

# 알고리즘적 사고 중심 프로그래밍 활동을 위한 교육용 텐저블 코딩 도구 개발

심재권<sup>†</sup> · 권대용<sup>††</sup>

## 요 약

4차 산업혁명에 대응하기 위한 인재양성을 목적으로 초등학교에서부터 소프트웨어 교육을 필수적으로 수행하고 있다. 소프트웨어 교육은 알고리즘적 사고의 향상을 위해 컴퓨팅을 기반으로 문제해결 절차를 설계하고, 코딩하여 결과를 확인하는 프로그래밍 활동을 제공하고 있다. 일반적으로 프로그래밍 활동에서는 언플러그드 활동을 통해 알고리즘을 설계하고, 이후 코딩하는 단계로 진행된다. 하지만 학생은 알고리즘 설계 활동과 코딩 활동이 중복되는 것처럼 느낄 수 있으며 코딩에 사용되는 도구의 사용법에 따라 설계한 내용과 표현이 다를 수 있어 인지적 부담이 발생할 수 있다. 본 연구에서는 언플러그드 활동에서 설계한 알고리즘을 직관적으로 소스 코드로 변환하여 실행시킬 수 있는 교육용 텐저블 코딩 도구를 제안하고자 한다. 제안하는 도구는 종이 블록을 조합하는 언플러그드 활동을 통해 알고리즘을 설계하고, 사진을 찍어 코드로 변환하여 실행된 결과를 확인할 수 있도록 개발하였고 초등학교 학생을 대상으로 사용성을 검증하였다.

주제어 : 알고리즘적 사고, 텐저블 프로그래밍 도구, 프로그래밍 교육

## Development of an Educational Tangible Coding Tools for Algorithmic Thinking Focused on Programming Activities

Jaekwoun Shim<sup>†</sup> · Daiyoung Kwon<sup>††</sup>

## ABSTRACT

Software education is required from elementary schools to prepare students for the fourth industrial revolution, which aims to improve algorithmic thinking. In general, teaching is divided into two stages: using a flowchart to design algorithms and implementing them through programming. However, converting a flowchart into code and checking the results in an educational programming tool is time consuming and requires additional programming activities. This study proposes a tangible coding tool that enables elementary students to convert algorithms designed at the unplugged activity into educational programming tool codes. This tool was developed in order for students to design algorithms at the level of assembling paper blocks and input them into a programming tool by taking a picture. Sixth graders were participated in this activity to evaluate its usability.

**Keywords** : Algorithmic Thinking, Tangible Programming Tool, Programming Education

---

<sup>†</sup>정 회 원: 고려대학교 영재교육원 연구교수  
<sup>††</sup>총신회원: 고려대학교 교과교육연구소 연구교수(교신저자)  
논문접수: 2019년 8월 8일, 심사완료: 2019년 11월 6일, 게재확정: 2019년 11월 19일  
\* 본 논문은 한국연구재단의 지원으로 수행되었음(NRF-2017R1D1A1B03036378)

## 1. 서론

미래사회로의 진행을 의미하는 4차 산업혁명이 등장한 이후 인공지능, 빅데이터, 가상과 증강현실 등 컴퓨팅 환경에서의 인간의 삶을 다양성과 진보적으로 변화시킬 수 있는 인재를 양성하는 방안에 관한 관심이 증대되고 있다[1]. 주요국의 인재양성 방안을 살펴보면, 대학과 대학원을 중심으로 고급 IT 인력을 양성하는 방안과 더불어 초등학교에서부터 초기에 ICT 교육을 필수적으로 제공하는 등 인재양성과 관련된 다양한 노력을 기울이고 있다[2][3]. 우리나라도 대학을 중심으로 IT 인력양성에 중점을 두고 있을 뿐 아니라 2015 개정 교육과정을 통해 초등학교와 중학교에서도 컴퓨팅 환경에서의 창의적 문제해결력을 갖추기 위한 소프트웨어 교육을 필수적으로 수행하고 있다[4][5].

초등학교와 중학교의 소프트웨어 교육을 살펴보면, 실생활과 다양한 학문 분야에 컴퓨팅을 활용하여 창의적으로 문제를 해결하는 능력의 향상과 정보문화에 관한 소양을 갖추고 협력적으로 문제를 해결하는 태도를 함양하는 데 목적을 두고 있다[4]. 특히, 컴퓨팅 파워를 사용하는 문제해결력을 기르기 위해 컴퓨터과학에 기반하여 문제해결 방법을 설계하고 소프트웨어로 구현하는 프로그래밍 교육을 중점적으로 수행하고 있다. 초등학생 대상 소프트웨어 교육의 목적을 달성하기 위해 학생의 수준에 적합한 스크래치, 엔트리와 같은 교육용 프로그래밍 도구와 함께 학생의 흥미와 관심을 유발하는 다양한 문제를 교육 콘텐츠로 제공하고 있다[6]. 소프트웨어 교육에 관한 연구와 관심이 증가하면서 다양한 교육 콘텐츠가 개발되어 적용된 결과, 프로그래밍 활동에 대한 내적 동기가 긍정적으로 변화하고, 논리적 사고력이 향상되는 등 교육적인 효과가 있는 것으로 보고되고 있다[7][8].

컴퓨팅 환경에서 문제를 해결하기 위한 문제해결 절차를 고안하고 프로그래밍하는 과정은 알고리즘적 사고와 사용하는 프로그래밍 도구의 능숙함을 동시에 요구하고 있어 학습에 부담이 발생하게 된다[9]. 또한, 프로그래밍 활동에서 학생은 문법에 대한 생소함, 의도와 결과 간의 오차, 오류의 수정 등 프로그래밍 학습에 다양한 어려움을 경험하게 된다[10][11]. 소프트웨어 교육에서도 일반적으로

순서도 수준의 도구를 사용하여 알고리즘을 설계하는 활동을 먼저 수행하고, 이후 프로그래밍 활동을 수행하는 형태로 학습의 단계를 제시하고 있다[12]. 즉, 문제를 해결하는 과정을 언플러그드 수준에서 점차적으로 표현하는 활동과 설계한 알고리즘에 따라 교육용 프로그래밍 도구를 사용하여 코딩하는 활동을 통해 체계적으로 알고리즘적 사고 중심의 프로그래밍 활동을 경험할 수 있다[13][14].

초등학생의 수준에서 알고리즘 설계 활동에 주로 사용되는 언플러그드 활동은 순서도를 활용하여 문제를 해결하는 과정을 표현하는 것이다[15]. 학생은 순서도로 문제를 해결하는 절차를 작성한 이후, 결과를 확인하기 위해 교육용 프로그래밍 도구에 알고리즘에 따라 코드를 완성하는 두 가지 활동을 일반적으로 수행한다. 교육의 목적상 알고리즘 설계의 단계와 코딩 활동 단계를 구분하여 제공하고 있지만, 학생의 관점에서 보면 설계한 알고리즘을 즉각적으로 확인할 수 없기 때문에 코딩한 결과를 시청각적으로 쉽게 확인할 수 있는 교육용 프로그래밍 활동에 더 큰 관심을 두게 된다[16]. 이로 인해, 알고리즘 설계 단계에 집중하기보다는 교육용 프로그래밍 도구를 사용하는 방법과 시청각적인 결과에 집중하게 된다. 따라서 컴퓨팅 환경에서 문제를 해결하기 위한 절차적 사고의 학습에 집중하기 위해서는 학생이 알고리즘을 설계하는 단계와 코딩 활동 단계가 분리되어 있다고 느끼지 않게 할 필요가 있다. 즉, 알고리즘 설계와 코딩하는 과정을 자연스럽게 하나의 과정으로 연계하여 확인할 수 있는 교육용 프로그래밍 도구가 필요하다.

본 연구에서는 학생이 언플러그드 활동으로 설계한 알고리즘을 직관적인 과정을 통해 코드로 변환하여 실행시킬 수 있는 텐저블 프로그래밍 도구를 제안하고자 한다. 제안하는 도구는 학생이 종이에 인쇄된 블록을 손으로 조합하여 순서도를 작성하는 수준에서 알고리즘을 설계하고, 종이 블록을 사진 찍어 즉각적으로 블록형 프로그래밍 도구에 소스 코드로 변환할 수 있도록 개발하였다. 학생은 제안하는 도구를 사용하여 설계한 알고리즘의 결과를 즉시적으로 확인할 수 있고, 디버깅할 수 있다. 제안하는 도구는 초등학교 6학년을 대상으로 적용하여 사용성을 검증한 결과, 알고리즘 중심의 프로그래밍 활동에 활용 가능한 것으로 분석되었다.

## 2. 관련연구

### 2.1 알고리즘 설계 중심 학습방안

교육용 프로그래밍 도구를 사용하여 프로그래밍 활동을 수행하기 이전에 문제해결 알고리즘을 설계하는 활동을 먼저 제공하는 알고리즘 사고 중심의 학습방안에 관한 연구를 살펴보면 다음과 같다.

초등학교 저학년을 대상으로 알고리즘 표현 도구인 순서도를 활용하여 놀이기반의 학습을 통해 순차, 선택, 반복의 절차적 사고를 가르칠 수 있는지 학습의 가능성을 분석하였다[13]. 분석 결과, 놀이 중심으로 학습한 학생의 성취도가 컴퓨터를 사용하여 학습한 학생들에 비해 높은 것으로 분석되었다.

알고리즘적 사고 능력의 향상을 목적으로 알고리즘적 사고 문제 모델을 제안하고, 초등학교 저학년의 수준을 고려하여 문제 모델에 기반한 학습 활동지를 제안하였다[9]. 학습 활동지는 출발지에서 목적지까지 로봇을 이동시키기 위한 문제를 제시하고, 순차, 분기, 반복의 알고리즘 표현 유형에 따른 명령어를 사용하여 시작과 종료의 논리적인 연결을 수행할 수 있도록 제작되었다. 동일한 문제를 제안하는 활동지로 풀이한 집단과 스크래치로 풀이한 집단을 비교한 결과, 제안하는 활동지를 이용한 집단이 풀이 시간이 더 적게 걸리고 문제 풀이 성공률도 더 높은 것으로 분석되었다.

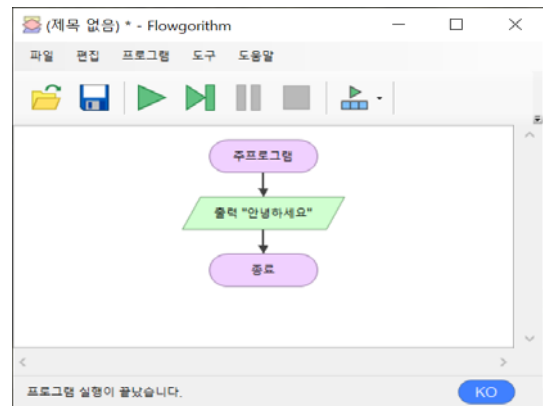
이전의 연구를 토대로 볼 때, 알고리즘을 설계하는 활동을 먼저 수행한 이후 교육용 프로그래밍 도구를 사용하여 프로그래밍하는 활동 단계는 타당하다고 할 수 있으며, 학생의 수준을 고려한 알고리즘 설계 중심의 학습방안에 관한 지속적인 연구가 필요하다.

### 2.2 알고리즘 설계 중심 도구

초등학생을 위한 소프트웨어 교육에서 알고리즘 표현 방법은 자연어로 작성하거나 도형과 기호로 작성해보는 언플러그드 활동을 제안하고 있다[15]. 하지만, 학생이 설계한 알고리즘을 확인하기 위해서는 프로그래밍 도구의 문법에 따라 코드를 작성하는 과정을 거쳐 실행된 결과를 확인하는 활동이 필요하다.

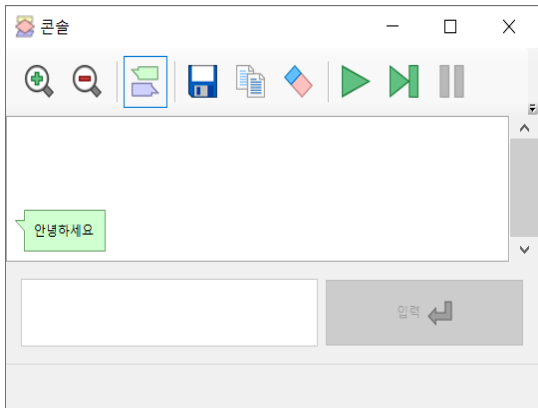
학생이 손으로 작성하던 순서도를 컴퓨터에서 작성하고 실행해 볼 수 있는 랩터(RAPTOR), 플로우고리즘(Flowgorithm) 등과 같은 다이어그램 도구가 등장하면서, 교육용 프로그래밍 도구로 코딩하는 과정을 생략하고, 알고리즘 설계에 집중할 수 있게 되었다[17].

랩터는 순서도 기반의 비주얼 프로그래밍 도구로 기호로 사용이 쉽고 범용 프로그램의 이해 없이도 사용할 수 있는 장점이 있다[18]. 학생이 손으로 작성한 순서도는 논리적인 형태로 실행이 가능하지만, 랩터는 작성한 순서도를 학생이 직접 실행시켜 결과를 직관적으로 확인할 수 있을 뿐 아니라 알고리즘의 각 단계를 하이라이트(Highlight)를 통해 관찰할 수 있다[19]. 더 나아가 학생이 작성한 순서도를 C, JAVA 등의 소스 코드로 변환할 수 있어 프로그래밍 활동과 자연스럽게 연결될 수 있다. 플로우고리즘 또한 랩터와 유사하게 순서도를 시각화하는 도구로 대학생들을 대상으로 알고리즘을 설계하는 교육에서 활용되고 있다[20].



[그림 1] Flowgorithm의 순서도 작성 모습

알고리즘 설계 중심의 도구를 정리하면 다음과 같다. 첫째, 순서도 수준의 알고리즘 표현이 가능하고, 알고리즘 수행의 시각화를 제공하고 있다. 둘째, 순서도를 제작하는 과정에서 일정 수준의 문법을 요구하고 있다. 블록형 프로그래밍 도구와 비교하여 기호 내부에 텍스트를 작성할 필요가 있고 정해진 형식에 따라 도구를 사용해야 한다. 마지막으로 작성한 순서도를 일반 프로그래밍 언어의 소스 코드로 변환이 가능하다.



[그림 2] Flowgorithm의 순서도 실행 모습

초등학생 수준에서 기존에 제안된 알고리즘 설계 중심의 도구는 다음과 같은 측면에서 사용에 어려움이 있다. 첫째, 알고리즘을 설계하는 도구의 사용법이 쉽다고는 하지만 별도의 응용프로그램의 수준에서 학습을 추가로 해야 한다는 부담이 발생한다. 언플러그드 수준에서 종이와 연필로 알고리즘을 작성하는 것이 아니라 컴퓨터를 사용하는 환경에서 추가적인 프로그램의 사용법을 익히는 것은 초등학생의 수준에서 학습량에 부담이 발생한다. 부담을 감소시키기 위해서 언플러그드 수준에서 학생이 간단한 조작만으로 알고리즘을 설계할 수 있는 도구가 요구된다. 둘째, 순서도를 작성하는 과정에서 키보드와 마우스를 이용하여 프로그래밍 활동과 유사한 수준으로 진행되고 있어 학생이 자유롭게 사고를 표현하는 데 어려움이 있다. 조작의 어려움을 해결하기 위해서 유아와 초등학생이 컴퓨팅 환경과 손쉽게 상호작용할 수 있는 텐저블 인터페이스 차원의 접근이 필요하다[21]. 마지막으로 순서도로 작성한 알고리즘을 초등학생의 수준에서는 블록형 프로그래밍 도구에서 사용할 수 있는 소스 코드로 변환될 필요가 있다. 학생이 설계한 알고리즘을 그대로 교육용 프로그래밍 도구에서 사용할 수 있도록 변환하는 기능을 제공할 필요가 있다.

### 3. 텐저블 코딩 도구 개발

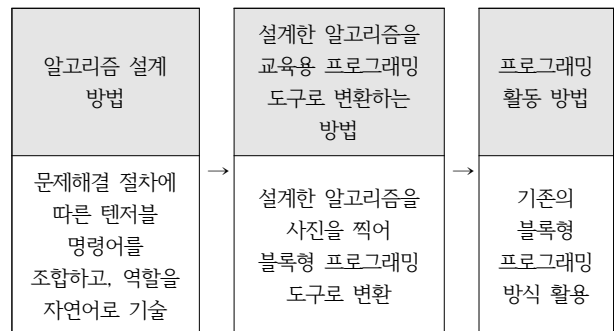
본 연구에서 제안하는 텐저블 코딩 도구는 초등학생이 언플러그드 활동을 할 수 있도록 종이에 인쇄하여 활용이 가능한 수준에서 설계하였다. 또한,

설계한 알고리즘을 사진을 찍는 간단한 조작으로 블록형 프로그래밍 도구의 코드로 변환할 수 있도록 앱을 설계하였다.

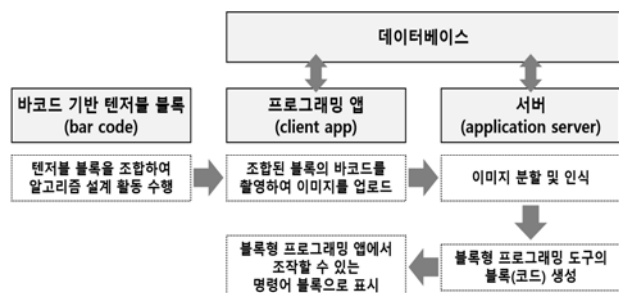
#### 3.1 설계

텐저블 코딩 도구는 블록형 프로그래밍 도구의 명령어를 각기 고유한 바코드로 구분하고, 바코드가 인쇄된 종이 블록을 문제해결 절차에 따라 조합하여 알고리즘을 설계할 수 있도록 제작하였다. 설계한 알고리즘을 스마트기기로 사진 촬영하여 분석된 바코드를 블록형 프로그래밍 도구의 코드로 변환하고, 스마트기기에서 기존의 블록형 프로그래밍 방식에 따라 계속하여 프로그래밍할 수 있다. 알고리즘 설계 활동과 프로그래밍 활동을 통합할 수 있는 텐저블 코딩 도구와 블록형 프로그래밍 도구의 연계방안을 단계적으로 정리하면 다음과 같다.

<표 1> 텐저블 코딩 도구와 블록형 프로그래밍 도구의 연계방안



텐저블 코딩 도구와 블록형 프로그래밍 도구가 연계된 구성도는 다음과 같다.



[그림 3] 텐저블 코딩 도구와 블록형 프로그래밍 도구의 연계방안 설계

텐저블 블록을 활용하여 알고리즘을 설계한 이후, 스마트기기의 카메라를 사용하여 촬영하고 이미지를 분석하여 블록형 프로그래밍 도구의 명령어 블록으로 변환한다. 이후 변환된 명령어 블록을 실행하여 학생이 설계한 알고리즘을 즉각적으로 확인하거나 알고리즘을 수정할 수 있다.

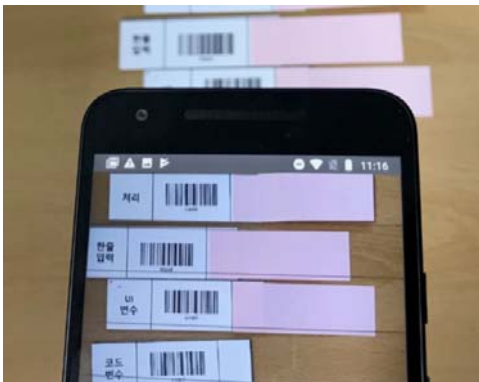
### 3.2 구현

텐저블 코딩 도구는 블록 이름, 명령어에 따른 바코드, 문제해결 절차를 칸으로 구성하여 인쇄할 수 있도록 제작하였다. 사용하는 텐저블 블록의 종류는 변수선언, 화면(UI)변수선언, 조건문(if) 시작과 종료, 반복문(for) 시작과 종료, 처리 명령어와 실행화면에서 사용하는 텍스트 입력칸, 버튼 등의 컴포넌트로 구성하였다.

텐저블 블록 이름	명령어에 따른 바코드	문제해결 절차를 자연어로 작성
처리	 casti	

[그림 4] 텐저블 명령어 예(처리 블록)

텐저블 블록을 위에서 아래로 순차적으로 조합하여 알고리즘을 설계한 내용을 사진 찍어 블록형 프로그래밍 도구로 변환하기 위해 리액트 네이티브(React Native), 바코드 인식 라이브러리, 블록키(Blockly)를 사용하여 자바스크립트 언어에 기반한 블록형 프로그래밍 앱을 제작하였다.



[그림 5] 설계한 알고리즘을 블록형 프로그래밍 도구에 입력하기 위해 사진 촬영하는 모습

개발한 도구를 사용하는 방법은 다음과 같다. 첫째, 학생이 문제해결 절차를 포스트잇에 작성하고, 작성한 내용에 따라 필요한 텐저블 블록을 찾아 포스트잇을 붙이면서 블록을 조합한다. 둘째, 블록형 프로그래밍 도구를 사용하여 조합한 텐저블 블록을 촬영하고 블록형 코드로 변환한다. 셋째, 변환된 블록에 변수와 값을 설정하고, 실행하여 학생이 설계한 알고리즘을 확인한다. 결과에 따라 블록형 프로그래밍 도구에서 디버깅 활동을 수행한다.



[그림 6] 버튼이 클릭 되면 입력된 값을 처리하여 출력하는 알고리즘 설계 예시

예시는 버튼이 클릭 되었을 때, 사용자가 입력한 값(a)을 불러오고 처리( $b = a \times 0.39$ )하여 결과를 출력(b)하는 알고리즘을 설계하고 이를 블록형 프로그래밍 도구로 변환한 모습이다.

## 4. 연구방법

### 4.1 연구대상

개발한 도구의 사용성 평가를 위한 목적으로 소프트웨어 교육을 통해 블록형 프로그래밍 활동을 경험한 적이 있는 초등학교 6학년 27명을 대상으로 텐저블 코딩 도구를 적용하였다.

<표 2> 연구대상

구분	남학생	여학생	합계
초등학교 6학년	15(55.6%)	12(44.4%)	27(100.0%)

### 4.2 연구절차

연구는 텐저블 코딩 도구로 알고리즘을 설계한 이후 블록형 프로그래밍 도구 앱으로 사진을 찍어 프로그래밍하여 문제를 해결하는 방법과 별도의 알고리즘 설계 없이 블록형 프로그래밍 도구 앱으로 프로그래밍하여 문제를 해결하는 방법을 모든 학생이 3시간 동안 경험하고 텐저블 코딩 도구에 대한 사용성 평가를 수행하였다.

<표 3> 연구절차

G	X	O <sub>1</sub> , O <sub>2</sub>
---	---	---------------------------------

G : 실험반(무작위로 선정) n=27

X : 텐저블 코딩 도구와 블록형 프로그래밍 도구로 문제풀이

O<sub>1</sub> : 텐저블 코딩 도구 사용성 설문

O<sub>2</sub> : 텐저블 코딩 도구를 사용해본 경험 인터뷰

### 4.3 연구도구

제안하는 도구가 초등학생의 수준에서 사용하기 적합한 것인지 판단하기 위한 목적으로 기존의 초등학교를 대상으로 하는 사용성 설문도구를 제안하는 도구에 적합하도록 수정하여 5점 척도로 설문하였다[22]. 개발된 텐저블 코딩 도구에 대한 사용성 설문 도구의 신뢰도는 Cronbach  $\alpha$  계수를 산출한 결과 .845로 나타났다.

<표 4> 텐저블 코딩 도구 사용성 설문

번호	질문
1	나는 바코드 사진을 찍어 필요한 종이 블록을 선택하는 것은 편리하다고 생각한다.
2	나는 바코드 사진을 찍어 필요한 여러 종이 블록들을 연결하는 것은 편리하다고 생각한다.
3	나는 바코드 사진을 찍는 종이 블록들을 사용하는 방법은 쉽다고 생각한다.
4	나는 바코드 사진을 찍는 종이 블록들을 사용하는 방법을 기억할 수 있다.
5	나는 바코드 사진을 찍는 종이 블록들을 사용하는 방법을 다른 친구들에게 설명할 수 있다.
6	나는 바코드 사진을 찍는 종이 블록들을 사용하는 것은 재미있다고 생각한다.

## 5. 연구결과

텐저블 코딩 도구에 대한 사용성 평가 결과, 4.13점으로 나타나 만족하는 것으로 분석되었다.

<표 5> 텐저블 코딩 도구에 대한 사용성 분석

구분	텐저블 코딩 도구의 사용성 M(SD)
전체(n=27)	4.13(0.80)

성별에 따른 차이 분석 결과 여학생이 4.21점, 남학생이 4.07점으로 나타났지만, 통계적으로 유의한 차이는 없는 것으로 분석되었다.

<표 6> 성별에 따른 사용성 차이 분석

구분	남학생 M(SD) (n=15)	여학생 M(SD) (n=12)	Mann-Whitney Z
사용성	4.07(0.88)	4.21(0.73)	.789

텐저블 코딩 도구로 알고리즘을 설계하고 블록형 프로그래밍 도구로 프로그래밍하는 활동과 알고리즘 설계 없이 블록형 프로그래밍 도구로 프로그래밍하는 두 활동을 모두 수행한 이후, 새로운 문제를 접한다면 어떤 방법으로 해결하고 싶은지 하나의 방법만 선택하도록 설문하였다. 설문 결과, 텐저블 코딩 도구를 사용하여 알고리즘을 설계한 이후 프로그래밍 활동을 하는 방법을 선호하는 학생

이 17명(63.0%), 별도의 알고리즘 설계 없이 프로그래밍 활동만 하는 방법을 선호하는 학생이 10명(37.0%)으로 나타났다. 두 가지 프로그래밍 활동에 대한 남학생의 선호도는 비슷하였지만, 여학생의 경우 프로그래밍 활동 전 텐저블 코딩 도구를 사용하는 것을 선호하는 것으로 나타났다.

<표 7> 텐저블 코딩 도구 선호도 분석

구분	남학생 (n=15)	여학생 (n=12)	합계 (n=27)
텐저블 코딩 도구로 알고리즘 설계 후 프로그래밍 활동	8(53.5%)	9(75.0%)	17(63.0%)
알고리즘 설계 없이 프로그래밍 활동	7(46.7%)	3(25.0%)	10(37.0%)

인터뷰를 통해 두 방법 중 선호하는 이유를 문의한 결과, 블록형 프로그래밍 도구에서 충분히 알고리즘을 설계하면서 동시에 프로그래밍할 수 있을 뿐 아니라 별도로 제시된 텐저블 코딩 도구를 사용하고 자연어로 알고리즘을 기술하는 활동이 번거로워 알고리즘 설계 없이 즉각적으로 프로그래밍하는 방법을 선호하는 것으로 응답하였다. 반면, 제시된 문제가 복잡하거나 아직 알고리즘을 설계하는 것에 익숙하지 않은 경우, 책상 위에서 텐저블 코딩 도구를 조작하여 문제해결 절차를 고민하고 다른 친구와 상호작용을 할 수 있어 텐저블 코딩 도구를 선호하는 것으로 분석되었다. 즉, 제안하는 텐저블 코딩 도구가 프로그래밍 활동에 익숙하지 않는 학생에게 효과적이라고 할 수 있다.

연구의 결과를 종합해보면, 제안하는 텐저블 코딩 도구는 알고리즘 설계와 프로그래밍 활동을 동시에 수행하기 어려운 학생에게 설계와 구현을 연계하는 학습경험을 제공하여 알고리즘적 사고의 학습에 도움을 줄 수 있는 것으로 해석할 수 있다.

## 6. 결론

본 연구는 초등학생을 위한 소프트웨어 교육에서 알고리즘적 사고를 가르치기 위한 목적으로 알고리즘을 설계하고 블록형 프로그래밍 도구로 구현하는 단계를 구분하여 제공할 수 있는 도구를 제안하고 사용성을 평가하였다.

연구 결과, 제안하는 텐저블 코딩 도구의 사용성은 만족하는 것으로 분석되었고, 남학생보다 여학생의 사용성이 높은 것으로 나타났다. 인터뷰를 통해 텐저블 블록을 사용하여 알고리즘을 설계하고 블록형 프로그래밍 도구로 프로그래밍하는 활동은 컴퓨터로 문제를 해결하는 활동에 익숙하지 않은 학생에게 적합한 것으로 분석되었다.

향후 연구로 알고리즘 설계에 필요한 텐저블 블록의 명령어의 종류와 조작 방법에 대한 구체적인 연구가 필요하고, 학생의 알고리즘적 사고 수준에 따라 알고리즘 설계와 프로그래밍 활동을 통한 구현의 단계를 보다 세부적으로 구분할 필요가 있다.

## 참 고 문 헌

- [1] Manyika, J., Lund, S., Chui, M., Bughin, J., Woetzel, J., Batra, P., & Sanghvi, S. (2017). *Jobs lost, jobs gained: Workforce transitions in a time of automation*. McKinsey Global Institute.
- [2] 김용민(2019). 주요 국가별 인공지능(AI) 인력 양성 정책 및 시사점. *보건산업브리프*, 276,
- [3] 정보통신기술진흥원(2018). 해외 주요국의 4차 산업혁명 대응 인재양성 정책 동향. 2018-02 호
- [4] 교육부(2015). 제2015-74호 실과(기술·가정)/정보과 교육과정.
- [5] 교육부·과학기술정보통신부(2016). 소프트웨어 교육 활성화 기본계획.
- [6] Popat S., &Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computer & Education*. 128, 365-376.
- [7] 김태훈·김종훈(2013). Kodu를 이용한 프로그래밍 학습이 초등학생의 논리적 사고력과 학습 흥미에 미치는 영향. *컴퓨터교육학회논문지*. 16(3), 13-22.
- [8] 오소희·김응진·김성식(2017). 메타인지 향상을 위한 엔트리 프로그래밍 교육 콘텐츠 개발 및 적용. *컴퓨터교육학회논문지*. 20(5), 61-68.

[9] 김용천·최지영·권대용, 이원규(2013). 초등 학생의 프로그래밍 학습을 위한 알고리즘적 사고 문제 모델 기반의 활동지 개발 및 적용. **정보교육학회논문지**. 17(3), 233-242.

[10] 오미자(2017). 스크래치 프로그램을 활용한 프로그래밍 교육에 대한 비전공자의 인식 연구. **컴퓨터교육학회논문지**. 20(1), 1-11.

[11] 최정원·이영준(2014). 프로그래밍 학습에서 학습자의 어려움 분석. **컴퓨터교육학회논문지**. 17(5), 89-98.

[12] 김은지·이태욱(2017). 블록형 프로그래밍 언어 학습을 위한 순서도 적용 방안: 엔트리 미션 해결하기. **한국컴퓨터교육학회 학술발표대회논문집**. 21(2), 21-24.

[13] 이용배·박지은(2012). 초등학생 대상의 활동 중심 순서도 교육 방법. **정보교육학회논문지**. 16(4), 489-502.

[14] 정은숙·허민·진영학·김영식(2009). 중학교 프로그래밍 수업에서 순서도학습이 논리적 사고력과 성취도에 미치는 영향. **컴퓨터교육학회논문지**. 12(6), 11-19.

[15] 김정아·오민우·양영훈·김태훈·김종훈(2019). 언플러그드 방식을 활용한 알고리즘 수행시간 분석이 초등학교 3학년 학생의 컴퓨팅 사고력에 미치는 효과. **정보교육학회논문지**. 23(1), 19-27.

[16] 김수환·이원규·김현철(2009). 개정된 정보교육과정에서 교육용프로그래밍언어의 교육적 적용방안. **컴퓨터교육학회논문지**. 12(2), 23-31.

[17] 김명민·이민정(2019). 비전공자를 위한 교육용 프로그래밍 언어의 비교 연구: 프로그래밍 언어 설계 원칙의 관점으로. **컴퓨터교육학회논문지**. 22(1), 47-61.

[18] Carlisle, M. C., Wilson, T. A., Humphries, J. W., & Hadfield, S. M. (2005). RAPTOR: a visual programming environment for teaching algorithmic problem solving. *In Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*. 176-180.

[19] 김성울·이종연(2015). RAPTOR 프로그래밍 교육도구를 이용한 알고리즘 교육. **컴퓨터교육학회논문지**. 18(6), 23-31.

[20] Thangavelu, S., Rao, V., Shyamala, C. K., & Velayutham, C. S. (2019). Introductory Programming Using Non-Textual Modalities. *2019 IEEE Global Engineering Education Conference (EDUCON)*. 867-871.

[21] 박만구·김동훈·김지영·김혜진·이보람·조영석·홍제훈(2018). 텐저블 코딩 교육 프로그램의 효과성 분석. **한국초등교육**. 29(4), 23-49.

[22] 심재권·이원규·권대용(2015). LED 기반 텐저블 프로그래밍 도구 개발 및 적용. **컴퓨터교육학회논문지**. 18(1), 35-43.

## 심재권



2007 경인교육대학교  
초등교육과(교육학학사)  
2012 고려대학교  
컴퓨터교육학과(이학석사)  
2017 고려대학교 컴퓨터교육학과(이학박사)  
2017~현재 고려대학교 영재교육원 연구교수  
관심분야: 컴퓨터교육, 프로그래밍 교육  
E-Mail: jaekwoun.shim@gmail.com

## 권대용



2003 고려대학교  
컴퓨터교육과(이학사)  
2006 고려대학교  
컴퓨터교육학과(이학석사)  
2011 고려대학교 컴퓨터교육학과(이학박사)  
2011~현재 고려대학교 교과교육연구소 연구교수  
2015~현재 (사)미래융합교육연구소 이사장  
관심분야: SW융합교육, 학습과학, 지능형학습시스템, 영재교육  
E-Mail: dykwon.edu@gmail.com