

계산적 사고 향상을 위한 창의적 스크래치 프로그래밍 학습

이은경[†]

요 약

최근 계산적 사고(Computational Thinking)의 중요성에 관한 인식이 확산됨에 따라 초중등학생들을 대상으로 계산적 사고 능력 향상을 위한 교육 연구가 활발히 이루어지고 있다. 본 연구에서는 스크래치를 활용한 창의적 프로그래밍 학습을 설계하고 해당 학습 활동이 중학교 1학년 학생들의 창의성 잠재력 및 계산적 사고 능력 발달에 미치는 영향을 확인하였다. 연구 결과, 학습자들의 창의성 잠재력은 유의하게 증진되었으며, 계산적 사고 능력의 토대가 되는 프로그래밍의 기본 개념과 원리를 습득하였음을 확인하였다. 특히, 창의성 잠재력의 하위 요인 중 창의적 성격 및 아이디어 생성 행동 요인에서 유의한 향상을 보였다. 계산적 사고의 경우 프로그래밍의 기본 개념인 순차, 반복, 조건, 이벤트 처리 및 연산자에 대한 개념 이해 수준이 높게 나타났다.

주제어 : 계산적 사고, 창의성 잠재력, 스크래치, 프로그래밍

Creative Programming Learning with Scratch for Enhancing Computational Thinking

Eunyoung Lee[†]

ABSTRACT

Computational thinking has been recently highlighted as an essential ability of the 21st Century so that many educational efforts have focused on broadening participation in computing and promoting computational thinking in K-12 settings. This paper describes the impact of creative learning activities with the Scratch on middle school students' computational thinking and creative potential. The learning activities were designed and implemented in 12 sessions with 34 middle school students. The pre and post creative potential assessment results show that students' creative personality and ideational behavior were significantly enhanced. Also, project portfolio analysis shows that students came to understand several computational concepts that are useful in a wide range of programming contexts: sequences, loops, conditionals, events, and operators.

Keywords : Computational Thinking, Creativity Potential, Scratch, Programming

[†] 중신회원: University of Pennsylvania Visiting Scholar
논문접수: 2012년 10월 17일, 심사완료: 2012년 12월 29일, 게재확정: 2013년 01월 11일

1. 서론

계산적 사고(Computational Thinking)는 복잡한 실세계의 문제를 다루기 위한 사고 과정이며 컴퓨터과학의 기본 개념과 원리에 기초한 문제 정형화, 문제 인식, 문제 분석, 해법의 구현 및 자동화 등의 요소를 포함한다[1][2][3][4][5]. 계산적 사고는 2006년 Wing에 의해 처음 제시된 이후, 지난 몇 년간 교육의 여러 분야에서 광범위하게 논의되어 오고 있으며, 프로그래머나 컴퓨터과학자 뿐 아니라 21세기의 모든 사람이 필수적으로 갖추어야 할 새로운 역량으로 간주되고 있다[6][7][8][9]. 특히, 미국의 경우, 계산적 사고 능력을 핵심으로 한 K-12 표준 컴퓨터과학 교육과정(CSTA K-12 Computer Science Standards)개발[10], 21세기를 위한 컴퓨터교육(Computing Education for the 21st Century)[11], 컴퓨팅 분야로의 광범위한 참여 증진(Broadening Participation in Computing)[12]등을 위한 국가 단위 지원 연구가 활발히 이루어지고 있다.

그럼에도 불구하고, 컴퓨팅 관련 분야는 여전히 청소년들의 일상생활, 진로, 관심 및 창의적 활동으로부터 단절된 것으로 인식되고 있다. 특히, 프로그래밍은 계산적 사고 발달을 위한 핵심 영역임에도 불구하고 여전히 특정 소프트웨어를 개발하기 위한 과정이며 프로그래머를 위한 영역이라는 편협한 인식으로 인해 학습자들은 프로그래밍 학습의 기회를 보장받지 못하고 있다.

프로그래밍 학습은 청소년들을 단지, 컴퓨터과학자나 프로그래머로 만드는 것이 아니라, 계산적 사고자(Computational Thinker)로 발전시키기 위한 것으로, 계산적 사고자는 일상생활 뿐 아니라 모든 학문 영역과 상황에서 요구되는 문제들을 해결하기 위해 계산적 사고를 적용할 수 있는 개인을 의미한다[13].

따라서 프로그래밍 학습은 학습자들이 창의적 생산자로서 실제적 컴퓨팅 창작물을 개발할 수 있도록 설계되어야 한다. 또한 각 창작물의 설계와 개발은 각 개인의 관심과 흥미를 반영한 창의적 활동과 연결되도록 설계되어야 하며, 무엇보다 어린 학습자들이 쉽게 접근할 수 있는 프로그래밍 환경을 제공하는 것이 중요하다.

스크래치는 모든 연령대, 다양한 관심사와 배경을 가진 모든 사람의 창의적 프로그래밍 활동을 지원하기 위해 개발된 프로그래밍 도구이다. 따라서 스크래치를 통해 누구나 쉽게 프로그래밍 학습이 가능하며, 개인의 창의적 아이디어를 반영한 상호작용 디지털 미디어 창작 활동이 가능하다. 뿐만 아니라, 스크래치 온라인 커뮤니티(<http://scratch.mit.edu>)는 전 세계 스크래치 개발자들이 프로젝트를 공유하고 협력, 토론 등의 상호작용이 가능한 환경을 제공한다. 즉, 스크래치를 활용한 창의적 프로그래밍 학습은 청소년들을 컴퓨팅 기술의 소비자가 아닌 창의적 생산자로서 컴퓨팅 영역에 접근할 수 있도록 지원함으로써 컴퓨팅 영역에 관한 청소년들의 진로와 관심을 유도할 수 있을 뿐 아니라, 창의성 및 계산적 사고 능력의 발달을 촉진시킬 수 있다[14]. 청소년들은 스크래치를 활용하여 구체적 창작물(artifacts)(예를 들어, 동적 상호작용 애니메이션, 동적 상호작용 게임, 스토리텔링 등)을 직접 설계하고 개발하는 창의적 생산자로서의 역할을 통해 계산적 사고 능력을 자연스럽게 습득할 수 있다.

본 연구에서는 스크래치를 활용한 창의적 프로그래밍 학습 활동을 구성하고 해당 학습 활동이 실제 학습자들의 창의성 잠재력 및 계산적 사고 능력 발달에 미치는 영향을 확인하고자 하였다. 창의성 잠재력은 현재 상황에서 실제적으로 관찰 가능한 창의적 수행 능력은 아니지만, 향후 교육과 훈련을 통해 개발, 발전될 수 있는 창의성에 대한 내재적 가능성을 의미한다[15]. 따라서 학습자의 창의성 잠재력을 확인하는 것은 창의성 향상 교육 설계를 위한 고려사항과 시사점을 제공한다. 이 점에서 매우 중요한 의의를 지닌다. 이를 위해, 12차시의 학습 프로그램을 개발하고 이를 2개교 중학교 1학년 학습자들을 대상으로 적용하였으며 다음 연구 문제 해결을 위한 자료 수집 및 분석을 실시하였다.

첫째, 스크래치를 활용한 창의적 프로그래밍 학습은 학습자의 창의성 잠재력을 향상시키는가?

둘째, 스크래치를 활용한 창의적 프로그래밍 학습은 학습자의 계산적 사고를 발달시키는가?

2. 관련연구

현대 사회의 청소년들은 기성세대에 비해 새로운 테크놀로지의 사용에 익숙한 편이다. 대부분의 청소년들은 채팅, 게임, 웹 검색, 자료 공유 등 온라인 환경의 디지털 미디어와 상호작용하는데 많은 시간을 소비하지만, 디지털 미디어의 창의적 생산 활동에 참여하는 경우는 많지 않다[14][16].

디지털 유창성(digital fluency)은 디지털 미디어와 상호작용하는 능력 뿐 아니라, 새로운 미디어를 설계하고 창작하는 능력을 포함한다[17]. 특히 동적 상호작용 미디어를 설계하고 창작하기 위해서는 프로그래밍 능력이 요구되며, 청소년들은 프로그래밍 학습을 통해 개인의 창의적 아이디어를 디지털 미디어로 표현하고 공유함으로써 개인의 정체성을 표현하는 방법을 습득하게 된다. 또한 프로그래밍 학습은 모든 분야의 문제 해결 및 시스템 설계 등에 유용한 계산적 사고의 발달을 촉진시킬 수 있다. 프로그래밍은 단지 특정 소프트웨어의 설계 과정이 아니라 문제 해결 과정의 외적 표현의 생성 과정이며 개인의 아이디어와 사고 과정에 관한 성찰의 기회를 제공하기 때문이다[18].

그러나 과거 프로그래밍 교육은 학습자 수준과 연령에 맞지 않은 어려운 프로그래밍 언어의 사용, 학습자의 관심이나 경험과 무관한 프로그래밍 활동 과제(예를 들어, 홀수의 합 구하기, 사각형 면적 구하기 등)의 제시, 학습 결과에 따른 적절한 안내와 피드백의 부재 등의 이유로 학습자의 관심과 진로를 유도하지 못하고 있다[14][19][20][21].

스크래치는 이러한 세 가지 과거 프로그래밍 교육의 문제점을 해결하기 위해 개발된 새로운 프로그래밍 학습 환경으로 다음과 같은 특성을 지닌다[14].

첫째, 누구나 쉽게 사용할 수 있다. 스크래치 프로그램의 생성은 특정 코드를 입력하는 방식이 아니라, 그래픽으로 구성된 프로그래밍 코드 블록들을 조립하는 과정을 통해 이루어진다. 학습자들은 스크래치 환경에서 제공되는 시각적 프로그래밍 코드 블록들을 스크립트 영역으로 가져와 순서에 따라 연결하는 간단한 동작을 통해 프로그

램을 완성할 수 있다. 특히, 각 프로그래밍 블록들은 문법적 구문에 맞을 경우에만 서로 연결되므로, 구문 오류에 대한 고려 없이 프로그래밍이 가능하다. 따라서 학습자들은 특정 프로그래밍 언어의 문법적 특성 및 사용법 습득이 아니라, 창의적 아이디어의 생성 및 문제 해결 과정 설계에 집중할 수 있다[20][21].

둘째, 다양한 유형 및 개인적인 관심과 경험을 반영한 프로젝트 생성이 가능하다. 학습자들은 자신에게 유의미한 프로젝트를 직접 생성하는 과정에서 최상의 학습을 경험하고 몰입하게 된다[13]. 스크래치는 학습자 개인의 관심 분야와 아이디어를 반영한 디지털 미디어 창작이 가능한 프로그래밍 환경을 제공한다[20]. 학습자들은 스크래치를 활용하여 다양한 유형의 프로젝트(예를 들어, 스토리텔링, 게임, 애니메이션, 시뮬레이션 등)를 개발할 수 있으며, 스크래치 프로그래밍 환경에서 통합적으로 제공되는 사진 촬영, 소리 녹음, 이미지 편집 기능을 이용하여 자신의 사진이나 목소리를 프로젝트에 쉽게 적용할 수 있다. 이러한 특성은 학습자들에게 프로젝트 개발을 위한 프로그래밍 활동에 동기를 부여하며 학습을 지속할 수 있게 한다.

셋째, 사회적 상호작용 활동이 가능하다. 스크래치 웹사이트(<http://scratch.mit.edu>)는 스크래치 개발자들의 사회적 상호작용 지원을 위한 온라인 커뮤니티로 학습자들은 자신의 프로젝트를 해당 웹사이트를 통해 공유하고 다른 사용자들의 비평가 의견을 수용하여 발전시킬 수 있다. 또한 다른 개발자들의 프로젝트를 자유롭게 재사용할 수 있을 뿐 아니라, 협력을 통한 프로젝트 개발이 가능하다[22].

스크래치의 이러한 기능적, 환경적 특성은 어린 학습자들을 소비자가 아닌 창의적 생산자로서 컴퓨팅 영역에 접근할 수 있도록 지원함으로써 컴퓨팅 영역으로의 진로와 관심을 유도할 수 있을 뿐 아니라, 창의성 및 계산적 사고 능력의 발달을 촉진시킬 수 있다.

스크래치 활용의 효과성 관련 연구들은 초중등 학생의 프로그래밍 수업에 스크래치를 활용하고 학습자의 창의성, 문제해결력, 동기 등에 미치는 효과를 제시하고 있다[20][21][23][24]. 특히, 창의

성과 관련된 연구 결과들은 스크래치 활용 학습이 학습자의 확산적 사고 향상 측면에 유의한 것으로 보고하고 있지만, 창의성의 속성 중 일부 요인에서만 유의한 차이를 보이고 있다[23][24]. 따라서 학습자의 창의성 잠재력을 평가함으로써 창의성 발현을 위한 강점과 약점을 파악하고 이를 토대로 교수 학습 전략을 개선해나간다면 스크래치 활용의 효과성을 보다 극대화할 수 있을 것이다.

3. 연구방법

3.1 연구대상 및 절차

본 연구는 ○○도 ○○군 지역 2개 중학교 1학년 34명을 대상으로 4주(12차시)에 걸쳐 실시되었으며, 세부 연구 절차는 다음과 같다.

먼저, 실험 처치 전 학습자의 창의성 잠재력을 확인하기 위한 사전검사를 10분간 실시하였다. 사전검사 실시 이후, 스크래치를 활용한 프로그래밍 학습 활동이 매주 3차시 4주에 걸쳐 진행되었다. 모든 학습자들의 프로젝트 결과물은 각 차시가 끝날 때 마다 온라인을 통해 공유, 수집되었다. 모든 학습이 종료된 직후, 창의성 잠재력 사후검사를 10분간 실시하였으며, 사전검사와 동일한 문항을 사용하였다.

3.2 자료수집 및 분석

3.2.1 창의성 잠재력 평가

스크래치 활용 프로그래밍 학습이 학습자의 창의성 잠재력에 미치는 영향을 확인하기 위해 사전, 사후 창의성 잠재력 평가를 실시하고, 평가 결과의 전후 비교를 위해 대응표본 t-검정을 실시하였다.

창의성 잠재력 검사를 위한 도구는 창의성 학습 평가 연구소(<http://www.clec.co.kr>)에서 개발한 검사지를 활용하였다. 해당 검사지는 창의성 발현과 관련된 창의적 사고유형, 창의적 성격, 아이디어 생성 행동을 측정하는 51개의 문항으로 각 문항은 5점 평정 척도로 구성된다. 가능한 점수 분포는 최고 255점에서 최저 51점이다. 구체적

인 문항 구성은 <표 1>과 같다.

<표 1> 창의성 잠재력 평가 문항 구성

요인	문항수
창의적 사고유형 (하위 5요인: 기능의 입법, 형식의 무정부, 수준의 전체, 범위의 외부, 경향의 진보)	15
창의적 성격 (하위 6요인: 상상, 심미, 감수성, 신기, 지성, 가치)	18
아이디어 생성 행동	18
계	51

3.2.2 계산적 사고 평가

스크래치 활용 프로그래밍 학습이 학습자의 계산적 사고 능력에 미치는 영향을 확인하기 위해 모든 학습자의 프로젝트 결과물을 수집, 분석하였다. 분석 기준으로 Brennan과 Resnick(2012)이 제시한 계산적 사고 능력 평가 틀 중 개념적 영역(Computational Concepts) 평가를 위한 요소를 도입하여 순차, 반복, 조건, 이벤트, 자료, 연산자의 총 6개의 기준을 설정하였으며[25], 각 기준에 따라 프로젝트 개발을 위해 각 개념을 정확히 이해하고 활용한 경우 1점, 그렇지 못한 경우 0점을 부과하였다. 가능한 점수 분포는 최고 6점에서 최저 0점이다.

4. 연구결과

4.1 창의적 프로그래밍 학습

스크래치를 활용한 창의적 프로그래밍 학습 활동은 4주(12차시) 분량으로 설계되었으며, 교수 학습 단계 및 활동 특성은 다음과 같다.

첫째, 학습의 단계는 과제 제시, 관련 지식과 경험의 활성화, 시연의 관찰, 새로운 지식과 기술의 적용을 통한 과제 해결, 과제 해결 경험의 공개적 시연의 단계로 이루어진다. 각 단계는 Merrill(2007)이 제시한 효율적인 교수 학습의 5가지 핵심 원리(과제중심, 활성화, 시연, 적용, 통합)를 토대로 설계되었다[26].

둘째, 학습자의 관심과 경험을 반영한 창작 활동의 기회를 제공하되, 적절한 안내와 피드백 제

공을 위해 정보와 과제가 결합된 형태의 학습 과제들을 설계하였다. 매주 다양한 형태의 프로젝트를 설계하고 개발할 수 있도록 구성하였으며, 각 프로젝트 유형은 다시 두 개 이상의 세부 학습과제로 구성된다. 학습과제는 이미 완성되어 있는 프로그램을 분석하고 따라할 수 있는 모델링 예제, 프로젝트 완성을 위한 단계별 절차와 정보가 포함되어 있는 과정 활동지, 예제와 정보가 포함되어 있지 않은 전통적 과제의 세 가지 유형으로 설계되었다. 모든 프로젝트 유형은 전통적 과제를 포함하고 있으며, 유형별 특성에 따라 모델링 예제 또는 과정 활동지가 포함된다.

셋째, 모든 학습과제들은 과제의 난이도와 지원 정도에 따라 순차적으로 제시될 수 있도록 설계하였다. 과제의 제시는 모델링 예제, 과정 활동지, 전통적 과제 순으로 제시함으로써 시간이 경과함에 따라 과제의 난이도는 증가하되, 교수자의 지원이나 관련 정보의 제공은 감소되도록 하였다.

전통적 과제 해결의 과정은 설계 기반 학습 원리(Design-based Learning)에 따라 설계(새로운 프로젝트의 생성), 개인화(개인적으로 의미 있는 프로젝트의 생성), 협력(협력을 통한 창의적 프로젝트의 생성), 반성(창의적 프로젝트의 비평가 재고)의 과정을 거쳐 이루어진다[13].

학습 활동 및 단계는 이영준과 이은경(2010)의 저서에서 제시한 스크래치 프로그래밍 프로젝트들을 토대로 구성하였으며[27], 구체적인 내용은 <표 2>와 같다.

<표 2> 창의적 프로그래밍 학습 활동

기간	프로젝트 유형	학습과제(과제유형)
1주	객체 동작 및 형태 제어	과제 1(모델링 예제): 움직이는 고양이 과제 2(과정 활동지): 유령 학교 과제 3(전통적 과제): 춤추는 발레리나
2주	상호작용 스토리텔링	과제 1(모델링 예제): 말하는 고양이 과제 2(과정 활동지): 나의 하루 과제 3(전통적 과제): 뮤직 비디오
3주	상호작용 애니메이션	과제 1(모델링 예제): 수족관 애니메이션 과제 2(전통적 과제): 슬라이드 쇼 애니메이션
4주	상호작용 게임	과제 1(과정 활동지): 물고기 잡기 게임 과제 2(전통적 과제): 재활용품 분리수거 게임

4.2 창의성 잠재력

창의성 잠재력 향상 정도가 통계적으로 유의한지 알아보기 위해 대응표본 t-검정을 실시하였으며 결과는 <표 3>과 같다.

<표 3> 창의성 잠재력 향상 정도

	사전 창의성 잠재력 - 사후 창의성 잠재력				
	M	SD	t	df	p
창의적 사고유형	-2.26	9.940	-1.328	33	.193
창의적 성격	-6.47	9.989	-3.777*	33	.001
아이디어 생성행동	-8.71	13.719	-3.700*	33	.001
총점	-17.44	25.653	-3.964*	33	.000

*p < .05, M:평균, SD: 표준편차

<표 3>에서와 같이 사후 창의성 잠재력 평가 점수는 사전 점수에 비해 유의하게 향상되었으나 (평균차=17.44, p=.000), 하위 요인별로 살펴보면, 창의적 성격(평균차=6.47, p=.001)과 아이디어 생성행동(평균차=8.71, p=.001)의 두 요인에서 유의한 차이를 보이고 있다.

창의적 성격 요인은 창의성 발현에 유리한 성격적 특성을 소유한 정도를 판별하기 위한 것으로, 특히 창의성과 관련된 지적 개방성(openness to experience)에 해당하는 6개의 하위 요인인 상상, 심미, 감수성, 신기, 지성, 가치로 구성된다. <표 4>는 창의적 성격의 하위 요인별 향상 정도를 나타낸 것으로, 상상(평균차=1.15, p=.011) 심미(평균차=1.68, p=.006), 가치(평균차=2.62, p=.000) 요인에서 유의한 향상을 보이고 있음을 확인할 수 있다.

<표 4> 창의적 성격 향상 정도

	사전 창의적 성격 - 사후 창의적 성격				
	M	SD	t	df	p
상상	-1.15	2.488	-2.688*	33	.011
심미	-1.68	3.355	-2.914*	33	.006
감수성	-1.00	3.172	-1.838	33	.075
신기	.35	3.034	.678	33	.502
지성	-.38	2.871	-.777	33	.443
가치	-2.62	2.861	-5.336*	33	.000
총점	-6.47	9.989	-3.777*	33	.001

*p < .05, M:평균, SD: 표준편차

아이디어 생성행동 요인은 창의적 사고의 산물인 아이디어를 산출해 내는 행동을 직접 측정하기 위한 것으로, 일상적인 환경에서 개인이 아이디어를 사용하고 평가하는 기술을 명확하게 나타내는 실제적 행동을 표현하고 있다. Plucker, Runco와 Lim(2006)은 이러한 아이디어 생성행동 요인이 확산적 사고를 구성하는 변인들과 유의한 관련성이 있음을 확인하였다[28].

창의성 잠재력 하위 요인 중, 아이디어 생성행동 요인이 다른 요인에 비해 가장 큰 유의한 향상을 보였다. 이는 스크래치의 도구적 특성 및 본 연구에서 설계한 학습 활동의 전략적 특성에 기인한 것으로 판단된다.

스크래치는 프로그래밍 기능 뿐 아니라, 다양한 멀티미디어 객체를 삽입, 생성, 변경할 수 있는 창의적 산물의 통합적 개발 환경을 제공한다. 예를 들어, 다양한 종류의 외부 이미지, 소리 자료들을 손쉽게 삽입할 수 있으며, 내장된 그래픽 편집기와 녹음기 등을 활용하여 프로그래밍 가능한 멀티미디어 객체를 생성할 수 있다. 따라서 학습자들은 프로그래밍 과정 중 자연스럽게 추상적 아이디어를 구체적 산물로 생성하는 활동을 수행함으로써 아이디어 생성, 구현, 평가의 과정을 반복적으로 경험하게 된다. 이러한 경험이 아이디어를 산출, 사용, 평가하는 기술을 명확하게 나타내는 실제적 행동 요인인 아이디어 생성행동 요인에 유의한 영향을 준 것으로 판단된다.

또한, 창의적 성격 요인 중 새롭고 재미있는 것을 상상하고, 미적인 것을 감상하고 즐기며, 가치 판단의 기준은 바뀔 수 있다고 판단하는 지적 개방성 요인이 유의하게 향상되었다. 이는 감수성, 신기, 지성 요인이 개인적 성향을 표현하는 반면, 상상, 심미, 가치 요인은 교육 활동이나 경험에 의해 변경될 수 있음을 시사한다.

창의적 사고유형의 경우, 사후 점수가 사전 점수에 비해 향상되었으나, 통계적으로 유의하지 않은 것으로 나타났다. 이는 창의적 사고유형의 증진을 위해서는 스크래치와 같은 창의적 활동이 가능한 학습 환경의 제공 뿐 아니라, 창의적 사고유형 연습을 위한 추가적인 교육적 처치가 필요하다는 것을 보여준다.

4.3 계산적 사고

스크래치를 활용한 창의적 프로그래밍 학습을 통한 학습자의 계산적 사고 발달 과정, 즉 프로그래밍의 기본 개념의 습득과 활용 능력을 확인하기 위해 학습자들의 프로젝트 결과물을 수집, 평가하였으며 결과는 <표 5>와 같다.

<표 5> 계산적 사고 능력 평가 결과

계산적 사고	최저점	최고점	N	M	SD
순차	0	1	34	.71	.462
반복	0	1	34	.68	.475
조건	0	1	34	.68	.475
이벤트	0	1	34	.68	.475
자료	0	1	34	.41	.500
연산자	0	1	34	.71	.462
총점	0	6	34	3.85	1.956

N: 사례 수, M:평균, SD: 표준편차

순차, 반복, 조건 처리는 프로그램의 동작 순서와 흐름 제어를 위해 필수적인 개념이다. 특히, 스크래치 프로그래밍 환경은 시각적인 객체의 동작(이동, 회전) 및 형태 변화(모양, 색상, 크기 등)를 손쉽게 제어할 수 있는 환경과 프로그래밍 블록들을 제공한다. 따라서 학습자들은 시각적 객체의 동작과 형태를 시간적 순서에 따라 제어하기 위해 프로그래밍 블록들을 직접 구성하고 실행 결과를 즉각적으로 확인할 수 있다. 즉, 프로그램 실행 결과 뿐 아니라 프로그램 코드를 동시에 시각적으로 확인 가능한 스크래치 프로그래밍 환경은 추상적 개념을 구체적 산물을 통해 관찰 가능하게 함으로써 학습자들의 추상적 개념 습득을 효과적으로 지원할 수 있다. 본 연구 결과는 비교적 짧은 시간임에도 불구하고 학습자들이 순차(평균=.71), 반복(평균=.68), 조건(평균=.68) 처리의 개념을 양호한 수준으로 습득하였음을 보여준다.

이벤트 처리는 상호작용 멀티미디어 개발을 위해 필수적인 개념으로, 동적 상호작용 애니메이션이나 게임 등의 설계에 활용된다. 스크래치 프로그래밍 환경은 마우스, 키보드 및 특정 센서를 활용한 외부 입력 값에 따라 객체의 동작을 손쉽게 제어할 수 있는 환경을 제공한다.

본 연구 결과는 학습자들이 이벤트 처리를 통한 동적 상호작용 개념을 양호한 수준(평균=.68)으로 습득하였음을 보여준다.

자료 처리는 자료를 저장하고, 사용하고, 갱신하는 과정들을 포함하며 추상적인 특성을 지닌다. 스크래치는 변수와 리스트의 두 가지 자료 유형을 설계하고 사용할 수 있는 환경을 제공한다. 본 연구 결과, 학습자들은 주로 게임 설계 시 점수를 생성하고 유지하기 위해 변수를 사용하는 것으로 드러났다. 그러나 자료 처리와 관련된 학습자들의 개념 이해 수준(평균=.41)은 다른 요소들에 비해 상대적으로 낮게 나타났다. 이는 스크래치를 활용한 학습이 학습자들에게 변수 사용에 관한 동기를 부여하고, 변수 사용의 목적을 이해시키는데 효과적임에도 불구하고, 자료 처리의 추상적 특성이 여전히 어려운 요소로 작용하고 있다는 것을 보여준다.

프로그래밍 과정에서 연산자는 특정 자료 값을 조정하거나, 조건 판단을 위한 논리식을 구성하는데 주로 사용된다. 스크래치는 산술, 논리, 비교연산을 위한 프로그래밍 블록 뿐 아니라 다양한 수학함수, 문자열 조정을 위한 연산자 블록들을 제공한다. 본 연구에서 학습자들은 주로 조건 판단을 위한 논리식 구성을 위해 연산자를 활용하는 것으로 드러났으며, 객체의 동작 범위를 임의로 지정하는데 난수 발생 함수를 사용하는 것으로 드러났다. 연산자와 관련된 개념 이해 수준(평균=.71)은 양호하게 나타났다.

계산적 사고는 디지털 미디어의 창의적 생산 활동을 통해 증진될 수 있다. 즉, 개인의 추상적 아이디어를 구체적 산물로 개발하는 과정은 추상화(abstraction)와 자동화(automation)로 대표되는 계산적 사고 과정이다[2]. 이러한 측면에서 프로그래밍 학습은 계산적 사고 과정 경험을 위한 핵심 영역이지만, 보다 효율적인 학습 지원을 위한 도구의 선택이 요구된다. 특히, 초보 학습자의 계산적 사고 능력 향상을 위한 프로그래밍 도구는 누구나 쉽게 시작할 수 있도록 쉬워야 한다. 그러나 복잡한 단계의 프로젝트 구현이 가능해야 하며 모든 종류의 프로젝트 구현이 가능해야 한다. 스크래치는 프로그래밍 초보 학습자를 위한 도구로 이러한 세 가지 조건을 갖추고 있으며, 본 연구

결과, 프로그래밍의 기본 개념 습득에 긍정적인 영향을 준 것으로 드러났다.

5. 결론

본 연구에서는 스크래치를 활용한 창의적 프로그래밍 학습이 어린 학습자들의 창의성 잠재력과 계산적 사고에 미치는 영향을 확인하고자 하였다.

스크래치는 어린 학습자 뿐 아니라 모든 연령대와 다양한 관심사와 배경을 가진 모든 사람의 창의적 프로그래밍을 지원하기 위해 개발된 프로그래밍 도구이다. 따라서 누구나 쉽게 사용할 수 있으며, 개인의 관심과 경험을 반영한 상호작용 디지털 미디어 생성을 지원한다. 또한 온라인 웹 사이트를 통해 전 세계 스크래치 개발자들이 생성한 프로젝트의 공유, 협력, 토론, 재생성 등이 가능한 환경을 제공한다. 스크래치의 이러한 기능적, 환경적 특성은 어린 학습자들의 프로그래밍에 관한 관심을 증진시키고, 누구나 쉽게 프로그래밍을 시작할 수 있을 뿐 아니라, 개인의 창의적 아이디어를 구체적 산물로 생성하고 표현할 수 있는 기회를 제공함으로써 기존 프로그래밍 교육의 문제점을 해결할 수 있다. 그러나 단지 도구의 전환이 창의성 및 계산적 사고와 같은 고차원적인 사고 능력의 발달을 보장하는 것은 아니다. 즉, 스크래치를 이용하여 보다 쉽고 다양한 프로젝트 개발이 가능한 환경을 제공하더라도 실제 학습자들의 지식과 기술 수준을 향상시키기 위한 교육적 접근이 필요하다.

따라서 본 연구에서는 스크래치 프로그래밍 환경의 효과를 최대화하고 높은 수준의 성취 경험을 보장하기 위한 학습 활동 및 전략을 설계하였다. 학습 활동은 다양한 유형의 프로젝트 생성을 지원하기 위해 객체 동작 및 형태 제어 뿐 아니라, 상호작용 애니메이션, 게임, 자료 처리의 다섯 가지 유형으로 구성하였다. 각 유형별 제시되는 학습 과제들은 낮은 수준에서부터 높은 수준으로의 성취를 보장하기 위해 모델링 예제, 과정 활동지, 전통적 과제의 순으로 구성하였으며, 학습자들은 마지막 단계에서 제시되는 전통적 과제를 해결하기 위해 모델링 예제와 과정 활동지 형태의 과제 수행을 통해 습득한 지식과 기술을 활용

하여 스스로 창의적 프로젝트를 설계하고 개발하게 된다. 해당 학습 활동은 2개 중학교 1학년을 대상으로 4주(12차시)에 걸쳐 진행되었으며, 학습자의 창의성 잠재력 및 계산적 사고 능력 발달에 긍정적인 영향을 준 것으로 확인되었다.

사전, 사후 창의성 잠재력의 향상 정도를 비교한 결과, 창의적 성격과 아이디어 생성행동 요인에서 유의한 향상을 확인하였다. 특히, 창의적 사고의 산물인 아이디어를 산출해 내는 행동을 직접 측정하기 위한 아이디어 생성행동 요인의 향상은 각 과제 수행을 통해 자신의 아이디어를 생성하고 생성한 아이디어를 실제 프로그램으로 구현하는 활동이 긍정적인 영향을 준 것으로 판단되며, 이는 교육적 처치를 통한 창의성, 특히 확산적 사고의 향상 가능성을 시사한다.

또한, 해당 학습 활동을 통한 계산적 사고의 발달을 확인하기 위해 학습자 프로젝트 결과물을 수집, 분석하였다. 분석 결과, 순차, 반복, 조건, 이벤트 처리 및 연산자의 이해와 사용에 있어 약 70% 정도의 높은 성취 수준을 보였다. 그러나 변수, 리스트와 같은 자료 처리의 개념 이해도의 경우 50% 이하의 낮은 성취 수준을 보였다. 이는 자료 처리의 추상적 특성이 어린 학습자들에게 여전히 어려운 요소로 작용하고 있음을 나타낸다.

연구 결과에 따른 향후 연구를 위한 제언은 다음과 같다.

첫째, 창의성 및 계산적 사고 향상을 위한 학습 개선 전략에 관한 추가적인 연구가 필요하다. 효과적인 프로그래밍 학습을 위한 세 가지 조건(쉽게 시작하기, 높은 단계의 성취 경험하기, 다양하고 유의미한 프로젝트 개발하기)을 모두 갖추기는 어렵지만[18], 도구의 개선과 더불어 교육적 전략에 관한 연구가 이루어진다면, 어린 학습자들을 위한 보다 나은 프로그래밍 학습 환경을 제공할 수 있을 것이다.

둘째, 계산적 사고 능력 평가를 위한 전략에 관한 연구가 필요하다. 계산적 사고를 평가하기 위한 명확한 평가 틀이나 전략에 관한 합의는 아직 미흡한 실정이며 학습 상황별 평가에 관한 연구들이 이루어지고 있다. 그러나 계산적 사고의 평가를 위해 개념 요소의 측정 뿐 아니라, 실제 문제 해결 상황에서 개념을 적용하고 수행하는 과

정 및 컴퓨팅에 관한 견해를 종합적으로 측정할 수 있는 전략이 요구된다.

참고 문헌

- [1] 이영준 · 이은경 (2008). 정보교육의 본질과 전망. *컴퓨터교육학회논문지*, 11(3), 1-11.
- [2] 이은경 (2009). *Computational Thinking 능력 향상을 위한 로봇 프로그래밍 교수 학습 모형*. 박사학위 논문, 한국교원대학교.
- [3] ISTE & CSTA (2011). *Operational Definition of Computational Thinking for K-12 Education*.
- [4] Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49, 33-35.
- [5] Wing, J. (2010). *Computational thinking: What and Why?* Available online at <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>, Accessed August 20, 2012.
- [6] Henderson, P, B. (2009). Ubiquitous computational thinking. *Computer*, 42(10), 100-102.
- [7] Lee, I. et al. (2011). Computational Thinking for Youth in Practice. *ACM Inroads*, 2(1), 32-37.
- [8] Lu, J. J. & Fletcher, G. H. L. (2009). Thinking about computational thinking. *ACM Special Interest Group on Computer Science Education Conference, (SIGCSE 2009)*, Chattanooga, TN, USA.
- [9] NAS (2011). *Report of a workshop on the scope and nature of computational thinking*. Washington DC: National Academies Press.
- [10] Seehorn, D. et al. (2011). *K-12 Computer Science Standards Revised 2011*. ACM & CSTA.
- [11] NSF (2012). *Computing Education for 21st Century (CE21)*. Available online at <http://www.nsf.gov/pubs/2012/nsf12609/nsf12609.pdf> Accessed September 10, 2012.
- [12] BPC Portal (2009). *Broadening Participation in Computing (BPC)*. Available online at <http://bpcportal.org>. Accessed October 10, 2012.
- [13] Brennan, K. (2011). *Creative Computing: A design-based introduction to computational*

- thinking*. Available online at <http://scratched.media.mit.edu/sites/default/files/CurriculumGuide-v20110923.pdf>, Accessed September 12, 2012.
- [14] Resnick, M. et al. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11), 60-67.
- [15] Runco, M. A. (2004). Everyone has Creative Potential. In Sternberg, R. J., Grigorenko, E. L. and Singer, J. L.(Eds.), *Creativity: From Potential to Realization*(21-30, 147-148). American Psychological Association, Washington. DC.
- [16] Prensky, M. (2001). *Digital natives, digital immigrants*. On the Horizon, 9(5), 1-6.
- [17] Resnick, M. (2002). *Rethinking Learning in the Digital Age*. Available online at <http://llk.media.mit.edu/papers/mres-wef.pdf> Accessed August 22, 2012.
- [18] diSessa, A. (2000). *Changing Minds: Computers, Learning, and Literacy*. MIT Press, Cambridge, MA.
- [19] Guzdial, M. (2004). Programming environments for novices. In S. Fincher and M. Petre(Eds.), *Computer Science Education Research*(127-154). Taylor & Francis, London, UK.
- [20] 이은경 · 이영준 (2008). Scratch 활용 프로그래밍 교육이 중학생의 몰입수준과 프로그래밍 능력에 미치는 영향. *중등교육연구*, 56(2), 359-382.
- [21] 배학진 · 이은경 · 이영준 (2008). 문제 중심 학습을 적용한 스크래치 프로그래밍 교수 학습 모형. *컴퓨터교육학회논문지*, 12(3), 11-22.
- [22] Brennan, K., Resnick, M., & Monroy-Hernandez, A. (2011). Making projects, making friends: Online community as catalyst for interactive media creation. *Journal of New Directions of Youth Development*, 128, 75-83.
- [23] 오정철 · 이지훤 · 김정아 · 김종훈 (2012). 스크래치를 활용한 STEAM 기반 교육 프로그램 개발 및 적용-초등학교 6학년 과학교과를 중심으로. *컴퓨터교육학회논문지*, 15(3), 11-23.
- [24] 전성균 · 이영준 (2012). 초등학생의 확산적 사고 촉진을 위한 CPS 프로그래밍 수업의 효과 분석. *컴퓨터교육학회논문지*, 15(2), 1-8.
- [25] Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *The Annual meeting of the American Educational Research Association*, Vancouver, BC, Canada.
- [26] Merrill, M. D. (2007). First Principles of Instruction, *ETR&D*, 50(3), 43-58.
- [27] 이영준 · 이은경 (2010). **상상의 실현을 위한 Scratch programming**. 서울:교학사.
- [28] Plucker, J., Runco, M. A., & Lim, W. (2006). Predicting Ideational Behavior From Divergent Thinking and Discretionary Time on Task, *Creative Research Journal*, 18(1), 55-63.



이 은 경

1998 한국교원대학교
컴퓨터교육과(교육학학사)

2005 한국교원대학교
컴퓨터교육과(교육학석사)

2009 한국교원대학교 컴퓨터교육과(교육학박사)
2012~현재 University of Pennsylvania

Visiting Scholar

관심분야: 컴퓨터교육, 학습과학

E-Mail: eunklee@upenn.edu