

基于 Unity3D 的虚拟化学智能课堂系统^①



孙俊辉¹, 高 岭², 高全力², 曹 瑞¹

¹(西北大学 信息科学与技术学院, 西安 710127)

²(西安工程大学 计算机科学学院, 西安 710600)

通讯作者: 高 岭, E-mail: gl@nwu.edu.cn

摘 要: 针对传统化学教学的局限性及现有虚拟化学课堂的不足, 以 Unity3D 为基础设计并实现了虚拟化学智能课堂系统. 构建虚拟场景并导入 Unity3D 引擎, 通过 C#脚本和粒子系统开发虚拟仿真实验; 为解决多人虚拟场景下权限分配问题, 实现了采用权限策略的智能辅助模块, 为不同学生分配不同权限并提供文字视频等提示功能; 通过 Spring Cloud 框架对系统功能进行拆分, 同时使用 Redis 优化系统. 系统适用于中学化学教学, 实现了具有权限控制及高仿真的虚拟实验, 有效解决虚拟场景下权限问题, 提升学习效率, 且具有高可用性等优点.

关键词: 智能课堂; Unity3D; 权限策略; 虚拟仿真; Spring Cloud

引用格式: 孙俊辉, 高岭, 高全力, 曹瑞. 基于 Unity3D 的虚拟化学智能课堂系统. 计算机系统应用, 2021, 30(6):68-74. <http://www.c-s-a.org.cn/1003-3254/7988.html>

Virtual Chemistry Intelligent Classroom System Based on Unity3D

SUN Jun-Hui¹, GAO Ling², GAO Quan-Li², CAO Rui¹

¹(School of Information Technology, Northwest University, Xi'an 710127, China)

²(School of Computer Science, Xi'an Polytechnic University, Xi'an 710600, China)

Abstract: With regard to the limitations of traditional chemistry teaching and the deficiencies of existing virtual chemistry classrooms, a virtual chemistry intelligent classroom system based on Unity3D has been designed and implemented. A virtual scene is constructed and imported into the Unity3D engine, and virtual simulation experiments are performed through the C# script and a particle system. For authority allocation in multi-user virtual scenes, the intelligent auxiliary module with authority strategy is designed to assign different ranges of authority to students and provide prompt functions such as text and video. Besides, the system functions are split through the Spring Cloud-based framework, and the system is optimized by Redis. This system is suitable for chemistry teaching in the middle school. It implements highly-simulated virtual experiments with authority control to solve authority problems in virtual scenarios and improve learning efficiency, with high availability, etc.

Key words: intelligent classroom; Unity3D; authority strategy; virtual simulation; Spring Cloud

信息时代下, 各领域科学技术都处于迅猛发展进程中, 虚拟现实也是其中之一. 从 1963 年的萌芽阶段至今, 其不论在理论还是实践上都有质变, 应用领域也不再仅限于娱乐, 而涉及教育、医疗等诸多方面^[1,2]. 在

该背景下, 为弥补传统教学的不足, 国家开始将信息技术和教育进行融合. 学生的学习方式正发生颠覆性的变化, 在 5G 的发展和如今疫情的影响下, 虚拟教学的发展势头将更加明显^[3].

① 基金项目: 国家科技支撑计划 (2018YFB1004500); 国家自然科学基金 (61672426); 陕西省教育信息化项目 (18JX006)

Foundation item: National Sci-Tech Supporting Plan of China (2018YFB1004500); National Natural Science Foundation of China (61672426); Project for Education Informatization of Shaanxi Province (18JX006)

收稿时间: 2020-10-07; 修改时间: 2020-11-08, 2020-12-12; 采用时间: 2020-12-18; csa 在线出版时间: 2021-06-01

化学是以实验为核心的重要基础学科,但由于中学阶段授课受实验条件、安全隐患等限制,许多实验难以达到理想效果,甚至无法进行。如不进行有效实验,学生仅从书本中难以对复杂反应有深刻理解,无法结合理论与实践。但目前尚无功能成熟的虚拟化学课堂,已知系统中所实现功能大多单一、三维视觉效果差,缺乏权限控制策略及系统优化。

本文基于以上背景构建适用中学化学教学的虚拟课堂系统,开发三维视觉效果良好的虚拟化学实验,实现了采用权限策略的智能模块,根据学生情况为其分配不同权限,提供自动文字视频实验提示。此外,通过 Spring Cloud 框架拆分系统功能,将其部署于不同接口,使用 Redis 缓存优化系统。

1 系统设计

1.1 系统设计原则

根据虚拟现实的技术特性、传统课堂的教学特点及现有虚拟课堂的系统特点,将设计原则总结如下:

(1) 构建虚实结合的三维教学空间:构造符合真实世界比例的虚拟场景,注重用户体验;将二维平面的抽象内容以三维空间立体展示,提高教学趣味性^[4-6]。

(2) 充分考虑学生用户的群体特点:目标用户对事物审美要求较高,为吸引其学习兴趣,设计更符合其审美特点的用户界面,让其更好地获得强沉浸感。

(3) 最大化利用虚拟教学优势:传统课堂中教师受时间限制不能充分指导每位学生,在虚拟教学中该问题应得到充分解决;应当针对现实中难以进行的实验进行开发,充分挖掘可教学内容,丰富学生知识广度。

(4) 力求解决虚拟场景下的权限问题:同现实环境一样,虚拟多人场景下的模型交互也存在竞争。因此应提出一种应用于虚拟多人场景下的权限策略,提供合理的权限处理方法。

(5) 严格保证系统质量:应当对网络和系统故障等情况进行预处理,设计服务熔断策略,避免影响用户体验。不仅要保证系统当下的可靠性,同时要采用合理的技术架构进行开发,方便系统未来进行扩展。

1.2 系统需求分析

进行系统功能开发前,需要针对传统课堂教学的局限性及现有的虚拟化学课堂的不足进行分析并提出对应解决方案,明确系统需求。

(1) 传统化学教学存在对实验材料的条件限制,例如需要铍锂等稀有金属参与的反应在现实中难以进行。

该问题在虚拟环境中不复存在,主要难点转变为对虚拟场景中模型材料材质的高度还原。

(2) 由于传统课堂的教学时间有限,教师无法指导全部问题学生。如果学生的问题无法及时解决,课后随时间积累就将更加严重。针对该问题设计一种智能模块,学生可观看并重复点播教学视频。

(3) 传统课堂中难以进行高危实验,即使可以也必须严格遵循要求,在已知系统中也仅有进行正确操作的实验。这种情况下,实验错误进行的结果变得抽象,学生无法深入了解实验。针对该问题,考虑开发以非标准手段进行的化学实验,例如在硫酸稀释实验中增设将水倒入浓硫酸的错误对比实验。

(4) 现有虚拟化学课堂中的实验数量少,且大部分都是现实中可轻易进行的实验,没有涉及现实中受材料限制和安全考虑而难以进行的实验,严重局限虚拟现实的技术优势。针对该问题,设计开发多项现实中难以进行的化学实验,尤其是具有强烈视觉效果的实验,丰富课堂趣味性,提高学生沉浸感。

(5) 现有虚拟课堂并未涉及权限问题,未充分考虑不同学生差异,无法做到针对性教学。此外,虚拟多人环境下的角色和模型属于不可剥夺资源,无法在同一时间节点共享。为此提出一种适用虚拟教学的权限策略,对现实用户与虚拟权限进行逻辑分离,并对虚拟角色及权限类型进行细粒度划分,借助事件监听机制,提供一种合理的动态权限控制方法。

(6) 现有虚拟课堂缺乏优化,伴随项目商业化及用户数量增加,数据库冗余会越来越高。高并发情况下系统运行效率低,发生故障时难以提供正常服务,严重影响用户体验。对此提出一套保证系统效率的服务端技术架构,使用 Spring Cloud 进行开发,将 Redis 作为缓存,增强系统鲁棒性,方便系统扩展。

1.3 系统功能

根据上述设计原则及需求分析,设计如下功能:

(1) 注册登录:用户首先需要进行用户注册,注册成功后系统会自动将其信息录入后端数据库中。登录成功后,在登录界面输入用户信息进行实验选择。

(2) 角色控制:学生用户在系统中通过键盘及鼠标等输入设备对其对应的虚拟角色进行控制,能够移动、进行虚拟实验以及观看教学视频等。

(3) 虚拟实验:虚拟角色可进行的实验共包括4项,分别为金属置换反应、火焰燃烧反应、硫酸稀释反应及活泼金属与水的反应。

(4) 智能模块: 系统将根据权限策略为虚拟角色分配对应的实验权限, 具有对应权限的角色才能进行实验, 实现针对性教学; 进行虚拟实验时, 系统将自动为虚拟角色提供文字视频等提示, 帮助学生进行自主学习, 在缺乏教师指导的情况下保证教学质量. 虚拟角色可以查询对应成绩和排名, 促进良性学习竞争.

1.4 系统整体结构

图1描述了本文所提出系统的结构, 包括客户端以及服务器端. 客户端负责实现虚拟场景下的功能, 包括用户管理模块、虚拟实验模块以及智能辅助模块. 服务器端负责处理系统运行时产生的数据, 包括用户数据、虚拟角色数据以及虚拟权限的数据.

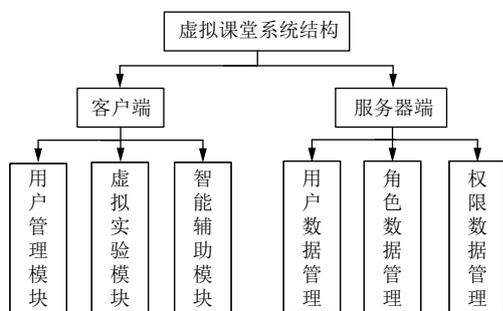


图1 系统整体结构

首先搭建系统场景, 采用三维建模软件 3dMAX 构建虚拟场景, 导入模型贴图后进行格式转换, 以 FBX 格式导出到 Unity3D 引擎^[7]. 其次进行客户端开发, 完成数据库设计, 利用 C#脚本进行组件化开发, 结合粒子系统和 NGUI 插件开发 GUI 界面和虚拟化学实验^[8-10], 设计权限策略, 并以此为基础开发智能模块. 最后通过 IDEA 进行服务器端开发, 利用 Spring Cloud 框架拆分微服务, 利用 Redis 缓存优化系统.

2 系统实现及关键技术

2.1 技术架构

系统技术架构如图2所示. 客户端采用 Unity3D 开发. 表现层对控制模型材质的 Shader 进行渲染并以 GUI 形式作为交互接口; 业务层完成虚拟课堂的核心功能, 包括用户的注册登录、虚拟实验及智能模块. 服务器端采用 IDEA 开发, 客户端通过 HTTP 接口和服务器端进行数据交互, 应用权限策略. MySQL 存储学生基本信息等持久数据, Redis 存储热点数据作为缓存.

2.2 场景建模

通过 3dsMax 制作虚拟化学课堂所需的三维模型^[11].

构建模型时, 既要保证模型质量, 也要保证模型的顶点总数和三角切面总数不至过高, 避免系统加载时占用大量内存开销, 影响系统运行效率. 制作实验仪器等小模型时, 提高模型精细程度; 制作教室墙体等大模型时, 降低模型精细程度.

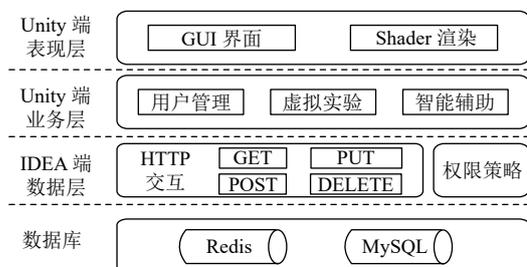


图2 系统技术架构

模型构建完成后, 在其材质贴图属性中添加贴图, 实现高仿真视觉效果. 此外还需将虚拟场景与真实课堂保持 1:1 比例, 增强学生使用系统时的沉浸感.

2.3 系统主场景

系统运行后, 首先加载如图3所示的负责用户注册及登录的主场景, 该场景通过 GUI 即图形用户界面实现, 用户通过鼠标和其他输入设备控制屏幕上的菜单和图标, 完成一系列交互活动.

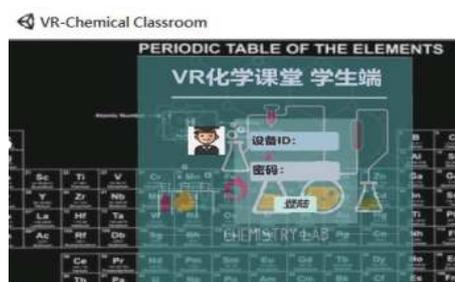


图3 系统登录场景

GUI 包含面板、按钮、输入框等属性, 完成界面后添加 C#脚本组件, 使 UI 可处理用户输入, 用户点击登录按钮后, 系统将根据用户及密码文本框的输入信息判断是否符合数据库中已录入的用户信息, 如符合则通过 Invoke() 方法调用 SceneManager.LoadScene() 系统 API 触发虚拟教室场景的加载并提示绿色的成功信息, 否则清空用户输入并提示红色的错误信息, 处理用户登录的 Login.cs 脚本的部分代码如图4所示.

2.4 虚拟化学实验

系统共包括 4 项化学实验, 分别为金属置换反应、火焰燃烧反应、硫酸稀释反应及活泼金属与水反应.

```

0 个引用
public void EndInput ()
{
    string user = userInput.text;
    string password = passwordInput.text;
    // 获取已注册用户信息
    ArrayList nameList=DataUtil.GetUserMsg();
    // 用户登录成功
    if (nameList.Contains(user)){
        text.color = Color.green;
        text.text = "登陆成功 / + "\n正在加载...请稍后...";
        Invoke("EnterClass", 2);
    }else{// 用户登录失败
        text.text = "抱歉, 您的 ID 或密码错误 \n请重试";
        text.color = Color.red;
        Invoke("ResumeText", 2);
    }
}

// 进入虚拟教学场景
0 个引用
public void EnterClass () {
    SceneManager.LoadScene(0);
}

// 恢复提示为空
0 个引用
public void ResumeText () {
    text.text = "";
}

```

图 4 处理用户登录的核心代码

金属置换反应 UI 如图 5(a) 所示, 该反应通过更新对液体颜色的渲染实现. Unity 中每个对象都具有坐标、渲染器、脚本等组件, 在对象的 C#脚本中可以获取其渲染器组件及具体负责渲染颜色的颜色组件, 再通过调整该组件的 RGB 属性改变颜色.

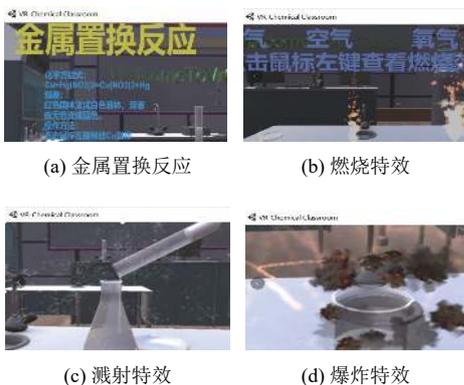


图 5 虚拟化学实验 UI

为铜块和液体分别添加碰撞器组件, 并勾选触发器属性. 当某游戏对象具有碰撞器组件时, 其可以和其它具有碰撞器组件的对象在场景中接触, 而触发器则可以在对象发生相关碰撞事件时触发指定逻辑.

通过 Input.GetMouseButtonDown() 系统 API 检测用户是否单击鼠标左键, 当检测到该事件发生时, 通过生成对象实例的方法在指定位置和角度触发铜块固体的生成事件, 并通过 NumberChange() 方法更新当前铜物质量. 之后调用 ColorChange() 方法, 当铜块接触液体时根据铜的数量决定液体颜色的改变程度, 控制该反应的 Test1.cs 脚本的核心代码如图 6 所示.

火焰燃烧反应模拟了火焰在氮气、空气以及氧气

3 种气体当中不同的燃烧效果, 主要通过粒子系统实现, 粒子系统是一种计算机图形学中模拟特定模糊现象的技术, Unity3D 中的粒子系统是参数化的, 通过调整参数值模拟不同效果, 难点在于确认合适的仿真参数, 在测试中不断调整参数值, 最终确定具有高度仿真效果的参数, 该反应的参数面板如图 7 所示. 运行时首先在脚本中获得粒子系统组件, 然后在 Update() 方法中每帧更新粒子生命周期、尺寸、速度等属性来渲染火焰燃烧的不同效果. 火焰燃烧特效如图 5(b) 所示.

```

@Unity 消息 | 0 个引用
void Update ()
{
    // 场景激活
    if (text.gameObject.activeInHierarchy)
    {
        // 点击鼠标左键
        if (Input.GetMouseButtonDown(0))
        {
            // 生成Cu固体
            Instantiate(cu, position.position, position.rotation);
            // 改变Cu数量
            NumberChange(cu);
            // 根据当前Cu数量改变其颜色
            ColorChange(cu, count);
        }
    }
}

```

图 6 处理金属置换反应的核心代码

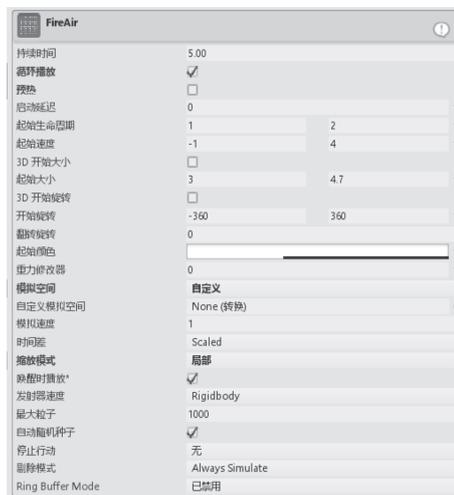


图 7 火焰粒子特性的参数面板

硫酸稀释反应同样通过粒子系统实现, 液体模型中的温度计数值随实验实时更新并显示, 该功能通过自定义 Shader 实现, Shader 用于控制 GPU 的图形渲染. 其中温度计 Shader 控制颜色变化的 surf() 方法的具体代码如图 8 所示. 除标准实验外, 开发非标准实验作为对比实验, 通过粒子系统模拟硫酸溅射特效帮助学生对该实验有更深刻的了解. 硫酸稀释对比实验中, 硫酸溅射特效如图 5(c) 所示.

```

// surf函数的参数是定死的,即input和input, 分别代表输入和输出
void surf(input i, input SurfaceOutputStandard o) {
    // uv_MainTex表示贴图坐标
    // tex2D 返回一个float4类型的值,表示贴图对应坐标的一点的颜色和Alpha值。
    fixed4 c = tex2D(_MainTex, i.uv_MainTex) * _Color;

    if (i.localPos.x >= _Clip && _Mode == 1 || i.localPos.y >= _Clip && _Mode == 2 ||
        i.localPos.z >= _Clip && _Mode == 3) {
        // Albedo 是像素的主颜色
        o.Albedo = _Color;
    } else {
        o.Albedo = c.rgb;
    }

    // 金属感
    o.Metallic = _Metallic;
    // 平滑度
    o.Smoothness = _Glossiness;
    // 透明度
    o.Alpha = c.a;
}

```

图8 温度计 Shader 控制颜色变化的具体代码

活泼金属与水的化学实验由于其巨大风险在现实中几乎无法进行,尤其是越活泼的金属反应强度越剧烈.为充分挖掘教学内容并通过强视觉冲击的三维效果吸引学生学习兴趣,选择了金属中最活泼的Cs作为实验材料,该金属与水反应时将产生极强的爆炸现象.该反应仍通过粒子系统实现,该实验中的粒子系统用以模拟爆炸及烟雾效果,其爆炸特效如图5(d)所示.

2.5 智能模块

2.5.1 权限策略

由于虚拟现实在教育应用的发展不够完善,缺乏对交互的权限控制,本系统提出一种基于RBAC的虚拟教学环境下的细粒度权限动态控制策略.该策略包括用户、角色、模型、权限、事件5大单元,其中对角色、模型以及权限单元进行了细粒度划分.

角色单元,即虚拟场景中的角色,包括3项信息:角色标识符、角色名称、角色类型.不同层次间的角色类型不同,如学生和教师角色;相同层次也具有不同的角色类型,如普通学生和班长角色.

模型单元,即虚拟场景中的模型,包括4项信息:模型标识符、模型权限、操作限制数.

权限单元,指虚拟场景中角色及模型的权限,包括4项信息:权限标识符、权限层级、权限内容、权限约束.权限内容划分为角色管理、模型管理.权限约束包括数量、时间约束,分别限制权限最大被拥有角色数及权限最长分配时长,防止“教学饥饿”.

事件单元,采用事件监听机制,对虚拟场景下全部事件进行监控,按照事件类型和处理原则处理.

为解决虚拟多人场景下虚拟模型所属权的问题,提出以下原则:

(1) 时间优先原则,虚拟角色发起模型交互请求时,发起时间节点较早的角色优先获得模型交互权.

(2) 权限优先原则,对模型交互具有不同权限的角

色在同一时间节点(时间间隔小于1ms)发起模型交互请求时,权限较高的角色优先获得模型交互权.

(3) 随机分配原则,为避免具有相同权限的角色于同一时间节点发起模型交互请求的而导致系统故障,使用随机生成数解决该问题.

为解决多人环境非实验条件下进行获取虚拟模型所属权的问题,提出课堂进度原则:综合学生角色的成绩与课堂进度计算出权限优先值 T , T 值最小的虚拟角色优先获得模型交互权.其中学生角色 s 的权限优先值 $T_s=(p_s/p)w_p+(g_s/g)w_g$, p_s 为学生角色 s 的实时课堂进度, p 为全体学生角色的平均实时课堂进度, g_s 为学生角色 s 的课程成绩, g 为全体学生角色的平均课程成绩, w_p 为课堂进度对 T 的权重值, w_g 为成绩对 T 的权重值,系统通过该原则确定所有虚拟角色的 T 值,对其升序排列入队,依次出队分配权限.

上述原则的具体实现依赖Unity3D脚本Update()方法每帧更新的特性,利用该特性实现事件监听机制,持续监听交互请求,当收到请求时通过HTTP接口将请求数据交由服务器端进行处理,服务器端处理权限分配问题的核心代码如图9所示.

2.5.2 智能提示

智能模块除对学生角色进行权限控制外,还支持自主学习,学生可根据场景中智能提示观看教学视频.不但解决传统教育模式中教师有限时间内无法顾全所有学生的缺陷,又以虚拟现实这种新技术丰富了教育趣味性,扩展了可教学内容.

该功能主要通过C#脚本及NGUI插件实现.NGUI是C#编写的Unity插件,提供强大的UI功能,开发时不用点击Play按钮进入运行状态就能即时查看结果.

利用NGUI为4项化学实验设置具体的提示UI,利用C#脚本指定文字提示触发的具体时机.Unity中每个对象都具有Transform组件,该组件中包含了模型的坐标信息.系统运行时首先在控制文字提示脚本的Start()方法中获取虚拟角色与各实验桌的坐标信息,再在Update()方法中通过Vector3向量API计算两者间的向量距离 d ,当 d 小于虚拟场景中指定距离时通过SetActive()方法激活提示组件,将UI呈现给虚拟角色.

视频提示通过按键手动实现,系统支持用户自定义触发按键.当控制视频组件脚本的Update()方法检测到指定输入时,将激活虚拟场景中的教学视频对象,获取其附属的Video视频组件,然后调用Play()方法

播放视频, 控制视频组件播放脚本方法的具体代码如图 10 所示, 硫酸稀释实验的智能提示如图 11 所示。

```
private Map<String, Data> permissionControl(Event event, Data data)
throws HTTPException {
    Map<String, Data> handleMsg = null;
    // 多人环境实验条件下的事件
    if (event.type == EventType.WITH_EXPERIMENT) {
        Principle p = data.getPrincipleType();
        if (p == Principle.TIME) {
            // 时间优先原则
            handleMsg = timeFirstHandle(event, data);
        } else if (p == Principle.PERMISSION) {
            // 权限优先原则
            handleMsg = permissionFirstHandle(event, data);
        } else if (p == Principle.RANDOM) {
            // 随机分配原则
            handleMsg = randomHandle(event, data);
        }
    }
    // 多人环境非实验条件下的事件
    if (event.type == EventType.NO_EXPERIMENT) {
        List<Model> models = data.getModelList();
        int minPrior = Integer.MAX_VALUE;
        for (Model model : models) {
            minPrior = getMinPrior(model, minPrior);
        }
        // 课堂进度原则
        handleMsg = classHandle(minPrior, models, event, data);
    }
    // 出现错误
    if (handleMsg != null) {
        handleMsg = errorHandle();
    }
    return handleMsg;
}

```

图 9 服务器权限策略代码

```
void Update()
{
    // 发生用户输入事件
    if (EndInput())
    {
        // 激活教学视频对象
        gameObject.SetActive(true);
        // 获取视频组件
        videoPlayer = gameObject.GetComponent<VideoPlayer>();
        // 播放教学视频
        videoPlayer.Play();
    }
}

```

图 10 触发视频播放的核心代码

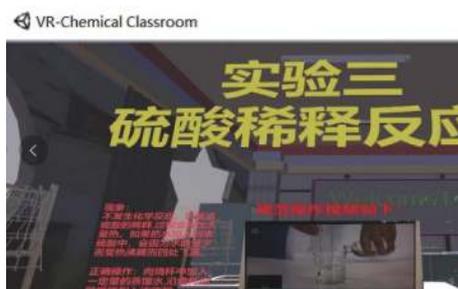


图 11 硫酸稀释实验的智能提示

3 系统优化

针对现有虚拟课堂缺乏系统优化的问题, 提出运用 Spring Cloud 微服务及 Redis 缓存解决, 对系统服务器端的功能进行拆分, 并缓存数据处理结果。

微服务是一种构建应用的架构方案, 可将应用拆分成多个功能, 每个功能都被称为一项服务, 各项服务不会相互影响。依据功能不同, 将本系统服务器端整体服务拆分为 3 部分: 用户服务: 处理用户的注册与登录问题; 角色服务: 处理虚拟角色进行各项虚拟实验的问题; 权限

服务: 处理权限策略应用于当前虚拟场景的问题。

本系统后端微服务的服务处理流程如图 12 所示。

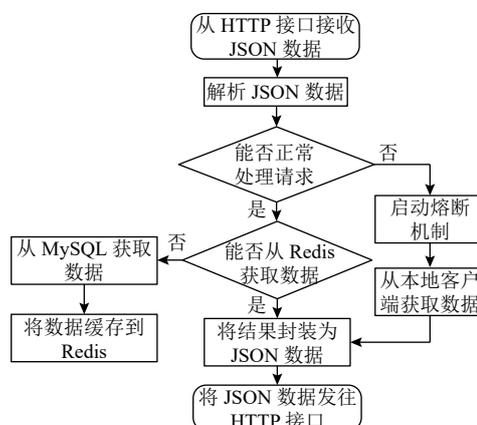


图 12 服务处理流程

相比现有虚拟课堂的单体架构, 使用微服务框架对系统功能进行拆分与部署有诸多优势, 概括如下:

- 1) 分布式部署, 后期可以将处理不同模块数据的服务部署于不同服务器, 大幅提升数据处理效率。
- 2) 高度可扩展, 随项目进行, 虚拟课程和实验数量都将增加, 微服务架构下可以完全不影响当前项目部署而增设物理生物等其它课程, 充分满足用户需求。
- 3) 高可用, 服务互相独立, 彼此不存在影响, 当某个微服务出现故障时, 不会影响其它微服务。

3.1 数据交互

虚拟场景中产生的数据依据功能划分提交至不同网络接口处理, 当有数据处理请求发起时, 在 C# 脚本的 Update() 方法中启动一个协程, 通过 Unity3D 的 HTTP REST API 访问微服务对应 HTTP 接口, 将数据封装为 JSON 格式并提交; 当数据处理请求返回时, 接收 JSON 数据并解析, 将数据填充至 UI 中的数据域, 如查询学生成绩功能中客户端和服务端进行数据交互的具体代码如图 13 所示。

3.2 服务降级

如果发生异常, 系统将进行服务降级, 从本地客户端获取实验相关信息并填充 UI, 呈现用户友好界面, 避免影响用户体验。服务降级通过 Spring Cloud 的 hystrix 熔断器组件实现, 该组件的作用是在不改变各微服务调用关系的前提下, 针对错误情况进行预先处理。

首先在项目的 pom.xml 文件中引入 hystrix 组件的坐标, 该文件主要在 JavaWeb 项目中对网络中的组件进行下载引用, 使项目可以使用第三方类或 API; 其

次, 在项目中对服务熔断的服务接口进行注册, 将其部署于默认的 8070 端口, 并配置该组件的 `enabled` 属性为 `true`, 表示启用熔断器; 之后利用注解对表现层的控制器类进行注解, 注解的 `fallback` 属性则指明了服务降级所依赖的错误处理类, 在该类中直接返回带有错误标识信息的 JSON 数据, 客户端接收后, 如判断有服务降级的标识, 则直接从本地客户端获取数据。

```
public IEnumerator GetRequest(int sid, int cid)
{
    string url = "http://localhost:8010/unity/findStudentById/" + sid + "/" + cid;
    Debug.Log("请求的url: " + url);
    // 以下以JSON格式请求数据
    using (UnityWebRequest webRequest = UnityWebRequest.Get(url))
    {
        yield return webRequest.SendWebRequest();

        // 接受返回的数据
        Dictionary<String, object> data = null;

        // 出现网络错误
        if (webRequest.isHttpError || webRequest.isNetworkError)
        {
            Debug.LogError(webRequest.error + "\n" + webRequest.downloadHandler.text);
        }
        else
        {
            //webRequest.downloadHandler.text是获取到的JSON数据, 对其进行解析
            Student student = JsonConvert.DeserializeObject<Student>(webRequest.downloadHandler.text);
            nameText.text = student.name;
            courseText.text = student.cname;
            scoreText.text = "" + student.score;
        }
    }
}
```

图 13 客户端和服务端进行数据交互的代码

事先在服务器端正常返回 JSON 数据前的代码中插入 `int i = 1/0;` 的异常代码, 当在进行金属置换反应实验场景中希望查询成绩时, 当点击查询成绩后, 服务器端产生异常, 将返回带有错误信息的数据处理结果, 客户端接收后, 通过 `SetActive(true)` 这个 Unity3D 当中用于激活游戏对象组件的系统 API 将含有本地存储的关于铜置换反应的化学方程式及现象的提示 UI 组件激活, 同时调用 `SetActive(false)` 隐藏实验组件, 减小资源开销, 这种用户友好的错误提示 UI 能够避免在故障情况下影响用户体验, 其效果如图 14 所示。



图 14 服务熔断后的金属置换反应

4 结语

本文实现了虚拟化学智能课堂系统, 操作简易, 三维视觉效果良好, 有助于中学学生进行化学课的虚拟学习, 更直观地学习化学反应知识, 了解各种化学反应的实验效果, 更好地构建完整的知识体系框架。

该系统的智能辅助模块能够依据权限策略对不同学生分配不同权限, 进行针对性教学; 通过文字视频等提示帮助学生在没有教师指导的情况下进行自主学习, 保证虚拟教学质量。其多项化学实验挖掘了可教学内容, 包括爆炸在内的粒子特效为学生提供高沉浸体验, 吸引学生学习兴趣。该系统运行时, 服务端数据采用微服务框架及 Redis 缓存处理, 相比单体架构效率更高, 同时也能够通过熔断机制实现高可用特性。

后期工作是增设物理、生物等虚拟课程, 开发支持项目商业化的内置商城模块, 寻找并解决权限策略的漏洞, 严格保障系统质量。

参考文献

- 刘崇进, 吴应良, 贺佐成, 等. 沉浸式虚拟现实的发展概况及发展趋势. 计算机系统应用, 2019, 28(3): 18–27. [doi: 10.15888/j.cnki.csa.006732]
- 蔡宝, 石坤举, 朱文华. 基于虚拟现实技术的车床仿真系统. 计算机系统应用, 2018, 27(5): 86–90. [doi: 10.15888/j.cnki.csa.006342]
- Wang SSY, Teo WZW, Teo WZY, *et al.* Virtual reality as a bridge in palliative care during COVID-19. *Journal of Palliative Medicine*, 2020, 29(1): 266–270. [doi: 10.1089/jpm.2020.0212]
- 胡芳睿, 赵梦, 邱鹏, 等. 基于 Unity3D 的药用植物学实训虚拟仿真系统. 计算机系统应用, 2020, 29(1): 266–270. [doi: 10.15888/j.cnki.csa.007221]
- 郭立威, 王权, 高克斌, 等. 基于虚拟现实的装甲车辆运动仿真研究. 计算机系统应用, 2019, 28(6): 221–227. [doi: 10.15888/j.cnki.csa.006943]
- Mills B, Dykstra P, Hansen S, *et al.* Virtual reality triage training can provide comparable simulation efficacy for paramedicine students compared to live simulation-based scenarios. *Prehospital Emergency Care*, 2020, 24(4): 525–536. [doi: 10.1080/10903127.2019.1676345]
- 陈阳, 胡永斌. 基于 VR 的沉浸式自闭症康复系统的设计与开发. 中国医学教育技术, 2020, 34(4): 463–466, 470. [doi: 10.13566/j.cnki.cmet.cn61-1317/g4.202004016]
- 郝孜奇, 张文胜. 基于 Unity3D 的铁路实训虚拟仿真系统开发. 计算机仿真, 2020, 37(6): 99–103, 241. [doi: 10.3969/j.issn.1006-9348.2020.06.022]
- 陈浩, 索南尖措, 李红, 等. Vuforia 在中学化学仿真实验系统中的运用. 电子技术与软件工程, 2020, (10): 36–37.
- 俞磊, 韩春香. 基于 Unity3D 和 3DSMax 的虚拟医院漫游系统的设计与实现. 长春大学学报, 2020, 30(4): 40–44.
- 熊巍, 何蔚珊. 基于 Unity3D 的化学虚拟实验系统设计与实现. 实验技术与管理, 2020, 37(2): 28–31. [doi: 10.16791/j.cnki.sjg.2020.02.007]