

游戏开发驱动的 C++ 教学探索与实践

张晓东 焦阳 唐继军*

深圳理工大学计算机科学与人工智能学院, 深圳 518107

摘要 针对 C++ 语言教学中内容抽象、学习动机不足等问题, 本文提出一种以海战游戏开发为核心案例的游戏驱动 C++ 教学模式。学生在 Ubuntu 终端环境下完成一款包含多类型敌舰、武器系统、补给机制和关卡系统的游戏开发, 基于情境学习、反馈学习、自我决定和认知负荷理论指导, 优化 C++ 教学模式。通过分析 291 名学生的项目完成度与课程综合成绩, 显示两者统计上较高的正相关 ($r=0.41$, $p<0.001$)。本文为程序设计课程引入真实场景游戏案例, 提供可复用的教学设计和实证依据。

关键字 C++ 教学, 游戏驱动学习, 情境学习, 反馈学习, 自我决定, 认知负荷

Exploration and Practice of Game Development-Driven C++ Education

Zhang, Xiaodong Jiao, Yang Tang, Jijun

Faculty of Computer Science and Artificial Intelligence
Shenzhen University of Advanced Technology
Shenzhen 518107, China

zhangxiaodong@suat-sz.edu.cn jiaoyang@suat-sz.edu.cn tangjijun@suat-sz.edu.cn

Abstract—Addressing the issues of abstract content and lack of learning motivation in C++ language teaching, this paper proposes a game-driven C++ teaching model centered on the development of a naval battle game. Students complete the development of a game featuring multiple types of enemy ships, weapon systems, supply mechanisms, and a level system within the Ubuntu terminal environment. The C++ teaching model is optimized through the theories of situated learning, feedback learning, self-determination, and cognitive load. Analysis of the project completion scores and comprehensive course grades of 291 students reveals a statistically significant positive correlation ($r = 0.41$, $p < 0.001$). This paper provides a reusable instructional design and empirical evidence for introducing real-world game scenarios into programming courses.

Keywords—C++ education, game-driven learning, situated learning, feedback learning, self-determination theory, cognitive load theory

1 引言

C++ 作为计算机专业的核心编程语言, 是计算机核心专业课程的先修课程, 如数据结构、算法设计、操作系统等。然而, 由于其复杂的语法体系和抽象的编程范式给教学实践带来了较大挑战。长期调研发现, C++ 初学者普遍面临如下困惑。1) 概念抽象, 如动态内存、指针与引用、多态与继承、模板等, 学生难以建立直观理解, 导致学习停留在知其然层面, 在实际编程中不知如何运用。特别是, 传统 C++ 教学多以 Windows 环境下的集成开发环境 Visual Studio (VS) 为主, 学生虽然可以较快的运行代码, 但对于编译链接过程、内存布局、底层系统调用缺乏直观认识。2) 传统教材的教学案例脱离真实应用场景, 并缺乏趣味

性, 导致学生主动学习和实践动力不足, 存在懂语法而不会写程序等现象[1]。

近年来, 项目驱动学习[2]已被证明能够有效地提高编程语言学习效果。郭文佳等将 CDIO 理念(构思 - 设计 - 实现 - 运作)融入 C++ 项目式教学[3], 取得了良好效果, 为本方案的设计提供了参考。在新工科建设背景下, 孟文龙等进一步指出程序设计基础课程亟需向实践导向转型[4]。将游戏项目引入 C++ 教学, 将有助于学生在创造游戏的过程中主动学习语言特性, 显著增强内在动机和知识迁移能力[5-6]。然而, 现有案例多集中在规则简单的游戏, 缺乏对继承、多态、内存管理等复杂 C++ 特性的深度覆盖。李国和等在程序综合实践课程中探索了案例教学法的应用, 证实了高质量案例对提升学生综合实践能力的价值, 但具体案例的设计仍有待进一步丰富和深化[7]。

*通讯作者: 唐继军 tangjijun@suat-sz.edu.cn。

此外,基于VS的开发环境,学生在图形用户接口(GUI)等方面投入过多精力,而忽略对C++语言本质特性的融会贯通。而以Ubuntu为代表的Linux环境因其开源、透明、贴近工业界特点,其在编程教学中的应用得到越来越多关注。在Ubuntu系统下,学生需要手动调用g++编译器、编写Makefile文件、管理动态库等,将有助于理解C++程序的构建过程和底层运行机制。同时,Ubuntu系统丰富的终端库(如libncurses)为C++课程项目实践提供了有力支持,使学生在无需学习复杂GUI框架情况下即可实现可视化交互。

针对传统C++语言教学存在的挑战,本文将探索游戏开发驱动C++语言教学的作用机制和实践效果。本工作响应新工科对实践能力的要求,为程序设计类课程的项目化教学改革提供了一条可复用的路径,主要贡献包括:

1) 提出一个以海战游戏为核心的C++综合教学案例。游戏开发覆盖C++教学全过程,融合情境学习、反馈学习、自我决定及认知负荷等理论,实现以学生为主体的主动学习模式。案例覆盖C++类继承、多态、内存管理、标准模板库(STL)容器、异常处理等核心知识点;并具有较强的可扩展性,在保证多数学生能够完成基本任务的同时,为适应能力快的学生提供个性化开发选择。

2) 构建基于Ubuntu和libncurses环境的游戏开发驱动教学方案,将学生注意力集中于C++特性和底层运行逻辑,并获得具有较高成就感的实践作品。为了评估教学效果,采用单组相关实验设计,通过项目完成度与课程综合成绩的相关性分析,检验游戏开发对于知识点理解与运用的促进作用。该方案为高校计算机专业在Linux环境开展C++教学提供可复用参考,同时也为其他编程语言教学提供经验。

2 游戏开发驱动C++教学的设计理念

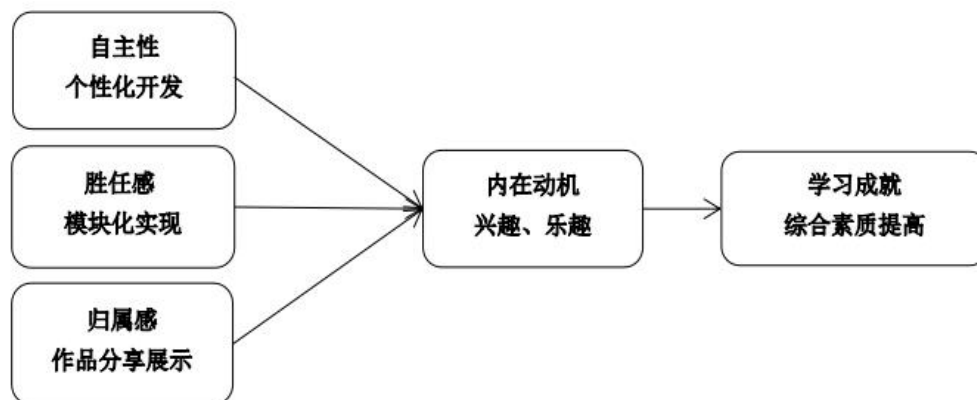


图1. 自我决定理论模型应用于游戏开发

自主性。游戏设定基本任务、扩展任务及个性化任务。学生可以在实现基本任务后,自由选择个性化任务,例如,在游戏中加入更多的智能元素,实现个

C++教学改革,在教师端将侧重基于真实案例构建应用场景,探索实际需求引导的渐进式教授模式;在学生端则逐步从以外在动机(如考试奖惩)为主导的学习模式向以内在动机为主导(学习兴趣)的学习模式过渡,在实践中反哺理论知识和编程技能学习[8]。而游戏开发则是驱动C++教学改革的一个重要内容。

2.1 情境学习

情境学习理论指出[9],知识如果脱离其应用情境,学习者将难以理解其意义和用途。而游戏提供了一个真实的问题情境,使得每个C++语法元素或者特性,成为游戏设计中的不可或缺的解决工具,为知识赋予意义和生命力。如在游戏中不同敌方角色,需要利用C++类和继承特性进行抽象。角色可以基于相同的接口函数,实现不同的外观和动作。游戏过程中敌方角色生存期的动态管理,则需要考虑对象的动态申请与释放,同时借助于动态数组统一管理,将有助于理解C++内存动态管理机制。

2.2 反馈学习

编程学习高度依赖反馈,而游戏开发具有即时反馈特性。每完成一个模块,排除语法错误后即可编译运行,实时看到游戏行为的变化,如游戏的攻击判定是否正确,胜负判断是否触发等。游戏逻辑中的逻辑错误(Bug)往往以直观形式表现出来,学生必须通过调试追踪问题根源。这一过程不仅锻炼了调试技能,更深化了对程序状态变化的理解。这种编码-运行-观察-修改的逻辑循环,符合认知科学中的测试效应,即主动检索和即时纠错比被动学习更能促进记忆[10]。

2.3 内在动机

根据自我决定理论,人的内在动机来源于三种基本心理需求的满足,即自主性、胜任感和归属感[11-12],如图1所示。而游戏开发课题则满足这三种需求:

性化扩展,如增加难度选择、加入网络协同等,满足不同层次学生的自主性需求。**胜任感。**配合理论教学进度,游戏被分解为多个模块。在完成知识点学习后,

将其应用在子模块的实现，同时获得一次胜任体验。当最终游戏完整运行时，学生获得的成就感远高于完成一道编程题。**归属感**。游戏是有趣的、可分享的作品。学生可以通过朋友圈、抖音等社交媒体在线上展示自己的游戏作品，或者线下与同学、朋友进行对战和比赛，并可以组队继续完善作品。代码不再是孤独的作业，而是社交互动的媒介。

当以上三种基本心理需求得支持和满足时，个体的内在动机得到增强和激发，学生将因为游戏开发本身的兴趣和乐趣而投入，而非追求任何分离的结果，有助于取得更大的学习成就。

2.4 认知负荷

认知负荷理论中，学习材料的复杂度如果超出学习者的工作记忆容量，学习就会失败[13]。传统 C++ 教学常同时引入过多新概念，比如类、继承和多态，学生难以一次性掌握，超出学生接受能力。游戏开发通过渐进式需求实现了认知负荷的管理，它将复杂内容分解为若干小步骤，每步只强调一个新知识点。学生先基于已有知识完成一个初始版本，如基于函数实现。然后，当代码变得庞大和混乱时，自然产生重构的需求，顺势引入 C++ 类及其特性，使得学生在具体情境中

理解抽象概念。这种顺序显著降低了认知负荷，让学习更加高效。

3 海战游戏案例的教学设计

基于游戏开发在 C++ 教学方面的优势，我们设计了一个贯穿 C++ 教学过程的海战游戏。学生利用 C++ 语言、libncurses 库及标准模板库，在 Ubuntu 环境下基于文本界面完成海战游戏开发。游戏中，玩家控制一艘战列舰，消灭随机生成的敌方舰船（炮艇、驱逐舰、巡洋舰）和轰炸机，并躲避敌方单位的攻击。玩家通过方向键移动，使用不同按键发射炮弹、散射子弹、发射导弹攻击敌方单位。击沉敌舰可获得金币。累计 1000 金币通关，生命值归零则游戏失败。游戏还包含武器补给包、医疗包以及可选的障碍物、颜色、多关卡等拓展任务，并支持学生加入个性化功能和特色。

3.1 教学目标

在知识层面，强化学生对 C++ 类的继承与多态、动态对象创建与销毁、STL 容器管理动态对象、递归算法、异常处理等知识点的理解和使用，实现不同游戏模块。模块与知识点映射表详见表 1。

表 1 海战游戏知识点覆盖映射表

游戏模块	C++ 特性	教学重点
敌舰类型（炮艇、驱逐舰、巡洋舰）	继承、虚函数、纯虚函数	多态实现不同敌舰的显示、移动和攻击行为
武器系统（子弹、鱼雷、导弹）	抽象基类、派生类重写	运行时多态与工厂模式思想
动态生成敌舰和武器	动态数组、new/delete 操作	内存管理与内存泄漏
游戏主循环与碰撞检测	循环、条件判断、坐标计算	复杂逻辑的组织与性能优化
补给包随机出现与拾取	随机数、范围判断	伪随机数生成与实际应用
关卡系统（1000 金币晋级）	状态机、枚举类型	状态转换与游戏流程控制
异常情况处理（越界、非法输入）	try/catch、自定义异常类	异常安全编程
轰炸机飞行路径	递归函数	递归思想与栈的理解

在能力层面，培养学生模块化设计、C++ 调试技巧、代码重构和文档撰写能力，培养软件开发逻辑思维与规范。提高 Ubuntu 环境下 C++ 程序开发与调试技能。参考刘佳芸等在 Java 实验教学中强调软件工程思想的渗透[14]，本课程同样要求学生在游戏开发中遵循模块化设计原则。

在素养层面，首先，游戏开发项目的趣味性与适度挑战性有助于激发学生的内在学习动机，使其从被动接受知识转向主动探究问题，形成持续投入的学习习惯。其次，学生在经历“需求分析—模块设计—编码实现—调试优化—作品展示”的完整开发流程后，能够获得强烈的工程成就感，这种正向情感体验将转化为后续学习的内在驱动力。再次，游戏项目要求学

生综合运用 C++ 的多项语言特性解决实际问题，有助于培养计算思维和系统设计能力。最后，作品的分享与展示环节能够增强学生的学术表达与交流能力，并为其后续学习专业课程奠定坚实的实践基础与心理准备，全面提升计算机专业综合素养。

3.2 教学实施

游戏开发贯穿于 C++ 教学全阶段。根据理论课教学内容，将海战游戏进行模块化划分。学生逐步实现相应模块，最终完成一个完整的海战游戏系统，并组织学生演示自己的作品。

第一阶段为需求分析设计。教师作为甲方，提出需求并讲解海战游戏规则；学生作为乙方完成设计文

档,占项目总分的 20%。理论教学进度为完成 C++基础语法。

第二阶段为搭建核心框架。教师提供 libncurses5 初始化和清屏辅助函数,降低入门门槛,实现控制台文本界面;实现 Game 类管理游戏主循环,并处理输入与状态;完成 Player 类,实现玩家的移动、射击和显示。占项目总分 40%。理论教学进度为完成 C++控制语句、动态数组、函数等。

第三阶段为构建敌方单位。Enemy 基类及派生类定义,完成所有敌舰类型、武器系统、攻击与胜负判定等;使用 vector 管理动态敌方单位和子弹,确保内存正确释放,占项目总分 40%。理论教学进度为 C++类、继承、多态、异常调试等。

第四阶段游戏拓展优化。鼓励学生实现扩展特性和个性化内容,包括补给包、颜色显示、随机障碍物、从文件读地图、多关卡难度提升等。此部分为加分项,最高 20 分。

4 研究方法

4.1 研究方法

为进一步检验游戏项目的区分度,采用极端组法,将项目得分前 27% (N=79) 与后 27% (N=79) 进行独立样本 t 检验。图 3 结果显示,高分组期末均分为 91.9±5.0,低分组为 85.9±8.1,差异具有高度统计学意义 (统计 p 值<0.001),效应量 Cohen's d=0.89。该结果表明,游戏项目的完成质量能够显著区分学生对 C++核心理论的掌握层次。

为了验证游戏开发任务与期末总成绩之间的关系,我们采用了 PEARSON 相关系数进行评估。评估结果发现,游戏项目分数与综合成绩之间存在较高的正相关 (相关系数为 0.41,统计 p<0.001)。由此说明,在真实场景游戏开发中投入更多、完成度更高的学生,对 C++核心理论知识的掌握也更加扎实。该结果支持“做中学”的教育理念,即通过解决真实、复杂的编程问题,学生能够更深刻地理解语言特性和设计思想。291 人平均分为 89.6 分 (SD=7.0),全部学生通过课程考核并获得了学分。超过 260 人取得了 80 分以上的综合成绩,占比 89.3%,对于后续专业课程的学习奠定了良好的基础。

本文进一步构建以游戏项目成绩 (X) 预测期末成 (Y) 的一元线性回归方程:

Y = 67.077 + 0.201 * X (1)

本文教学方案在深圳理工大学计算机科学与技术专业 2025 级程序设计语言 A (C++) 课程上实施,选课学生人数为 291 人,课程时间为 2025 年秋季学期。实验环境为 Ubuntu 24.04 LTS,编译器为 g++ 13.3.0,项目通过 Makefile 进行配置和管理。调试工具采用 gdb,内存调试工具采用 valgrind。每位学生的游戏作品由 2 名助教根据 3.2 中各模块分数占比进行评分,并取两者的平均值作为该生的最终评分。每个项目总评分为 120 分,其中 20 分扩展分。本研究采用单组相关实验设计,通过收集学生项目完成度与课程综合成绩 (由期中、期末理论考试成绩、编程作业成绩等构成),分析两者之间的相关性。

4.2 数据统计与分析

游戏项目得分和综合成绩情况如表 2 所示。291 名学生的项目完成度平均分为 111.6 分 (标准差 SD=14.3)。其中,超过 80 分人数为 272,占总人数比例为 93.5%,说明绝大部分学生能够较好的完成游戏基本任务;超过 100 分人数为 255,超过 87%的学生对于游戏开发项目具有较高的开发兴趣,完成扩展任务和个性化任务开发。

表 2 成绩分布情况表

	平均分	最低分	>80 人数	>100 人数
海战游戏分数	111.6 (SD=14.3)	60	272	255
综合成绩分数	89.6 (SD=7.0)	61.7	260	\
Pearson 系数	0.41 (P<0.001)	\	\	\

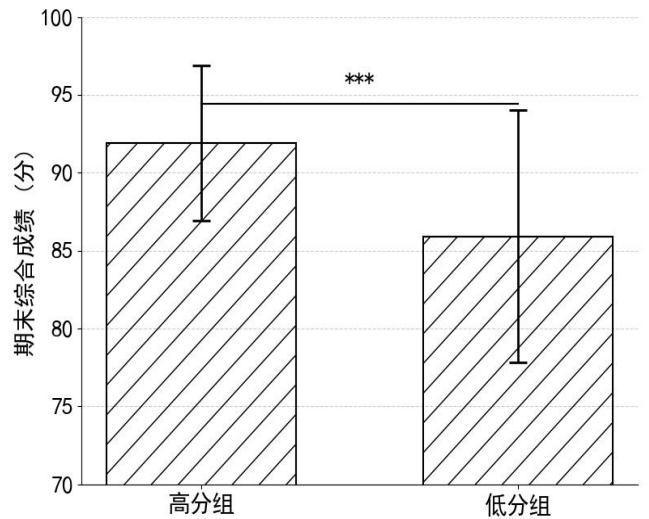


图 3. 高、低分组学生期末综合成绩均值比较 (**p<0.001)

这意味着游戏项目每增加 10 分,期末总成绩预期提升约 2.0 分。该结果量化了“做中学”的转化效率,为课程评分权重设置提供了实证依据。如图 4 所示,模型拟合优度 R²=0.172,回归系数显著 (t=7.74, p<0.001),反映了学生在该游戏开发项目中的投入与表现,与其对 C++核心理论知识的掌握程度之间存在

稳定的正相关关系。这验证了游戏驱动的编程实践具有可量化的正向迁移效应。

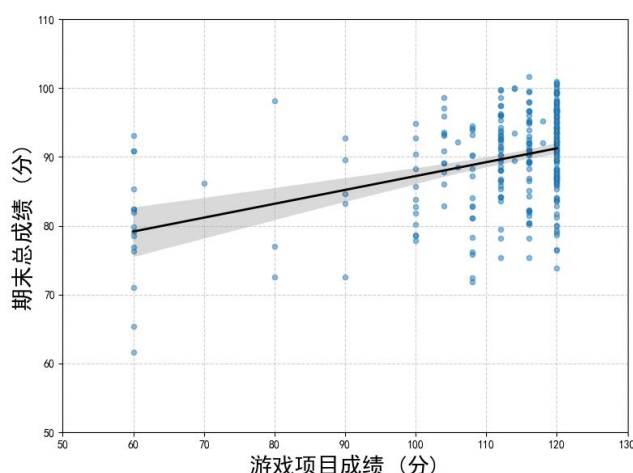


图 4. 游戏项目成绩与期末总成绩的散点图及线性回归拟合

4.3 研究局限性与替代验证策略

本研究采用了单组相关设计，未设置传统教学对照组，因此无法直接断言游戏开发教学在统计上优于传统教学模式，这是本文的一个研究局限。该设计的选择源于课程培养方案的刚性约束，即本课程为 2025 级必修核心课程，且目前为第二届开课，缺乏往届同质数据进行纵向对比。为在现有条件下最大限度弥补对照组缺失带来的论证短板，本文从以下两个维度提供替代性佐证。

首先是项目区分效度检验。尽管缺失对照组，但是极端组差异分析（图 3）表明，项目得分前 27% 与后 27% 学生的期末综合成绩存在显著差异（ 91.9 ± 5.0 比 85.9 ± 8.1 , $p < 0.001$, Cohen's $d = 0.89$ ）。这表明游戏项目的完成质量能够有效区分学生对 C++ 核心理论的掌握层次，具备优良的区分效度。同时，项目成绩与总成绩的中等程度正相关（ $r = 0.41$ ）进一步印证了项目任务与课程核心知识体系的内在一致性，说明该实践任务能够有效承载课程综合能力考核的功能。

第二是高阶学习行为佐证。本游戏项目设置了 20 分的扩展加分项，内容非强制完成。统计显示，超过 87%（255 人）的学生主动完成了该部分。在缺乏外部强制奖惩的情况下，如此高的拓展完成率客观反映了学生基于自主性与胜任感的内在动机投入。这一高参与度现象为自我决定理论在游戏教学情境下的适用性提供了间接的行为证据，也侧面印证了游戏驱动设计在激发持续学习投入方面的独特优势。

当然，本研究还存在另一不可忽视的外部干扰因素。随着 LLM 技术的不断进步，对于编程语言类课程带来了较大的冲击和影响。随着生成式 AI 的兴起，已有研究探索了 AI 驱动下的 Python 教学案例，如何在 C++ 教学中合理利用此类工具，已成为亟待解决的问题

[15]。存在部分学生使用 LLM 完成开发任务的情况，这也可能是导致游戏得分与综合成绩之间相关性不能进一步提高的原因。因此，在下一轮教学中将探索 LLM 作为“结对编程伙伴”或“智能辅导工具”整合进教学过程，并进行对比分析。如吴怀广等提出利用 LLMs 制造认知冲突[16]，这一思路可为本课程后续引入“结对编程伙伴”模式提供借鉴。此外，在未来工作中，将在下一届教学中开展包含游戏开发在内的不同类型项目的平行对照实验，为课程教学提供更多自主性元素，激发学生对于 C++ 编程语言学习的内在动机，也为验证游戏开发在 C++ 课程教学中的有效性提供技术支持。

5 结束语

本研究从情境学习、自我决定和认知负荷等理论视角，系统分析了游戏开发融入 C++ 教学的作用机制，并以海战游戏为代表案例进行了准实验验证。量化数据和定性反馈共同表明，游戏开发能够通过情境化学习、即时反馈和内在动机激发，显著提升 C++ 学习者的综合编程能力和学习动机，尤其是在面向对象和 STL 等中高阶知识点的掌握上具有明显优势。

本研究的主要贡献在于，理论上阐明了游戏开发促进 C++ 教学的多重作用机制；实践上，提供了可复用、经过验证的海战游戏教学框架和详细的教学设计；方法上，采用单组相关实验设计提供了初步的量化实证依据。因此，在程序设计教学中，游戏开发应作为综合实践的核心组成部分。让学生在创造游戏的快乐中掌握系统编程的语言，实现教育方法创新。

参考文献

- [1] 裴壮, 田秀霞, 李冰雪. 知识图谱赋能的面向对象程序设计 C++ 教学改革与实践[J]. 华东师范大学学报(自然科学版), 2024, 5: 104-113.
- [2] 傅佑铭, 梁超, 何政, 等. 程序设计课程教学双轮驱动新模式探索[J]. 计算机技术与教育学报, 2025, 13(3): 62-68.
- [3] 郭文佳, 傅向华, 吴庆阳, 等. 基于 CDIO 理念的 C++ 程序设计课程项目式教学实践[J]. 计算机技术与教育学报, 2021, 9(2): 69-73.
- [4] 孟文龙, 张策, 张小东. 新工科背景下“计算机程序设计基础”程教学改革探索[J]. 计算机技术与教育学报, 2024, 12(1): 85-90.
- [5] 康长青, 朱丽娟. 基于游戏化学习的程序设计实践课程改革[J]. 计算机教育, 2024, 8(356): 127-132.
- [6] 徐婉珍, 李强, 魏菊霞. 在游戏开发实践中培养程序设计能力[J]. 计算机教育, 2017, 266(2): 104-106.
- [7] 李国和, 董丹丹. 面向胜任力培养的程序设计综合实践教学探索[J]. 计算机技术与教育学报, 2023, 11(4): 127-132.
- [8] 汪淼, 钟娟, 孙全玲, 等. 案例-项目-实战: 师生协同构建教学闭环[J]. 计算机技术与教育学报, 2025, 13(6): 73-77.

-
- [9] 魏娟. 情境学习理论对信息管理专业人才培养的启示[J]. 计算机教育, 2008, 78(18): 47-48+44.
- [10] 周爱保, 杨小娥, 马小凤, 等. 学习方式的变革:提取促进学习[J]. 心理科学. 2017, 40(4): 913-919.
- [11] 杨弋, 荀启峰. 自我决定理论视域下的人工智能课程教学策略与实践[J]. 计算机教育, 2025, 12: 195-199.
- [12] 罗珣, 胡学钢, 方宝富, 等. 基于兴趣度-持久度的程序设计课程教学改革探讨[J]. 计算机技术与教育学报, 2021, 9(2): 80-85.
- [13] 辛自强, 林崇德. 认知负荷与认知技能和图式获得的关系及其教学意义[J]. 华东师范大学学报(教育科学版), 2002, 20(4): 55-60.
- [14] 刘佳芸, 张思佳, 胡泽元, 等. 融合软件工程思想的Java程序设计实验教学改革与探索[J]. 计算机技术与教育学报, 2025, 13(1): 76-81.
- [15] 郭曦, 王建勇. 生成式人工智能在Python教学中的作用与思考[J]. 计算机技术与教育学报, 2024, 12(2): 31-36.
- [16] 吴怀广, 王晓, 金松河, 等. LLMs赋能Python认知冲突的教学设计与实践探索[J]. 计算机技术与教育学报, 2025, 13(2): 79-83.