

피지컬 컴퓨팅 수업에서 나타난 초등학생의 컴퓨팅 실천 (Computational Practices)에 관한 연구

민선희[†] · 김민경^{††}

요 약

본 연구는 교육부(2015)의 2015 개정교육과정에서 다루고 있는 소프트웨어 교육지침을 토대로 초등학교 6학년을 대상으로 구체적인 매체를 활용하여 물리적 모델을 구성하고 이를 프로그래밍으로 제어함으로써 디지털 세계와 물리적 세계의 능동적 상호작용을 지원하는 피지컬 컴퓨팅 수업을 설계하여 적용하였다. 이를 통해 피지컬 컴퓨팅 수업의 과정 및 결과에서 컴퓨팅 실천을 중심으로 컴퓨팅 사고의 변화를 분석하였다. 연구결과 피지컬 컴퓨팅 수업은 학생들로 하여금 컴퓨팅 실천(computing practices)을 통한 학습을 지원하여 컴퓨팅 사고의 형성을 지원하였다. 또한 학생들의 아이디어를 실제화하도록 피지컬 컴퓨팅 도구를 활용해서 컴퓨팅 기술로 구현하도록 하였고, 실제적 맥락의 문제상황은 학생들의 컴퓨팅 사고가 발현될 기회를 마련하였다. 본 연구를 통해 개발된 피지컬 컴퓨팅 수업과 분석 도구들이 앞으로 교육현장에서 실행될 소프트웨어 수업에 관한 사례연구로서 교육적 의미와 시사점을 제시할 것이라 기대한다.

주제어 : 피지컬 컴퓨팅, 컴퓨팅 사고, 컴퓨팅 개념, 컴퓨팅 실천, 컴퓨팅 관점

A Study on Computational Practices of Elementary School Students in Physical Computing Lessons

Sun Hee Min[†] · Min Kyeong Kim^{††}

ABSTRACT

This purpose of this study is to develop and apply the physical computing lessons based on the software guidelines from the Ministry of Education (2015). In this study, I research how computational thinking occurs in class by applying the physical computing lessons to elementary students based on computational practices. The physical computing lessons and analytic methods for computational thinking in this study can be used as a sample and case-study to develop the lessons in the educational field.

Keywords : Physical Computing, Computational Thinking, Computational Concepts, Computational Practices, Computational Perceptions

[†]정 회 원: 이화여자대학교 초등교육학과 대학원생
^{††}정 회 원: 이화여자대학교 초등교육과 교수(교신저자)
논문접수: 2019년 3월 8일, 심사완료: 2019년 4월 22일, 게재확정: 2019년 5월 2일
* 본 논문은 민선희의 박사학위 논문(2018)에서 일부를 요약·재구성한 것임.

1. 서론

4차 산업혁명이 가져온 변화로 인해 미래 직업과 삶의 변화와 함께 첨단과학기술 및 소프트웨어 교육에 관한 관심이 높아졌다. 이런 변화에 대비하기 위해 미국, 영국, 호주, 인도, 중국 등의 나라에서는 컴퓨팅 사고(Computational Thinking)를 미래핵심역량으로 제시하면서 정규교육과정에 반영하고 있다[1]. 한국도 컴퓨팅 사고를 핵심역량으로 강조하면서, 정보윤리 및 태도를 포함한 역량 강화를 목표로 2018년도부터 점진적으로 정규교과에서의 적용을 제시하였다[2].

특히 초등학교의 소프트웨어교육은 2015 개정교육과정[2]에 근거하여 초등 5~6학년 실과교과에 소프트웨어교육이 도입되면서 현재 제시된 17시간이라는 시간적 제약[1][3], 수업에 적용하기 위한 콘텐츠, 평가도구 등과 같은 실제적인 어려움도 함께 보고되는 상황이다[4][5]. 이러한 제약들을 해결하기 위해 수학 및 사회 등과 같은 다양한 교과 연계를 하거나[6], 창의적 체험 및 동아리 활동에 적용하는 방안들이 연구되고 있다[7][8][9].

이와 관련하여 변화하는 시대상을 고려하여 학습자의 능동적 지식구성을 지원할 실제적 체험(Hands-on Learning/Learning by Making)을 지원하도록 학습자의 컴퓨팅 실천이 가능한 미래형 교실 환경의 조성이 필요하다는 것을 제시하고 있다[10]. 또한 초등학생의 발달단계 및 국가별 교육과정에서 요구하는 학습 목표를 고려하여 교육용 프로그래밍 언어의 적용을 확산시키고 있고[11][12], 특히 한국의 경우 스크래치와 엔트리학교현장에서 널리 활용되고 있다. 그리고 로봇을 포함하여 피지컬 컴퓨팅(Physical Computing) 도구와 같은 교육용 매체의 활용을 강조하고 있다[13][14][15].

특히 피지컬 컴퓨팅(Physical Computing) 도구는 컴퓨팅 환경과 물리적 환경을 연결할 수 있어 학생들의 실천과 이해를 돕기 위한 유의미한 학습의 도구로 활용할 수 있고[16][17], 문제상황을 파악하기 위한 핵심단서를 시각화 및 청각화시킬 수 있다는 점[18]에서 향후 교육적 활용도가 높아질 것으로 기대된다.

피지컬 컴퓨팅 도구에 포함된 센서, 모터를 포함

한 출력 장치들은 학생들의 흥미를 유발하고 능동적 체험을 지원함으로써 실생활의 복잡한 문제를 해결하도록 지원하며[19], 능동적 지식구성을 지원하는 도구(mental model)로서 의미가 있다[20]는 점에서 미래형 교실 환경에 필요한 도구로 활용될 수 있다. 또한 피지컬 컴퓨팅에서 지원하는 프로그래밍은 수학 및 과학의 지식과 개념을 컴퓨팅 기술로 구현하고 문제해결을 위한 개념 획득이 가능하다는 점에서 교육적 활용도가 높다[21][22][23].

본 연구는 초등학생을 대상으로 피지컬 컴퓨팅 수업을 설계 및 적용하여 수업에서 나타난 컴퓨팅 사고의 특징과 수업의 양상을 컴퓨팅 실천 관점에서 분석하였다. 이를 통해 피지컬 컴퓨팅 수업을 위한 수업 설계 및 구체적인 적용예시를 제시하고 평가방법 및 후속 연구에 대한 아이디어를 제공하고자 한다.

2. 이론적 배경

2.1 소프트웨어교육과 피지컬 컴퓨팅

2.1.1 국내외 소프트웨어교육

소프트웨어교육에 대한 정의를 살펴보면, 소프트웨어에 대한 지식, 프로그래밍, 코딩, 컴퓨팅 관련 기초 등을 배우는 것을 의미한다[24]. 이와 관련하여 2015 개정교육과정에서는 정규교육과정에서 소프트웨어교육을 위한 핵심역량으로 컴퓨팅 사고력(Computational Thinking)을 강조하고 있다[25].

특히 바른 인성의 함양과 미래사회가 요구하는 역량 개발을 위해 인문·사회·과학기술의 균형적 발달을 제시하고 있다[2][26]. 이를 위해 미래 핵심역량으로 창의융합사고역량, 의사소통역량, 자기관리역량, 지식정보처리역량, 심미적 감성역량을 강조하면서 이런 핵심역량을 기르는 방안으로 소프트웨어교육을 정규교과에 도입하고 있다[2].

현재 초등학교는 17시간 이상, 중·고등학교의 경우 34시간 이상을 실시하도록 하고 있으며 컴퓨팅 사고력을 가진 창의적 융합인재 양성을 강조하고 있다. 소프트웨어를 위한 학교급별 교육목표를 살펴보면 초등학교는 생활과 소프트웨어, 알고리즘과 프로그래밍을 다루도록 하고 있으며, 중·고등학교에서는 컴퓨팅과 문제해결 영역을 추가할 것을 제

시하고 있다[2][26].

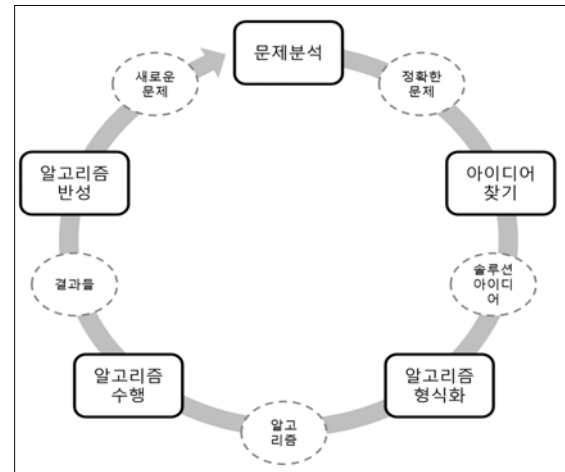
초등학교에서는 실과과목을 바탕으로 교육과정을 재구성하도록 다음과 같이 제시하고 있다[26]. 첫째, 5학년에서 다루고 있는 [생활과 정보] 단원의 경우는 문제해결과정의 체험, 알고리즘의 체험, 생활과 소프트웨어를 실시하도록 한다. 다만 창의적 체험활동의 ‘자율영역’ 시간을 활용해서 프로그래밍 체험을 진행한다. 둘째, 5학년의 [생활과 정보] 및 6학년의 [생활과 전기전자] 단원을 재구성하여 진행한다. 이때 프로그래밍의 체험은 ‘로봇의 이해’에서 실습을 통해 다루도록 한다. 공통으로 2가지 방안 모두 체험시간의 확보를 지적하고 있다.

이와 비교하여 미국은 컴퓨터과학의 중요성이 대두되면서 STEAM(Science, Technology, Engineering, Mathematics 이하 STEM)교육에서 컴퓨팅 사고가 핵심이 되고 있다[27]. 또한 학교현장에서 수학, 공학 등과 같은 다양한 교과를 융합적으로 접근하여 혁신적으로 수업을 개선하려는 시도를 지속하고 있다[28].

영국은 2014년부터 소프트웨어교육을 실시하면서 초등수준의 학생들을 위한 피지컬 컴퓨팅의 필요성을 강조하고 있다[11]. 한편 중국은 초등 3학년년부터 고등학교까지 68~140시간을 가르치도록 하고 있으며, 일본은 2009년부터 유치원부터 소프트웨어교육을 필수적으로 가르치도록 교육을 확대하고 있다[24].

이러한 소프트웨어교육의 도입 및 확대와 관련하여 현장 적용을 위해 교수학습모형에 관한 연구들이 진행되고 있다[29][30][31][32][33]. 컴퓨팅 사고 기반 소프트웨어 교수학습모형들은 공통적으로 제작, 개발 구현 등과 같은 실제적인 체험을 강조하고 있는데[27], 특히 알고리즘 형성을 기반으로 한 프로그래밍을 강조하고 있다.

알고리즘 발명 수업모형은 알고리즘의 형식화를 중심으로 구성되었는데, 다음 [그림 1]과 같다[34]. 알고리즘 발명 수업모형은 문제해결 절차를 기반으로 진행되어 학습자들이 로봇제작보다는 로봇을 제어하는 알고리즘과 프로그래밍에 초점을 두도록 구성되어 있다.



[그림 1] 알고리즘 발명 수업모형 (Futschek & Moschitz, 2010, p.4)

2.1.2 피지컬 컴퓨팅(Physical Computing)

소프트웨어교육에서 핵심역량으로 강조하고 있는 컴퓨팅 사고를 기르기 위해서는 프로그래밍 언어의 단순구현이 아닌 실제 생활에서 발견되는 문제를 해결하기 위한 실제적인 솔루션을 구축하도록 알고리즘을 고안하고 프로그래밍을 통해 구체화하는 과정이 필요하다[12]. 피지컬 컴퓨팅은 컴퓨터와 인간의 상호작용을 지원하는 물리적인 시스템을 구축하도록 소프트웨어와 하드웨어를 사용하는 측면 [35]에서 물리적 세계와 컴퓨팅 세계를 연결하고 자신의 아이디어를 객관적으로 시각화시키고 모델링할 수 있다는 측면에서 학습자의 이해와 실천을 촉진할 수 있다.

피지컬 컴퓨팅 도구는 형태에 따라서 세가지 형태로 구분되는데, 일반적으로 알려진 로봇형을 포함하고 있다. 이외 보드형, 모듈형으로 구분될 수 있다[12]. 이중 모듈형은 학습자 스스로 자신만의 로봇을 디자인하고 설계 및 프로그래밍을 경험할 수 있어, 소프트웨어교육에서 강조하는 체험, 개발, 알고리즘과 프로그래밍을 시행할 수 있다는 측면에서 학습자의 흥미 유발과 능동적인 참여를 유도할 수 있다. 특히 일상환경에서 사용되는 익숙한 재료의 활용 및 실제적인 해결방법을 경험하는 시간은 각자의 생각을 수정 및 보완하는 시간(Tinkering time)이 되어 학습자의 깊은 이해를 지원하는 도구로 활용될 수 있다[36].

2.2 컴퓨팅 사고(Computational Thinking)

2.2.1. 컴퓨팅 사고의 정의와 구성요소

Wing(2006)은 컴퓨팅 사고에 대해서 21세기를 살아가는 데 필요한 기술로서 읽기, 쓰기, 셈하기와 같이 기본적이고 필수적인 기술이라고 강조하였다. 또한 컴퓨팅 사고란 문제해결에 컴퓨팅 기술을 활용하는 것이라고 소개하면서 핵심요소로 추상화, 자동화를 제시하였다[37].

한편 국내의 경우, 컴퓨팅 시스템을 활용해 문제를 해결하는 것을 포함하여 문제를 발견하고 이해하면서 해결방법을 찾아서 시행하는 것이라고 컴퓨팅 사고를 정의하였다. 특히 해결된 문제의 평가를 통해 문제해결의 방법을 점검하고 오류를 수정할 수 있다고 지적하였다[38]. 그외 컴퓨팅 사고에 대해서 컴퓨팅 시스템의 역량을 활용해서 문제를 효과적, 효율적으로 해결할 수 있는 절차적 사고능력으로 정의하기도 하였다[39].

반면 컴퓨팅 사고가 1950~60년대에 알고리즘적 사고(Algorithmic thinking)으로 사용되던 개념으로 오늘날에는 다양한 유형의 문제를 효율적으로 해결하는 방법, 알고리즘의 개발을 위한 수학의 사용, 다양한 단계의 추상화를 포함하고 있다고 정의하기도 하였다[40]. 이와 관련하여 컴퓨팅 사고의 컴퓨팅 사고의 핵심요소를 알고리즘, 추상화, 자동화로 제시하면서 프로그래밍 교육만으로 컴퓨팅 사고와 관련된 아이디어를 가르칠 수 없으므로 다양한 교과와 연계된 교육을 강조[41]하였다. 또한 컴퓨팅 사고와 관련된 과제를 해결하기 위해서는 프로그래밍이 중요한 도구로 사용된다고 지적하면서 패턴의 정의 및 추상화가 중심이 되는 점이 컴퓨팅 사고가 다른 사고와 구별되는 점이라고 제시[42]하였다.

컴퓨팅 사고의 구성요소로 공통적으로 추상화 및 자동화를 지적[26][29][37][40][41][42]하고 있으며, 그 외 알고리즘[26][29][40][41], 문제분해[26][29]를 포함시키기도 하였다. 이를 바탕으로 컴퓨팅 사고에 대한 학자들의 정의를 종합해보면, 단순히 컴퓨터를 활용하거나 프로그래밍과 같은 협의의 개념보다는 컴퓨팅 사고를 문제해결을 위한 생각 도구(mental tools)의 관점[43]에서 문제해결의 틀로

접근하고 있다[29][37]. 이런 측면에서 컴퓨팅 사고는 사고과정(mental process)과 유형의 산출물(tangible outcomes)을 포함하는 것으로 정리할 수 있다.

2.2.2. 컴퓨팅 사고의 평가

컴퓨팅 사고를 평가하기 위해서 우선 학습자들의 프로그래밍 능력, 사고력, 문제해결력을 평가하기 위한 선다형 문항들을 기반으로 한 평가도구들이 활용되기도 하였다[44][45][46]. 특히 비버챌린지 문항을 이용하여 컴퓨팅 사고의 개념적 이해를 평가하는 방법은 평가의 방법으로 유용하다는 측면에서 한국에서도 활용되고 있다[47]. 또한 선다형 평가도구를 보완하기 위해 루브릭을 활용한 연구들도 시도되었다[32][48].

한편 컴퓨팅 사고를 인지적 측면에서만 평가하는 것이 아니라 개인적 역량, 역량을 배양하는 행동, 행동이 실행되는 환경으로 구분해 접근하기도 하였다[43]. 그리고 컴퓨팅 사고를 평가하기 위해 컴퓨팅 사고를 실천(practices), 개념(concepts), 관점(perspectives)으로 나누어 평가할 것을 지적하는 연구들도 진행되었다[32][49]. 이와 관련해서 프로젝트의 분석, 산출물을 활용한 인터뷰, 시나리오 설계를 평가할 것을 제시하면서 각 방법의 강점과 한계점을 다음 <표 1>과 같이 지적하였다[49].

<표 1> 컴퓨팅 사고 평가방법의 강점과 한계점

평가 방법	컴퓨팅 개념	컴퓨팅 실천	컴퓨팅 관점
프로젝트 분석	개념적 지표로 프로그래밍 명령어(블록)을 활용할 수 있음	평가가 불가능	평가가 불가능
산출물에 기반한 인터뷰	개념적 이해와 차이가 있을 수 있으며, 프로젝트 결과물에 대한 한계를 가짐	가능하지만 자신이 실제로 설계한 경험에 기반해야 하며, 기억의 한계가 있음	가능할 수 있으나 직접적으로 묻기 어려움
시나리오 설계	개념적 이해의 차이와 범위를 가지며, 외부적으로 선택된 프로젝트에 한함	가능하지만 실시간으로 새로운 상황이어야 하며, 외부적으로 선택한 프로젝트에 한함	가능할 수 있으나 직접적으로 묻기 어려움

(Brennan & Resnick, 2012, p.22)

프로젝트 분석은 컴퓨팅 개념을 평가하는데 적절하며 컴퓨팅 실천과 관점의 평가는 불가능하다고 강조하였다. 컴퓨팅 실천을 분석하기 위해서는 산출물을 기반으로 한 인터뷰와 시나리오 설계를 활용할 것을 제안하였다.

컴퓨팅 실천을 분석하는 데 있어서 추상화와 모듈화, 실험과 반복, 테스트와 디버깅, 재사용과 리믹스로 구분하여 컴퓨팅 실천 루브릭[32]을 활용하여 분석하기도 하였다. 그 외 컴퓨팅 실천 차원에서 컴퓨팅 사고를 데이터 연습, 모델링과 시뮬레이션, 컴퓨팅 문제 연습, 시스템적 사고 연습으로 구분[50]하여 교과 연계 및 문제해결 측면에서 접근하기도 하였다.

학교현장에서 교사들이 겪는 평가의 어려움을 극복하기 위해[5] 학습자들의 컴퓨터 기반 역량을 평가하기 위한 연구들이 지속되고 있다[39][43]. 컴퓨팅 개념의 이해[12] 뿐만 아니라 수업 전반에서 나타나는 컴퓨팅 실천의 변화를 파악하기 위한 루브릭에 관한 연구들이 확산되고 있다[32][48]. 또한 의사소통, 동기유발 등과 같은 컴퓨팅 사고의 정성적 측면에 관한 연구들도 실행되고 있다[22][23][43].

3. 연구방법

3.1 연구대상

피지컬 컴퓨팅 수업에서 나타나는 개인의 컴퓨팅 실천 변화를 파악하기 위해 단일 사례를 선정하여 분석하였다. 사례의 선정은 피지컬 컴퓨팅 수업에 참여한 10명의 남학생 중에서 비버챌린지를 활용한 컴퓨팅 개념 검사에서 성취가 뛰어난 상위 4명을 선정하였다.

연구에 참여한 학생 모두 로봇 및 프로그래밍에 대한 경험이 있었으며, 아두이노, 스크래치, C언어 등을 활용하였다고 응답하였다. 본문에 표기된 학생 이름은 모두 가명으로 표기하였다.

3.2 피지컬 컴퓨팅 수업 설계

본 연구에서는 학생들의 실제적 이해를 파악하기 위해 수업 과정에서 나타난 컴퓨팅 실천을 중심으로 컴퓨팅 사고를 분석하고자 하였다. 이를 위해 물리적

환경과 컴퓨팅 환경의 적극적 상호작용을 지원하는 피지컬 컴퓨팅 도구를 활용함으로써 학습자들에게 자신의 아이디어를 테크놀로지와 소프트웨어를 활용하여 구체화시킬 수 있는 학습환경을 제공하였다.

문헌 연구를 통해 학생들에게 친숙한 부품들을 사용하고 로봇의 제작 및 창의적 변형이 가능한 모듈형 피지컬 컴퓨팅 도구를 활용하였다[11][14][23]. 그리고 컴퓨팅 사고의 향상을 위해 알고리즘과 프로그래밍의 체험을 강조하는 점[31][34]을 고려하여 다음에 제시된 <표 2>와 같이 피지컬 컴퓨팅 수업의 단계별 활동내용을 구성하였다.

<표 2> 피지컬 컴퓨팅 수업의 단계별 활동내용

구분	수업 단계	활동내용
1	문제분석(AP) Analyze Problem	- 문제상황을 읽고 주어진 상황과 조건을 파악하여 문제를 명료화한다
2	아이디어 찾기(FI) Finding Idea	- 아이디어를 생성하고 자료를 수집한다 - 최적의 아이디어를 생성한다 - 수업에 필요한 매체를 사용하여 문제 해결을 위한 아이디어를 수집한다
3	알고리즘 형식화(FA) Formulate Algorithm	- 실현방법과 과정을 설계한다 - 알고리즘을 형식화한 내용을 프로그램 형태로 변환한다
4	알고리즘 수행(PA) Play Algorithm	- 계획한 대로 실현한다 ⇒ 작성한 프로그램을 시행한다
5	알고리즘 반성(RA) Reflect Algorithm	- 문제해결과정 및 결과를 평가한다 ⇒ 프로그램의 실행 결과를 알고리즘과 비교한다. ⇒ 시행결과를 토대로 알고리즘을 수정 및 보완한다. - 상호평가 및 자기평가를 실시한다

피지컬 컴퓨팅 수업설계의 특징은 로봇의 사용, 응용, 창조의 과정을 통해 소프트웨어교육지침[2]에서 강조하는 알고리즘과 프로그래밍 영역을 경험하도록 수업의 주제, 내용, 평가방법을 설정한 것이다. 일상생활과 연계된 수업주제를 선정하여 학생들의 감성적 체험이 가능하도록 하였고 각자의 산출물을 각자의 의도에 맞춰서 구성할 수 있도록 피지컬 컴퓨팅 도구를 활용하여 탐구, 실험, 수정과 창작이 가능한 학습환경을 제공하였다.

본 연구에서는 초등학교 실과교과의 핵심개념, 교과내용, 성취기능을 탐구하여 수학 및 사회교과와 연계하여 ‘미로탈출, 스쿨버스’ 수업을 개발하고 적용하였다. 수업주제를 개발하기 위해 실제 생활, 탐구, 즐거움, 협력이 가능[51]하며, 지역과의 연계가 가능

<표 3> 컴퓨팅 실천 평가용 루브릭

컴퓨팅 실천		1수준(1점)	2수준(2점)	3수준(3점)
문제의 이해와 구조화	제시된 문제를 이해할 수 있다	문제에 설명하지 못하거나 제시된 문제를 단순반복한다	문제에서 요구하는 조건을 1가지 이상 찾아낼 수 있다	문제의 의도를 파악하여 필요한 조건들을 구체적으로 제시한다
	제시된 문제를 구체화시킬 수 있다	자신의 프로젝트에서 시도했던 구체적인 예시를 제공하지 못한다	자신의 프로젝트안에서 시도했던 단편적인 예시를 제공한다	자신의 프로젝트에서 시도했던 구체적인 예시를 제시한다
문제 해결 방법 탐색	문제를 해결하기 위한 방법을 순서에 따라 설명할 수 있다	기본적인 설명을 할 수 있지만 특정 프로젝트에 대한 구체적인 설명은 없다	어떤 순서에 따라 특정 프로젝트를 설계하는 단편적인 예시를 제시한다	특정 프로젝트의 세부사항을 제시하며, 어떤 순서에 따라 어떻게 개발했는지 설명한다
	제한한 문제해결 방법의 문제점과 개선방법에 대해 설명할 수 있다	어떤 수정도 하지 않았거나 수정했다는 것을 이야기했으나 근거를 제시하지 못한다	자신의 프로젝트에서 수정한 하나의 구체적인 사례를 묘사한다	자신의 프로젝트에서 수정한 것들을 한 개 이상 제시하고 그 이유를 적는다
알고리즘의 체험	순차, 선택, 반복 구조를 이용하여 문제해결 절차를 그림이나 기호를 이용하여 표현할 수 있다	문제해결 절차를 그림이나 기호를 이용해 표현하지 못한다	순차, 선택, 반복 구조 중 1개 이상을 이용하여 문제해결의 절차를 그림이나 기호를 이용하여 표현할 수 있다	순차, 선택, 반복 구조 중 2개 이상을 이용하여 문제해결의 절차를 그림이나 기호를 이용하여 표현할 수 있다
프로그래밍의 이해	프로그래밍 명령어의 기본 요소를 알 수 있다	명령어의 세부 기능을 모르고 프로그램을 임의로 구성한다. 로봇이 작동되지 않는다	프로그램에 사용된 명령어 기능을 일부 알고 사용할 수 있다. 로봇이 일부 기능을 수행한다	프로그램에 사용된 명령어 기능을 모두 알고 적절히 사용할 수 있다. 로봇이 전체 기능을 수행한다

한 주제[52]인지 고려하였다. 1개의 주제당 3회차 수업으로 진행되었고, 1회차당 80분이 소요되었다.

3.3 연구설계

본 연구에서는 피지컬 컴퓨팅 수업에서 나타나는 학생들의 컴퓨팅 실천을 분석하기 위해 양적 및 질적 자료를 수집하여 연구하였다. 피지컬 컴퓨팅 수업 단계를 바탕으로 개인별 컴퓨팅 실천이 어떻게 나타났는지 파악하였고, 활동지 및 인터뷰에서 취합한 자료를 활용하여 분석하였다.

이를 위해 컴퓨팅 실천 평가용 루브릭을 개발하였는데, 소프트웨어교육지침[2][27]을 바탕으로 선행연구들[32][48][53]을 참고하여 재구성하였다. 컴퓨팅 실천 평가용 루브릭을 개발하기 위해 초등학교 6학년 대상의 소그룹(4명) 연구와 초등5~6학년의 대집단(20명)의 예비연구를 통해 분석기준과 내용을 수정 및 보완하였다.

예비연구의 내용을 바탕으로 수정된 평가용 루브릭은 로봇교육 및 초등교육 전문가들, 초등교사들의 자문을 받아서 타당도를 검증받았다. 다음 <표 3>은 본 연구에서 개발하여 사용한 컴퓨팅 실천 평가용 루브릭이다.

학습자의 산출물만으로는 평가의 한계가 있다는 지적[32]에 따라 수업이 종료된 직후에는 학생들을 대상으로 산출물을 활용한 인터뷰를 시행하였다. 인터뷰 결과는 모두 동영상으로 녹화 후 전사하여 컴퓨터 실천 루브릭에 근거하여 3인의 전문가에 의해 채점되었다.

컴퓨팅 실천 검사결과에 대해서 채점자간 신뢰도를 파악하기 위해 Person 상관계수를 이용하여 분석하였다. 다음 <표 4>는 활동지와 인터뷰를 활용한 컴퓨팅 실천 분석결과에 대한 신뢰도이다.

<표 4> 컴퓨팅 실천에 대한 채점자간 신뢰도

평가자료 유형	활동주제	A-B	A-C	B-C
활동지	미로탈출	.994**	.797**	.833**
	스쿨버스	.748*	.795**	.808**
인터뷰	미로탈출 + 스쿨버스	.934**	.911**	.875**

4. 연구 결과

4.1 수업설계 및 적용

피지컬 컴퓨팅 수업은 ‘미로탈출’과 ‘스쿨버스’의 2가지 테마를 중심으로 각 3회차로 진행되었다. 다음 <표 5>는 본 연구에서 진행된 수업주제 및 활동내용이다.

<표 5> 본 연구에서 시행된 피지컬 컴퓨팅 수업

구분	피지컬 수업단계	활동내용
미로탈출	문제분석 및 아이디어 찾기	로봇의 움직임 탐구
	알고리즘의 형식화 및 수행	로봇을 이용해 길찾기
	알고리즘의 수행 및 반성	미로탈출 실행
스쿨버스	문제분석 및 아이디어 찾기	마을 지도 및 스쿨버스 제작하기
	알고리즘의 형식화 및 수행	스쿨버스 제작 및 프로그래밍
	알고리즘의 수행 및 반성	스쿨버스의 움직임 최적화

미로탈출 수업의 경우 문제를 해결하기 위해 로봇의 형태를 수정할 수 있게 조립아이디어가 포함된 반구조화된 로봇을 활용해 프로그래밍하도록 수업이 진행되었다. 다음 [그림 2]는 미로탈출 수업에서 학생들이 완성한 로봇 및 미로이다.



[그림 2] 미로탈출 수업에서 사용된 로봇 및 미로

학생들이 알고리즘과 프로그래밍에 집중하도록 기본적인 로봇 형태를 제시하였고, 미로를 이동하기 위해 학생들이 각자의 방식으로 로봇을 변형하여 사용하였다. 제시된 미로는 로봇들이 중간지점과 최종 골인 지점으로 2단계로 나누어 수행 목표를 제시하여 학생들이 단계적으로 과제를 해결하도록 하였다.

다음 [그림 3]은 학생들이 창작한 스쿨버스들과 스쿨버스의 이동 모습이다. 스쿨버스 수업은 학생들이 살고있는 지역상황을 고려하여 문제해결에 필요한 스쿨버스(로봇)를 직접 창작하고 스쿨버스가 이동하는 거리를 지도형태로 구체화시켜서 문제를 해결하였다.

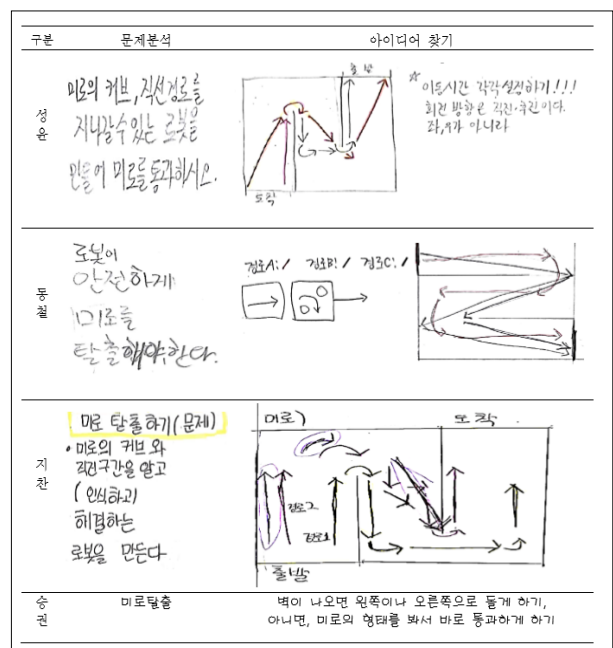


[그림 3] 학생들이 창작한 스쿨버스들과 스쿨버스의 이동 모습

학생들에게 각 수업별 3주간 작성하도록 활동지를 제공하였다. 활동지는 피지컬 컴퓨팅 수업단계에 맞춰서 학생들이 직접 기록하도록 구성되었다. 수업 과정은 사진 및 영상으로 기록되었고 각 수업이 종료시 산출물을 활용한 인터뷰를 진행하였다.

4.2 적용 결과

미로탈출 수업을 중심으로 학생들의 컴퓨팅 실천의 변화를 살펴보면, 학생들의 문제분석(AP) 및 아이디어 찾기(FI) 단계에서는 문제상황을 분석하여 문제를 재정의하고 미로를 탈출하기 위한 아이디어들을 찾았다. 이를 위해 미로가 어떻게 생겼는지 관찰하고 로봇을 이동시키는 방법을 찾아보면서 아이디어를 수집하였다. 학생들이 작성한 세부적인 내용은 다음 [그림 4]와 같다.



[그림 4] 미로탈출 수업에서 나타난 학생들의 문제분석 및 아이디어 찾기 유형

문제분석(AP) 단계는 문제의 이해와 구조화 수준을 파악하는 것이다. 1수준의 경우 문제에 대해 설명하지 못하는 경우 또는 제시된 문제를 단순 반복하는 경우인데, 동철과 승권의 응답유형은 1수준으로 분석되었다.

이와 비교하여 성윤과 지찬의 경우는 문제분석 단계부터 미로탈출을 위해서 로봇의 움직임이 달라져야 함을 인지하고 그에 따라서 2가지 형태의 이동 경로를 구상하였다. 이는 문제분석단계에서는 핵심 정보를 찾아내는 것이 필요하며, 이는 아이디어를 찾기 위한 근거가 되는 것으로 파악되었다. 또한 컴퓨팅 사고에서 강조하는 문제분해와 추상화가 나타난 것으로 분석되었다.

관계를 이해하고 있는 것으로 파악되었다.

학생들의 활동내용과 생각을 파악하기 위해 산출물을 활용한 인터뷰를 진행하였다. 인터뷰 결과 학생들은 로봇을 이동시키기 위해 모터를 제어하는 것의 필요성을 인지하고 있었다.

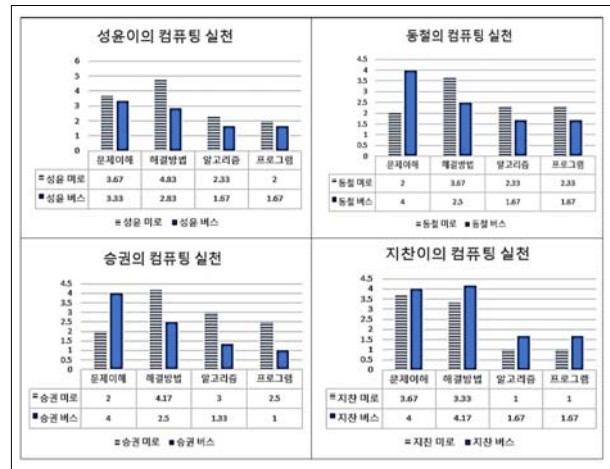
학생들은 로봇의 이동 방향을 제어하기 위해 모터의 회전 방향 및 지속시간을 변경하는 것이 필요하다는 것을 이해하고 있음이 나타났다. 또한 단위행동의 수행을 위해 명령어를 순서대로 배열하는 것과 동작을 반복하기 위한 활용하고 있어 순차명령 및 반복명령을 활용한 프로그래밍 방식을 이해하고 있는 것으로 나타났다.

한편 학생들은 자신의 로봇이 교실 바닥에서 움직일 때 로봇의 구조에 따른 바닥의 저항 등과 같은 물리적 오류를 해결하기 위해 반복적인 실험을 통해 프로그램을 보완하고 수정했다고 보고하였다. 이는 학생들이 직접 설계 및 제작을 할 수 있는 로봇을 활용하여 실시간으로 제어하는 활동을 통해 문제해결을 위해 능동적으로 대처하는 양상을 보여주는 것이라 할 수 있다.

로봇을 활용하여 문제를 해결하는 과정에서 아이디어를 나누고 다른 사람의 아이디어를 적극적으로 활용함으로써 로봇을 통해 동료와 협력하는 컴퓨팅 관점의 변화를 경험하고 있음이 관찰되었다. 다만 기존 알고리즘 형식화(FA) 단계에서 프로그래밍 명령어로 바로 전환하여 표현하였기에 일의 순서를 점검하기보다는 명령어의 개별기능에 치중하는 모습이 나타났다.

향후 학생들을 지도할 때 알고리즘 형식화(FA) 단계에서 우선 글, 그림, 기호 등의 방식으로 일의 순서를 표현하는 방안을 모색해야 할 것이다. 이러한 수학적 표현의 활성화를 통해 일의 순서를 시각적으로 구체화시킬 수 있어 자신의 아이디어를 점검하고 반성할 기회를 마련하게 될 것이다.

피지컬 컴퓨팅 수업에서 나타난 개별 학생들의 컴퓨팅 실천을 영역별로 구분해서 평균점으로 나타낸 결과는 다음 [그림 7]과 같다. 미로탈출 수업이 컴퓨팅 실천의 점수가 스쿨버스 수업과 비교하여 상대적으로 높은 편으로 보고되었다.



[그림 7] 개별 학생들의 컴퓨팅 실천 변화

미로탈출 수업의 경우 로봇 몸체를 제작하기 위한 조립도를 제공하여 학생들이 로봇 제작에 어려움을 겪지 않고 프로그래밍에 집중할 수 있었다. 이와 비교하여 스쿨버스 수업의 경우는 학생 스스로 로봇을 창작할 기회가 제공되어 창의적 성향을 발휘할 기회가 되었다.

그렇지만 로봇제작에 과도하게 집중하는 경우 프로그래밍에 할애할 시간이 상대적으로 부족하여 로봇이 제대로 기능하지 못하는 한계점이 나타났다. 특히 학생들이 프로그래밍 초보자인 경우 로봇제작의 부담을 느끼게 되면 문제해결에 어려움을 느낄 수 있어 수업에 대한 흥미와 참여에 영향을 미치게 되었다.

학생들의 개별적인 변화를 살펴보면, 지찬은 미로탈출 수업과 비교하여 스쿨버스 수업에서 전체적으로 향상이 나타났다. 문제이해와 구조화영역의 경우 동철, 승권, 지찬은 미로탈출 수업과 비교하여 스쿨버스 수업에서 향상이 나타났는데, 특히 동철과 승권은 높은 향상이 보고되었다. 이는 스쿨버스 수업이 학생들이 살고 있는 지역사회의 교통문제와 연관되어 흥미를 유발하였고 문제를 파악하는데 도움이 되는 것으로 분석되었다.

연구결과 피지컬 컴퓨팅 수업에 참여한 학생들은 로봇을 제어하기 위해 프로그래밍을 경험하면서 컴퓨팅 개념에서 다루고 있는 추상화, 문제분해, 알고리즘의 개념을 획득하게 되었다. 이는 프로그래밍 초보자들에게 학습매체를 직접 설계 및 생성하고 개발하는 활동을 제공하는 것이 최상의 학습효

과를 얻을 수 있으며[54], 컴퓨팅 사고력[12] 및 프로그래밍 언어에 대한 이해가 향상되었다는 연구결과들[55][56]과 일치한다.

5. 결론 및 제언

본 연구에서 나타난 결과로부터 도출된 결론은 다음과 같다. 첫째, 본 연구에서 개발된 피지컬 컴퓨팅 수업은 컴퓨팅 실천을 통해 문제분해, 추상화, 알고리즘의 형식화를 경험하도록 하여 컴퓨팅 사고의 형성을 지원한다.

특히 피지컬 컴퓨팅 수업은 로봇 제작 및 프로그래밍이 가능하여 로봇과 같은 실제 모델(real model)을 각자의 멘탈 모델(mental model)과 비교하고 수정할 기회를 제공하였다. 이는 능동적인 지식 구성을 돕고 컴퓨팅 실천을 통해 컴퓨팅 개념을 이해하는 데 도움이 되었다. 또한 스쿨버스의 이동 경로를 표현한 지도 만들기 활동은 핵심정보를 추출하여 구체화함으로써 문제분해 및 추상화와 같은 컴퓨팅 개념을 형성하는 기회가 되었다.

둘째, 피지컬 컴퓨팅 수업은 물리적 모델을 제어하는 하드웨어적인 방식과 프로그래밍과 같은 소프트웨어적인 방법으로 문제를 해결하는 방법을 시도해보면서 각자 구상한 아이디어를 컴퓨팅 기술로 실현함으로써 컴퓨팅 사고가 발현되도록 하였다. 특히 알고리즘을 형식화하는 활동에서 학생들이 기호, 표, 그림 등을 활용함으로써 수학적 표현을 활성화시킨다는 점에서 교육적 의미를 발견할 수 있었다.

셋째, 피지컬 컴퓨팅 수업에서 제시한 미로탈출과 스쿨버스와 같은 실제적 맥락의 문제상황은 학생들이 컴퓨팅 기술을 통해 문제를 해결하도록 기회를 제공하였다. 미로탈출 수업에서 학생들은 로봇 2대를 함께 제어하면서 협력의 필요성을 인식하였다. 활동과정에서 자신이 발견한 해결책을 적극적으로 공유하였고, 스쿨버스 수업의 문제를 고민하면서 컴퓨팅 기술이 지역사회의 문제를 해결하는데 활용될 수 있다고 인식하였다.

지금까지 제시한 결론을 기반으로 다음과 같이 제언하고자 한다. 첫째, 피지컬 컴퓨팅 수업에서 학생들이 로봇을 제작하고 프로그래밍하는 활동이 컴퓨팅 개념의 형성을 지원한 점을 고려하여, 향후

소프트웨어교육에서 학생들이 물리적으로 제어가능한 컴퓨팅 실천 활동을 포함해야 할 것이다.

본 연구의 경우 학생들에게 친숙한 매체를 활용함으로써 수업의 흥미와 참여도가 높아진 점을 고려하여 학습자의 사전 경험을 파악하여 적절한 테크놀로지와 매체를 활용해야 할 것이다. 한편, 본 연구에서는 모듈형 피지컬 컴퓨팅 도구를 활용해 연구를 진행하였는데, 앞으로 로봇형 도구 및 보드형 도구와 같은 다른 유형의 피지컬 컴퓨팅 도구에 대한 추가적인 연구가 필요하다.

둘째, 성취도가 높은 학생들이 알고리즘을 구체화시키는 단계에서 자신의 아이디어를 프로그래밍 언어로 즉각적으로 표현한 점을 고려해야 할 것이다. 학생들은 전체적인 일의 흐름을 파악하고 분석하기 보다는 로봇의 작동여부 및 세부 기능에 집중하였다.

이러한 점을 고려하여 학생들에게 자신의 생각을 글자, 그림, 기호 등을 이용해 표현하는 방식을 우선 지도할 필요가 있다. 특히 프로그래밍 초보자들의 경우에 알고리즘의 형식화가 쉽지 않다는 연구결과[34]를 고려하여, 수학교과에서 수학적 표현의 활성화와 함께 알고리즘을 형식화하고 수행 및 반성하는 과정을 경험하도록 기회를 제공해야 할 것이다.

셋째, 학생들을 지도할 때 추상화의 과정을 경험한 후에 자동화 단계를 접근하는 것이 필요하다. 피지컬 컴퓨팅 수업에 참여한 학생들은 문제상황을 인식하는 관점과 방식에 따라 문제해결을 위한 아이디어 및 알고리즘이 달라짐이 나타났다.

이러한 점을 고려하여 문제분해 및 패턴인식을 포함한 추상화에 대한 교육이 우선 전개되어야 할 것이다. 이와 관련하여 지도 그리기와 같은 활동이 핵심요소의 파악과 패턴을 추출하는 추상화에 도움이 된 점을 고려하여 향후 사회교과 및 미술교과 등과 연계한 수업을 개발할 필요가 있다.

넷째, 본 연구는 자발적 참여를 전제로 하였기에 남학생만을 대상으로 연구가 실행되었다. 따라서 여학생을 대상으로 후속연구가 필요하다. 본 연구에 참여한 학생들의 경우 관련 경험이 있었고, 로봇과 프로그래밍에 대한 흥미가 높았는데, 향후 여학생들에 대상으로 관련 경험이 어떠한지 파악하는 것과 더불어 피지컬 컴퓨팅 수업 경험에 따른 컴퓨

팅 사고의 변화와 특징에 대한 후속 연구가 필요하다.

참 고 문 헌

- [1] 류미영·한선관 (2015). 초등 SW 교육을 위한 CT 교육 프로그램 개발. **한국정보교육학회논문지**, 19(1), 11-20.
- [2] 교육부 (2015a). **초·중등학교 교육과정 총론**. 교육부 고시 제 2015-74호(별책1). 서울: 교육부.
- [3] 김갑수 (2017). 한국 초등교사 양성 기관의 소프트웨어 교육과정 분석. **한국정보교육학회논문지**, 21(6), 723-732.
- [4] 김수환 (2015). Computational Thinking 교육에서 나타난 컴퓨터 비전공 학습자들의 어려움 분석. **한국컴퓨터교육학회논문지**, 18(3), 49-57.
- [5] 김한성·전수진 (2017). SW 교육 평가 방법의 적합성 및 현장 유용성에 대한 초등교사의 인식분석. **한국정보교육학회논문지**, 21(3), 267-275.
- [6] 신승용·조혜경·김미량 (2013). 초·중등학생의 로봇교육을 위한 수학, 과학과 교육과정 연계 로봇 소양 교육과정 개발. **한국컴퓨터교육학회논문지**, 16(6), 55-70.
- [7] 김동정·심석희·유현창 (2013). 피지컬 컴퓨팅 기반 STEAM 교육 아웃리치 프로그램. **한국컴퓨터교육학회 학술발표대회 논문집**, 17(2), 279-283.
- [8] 김태훈 (2015). **컴퓨팅 사고력 신장을 위한 프로그래밍 중심 STEAM 교육 프로그램**. 박사학위논문. 제주대학교.
- [9] 송의성 (2013). 교육용 로봇을 활용한 예비초등교사 로봇교육프로그램의 개발 및 적용. **디지털콘텐츠학회논문지**, 14(3), 333-341.
- [10] 정종원·박인우·임병노·이영준·유현창·김현진, 고범석 (2013). **테크놀로지 발전방향과 교육의 미래 모습을 통한 미래교실 구성**. 연구자료 RM 2009-10. 서울: 한국교육학술정보원.
- [11] 송의성·길준민 (2017). 초등예비교사의 컴퓨팅 사고력 향상을 위한 블렌디드 러닝 기반의 소프트웨어교육 프로그램 개발 및 적용. **한국정보처리학회논문지**, 6(7), 353-360.
- [12] 신승기·배영권 (2015). 인도의 초등학교 컴퓨터 교육에 대한 분석 및 시사점. **한국정보교육학회논문지**, 18(4), 585-594.
- [13] 김재휘·김동호 (2016). 컴퓨팅 사고력 향상을 위한 초등 피지컬 컴퓨팅 교육과정 개발. **한국정보교육학회논문지**, 20(1), 69-82.
- [14] Alimisis, D. (2013). Educational Robotics: Open questions and new challenges. *Themes in Science & Technology Education*, 6(1), 63-71.
- [15] Felica, A., & Sharif, S. (2014). A reivew on educational robotics a assistive tools for learning mathematics. *International Journal of Computer Science Trends and Technology*, 2(2), 62-84.
- [16] 김성종·김현진·이승진 (2014). **미래 핵심역량 증진을 위한 교육정보화의 역할**. 연구자료 RM 2014-16. 대구: 한국교육학술정보원.
- [17] 이영주·조영환·조규락·최재호 (공역) (2015). **테크놀로지와 함께하는 유의미학습**. Howalnd, J., Jonassen, D. H., & Mara, R. M의 **Meaningful Learning With Technology**. 서울:아카데미프레스. (원서출판 2011)
- [18] 박성익·임철일·이재경·최정임·임정훈·정현미·송해덕·장수정·장경원·이지연·이지은 (2012). **교육공학의 원리와 적용**. 서울: 교육과학사.
- [19] 김민경·민선희 (2008). 로봇틱스 탐구에서 나타난 아동의 반응 및 교육적 의미 분석. **한국실과교육학회지**, 21(3), 181-200.
- [20] Papert, S. (1980). *Mindstorms: children, computer and powerful ideas*. NY: Basic Books.
- [21] Elkin, M., Sulilivan, A., & Bers, M. U. (2014). Implementing a robotics curriculum in and early childhood Montessori classroom. *Journal of Information Technology Education: Innovations in Practice*, 13, 153-169.
- [22] Kabábotá, M., & Peárová, J. (2010). Learning how to teach robotics. *Constructionism*, 1-18.
- [23] Romero, M., & Dupont, Y. (2016). Educational robotics: From procedural

- learning to co-creative project oriented challenges with LEGO WeDo. 6159-6163.
- [24] 김대수 (2017). **소프트웨어와 컴퓨팅 사고**. 파주: (주)생능출판사.
- [25] 양재명 · 김자미 · 이원규 · 윤일규 · 서정화 · 유효성 · 양혜지 · 김민정 · 최희정 (2017). **초·중등 소프트웨어(SW) 교육 역량 진단도구 개발 연구**. 연구보고 RR 2017-3. 대구: 한국교육학술정보원.
- [26] 교육부 (2015b). **소프트웨어 교육 운영지침**. 서울: 교육부.
- [27] 김진숙 · 한선관 · 김수환 · 정순원 · 양재명 · 장의덕 · 김정남 (2015). **SW 교육 교수학습 모델 개발 연구**. CR 2015-35. 서울: 한국교육개발원.
- [28] K-12 computer science standards recommended by Computer Science Teachers Association (2017). Retrieved June 20, 2018, from <https://csteachers.org/page/standards>
- [29] 김영애 · 김갑수 · 김재현 · 김한성 · 양재명 · 이승진 · 정진명 · 최현종 · 채경화 · 최영선 · 장원영 (2015). **소프트웨어교육 운영지침 개발 연구**. 연구보고 CR 2015-3. 대구: 한국교육학술정보원.
- [30] Allen, W., Coulter, B., Denner, J., Lee, I., Malyn-Smith, J., & Martin, F. (2010). *Computational thinking for youth*. White paper for the ITEST small working group on computational thinking (CT).
- [31] Lee, I., Martin, F., Denner, B., Allen, W., Erickson, J., Malyn-Smith, J., & Wenner, L. (2011). Computational thinking for youth in practice. *Association for Computing machinery*, 2(1), 32-37.
- [32] Creative Computing recommended by Brennan, K., Balchm, C., & Chung, M. (2015). Retrieved June 9, 2018, from <http://scratched.gse.harvard.edu/guide/>
- [33] 전수진 (2017). SW 교육에서의 컴퓨팅 사고력 신장을 위한 개발 중심 모형의 설계 및 효과. **한국정보교육학회논문지**, 21(6), 619-627.
- [34] Futschek, G., & Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing Algorithms. *Constructionism*, 1-10.
- [35] 김지현 · 김태영 (2016). 중등 수학과학 영재를 위한 피지컬컴퓨팅 교육이 융합적 역량 향상에 미치는 영향. **한국컴퓨터교육학회논문지**, 19(2), 87-98.
- [36] 강현석 · 이원희 · 허영식 · 이자현 · 유재순 · 최윤경 (공역) (2008). **거꾸로 생각하는 교육과정 개발-이론편, 교과에 대한 진정한 이해를 목적으로**. Wiggins, G., & McTighe, J의 *Understanding by Design*. 서울: 학지사 (원본출판 2005)
- [37] Wing, J. M. (2006). Computational thinking. *Communications of the Association for Computing Machinery*, 19(3), 33-35.
- [38] 한국컴퓨터교육학회 (2014). **Computational thinking & 창의적 문제해결 방법론**. 고양: 이한미디어.
- [39] 김자미 (2017). 컴퓨팅 사고력 기반의 문제해결 능력. **한국정보처리학회논문지**, 24(2), 13-21.
- [40] Denning, P. J. (2009). Beyond computational thinking. *Communication of the ACM*, 52(6), 28-30.
- [41] Yadav, A., Hong, H., & Stepheson, C. (2016). Computational thinking for all: Pedagogical approached to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, 60(6), 565-568.
- [42] Grover, S., & Pea, R. (2013). Computational thinking in K-12: Review of the state of the field. *Educational Research*, 42(1), 38-43.
- [43] Bers, M. U. (2010). The TangibleK robotic program: Applied computational thinking for young children. *Early Childhood Research & Practice*, 12(2), 1-19.
- [44] 김종혜 (2009). **정보과학적 사고 기반의 문제 해결 능력 향상을 위한 중등교육 프로그램**. 박사학위논문. 고려대학교.
- [45] 양재명 · 이원규 · 김자미 · 서정희 · 유효성 · 양혜지 · 김민정 · 최희정 (2017). **2017년도 소프트웨어(SW)교육 효과성 측정도구 개발연구**. 연구보고 CR 2017-8. 대구: 한국교육학술정보원.

[46] Bebras Computational Challenge recommended by UK Bebras challenges (2015). Retrieved Jan. 10, 2016 from <http://bebras.uk>

[47] 전용주 · 정웅열 · 김인주 · 김지혜 · 이현아 · 김동윤 (2018). 소프트웨어 교육에서 비버챌린지를 활용한 교수학습 및 평가 방법 탐색. *한국컴퓨터교육학회논문지*, 21(6), 63-82.

[48] 최형신 (2014). Computational Thinking 역량 계발을 위한 수업 설계 및 평가 루브릭 개발. *한국정보교육학회논문지*, 18(1), 57-64.

[49] Brennan, K., & Resnick, M. (2012). Using artifact-based interviews to study the development of computational thinking in interactive media design. *Paper presented at annual American Educational Research Association meeting*, Vancouver, BC, Canada.

[50] Wenitrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Educational and Technology*, 25(1), 127-147.

[51] 신승기 · 배영권 (2014). 인도의 초등학교 컴퓨터 교육에 대한 분석 및 시사점. *한국정보교육학회논문지*, 18(4), 585-594.

[52] 최돈형 · 조성화 · 안재정 · 홍현진 · 정국초 (2015). 교사가 실천하는 지속가능발전교육: 미래세대와 동행하기. 서울: 유네스코한국위원회.

[53] 최형신 (2014). Computational Thinking (CT) 역량 계발 방안: CT의 5가지 핵심역량 분석. *한국정보교육학회 논문지*, 20(6), 553-562.

[54] Kafari, Y. B., & Resnick, M. (1996). *Constructions in practice: Designing, thinking and learning in digital world*. Routledge.

[55] Ricca, B., Lulis, E., & Bade, D. (2006). *Lego mindstorms and the growth of critical thinking*. Intelligent tutoring systems workshop on teaching with robots. Agents, and NLP.

[56] Resnick, M. (2006). Computer as pain

brush: Technology, play and the creative society. *Play=learning: How play motivated and enhances children's cognitive and social-emotional growth*, 192-208.



민 선 희

2005 숙명여자대학교
유아교육정보
(유아교육학석사)
2018 이화여대 초등교육학과
(문학박사)

2006~2009 숙명여대 창조적 리더십센터 연구원
2011~2017 LEGO education Development
Manager

관심분야: Computational Thinking, 융합교육,
피지컬 컴퓨팅, 창의성, Playful
Learning

E-Mail: sunnym73@naver.com



김 민 경

1985 이화여자대학교
수학과(이학사)
1987 이화여자대학교
수학교육전공(교육학석사)

1997 Iowa State University
Dept. of Curriculum & Instruction
(Ph.D.)

관심분야: 교육공학, 컴퓨터교육, 문제해결,
창의성 및 융합교육

E-Mail: mkkim@ewha.ac.kr