

# 컴퓨팅 사고력 향상을 위한 초등 피지컬 컴퓨팅 교육과정 개발

김재휘\* · 김동호\*\*

청주교육대학교 교육대학원 로봇전공\* · 청주교육대학교 컴퓨터교육과\*\*

## 요 약

소프트웨어교육의 도구로서 블록형 교육용 프로그래밍 언어(EPL)는 무료로 이용할 수 있고, 피지컬 컴퓨팅은 현실세계와 컴퓨팅 환경을 연결시켜줌으로써 실생활 문제 해결이라는 목표에 더욱 가깝게 다가갈 수 있도록 해주기 때문에, 특히 초등학교 단계에서 학생들이 직관적으로 이해하기 쉬우며 알고리즘적 사고를 향상시키기 때문에 많이 이용되고 있다. 본 연구에서는 초등학생도 회로지식 없이 쉽게 접할 수 있고 가격적인 측면에서도 큰 부담이 가지 않는 엔트리와 센서보드를 활용한 EPL 및 피지컬 컴퓨팅 교육과정을 제시하고, 초등학교 6학년 학생을 대상으로 12차시의 EPL 교육과정과 8차시의 피지컬 컴퓨팅 교육과정을 개발하고 적용하였다. 본 연구의 결과 EPL 교육과정을 적용한 집단과 피지컬 컴퓨팅 교육과정을 추가로 적용한 집단 모두 학생들의 컴퓨팅 사고력 향상에 유의미한 효과가 있었으며, 피지컬 컴퓨팅 교육과정을 추가로 적용한 집단에서 컴퓨팅 사고력 요소의 자료표현, 수행 및 검증, 일반화에서 유의하게 더 큰 향상을 보였다.

키워드 : 소프트웨어교육, 교육용 프로그래밍 언어(EPL), 피지컬 컴퓨팅, 컴퓨팅 사고력, 문제해결력

## Development of Physical Computing Curriculum in Elementary Schools for Computational Thinking

Jaehwi Kim\* · Dongho Kim\*\*

Dept. of Computer Education, Chongju National University in Education

## ABSTRACT

Block-based educational programming language(EPL) is commonly used due to its availability at low or no cost. It is also preferred tool of computing education due to its intuitive design, ease-of-use and its effectiveness in increasing algorithmic thinking abilities especially in elementary students. Physical computing is also necessary because it brings students closer to real-world problem solving by connecting the real world with the computing environment. However, due to high-cost and required knowledge in electrical engineering, many schools find the education difficult to access. The study shows significant increase in computational thinking abilities in both groups treated with EPL and additional physical computing education.

Keywords : Educational Programming Language(EPL), Physical Computing, Computational Thinking, Problem Solving Ability

교신저자 : 김동호(청주교육대학교 컴퓨터교육과, dhkim@cje.ac.kr)

논문투고 : 2016-01-22

논문심사 : 2016-01-24

심사완료 : 2016-02-16

### 1. 서론

최근 미래 사회의 중요한 능력으로 컴퓨팅 사고력 (Computational Thinking)의 중요성이 전 세계적으로 강조되며 이를 중심으로 국가교육과정을 개정하는 사례가 늘고 있다. 영국에서는 만 5세부터 컴퓨팅 교육을 필수로 이수하게 되며, 인도에서는 초등6년 동안 주당 1~2시간씩 소프트웨어 교육을 받는다. 그 외에도 미국, 일본, 이스라엘, 에스토니아 등 여러 나라에서 앞 다퉈서 소프트웨어 교육을 의무화하는 추세다. 한국에서도 이러한 움직임의 일환으로 소프트웨어 교육을 초등학교 및 중학교 전체 학생을 대상으로 의무화하는 방안이 발표되었다[12].

소프트웨어교육의 목표는 컴퓨팅 사고력을 가진 창의·융합 인재이다. 컴퓨팅 사고력이란 컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력을 뜻한다. 이러한 컴퓨팅 사고력을 기르기 위해서는 기존의 프로그래밍 교육처럼 예제를 그대로 따라하며 기능을 익히는 것이 아니라, 일상생활 주변의 문제를 발견하고 이를 해결하기 위한 알고리즘을 고안하고 프로그래밍을 통해 구현하고 디버깅하는 과정이 필요하다. 이러한 관점에서 기존의 프로그래머를 양성하기 위한 직업교육적인 측면에서 벗어나, 일반 학생들의 사고력을 향상을 위해 교육용 프로그래밍 언어(EPL)를 활용한 다양한 교육 프로그램이 개발 및 적용되어왔다. 교육용 프로그래밍언어는 쉽고 직관적이면서 전문 언어와 유사한 알고리즘적 사고과정을 경험할 수 있기 때문에 초등학교 저학년에서부터 중등 이상의 교육에까지 다양하게 활용되고 있다. 그러나 소프트웨어교육의 목표인 컴퓨팅 사고력을 기르는 데 있어 교육용 프로그래밍언어 교육만으로는 한계가 있다. 학생들의 교육을 받으며 만들어내는 결과물이 화면 속에만 머무르기 때문에 배운 것을 실생활과 관련짓기 어려워하는 경우가 많기 때문이다. 학생들이 학습한 것을 바탕으로 실생활의 문제를 해결할 수 있는 아이디어를 떠올리고 컴퓨팅 교육에 대한 지속적인 흥미를 이끌어내기 위해서는 컴퓨팅 활동이 현실세계와 연결될 수 있다는 사실을 경험할 수 있도록 하는 것이 중요하다[19]. 피지컬 컴퓨팅의 핵심은 컴퓨팅 환경과 현실세계 사이의 상호작용에 있다. 다양한 센서를 통해 현실세계로부터

터 받아들이는 정보를 이용하여 컴퓨팅을 하거나 컴퓨팅의 결과를 다양한 형태의 액추에이터(대표적으로 모터)를 통해 출력하는 것이다. 피지컬 컴퓨팅은 이를 경험한 학생들이 컴퓨팅이 우리 생활과 밀접하게 관련되어 있음을 느낌으로써 생활 속 문제를 해결하는 창의적인 아이디어를 얻거나 실제로 해결하는데 도움을 준다.

본 연구에서는 교육용 프로그래밍 언어 교육과정과 함께 센서보드를 활용한 피지컬 컴퓨팅 교육과정을 개발하고, 해당 교육과정을 초등학교 6학년 두 학급에 차등하여 적용한 후 그 효과성을 검증하고자 한다. 한 학급은 교육용 프로그래밍 언어 교육과정만을 적용하고, 다른 학급은 같은 교육 이후 피지컬 컴퓨팅 교육과정을 추가로 적용한 후 컴퓨팅 사고력과 관계된 문제해결력 검사를 실시하고 그 결과를 비교·분석한다.

### 2. 이론적 배경 및 시사점

#### 2.1 문제해결력과 컴퓨팅 사고력

문제해결력이란 일상생활에서 부딪히는 여러 가지 문제를 창의적, 논리적, 비판적 사고를 통해 올바르게 인식하고 적절히 해결하는 능력을 말한다. Wing(2006)은 컴퓨팅 사고력에 대한 논의를 본격화하면서 컴퓨팅 사고력의 개념에 대한 전반적인 틀을 설정했다. 그는 (그림 1)과 같이 컴퓨팅 사고력을 크게 추상화(abstraction)와 자동화(automation)로 구분하였다.



(Fig. 1) Components of Computational Thinking

추상화는 실제 세계의 문제를 해결 가능한 형태로 표현하기 위한 사고과정이라고 할 수 있으며, 문제를 해결하기 위하여 필요한 자료를 수집 및 분석하고, 필요한 표현 방법(도표, 그래프 등)을 활용하여 눈으로 보기 쉽게 나타내고, 복잡한 요소를 작은 단위로 분해하고, 해결에 필요한 변수들을 추출하여 적절한 해결 모델을 설계

하는 과정이다. 이러한 추상화 과정에서 만들어진 해결 모델을 컴퓨터가 이해할 수 있는 프로그래밍 언어로 표현하여 인간이 처리하기 어려운 많은 양의 반복된 작업이나 시뮬레이션을 실시하는 것을 자동화라고 한다[19].

미국의 ISTE(International Society for Technology in Education)와 CSTA(Computer Science Teachers Association)는 2011년 컴퓨팅 사고력을 자료 수집, 자료 분석, 자료 표현, 문제 분해, 추상화, 알고리즘과 절차, 자동화, 시뮬레이션, 병렬화의 아홉 가지의 세부요소로 나누어 제시하였다[19].

교육부(2015.2)에서는 이러한 앞선 연구들을 바탕으로 컴퓨팅 사고력을 ‘컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력’으로 정의하고, ISTE와 CSTA의 요소를 인용하여 그 구성요소를 아래와 같이 제시하였다[19].

- 문제를 컴퓨터로 해결할 수 있는 형태로 구조화하기
- 자료를 분석하고 논리적으로 조직하기
- 모델링이나 시뮬레이션 등의 추상화를 통해 자료를 표현하기
- 알고리즘적 사고를 통하여 해결방법을 자동화하기
- 효율적인 해결방법을 수행하고 검증하기
- 문제 해결 과정을 다른 문제에 적용하고 일반화하기

## 2.2 교육용 프로그래밍 언어(EPL)

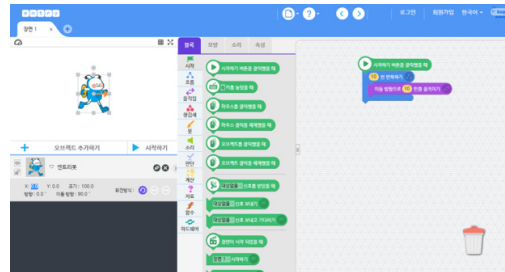
교육용 프로그래밍 언어(Educational Programming Language, EPL)란 전문 프로그래머들이 사용하는 것이 아니라, 학생들의 고등사고력(논리적 사고력, 문제해결력, 창의성)을 길러주기 위해 교육용으로 만들어진 프로그래밍 언어를 말한다. 교육용 프로그래밍 언어의 종류는 매우 다양하며 블록형 프로그래밍 언어인 ‘스크래치(Scratch)[11]’, ‘엔트리(Entry)[7]’, ‘앨리스(Alice)’, 텍스트형 프로그래밍 언어인 ‘로고(Logo)’ 등이 있다. ‘엔트리(Entry)’는 스크래치와 같은 블록 코딩 방식의 언어를 기반으로 하여, 교사들이 쉽게 학생들을 교육하고 관리할 수 있도록 만들어진 ‘소프트웨어 교육 플랫폼’이다. 기존에는 화면에 텍스트를 입력하여 프로그램을 만들었다면 교육용 프로그래밍 언어는 블록을 드래그&드롭하는 형식으로 프로그래밍할 수 있다. 이는 아이들의 흥미와 몰입도가 텍스트 언어보다 높으며, 쉽게 프로그래

밍의 개념을 익힐 수 있는 장점이 있다. 특히 엔트리는 다음과 같은 장점을 지닌다.

첫째, 문제 해결 형식의 다양한 학습 콘텐츠를 제공한다. 미로를 빠져나가거나 우주를 여행하며 만나는 문제들을 게임을 하듯 해결해 나가며 자연스럽게 프로그래밍의 기본 원리를 익힐 뿐 아니라 블록형 언어의 사용에 익숙해지게 된다.

둘째, 교사가 직접 학생들에게 필요한 강의나 코스를 제공할 수 있다. 엔트리에서 제공하는 강의 기능은 교사가 수업을 준비할 때, 캐릭터나 배경, 일부 코드가 주어진 상태에서부터 수업을 시작할 수 있게 해주거나 학생들에게 해당 차시에 필요한 블록들만을 제한하여 보여줌으로써 수업을 효율적으로 진행할 수 있도록 돕는다.

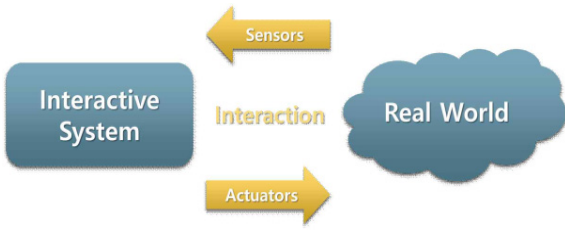
셋째, 다양한 교육 자료를 무료로 제공한다. EBS 소프트웨어야 놀자와 함께 제작한 방송영상과 특별영상을 교육 또는 학습 자료로 활용할 수 있다.



(Fig. 2) Screen of Making Entry

## 2.3 피지컬 컴퓨팅

피지컬 컴퓨팅의 개념은 덴 오설리번(Dan O'Sullivan)[6]과 탐 아이고(Tom Igoe)[22] 교수가 NYUIITP(Interactive Telecommunications Program, New York University)에서 인터랙티브 피지컬 시스템(interactive physical systems)를 가르치는데서 시작되었다. 그는 피지컬 컴퓨팅을 “피지컬한 실제 세계와 컴퓨터의 가상 세계가 서로 대화할 수 있도록 하는 것이다”라고 설명하고 있다. 즉, 현실세계의 여러 현상들을 센서나 여러 장치들을 통해 감지하고, 감지된 값들을 사용하여 컴퓨터를 통해서 물리적인 장치를 제어하는 것을 말한다.



(Fig. 3) Concept of Physical Computing

피지컬 컴퓨팅 도구는 형태에 따라 로봇형, 모듈형, 보드형으로 나눌 수 있다. 먼저 로봇형은 모터 등의 물리적 출력장치가 강화된 완성된 형태의 도구를 말한다. 프로그래밍의 결과로 로봇을 움직이거나 소리나게 할 수 있으며, 다양한 센서를 포함하고 있는 경우 이를 활용하여 로봇이 라인을 따라가게 하거나 장애물이 있으면 피하게 하는 등 현실세계와 상호작용하는 컴퓨팅을 할 수 있다. 로봇형 피지컬 컴퓨팅 중 블록형 교육용 프로그래밍 언어로 제어 가능한 도구의 예로는 햄스터와 알버트가 있다. 로봇형 도구는 완제품 형태이기 때문에 쉽게 사용할 수 있다는 장점이 있지만, 가격이 비싸고 활용에 한계가 있다는 단점이 있다. 모듈형은 다양한 입력장치와 출력장치를 조립하여 마이크로 컨트롤러에 연결하고 교육용 프로그래밍 언어를 통해 제어할 수 있는 도구를 말한다. 레고 형태로 주어진 블록들만을 가지고 조립을 할 수도 있고 종이, 철사 등 다른 재료들을 통해 원하는 형태를 만들어낼 수도 있다. 그 예로는 레고 위두(Lego WeDo)와 비트브릭(BitBrick) 등이 있다. 원하는 대로 조립하고 때로는 다른 재료들을 함께 쓸 수 있기 때문에 활용 범위가 넓지만, 가격이 비싸고 제품 간 호환이 어렵다는 단점이 있다.

마지막으로 보드형은 마이크로 컨트롤러를 포함한 전자보드 형태의 기판을 말한다. 아두이노는 대표적인 오픈 하드웨어 보드이다. 다양한 형태 중 아두이노 UNO 보드는, 마이크로컨트롤러를 내장하고 6개의 아날로그 입력 핀과 13개의 디지털 입출력 핀을 제공하며 아두이노 프로그램 외에도 다양한 교육용 프로그래밍 언어로 각 핀들의 제어가 가능하다. 아두이노는 필요한 전자소자나 확장 쉴드를 활용하여 상상하는 것을 제한 없이 자유롭게 만들어낼 수 있으나, 이를 위해서는 전기회로 및 전자지식이 필요하기 때문에 초등 교육현장

에서 활용되기에는 어려운 점이 많다. 이에 피지컬 컴퓨팅의 초기 단계에서는 센서보드를 많이 활용한다. 센서보드란, 다양한 입력장치와 출력장치를 집약된 보드 형태로 만들고 마이크로컨트롤러를 통해 컴퓨터와 실시간으로 통신하며 하여 회로의 구성없이 컴퓨팅을 할 수 있도록 만들어주는 도구이다. 블록형 교육용 프로그래밍 언어인 스크래치나 엔트리를 통해 제어할 수 있는 센서보드는 대표적으로 스크래치용 피코보드와 엔트리용 센서보드가 있다.

### 2.4 선행 연구 및 시사점

Clement(1989)는 사고력과 관련하여 수차례의 연구를 통해 로고 프로그래밍이 어떤 영향을 끼치는가를 조사하여 로고가 사고력 신장(문제해결력, 창의력)과 수학 지식 습득에 영향을 미친다는 연구보고를 하였다.

이점순(2008)은 진주시내 초등학교 5학년 2개 학급을 대상으로 로고 프로그래밍 교육을 실시하고 로고가 아동들의 창의력 신장에 미치는 영향을 보았다. 또한, 김종진(2010)은 교육용 프로그래밍 언어의 종류에 따른 창의성 향상도를 비교하기 위해 로고와 스크래치의 교재를 개발하고 창의성 향상도를 비교 실험한 결과, 로고는 창의성 영역 중 유창성 영역에, 스크래치는 추상성과 지향 영역에 긍정적인 영향을 주는 것으로 나타났다.

조준필(2012)은 기술교육에서 중학생의 논리적 사고력 함양을 위한 학습 프로그램을 개발하고 적용한 결과 스크래치 학습 프로그램이 논리적 사고력 함양에 효과가 있으며, 특히 명제논리에 있어 큰 향상을 보였다. 양영훈(2010)은 초등학교 4학년 학생을 대상으로 스크래치 활용 알고리즘 수업을 실시한 결과 실시하고 비교집단과 비교한 결과 논리적 사고력 향상에 효과가 있었으며, 특히 조합논리에 유의미한 효과를 보이는 것으로 나타났다.

신갑천(2010)은 스크래치를 활용하여 로봇의 라인트레이싱을 시뮬레이션하는 교육 프로그램을 개발, 적용한 결과 학생들의 논리적 사고력과 문제 해결력이 향상되었음을 확인하였다. 또한, 송정범(2008)은 스크래치를 활용한 프로그래밍 학습이 문제해결력 향상에 미치는 영향을 국제비교조사(PISA)의 문제해결력 문항을 통해 연구한 결과 학습자의 내재적 동기와 문제해결력 향상

에 효과가 있는 것으로 나타났다.

김수환과 한선관(2012)은 초등학교 4학년 20명을 대상으로 5일간 스크래치 수업을 실시 후, 컴퓨팅 사고력의 컴퓨터적 관점 영역의 인식이 제대로 형성되었는지 알아보기 위해 문항지를 기반으로 하는 자기 설문 평가를 실시하였다. 이은경은 스크래치를 활용한 창의적 프로그래밍 학습을 설계하여 중학교 1학년 학생들 34명의 창의성 잠재력 및 컴퓨팅 사고력 능력 발달에 미치는 영향을 창의성 학습평가 연구소(<http://www.clec.co.kr>)에서 개발한 검사 도구를 사용하여 평가하였다. 최형신(2012)은 교육대학교 학생들의 스크래치 수업 학기프로젝트 산출물에서 다양한 측면의 컴퓨팅 사고력 능력의 세부요소를 보이는 증거들을 기초, 발달, 능숙의 3단계의 수준으로 평가할 수 있는 루브릭을 고안 제안하였다.

이스라엘 정부는 중학교 1학년 교육 과정에 컴퓨팅 사고력을 강화시키기 위해 새로이 도입된 커리큘럼인 ‘스크래치를 이용한 컴퓨터 과학 개론’ 과목을 평가하기 위해 5,194명의 학생들을 대상으로 전국적인 스크래치 시험을 실시했다. 최형신과 김기범(2015)은 예비교사들에게 스크래치 프로그래밍 교육을 실시하고, 학생들의 컴퓨팅 사고력을 측정하기 위해 컴퓨팅 사고력 평가 설문, 스크래치 지필 시험, 산출물 기반 인터뷰를 실시하였다. 컴퓨팅 사고력 평가 설문지는 Brennan, K., & Resnick, M. (2012)이 제시한 컴퓨팅 사고 프레임워크의 세 가지 차원에 기반을 두고 14문항을 개발하였다.

이상의 선행 연구에서는 컴퓨팅 교육의 도구로서 대부분 교육용 프로그래밍 언어만을 다루고 있으며, 피지컬 컴퓨팅의 효과를 분석한 연구는 부족한 것으로 확인되었다. 또한 컴퓨팅 사고력의 평가에 대한 선행 연구들을 살펴본 결과 컴퓨팅 사고력의 정의와 프레임워크에 기반하여 다양한 시도들이 이루어지고 있으나 관련 연구가 충분하지 않아 검증된 평가 방법을 찾기 어려웠다. 대부분의 경우 설문 형식으로 이루어지는 자기평가의 방법이나 프로그래밍 언어에 대한 지필평가, 또는 산출물 평가의 방식으로 이루어지고 있었다. 이러한 평가 방법의 경우, 컴퓨팅 교육을 받지 않은 학생들과 받은 학생들 간의 차이를 검증할 수 없어, 교육의 필요성에 대한 근거를 제공하기 어렵다. 컴퓨팅 사고력에 대한 정의와 요소가 논의되기 전까지 컴퓨팅 교육의 효과성 검증은 창의성, 논리적사고력, 문제해결력 검사 등으

로 이루어져 왔는데, 이러한 일반적인 고등 사고력들과 컴퓨팅 사고력과의 연관성을 찾아 객관적인 평가 방법을 마련하는 것이 필요하다.

### 3. 교육과정 및 효과성 경증을 위한 검사지 개발

#### 3.1 EPL 및 피지컬 컴퓨팅 교육과정 개발

##### 3.1.1 교육 도구 선정

본 연구에서는 교육용 프로그래밍 언어 교육과 피지컬 컴퓨팅 교육의 효과성을 분석하기 위한 도구로 블록형 교육용 프로그래밍 언어인 엔트리와 엔트리와 연결하여 제어할 수 있는 피지컬 컴퓨팅 도구인 센서보드를 선정하였다.

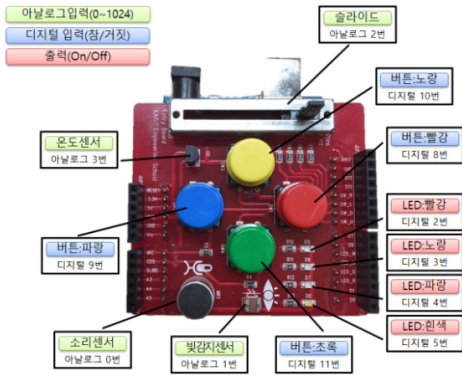
엔트리는 기본적으로 스크래치와 유사하면서도 EBS 교육방송에서 방영된 ‘소프트웨어야 놀자’와 연계하여 영상자료를 수업에 활용할 수 있기 때문에 선택하였다.



(Fig. 4) EBS-Play SW Entry Webpage

피지컬 컴퓨팅 도구로는 엔트리와 연결하여 사용할 수 있는 다양한 피지컬 컴퓨팅 도구 중에서 센서보드를 선정하였다. 센서보드는 저렴하면서도 별도의 회로 지식이 필요 없고 조작이 간편하여 초등학교 수준에서 적용하기에 적합하다고 판단하였기 때문이다. 카이스트 공

학교에서 만든 엔트리 센서보드는 아두이노 보드를 기반으로 쉴드 형태로 끼워 사용할 수 있는 형태로, (그림 5)에서는 센서보드의 기능 요소를 보여준다.



(Fig. 5) Entry Sensor-board & Elements

3.1.2 교수·학습 모형 설계

소프트웨어 교육의 목표인 컴퓨팅 사고력 향상을 위해 정형화된 수업모형은 아직 합의된 것이 없으므로, 본 연구에서는 교육부의 소프트웨어 교육 운영 지침 (2015.2)의 초등학교의 운영 지침의 ‘알고리즘과 프로그래밍’ 영역의 내용 요소와 성취기준에 근거하여 교수·학습 모형을 설계하고자 하였다. 교육부 지침의 ‘알고리즘과 프로그래밍’ 영역의 중영역과 내용요소에서 초등학교생이 소프트웨어교육을 통해 경험해야 하는 내용 요소를 재구성하여 문제 중심 소프트웨어교육 교수·학습 모형을 <표 1>과 같이 제시하였다.

<Table 1> Problem Based SW Education Learning

Stage	Activity
Problem presentation	present problem related to SW in our life
Problem solving research	structure problem and research solution method procedurally
Algorithm representation	represent solution with sequence, selection, repetition structure
Programing experience	Automate problem solution with restricted instruction
Share & Evaluation	share with colleague and evaluate program

3.1.3 문제 중심 EPL 교육과정 개발

컴퓨팅 사고력 향상을 위한 교육용 프로그래밍 언어 교육과정을 문제 중심으로 구성하기로 하고 ‘EBS 소프트웨어야 놀자’의 교육과정을 바탕으로 8가지 문제를 선정하였다. 문제 해결을 위해 학생들이 사용하는 프로그래밍 개념 요소는 교육부의 성취기준에 따라 ‘순차, 선택, 반복구조’로 제한하였으며 필요한 경우 기본적인 수준에서 이벤트, 변수, 연산의 개념을 추가하기로 하였다. 먼저, 선정된 8가지 문제를 해결하며 배울 수 있는 프로그래밍 개념을 분석하고, 분석 결과에 따라 8가지 주제를 개념의 위계와 난이도에 따라 순서를 재구성하였다. 소프트웨어 교육이 효과적으로 이루어지기 위해서는 학생들이 교육의 필요성과 소프트웨어와 우리 생활의 관련성을 인식하는 것이 중요하다. 또한 다양한 문제 중심 학습을 통해 배운 프로그래밍 개념들을 적용하여 스스로 문제를 설정하고 계획하고 해결해보는 경험도 필요하다. 이러한 부분들을 포함하여 구성된 문제 중심 교육용 프로그래밍 언어 교육과정 12차시의 구성과 활동 내용은 <표 2>와 같다.

<Table 2> Problem Based EPL Curriculum & Activity

class	contents
<b>우리 주변의 소프트웨어</b>	
1	- 소프트웨어의 의미를 알고 우리 주변의 소프트웨어 찾아보기 - 소프트웨어로 변화하고 있는 세상 알아보기(생활, 경제, 산업) - 소프트웨어와 나의 미래와의 관련성 찾기
<b>미로를 따라보물찾기</b>	
2	- 교육용 프로그래밍언어(블록형)에 익숙해지기 - SW의 기본 원리(순차, 반복, 선택)를 활용하여 엔트리봇이 미로를 빠져나가는 미션 해결하기
<b>나만의 애니메이션 만들기</b>	
3	- 여러 가지 이벤트에 따른 애니메이션 계획하기 - 여러 가지 이벤트와 순차구조를 활용하여 애니메이션 만들기 (시작하기, 마우스클릭, 키보드 키 입력, 신호)
<b>도형을 그려요</b>	
4	- 여러 가지 다각형의 성질 알아보기 - 반복 명령어로 여러 가지 다각형 그리기(~회 반복)

<b>쑹쑹쑹쑹로봇청소기</b>	
5	<ul style="list-style-type: none"> <li>- 로봇청소기를 관찰하고 문제해결방법 찾아보기</li> <li>- 주어진 명령어로 활용하여 로봇청소기 알고리즘 만들기</li> <li>- 반복과 선택구조를 활용하여 로봇청소기 프로그램 구현하기 (계속 반복하기, 만약 ~라면)</li> </ul>
<b>열려라자동문</b>	
6	<ul style="list-style-type: none"> <li>- 자동문을 관찰하고 자동문 역할극 하기</li> <li>- 자동문 알고리즘을 표현하기 위한 오브젝트를 구성하고 오브젝트간 상호작용을 고려하여 주어진 명령어로 알고리즘 만들기</li> <li>- 선택구조와 신호를 활용하여 자동문 프로그램 구현하기 (만약 ~라면, ~신호보내기, ~신호를 받았을 때)</li> </ul>
<b>비밀번호로상자를열어라</b>	
7	<ul style="list-style-type: none"> <li>- 전자도어락에 비밀번호를 입력하여 문을 여는 과정 생각해보기</li> <li>- 주어진 명령어를 사용하여 전자도어락 알고리즘 적기</li> <li>- 변수와 선택구조를 활용하여 전자도어락 프로그램 만들기 (변수를 ~로 정하기, 만약 ~라면 아니면~)</li> </ul>
<b>숫자맞히기놀이</b>	
8	<ul style="list-style-type: none"> <li>- 업앤다운 숫자 맞히기 놀이하기</li> <li>- 주어진 명령어로 숫자맞히기 놀이 알고리즘 적기</li> <li>- 변수와 선택구조, 논리연산을 활용하여 숫자 맞히기 놀이 프로그램 만들기</li> </ul>
<b>누르면나온다!자동판매기</b>	
9	<ul style="list-style-type: none"> <li>- 자동판매기 역할극 하기</li> <li>- 자동판매기 작동 절차 적기</li> <li>- 자동판매기 프로그램 만들기</li> </ul>
<b>자유창작활동</b>	
10-11	<ul style="list-style-type: none"> <li>- 내가 만들고 싶은 프로그램 아이디어 내기</li> <li>- 프로그램 화면 구상하기</li> <li>- 프로그램에 필요한 오브젝트를 파악하고 프로그램에서의 역할 적기</li> <li>- 계획에 따라 배운 개념을 활용하여 프로그램 만들기</li> </ul>
<b>프로그램공유및발전</b>	
12	<ul style="list-style-type: none"> <li>- 내가 만든 프로그램을 친구들과 공유하기</li> <li>- 친구들의 프로그램에 의견 주기(잘된 점, 개선점)</li> <li>- 친구들의 의견을 반영하여 프로그램 발전시키기</li> </ul>

교육부 지침을 바탕으로 재구성한 수업의 흐름에 문제 중심 교육과정의 내용을 적용한 구체적인 수업 지도안은 <표 3>과 같다. 1~2차시와 10~12차시를 제외한 나머지 차시에 대한 수업의 흐름도 이와 유사하다.

<Table 3> Lesson Plan of Problem Based EPL Curriculum

stage	Flow	teaching · learning Activity
도입	생활 속 문제 제시	<ul style="list-style-type: none"> <li>◎ 로봇청소기 문제 제시</li> <li>• 로봇청소기 관찰하기</li> <li>• 로봇청소기의 움직임 분석하기</li> </ul>
전개	문제 해결 방법 탐색	<ul style="list-style-type: none"> <li>◎ 로봇청소기 문제 해결방법 탐색</li> <li>• 학습지를 통해 방을 효과적으로 청소하기 위한 경로 그려보기</li> <li>• 로봇청소기가 반복적으로 수행해야 하는 명령 적어보기</li> <li>• 로봇청소기가 조건에 따라 수행해야 하는 명령 적어보기</li> </ul>
	알고리즘 표현	<ul style="list-style-type: none"> <li>◎ 로봇청소기 알고리즘 표현</li> <li>• ‘벽에 닿았는가?’의 판단에 따른 로봇청소기의 알고리즘 순서도 그리기</li> </ul>
정리	프로그래밍의 체험	<ul style="list-style-type: none"> <li>◎ 로봇청소기 프로그램 만들기</li> <li>• 반복, 선택의 개념을 활용하여 로봇청소기 프로그램 만들기</li> </ul>
	공유 및 평가	<ul style="list-style-type: none"> <li>◎ 프로그램 공유 및 평가</li> <li>• 만든 프로그램을 친구들과 공유하기</li> <li>• 친구들의 프로그램과 나의 프로그램의 다른 점을 찾거나, 친구의 프로그램에 의견 주기</li> <li>• 친구들의 의견을 반영하여 나의 프로그램을 발전시킬 수 있는 아이디어 찾기</li> </ul>

### 3.1.4 센서보드를 활용한 피지컬 컴퓨팅 교육과정 개발

12차시의 문제 중심 EPL 교육과정을 통해 소프트웨어의 기본 원리를 익힌 학생을 대상으로 추가로 실시할 피지컬 컴퓨팅 교육과정 8차시를 개발하였다. 교육과정의 구성에 있어 센서보드 교육을 받는 학생들은 이미 12차시 교육을 통해 기본적인 프로그래밍 원리를 이해하고 있다고 가정하고, 센서보드를 통해 피지컬 컴퓨팅을 활용하는 방법을 아는 것 외에 추가적인 프로그래밍 개념의 학습을 하지는 않도록 하는 데 유의하였다.

피지컬 컴퓨팅 교육과정도 문제 중심으로 이루어질 수 있도록 센서와 관련된 실생활 문제를 뽑아냈다. 1차시에서는 우리 주변에서 센서를 활용하는 예들을 통해 센서를 이해할 수 있도록 하고, 2~6차시에서는 센서보드의 각 센서들을 활용하여 다양한 예시작품을 만들어 봄으로써 센서의 활용 방법을 익힐 수 있도록 구성하였다. 센서보드에게 제공하고 있는 기능들은 크게 아날로

그센서, 디지털입력, 출력으로 나누어볼 수 있는데, 모든 기능을 다 다루되, 아날로그 센서에 더 초점을 두고, 프로그램이 현실세계와 어떻게 상호작용 할 수 있을지 아이디어를 낼 수 있는 방향으로 내용에 중점을 두었다. 7-8차시에서는 센서보드를 활용하여 자유로운 창작 활동을 하도록 함으로써 이전 차시들에서 냈던 아이디어를 실제로 구현해 보고, 친구들과 공유하며 발전시켜나가도록 했다.

<Table 4> Physical Computing Curriculum using Sensor-board

class	field	contents
<b>우리의 감각기관과 센서</b>		
1	센서의 이해	<ul style="list-style-type: none"> <li>- 우리의 감각기관을 대신하는 센서들에 대해 생각하기</li> <li>- 우리 주변에서 쓰이는 센서들 찾아보기</li> <li>- 센서보드의 부품들 알아보기</li> </ul>
<b>소리센서로 풍선을 터뜨려요</b>		
2	아날로그 센서 입력 활용 (소리센서)	<ul style="list-style-type: none"> <li>- 소리센서의 사용법 알아보기</li> <li>- 소리센서를 붙여 풍선을 터뜨리는 프로그램 만들기</li> <li>- 소리센서를 활용할 수 있는 재미있는 아이디어 적기</li> </ul>
<b>빛감지 센서로도 난방지기를 만들어요</b>		
3	아날로그 센서 입력 활용 (빛감지 센서)	<ul style="list-style-type: none"> <li>- 빛 감지 센서의 사용법 알아보기</li> <li>- 빛 감지 센서를 사용하여 난방지기 프로그램 만들기</li> <li>- 빛 감지 센서를 활용할 수 있는 재미있는 아이디어 적기</li> </ul>
<b>온도센서로 자동으로 온도를 조절해요</b>		
4	아날로그 센서 입력 활용 (온도센서)	<ul style="list-style-type: none"> <li>- 온도 센서의 사용법 알아보기</li> <li>- 온도 센서를 사용하여 자동으로 히터나 에어컨을 켜는 프로그램 만들기</li> <li>- 온도센서를 활용할 수 있는 재미있는 아이디어 적기</li> </ul>
<b>슬라이더로 날아라 독수리 게임을 완성해요</b>		
5	아날로그 센서 입력 활용 (슬라이더)	<ul style="list-style-type: none"> <li>- 슬라이더 사용법 알아보기</li> <li>- 슬라이더로 날아라 독수리 게임 만들기</li> <li>- 슬라이더를 활용할 수 있는 재미있는 아이디어 적기</li> </ul>
<b>LED와 버튼으로 미로를 탈출해요</b>		
6	디지털 입출력 활용 (버튼, LED)	<ul style="list-style-type: none"> <li>- LED와 버튼의 사용법 알아보기</li> <li>- LED와 버튼으로 미로탈출 게임 만들기</li> <li>- LED와 버튼을 활용할 수 있는 재미있는 아이디어 적기</li> </ul>

**센서보드를 활용한 나만의 작품 만들기**

**7-8 자유창작 활동**

- 미래의 스마트 하우스, 스마트 장난감 등 자유로운 주제로 센서보드를 활용한 작품 계획하기
- 센서보드를 활용한 작품 만들기
- 제대로 동작하지 않는 부분을 친구와 협력하여 해결하기
- 만든 작품 친구들과 공유하기

센서보드를 활용한 피지컬 컴퓨팅 수업 지도안의 예시는 <표 5>과 같으며, 1차시와 7~8차시를 제외한 나머지 차시에 대한 수업의 흐름도 이와 유사하다.

<Table 5> Lesson Plan using Sensor-board

stage	Flow	teaching · learning Activity
도입	생활 속 문제 제시	<ul style="list-style-type: none"> <li>◎ 소음측정기 문제 제시</li> <li>• 소음측정기 관찰하기</li> <li>• 소음측정기의 원리 생각해보기</li> </ul>
	문제 해결 방법 탐색	<ul style="list-style-type: none"> <li>◎ 소리센서의 사용법 알기</li> <li>• 오브젝트가 소리센서값을 말하게 하기</li> <li>• 주변 소음에 따른 소리센서값의 변화 살펴보기</li> <li>• 소리센서값을 활용하여 풍선을 터뜨리는 아이디어 내기</li> </ul>
전개	알고리즘 표현	<ul style="list-style-type: none"> <li>◎ 소리센서로 풍선 터뜨리기 알고리즘 표현</li> <li>• 소리센서값의 변화에 따라 풍선이 커지는 알고리즘을 일상 언어로 표현하기</li> <li>• 풍선의 크기에 따라 풍선이 터진 모양으로 바뀌는 알고리즘을 일상 언어로 표현하기</li> </ul>
	프로그램의 명칭 체험	<ul style="list-style-type: none"> <li>◎ 풍선 터뜨리기 프로그램 만들기</li> <li>• 소리센서값과 반복, 선택의 개념을 활용하여 풍선 터뜨리기 프로그램 만들기</li> </ul>
정리	공유 및 평가	<ul style="list-style-type: none"> <li>◎ 프로그램 공유 및 평가</li> <li>• 만든 프로그램을 친구들과 공유하기</li> <li>• 친구들의 프로그램과 나의 프로그램의 다른 점을 찾거나, 친구의 프로그램에 의견 주기</li> <li>• 친구들의 의견을 반영하여 나의 프로그램을 발전시킬 수 있는 아이디어 찾기</li> </ul>

**3.2 교육과정 효과성 검증을 위한 검사지 개발**

**3.2.1 연구 대상 선정**

서울 소재 B 초등학교 6학년 전체 학생에게 개발된



교육과정을 적용하기로 하였다. 단, 학생들의 부담을 줄이고 실험집단과 비교집단의 사례수를 맞추기 위해 전체 학급 중 두 학급만을 대상으로 효과성 검증을하기로 하였다. 먼저, B 학교 6학년 학급 담임을 대상으로 개발한 12차시 교육과정 적용을 위한 연수를 실시하면서 교육 방법에 있어 동일하도록 주의를 요하고 구체적인 교육자료 및 교재를 함께 전달하였다. 이후 각 학급 담임들의 주도 하에 6학년 전체 학급을 대상으로 개발된 EPL 12차시의 교육과정을 통해 주 1차시씩 12차시의 교육을 실시하였다. 전체 학급을 대상으로 한 공통 교육과정 적용 이후, 미리 선정했던 한 학급을 대상으로 방과 후 4일간 8차시의 피지컬 컴퓨팅 교육을 추가로 실시하였다. 그 구체적인 내용은 <표 6>과 같다.

<Table 6> Sample of Experimental Groups

group	experience group	comparative group	total
class	6	6	
curriculum	Problem based EPL 12 class, physical computing	Problem based EPL 12 class	
	8 class		
total class	20	12	
student number	21	23	44

3.2.2 검사 도구 선정 및 검사지 개발

EPL 교육과정만 적용한 학생과 피지컬 컴퓨팅 교육과정을 추가로 적용한 학생이 스스로 느끼는 효과성을 비교·분석하기 위해 문제 해결, 학습 만족도, 진로·직업 관련의 세 영역으로 나누어 두 가지씩 설문 문항을 구성하였다. 그 내용은 <표 7>과 같다.

<Table 7> General Questionnaire

Num.	field	question
1	problem solution	엔트리(와 센서보드)를 활용한 프로그래밍 수업이 우리 주변의 소프트웨어를 이해하는 데 도움이 된다.
2	problem solution	엔트리(와 센서보드)를 활용한 프로그래밍 수업이 일상생활의 속의 문제 해결 아이디어를 떠올리는 데 도움이 된다.
3	learning satisfaction	엔트리(와 센서보드)로 더 많은 작품들을 만들어 보고 싶다.

4	learning satisfaction	친구에게 엔트리(와 센서보드)를 배우도록 추천하고 싶다.
5	career	소프트웨어 교육이 나의 미래에 도움이 된다고 생각한다.
6	career	소프트웨어 관련 직업에 관심을 가지게 되었다.

학생들의 컴퓨팅 사고력을 측정하기 위하여 OECD/PISA 국제학업성취도평가의 문제해결력 영역 문항 2003년도 19문항과 2012년도 6문항의 총 25문항을 바탕으로 컴퓨팅 사고력과 밀접한 관련을 가진 문항들을 추출하였다. 컴퓨팅 사고력의 개념은 교육부의 운영 지침에 따라 제시된 컴퓨팅 사고력의 요소에 따라 6가지 요소로 구분하고 편의성을 위해 <표 8>과 같이 각 요소를 약식으로 표현하였다. 문제해결력 검사 문항 25개 중, 컴퓨팅 사고력 요소와 관련하여 추출한 문항 17가지와 관련 컴퓨팅 사고력 요소는 <표 9>와 같다.

<Table 8> Computational Thinking Elements

Computational Thinking Elements	abbreviation
structure problem with computer	→ structuring
analyze data and organize logically	→ data analysis
represent data through abstraction with modeling or simulation	→ data representation
automate solution method with algorithmic thinking	→ algorithm & automation
execute efficient solution method and verify	→ execution & verification
generalize and apply problem solving process for other problem	→ generalization

<Table 9> Problem Solving Contents on Computational Thinking

Num.	subject	content	related computational thinking
1	교통	최적의 미팅 장소 찾기	execution & verification
2	교통	가장 빠른 길 찾기	execution & verification
3	날씨 컨트롤	각각의 컨트롤러가 영향을 미치는 요소 찾기	data analysis
4	티켓	티켓 구매 머신으로 필요한 티켓 구매하기	algorithm & automation
5	티켓	티켓 구매 머신으로 가장 저렴한 티켓 구매하기	algorithm & automation
6	로봇 청소기	로봇청소기의 움직임을 관찰하고 절차적으로 나타내기	data analysis

7	도서관 관리체계	다양한 조건들을 고려하여 가능한 책 대여일수 구하기	algorithm & automation
8	도서관 관리체계	다양한 조건들을 예/아니오로 판단할 수 있는 순서도로 나타내기	structuring
9	수에 의한 디자인	다양한 명령어에 의한 결과를 보고 새로이 주어진 명령어에 대한 결과 예측하기	generalizati on
10	수에 의한 디자인	다양한 명령어에 의한 결과를 보고 새로이 주어진 명령어에 대한 결과 예측하기	generalizati on
11	수에 의한 디자인	예시자료를 보고 반복명령을 활용하여 주어진 그림을 그리는 명령어 쓰기	structuring
12	교육과정 설계	주어진 조건을 만족하도록 시간표 완성하기	execution & verification
13	냉동고	수행 절차를 살펴보고 수행 결과의 원인 찾기	data analysis
14	냉동고	문제의 원인 찾아 오류를 수정하고 결과 검증하기	execution & verification
15	관계	수로에서 주어진 수문의 개폐에 따른 물의 흐름 나타내기	data representati on
16	관계	수로에서 다양한 문제 상황을 가정하고 물이 통과하여 흐를 수 있는지 검증하기	execution & verification
17	관계	특정 수문이 고장 났는지를 점검할 수 있는 실험 설계하기	generalizati on

3.2.3 실험 설계 및 처치

본 연구에서는 실험집단과 비교집단을 임의로 선정하여 실시하는 이질 비교집단 전후검사 설계를 사용하였다. 연구대상으로 실험집단과 비교집단을 선정하여 사전검사를 통해 실험집단과 비교집단이 동질집단임을 보이고, 실험처치 후 사후검사로 일반설문과 컴퓨팅 사고력 관련 문제해결력 검사를 실시하여 실험의 효과성 여부를 분석했다. 이러한 연구의 실험설계를 도식화 하면 <표 10>와 같다.

<Table 10> Experimental Design

experience group	O <sub>1</sub>	X <sub>1</sub>	X <sub>2</sub>	O <sub>2</sub> , O <sub>3</sub>
comparative group	O <sub>1</sub>	X <sub>1</sub>		O <sub>2</sub> , O <sub>3</sub>

- O<sub>1</sub> : 사전검사(컴퓨팅 사고력 검사)
- O<sub>2</sub> : 사후검사(일반 설문)
- O<sub>3</sub> : 사후검사(컴퓨팅 사고력 관련 문제해결력 검사)
- X<sub>1</sub> : 문제 중심 EPL 교육과정(12차시)
- X<sub>2</sub> : 센서보드 활용 피지컬 컴퓨팅 교육과정(8차시)

4. 교육과정 적용 결과 및 분석

4.1 일반 설문 결과 분석

실험 집단과 비교집단을 대상으로 개발한 두 가지 교육과정을 차등 적용 후 문제해결, 학습 만족도, 진로·직업 관련의 세 영역에 관한 일반 설문을 실시하였다. 각 영역의 최저점은 2점, 최고점은 10점이었다. 실험집단과 비교집단에 유의미한 차이가 있는지 보기 위해 독립표본 t검증을 실시한 결과는 <표 11>과 같다. 설문의 모든 영역에서 실험집단의 평균이 비교집단보다 높았으나, 통계적으로 유의미한 차이를 보이는 영역은 문제해결과 진로 직업 관련으로 나타났다. 피지컬 컴퓨팅 교육과정이 학생들에게 실생활 문제를 해결하는 아이디어를 내는 데 도움을 주고, 배운 내용을 자신의 진로와 연계시키도록 만드는데 효과적이었음을 보여준다.

<Table 11> Result of General Questionnaire

related area	group	n	M	SD	t	P
problem solution	experience	21	9.0952	1.0911	-2.9163	**0.0057
	comparative	23	7.4348	2.3899		
learning satisfaction	experience	21	8.4286	1.3990	-1.6952	0.0974
	comparative	23	7.3913	2.4632		
career	experience	21	8.3532	1.2890	-2.2988	*0.0266
	comparative	23	6.9565	2.4951		

\*P<.05, \*\*P<.01

4.2 컴퓨팅 사고력 관련 문제 해결력 검사 결과 분석

4.2.1 사전검사

실험집단과 비교집단을 대상으로 컴퓨팅 사고력 관련 문제 해결력 검사지로 사전 검사를 실시한 결과, 실험집단은 비교집단에 비해 컴퓨팅 사고력의 3개의 하위 요소 자료 분석, 수행 및 검증, 일반화에서 평균수가 높고, 그 외 3개의 하위 요소에서는 낮게 나타났으나 통계적으로 유의한 차이는 없었다. 따라서 각 항목에 해서 유의한 차이가 없으므로 두 집단이 동질집단임이 확인되었다.

### 4.2.2 집단별 사전-사후검사 결과 비교

실험집단과 비교집단을 대상으로 각각의 처치 적용 후, 실험 처치 효과성을 분석하기 위해 사전-사후검사 데이터로 종속표본 t검정을 실시하였다. 그 결과는 <표 12>, <표 13>과 같다.

<Table 12> Validation of Computational Thinking on Comparative Group(n=21)

element	test period	M	SD	t	P
structuring	pretest	1.1304	0.7570	-3.1096	**0.0051
	posttest	2.2174	1.6776		
data analysis	pretest	4.8261	1.0725	-2.3361	*0.0290
	posttest	5.5217	0.9941		
data representation	pretest	1.1304	0.7570	-0.1958	0.8466
	posttest	1.1739	0.8341		
algorithm & automation	pretest	2.0000	0.8528	-2.3133	*0.0304
	posttest	2.5217	0.5108		
execution & verification	pretest	3.5217	0.8980	-2.1786	*0.0404
	posttest	4.3913	1.5591		
generalization	pretest	1.1739	1.0292	-2.1817	*0.0401
	posttest	1.8261	0.9841		

\*P<.05, \*\*P<.01

<Table 13> Validation of Computational Thinking on Experimental Group(n=21)

element	test period	M	SD	t	P
structuring	pretest	1.0952	0.6249	-3.8730	**0.0009
	posttest	2.3810	1.5645		
data analysis	pretest	4.8571	0.7928	-2.3070	*0.0319
	posttest	5.6190	1.2836		
data representation	pretest	0.9524	0.7400	-3.5675	**0.0019
	posttest	1.6190	0.4976		
algorithm & automation	pretest	1.8095	0.7496	-2.5856	*0.0177
	posttest	2.3333	0.4830		
execution & verification	pretest	4.0000	1.2247	-3.9241	**0.0008
	posttest	5.2381	0.8891		
generalization	pretest	1.2857	0.9024	-3.8602	**0.0010
	posttest	2.3810	0.6690		

\*P<.05, \*\*P<.01

비교 집단의 컴퓨팅 사고력 사전·사후 검사를 분석한 결과 모든 컴퓨팅 사고력의 모든 요소에서 평균이 상승하였으며, t 검증의 결과, 자료 표현을 제외한 모든 요소에서 p<.05로 통계적으로 유의한 것으로 확인되었다. 실험 집단의 데이터를 분석한 결과 모든 컴퓨팅 사고력의 모든 요소에서 평균이 상승하였으며, t 검증의 결과 컴퓨팅 사고력의 모든 요소에서 p<.05로 통계적으

로 유의한 것으로 확인되었고, 특히 구조화, 자료 표현, 수행 및 검증, 일반화 요소에서 p<.01로 큰 상승폭을 보였다. 이는 두 집단의 실험처치가 모두 컴퓨팅 사고력의 향상에 유의미한 긍정적인 효과를 보이는 것으로 볼 수 있다.

### 4.2.3 집단 간 컴퓨팅 사고력 사후검사 차이 검증

실험처치 후 실험집단과 비교집단의 논리 사고력 각각의 하위요소들의 변화가 두 집단 간 통계적으로 유의한 차이를 보이는지 알아보기 위해 독립표본 t검정을 실시하였다. 그 결과는 <표 14>와 같다.

실험집단과 비교집단의 사후 결과를 비교·분석한 결과, 알고리즘과 자동화를 제외한 모든 영역에서 실험 집단의 평균값이 더 높게 나타났다. 이 결과가 유의한 지 살펴보는 t 검증에서 자료표현, 수행 및 검증, 일반화 영역에서 P<.05로 유의한 차이를 보이는 것으로 나타났다.

실험 결과, 센서보드를 활용한 피지컬 컴퓨팅 수업은 EPL만을 활용한 교육에 비해 컴퓨팅 사고력 요소 중 자료표현, 수행 및 검증, 일반화 능력 향상에 더 효과적이었다고 볼 수 있다.

<Table 14> Result of Computational Thinking(n=44)

element	group	n	M	SD	t	P
structuring	experience	21	2.3810	1.5645	-0.3335	0.7404
	comparative	23	2.2174	1.6776		
data analysis	experience	21	5.6190	1.2836	-0.2825	0.7789
	comparative	23	5.5217	0.9941		
data representation	experience	21	1.6190	0.4976	-2.1236	*0.0396
	comparative	23	1.1739	0.8341		
algorithm & automation	experience	21	2.3333	0.4830	1.2541	0.2167
	comparative	23	2.5217	0.5108		
execution & verification	experience	21	5.2381	0.8891	-2.1843	*0.0346
	comparative	23	4.3913	1.5591		
generalization	experience	21	2.3810	0.6690	-2.1660	*0.0360
	comparative	23	1.8261	0.9841		

\*P<.05, \*\*P<.01

### 4.3 종합적 분석 결과

이상의 적용 결과에 대하여 종합적으로 정리하면 다음과 같다.

- 일반 설문을 통한 교육의 만족도 조사에서는 학습 만족도는 유사했으나 실험 집단이 문제해결과 진로 직업의 영역에서 더 큰 효과를 느끼는 것으로 나타났다.
- 실험 집단과 비교 집단 모두 학생들의 컴퓨팅 사고력 향상에 유의미한 긍정적인 효과가 있었다.
- 실험집단은 비교집단에 비해 자료표현, 수행 및 검증, 일반화 영역에서  $P < .05$ 로 유의한 차이를 보이는 것으로 나타났다. 이는 센서보드를 활용한 피지컬 컴퓨팅 수업은 EPL만을 활용한 교육에 비해 컴퓨팅 사고력 요소 중 자료표현, 수행 및 검증, 일반화 능력 향상에 더 효과적이었다고 볼 수 있다.
- 이를 종합적으로 정리해보면 문제 중심 EPL 교육과정과 센서보드를 활용한 피지컬 컴퓨팅 교육과정이 학습자의 컴퓨팅 사고력 관련 문제해결력을 신장시키는데 기여한다고 볼 수 있으며, 프로그래밍 교육만 받을 때보다 피지컬 컴퓨팅 교육이 함께 이루어질 때 만족도와 컴퓨팅 사고력 관련 문제해결력의 향상 측면에서 더 효과적이라고 할 수 있다.

## 5. 결론

본 연구에서는 엔트리와 센서보드를 활용한 문제 중심의 EPL 교육과정과 센서보드를 활용한 피지컬 컴퓨팅 교육과정을 개발하고 적용하였다.

먼저 문헌 연구를 통하여 컴퓨팅 사고력의 개념과 요소, 교육부의 소프트웨어교육 운영 지침을 바탕으로 문제 중심의 소프트웨어교육을 위한 교육 모형을 제시하였다. 이후 소프트웨어 교육 성취 기준에 근거하여 문제 중심 소프트웨어 교육에 활용할 수 있는 주제들을 선정하고 제시한 모형을 적용한 12차시의 EPL 교육과정을 개발하고 구체적인 수업 지도안의 예를 제시하였다. 또한 배운 개념을 적용할 수 있는 피지컬 컴퓨팅 교육과정 8차시를 개발하고 마찬가지로 수업 지도안의 예를 제시하였다.

개발된 교육과정 적용의 효과를 알아보기 위해 기존에 문제해결력 측정을 위해 사용되어오던 OECD/PISA 국제학업성취도 평가의 문제해결력 영역 문항을 컴퓨팅 사고력의 요소와 관련된 문항 중심으로 재구성하고 영역을 나눔으로써 컴퓨팅 사고를 측정할 수 있는 검사로

활용하였다. 이를 통해 EPL 교육과정만 적용한 집단과 피지컬 컴퓨팅 교육과정을 추가로 적용한 집단의 결과를 비교 분석하였다. 또한 일반 설문을 통해 각 교육과정을 적용한 집단의 교육 만족도도 함께 분석하였다.

연구 결과에 따르면 문제 중심 EPL 교육과정을 적용한 집단과 센서보드를 활용한 피지컬 컴퓨팅 교육과정을 추가로 적용한 집단 모두 학생들의 컴퓨팅 사고력 향상에 유의미한 효과가 있었다. 단, 피지컬 컴퓨팅 교육과정을 추가로 적용한 집단에서 컴퓨팅 사고력 요소의 자료표현, 수행 및 검증, 일반화에서 유의하게 더 큰 향상을 보였다. 또한 일반 설문을 통해 교육의 만족도를 조사하여 비교 분석한 결과, 학습 만족도는 유사했으나 문제해결과 진로 직업의 영역에서 더 큰 효과를 느끼는 것으로 나타났다.

이를 종합으로 고려하면 문제 중심 EPL 교육과정과 센서보드를 활용한 피지컬 컴퓨팅 교육과정이 학습자의 컴퓨팅 사고력 관련 문제해결력을 신장시키는데 기여한다고 볼 수 있으며, 프로그래밍 교육만 받을 때보다 피지컬 컴퓨팅 교육이 함께 이루어질 때 만족도와 컴퓨팅 사고력 관련 문제해결력의 향상 측면에서 더 효과적이라고 할 수 있다.

## 참고문헌

- [1] Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the annual 232 American Educational Research Association (AERA) meeting, Vancouver, BC, Canada.
- [2] Cho Junpil (2012). The Development of Learning Program Using Scratch to Foster Logical Thinking Ability of Middle School Students in Technology Education. *Journal of Korean Technology Education Association*, 12(1), 2012.4, 213-233.
- [3] Choi Hyungsin, & Kim Gibum (2015). The Effects of Scratch Programming on Preservice Teachers : Assessment Utilizing Computational Thinking and Bloom's Taxonomy. *Journal of The Korean Association of Information Education*, 19(2), June 2015, 225-232.

[4] Choi, H. (2014). Developing Lessons and Rubrics to Promote Computational Thinking. *Journal of the Korean Association of Information Education*, 18(1), 57-64.

[5] D. H. Clements (1989). Learning of geometric concepts in a LOGO environment. *Journal for Research in Mathematics Education*

[6] Dan O'Sullivan. <http://itp.nyu.edu/~dbo3/blog/>

[7] GNUE Institute of future Talent (2015.8). 2015 Creative computing issue report.

[8] <http://hamster.school>

[9] <http://neweducation.co.kr/entry/>

[10] <http://play-entry.org/tt#!/advance/software>

[11] <http://scratch.edu/>

[12] [http://www.hani.co.kr/arti/economy/economy\\_general/648219.html](http://www.hani.co.kr/arti/economy/economy_general/648219.html)

[13] <http://www.play-entry.org/>

[14] <http://www.slideshare.net/KisoonEom/ss-50151290>

[15] Kim Jongjin (2010). Developing the Teaching Material and Comparative Experiment of LOGO and Scratch. *Journal of The Korea Digital Contents Society*, 10(7), 459-469.

[16] Kim, S., & Han, S. (2012). Design-Based Learning for Computational Thinking. *Journal of the Korean Association of Information Education*, 16(3), 319-326.

[17] Lee Gemsun (2008). Effects of LOGO Programming Language on Elementary School Students' Creativity, JNUE Master Thesis.

[18] Lee, E. (2013). Creative Programming Learning with Scratch for Enhancing Computational Thinking. *Journal of Korean Association of Computer Education*, 16(1), 1-9.

[19] Ministry of Education (2015. 2). Guide of SW education.

[20] S. Jungbum etc (2008). The Effect of Learning Scratch Programming on Students' Motivation and Problem Solving Ability. *Journal of the Korean Association of Information Education*, 2(3), 323-332.

[21] Shin Gabchun (2010). Effects of Robot Programming Education using Scratch on Growth of Logical Thinking Abilities and Problem Solving Abilities, GNUE Master Thesis.

[22] Tom Igoe. <http://tigue.net/>

[23] Yang Younghun (2011). Study on Development and Application of Algorithm Learning Program for Thinking Ability Improvement Using Scratch, Jeju University Master Thesis.

[24] Zur-Bargury, I., Parv, B., & Lanzberg, D. (2013). A National Exam as a Tool for Improving a New Curriculum. Proceedings of ITiCSE Conference, 267-272.

#### 저자소개



##### 김재휘

2012 청주교육대학교 졸업  
 2014 청주교육대학교 교육대학원 졸업  
 2012~2015 대전광역시 초등학교사로 근무  
 2015~현재 엔트리교육연구소 연구원  
 관심분야: 소프트웨어교육, 로봇교육  
 e-mail: jaewhi23@gmail.com



##### 김동호

1999 서울대학교 전산학과(이학박사)  
 1990~현재 청주교육대학교 교수  
 관심분야: 컴퓨터교육, 로봇교육  
 e-mail: dhkim@cje.ac.kr

