

T/EJCCCSE

中国商业股份制企业经济联合会团体标准

T/EJCCCSE XXXX—XXXX

教学训练三维交互资源快速开发标准化流程及工具

Teaching and training: 3D interactive resources, rapid development of standardized processes and tools

（征求意见稿）

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

2024 – XX – XX 发布

2024 – XX – XX 实施

中国商业股份制企业经济联合会 发布

目 次

前言	II
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 开发流程	1
5 工具	6

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由武汉蓝海科创技术有限公司提出。

本文件由中国商业股份制企业经济联合会归口。

本文件起草单位：武汉蓝海科创技术有限公司

本文件主要起草人：

教学训练三维交互资源快速开发标准化流程及工具

1 范围

本文件规定了教学训练三维交互资源快速开发标准化流程及工具的术语和定义、开发流程、工具。本文件适用于教学训练三维交互资源的快速开发。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语和定义

本文件没有需要界定的术语和定义。

4 开发流程

4.1 需求分析

4.1.1 明确教学目标：确定三维空间要辅助达成的教学训练目的。

4.1.2 分析用户需求：考虑学习者的年龄、知识水平等。

4.1.3 确定空间特性：如大小、复杂度、环境类型等。

4.2 模块设计

4.2.1 三维空间设计

4.2.1.1 明确设计目标

应依据教学训练内容确定三维空间主题，明确目标是知识讲解、技能训练还是体验式学习，以此指导空间设计。

4.2.1.2 规划布局与比例

应根据主题合理安排物体和场景元素；确保空间内物体比例准确、符合实际。

4.2.1.3 设计导航与视角控制

应创建直观便捷的导航方式；提供多种视角选择和控制方式，包括自由视角、跟随视角等。

4.2.1.4 营造环境与氛围

应模拟真实的环境条件；通过光照、色彩、音效等营造合适氛围。

4.2.2 三维空间仿真设计

4.2.2.1 物理环境仿真

物理环境仿真设计主要包含以下内容：

- 重力模拟：根据教学内容需求，合理设置重力参数；
- 摩擦力与阻力：在机械操作训练中，应精确模拟物体间的摩擦力，使操作手感真实。

4.2.2.2 空间特性仿真

空间特性仿真设计主要包含以下内容：

- 空间维度：应通过特殊的视觉和交互设计来体现空间的多维特性，辅助学习者理解抽象概念；

- b) 空间限制与边界：应明确呈现空间的边界和限制，使学习者能体验到实际操作中的空间约束。

4.2.2.3 环境变化仿真

环境变化仿真设计主要包含以下内容：

- a) 天气变化：如在地理教学训练中，应模拟雨、雪、风、雾等天气变化对环境和行动的影响；
- b) 昼夜交替与时间流逝：如在农业种植教学中，应展示昼夜交替对植物生长的影响；在历史场景重现中，通过时间流逝体现不同时段的环境变化。

4.2.2.4 特殊效果仿真

特殊效果仿真设计主要包含以下内容：

- a) 光影效果：如在艺术鉴赏教学中，模拟不同光照角度和强度下艺术品的视觉效果；在建筑设计教学中，体现光影对建筑外观和内部空间的影响；
- b) 音效模拟：如在音乐教学中，营造不同空间的声学效果；在灾难逃生训练中，通过声音的传播和反射效果，辅助学习者判断环境情况。

4.2.3 API 设计

4.2.3.1 教育数据交互

教育数据交互 API 设计主要包含以下内容：

- a) 用户信息接口：用于获取和更新用户的基本信息、学习进度、历史训练记录等，方便系统根据用户特征调整三维交互内容；
- b) 课程数据接口：使三维交互资源能与教学课程管理系统连接，获取课程大纲、教学目标、知识点分布等数据，确保资源内容与课程紧密结合。

4.2.3.2 三维场景控制

三维场景控制 API 设计主要包含以下内容：

- a) 场景加载接口：根据教学训练需求，快速加载指定的三维场景，包括场景的初始状态、物体布局等，提高资源的响应速度；
- b) 场景切换接口：在多场景教学中，实现场景之间的平滑切换；
- c) 视角控制接口：提供对三维场景视角的控制功能，包括旋转、缩放、平移等操作，满足用户从不同角度观察场景的需求。

4.2.3.3 交互功能

交互功能 API 设计主要包含以下内容：

- a) 操作交互接口：处理用户在三维场景中的各种操作，将用户操作反馈给系统，实现相应的交互效果，如操作物体的移动、组装等；
- b) 反馈交互接口：当用户完成某个操作或任务时，接收系统的反馈信息，并将其以合适的方式呈现给用户。

4.2.3.4 资源管理

资源管理 API 设计主要包含以下内容：

- a) 模型资源接口：负责管理三维模型资源的加载、卸载和更新，确保模型的准确性和高效利用；
- b) 多媒体资源接口：处理与教学内容相关的多媒体资源，使多媒体资源与三维场景和交互功能协同工作。

4.3 模块开发

4.3.1 三维空间开发

4.3.1.1 规划三维空间

根据教学训练目标确定空间类型、规模及空间布局。

4.3.1.2 创建三维模型

应按照规定建立三维模型，模型精度依据教学需求而定。

4.3.1.3 设计空间交互

确定用户在三维空间中的交互方式，如点击、拖拽、缩放等基本操作。

4.3.1.4 构建空间环境

营造适合的物理环境，包括重力、光照、温度等因素。

4.3.1.5 集成与测试

4.3.1.5.1 将三维空间模块与教学训练系统的其他部分集成，确保数据传输和功能调用正常。

4.3.1.5.2 进行全面测试，检查空间模块在不同硬件设备和软件环境下的兼容性、稳定性和性能表现；同时收集用户反馈，对空间模块进行优化和改进。

4.3.2 API 模块开发

4.3.2.1 应选择合适的开发技术和工具。根据需求和设计，选择编程语言、开发框架等进行 API 的开发。

4.3.2.2 应按照规定实现各个接口的功能。对于数据交互接口，要确保数据的准确传输和存储；对于三维场景控制接口，要实现对场景的加载、切换、视角控制等操作。

4.3.2.3 进行严格的代码测试，包括单元测试、集成测试等，确保 API 的稳定性和可靠性。

4.4 模块代码编写

4.4.1 代码结构

4.4.1.1 模块划分

应将代码按照功能划分为不同模块，如三维场景初始化模块、模型加载模块、交互功能模块、数据处理模块等。每个模块的代码应存放在独立的文件中，便于维护和管理。

4.4.2 层次划分

在每个模块内，代码的逻辑层次应清晰。例如，先进行变量定义和初始化，再进行主要功能的实现，最后是清理和释放资源相关的代码。

4.4.3 代码兼容性

4.4.3.1 考虑目标环境：明确教学训练三维交互资源的运行环境，包括不同的浏览器、操作系统和设备类型。编写代码时，要进行兼容性测试和处理，避免使用特定环境下不支持的特性。

4.4.3.2 使用兼容性库或技术：如果需要使用较旧的浏览器或设备，可以使用一些兼容性库处理代码的兼容性问题，或者采用渐进增强的开发策略。

4.5 功能测试

4.5.1 场景加载测试

4.5.1.1 测试目标

确保三维场景能够正确、快速地加载，无卡顿或报错。

4.5.1.2 测试步骤

启动资源，检查各个教学训练场景是否能完整呈现，包括场景中的物体、环境等元素。同时，记录加载时间，在不同设备上多次测试，确保加载时间在合理范围内。

4.5.2 交互功能测试

4.5.2.1 基本交互操作测试

4.5.2.1.1 测试目标：验证用户在三维空间中的点击、拖拽、缩放等操作是否流畅、准确。

4.5.2.1.2 测试步骤：在场景中尝试对物体进行各种基本操作，检查物体的响应是否符合预期。

4.5.2.2 教学相关交互测试

4.5.2.2.1 测试目标：检查针对教学训练内容设计的特殊交互功能是否正常工作。

4.5.2.2.2 测试步骤：在不同教学场景中进行相关交互，确认交互能正确触发相应的效果和反馈。

4.5.3 教学内容呈现测试

4.5.3.1 测试目标：保证教学内容在三维空间中的整合与呈现符合设计要求。

4.5.3.2 测试步骤：查看文本、图像、音频、视频等教学素材是否能在正确的位置和时间显示或播放。检查教学内容与三维场景、交互功能的结合是否紧密，是否有助于学习者理解教学点。

4.5.4 性能测试

4.5.4.1 硬件性能测试

4.5.4.1.1 测试目标：评估资源在不同硬件配置下的运行性能，确保满足最低要求。

4.5.4.1.2 测试步骤：在多种硬件设备上运行资源，监测帧率、内存占用等指标，避免出现严重的卡顿或崩溃现象。

4.5.4.2 兼容性测试

4.5.4.2.1 测试目标：确认资源与常用教学设备、软件和操作系统的兼容性。

4.5.4.2.2 测试步骤：在不同操作系统和常用教学软件环境中运行资源，检查是否存在兼容性问题，如显示异常、功能失效等。

4.5.5 用户体验测试

4.5.5.1 测试目标：从用户角度评估资源的易用性和沉浸感。

4.5.5.2 测试步骤：邀请目标用户群体进行试用，收集用户对操作便捷性、界面友好度、学习体验等方面的反馈，如是否容易上手、是否能长时间保持注意力。

4.6 模块迭代开发

4.6.1 收集反馈

4.6.1.1 用户反馈收集：通过问卷调查、用户评价、现场观察等方式，收集学习者和教师在使用三维交互资源过程中的问题、建议和新需求。

4.6.1.2 系统数据收集：分析资源使用数据，包括使用频率、停留时间、操作路径等，了解用户行为模式和资源的薄弱环节。

4.6.2 需求评估

4.6.2.1 重要性排序：对收集到的反馈和数据进行整理，评估各项需求的重要性。

4.6.2.2 可行性分析：结合技术、时间和资源等因素，分析满足需求的可行性。

4.6.3 设计改进

4.6.3.1 功能优化设计：根据需求评估结果，对三维交互资源的功能进行优化设计。

4.6.3.2 架构调整设计：如果需求涉及到系统架构的变化，对资源的架构进行相应调整，确保系统的可扩展性和灵活性。

4.6.4 代码开发与测试

4.6.4.1 模块更新开发：按照设计方案，对相关模块进行代码更新。在开发过程中，遵循原有的代码编写规范，确保代码质量和可维护性。

4.6.4.2 回归测试与新功能测试：对更新后的模块进行全面测试，包括回归测试以确保原有功能未受影响，以及新功能测试以验证新需求是否得到满足。对测试中发现的问题及时修复。

4.6.5 部署与持续监测

4.6.5.1 更新部署：将迭代后的模块部署到教学训练环境中，确保更新过程对正常教学活动的影响最小化。

4.6.5.2 持续监测效果：在部署后继续收集用户反馈和系统数据，监测迭代效果，为下一轮迭代开发提供依据。

4.7 交付物开发与测试

4.7.1 三维交互资源安装包

4.7.1.1 开发

三维交互资源安装包的开发应符合以下要求：

- 内容完整性：将所有三维交互资源相关的文件、模型、纹理、脚本等打包成一个完整的安装包。确保不同教学训练场景、功能模块在安装后都能正常运行，没有缺失文件导致的错误；
- 安装便捷性：设计简单易懂的安装向导，对于普通用户来说，能够轻松完成安装过程，无需复杂的技术操作。

4.7.1.2 测试

三维交互资源安装包的测试主要包含以下内容：

- 兼容性测试：在多种目标操作系统（如 Windows、Mac、Linux 的不同版本）和设备类型（如台式机、笔记本、平板电脑）上测试安装包的安装过程，确保没有兼容性问题；
- 完整性测试：安装完成后，检查所有资源文件是否正确安装到指定位置，启动资源，检查是否有文件损坏或缺失导致的运行错误。

4.7.2 用户手册

4.7.2.1 开发

用户手册的开发应符合以下要求：

- 功能介绍：详细描述三维交互资源的各种功能，包括三维场景的浏览方式、交互操作（如点击、拖拽、特定教学交互动作）、教学内容呈现形式（如文本提示、动画演示）等；
- 操作指南：为用户提供清晰的操作步骤，从启动资源到完成教学训练任务，例如如何在不同场景间切换、如何使用各种工具进行训练，配有相应的截图或示意图；
- 常见问题解答：收集可能出现的问题，如安装问题、运行时的报错、交互不响应等，并给出解决方案，帮助用户快速解决使用过程中的困难。

4.7.2.2 测试

用户手册的测试主要包含以下内容：

- 准确性测试：按照用户手册的内容进行实际操作，检查手册中描述的功能和操作步骤是否与实际情况一致，有无误导用户的信息；
- 清晰度测试：邀请非技术人员阅读用户手册，评估手册内容是否易于理解，语言是否简洁明了，截图和示意图是否清晰准确地辅助理解。

4.7.3 教学案例文档

4.7.3.1 开发

教学案例文档的开发应符合以下要求：

- 案例设计：创建一系列具有代表性的教学案例，这些案例应涵盖不同的教学训练目标和场景，如针对某个知识点的讲解案例、某种技能的训练案例；
- 教学步骤：详细阐述每个教学案例的实施步骤，包括教师在使用三维交互资源时的引导步骤和学生的操作步骤，展示如何利用资源实现教学效果。

4.7.3.2 测试

教学案例文档的测试主要包含以下内容：

- a) 可行性测试：教师按照教学案例文档尝试在真实教学场景中使用三维交互资源，检查案例中的教学步骤是否可行，是否能达到预期的教学效果；
- b) 有效性测试：观察学生在使用案例进行学习后的反馈和学习成果，评估教学案例是否有效帮助学生理解知识和掌握技能，是否充分发挥了三维交互资源的优势。

5 工具

教学训练三维交互资源快速开发工具主要包括以下几种：

- a) 三维交互资源仿真开发及测试工具；
 - b) 三维交互资源业务逻辑开发及测试工具；
 - c) 训练三维交互资源集成测试工具。
-