

초중등학습자의 특성에 따른 텐지블 프로그래밍 성취도 차이분석

심재권[†] · 이원규^{††} · 권대용^{†††}

요 약

컴퓨팅 교육에서의 주된 목적은 컴퓨팅 사고력(Computational Thinking)의 향상으로 컴퓨팅 사고력의 핵심적인 요소는 추상화(Abstraction)와 자동화(Automation)이다. 초중등학생을 대상으로 하는 컴퓨팅 교육에서는 주로 알고리즘 설계 학습을 목적으로 프로그래밍 활동을 제공하고 있다. 본 논문에서는 초중등 학습자가 텐지블 프로그래밍 도구를 사용하여 알고리즘을 학습한 결과를 분석하였다. 분석은 초중등 학습자의 특성인 학교급과 성별에 따라 성취도와 사용성에 차이가 있는지 분석하였다. 분석결과, 학년에 따라 추상화 능력은 비례하지 않는 것으로 분석되었고, 성별에 따라 텐지블 도구의 사용성에 차이가 있는 것으로 분석되었다.

주제어 : 프로그래밍 교육, 알고리즘 설계, 컴퓨팅 사고력

Analysis of the Difference of Tangible Programming Achievement on Learner' s characteristics

Jaekwoun Shim[†] · WonGyu Lee^{††} · DaiYoung Kwon^{†††}

ABSTRACT

The main purpose of computing education is to teach the Computational Thinking which is based on Abstraction, Automation and etc. In Elementary and Secondary Education, programming activity is provided for the purpose of learning algorithm design. In this paper, the achievements of learning concepts of algorithm design are analyzed on the learner's characteristics in the programming activities using the Tangible programming tools for elementary and secondary school students. As a result, the achievement did not show the difference on grades but the usability showed the difference on genders.

Keywords : Programming Education, Algorithm Design, Computational Thinking

† 정 회 원: 고려대학교 컴퓨터교육학과
 †† 종신회원: 고려대학교 컴퓨터학과
 ††† 종신회원: 고려대학교 컴퓨터교육과(교신저자)
 논문접수: 2015년 10월 6일, 심사완료: 2015년 11월 5일, 게재확정: 2015년 12월 21일
 * 본 논문은 2014년 한국연구재단 이공분야기초연구사업의 지원으로 수행되었음(NRF-2013R1A1A2061352)

1. 서론

정보사회에서 정보기기를 활용하여 효율적으로 문제를 해결 할 수 있는 능력은 누구나 갖추어야 할 기본적인 소양이다[1]. 따라서 세계적으로 초중등 교육과정에서 컴퓨팅기반 문제 해결에 핵심이 되는 프로그래밍 관련 교육을 필수적으로 하도록 구성하고 있으며 나아가 소프트웨어 역량 위한 교육도 강화하고 있다[2][3]. 특히 최근의 프로그래밍 교육은 프로그래밍 도구의 사용법과 활용 방안을 바탕으로 컴퓨팅 기반의 문제 해결력 향상을 위한 사고력 중심의 교육을 목표로 하고 있으며 대표적으로 컴퓨팅 사고력(Computational Thinking) 향상을 제시하고 있다[4].

컴퓨팅 사고력(Computational Thinking)에서는 핵심적으로 추상화(Abstraction)와 자동화(Automation)를 제안하고 있다. 추상화는 주어진 문제나 업무에 관한 데이터 추상화(Data Abstraction)와 절차 추상화(Procedure Abstraction)로 구분할 수 있다. 데이터 추상화는 현실세계의 대상과 개념을 주어진 목적에 적합한 데이터의 속성(Attribute)과 관계(Relation)를 도출하여 컴퓨팅 환경에서 사용할 수 있는 데이터 모델을 구축하는 과정을 의미한다[5]. 절차 추상화는 문제해결 과정을 컴퓨팅 형태로 결과를 도출하거나 시뮬레이션 할 수 있도록 알고리즘을 설계하고 모델링하는 과정을 의미한다[6]. 자동화는 추상화 과정을 통해 정의된 데이터와 설계한 알고리즘을 프로그래밍 언어를 사용하여 최종적으로 프로그램(소프트웨어)으로 구현하는 과정을 의미하며 프로그래밍 활동을 통해 명확하게 학습할 수 있다.

컴퓨팅 환경에서 주어진 문제를 해결하거나 업무 목적을 자동적으로 처리하기 위해서는 알고리즘 설계가 필수적이며 우선적으로 요구된다[7][8]. 문제 상황에 적합하고, 업무에 효율적인 알고리즘을 설계하기 위해서는 순차적인 처리, 조건에 의한 분기와 반복적인 처리에 대한 개념을 논리적으로 구조화하여 절차적으로 추상화할 필요가 있다[9]. 이후 자동화를 위한 프로그래밍은 설계한 알고리즘을 코드화 하는 것으로 구현 환경에 대한 지식과 경험이 많이 요구 되는 작업이기 때문에 학습자에게 매우 어려운 과정으로 인식되고

있다[10]. 따라서 초중등 학습자를 대상으로 하는 프로그래밍 활동은 절차적 추상화에 집중할 수 있도록 학습자가 설계한 알고리즘의 결과를 확인 하는 것에 초점이 맞춰 교육할 필요가 있다[11].

초중등 학습자가 문제나 업무에 적합한 알고리즘을 설계하고 프로그래밍 언어의 문법과 사용법을 배워 올바른 시뮬레이션 결과를 도출하는 전 과정을 학습하는 것은 쉽지 않은 과정이다[10]. 프로그래밍 과정에서 학습자의 사용법에 대한 인지적인 부담감을 감소시키고, 학습의 동기를 유발하기 위한 목적으로 Scratch, Etoys, 엔트리와 같은 교육용 프로그래밍 도구들이 개발되었고, 프로그래밍 교육에서 다양하게 활용되고 교육적인 효과가 있는 것으로 나타났다[12][13]. 하지만 이러한 교육용 프로그래밍 도구들은 절차적 추상화에서 다루는 개념과 방법을 학습하기 위해 알고리즘을 설계하기 보다는 GUI기반으로 오브젝트나 스프라이트가 동작하는 것을 기반으로 학습자의 흥미와 관심을 유발하는 기능의 활용에 집중되고 있다. 이는 문제해결이라는 프로그래밍 활동의 본연의 목적에 부합하지 않을 뿐더러 도구의 사용법과 활용의 능숙도에 교육 효과가 집중될 수 있다. 따라서 사용법 학습에 대한 부담을 줄이고 절차적 추상화를 통해 알고리즘 설계에 집중할 수 있는 프로그래밍 도구에 대한 연구가 필요하다.

텐지블 프로그래밍 도구는 손을 사용하여 직관적으로 프로그래밍 할 수 있어 조작은 단순하지만, 학습내용과 콘텐츠는 추상적인 내용에서부터 단순한 내용까지 포괄적으로 활용할 수 있다[14]. 절차적 추상화에서 다루는 개념을 위계화하여 콘텐츠를 구성하고, 도구적인 사용이 단순한 텐지블 프로그래밍 도구를 활용한다면 보다 알고리즘 설계에 집중하여 프로그래밍 학습을 할 수 있는 장점이 있다[15]. 텐지블 프로그래밍 환경은 텐지블 인터페이스를 기반의 도구를 조작하여 프로그래밍 활동을 수행할 수 있는 환경을 의미한다. 신체를 사용하여 프로그래밍 활동을 할 수 있는 환경은 키보드 마우스를 사용하는 일반적인 컴퓨터 환경과 비교하여 직관적인 조작과 물리적인 환경에서 결과를 확인 할 수 있다는 장점 때문에 높은 교육적 효과를 보여 주고 있다[16]. 특히 텐지블 프로그래밍 환경은 도구 사용에 있어서 인지

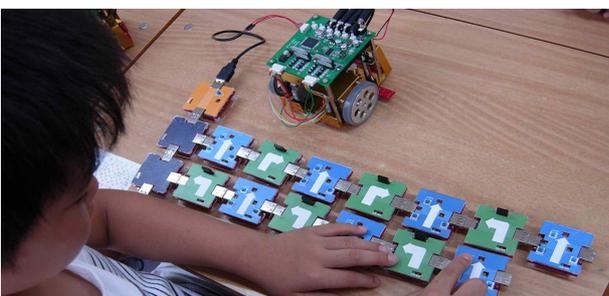
적 부담을 많이 느끼는 초보 학습자에게 큰 효과가 있는 것으로 나타났다[17]. 또한 교육적인 효과를 증대시키기 위해 학습자의 흥미와 관심을 고려하여 로봇과 함께 연계되어 제공되고 있다. 교육용 로봇의 활용은 학습자가 프로그래밍한 결과를 실세계에서 로봇의 움직임을 통해 확인할 수 있고, 학습자의 창의적인 아이디어나 설계를 현실에서 구현할 수 있다. 이처럼 텐지블 프로그래밍 환경은 프로그래밍 학습의 동기를 유발과 더불어 학습자에게 유의미한 경험을 제공할 수 있어 초중등 학습자를 대상으로 하는 프로그래밍 교육에서 널리 사용되고 있다[18].

본 연구의 목적은 텐지블 프로그래밍 도구를 활용한 알고리즘 학습에서 학습자의 성별과 학교급에 따라 성취도에 차이가 있는지를 분석하는 것이다. 이를 위해 절차적 추상화의 개념들을 초중등 학습자에게 가르친 이후 성별과 학교급에 따라 절차적 추상화 능력에 차이가 나타나는지 분석하였다. 실험도구는 기존에 개발된 LED 기반의 텐지블 프로그래밍 도구를 사용하였고 절차적 추상화 개념을 학습할 수 있는 콘텐츠를 개발하여 적용하였다.

2. 선행연구

2.1 Algorithmic Bricks (A-Bricks)

Algorithmic Bricks는 초등학생을 대상으로 로봇을 제어할 수 있는 텐지블 프로그래밍 도구이다[15]. 블록은 어린 학생이 한 손으로 잡을 수 있는 크기로 블록들을 좌우로 연결하거나 쌓는 조작을 통해 프로그래밍 할 수 있고, 순차, 반복, 조건, 매개변수의 프로그래밍 개념을 구체적인 형태로 조작할 수 있다.

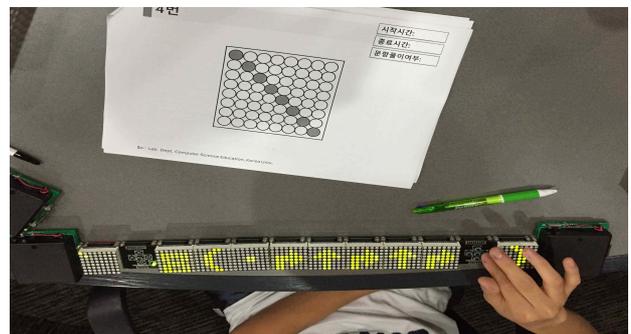


<그림 1> A-Bricks를 이용한 프로그래밍 모습

초등학교 저학년을 대상으로 동일한 문제를 Scratch 와 A-Bricks로 풀이한 결과, 사용성에서는 차이가 없었지만 논리적 오류와 조작 오류가 A-Bricks를 사용하는 집단이 더 낮게 발생하는 것으로 분석되었다[15]. 또한 A-Bricks를 활용하여 프로그래밍 할 수 있는 논리적 수준에 따른 로봇 과제를 개발하여 초등학교 1학년에게 적용한 결과, 프로그래밍 후에 논리적 사고력의 수준이 향상된 것으로 나타났고, 사용성 중 즐거움에 대한 항목이 4.5점 이상으로 분석되었다[19]. 즉, 초등학생을 대상으로 텐지블 프로그래밍 도구인 A-Bricks를 사용하여 프로그래밍을 가르치는 활동이 충분히 가능하고 효과적임을 시사한다.

2.2 B-Bricks

B-Bricks는 8*8 dot-matrix LED에서 개별적으로 LED의 점등을 제어할 수 있는 텐지블 프로그래밍 도구이다[20]. 학생에게 출력할 이미지를 문제로 제공하고 프로그래밍한 결과와 비교하여 정답의 유무를 즉시적으로 확인할 수 있다. 또한 텐지블 유저 인터페이스에 기반 하여 도구의 사용에 대한 학생의 인지적인 부담을 감소시킬 수 있고 전진, 우회전, 좌회전 등과 같은 간단한 이미지로 명령어의 기능을 전달 할 수 있어 주어진 문제를 자동적으로 처리하기 위한 절차적인 추상화에 보다 집중할 수 있는 장점이 있다.



<그림 2> B-Bricks를 이용한 프로그래밍 모습

초등학생을 대상으로 B-Bricks는 순차, 반복, 이중반복으로 문제의 난이도를 구분하여 적용한 결과 충분히 프로그래밍 활동이 가능하고, 사용성에 대한 설문에서 5점 만점 중 모두 4점 이상으로 응답한 것으로 분석되었다[20].

3. 연구방법

3.1 연구대상

연구대상은 서울 소재 초등학교와 중학교를 대상으로 학생을 모집하여 32명의 학생을 선정하였다.

<표 1> 연구대상

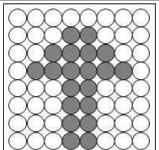
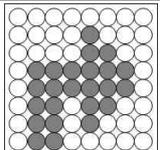
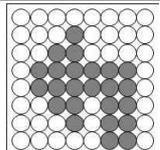
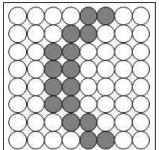
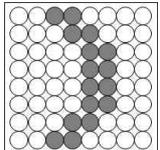
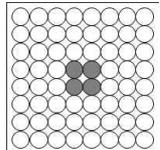
구분	학년	남	여	합계
초등학생	4	•	3	3
	5	10	5	15
	6	4	2	6
중학생	1	7	1	8
합계		21	11	32

3.2 연구도구

3.2.1 B-Bricks

학습자에게 출력해야 하는 이미지를 문제로 제공하고, B-Bricks의 명령어들을 조합하여 문제와 동일한 이미지를 출력하여 정답의 유무를 확인할 수 있다.

<표 2> B-Bricks 명령어

		
1칸 전진	우회전	좌회전
		
반복 시작	반복 종료	LED 점등

<그림 3>의 모습은 왼쪽 위의 모서리를 시작점으로 하여 “우회전, 7칸 전진, LED 점등”을 4회 반복하는 프로그래밍을 하는 모습이다.

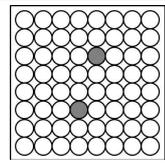
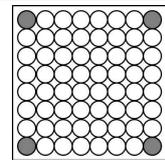
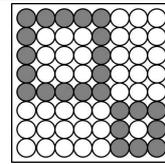


<그림 3> B-Bricks 사용예제

3.2.2 프로그래밍 문제

B-Bricks를 사용하여 풀이하는 프로그래밍 문제는 순차적 처리, 반복적 처리의 관점에서 난이도를 구분하였다. 난이도 하는 순차적으로 명령어를 나열하여 해결하는 수준이고, 난이도 중은 횟수에 의한 반복적인 처리를 통해 해결하는 수준이다. 마지막으로 난이도 상은 횟수에 의한 반복을 중첩하여 사용하여 해결하는 수준이다.

<표 3> 프로그래밍 문제의 난이도 예시

		
난이도 하	난이도 중	난이도 상

3.2.3 사용성 평가도구

사용성 평가도구는 ISO 9241-11:1998를 참고하여 개발한 도구를 사용하였다[20]. 실험에 참가한 학습자를 대상으로 Likert 5점 척도로 설문하였고, 검사도구의 신뢰도(Cronbach α)는 .87로 신뢰로운 것으로 분석되었다.

<표 4> 사용성 평가 문항

번호	문항
1	블록의 사용방법이 편리하다.
2	블록은 내가 문제를 푸는데 사용하기 쉽다.
3	블록을 사용하는 방법이 쉽다.
4	블록의 사용방법을 다른 친구에게 설명할 수 있다.
5	블록의 사용방법을 기억하기 쉽다.
6	블록을 사용하는 것은 재미있다.
7	블록은 흥미롭고, 계속 사용해 보았으면 좋겠다.

3.3 실험절차

실험은 B-Bricks 사용법과 예시 문제 풀이를 1시간 동안 진행한 이후, 2시간 동안 6문제를 학습자가 개별적으로 풀이하였다. 문제 풀이는 문제에서 제공하는 이미지와 출력한 결과가 일치할 경우에만 풀이시간을 기록하였고, 난이도(하->중->상) 순으로 풀이하였다.

<표 5> 실험절차

수업순서	실험내용
1교시	B-Bricks사용법 및 예시 문제 풀이
2~3교시	난이도 하->중->상 순으로 각 2문제씩 총 6문제 풀이
종료 후	사용성 평가

4. 연구결과

학습자의 특성인 성별(남학생: 21명, 여학생 11명)과 학교급(초등학교4,5학년: 18명, 초등학교6,중학교1학년: 14명)에 따라 절차적 추상화 능력에 대한 차이를 분석한 결과는 다음과 같다.

4.1 성별에 따른 차이분석

4.1.1 프로그래밍 성취도

성별에 따른 정답률의 차이를 분석한 결과 순차, 반복, 중첩반복 문제에서 차이가 나타나지 않았다. 성별 내에서 프로그래밍 문제의 난이도간 정답률의 차이를 분석한 결과, 남학생과 여학생 모두 순차와 반복 간에는 차이가 없었지만 순차와 중첩반복, 반복과 중첩반복 간에는 차이가 있는 것으로 분석되었다.

<표 6> 성별에 따른 프로그래밍 정답률(%) 차이분석

구분	남학생 (%)	여학생(%)	t	p
순차	73.8	77.3	.361	.721
반복	88.1	86.4	.147	.884
중첩반복	40.5	45.5	.513	.611
F	16.458***	6.204**		
p	.000	.006		
post-hoc	(순,중), (반,중)	(순,중), (반,중)		

p < .01, *p < .001

문제 풀이의 결과가 정답일 경우에 한하여 성별에 따른 문제 풀이 시간의 차이를 분석한 결과 성별에 따른 문제 해결 시간에는 차이가 없는 것으로 분석되었다. 성별 내에서 프로그래밍 문제의 난이도 간 문제 해결 시간의 차이를 분석한 결과 순차와 반복 간에는 차이가 없었지만 순차와 중첩반복, 반복과 중첩반복 간에는 차이가 있는 것

으로 분석되어 성별에 따른 정답률과 유사한 경향을 나타내었다.

즉, 문제를 해결하기 위한 단계를 학습자가 설계하는 과정에서 순차적인 처리라 하더라도 단순한 반복처리 보다 코드(블록)가 길어지거나 많아지는 경우에는 난이도가 오히려 상승하는 것으로 해석할 수 있다.

<표 7> 성별에 따른 프로그래밍 시간(초) 차이분석

구분	남학생(초)	여학생(초)	t	p
순차	380.9	391.7	.078	.938
반복	251.1	386.5	1.907	.067
중첩반복	728.6	940.9	.922	.366
F	7.725**	5.788**		
p	.001	.007		
post-hoc	(순,중), (반,중)	(순,중), (반,중)		

**p < .01

4.1.2 사용성

성별에 따라 프로그래밍에 사용한 B-Bricks에 대한 사용성 평가결과 남학생은 4.39점, 여학생은 4.85점으로 모두 4점 이상으로 높은 응답을 나타내었다.

성별에 따른 텐지블 프로그래밍 도구의 사용성의 차이분석 결과, 사용성에 대한 인식에서 차이가 있는 것으로 분석되었다. 즉, 텐지블 프로그래밍 도구의 활용은 남학생 보다는 여학생에게서 도구의 효과가 더 크게 나타난다고 할 수 있다.

<표 8> 성별에 따른 텐지블 프로그래밍 도구의 사용성 차이분석

구분	평균(표준편차)	t	p
남학생	4.39(0.63)	2.921**	.007
여학생	4.86(0.25)		

** p < .01

4.2 학교급에 따른 차이분석

4.2.1 프로그래밍 성취도

학년에 따른 정답률의 차이를 분석한 결과, 중첩반복에 대한 개념의 학습에서 초등학교 4,5학년 집단과 초등학교 6학년, 중학교 1학년 집단에 차이가 있는 것으로 나타났다.

초등학교 4,5학년 집단의 프로그래밍 문제 난이도간에 정답률의 차이를 분석한 결과, 순차와 반복, 반복과 중첩반복 간에 차이가 있는 것으로 나타났다. 초등학교 6학년, 중학교 1학년 집단의 프로그래밍 문제 난이도간에 정답률의 차이를 분석한 결과, 순차와 중첩반복, 반복과 중첩반복 간에 차이가 있는 것으로 나타났다.

<표 9> 학교급에 따른 프로그래밍 정답률(%) 차이분석

구분	초4,5학년 (%)	초6,중1학년 (%)	t	p
순차	69.4	82.1	1.426	.164
반복	91.7	82.1	.855	.399
중첩반복	50.0	32.1	2.044*	.050
F	12.491***	12.637***		
p	.000	.000		
post-hoc	(순,반), (반,중)	(순,중), (반,중)		

*p < .05, ***p < .001

문제 풀이에 걸리는 시간을 분석한 결과 두 집단의 차이는 나타나지 않았다. 학교급에 따른 프로그래밍 시간을 각 집단 내에서 프로그래밍 문제의 난이도간에 차이를 분석한 결과, 초등학교 4,5학년 집단에서는 순차와 중첩반복, 반복과 중첩반복 간에 차이가 있었고, 초등학교 6학년, 중학교 1학년 집단에서는 반복과 중첩반복 간에 차이가 있는 것으로 분석되었다.

<표 10> 학교급에 따른 프로그래밍 시간(초) 차이분석

구분	초4,5학년 (초)	초6,중1학년 (초)	t	p
순차	298.3	495.6	1.411	.177
반복	342.7	234.1	1.553	.132
중첩반복	835.4	751.0	.361	.721
F	10.843***	4.026*		
p	.000	.028		
post-hoc	(순,중), (반,중)	(반,중)		

*p < .05, ***p < .001

4.2.2 사용성

학교급에 따른 텐지블 프로그래밍 도구에 대한 사용성을 분석한 결과, 두 집단에서 차이가 있는 것으로 나타나지 않았다.

<표 11> 학교급에 따른 텐지블 프로그래밍 도구의 사용성 차이분석

구분	평균(표준편차)	t	p
초4-5학년	4.56(0.61)	.022	.983
초6/중1학년	4.55(0.55)		

5. 결론 및 논의

본 연구는 텐지블 프로그래밍 도구를 사용하여 알고리즘 학습한 이후, 학습자의 특성에 따라 성취도와 사용성에 차이가 있는지를 분석하고자 하였다.

성취도에서 성별에 따른 차이는 나타나지 않았지만 학교급에서는 오히려 학년이 낮은 학습자가 높은 학습자에 비해 정답률이 더 높은 것으로 분석되었다. 2009년 개정 교육과정에서는 중첩 반복을 포함한 알고리즘 설계 개념 학습을 중학교 1학년에 수행 하도록 제시하고 있다[21]. 본 연구의 결과를 기반으로 중첩반복과 같은 수준의 알고리즘 학습도 환경적이 부분에서 학습자에게 인지적 부담을 주지 않는 도구를 사용한다면 초등 4학년에서도 충분히 가능함을 보여주었다.

텐지블 프로그래밍 도구의 사용성에서는 여학생이 남학생에 비해 높게 인식하는 것으로 나타났다. 선행연구들에서는 프로그래밍 활동에서 여학생이 남학생보다 학습에 어려움을 겪는 것으로 보고하고 있다. 하지만 본 연구에서는 정답률과 문제 풀이 시간에서 남학생과 여학생 간에 차이가 없는 것으로 나타났고, 도구의 사용성에서 남학생보다는 여학생이 더 높은 사용성을 나타내어 텐지블 프로그래밍 도구가 가져다준 효과라고 해석할 수 있다.

추가적으로 연구에 사용하였던 텐지블 프로그래밍 도구는 다음과 같은 사항을 고려할 필요가 있는 것으로 보인다. 첫째, 프로그래밍 도구의 사용방법은 직관적으로 조작할 수 있는 수준으로 제공되어 학습자가 추상화한 내용을 도구사용에 어려움 없이 표현할 수 있어야 한다. 둘째, 프로그래밍 도구에서 다루는 알고리즘은 쉬운 수준에서 어려운 수준까지 폭넓게 다룰 수 있어야 한다.

향후 연구로는 텐지블 프로그래밍 도구를 사용하여 알고리즘 학습을 위한 구체적인 평가기준의

개발과 텐지블 프로그래밍 문제의 난이도를 구분할 수 있는 세분화된 요소의 연구가 필요하다.

참 고 문 헌

- [1] Wing, J. M. (2006). Computational Thinking, *Communications of the ACM*, 49(3), 33-35.
- [2] 김자미, 이원규 (2014). 영국의 교육과정 개정으로 본 정보교과의 지식과 문제해결력에 대한 쟁점, *컴퓨터교육학회논문지*, 17(3), 54-64.
- [3] 김자미, 이원규 (2014). 통합에서 독립으로, 이스라엘 컴퓨터과학 교과과정의 진화, *한국컴퓨터교육학회*, 17(4), 33-44.
- [4] Wing, J. M.(2008). Computational thinking and thinking about computing, *Philosophical transactions of the royal society A*, 366, 3717-3725.
- [5] Robert C. G., Veda C. S.(1999), Data abstractions: Why and how?, *Data & Knowledge Engineering*, 29, 293-311.
- [6] Dinesh B., Davis J. G.(1992), Conceptual data modelling in database design: similarities and differences between expert and novice designers, *Man & Machine Studies*, 37, 83-101.
- [7] Marvin L. Minsky(1967). *Computation: finite and infinite machines*, Prentice Hall, Upper Saddle River, NJ, USA.
- [8] Futschek, G. (2006). Algorithmic thinking: the key for understanding computer science. Informatics Education? *The Bridge between Using and Understanding Computers*, 159-168.
- [9] Yuri Gurevich(2000). Sequential abstract state machines capture sequential algorithms. *ACM Transactions on Computational logic*, 1(1), pp. 77-111.
- [10] Liu, C., Cheng, Y., & Huang, C. (2011). The effect of simulation games on the learning of computational problem solving. *Computers & Education*, 57(3), 1907-1918.
- [11] Rogozhkina, I., & Kushnirenko, A. (2011). PiktoMir: teaching programming concepts to preschoolers with a new tutorial environment. *Procedia Social and Behavioral Sciences*, 28, 601-605.
- [12] Maria K.(2012). Diverse categories of programming learning activities could be performed within Scratch, *Procedia Social and Behavioral Sciences*, 46, 1162-1166.
- [13] 정미연, 이은경, 이영준 (2008). Squeak Etoys 활용 알고리즘 학습이 중학생의 문제해결력에 미치는 영향, *대한공업교육학회*, 33(2), 170-191.
- [14] Hiroshi I., Brygg U. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *Proceedings of Conference on Human Factors in Computing Systems(CHI '97)*, 1-8.
- [15] Kwon D. Y., Kim H. S., Shim J. K., & Lee W. G. (2012). Algorithmic Bricks: A Tangible Robot Programming Tool for Elementary School Students. *IEEE transactions on Educations*, 55(4), 474-479.
- [16] Sapounidis T., Demetriadis S. (2013). Tangible versus graphical user interfaces for robot programming: exploring cross-age children's preferences, *Personal and Ubiquitous Computing*, 17(8), 1775-1786.
- [17] Bers, M. & Horn, M. (2010). *Tangible programming in early childhood: Revisiting developmental assumptions through new technologies*. Greenwich, CT: Information Age Publishing.
- [18] M. U. Bers, L. Flannery, E. R. Kazakoff, A. Sullivan (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum, *Computer & Education* 72, 145-157.
- [19] 권대용 (2013). 텐지블 프로그래밍 도구를 활용한 논리적 사고력기반의 초등 로봇 과제 개발 및 적용, *컴퓨터교육학회논문지*, 16(4), 13-21.
- [20] 심재권, 이원규, 권대용 (2015). LED 기반 텐지블 프로그래밍 도구 개발 및 적용, *컴퓨터*

교육학회논문지, 18(1), 35-43.

[21] 교육과학기술부(2011). 제 2011-361호[별책 18] 중학교 선택 교과 교육과정. 정보.



심재권

2007 경인교육대학교
컴퓨터교육과(교육학학사)
2012 고려대학교
컴퓨터교육학과(이학석사)

2012 ~ 현재 고려대학교 컴퓨터교육학과 박사수료
관심분야: 컴퓨터교육, 프로그래밍 교육
E-Mail: jaekwoun.shim@inc.korea.ac.kr



이원규

1985 고려대학교 문과대학
영어영문학과(문학사)
1989 筑波大学 大学院
理工学研究科(공학석사)

1996 ~ 현재 고려대학교 컴퓨터학과 교수
관심분야: 컴퓨터교육, 정보검색, 데이터베이스
E-Mail: lee@inc.korea.ac.kr



권대용

2003 고려대학교 사범대학
컴퓨터교육과(이학사)
2006 고려대학교 대학원
컴퓨터교육학과(이학석사)

2011 고려대학교 대학원 컴퓨터교육학과(이학박사)
2011 ~ 현재 고려대학교 컴퓨터교육과 연구교수
관심분야: 컴퓨터교육, EPL, 교육용 로봇, 학습과학,
영재교육
E-Mail: daiyoung.kwon@inc.korea.ac.kr