

基于图形化工具的编程教学促进初中生 计算思维发展的实证研究

傅 骞¹, 解博超², 郑娅峰³

(1.北京师范大学 教育技术学院, 北京 100875;

2.对外经济贸易大学 网络安全和信息化处, 北京 100029;

3.河南财经政法大学 计算机与信息工程学院, 河南 郑州 450046)

[摘 要] 计算思维已经成为中小学信息技术课程中核心素养涵盖的重要组成部分。基于编程工具培养中学生计算思维作为一种重要的教学手段得到了学术界的广泛认可。近几年来,图形化编程工具不断涌现,并大量进入 K12 教育阶段的课堂中。然而,使用图形化编程工具是否能够更好地提升学生的计算思维还有待通过实证研究进行验证。为此,研究通过对 45 名初中学生进行为期 16 个学时的教学实验,探究基于图形化编程工具和文本编程工具两种不同教学工具对计算思维培养的影响及其差异。研究结果表明,较之于传统文本编程环境,采用图形化编程工具的学生计算思维能力提升更为明显,并能够完成更为复杂的创意作品。这一研究发现将为中小学阶段采用图形化工具开展编程教育,培养计算思维能力提供有效的实践指导。

[关键词] 计算思维; 图形化编程; 问题解决能力; 编程教育

[中图分类号] G434

[文献标志码] A

[作者简介] 傅骞(1978—),男,浙江金华人。副教授,主要从事物联网技术及教育应用、创客教育等方面研究。E-mail:fredqian@bnu.edu.cn。郑娅峰为通讯作者,E-mail:zlzyf@126.com。

一、引 言

计算思维是人类经济和社会发展的一种基本思维方式^[1],是指个体运用计算机科学领域的思想方法,在形成问题解决的过程中产生的一系列思维活动^[2]。信息时代,计算思维已成为个体在复杂的技术文化中获得成功应具备的基本技能^[3]。研究指出,计算思维与学生的创造力、批判性思维、问题解决等诸多技能高度相关^[4]。因此,培养计算思维有助于提高学生信息技术知识与技能,培养学生跨学科综合问题解决能力^[5]。

当前,我国教育领域的计算思维研究与培养还处于初级阶段,主要对计算思维的特征、内涵、价值等进行理论探讨,并且多集中在高等教育领域^[6]。国外研究

则多在 K12 阶段通过引入计算工具,采用游戏化或情境化教学方式,在活动中融入计算思维思想来辅助学生的知识学习和问题解决,从而达到培养学生计算思维的目的^[6]。而在诸多提升计算思维的手段中,编程教育以其特有的逻辑思维 and 创新能力培养的潜能被国际教育界广泛研究。

我国研究者随之也逐步重视基于编程教育对计算思维的能力培养,开展编程教育成为培养学生计算思维的重要手段^[7]。尤其近几年我国政府大力倡导创客教育、STEM 教育等新型教育模式,使得作为主力军的编程教育在中小学阶段广泛开展^[8],图形化编程工具不断涌现,并成为中小学生学习编程教育的主要载体。诸多研究表明,图形化编程工具在降低学生认知负荷、提升学习兴趣以及提高编程思维等方面具有明显

基金项目:国家自然科学基金资助项目“基于教育大数据的在线协作讨论过程自动化分析与实时可视化呈现研究”(项目编号:61841702)

优势,更适合中小学生的认知水平^[5,9]。然而,编程教育作为计算思维培养的重要载体,虽然其有效性已经得到了诸多学术文章基于理论上的推演,但较少采用实证研究的方式检验。同时,尚未发现有研究者尝试比较图形化编程工具与传统文本编程工具对中小学生学习计算思维的提升作用。

因此,本文通过开展使用图形化编程工具和文本编程工具的编程教学,尝试对比二者在提升学生计算思维能力上的表现是否有显著差异。研究通过对初中学生开展为期16个学时的准实验研究,采用计算思维测试题、量表等工具对学生的计算思维进行更细维度的考察,探究基于图形化编程和文本编程工具在计算思维培养上的差异。同时,基于研究结果,给出具体的建议,为一线教学及计算思维研究的开展提供思路和借鉴。

二、文献综述

(一)计算思维培养

计算思维是一种解决问题的思维过程,指能够清晰、抽象地将问题和解决方案用信息处理代理(机器或人)所能有效执行的方式表述出来^[2]。开展计算思维教育有助于提高学生信息技术知识与技能,培养学生跨学科、综合应用学科知识解决问题的能力,提高学生的内驱力和创新力^[5]。

各国政府及教育界都对中小学阶段计算思维能力的培养给予了极大的重视。其中,英国于2014年9月推行了以培养计算思维能力为目标的计算机课程,美国于2014年将发展学生计算思维能力列入了高中计算机科学概论课程目标,我国在2017年公布的《普通高中信息技术课程标准》中也明确将“计算思维”素养的培养列入课程的核心素养。

与此同时,国内外大量研究者开展了对于K-12阶段计算思维培养的多维度探索。国外研究者在计算思维的相关探索中,较多地尝试将计算思维培养融入中小学的多学科教学中。如Swanson等人经研究发现,具有丰富计算环境的生物课程对高中学生的模型识别能力有较好的培养效果^[10];Wolz等人则通过与语言、艺术、数学、技术等学科教师的合作,将计算思维培养融入中学生交互式杂志设计项目中^[11];Sengupta等人提出将计算思维融合到中小学科学课程中,强调通过对建模和模拟的支持培养学生的计算思维,帮助中学生学习物理和生物,并通过实证研究来证明学习环境的有效性^[12]。与国外研究者侧重学科融合不同,国内研究者更侧重将计算思维与PBL、协作学习、游

戏教学等不同教学法结合,设计融入计算思维的教学案例或教育游戏^[13]。如生诗蕊构建了基于PBL的计算思维培养模型^[14];葛明珠将计算思维的教学理念与协作学习模式相结合,构建了CLMCT(Collaborative Learning Model Based on Computational Thinking)教学模型^[15];张立国等通过对计算思维不同概念界定的特征分析,提出了基于游戏化理念、问题解决理念、可视化理念的计算思维培养教学策略^[16]。

从这些研究可以看出,我国研究者当前关注的重点为构建能够有效培养计算思维的教学模式,通过以计算思维能力培养为核心的教学目标,尝试将多种不同的思维方法融合到具体的、多样的教学过程中,使得学生在多样的学习过程中获得计算思维的能力。

(二)基于编程教育的计算思维培养

计算思维能力的本质是将复杂的问题通过计算信息处理的方式进行表达的一种能力,因此,大量研究者将编程教育和基于技术支持的产品相结合,开展了对计算思维技能培养的研究^[6]。这些研究表明,编程教育是较为有效地促进计算思维能力提升的途径。

伴随技术领域的不断发展,中小学编程教育领域所使用的编程工具也从基于文本的传统形式发展到目前基于块的图形化编程形式^[17],如Scratch、APP Inventor、Mixly等工具被K12教育领域广泛接受。研究表明,图形化编程能够提供更为灵活的学习过程,从而帮助学生在保持其创造力^[18]。国内不同研究者也尝试开展基于图形化工具编程的计算思维培养探索。如赵兰兰开展了使用Scratch提升小学生计算思维能力的研究,通过分析不同类型代码块的个数证明图形化编程提升了学生的计算思维能力^[19];宁可为针对初中信息技术课程特点,利用编程工具App Inventor开展课堂教学,发现学生的计算思维能力及学习兴趣得到明显提升^[20]。

尽管编程教育对计算思维的重要性已经得到认同,然而在这一领域开展的实验研究仍显不足。Lye等人任在其调查研究中就指出,基于编程的计算思维培养的研究更多地体现在概念解释、文献综述,而缺乏实证的研究^[21]。另外,计算思维是一个综合的能力,包含创造、逻辑、抽象、算法、问题解决、综合、调试等多个维度^[18]。虽然一些研究表明图形化编程可以有效培养计算思维、促进问题解决,然而其与文本编程在培养计算思维更细维度上的差异,以及是否存在教学缺陷等相关实证研究还非常少。因而,本研究通过图形化工具与文本编辑工具教学的对比,探索使用图形化编程工具学习编程对初中生计算思维能力提升的影响。

三、研究设计

(一) 研究对象

研究对象为公开招募的 45 名初中学生, 来自北京地区多所中学, 平均年龄为 13 岁。本研究采用准实验法, 根据学生的年级、性别及是否学过程序设计进行随机分组。22 名学生为实验组, 采用图形化编程工具教学; 23 名学生为控制组, 采用文本编程工具教学。两组学生使用相同的教学方案。所有学生都能够熟练操作计算机。实验结束后, 共回收有效数据 39 份, 其中实验组回收 18 份, 控制组回收 21 份, 其余被试的数据因未能完成测试或完整参加整个教学环节而被舍弃。

(二) 研究工具

实验采用研究者自编的计算思维试卷(A、B 卷)作为测量工具进行前后测, 试卷中的试题全部选自 2015 和 2016 年《Bebras 国际计算思维挑战赛试题册》, 因而试题具有良好的区分度。为进一步刻画计算思维, 研究选用 Selby 等人于 2013 年提出的标准, 将计算思维水平进一步分解为抽象、算法思想、分解、评估和泛化这五个层面^[22]。因此, A、B 两套试卷各包含 6 道涉及思维能力的情景题目, 所选每道试题均覆盖抽象思维、分解思维、算法思维、评估思维和泛化思维五个思维层面中的两个或两个以上, 题目示例如图 1 所示。

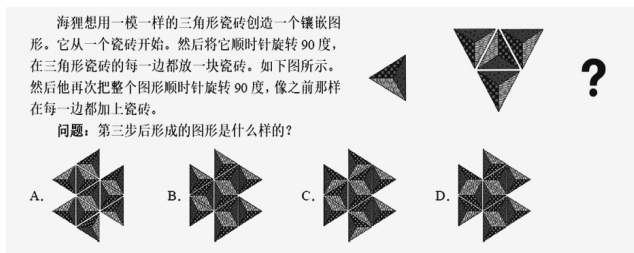


图 1 计算思维试卷试题示例

在编程工具的选择上, 实验组使用 Mixly for microPython, 该工具是由北京师范大学米思齐团队开发的基于块的图形化编程工具。工具将编程代码“包装”为可拖拽的块, 学生通过拖拽、拼接图形块的方式编写程序。控制组采用 Python Editor 文本编辑工具 (Mu), 该工具提供了自动缩进、自动补全、语法检查等功能, 降低了初学者发生格式和拼写错误的可能性。两组均使用 micro:bit 作为编程教学的硬件载体。Micro:bit 是一款 ARM 架构的单片机, 集成 5×5 LED 点阵、按钮、无线、加速度计和电子罗盘等, 主要用于青少年编程教育, 对于入门者而言是较易上手的编程硬件。

(三) 实验流程

实验流程主要分为计算思维前测、教学阶段、创意设计、计算思维后测四个阶段。

前测阶段: 在正式课程开始前, 实验组和控制组的学生作答计算思维试卷 A 卷, 时间为 30 分钟。该阶段的目的是测量两组学生初始的计算思维水平是否相当。

教学阶段: 实验组和控制组的学生分别参与 16 课时, 每两个课时完成一个教学单元, 两组均采用任务驱动的教学模式, 由同一名教师授课; 讲授内容相同, 涵盖程序设计基础知识和 Micro:bit 板载无线等电子元件的控制知识。

创意设计: 实验组和控制组的学生自选主题, 编写程序完成一个创意电子作品, 时间为 80 分钟 (2 学时)。

后测阶段: 实验组和控制组的学生作答计算思维试卷 B 卷, 时间为 30 分钟。

四、研究结果

(一) 实验组和对照组计算思维前后测结果比较

为研究实验前后学生计算思维水平的变化, 分别对两组学生计算思维水平的前后测进行分析。对于小样本研究, 异常值的影响尤其显著, 因此, 需检验数据中的异常值。实验组计算思维前后测差值正态检验显著性水平低于 0.05, 数据不服从正态分布, 因此, 使用 Wilcoxon 符号秩检验分析实验组计算思维前后测的差异, 数据分析见表 1。控制组计算思维前后测差值正态检验显著性水平平均高于 0.05, 认为数据服从正态分布, 使用配对样本 *T* 检验分析前后测的差异, 数据分析结果见表 2。数据结果表明, Wilcoxon 符号秩检验分析的显著性水平 ($p = 0.001$) 和配对样本 *T* 检验的显著性水平 ($p = 0.009$) 均低于 0.05, 表明实验后实验组和控制组的计算思维水平较实验前均有显著提高。说明无论采用图形化编程还是文本编程, 学生经过 16 学时的学习后, 计算思维都有较为明显的提升。

表 1 实验组计算思维前后测 Wilcoxon 符号

秩检验结果 ($N = 18$)

| 测试项 | 平均值 | 标准误 | Sig.(双侧) |
|--------------|------|--------|----------|
| 实验组计算思维前后测比较 | 0.89 | 16.545 | 0.001** |

注: * $p < 0.05$, ** $p < 0.01$, 以下同上。

表 2 控制组计算思维前后测配对样本 *T* 检验结果 ($N = 21$)

| 测试项 | 平均数 | 标准差 | <i>T</i> | Sig.(双侧) |
|--------------|------|------|----------|----------|
| 控制组计算思维前后测比较 | 0.86 | 1.35 | 2.91 | 0.009** |

(二) 两组学生学业水平后测结果比较

本研究项目的学业水平包含计算思维水平及创意电子作品质量两个纬度。为研究两组学生在编程学习前后计算思维水平是否均有提升, 研究使用 SPSS 分别对两组学生计算思维测试题(A、B 卷)的数据进行统计分析, 分别比较两组在前后测的表现是否存在差异。

本研究将被试分为初始能力相当的两组样本, 采取不同的干预手段, 得到两个相互独立的实验样本。由于样本容量均小于 30, 属于独立小样本, 因此, 采用 Shapiro-Wilk Test 进行正态检验, 显著性水平均大于 0.05, 表明实验组及控制组数据呈现正态分布。采用 Levene Test 进行方差齐性检验, 显著性水平均大于 0.05, 表明各组方差没有显著性差异。因此, 可以采用独立样本 T 检验对前测数据进行分析, 结果见表 3; 可以采用单因素方差分析对后测数据进行分析, 结果见表 4。

表 3 前测独立样本 T 检验结果

| 测试项 | 类别 | 人数 | 平均值 | 标准差 | T | Sig.(双侧) |
|------|-----|----|------|------|------|----------|
| 计算思维 | 实验组 | 18 | 3.50 | 1.34 | 1.58 | 0.123 |
| | 控制组 | 21 | 2.76 | 1.55 | | |

表 4 后测单因素方差分析检验结果

| 测试项 | 类别 | 人数 | 平均值 | 标准差 | 标准误 | F | η^2 |
|--------|-----|----|-------|------|------|--------|----------|
| 计算思维 | 实验组 | 18 | 4.39 | 1.04 | 0.24 | 15.35* | 0.41 |
| | 控制组 | 21 | 3.62 | 1.12 | 0.24 | | |
| 创意电子作品 | 实验组 | 18 | 20.28 | 2.32 | 0.55 | 8.69* | 0.44 |
| | 控制组 | 21 | 17.05 | 2.13 | 0.46 | | |

前测独立样本 T 检验结果表明, 实验前实验组和控制组学生在计算思维水平上没有显著差异 ($T = 1.58, p = 0.123$); 而后测单因素方差分析结果表明, 实验后实验组和控制组在计算思维水平 ($F = 15.35, p < 0.05, \eta^2 = 0.41$) 以及创意电子作品表现 ($F = 8.69, p < 0.05, \eta^2 = 0.44$) 上均存在显著性差异, 并且实验组的数据显著高于控制组。数据结果表明, 实验前两组的计算思维水平相当; 而从后测得分以及创意电子作品得分看, 实验组的表现明显优于控制组。

(三) 实验组与控制组创意电子作品中计算思维五维度差异对比分析

为研究实验后实验组和控制组学生的创意电子作品在计算思维的五维度上水平是否有差异, 将每个创意电子作品覆盖的条目所对应的计算思维概念进行计数, 并对计数结果进行描述性统计, 见表 5。

表 5 实验组和控制组创意电子作品在计算思维五个概念上的数量统计结果

| 计算思维概念 | 组别 | 项目数 | 平均数 | 标准差 | 标准误 | 最大值 | 最小值 |
|--------|-----|-----|-------|------|------|-----|-----|
| 抽象 | 实验组 | 11 | 12.64 | 1.32 | 0.36 | 17 | 9 |
| | 控制组 | 13 | 10.15 | 2.91 | 0.88 | 13 | 7 |
| 算法思想 | 实验组 | 11 | 28.18 | 1.91 | 0.53 | 32 | 23 |
| | 控制组 | 13 | 23.46 | 3.25 | 0.98 | 30 | 12 |
| 分解 | 实验组 | 11 | 5.55 | 5.67 | 1.57 | 7 | 4 |
| | 控制组 | 13 | 4.92 | 1.13 | 0.34 | 7 | 4 |
| 评估 | 实验组 | 11 | 8.64 | 1.12 | 0.31 | 11 | 6 |
| | 控制组 | 13 | 6.23 | 1.75 | 0.53 | 9 | 3 |
| 泛化 | 实验组 | 11 | 5.82 | 1.69 | 0.47 | 8 | 4 |
| | 控制组 | 13 | 5.31 | 1.08 | 0.33 | 8 | 4 |

由于使用 Shapiro-Wilk Test 检验该组数据不服从正态分布, 故采用 Wilcoxon 符号秩检验分析实验组和控制组的创意电子作品覆盖的计算思维概念数量是否有差异。数据结果见表 6。

表 6 实验组和控制组创意电子作品在计算思维五个维度上的 Wilcoxon 检验结果

| | 抽象 | 算法思想 | 分解 | 评估 | 泛化 |
|---------------------|--------|--------|--------|---------|--------|
| Mann-Whitney U 统计资料 | 38.00 | 36.50 | 49.50 | 24.00 | 52.50 |
| Wilcoxon W | 129.00 | 127.50 | 140.50 | 115.00 | 143.50 |
| Z | 1.962 | 2.042 | 1.329 | 2.784 | 1.134 |
| Sig.(双侧) | 0.051 | 0.041* | 0.184 | 0.005** | 0.257 |

数据结果显示, 实验组和控制组学生的创意电子作品在计算思维五个维度中的算法思想 ($Z = 2.042, p < 0.05$) 和评估 ($Z = 2.784, p < 0.01$) 这两个维度上出现了显著性差异, 且实验组得分高于控制组。其他三个维度虽未出现显著性差异, 但从平均分上看, 实验组均高于控制组。

(四) 创意电子作品得分与计算思维后测得分的相关分析

单纯使用一种测评方式很容易导致研究者对学生计算思维能力发展的理解出现偏差, 因此, 本研究通过计算创意电子作品得分与计算思维试题得分的一致性, 探究计算思维得分与学生创意电子作品的表现是否相关。数据结果显示, 创意电子作品得分和计算思维试题后测得分在 0.01 水平上存在显著的相关关系, 相关系数为 0.454, 见表 7。

表 7 创意电子作品与计算思维后测的相关分析

| | 创意电子作品得分 |
|----------|----------|
| 计算思维后测得分 | 0.454** |

五、讨 论

(一)图形化编程能够促进初中生计算思维能力培养

在当前的研究中,通过采用实证研究的方法对比不同编程工具对于初中生计算思维能力的影 响。数据结果表明,基于编程教育的教学方式可以有效提升学生的编程思维能力。这也与先前的研究所获得的结果较为一致^[21]。更进一步,数据结果表明,与文本编程相比较,采用图形化编程工具的教学无论在计算思维能力提升还是学生最终的创意作品质量表现上都存在显著性差异。这一结果也说明,对于初中生而言,采用更易于理解的图形化编程工具可以有效降低复杂代码编写所带来的认知难度,从而使得学生的注意力更关注于问题解决本身,最终使得基于图形化编程的教学更有助于提升学生的计算思维能力,以及更容易完成更为复杂的创意作品。进一步探讨其深层次的原因,这是因为传统文本编程语言虽然具有与计算机思维极其相似的表示形式,但其晦涩的语法、关键字拼写等细节常常加重了学生的认知负荷,使得学生产生畏难情绪。而图形化编程工具可以有效减轻学生的认知负荷,让学生“专注于编程所涉及的逻辑和结构,而不用担心编写程序的机制”^[23]。同样地,在 Weintrop 等人的研究中也发现,使用图形化编程的学生会表现出更强烈的学习欲望、更高的学习兴趣以及对未来学习更复杂内容的期待^[24]。因而使用图形化编程工具编程的学员获得了更大的计算思维能力提升。

(二)图形化编程工具更有助于学生完成高质量的创意作品

创意作品作为课程最终作业的评估,是对学生综合运用所学知识解决复杂问题的一个考察,也体现了学生对计算思维能力、问题解决能力等的具体实践应用。在本研究中,教师分别给两个组 80 分钟的时间完成创意电子作品的设计与开发。采用图形化编程工具的实验组平均用时为 48 分钟,而采用文本编程工具的控制组在 80 分钟时仅有不到一半的学生完成了程序的编写。两组在用时上的差异与 Garner 等人的研究结果^[25]相似。同时,统计结果也表明,实验组与控制组在创意作品质量上存在显著差异,实验组平均成绩远高于控制组。通过观察及课后访谈,这一情况的出现可能存在的原因如下:第一,使用文本编辑工具时频繁的报错和调试使得学生只能拼凑代码从而使得作品本身不够完整。如在访谈中,控制组的李同学说:“为了帮助我们独立完成创意电

子作品,教师在上课过程中特意教给了我们一些很实用的调试技巧。然而在实际编程的过程中,调试一些因为粗心而产生的低级错误会让我分心,有时候就忘了接下来该做什么了。”第二,图形化编程工具有助于学生将复杂的任务分解为更小、更易实现且更为明确的内部逻辑块。这也与 Maloney 的实验结论相一致。Maloney 在其研究中表明,图形化编程中的模块化可以降低学生发生语法错误的可能性,使得学生更关注于问题本身,而非不断地与程序是否能编译通过“做斗争”^[26]。更进一步,实验组使用了更为丰富的功能块,任务复杂度和完成性都更好,并且部分学生开始尝试使用更为优化的逻辑完成作品。这也说明,实验组的学生已经开始有意识地评估自己的代码,这也与数据结果显示基于图形化编程的实验组所表现的算法思想和评估这两个维度上都与控制组出现了显著性差异相一致。

(三)高水平的计算思维有助于学生完成高质量的创意作品

数据分析表明,创意电子作品得分和计算思维试题后测得分呈现显著差异。这一结果说明,具有较高计算思维能力的学生能够完成更复杂的创意电子作品。统计结果表明,在创意电子作品中,学生对于变量和功能的使用,实验组均多于控制组,说明实验组程序的复杂度比控制组要高。这一发现也进一步印证,计算思维作为综合能力的表现体现了学生的逻辑思维能力、抽象能力、算法思维、问题解决能力等^[18],可以被扩展为有利的技术方法,降低学生完成创新作品的门槛。因而,在目前的中小学编程教育课程中,应该将计算思维能力的培养作为核心目标,贯穿于课程教学的整个环节。研究同时表明,在中小学编程教育的初级阶段,引入图形化编程工具将有效提升计算思维培养的效率,从而使学生获得更好的学习体验和成果。然而值得注意的是,图形化编程工具作为初学者编程教学中的一个脚手架,不能做到像文本编程工具一样强大。因此,在教学中,待学生掌握了基本的编程知识且认知水平达到一定层次时,应慢慢撤去这个脚手架,过渡到文本编程环境中。同时,过于依赖工具而缺乏独立思考,反而会阻碍计算思维的发展,特别是抽象、泛化等能力的形成。

六、结论与展望

本研究开展了初中生采用不同编程工具培养计算思维能力的实验研究。结果表明,使用图形化编程工具和文本编程工具开展创意电子设计课程均可以

提高初中生的计算思维水平,且使用图形化编程工具提升效果更为明显。同时,高水平的计算思维可以帮助学生完成更高质量的创意作品。这一研究发现将为中小学阶段采用图形化工具开展编程教育课程提供有力的证据支撑。

然而,由于时间和样本所限,项目研究还存在一定的局限性。在未来的研究中,可以采用每周1~2学

时的间断授课方式,进行常态课堂的整班教学,从而明确图形化编程工具在真实教学场景下的效果。同时,由于图形化编程工具本身也有其局限性,学生进入高中阶段最终还是要学习文本编程。因而,在编程教育中,何时从图形化编程过渡到文本编程以及如何使学生将学习图形化编程的经验迁移到文本编程学习中,也是值得研究的问题。

[参考文献]

- [1] KAFAY B, BURKE Q. Computer programming goes back to school[J]. Phi delta kappan, 2013,95(1):61-65.
- [2] WING J M. Computational thinking[J]. Communications of the ACM, 2006,49(3):33-35.
- [3] BUNDY A. Computational thinking is pervasive[J]. Journal of scientific & practical computing, 2007(2):67-69.
- [4] BINKLEY M, ERSTAD O, HERMAN J, et al. Defining twenty-first century skills[M]. Netherlands: Springer, 2012:17-66.
- [5] 陈鹏,黄荣怀,梁跃,等. 如何培养计算思维——基于2006—2016年研究文献及最新国际会议论文[J].现代远程教育研究, 2018(1):98-112.
- [6] 范文翔,张一春,李艺. 国内外计算思维研究与发展综述[J]. 远程教育杂志, 2018(2):3-15.
- [7] ORR G. Computational thinking through programming and algorithmic art[C]//International Conference on Computer Graphics and Interactive Techniques, New York:ACM, 2009.
- [8] MARGOLIS J, GOODE J, BERNIER D. The need for computer science[J]. Educational leadership journal of the department of supervision & curriculum development N.e.a, 2011,68(5):68-72.
- [9] RESNICK M, MALONEY J, MONROYHERNANDEZ A, et al. Scratch: programming for all[J]. Communications of the ACM, 2009,52(11):60-67.
- [10] SWANSON H, ANTON G, BAIN C, et al. Computational thinking in the science classroom[C]//Proceedings of the international conference on computational thinking education. Hong Kong: The Education University of Hong Kong, 2017:28-33.
- [11] WOLZ U, STONE M, PEARSON K, et al. Computational thinking and expository writing in the middle school[J]. ACM transactions on computing education, 2011,11(2):1-22.
- [12] SENGUPTA P, KINNEBREW J S, BASU S, et al. Integrating computational thinking with K-12 science education using agent-based computation: a theoretical framework[J]. Education and information technologies, 2013,18(2):351-380.
- [13] 张慧. 教育游戏对初中生计算思维能力的影响研究[D]. 延边:延边大学, 2017.
- [14] 生诗蕊. 基于PBL的计算思维培养研究[D]. 锦州:渤海大学, 2016.
- [15] 葛明珠. 基于计算思维的协作学习模式在中学信息技术课程中的实践与研究[D]. 西安:陕西师范大学, 2014.
- [16] 张立国, 王国华. 计算思维:信息技术学科核心素养培养的核心议题[J]. 电化教育研究, 2018(5):115-121.
- [17] TOPALLI D, CAGILTAY N E. Improving programming skills in engineering education through problem-based game projects with scratch[J]. Computers & education, 2018,120:64-74.
- [18] DURAK H Y. and SARITEPECI M. Analysis of the relation between computational thinking skills and various variables with the structural equation model[J]. Computers & education, 2017,116(2018):191-202.
- [19] 赵兰兰. 运用Scratch软件培养中学生计算思维的研究[D]. 上海:上海师范大学, 2013.
- [20] 宁可为, 杨晓霞. 基于App Inventor的初中计算思维培养实证研究[J]. 课程·教材·教法, 2018(2):110-115.
- [21] LYE S Y, KOH J H L. Review on teaching and learning of computational thinking through programming: what is next for K-12?[J]. Computers in human behavior, 2014, 41(C):51-61.
- [22] Selby C C. Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy[C]//Workshop in Primary & Secondary Computing Education. New York:ACM, 2015:80-87.
- [23] KELLEHER C, PAUSCH R. Lowering the barriers to programming: a survey of programming environments and languages for novice programmers[J]. Acm computing surveys, 2005, 37(2):83-137.

- [24] WEINTROP D, WILENSKY U. Comparing block-based and text-based programming in high school computer science classrooms[J]. ACM Transactions on computing education, 2017, 18(1):1-25.
- [25] GARNER S. A quantitative study of a software tool that supports a part-complete solution method on learning outcomes[J]. Journal of information technology education, 2009(8):285-310.
- [26] MALONEY J, RESNICK M, RUSK N, SILVERMAN B & EASTMOND. The scratch programming language and environment[J]. ACM transactions on computing education, 2010, 10(4): 1-15.

Empirical Research on Programming Education Based on Graphical Tools to Promote Students' Computational Thinking in Junior High School

FU Qian¹, XIE Bochao², ZHENG Yafeng³

(1.Faculty of Education, Beijing Normal University, Beijing 100875; 2.Department of Information Management, University of International Business and Economics, Beijing 100029; 3.School of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou Henan 450046)

[Abstract] Computational thinking has become an important part of the core literacy of information technology curriculum in primary and secondary schools, and to cultivate students' computational thinking based on programming tools has been widely recognized. In recent years, graphical programming tools have emerged and entered into K12 classroom teaching. However, whether the use of graphical programming tools can better improve students' computational thinking still remains to be verified by empirical research. Therefore, this study explores the differences in the cultivation of computational thinking based on graphical programming tools and text programming tools through a 16-hour teaching experiment with 45 junior high school students. The results show that compared with the traditional text programming, the students who use graphical programming tools can improve their computational thinking ability more obviously and complete more complex creative works. The research findings will provide practical guidance for primary and secondary schools to use graphic tools to carry out programming education and cultivate computational thinking ability.

[Keywords] Computational Thinking; Graphical Programming; Problem-solving Ability; Programming Education