

고등학생 대상 메이커 교육이 소프트웨어 교육에 대한 태도, 창의적 문제해결력, 컴퓨팅 사고에 미치는 영향

홍원준* · 최재성** · 이현**

선문대학교 로봇플랫폼 연구소* · 선문대학교 컴퓨터공학부**

요약

본 연구에서는 지역 고등학생들을 대상으로 아두이노를 기반으로 하는 인공지능 메이커 교육을 실시하고 소프트웨어 교육에 대한 태도, 창의적 문제해결력, 컴퓨팅 사고에 미치는 영향을 관찰하였다. 이를 위해 mBlock과 아두이노를 기반으로 인공지능 로봇을 만들 수 있는 교육을 설계하여 학교 외부의 메이커 스페이스에서 교육을 실시하였다. 교육에 참여한 20명의 학생 중 19명의 데이터를 분석하였고, 대응표본 t-검정 결과 모든 부분에 있어 향상하였음을 확인할 수 있었다. 또한 학생들의 학업성취도 및 만족도에 대한 예측변인을 확인하기 위해 다중회귀분석을 실시하였다. 그 결과 소프트웨어 교육에 대한 태도가 유의한 예측변인으로 확인되었다. 본 연구를 통해 메이커 교육이 소프트웨어 교육에 대한 태도, 창의적 문제해결력, 컴퓨팅 사고와 긍정적인 영향을 미친다는 사실을 확인할 수 있었다. 이러한 결과를 바탕으로 연구의 의의와 시사점을 도출하였고, 마지막으로 연구의 제한점과 추후 연구를 제안하였다.

키워드 : 메이커 교육, 인공지능, 소프트웨어에 대한 태도, 창의적 문제해결, 컴퓨팅 사고

Effects of maker education for high-school students on attitude toward software education, creative problem solving, computational thinking

Wonjoon Hong* · Jae-Sung Choi** · Hyun Lee**

Robot Platform Institute, Sunmoon University* · Department of Computer Engineering,
Sunmoon University**

Abstract

The purpose of this study is to examine effects of maker education for high-school students on attitude toward software education, creative problem solving, and computational thinking. The program was designed to develop an artificial intelligence robot using mBlock and Arduino and implemented at a maker space. We analyzed 19 students among 20 who participated in the program, the result of paired t-test indicated significant increase in all variables. Also, we performed a multiple regression analysis to investigate predictors of perceived achievement and satisfaction. The finding demonstrated an initial attitude toward software education was found to be the significant predictor of perceived achievement and satisfaction. With the results, we confirmed maker education enhances attitude toward software education, creative problem solving, and computational thinking. Lastly, we discussed the implications and limitations and suggested the direction for future research.

Keywords : maker education, artificial intelligence, attitude toward software education, creative problem solving, computational thinking

본 과제(결과물)는 교육부의 재원으로 지원을 받아 수행된 산학협력 선도대학(LINC) 육성사업의 연구결과입니다.

교신저자 : 최재성(선문대학교 컴퓨터공학부)

논문투고 : 2020-09-24

논문심사 : 2020-10-27

심사완료 : 2020-12-07

1. 서론

최근 IT(Information technology) 분야의 급속한 발달로 4차 산업혁명이 우리의 일상에 들어오면서 이제는 산업혁명 시대의 토지, 자본과 같은 유형의 자원이 아니라 창의성, 문제해결력 등과 같이 복잡한 문제를 효과적으로 해결할 수 있는 지적능력이 보다 중요해지고 있다. 즉, 사회가 복잡해짐에 따라 발생하는 문제 또한 과거에 비해 보다 많은 시간과 노력을 필요로 하게 되었다. 그러므로 이러한 새로운 유형의 문제를 효율적으로 해결하기 위한 사고능력의 필요성이 대두된다.

하지만 일반적으로 이러한 사고능력은 스스로 형성되기 힘들고, 더구나 기존의 교육환경 및 교육방식 하에서 배양하는 것 또한 쉽지 않다. 메이커 교육은 학생들이 이러한 고차원적인 사고능력을 효과적으로 학습할 수 있는 교육방법으로 최근에 많은 주목을 받고 있다[11][14]. 메이커 교육은 자율성과 창의성이 존중되는 환경에서 학생들이 스스로 자유롭게 3D 프린터, 아두이노(Arduino) 등 다양한 도구를 활용하여 문제를 해결하며 결과물을 만들어 다른 사람과 공유함으로써 사회적 발전에 이바지하는 전반적인 문화를 의미한다[8]. 또한 학생들에게 자율성을 부여하는 이러한 접근은 학생들로 하여금 교육 참여도와 흥미를 향상시키는 장점이 있다[3]. 따라서 기존의 경직되어 있던 교육환경에서 진행될 때와는 다르게 다양한 교육이 흥미로운 놀이 혹은 취미 활동처럼 인식되면서 학생들에게 교육에 대한 긍정적인 인상을 주고 이로 인해 학생들의 적극적인 참여와 나아가 높은 학업 성취도 및 만족도를 기대할 수 있다[5].

최근 메이커 교육을 통해 학습할 수 있는 다양한 사고 능력 중 창의적 문제해결력(creative problem solving)은 요즘 같은 복잡한 지식사회를 살아가는 대부분의 사람들이 갖추어야 할 필수적인 능력이 되어가고 있다. 이제는 과거처럼 수동적으로 교수자로부터 내용을 주입받고 단순히 암기만 해서는 현시대의 복잡한 문제들에 효과적으로 대응할 수 없다[26]. 문제에 대한 보다 효과적인 해결책을 찾기 위해서는 과거와는 달리 다양한 각도로 문제를 볼 줄 알아야 한다. 창의적 문제해결력은 발산적 사고와 수렴적 사고기술을 활용해 기존과는 다른 가장 효과적인 해결 방법을 체계적인 절차를 통해 찾을 수 있는 방법을 제시해준다[29].

이어서 최근 복잡한 문제를 해결하는 데에 있어서 컴퓨터의 효과적인 활용은 필수가 되었다. 이에 문제에 대한 창의적인 접근뿐만 아니라, 컴퓨터처럼 다양한 정보 처리 방식을 사용해 철차적으로 처리하는 컴퓨팅 사고(computational thinking)에 대한 관심 또한 크게 증가하고 있다. 즉, 컴퓨터가 문제를 해결할 때 사용하는 추상화(abstraction), 자동화(automation), 알고리즘(algorithm), 병렬화(parallelization) 등의 방식을 활용해 복잡한 문제를 효율적으로 해결하는 접근 방법에 주목하고 있다[31]. 또한 2018년도부터 초·중등학교 학생들은 4차 산업혁명 시대의 핵심 문제해결 도구인 코딩을 의무적으로 학교에서 교육받는다. 코딩의 효과적인 활용은 유연한 컴퓨팅 사고가 전제되어야 한다는 점에서 컴퓨팅 사고 능력은 청소년들에게 요구되는 필수 사고 능력이라고 할 수 있다.

따라서 본 연구에서는 고등학생들을 대상으로 하는 인공지능 메이커 교육이 학생들의 소프트웨어 교육에 대한 태도와 현대사회가 필요로 하는 창의적 문제해결력 및 컴퓨팅 사고에 어떠한 영향을 미치는지 알아보고자 한다. 이를 바탕으로 본 연구에서 탐색하고자 하는 연구문제는 다음과 같다.

첫째, 인공지능 메이커 교육이 고등학생들의 소프트웨어 교육에 대한 태도를 향상시키는가?

둘째, 인공지능 메이커 교육이 고등학생들의 창의적 문제해결력을 향상시키는가?

셋째, 인공지능 메이커 교육이 고등학생들의 컴퓨팅 사고를 향상시키는가?

넷째, 교육에 대한 태도, 창의적 문제해결력, 컴퓨팅 사고가 지각된 학업성취도와 만족도를 예측할 수 있는가?

2. 이론적 배경

2.1. 메이커 교육

메이커 교육은 구성주의(constructivism)에 뿌리를 두고 있으며, 성공적인 교육을 위해 능동적 학습(active learning)과 메이커 스페이스(maker space)라고 불리는 교육을 수행하기 위한 준비된 환경을 강조한다[3]. 메이커 스페이스는 학생들 스스로 동료들과 자유롭게 메이킹(making) 활동을 하며 구성주의 학습이 이루어질 수

있는 기회를 제공하는 공간을 의미한다[33]. 이러한 환경은 학교 교실과 같은 기존의 전통적인 학습공간과 차별화되며, 학생들로 하여금 주도적이고 창의적인 학습이 가능하도록 도와준다[14]. 즉, 학생들은 실생활 문제를 기반으로 다양한 질문을 통해 스스로 해결 방법을 찾고 남들과 공유할 수 있는 결과물을 만든다. 그 과정 속에서 교수자는 직접적으로 학생들에게 지식을 주입해주는 지식 전달자의 역할이 아니라 학생들이 스스로 탐구할 수 있도록 방향을 제시해 주는 가이드의 역할이 더 강조된다[8]. 메이커 교육에서 메이커(Maker)는 단순히 무언가를 만드는 사람이 아니라 스스로 문제를 구성하고, 탐색하며, 다양한 정보와 도구를 활용하여 완성된 결과물을 제작하는 주도적인 역할의 행위 주체를 말한다[7]. 이러한 과정 속에서 주변의 다른 메이커들과 서로 의견을 공유하며 더 나은 아이디어를 찾기 위해 노력한다[9].

메이커 교육의 효과는 기존의 연구결과에서 확인할 수 있다. 초등학교를 대상으로 한 메이커 교육과 일반적인 강의식 수업의 비교에서는 메이커 교육을 통해 수업을 들은 학생들이 그렇지 않은 학생들에 비해 창의적 문제해결력이 더 많이 향상된 것으로 나타났다[18]. Yin, Hadad, Tang, & Lin(2019)은 전자섬유(e-textile), 전자회로(electric circuit), 아두이노(Arduino)를 포함해 다양한 도구들을 활용하여 메이커 교육이 어떻게 컴퓨팅 사고에 영향을 미치는지 연구를 하였다[33]. 그 결과 메이커 수업을 받은 이후 학생들의 컴퓨팅 사고에 대한 능력이 받기 전에 비해 유의한 수준으로 향상되었음을 확인할 수 있었다. 최근 이루어지는 메이커 교육을 보면, 한국의 메이커 교육은 주로 3D 프린터, 로봇, 인공지능 등의 교육에 집중되어 있지만, 외국에서는 이외에도 바느질, 용접, 요리, 그림, 블록 등 매우 다양한 활동을 활용해서 교육을 실시하고 있다[24].

2.2. 창의적 문제해결력

창의성(Creativity)은 참신하고 유용한 아이디어를 생산할 수 있는 능력으로 정의할 수 있다[20]. 이렇듯 창의성이 인지 과정을 통한 창의적인 아이디어의 형성을 의미한다고 본다면, 창의적 문제해결력은 보다 문제해결(problem solving)에 초점을 맞춘 인지 과정이라고 할

수 있다[21].

Treffinger, Isaksen, & Dorval(2005)에 의하면 일반적으로 창의적 문제해결은 반복적인 발산적 사고와 수렴적 사고를 바탕으로 주어진 상황에서 문제를 발견하고, 관련된 자료를 탐색하며, 창의적으로 아이디어를 생성하여 문제를 해결하는 일련의 과정을 말한다[29]. 구체적으로 보면 가장 최근 모델인 Creative problem solving 6.1은 문제의 이해, 아이디어 생성, 행위를 위한 계획의 세 가지 행동요소를 포함하고 있으며, 각각의 요소는 하위 단계를 가지고 있다. 문제의 이해에는 관심문제 발견, 자료 발견, 문제 발견 단계가 있으며, 아이디어 생성에는 아이디어의 발견 단계가 있다. 마지막으로 행위를 위한 계획에는 해결 발견 단계와 수용 발견 단계가 있다.

이상수와 이유나(2007)에 따르면 발산적 사고는 문제해결과 관련된 다양한 창의적인 아이디어를 도출하기 위해 활용되고, 수렴적 사고는 도출된 아이디어들을 비판적으로 분석하여 정교화 시키고 결과적으로 가장 이상적인 아이디어를 선정한다[16]. 이 과정에서 사용되는 비판적 사고의 방식에는 분석적 사고, 추론적 사고, 종합적 사고, 대안적 사고 등이 포함되며[26], 이러한 인지적 사고과정을 통해 학생들은 하나의 정해진 해답이 아닌 다양한 후보군들을 탐색하고 더 나아가 분석적 비판능력을 바탕으로 그 안에서 최적의 결과를 도출하기 위해 노력한다[29].

메이커 교육을 통한 창의적 문제해결력 향상은 최근 많은 관심을 받고 있다[15][18][28]. 학생들은 주어진 환경 안에서 자유롭게 문제를 탐색하고 그에 맞는 답을 찾기 위해 창의적으로 본인들의 아이디어를 발전시켜 나가볼 수 있다. 그뿐만 아니라 발산적 사고로부터 다양한 아이디어를 도출하고, 나아가 비판적 사고를 통해 주어진 문제를 해결하기 위한 최적의 답을 찾아가게 된다[9]. 최종적으로 본인의 아이디어들이 반영된 실제 결과물을 제작 및 공유하며 학생들은 학습에 대해 흥미를 느끼며 집중할 수 있게 된다[27].

메이커 교육과 창의적 문제해결력의 관계에 관한 연구를 살펴보면, 윤정구와 김영식(2018) 아두이노를 활용하여 메이커 교육의 창의적 문제해결력에 미치는 효과에 대해 연구하였다[34]. 연구를 위해 한 집단은 기존의 텍스트 기반의 프로그래밍 언어를 활용한 수업을 진행

하였고, 다른 집단은 아두이노를 활용한 프로그래밍 학습을 진행하였다. 결과적으로 아두이노를 활용한 실험 집단이 그렇지 않은 통제집단보다 창의적 문제해결 능력의 모든 하위 요소(특정 영역의 지식/사고 기능/기술의 이해 및 숙달 여부, 확산적 사고, 비판적 논리적 사고, 동기적 요소)에서 유의한 수준으로 높게 증가하였다. 앱 인벤터와 아두이노를 이용한 최숙영과 김세민(2016)의 연구에서는 확산적 사고와 동기적 요소의 향상을 확인할 수 있었다. 그뿐만 아니라 많은 학생들이 교육을 통해 배운 지식을 본인들의 일상생활에 효과적으로 적용 가능하다고 답변하였다[5].

2.3. 컴퓨팅 사고

‘컴퓨팅 사고’라는 용어는 1980년 Papert(1980)의 저서 *Mindstorms*에서 처음 사용되었다[22]. 그에 따르면 컴퓨터는 인간의 사고력을 강화시키고 또한 지식에 접근하는 방식을 변화시킨다고 하였다. Wing(2006)은 컴퓨팅 사고는 이제 컴퓨터 과학자뿐만 아니라 읽기, 쓰기, 셈하기처럼 현재를 살아가는 모든 사람 사람들이 습득해야 할 기본 능력으로 컴퓨터 과학의 개념을 기반으로 인간의 행동을 이해하고 나아가 일상생활의 문제해결을 설명한다고 하였다[31]. 그에 따르면 크게 알고리즘(algorithm), 추상화(abstraction), 자동화(automation)의 세 가지 요소로 구성되어 있다. 알고리즘은 문제를 해결함에 있어서 순서대로 과정을 나열함을 의미하며, 추상화는 문제해결 방법들을 일반화해서 다른 비슷한 문제에 적용할 수 있도록 하는 것을 의미한다. 마지막으로 자동화는 디지털 도구를 활용하여 반복적인 작업을 효율적으로 처리하는 것을 의미한다.

이후, Computer Science Teachers Association와 International Society for Technology in Education(2011)가 컴퓨팅 사고를 구성하는 9가지 요소로 자료 수집/분석/표현(data collection/ analysis/ data presentation), 문제 분해(problem decomposition), 추상화(abstraction), 알고리즘과 절차(algorithms and procedures) 자동화(automation), 병렬화(paralleization), 시뮬레이션(simulation)을 제시하였다[6]. 자료 수집/분석/표현은 필요한 자료를 수집하여 패턴을 찾은 후 의미 있는 자료들을 차트 등으로 표현하는 과정을 말하며,

문제분해는 복잡한 문제를 이해하기 쉬운 수준으로 분해하고 그 안에서 중요 요소를 추출하는 과정을 의미한다. 병렬화는 빠른 수행을 위해 여러 가지 작업을 동시에 진행하는 것을 말하며, 마지막으로 시뮬레이션은 문제를 해결하기 위한 과정을 미리 수행해 보면서 실제 상황에서 일어날 수 있는 문제를 미연에 방지하는 것이다. 이를 종합해보면, 컴퓨팅 사고는 학생들에게 자료를 수집 및 분석하고, 이를 바탕으로 효과적인 전략을 사용해 복잡한 문제의 해결방안을 개발하는 능력을 길러준다[1].

하지만 현실적으로 학생들이 스스로 컴퓨팅 사고를 향상시키는 것은 쉽지 않다. 효과적인 방법 중 하나는 컴퓨팅 사고를 어떻게 적용할지 교수가 실제 생활의 문제를 기반으로 정확한 가이드라인을 주고 학생들에게 스스로 훈련할 수 있는 기회를 충분히 제공해 주어야 한다[32]. 이러한 이유로 최근 메이커 교육이 컴퓨팅 사고를 위한 효과적인 교육 방법 중 하나로 주목을 받고 있으며, 최근 많은 연구들이 어떻게 학생들의 컴퓨팅 사고력에 미치는지에 관심을 기울이고 있다.

김미영과 김성원(2020)은 스크래치와 비트브릭을 활용한 코딩수업을 통해 컴퓨팅 사고가 수업 전에 비해 유의한 수준으로 향상되었음을 확인하였다[14]. 특히 낮은 수준과 중간 수준의 컴퓨팅 사고력을 가지고 있는 학생에게서 효과적으로 나타났다. 또한 이정민 고은지(2018)의 연구에서는 중학교 1학년 학생들을 대상으로 소프트웨어 교육 후 컴퓨팅 사고의 하위 요인 중 비판적 사고력, 창의성, 알고리즘적 사고, 문제해결력이 유의한 수준으로 향상되었다[17]. 위와 같이 최근 소프트웨어를 활용한 메이커 교육이 학생들의 컴퓨팅 사고를 향상시키는데 효과적인 것으로 밝혀지고 있다.

3. 연구방법

3.1. 연구대상

인공지능 교육에 참여한 고등학생 20명 중 설문을 완료한 19명을 대상으로 데이터 분석을 실시하였다. 참여 학생 중 1학년이 5명(26.3%)이었으며, 2학년은 14명(73.7%)이었다. 성별은 남학생이 16명(84.2%)이었으며, 여학생은 3명(15.8%)이 참여하였다. 본 교육에 참여한 이

유(중복 선택 가능)를 보면, 개인적 호기심에 의해 참여한 학생이 7명(36.8%)이었으며, 본인의 프로그램 실력 향상을 참여한 학생이 3명(15.8%)으로 나타났다. 또한 대학 진학에 도움이 될 것 같아서 참여한 학생이 6명(31.6%)이었으며, 주변의 권유에 의해서 참여한 학생은 5명(26.3%)으로 나타났다.

3.2. 인공지능 메이커 교육 프로그램 설계 및 운영

본 연구는 고등학생들에게 소프트웨어와 인공지능(artificial intelligence, AI)에 대한 관심과 역량을 높이고 다가오는 4차 산업혁명 시대에 필요한 인재를 양성하기 위해 S 대학에서 진행한 “SW 융합 메이커 스쿨”프로젝트의 일환으로 실시되었다. 인공지능 메이커 교육은 <Table 1>와 같이 총 12차시로 구성되어 있으며, 2020년 8월 동안 진행되었다. 코로나로 인해 여유 있는 자리 배치를 위해 20명의 학생들만 수강하도록 하였고, 학생들에게 메이커 스페이스 환경을 제공하기 위해 학교가 아닌 외부의 전문 메이커 교육시설을 사용하였다.

소프트웨어 교육에 대한 태도, 창의적 문제해결력, 컴퓨팅 사고에 대한 자료를 수집하기 위해 교육이 시작하기 전과 교육을 마친 후 사전, 사후 설문을 실시하였고, 마지막 설문이 끝난 후 20명밖에 참여할 수 없는 상황에서 부족한 데이터를 보완하기 위해 학생들에게 인터뷰를 실시하였으며, 수강한 메이커 교육에 대한 전반적인 느낌과 소감 등과 관련해서 질문하였다.

<Table 1> Artificial intelligence maker program

Topic	No.	Detail
Foundation of AI & Machine Learning	1	Introduction to Artificial Intelligence & Machine Learning
	2	Understanding of training models using mBlock
	3	Developing my machine learning
Understanding of Arduino	4	Physical computing using Arduino
	5	Utilizing a sense of Arduino
	6	Utilizing an actuator of Arduino
Arduino & Machine Learning	7	Controlling Arduino using machine learning
	8	
Personal Project	9	Finding a solution to a real-life problem
	10	Realizing a training model with Arduino
	11	Develop a prototype
	12	Product presentation and evaluation

처음 1-3차시는 인공지능과 머신러닝의 기본 개념에 대해 학습하였고, 이러한 인공지능 로봇을 만드는 데에 필요한 도구에 대해 학습하였다. 본 교육에서는 프로그래밍에 사전지식이 많지 않은 학생들도 다루기 쉬운 mBlock을 가지고 진행하였다. 또한 본인들이 생각하는 실생활 로봇에 대해 “로봇 계획서”를 작성하게 함으로써 본인의 아이디어를 공유할 수 있게 하였다. 4-6차시는 아두이노에 대한 기본적인 설명과 중요 기능들에 대해 학습하였다. 특히 본인이 생각하는 기능들을 중요한 센서, 액추에이터 등과 같은 필수 기능들을 활용해 어떻게 구현해볼 수 있는지 직접 실습해보면서 특징을 충분히 학습할 수 있도록 하였다. 7-8차시에는 아두이노가 어떻게 머신러닝과 연결될 수 있는지 개념 및 간단한 원리에 대해서 실습과 함께 학습하였다. 이를 바탕으로 9-12차시에는 본인들이 발견한 실생활 문제를 기반으로 직접 아두이노를 이용해 프로토타입을 만들고 다른 학생들과 공유 및 평가를 하였다.

<Table 2> Understanding of machine learning

Topic	Learning Material
Understanding of Arduino sensors and machine learning	
Telling colors	
Voice recognition for LED	

특히 학생들에게 생소할 수 있는 머신러닝의 개념을 이해시키기 위해 7-8차시 교육을 통해 아두이노가 어떻게 머신러닝 학습을 위해 사용될 수 있는지 실제 실습을 통해 알아보았다. 우선은 머신러닝과 인공지능의 개념 및

관계를 Google의 Teachable machine과 같은 예를 들어 이해하기 쉽게 설명하였다. 그런 후 실제 아두이노를 가지고 간단한 실습을 통해 머신러닝과 인공지능의 기본적인 원리에 대해 이해하는 시간을 갖도록 하였다. 이를 위해 <Table 2>와 같은 실습을 진행하였다.

본 연구의 교육 프로그램이 컴퓨팅 사고의 향상에 있는 점을 고려해볼 때, 교육 프로그램의 주제에 따라 컴퓨팅 사고의 구성요소를 연관시켜 보면 <Table 3>와 같다. 학생들은 각 주제에 따라 컴퓨팅 사고의 구성요소들을 경험하게 되고, 이를 통해 자연스럽게 컴퓨팅 사고를 통한 효과적인 문제해결을 연습할 수 있게 된다.

<Table 3> Computational thinking and program topic

	Computational Thinking								
	Data Collection	Data Analysis	Data Presentation	Problem Decomposition	Abstraction	Algorithm	Automation	Simulation	Parallelization
Foundation of AI & Machine Learning	✓	✓	✓						
Understanding of Arduino	✓	✓	✓	✓	✓	✓	✓	✓	✓
Arduino & Machine Learning	✓	✓	✓	✓	✓	✓	✓	✓	✓
Personal Project	✓	✓	✓	✓	✓	✓	✓	✓	✓

이어서 컴퓨팅 사고의 구성요소별로 구체적인 학습전략은 <Table 4>과 같다. 각각의 컴퓨팅 사고 구성요소에 따라 학습자들에게 요구되는 수행에 기반한 학습목표가 있고, 이를 달성하면서 학생들은 자연스럽게 컴퓨팅 사고를 훈련하게 된다. 예를 들면 학생들은 순서도를 통해 자신의 프로그래밍 순서나 혹은 그림을 통해 자신의 아이디어를 표현하고 남들의 공유를 통해 자료표현(data presentation)을 연습할 수 있다. 또한 학생들은 반복문(loop)의 사용을 통해 자연스럽게 자동화(loop)의 개념에 대해 학습할 수 있게 된다.

<Table 4> Computational thinking and learning objectives

Computational Thinking	Learning objectives
Data Collection	<ul style="list-style-type: none"> • Data collection from a real life • Collecting data related to problems
Data Analysis	<ul style="list-style-type: none"> • Extracting important function • Pattern analysis
Data Presentation	<ul style="list-style-type: none"> • Sharing ideas by drawing them on the blackboard
Problem Decomposition	<ul style="list-style-type: none"> • Elaborating a problem • Decomposing problems into small parts
Abstraction	<ul style="list-style-type: none"> • Deleting unnecessary parts • Grouping common parts
Algorithm	<ul style="list-style-type: none"> • Listing functions in a proper order
Automation	<ul style="list-style-type: none"> • Using Loop (e.g., For, While, etc.) for repetitive jobs
Simulation	<ul style="list-style-type: none"> • Checking out errors
Parallelization	<ul style="list-style-type: none"> • Programming parallel processes

학생들은 본 교육을 통해 스스로 발견한 문제를 바탕으로 본인만의 로봇을 만들었다. (Fig. 1)과 같이 교수자는 기본적인 개념과 개발 방법에 대해서만 가르쳐준 후 학생들이 스스로 직접 경험해볼 수 있게 학습가이드 역할을 하였다. 또한 부족한 부분은 서로 의논하여 새로운 아이디어를 도출하려 하였으며, 결과물에 바로 반영하여 테스트할 수 있도록 하였다. 결과적으로 학생들은 실제로 작동하는 아두이노 인공지능 로봇의 처음 설계부터 완성품 제작까지 전 과정을 경험할 수 있었다.



(Fig.1) Making Activity

3.3. 연구도구

3.3.1. 소프트웨어 교육에 대한 태도

소프트웨어 교육에 대한 일반적인 태도를 조사하기 위해 사전, 사후 검사를 실시하였다. 검사를 위해 사용된 검사지는 Wiebe, Williams, Yang, & Miller(2003)가 컴퓨터과학에 대한 태도를 조사하기 위해 개발한 문항을 박정호(2015)가 수정 및 보완한 도구를 사용하였다[23][30]. 본 연구에서는 연구대상과 맥락에 맞게 문항을 수정 및 보완하여 총 5문항으로 구성하였고, 교육공학 전문가 1인의 검토를 거쳐 사용되었다. 구체적으로 “나는 SW를 잘 배울 수 있다고 확신한다.”와 같은 문항으로 구성되어 있으며, 전혀 그렇지 않다(1점)부터 매우 그렇다(5점)의 5점 Likert 척도로 측정하였으며, 내적 일치도 점수인 Cronbach's α 는 사전 검사에서는 .923, 사후 검사에서는 .897로 나타났다.

3.3.2. 창의적 문제해결력

창의적 문제해결력을 측정하기 위해 한국교육개발원에서 2001년 개발한 ‘간편 창의적 문제해결력 검사 개발 연구(2001)’을 기반으로 서울대 심리 연구실 MI연구팀 2004년에 개발한 검사지를 수정 및 보완하여 사용하였다[4]. 창의적 문제해결 검사지는 총 4개의 하위 영역, 1) 특정 영역의 지식/사고 기능/기술의 이해 및 숙달 여부, 2) 확산적 사고, 3) 비판적 논리적 사고, 4) 동기적 요소로 구성되어 있다. “나는 수업 시간 중 많은 것에 호기심을 가지고 계속 질문한다.”와 같은 문항을 포함하여, 총 20문항으로 구성되어 있으며, 전혀 그렇지 않다(1점)부터 매우 그렇다(5점)의 5점 Likert 척도로 측정하였다. 내적 일치도 점수인 Cronbach's α 는 사전 검사에서는 .934, 사후 검사에서는 .954로 나타났다.

3.3.3. 컴퓨팅 사고

컴퓨팅 사고를 측정하기 위해 본 연구에서는 김미영과 김성원(2018)이 개발한 문제해결 기반 컴퓨팅 사고 측정도구를 수정 및 보완하여 사용하였다[13]. 컴퓨팅 사고 측정문항은 자료표현, 추상화, 자동화, 시뮬레이션

총 4개의 하위 영역을 포함하고 있으며, “나는 문제를 단순화하고 그 관련성을 찾아 표현할 수 있다.”를 포함해 총 18문항으로 구성되어 있다. 모든 문항은 5점 Likert 척도로 측정하게 하였으며, 전혀 그렇지 않다(1점)부터 매우 그렇다(5점) 수준으로 구성되어 있다. 내적 일치도 점수인 Cronbach's α 는 사전 검사에서는 .952, 사후 검사에서는 .974로 나타났다.

3.3.4. 지각된 학업 성취도 및 만족도

지각된 학업 성취도 및 만족도를 조사하기 위한 문항은 강민석(2009)이 개발한 도구에서 추출하여 연구 대상과 맥락에 맞게 수정하여 활용하였다[12]. 학업성취도 검사는 “나는 이 수업을 통해 많은 것을 배웠다고 생각한다.”를 포함해 총 4문항으로 구성되어 있으며, 만족도 검사는 “나는 이 수업을 다른 사람에게 추천하고 싶다.”를 포함해 총 3문제로 구성되어 있다. 두 검사 모두 전혀 그렇지 않다(1점)부터 매우 그렇다(5점)의 5점 Likert 척도로 측정하였다. 지각된 학업성취도와 만족도 도구의 Cronbach's α 는 사후 검사에서 각각 .881과 .866으로 나타났다.

3.4. 분석 방법

데이터 수집을 마친 후 사전, 사후 문항들의 내적 일관성을 살펴보기 위해 Cronbach's α 를 계산하여 확인하였다. 또한 학생들이 지각한 학업성취도와 만족도를 측정하기 위해 사후 설문에만 해당 문항들을 추가하여 조사하였다. 이후 인공지능 메이커 교육의 효과성에 대해 알아보기 위해 각각의 질문에 대해 사전-사후 검사 대응표본 t-검정을 실시하였다. 마지막으로 사전 설문지에서 측정된 소프트웨어 교육에 대한 태도, 창의적 문제해결력, 컴퓨팅 사고가 지각된 학업성취도 및 만족도와 어떠한 관계가 있는지 살펴보기 위해 다중회귀분석을 하였다. 모든 분석은 R studio를 사용하여 수행하였다. 마지막으로 학생들의 인터뷰 녹음파일을 연구자가 분석하여 소프트웨어에 대한 태도에 반영하였다.

4. 연구결과

4.1. 소프트웨어 교육에 대한 태도

소프트웨어 교육에 대한 태도에 대한 변화를 살펴보기 위해 사전, 사후 검사를 바탕으로 대응표본 t-검정을 실시하였다. 그 결과 <Table 5>와 같이 인공지능 메이커 교육 후에 소프트웨어 교육에 대한 태도가 평균점수 4.126에서 4.389로 유의하게 향상되었다. 이러한 긍정적인 태도의 변화는 학생들의 사후 인터뷰에서도 나타났다. 학생들의 주요 내용은 다음과 같다.

“평소에 일단 접하기 힘든 수업인데... 정말 유익하고 재밌었던 시간이었습니다.”
 “소프트웨어 메이커 교육과 로봇에 대해 흥미가 높아졌고... 진로를 결정하는 데에 많은 도움이 되었습니다.”
 “처음으로 로봇기구도 만들어보고 ... 진로에 대해 생각해볼 수 있는 시간이어서 좋았습니다.”
 “로봇을 조립하고 만들면서 그리고 프로그래밍을 짜면서 매우 흥미로운 시간이었고... 앞으로 이런 시간이 많이 활성화 되었으면 좋겠습니다.”

종합해보면 교육 후 학생들은 소프트웨어 메이커 교육에 대해 긍정적인 태도를 보였음을 확인할 수 있었다. 구체적으로 보면, 학생들은 평소에는 접하기 어려운 교육이라 수업으로부터 많은 흥미를 느꼈고, 직접 제작과 프로그래밍 활동을 하면서 긍정적인 느낌을 받았다고 볼 수 있다. 특히 이러한 교육을 통해 학생들은 자신의 진로를 탐색할 수 있는 기회를 가질 수 있었다는 점에서 긍정적인 태도를 보였다.

<Table 5> Paired t-test of attitude toward software education

Variables		M	SD	t	p
Attitude toward SW education	Pre	4.126	0.684	-3.508	0.003
	Post	4.389	0.488		

4.2. 창의적 문제해결력

메이커 교육 후에 창의적 문제해결력의 향상 정도를 살펴보기 위해 사전, 사후 검사를 바탕으로 대응표본 t-검정을 실시하였다. 그 결과 <Table 6>와 같이 메이커 교육 후에 평균 3.755에서 4.009로 창의적 문제해결력이 유의하게 증가하였음을 볼 수 있다.

<Table 6> paired t-test of creative problem solving

Variables		M	SD	t	p
Creative Problem Solving	Pre	3.755	0.570	-3.657	0.002
	Post	4.008	0.583		

4.3. 컴퓨팅 사고

다음으로 컴퓨팅 사고에 대한 인공지능 메이커 교육 전후의 변화를 알아보기 위해 사전, 사후 검사를 바탕으로 대응표본 t-검정을 실시하였다. 그 결과 <Table 7>와 같이 소프트웨어 교육 이후 컴퓨팅 사고가 유의한 수준으로 증가했음을 알 수 있다.

<Table 7> paired t-test of computational thinking

Variables		M	SD	t	p
Computational Thinking	Pre	3.579	.738	-5.383	.000
	Post	4.056	.703		

4.4. 지각된 학업 성취도 및 만족도와와의 관계

마지막으로 교육 초기의 학생들이 가지고 있는 소프트웨어 교육에 대한 태도, 창의적 문제해결력, 컴퓨팅 사고와 본 교육에서 지각하는 학업 성취도 및 만족도에 어떠한 관계가 있는지 살펴보았다. 그 결과 <Table 8>과 같이 교육 초기의 소프트웨어 교육에 대한 태도가 학업성취도를 통계적으로 유의하게 예측하는 것으로 나타났다.

<Table 8> A multiple regression analysis of perceived academic achievement

	B	SE	t
Intercept	1.681	.574	2.927**
Attitude toward SW education	.461	.155	2.972**
Computational Thinking	-.164	.186	-0.883
Creative Problem Solving	.381	.203	1.874
Adjusted R ²	.424		
Model F	8.121***		

p<.01, *p<.001

마찬가지로, <Table 9>과 같이 교육에 대한 지각된 만족도 또한 초반의 소프트웨어 교육에 대한 태도에 의해서만 예측되었다.

<Table 9> A multiple regression analysis of perceived satisfaction

	B	SE	t
Intercept	2.547	.664	3.835***
Attitude toward SW education	.463	.180	2.575*
Computational Thinking	-.129	.215	-.598
Creative Problem Solving	.129	.235	.549
Adjusted R ²		.201	
Model F		3.426*	

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

5. 결론 및 제언

본 연구는 고등학생들을 대상으로 인공지능 메이커 교육이 소프트웨어 교육에 대한 태도, 창의적 문제해결력, 컴퓨팅 사고에 미치는 영향을 살펴보았다. 또한 학생들의 이러한 인지적, 정의적 특성이 학생들이 지각하는 학업 성취도 및 만족도와 어떠한 관계가 있는지 살펴보았다.

먼저, 메이커 교육을 통해 학생들의 소프트웨어 교육에 대한 태도는 유의한 수준으로 향상되었다. 이러한 결과는 메이커 교육 이후 소프트웨어 교육에 대한 태도의 긍정적인 변화를 확인한 기존의 연구들과 그 결을 같이 한다[35]. 우선 학생들의 소프트웨어 교육에 대한 태도가 향상된 이유는 메이커 교육이 기본적으로 학생들에게 자유로운 환경을 제공하여 정보를 탐색 및 분석하며, 결과적으로 스스로 문제를 해결할 수 있도록 지지해 주기 때문이라고 볼 수 있다[9]. 구체적으로 보면 학생들에게 문제를 해결하는 데에 있어서 충분한 자율성(autonomy)과 통제권(control)이 부여된다면, 스스로 문제를 해결해 나가는 과정으로부터 그들은 충분한 성취감과 흥미를 느끼게 된다[19]. 따라서 선생님들은 학생들에게 인공지능 로봇을 만드는 데에 필요한 기본적인 지식과 가이드만을 제공해 줌으로써 독립적인 학습활동을 통해 학생들은 소프트웨어 학습에 많은 흥미와 재미를 느꼈을 것이다.

이제는 소프트웨어의 활용이 거의 모든 분야에 걸쳐 필수적인 시대임을 고려할 때 학생들의 소프트웨어 교육에 대한 긍정적인 태도는 매우 중요하다고 볼 수 있다. 즉 흥미를 비롯한 긍정적인 태도는 학습에 대한 집중력을 높여주고 결과적으로 교육과정에서의 이탈을 방지하여 끝까지 완수할 수 있도록 도와준다는 점에서 중요하다[2]. 이러한 점에서 본 연구는 메이커 교육이 소프트웨어 교육에 대한 태도를 긍정적으로 향상시킬 수 있다는 점을 확인했다는 데에 그 의의가 있다.

다음으로 창의적 문제해결력은 학생들이 교육을 수강한 후 유의하게 증가하였음을 확인할 수 있었다. 프로그래밍 학습과 문제를 해결하는 과정은 문제를 발견하고 그것을 해결하기 위해 다양한 아이디어를 생산하며, 결과적으로 최적의 선택을 통해 완성된 결과물을 얻는다는 점에서 비슷하다[34]. 특히 창의적인 문제해결 과정은 자유로운 환경에서 보다 효과적으로 수행될 수 있고, 메이커 스페이스는 학생들로 하여금 스스로 문제를 창의적으로 탐구할 수 있는 기회를 충분히 제공한다는 점에서 최적의 환경이라고 할 수 있다[18]. 구체적으로 보면 시간과 공간의 제약이 교실에서 이루지는 전통적인 수업방식에 비해 덜하고 상대적으로 문제를 해결하는 데에 필요한 정보나 도구에 대한 접근 또한 상대적으로 더 자유롭다[28]. 그러므로 학생들은 자유로운 사고의 확산과 수렴 과정을 통해 창의적으로 자신만의 문제해결 방법을 구성하고 적용하는 기회를 충분히 가질 수 있다.

또한 주변의 동료들과 협업을 통해 자유롭게 의견을 교환할 때 창의적인 아이디어를 보다 효과적으로 생산할 수 있다[10]. 본 연구에서는 메이커 교육의 본질을 잘 반영하기 위하여 학교가 아닌 외부의 메이커 스페이스를 활용하여 교육을 진행하였다. 코로나 때문에 기존의 계획에 비하면 상대적으로 제약이 있었지만 이곳에서 학생들은 동료들과 자신의 결과를 자유롭게 공유하며 기존의 아이디어를 좀 더 창의적인 아이디어로 발전시키는 기회를 가질 수 있었다.

이어서, 교육을 통해 컴퓨팅 사고가 유의하게 향상되었음을 보았다. 이러한 결과는 메이커 교육과 컴퓨팅 사고와의 관계를 주제로 하는 기존의 많은 연구 결과들에 의해 지지된다[11][14][17] 구체적으로 학생들은 본 교육을 통해서 본인들이 문제를 해결하는 데에 필요한 자료

를 스스로 수집 및 분석하고 비슷한 주제를 기반으로 추상화하는 능력을 배울 수 있었다. 해당 메이커 교육의 교육목표는 전문적인 프로그래밍 기술을 가르치기보다는 아두이노와 앱인벤터와 같은 쉬운 도구를 활용해 컴퓨팅 사고의 향상에 보다 초점을 맞추었다. 때문에 기본적인 코딩을 통해 인공지능 로봇을 제작하며 추상화 및 자동화와 같은 컴퓨팅 사고력을 효과적으로 연습해 볼 수 있는 기회를 가질 수 있었다[17]. 그뿐만 아니라 프로그래밍의 기본 지식을 학습한 후 실생활에서 접할 수 있는 문제를 찾아 해결 방법을 탐색해 보도록 함으로써 학생들이 능동적 학습을 통해 컴퓨팅 사고의 향상을 가능하게 하였고 이렇게 학습된 사고방식은 효과적인 지식전이를 통해 다른 분야에서도 유용하게 활용될 수 있다[25].

마지막으로 교육 초기의 소프트웨어 교육에 대한 태도는 교육에 대한 지각된 학업성취도 및 만족도를 유의하게 예측하고 있음을 확인할 수 있었다. 다시 말하면 청소년들을 위한 성공적인 소프트웨어 교육에 있어서 학생들의 높은 창의적 문제해결력 또는 컴퓨팅 사고 능력보다, 학생들의 소프트웨어 교육에 대한 긍정적인 태도가 더 의미 있는 예측변인으로 나타났다. 즉, 소프트웨어에 관심이 많은 학생일수록 교육에 열심히 참여하고 만족스러운 결과를 얻어 간다고 생각해 볼 수 있다. 그러므로 소프트웨어 교육의 효과를 강화하기 위해서는 학생들이 소프트웨어에 대한 긍정적인 태도를 가질 수 있도록 평소에 소프트웨어의 가치와 필요성에 대해 학습할 수 있는 기회를 제공할 필요가 있다.

본 연구는 인공지능 메이커 교육의 효과에 따른 연구의 의의와 시사점을 제공하였지만, 다음과 같은 연구의 한계점과 그것들을 보완할 추후 연구를 확인할 수 있었다.

첫째, 집단 감염 전염병(코로나)으로 인해 국가적으로 집합교육의 인원 제한이 권고되었고 그에 따라 본 연구에 참여한 학생들의 수가 충분하지 않았다. 따라서 본 연구의 결과를 모든 고등학생들을 대상으로 일반화하기에는 한계가 있다. 그러므로 보다 많은 학생들을 대상으로 하는 후속 연구를 제안한다. 둘째, 학생들의 특성을 측정하는 데에 있어 사용된 데이터가 제한적이다. 본 연구는 학생들이 작성한 설문지를 기반으로 하는 양적 연구를 통해 수행되었다. 하지만 객관적인 학업성취도 검

사도구를 사용한 것이 아니라 설문을 통하여 지각된 학업성취도를 측정하였다. 이는 본 교육의 목적이 일정 수준의 지식이나 점수를 반드시 성취해야 하는 것이 아니라 고등학생들의 소프트웨어에 관한 관심 증진 및 인공지능 개념 및 활용에 대한 탐색이었기 때문에 학생들에게 심리적으로 부담을 줄 수 있는 객관적인 학업성취도를 수행하지 않았다. 하지만 이러한 교육을 통한 좀 더 객관적인 학생들의 변화를 측정하고자 한다면, 추후 연구에서는 지각된 학업성취도가 아닌 학습한 내용을 기반으로 하는 객관적인 학업성취도에 관한 연구를 수행해 볼 수 있을 것이다.

참고문헌

- [1] Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.
- [2] Andersen, L., & Ward, T. J. (2014). Expectancy value models for the STEM persistence plans of ninth grade, high ability students: A comparison between Black, Hispanic, and White students. *Science Education*, 98(2), 216-242.
- [3] Blikstein, P., & Worsley, M. (2016). Children are not hackers. Building a culture of powerful ideas, deep learning and equity in the maker movement. In K. Peppler, E. Halvorsen, & Y. Kafai (Eds.) *Makeology: Makerspaces as learning environments* (pp. 64 - 80). New York, NY: Routledge.
- [4] Cho, S., Jang, Y., Jung, T. & Lim, H. (2001). *A study on the development of simple and creative problem solving Test (1)*. Seoul: Korea Educational Development Institute.
- [5] Choi, S. Y., & Kim, S. M. (2016). Effects of physical computing education using app inventor and Arduino on industrial high school students' Creative and integrative thinking. *The Journal of Korean Association of Computer Education*, 19(6), 45-54.
- [6] Computer Science Teachers Association and the International Society for Technology in Education (2011). *Computational Thinking in K - 12*

Education, teacher resources.

- [7] Dougherty, D. (2012). The maker movement. *Innovations: Technology, governance, globalization*, 7(3), 11-14.
- [8] Halverson, E. R., & Sheridan, K. (2014). The maker movement in education. *Harvard educational review*, 84(4), 495-504.
- [9] Hsu, Y. C., Baldwin, S., & Ching, Y. H. (2017). Learning through making and maker education. *TechTrends*, 61(6), 589-594.
- [10] Jo, Y. J. (2019). Development of maker education program with creative problem solving(CPS). *Journal of Fisheries and Marine Sciences Education*, 31(3), 856-876.
- [11] Jung H. N., & Jhun, Y. S. (2020). Physical Computing Classes using a Flipped Learning Strategy. *Journal of Gifted/Talented Education*, 30(1), 1-23.
- [12] Kang, M. S. (2009). *Development of learners' perceived interaction model and scale between learner and instructor in e-learning environments*. Doctoral Dissertation, Korea University.
- [13] Kim, M. Y., & Kim, S. W. (2018). Korean secondary students' computational thinking based on problem solving. *Journal of Learner-Centered Curriculum and Instruction*, 18(16), 807-830.
- [14] Kim, M. Y., & Kim, S. W. (2020). The effect of scientific problem-solving education using physical computing on computational thinking. *Journal of Learner-Centered Curriculum and Instruction*, 20(8), 387-410.
- [15] Kim, H. J., Seo, J. H., & Kim, Y. S. (2016). The effect of scratch programming education using arduino on middle school students' creative problem solving ability. *Journal of Learner-Centered Curriculum and Instruction*. 16(12), 707-724.
- [16] Lee, S. S., & Lee, Y. N. (2007). Development of blended instructional model for creative problem solving. *Journal of Educational Technology*, 23(2), 135-159.
- [17] Lee, J. M., & Ko, E. J. (2018). The effect of software education on middle school students computational thinking. *Journal of the Korea Contents Association*, 18(12), 238-250.
- [18] Lee, S. C., Kim T. Y., Kim, J. S., Kang, S. J., & Yoon, J. Y. (2019). The Effect of a Design Thinking-based Maker Education Program on the Creative Problem Solving Ability of Elementary School Students. *Journal of The Korean Association of Information Education*, 23(1), 73-84.
- [19] Martin, L. (2015). The promise of the maker movement for education. *Journal of Pre-College Engineering Education Research (J-PEER)*, 5(1), 30-39.
- [20] Mumford, M. D., Supinski, E. P., Threlfall, K. V., & Baughman, W. A. (1996). Process based measures of creative problem solving skills: III. Category selection. *Creativity Research Journal*, 9(4), 395 - 406.
- [21] Osburn, H. K., & Mumford, M. D. (2006). Creativity and planning: Training interventions to develop creative problem-solving skills. *Creativity Research Journal*, 18(2), 173-190.
- [22] Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. New York, NY: Basic Books.
- [23] Park, J. H. (2015). Effects of storytelling based software education on computational thinking. *Journal of the Korean Association of information Education*, 19(1), 57-68.
- [24] Pepler, K., & Bender, S. (2013). Maker movement spreads innovation one project at a time. *Phi Delta Kappan*, 95(3), 22 - 27.
- [25] Ringberg, T., & Reihlen, M. (2008). Towards a socio cognitive approach to knowledge transfer. *Journal of Management Studies*, 45(5), 912-935.
- [26] Seo, M. G. (2012). Critical thinking and creative problem solving. *Korean Journal of General Education*, 6(3), 221-247.
- [27] Shim, S. Y., Kim, J. O., & Kim, J. S. (2016). Development of steam learning program using ar-

duino to improve technological problem-solving ability for middle school students. *The Korean Journal of Technology Education*, 16(1), 77-100.

[28] Taylor, B. (2016). Evaluating the benefit of the maker movement in K-12 STEM education. *Electronic International Journal of Education, Arts, and Science (EIJEAS)*, 2, 1-22.

[29] Treffinger, D. J., Isaksen, S. G., & Dorval, B. K. (2005). *Creative problem solving: An introduction* (4th ed.). Waco, TX: Prufrock Press.

[30] Wiebe, E. N., Williams, L., Yang, K., & Miller, C. (2003). *Computer science attitude survey*. Dept. of Computer Science, NC State University.

[31] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

[32] Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 14(1), 1-16.

[33] Yin, Y., Hadad, R., Tang, X., & Lin, Q. (2019). Improving and Assessing Computational Thinking in Maker Activities: the Integration with Physics and Engineering Learning. *Journal of Science Education and Technology*, 29(2), 189-214

[34] Yoon, J. K., & Kim, Y. S. (2018). Influence of programming education utilizing arduino on creative problem solving ability of high school students. *The SNU Journal of Education Research*, 27(3), 53-73.

[35] Yoon, J. H., Kwon, J. H., & Kang, S. J. (2019). Verification of effectiveness of design thinking-based maker education program for middle school students. *Journal of Learner-Centered Curriculum and Instruction*, 19(10), 561-584.

저자소개

홍 원 준



2008 공주대학교 컴퓨터교육학 학사
 2013 서울대학교 교육공학 석사
 2018 네바다대학 교육공학 박사
 2020~현재 : 선문대학교
 로봇플랫폼연구소 연구원
 관심분야 : 온라인교육, 학습분석학,
 동기, 메타인지
 E-mail : hongkey83@naver.com

최 재 성



2003 경성대학교 컴퓨터공학 학사
 2006 U. of Texas at Arlington
 Computer Engineering 석사
 2011 U. of Texas at Arlington,
 Computer Engineering 박사
 2016~현재 : 선문대학교
 컴퓨터공학부 조교수
 관심분야 : 컴퓨터교육, 사물인터넷,
 센서네트워크
 E-Mail : jschoi@sunmoon.ac.kr

이 현



1998 선문대학교 전자계산학 학사
 2002 선문대학교 대학원 전자계산학
 석사
 2010 U. of Texas at Arlington,
 Computer Engineering 박사
 2012~현재 : 선문대학교
 컴퓨터공학부 부교수
 관심분야: 실시간 의사결정시스템,
 자율컴퓨팅, 휴먼케어 시스템 등
 E-Mail : mahyun91@sunmoon.ac.kr