

游戏化策略对文科生编程能力与自我效能的影响

李珩, 黄璐

(重庆大学教师教学发展中心, 重庆 400030)

摘要: 为探讨游戏化策略对文科学生编程能力与自我效能感的影响, 提出一种游戏化学习活动促进模型。通过在一个学期内开展实验教学, 比较使用 CodeCombat 游戏化平台(实验组)和传统标准课程(对照组)学生的表现。研究表明, 实验组学生在编程能力与自我效能感方面提升显著, 两者间存在高度相关性, 为改善文科学生的编程教育提供了重要的指导和借鉴。通过深度整合游戏化学习理论、游戏化教学平台和教学实践, 为相关领域的教育研究提供了新的思路和方法。

关键词: 游戏化; 游戏化学习; 自我效能; 编程能力

DOI: 10.11907/rjtk.231637

中图分类号: G42

文献标识码: A

开放科学(资源服务)标识码(OSID):

文章编号: 1672-7800(2024)008-0301-07



Effects of Gamification Strategy on Liberal Arts Students' Programming Ability and Self-Efficacy

LI Heng, HUANG Lu

(Center for Enhancement Teaching and Learning, Chongqing University, Chongqing 400030, China)

Abstract: This study aims to investigate the impact of gamification strategies on the programming ability and self-efficacy of liberal arts students and proposes a gamified learning activity promotion model. Through conducting experimental teaching over one semester, the performances of the experimental group (using the CodeCombat gamified platform) and the control group (traditional standard curriculum) were compared. The results indicate that the students in the experimental group showed significant improvements in programming ability and self-efficacy, with a high correlation between the two. These findings provide important guidance and references for enhancing programming education among liberal arts students. This study deeply integrates gamification learning theories, gamified teaching platforms, and teaching practices, offering new insights and methods for educational research in related fields.

Key Words: gamification; gamified learning; self-efficacy; programming ability

0 引言

随着信息科技的快速发展, 就业市场对具备高级电脑技能的毕业生需求不断增加。市场分析显示^[1], 各行业在软件方面的投资大幅增加, 凸显了对学习编程的强烈需求^[1-3]。该趋势显示了社会对具备高级编程能力人才的迫切需求, 因此许多高等教育机构推出更具创新性的编程

课程。

如今, 中国本科教育中计算机编程已成为一门必修课程。2017年, 教育部正式提出建设新型文科专业的计划, 积极推动现代信息技术, 例如融合人工智能和大数据与文科专业、发展新兴文科专业、推动现有文科专业的转型升级。

然而, Suma等^[4-6]研究表明, 学习计算机编程需要进行更高层次的思维, 问题解决、规划和设计等认知过程在

收稿日期: 2023-06-19

扫描二维码阅读全文:



基金项目: 教育部新文科研究与改革实践项目(2021180027); 高等教育科学研究规划课题(2022); 重庆大学教育教学改革一般课题(2021Y57); 重庆市教育教学改革一般课题(213008)

作者简介: 李珩(1981-), 博士, 重庆大学教师教学发展中心副教授, 研究方向为教育信息化。

学习过程需要承担相当大的认知负荷^[7],因此编程课程被认为是最具挑战性的学科之一^[8-10],相较于其他学科而言辍学率和失败率普遍较高^[11-13]。

尽管,学习计算机编程在学生中越来越受欢迎,但事实证明掌握这一技能并非易事,需要付出大量努力。在影响学习成果的众多因素中,自我效能感被普遍认为是最为重要的因素,除了对具体编程知识的实质性测试,学生对自身能力和信心的认知也会对学习成果产生显著影响^[14-15]。因此,研究者们越来越关注如何提升学生的自我效能感,从而促使其在编程学习中的成功,这种关注也体现了心理和情感因素在教育中的重要性日益增强。

本文研究对象为重庆大学法学院的文科学生,在学习本专业必修课Python编程基础的过程中发现,大多数文科学生缺乏信息素养,导致其学习编程非常困难。为了激发文科学生学习编程的兴趣,以进一步提升学习效果,本文在游戏化学习理论的基础上提出一种游戏化学习活动促进模型,并结合一个具体的游戏化编程教学平台进行应用,结合游戏化学习理论与具体平台的应用和教学实践,将编程的抽象概念引入学生的“可见世界”,并利用游戏化的学习模式将基础编程知识具体化为各种任务^[16-19]。

首先,将不同的编程知识整合到平台的不同层次,让学生通过编写程序来消灭敌人、清除道路上的障碍物,并持续获取物体和武器等。其次,准备实验组和对照组,实验组通过游戏化方式进行编程学习,对照组则接受标准课程。最后,设计两个测试,通过综合考试评估编程知识和能力,另一个评估学生自我效能感。参与实验的两组学生在完成测试后,将对结果进行评估并得出相应结论,旨在通过游戏化教学方式促进文科学生的编程学习,评估其对学生编程知识、能力及自我效能感的影响。希望通过提出的游戏化学习活动促进模型和具体平台的应用,为教育实践提供有益的思路和方法,以适应当今编程学习的需求。

1 研究设计

1.1 框架设计

本文研究目的在于探讨游戏化学习策略对文科生编程能力及自我效能感的影响,以提升其编程水平和自我认知。为此,提出以下研究问题:①基于最终成绩,比较游戏化学习编程(实验组)与传统课程学习编程(控制组)之间的差异;②依据自我效能感反馈,分析实验组与控制组间的差异;③探究实验组学生的成绩与自我效能感之间的关联。研究框架如图1所示,包含前测、自我效能感评估、期末考试及课程结束时的访谈。

该研究于2022年春季在重庆大学进行,包含来自法学院的两个班级的64名学生,所有学生接受同一位教师教授的Python编程基础课程,共计17节课(16周教学与1周评估),每周教学3节课,每节课45min。在参与研究前,所

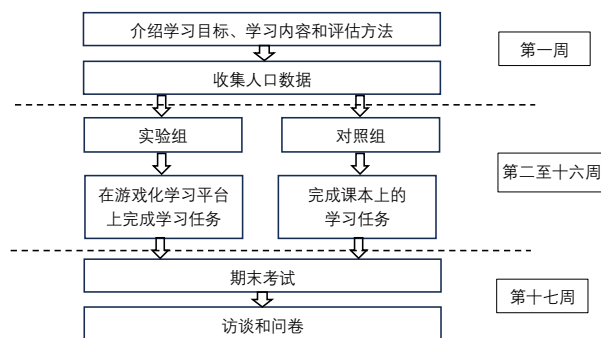


Fig. 1 Research framework

图1 研究框架

有学生均未接触过编程课程且不具备相似的编程背景与经验。其中一个班级(30名学生)被随机分配至实验组,而另一个班级则分配至控制组。

研究的第一周收集人口统计数据,包括性别、年龄、所在部门、家庭背景、计算机经验年限和计算机使用技能等信息。随后,实验组学生在教师的引导下,了解游戏化学习平台的基本功能,并通过该平台完成各个关卡的任务。对照组学生则学习相同课程,但被要求完成课本上的课堂作业。

在研究最后一周(第17周),所有学生参加期末考试,以考试成绩评估学生的编程能力,随后进行自我效能感的调查,以了解学生在学习编程过程中的主观感受。此外,本文还访谈了志愿者学生,以深入了解游戏化学习如何、为何提升了学生的编程能力和自我效能感。

本文研究基于为期17周的比较实验,旨在评估游戏化学习策略对文科学生编程能力和自我效能感的影响。通过评估编程能力与自我效能感,希望揭示游戏化学习能否有效提高文科学生的编程技能水平和自我认知水平。

1.2 游戏化教学设计

随着混合学习和远程学习的普及,学习者参与度变得至关重要,其中一种受欢迎的方法是采用游戏化学习来吸引学生。先前的教育研究和神经科学研究提供了全面的证据,表明游戏化学习可增强学习效果。研究人员对这种学习增强效应进行了综述,并认为游戏化学习通过情感(享受)、认知(认知负荷)和行为(愿意参与学习)机制来提升学生的学习表现和学习成果。

2021年Dicheva等^[20]提出一种设计高效的游戏化学习体验的方法,总结了游戏化教学设计中的关键要素和最佳实践,包括目标设定、规则设计、反馈机制等。2020年Xi等^[21]对游戏化与教育中的动机问题进行系统综述,探讨了不同游戏化元素和策略在激发学习动机方面的效果。2019年Carvalho等^[22]提供了关于游戏化教学设计的有效实践建议,例如选择适当的游戏化元素、平衡挑战和能力、提供有效的反馈等。这些研究丰富了游戏化学习设计的理论,但缺乏具体的教学实践验证其有效性。为此,本文在前期研究基础上,提出一个游戏化学习活动促进模型,

如图 2 所示。

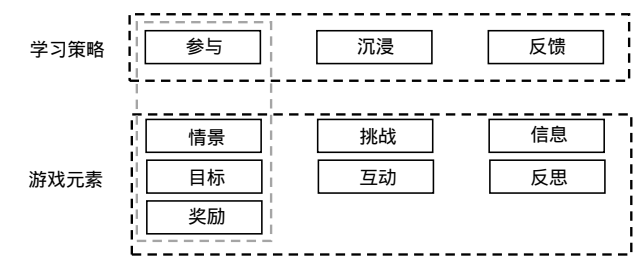


Fig. 2 Learning activities promoting model
图 2 学习活动促进模型

2 模型设计与效果评估

2.1 模型界面

模型由 3 个阶段组成,每个阶段都包含了若干游戏元素。为了实现游戏化的教学实践,本文选择通过冒险游戏学习编程的游戏化教育平台——CodeCombat,让学生通过掌握编码魔法来拯救他们的世界。具体为,综合任务包括 6 个计算机科学单元,基于项目单元来加强学习,学生将在从基本语法到递归的旅程中取得进步。

登录系统后,学生将进入一个特定关卡,关卡界面由 5 部分组成。其中,第一部分为关卡目标,描述每个关卡必须完成的任务;第二部分为代码预览区,学生可观察代码执行效果;第三部分为功能区,系统提供的各种功能;第四部分为代码编写区,学生在此编写 Python 代码;第五部分为关卡提示,如果学生在编程中遇到困难可点击此处查看提示。整个游戏分为计算机科学 1-6,共 214 个关卡,学生只有在完成当前关卡后才能进入下一个关卡,随着关卡深入,任务越来越复杂。

2.2 学习阶段

第一阶段采用了一系列游戏化要素来提升学生参与度,包括情境设定、明确目标和奖励机制。其中,情境设定通过创造故事背景,帮助学生更好地理解活动的实际意义;明确目标的设定有助于学习者明晰活动的目的和预期结果;奖励机制作为外部刺激,以激发学习者的外在动力,从而增强其参与意愿。在此阶段教师设置了一系列相对简单的任务,提供了丰富的提示和代码注释以引导学生顺利完成任务。学生完成任务后将获得相应的宝石奖励,这些宝石可用于购买新的装备。

沉浸阶段引入第二组游戏化元素——挑战和互动。其中,挑战是一系列难度逐渐增加的任务,学习者在满足基本要求后可逐步进行更高级别任务,通过学习者的能力范围设置活动,有助于产生沉浸感,促使学习者进入全神贯注的状态;互动包括人机互动和人际互动,在人机互动环节教师提供了逻辑挑战与组合挑战任务,要求学生编写完整的代码,综合运用所学知识。当学生在完成任务过程遇到困难时,可点击提示按钮进行人机互动,获取更多提

示减少挫折感。

反馈阶段信息和反思作为第三组游戏化要素。其中,信息是指一系列能帮助学习者顺利进行学习活动的因素,例如进度条、提示等;反思则是一系列能帮助学习者审视自身学习情况的因素,这些因素基于学习活动所得出的统计信息,例如准确率、成就获取、任务完成度等。通过反馈信息和反思,学习者可获得自身对整个学习过程的全面反馈,进而深入了解自身不足与优势所在,从而更好地开展学习活动以提升学习效率。

此外,教师在过程中可监测学生的背景数据并检查学习者的学习进展水平,对于愿意接受挑战但缺乏相应能力的学习者,教师提供一定的帮助既能减少其产生挫折感的可能性,还能保持学习动力,增强学生的沉浸感。在每个单元结束时教师开放战斗任务,让学生与同龄人进行代码竞技,以进一步增强互动,提升沉浸感。

2.3 效果评估

本文通过期末考试评估学生的编程能力,题目选自中国电子信息协会,包括选择题、填空题、简答题、代码分析题和程序设计题,涵盖了变量、循环、条件判断、字符串等编程知识,能考察学生对基本概念、语法、编程思维和问题解决能力的掌握程度。经过试测,这些题目在难易程度和判别性方面表现良好,具体考试双向细目如表 1 所示。

Table 1 Two way checklist
表 1 双向细目表

内容	填空	选择	简答	代码分析	程序设计	总分
	知识	知识、理解	知识、理解	分析	分析综合 应用评价	
数据类型	3	8	5	5		16
运算	3	5	5			12
条件	4	5	5	5	5	24
循环	3	5	5	5	5	32
函数	2	2	5		10	16
总分	15	25	25	15	20	100

为评估学生的自我效能感水平,在 Ramalingam 等^[23]设计的计算机编程自我效能量表(Computer Programming Self-Efficacy Scale, CPSES)的基础上进行调整,开发了一种用于评估 Python 编程自我效能的 PPSES 工具。该量表共包含 31 个项目,采用 7 点李克特式量表进行评估,分值从 1(完全不自信)到 7(非常自信),要求学生评估自己在实现编程入门的 5 个特定课程目标方面的程度:①分析、设计解决简单问题的方案;②制定处理较大问题的策略;③进行系统调试;④阅读、理解、修改他人编写的大量代码中的一小部分;⑤解释程序的结构与操作方式。

2.3 数据分析

本文使用 IBM SPSS 软件比较实验组与对照组学生的编程能力和编程自我效能感。首先将收集的数据进行描述性统计,分析组均值、数字和标准差等统计指标。其次,采用协方差分析(ANCOVA)检验游戏化学习对学生编程能力和编程自我效能感的影响,采用双变量线性相关分析探究

学生编程水平与自我效能感之间的相关性。最后,对访谈数据进行编码以挖掘学生的感知和体验,本文研究人员在阅读所有采访记录后在编码主题和引用方面达成了共识。

3 研究结果

3.1 编程能力

本文采用期末考试来测试编程能力,结果如表 2 所示。由此可知,数据样本显示实验组中高分(90 分以上)人数较多,对照组中更多学生的分数集中在平均水平以上。因此,初步推测实验组的得分可能优于对照组,但无法确定实验组的得分是否明显高于对照组。同时,实验组得分的样本数据明显偏离正态分布,因此在进行非参数统计分析时需要优先考虑准确性较高的方法。此外,对照组的性能数据也明显偏离正态分布,同样需要考虑采用非参数统计分析方法。

Table 2 Median analysis
表 2 中位数分析

对照组	实验组	差异值
77.50	84.50	7.00

对照组学生的成绩分布较为均衡,主要集中在中等水平并略有下降趋势,即有较多得分在 60–70 分的学生。实验组的平均得分为 83 分,对照组的平均得分为 76 分。实验组的得分波动范围主要在 78–94 分,对照组的得分则集中在 71–84 分,由此可见实验组结果更分散,对照组则更为集中。结合表 2 的中位数分析与 Wilcoxon 符号秩检验(见表 3)显示,在置信水平为 0.05 时,两组得分存在显著差异,由此可证明实验组的表现明显优于对照组。

Table 3 Wilcoxon signed ranks test
表 3 威尔科克森符号秩检验

项目	对照组与实验组差异
Z 值	-4.237 ^b
渐进显著性(双尾)	0.000

3.2 自我效能感

3.2.1 信度分析

量表的整体可靠性分析是使用 Cronbach 的 alpha 可靠性系数求得可靠性系数为 0.966。

3.2.2 效度分析

为了验证 CPSES 量表的结构有效性,首先进行巴特利特球度检验(KMO)和 Bartlett 检验。结果显示,KMO 系数为 0.94, Bartlett 检验中自由度为 465($p=0.000$),因此可对 31 个项目的量表进行因子分析。因子分析结果显示,31 个项目载荷在 0.448~0.651。

通过因子分析对问卷数据的整体有效性进行了验证,共提取了 6 个因子代表问卷中所有变量,分别为编程技能(因子 1)、Python 构造(因子 2)、简单性(因子 3)、时间意识(因子 4)、重新编码以更好理解的能力(因子 5)、自我激励(因子 6),如表 4 所示。

Table 4 Factor extraction
表 4 因子提取

因子	初始特征值			提取载荷平方和		
	总计	方差/%	累积/%	总计	方差/%	累积/%
1	17.329	55.899	55.899	17.329	55.899	55.899
2	3.635	11.726	67.625	3.635	11.726	67.625
3	1.656	5.342	72.967	1.656	5.342	72.967
4	1.414	4.562	77.529	1.414	4.562	77.529
5	1.153	3.719	81.248	1.153	3.719	81.248
6	1.084	3.496	84.744	1.084	3.496	84.744

3.2.3 相关性分析

本文计算了每个项目得分与自我效能感总得分间的相关性,以测试每个项目的普遍水平。相关系数用于衡量两个变量间程度和方向关系的指标,数值越接近 1 或 -1 代表相关性越强,表 5 汇总了每个项目的相关系数。由此可见,每个项目与自我效能感间均存在显著的正相关关系($p < 0.01$),代表每个项目均对整体评估提供了平衡和有意义的贡献。其中,相关系数在 0.501~0.866 之间变化,表明不同项目与自我效能感间的关联程度存在差异。

该发现也进一步支持了本文研究对衡量学生编程能力和编程自我效能感的有效性,通过分析每个项目与自我效能感的相关性,能全面了解不同项目与学生编程能力和自我效能感之间存在的关系,有助于评估学生在编程方面的水平与自我认知。

Table 5 Correlation analysis
表 5 相关性分析

项目	指标	相关系数
我可以编写语法正确的 python 语句	皮尔逊相关性	0.643**
	Sig.(双尾)	0.000
我能理解 python 的语法结构和保留字(如 def、and、for 等等)的用法	皮尔逊相关性	0.658**
	Sig.(双尾)	0.000
我可以使用 python 编写逻辑上正确的代码块	皮尔逊相关性	0.665**
	Sig.(双尾)	0.000
我可以编写一个 python 程序来显示问候消息	皮尔逊相关性	0.600**
	Sig.(双尾)	0.000
我可以编写一个 python 程序来计算三个数字的平均值	皮尔逊相关性	0.501**
	Sig.(双尾)	0.003

续表		
项目	指标	相关系数
我可以编写一个python程序来计算任意给定数字的平均值	皮尔逊相关性	0.560**
	Sig.(双尾)	0.001
我可以使用多种在python中可用的内置函数	皮尔逊相关性	0.570**
	Sig.(双尾)	0.001
我可以构建自己的python程序	皮尔逊相关性	0.762**
	Sig.(双尾)	0.000
如果碰到一个我熟悉的小问题,我可以写一个python小程序来解决	皮尔逊相关性	0.761**
	Sig.(双尾)	0.000
我可以编写一个适当大小的程序来解决一个对我来说有点困难的问题	皮尔逊相关性	0.814**
	Sig.(双尾)	0.000
我可以编写一个长而复杂的程序来解决任何给定的问题	皮尔逊相关性	0.803**
	Sig.(双尾)	0.000
我可以模块化的组织和设计我的程序	皮尔逊相关性	0.895**
	Sig.(双尾)	0.000
我明白什么是对象	皮尔逊相关性	0.737**
	Sig.(双尾)	0.000
我可以识别问题领域中的对象,并且可以声明、定义和使用它们	皮尔逊相关性	0.731**
	Sig.(双尾)	0.000
只要给出一个明确标记的函数声明,我就可以使用一个预先编写的函数。	皮尔逊相关性	0.755**
	Sig.(双尾)	0.000
我可以调试(纠正所有错误)我写的一个长而复杂的程序,并使它工作	皮尔逊相关性	0.866**
	Sig.(双尾)	0.000
我能理解一个长而复杂的程序	皮尔逊相关性	0.837**
	Sig.(双尾)	0.000
如果有人事先告诉我如何解决问题的方法,我可以完成一个编程项目	皮尔逊相关性	0.676**
	Sig.(双尾)	0.000
如果有语法参考手册的帮助,我可以完成一个编程项目	皮尔逊相关性	0.739**
	Sig.(双尾)	0.000
如果遇到困难可以打电话找人帮忙,我可以完成一个编程项目	皮尔逊相关性	0.707**
	Sig.(双尾)	0.000
只要有人帮助我开始,我就可以完成一个编程项目	皮尔逊相关性	0.674**
	Sig.(双尾)	0.000
如果我有足够的时间,我就能完成一个编程项目	皮尔逊相关性	0.863**
	Sig.(双尾)	0.000
如果我只有内置的帮助工具,我可以完成一个编程项目	皮尔逊相关性	0.735**
	Sig.(双尾)	0.000
在从事编程项目时,如果我陷入了困境,我可以找到克服问题的方法	皮尔逊相关性	0.848**
	Sig.(双尾)	0.000
我可以在短时间内为给定的编程项目找到合适的策略	皮尔逊相关性	0.859**
	Sig.(双尾)	0.000
如果程序项目截止日期非常紧迫,我可以有效地管理我的时间	皮尔逊相关性	0.813**
	Sig.(双尾)	0.000
如果交给我一个长而复杂的程序,我可以一直跟踪这个程序的执行过程	皮尔逊相关性	0.747**
	Sig.(双尾)	0.000
我可以重写代码的冗长和混乱的部分,以使其更容易阅读和清楚	皮尔逊相关性	0.756**
	Sig.(双尾)	0.000
即使周围有许多干扰物,我也能找到方法来集中注意力在我的程序上	皮尔逊相关性	0.747**
	Sig.(双尾)	0.000
我能找到激励自己去编程的方法,即使对问题领域不感兴趣	皮尔逊相关性	0.804**
	Sig.(双尾)	0.000
我可以编写一个其他人可以理解的程序,并在以后添加功能	皮尔逊相关性	0.850**
	Sig.(双尾)	0.000

注:**p<0.01。

3.2.4 辨别度分析

为了评估量表中各题项的区分度,首先对每个题项的原始得分进行从高到低排序。然后确定最高、最低得分群体,本文由最高得分的 27% 和最低得分的 27% 的学生组

成。接下来,使用独立样本 t 检验比较这两个群体的总分差异。针对 31 个题项的独立样本 t 检验结果显示,各差异水平非常显著($p < 0.001$),表明量表具有较高的项目判别力。

通过对高分群体和低分群体进行比较,能确定量表在不同学生群体中存在明显差异的项目,意味着这些题项在评估学生编程能力上具有较高的区分度,能有效识别能力较强或较弱的学生。如此,评估工具可深入了解学生的能力水平和个体差异,为后续教学和干预提供了有价值的依据。

3.2.5 自我效能感分析

表6为两组学生在编程自我效能感方面的结果。分析单因素方差发现,实验组的编程自我效能相较于对照组($p < 0.01$)提升明显。具体为,实验组的校正均值为72.39,标准误差为1.78;对照组的校正均值为64.46,标准误差为1.71。实验结果表明,游戏化学习方法在提高学生的编程自我效能感方面具有显著效果,进一步体现了游戏化学习方法对学生在编程领域自我效能感的积极影响。

Table 6 One-way ANOVA of programming efficacy between experimental and control group students

表6 实验组与对照组学生编程效能感的单向方差分析

Group	N	Mean	SD	Adjusted Mean	Adjusted SE	F
实验组	30	73.43	9.65	72.39	1.78	9.67**
对照组	34	64.25	12.87	64.46	1.71	

注:** $p < 0.01$ 。

综上,实验组学生在编程自我效能感上水平更高,证明了游戏化学习方法能激发学生学习动机与自信心,增强了学习者的编程技能与能力感知。

3.3 编程能力与自我效能感的关系

本文使用双变量线性相关分析,评估学生编程能力与自我效能感之间的关系,结果如表7所示。由此可见,实验组中编程能力与编程自我效能感呈现高度相关关系($r = 0.966, n = 30, p < 0.05$);对照组中编程能力与编程自我效能感存在高度相关关系($r = 0.890, n = 34$)。研究表明,学生编程能力与编程自我效能感之间存在紧密联系。在实验组和对照组中,学生编程能力水平与自身编程方面的能力感知密切相关。同时,高度正相关的结果表明,学生编程能力越强,其对自身编程能力的自我认知越高,这一关系对教育实践具有重要意义,提醒了教师在教学过程中既应注重培养学生的实际编程技能,也要关注其自我效能感的培养与提升。

Table 7 Analysis of the correlation between students' programming ability and self-efficacy

表7 学生编程能力与自我效能感的相关性分析

Group	N	Pearson Correlation coefficient(r)
实验组	30	.966*
对照组	34	.890*

注:** $p < 0.05$ 。

3.4 学生访谈

为了深入了解影响学生学习效果的因素,本文对实验组中的15名学生进行了一对一访谈,访谈问题包括本文所提方法相较于传统教学方式是否更好,何种因素对学习有帮助。

对于期末考试成绩高且自我效能感强的学生,研究发现这些学生能清晰地复述所学编程知识,认为编程学习是一种令人愉悦的经历,任务挑战激发了学生主动学习的意愿。随着任务复杂度增加,学生们对编程技能的信心不断增强,学生们在编程课程内外愿意接受更复杂的编程任务以挑战自己。此外,接受访谈的两名学生表示,通过提升编程技能可提升推理能力,使其更容易学习其他科目。

对于少数分数低且缺乏自我效能感的学生表现出以下特点:无法清楚地解释自己在编程任务中取得的成就;尽管这些学生对游戏化学习持积极态度,但在离开平台后,不知如何将所学编程知识应用于日常问题解决中。

在与对照组学生进行访谈时发现,部分学生在期末考试中成绩为中等水平,且对自身编程能力与自我效能感持有较低评价,虽然他们提到了在学习过程中遇到了各种问题和挑战,但并未及时得到解决,导致信心和动力下降,体现了对照组学生面临的困境,即学习难题的积累和对编程知识的应用不足。该研究结果进一步凸显了游戏化学习的重要性,即如何将游戏中的任务与现实世界的问题相结合,以促进学生学习兴趣和解决问题的能力。

综上所述,本文强调教育工作者应通过观察学生的学习表现和行为,尽可能地提供个性化支持和指导,及时帮助学生克服困难,重建他们的自我效能感。特别在提供编程任务时,应结合实际问题帮助学生进一步巩固所学知识。

实际上,在CodeCombat平台上教师可通过后台数据观察每个学生的代码完成时间和情况,对于那些需要花费大量时间仍未能完成任务的学生,如果能及时提供个性化指导,将有助于提升此类学生的自我效能感。

4 结语

现有研究表明,游戏化学习能显著提升文科学生的程序设计能力与自我效能感,并对其进行延伸发现显示游戏化学习对文科学生编程能力和自我效能感具有积极作用。通过访谈了解到,对于文科学生而言游戏化学习可提升他们的学习兴趣,帮助其取得更好的成绩并增强自我效能感。

然而,本文研究仍然存在一定的局限性。例如,研究仅在一所大学持续17周,应在更多样化的人群、更长时间周期或涵盖更多班级和专业范围等条件下进行实验验证。尽管在本研究中,实验组和对照组均由同一教师进行指导,但可能存在学生性别、先前知识和父母态度等其他因素对结果造成影响。

同时,本文课程设计基于教师和学生个人经验所开展,未来研究将根据反馈建议进一步改善教学设计,例如在不同层级间增加针对实际问题的编程练习。此外,本文建议在未来研究中考采用结对编程的方法,以提升学生

的计算机技能与自我效能感。

参考文献:

- [1] MACRIDES E, MILIOU O, ANGELI C. Programming in early childhood education: a systematic review[J]. *International Journal of Child-Computer Interaction*, 2022, 32: 100396.
- [2] LEE J, CHENG Y C. Change the face of software engineering education: a field report from Taiwan [J]. *Information and Software Technology*, 2011, 53(1): 51-57.
- [3] COLLINS A, SEDIG K. Challenges and opportunities in teaching programming to novice learners: a systematic literature review[J]. *ACM Transactions on Computing Education*, 2021, 21(1): 1-33.
- [4] SIVASAKTHI M, RAJENDRAN R. Learning difficulties of OOP paradigm using Java: students' perspective[J]. *Indian Journal of Science and Technology*, 2012, 4(8): 983-985.
- [5] NIKULA U, GOTEL O, KASURINEN J. A motivation guided holistic rehabilitation of the first programming[J]. *ACM Transactions on Computing Education*, 2011, 11(4): 1-38.
- [6] TAN P H, TING C Y, LING S W. Learning difficulties in programming courses: undergraduates' perspective and perception [C]// *International Conference on Computer Technology and Development*, 2009: 42-46.
- [7] GARNER S. Learning to program from Scratch[C]// *9th IEEE International Conference on Advanced Learning Technologies*, 2009: 451-452.
- [8] FANG X. Application of the participatory method to the computer fundamentals course, affective computing and intelligent interaction [J]. *Advances in Intelligent and Soft Computing*, 2012, 137: 185-189.
- [9] LAU W W F, YUEN A H K. Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy [J]. *British Journal of Educational Technology*, 2009, 40(4): 696-712.
- [10] MILNE I, GLENN R. Difficulties in learning and teaching programming—views of students and tutors[J]. *Education and Information Technologies*, 2002, 7(1): 55-66.
- [11] KORKMAZ Ö. Students' difficulties in and opinions about designing algorithms according to different instructional applications[J]. *Energy Education Science and Technology Part B: Social and Educational Studies*, 2013, 5(1): 209-218.
- [12] NILSEN H, LARSEN A. Using the personalized system of instruction in an introductory programming course[C]// *Norwegian Conference for Organizations' Use of IT*, 2011: 27-38.
- [13] CASPERSEN M E, KOLLING M. STREAM: a first programming process[J]. *ACT Transaction on Computing Education*, 2009, 9: 1-29.
- [14] ANASTASIADOU S D, KARAKOS A S. The beliefs of electrical and computer engineering students' regarding computer programming [J]. *The International Journal of Technology, Knowledge and Society*, 2011, 7(1): 37-51.
- [15] LUO J. A study on the influence of flipped classroom on the self efficacy of computer programming[J]. *Software Guide*, 2019, 18(5): 36-39.
罗剑. 翻转课堂对计算机编程自我效能的影响研究[J]. *软件导刊*, 2019, 18(5): 36-39.
- [16] ASKAR P, DAVENPORT D. An investigation of factors related to self-efficacy for Java programming among engineering students[J]. *The Turkish Online Journal of Educational Technology*, 2009, 8(1): 26-32.
- [17] ZHU X T, GE S L, WANG N X. Digital transformation: a systematic literature review [J]. *Computers & Industrial Engineering*, 2021, 162: 107774.
- [18] SUBHASH S, CUDNEY E A. Gamified learning in higher education: a systematic review of the literature [J]. *Computers in Human Behavior*, 2018, 87(1): 51873322.
- [19] DICHEVA D, DICHEVA C, AGRE G, et al. Gamification in education: a systematic mapping study[J]. *Educational Technology & Society*, 2015, 18(3): 75-88.
- [20] DICHEVA D, HSIN W J, DICHEV C, et al. Exploring the effect of virtual currency on learners engagement[C]// *2021 International Conference on Advanced Learning Technologies*, 2021: 83-87.
- [21] XI N, HAMARI J. Does gamification affect brand engagement and equity? a study in online brand communities [J]. *Journal of Business Research*, 2020, 109: 449-460.
- [22] CARVALHO C V D, BESSA P, COSTA R, et al. Learnit: a serious game to support study methods in engineering education[C]// *2019 IEEE Global Engineering Education Conference*, 2019: 959-967.
- [23] RAMALINGAM V, WIEDENBECK S. Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy [J]. *Journal of Educational Computing Research*, 1998, 19(4): 367-381.

(责任编辑:刘嘉文)