权摄像头的接口, 当我们在本地运行该系统可以方便的查看本地摄像头的实时监控信息, 这里也不在过多详述.

但是对于系统需要远程运行并查看数据中心监控主系统的情况, 目前并没有相关的三维场景远程监控的公开解决办法. 为了解决这一问题, 提出了设计了基于 UDP 传输+ffmpeg 解码+合成 texture 的解决方案.

流程图如图 6 所示.

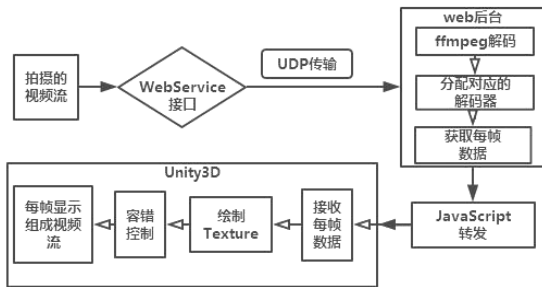


图 6 远程实时显示主控系统流程图

流程的主要难点有两个部分:

一、视频流转化为每帧数据. 本编辑器使用了 ffmpeg 进行解码. 首先穷举流格式, 找到其 VIDEO\_TYPE, 再找到对应解码器 DECODER, 最后从码流中读取 FRAME 数据.

二、Texture 的容错控制. 由于系统使用 UDP 无连接传输, 在传输时会丢失数据包, 这可能导致某个 Texture 无法绘制或者恢复顺序错误, 需要容错控制. 被恢复的视频流需要记录恢复的上一帧图像, 如某帧恢复错误, 视频流需要停留在上一帧进行等待, 直到后面某帧图像正确恢复时才继续执行播放. 对于每帧恢复顺序, 采用一个队列的 FIFO 特性来保证视频流被恢复的帧顺序不会发生改变.

### 3 编辑器运行效果

本系统借助 Unity3D 引擎实现了数据中心从物理到逻辑的三维场景可视化, 可以直观简单的看到层次化图片所对应的物体, 点击图片后向发送 Unity3D 消息, 通过射线碰撞得到鼠标位置信息并实例化物体. 借助物体上所附脚本来编辑物体, 最终通过编辑不同模型组建成一个三维可视化数据中心. 并将数据驱动的环境监控和远程显示主监控系统来达到多种数据融合到统一虚拟监控窗口, 减少数据孤岛的现象.

数据中心可视化场景编辑器运行效果如图 7.

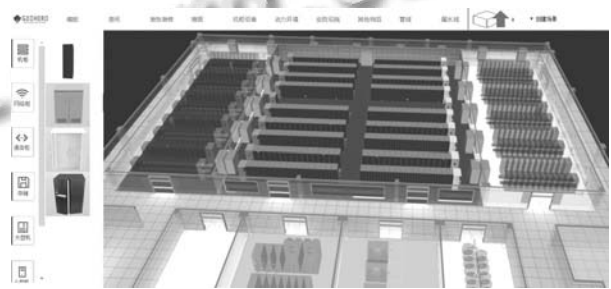


图 7 运行效果图

如果将该系统的 3D 模型替换成其他场景的模型, 同样可以使用该编辑器开发其他三维可视化场景.

### 4 结语

本系统通过提供直观的, 交互性的界面, 使用户可以快速的搭建三维可视化数据中心, 解决了可视化场景复用性较差的问题, 提高开发效率. 本文所述技术路线和功能部分并不局限适用于数据中心的可视化场景开发, 同样适用于其他场景可视化编辑系统的实现, 在三维场景开发中提供了一定的技术参考价值.

### 参考文献

- 1 龚昊,张琦.数据中心三维可视化管理软件的设计与实现 [硕士学位论文].北京:北京邮电大学,2012.
- 2 刘柱,李先毅,刘瑾.基于 UNITYWeb3D 虚拟校园的设计和实现.中国教育网络,2014,(11):77-78.
- 3 朱惠娟.基于 Unity3D 的虚拟漫游系统.计算机系统应用, 2012,21(10):36-39.
- 4 任国栋,陈林华,陶雪峰,方先行.基于Unity3D的虚拟博物馆信息三维可视化系统.计算机系统应用,2013,22(9):86-90.
- 5 张敏,陈云海.虚拟化技术在新一代云计算数据中心的应用研究.广东通信技术,2009,29(5):35-39.
- 6 Yang K, Jie J. The research of making scenic wandering system based on Unity 3D. IEEE Workshop on Electronics, Computer and Applications. 2014. 237-239.