

언플러그드 방식을 활용한 알고리즘 수행시간 분석이 초등학교 3학년 학생의 컴퓨팅 사고력에 미치는 효과

김정아 · 오민우 · 양영훈 · 김태훈 · 김종훈

제주대학교

요약

본 연구에서는 알고리즘의 수행시간 분석에 초점을 맞춘 언플러그드 방식의 교육이 초등학교 3학년 학생들의 컴퓨팅 사고력 향상에 미치는 효과를 분석해보았다. 현직 초등교사 70명과 초등학교 3학년 학생 120명을 대상으로 실시한 요구분석 결과를 바탕으로, ADDIE 모형의 절차에 따라 언플러그드 교육프로그램을 개발하였다. 개발한 교육프로그램으로 초등학교 3학년 학생 24명에게 총 6일 동안 36시간의 수업을 진행하였고, 같은 시간 동안 보드게임 교육을 받은 24명의 학생들과 사전·사후 비교 검사를 통해 효과를 분석하였다. 분석 결과, 알고리즘의 수행시간 분석에 초점을 맞춘 언플러그드 방식의 교육은 초등학교 3학년 학생들의 컴퓨팅 사고력에 긍정적인 효과를 미치는 것으로 확인되었다.

키워드 : 알고리즘, 컴퓨팅사고력, 수행시간 분석, 언플러그드

The Effect of Algorithm Execution Time Analysis using Unplugged Program on the Computational Thinking of Third Grade Students

Jungah Kim · Minwoo Oh · Younghun Yang · Taehun Kim · Jonghoon Kim

Jeju National University

ABSTRACT

This study analyzed the effect of an unplugged educational program which focuses on the algorithm execution time analysis on enhancing the computational thinking of the third grade students. An unplugged educational program was developed by following the procedures of the ADDIE model based on the result of the demand analysis conducted with 70 incumbent elementary school teachers and 120 third grade students. The developed program was applied to 24 third grade students for 36 classes in six days and a comparison was made through pre-post comparison tests with 24 students who were taught with a board game program for the same period of time. As a result, the unplugged education program which focuses on the algorithm execution time analysis is deemed to have positive effect on the computational thinking of the third grade students.

Keywords : Algorithm, Computational thinking, Execution time analysis, Unplugged

교신저자 : 김종훈(제주대학교 초등컴퓨터교육전공)

논문투고 : 2019-01-24

논문심사 : 2019-02-20

심사완료 : 2019-02-22

1. 서론

‘4차 산업혁명’ 시대의 교육은 기존의 교육과 같이 단순히 지식을 전달하는 형태가 아닌 ‘지식을 습득하는 방법’을 알 수 있도록 교육해야 한다. 이러한 능력을 위해서는 문제 해결을 위해 필요한 정보를 스스로 찾고, 선별하여, 필요한 경우 새로운 정보로 가공해낼 수 있는 사고를 할 줄 아는 컴퓨팅 사고력(Computational Thinking)이 필수적이다.[1][2]

컴퓨팅 사고력은 컴퓨터 과학 분야를 전공하는 사람만이 갖추어야 하는 역량이 아닌 모든 사람이 학습해야 하는 보편적인 능력으로 간주되며 세계 주요 선진국인 미국, 영국, 이스라엘에서는 컴퓨터과학을 정규교과로 실시하고 있다. 특히 만 5세 아동이 수를 배우듯 컴퓨팅 사고력 또한 어릴 때부터 체계적으로 배워야 한다고 강조되고 있다.(Wing 2006.)

이에 반해 우리나라는 소프트웨어 교육 시수가 매우 부족하다. 초등학교의 경우 5~6학년을 대상으로만 소프트웨어 교육이 실시되고 있기에 초등 저학년의 컴퓨팅 사고력이 함양된 소프트웨어 교육이 필요하다.[3] 따라서 이번 연구는 3학년에 초점을 맞추어 컴퓨팅 사고력을 기르기 위한 방안을 연구해보았다.

초등학생을 대상으로 한 컴퓨터과학 교육 현장은 학생 개개인이 교사의 지도에 따라 애니메이션이나 로봇을 활용하여 자료구조를 입력하고 알고리즘의 원리를 이해할 수 있는 형태가 대다수를 이루고 있다.[5] 그러나 프로그래밍 언어에 대한 지식이나 활용해본 경험이 거의 없는 학습자, 그 중에서도 초등학생들에게는 알고리즘을 학습하는 것보다 프로그래밍 언어를 활용해야 한다는 것이 더 큰 부담이 될 수 있고, 특히 초등학교 저학년에게는 더 큰 방해 요소로 작용할 수 있다.[6] 이 점을 고려하여 3학년 대상의 소프트웨어 교육 방법으로 언플러그드 방법을 선택하였다. 언플러그드 컴퓨팅 방법은 체험 활동과 구체물 사용의 방법을 통해 학생들에게 컴퓨터과학내용에 기반한 문제해결 상황을 해결함으로써 계산적사고를 기를 수 있는 기회를 제공할 수 있고,[7] 특히 저학년 소프트웨어 교육에 대한 흥미 유발에도 긍정적인 영향을 미친다.[6]

컴퓨팅사고력은 컴퓨팅의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력이

다.[4] 컴퓨터과학의 관점에서 효율적인 문제해결은 최소의 명령과 시간으로 문제를 해결하는 것이다. 이는 문제 해결에 필요한 알고리즘의 최소명령, 수행시간과 밀접한 연관이 있다. 세계적으로 많은 언플러그드활동이 개발되어 있지만, 수행시간에 초점을 맞춘 연구 사례가 미비하여 본 연구의 교육 내용을 컴퓨터과학에서 통용되는 다양한 알고리즘과 그 알고리즘의 수행시간을 분석하는데 초점을 맞추었다.

결론적으로 초등학교 3학년 학생을 대상으로 언플러그드 방식을 활용한 알고리즘 수행시간 분석에 초점을 맞춘 교육을 실시하였고 이 교육이 컴퓨팅사고력에 어떤 영향을 미치는지 연구해보았다. 교육 프로그램 실시 후, 컴퓨팅 사고력 검사를 실시하였으며 검사도구로는 김병수(2014)의 연구에서 개발한 계산적 인지력 검사(Computational Congnition Test) A, B형을 선정하였다.[8]

2. 이론적배경

2.1 언플러그드 교육

언플러그드 교육 방식은 뉴질랜드의 팀 벨(Tim Bell) 교수가 제안한 교수학습 방법으로 특정 소프트웨어나 하드웨어에 의존하지 않고 생활 속의 다양한 소재를 활용하여 컴퓨터 과학의 기본 개념을 가르치기 위한 활동들을 의미한다.[9]

국내에서도 언플러그드 컴퓨팅 방법을 이용한 다수의 연구들이 있는데, 이 연구들은 학생들이 놀이를 통해 컴퓨터과학의 원리를 학습할 수 있다는 것을 알려준다. 특히 추상적일 수 있는 컴퓨터의 작동 원리를 이해하는데 유용하다.[7] 또한, 컴퓨터 과학의 원리를 쉽고 재미있게 배울 수 있다는 점에서 컴퓨터 과학이 생소한 저학년 학생들에게 적합한 교육 방법이다.

언플러그드 교육 방식이 갖는 특징을 보면 다음과 같다.[11]

<Table 1> Unplugged principle

원리	내용
No computers required	언플러그드 활동은 컴퓨터를 필요로 하지 않는다.
Real computer science	구체적인 활동을 통해 실제적인 컴퓨터과학의 개념 원리를 학습한다.
Learning by doing	구성주의 방식으로 실제적이고 귀납적인 활동을 통해 학습한다.
Fun	학습자의 흥미와 호기심을 유발하는 요소를 포함한다.
No specialised equipment	우리 주변에서 쉽게 접할 수 있는 도구를 활용한다.
Variations encouraged	언플러그드 활동은 다양한 소재와 방법으로 새로이 개발 가능하다.
For everyone	컴퓨터과학을 배우고자 하는 모든 연령을 위한 방식이다.
Co-operative	언플러그드 활동은 협동학습이 더 적합하다.
Stand-alone activities	하나의 활동에 하나의 원리를 적용하여 교육한다.
Resilient	언플러그드 활동은 학생들의 다양한 행동에 융통성을 갖고 있다.

2.2 컴퓨팅 사고력

교육부가 소프트웨어 교육을 통해 길러내고자 하는 인재상은 컴퓨팅사고력을 가진 창의·융합 인재이다. 여기서 컴퓨팅 사고력이 의미하는 바는 컴퓨터 과학의 기본적인 개념과 원리를 기반으로 문제를 효율적으로 해결할 수 있는 사고 능력이다.[10] 컴퓨팅 사고력은 모든 학문 영역의 문제 해결을 위한 기초적인 사고 능력이며, 실생활과 관련된 복잡한 문제를 해결하기에 가장 적합한 인지 능력이다.[14]

컴퓨팅사고력(Computational Thinking, CT)은 2006년 미국 카네기멜론 대학의 Wing교수에 의해 소개되었다. Wing은 컴퓨팅사고력은 읽기, 쓰기, 말하기와 더불어 필수로 익혀야 하는 능력이라고 하였으며, ‘해결해야 할 문제를 만났을 때 컴퓨터 과학자처럼 사고하는 것’의 개념으로 소개하였다.

컴퓨팅 사고력을 구성하고 있는 핵심 요소들을 살펴보면 <Table 2>과 같다.

<Table 2> Elements of Computational Thinking

Component	Definition
Data Collection	The process of gathering appropriate information
Data Analysis	Making sense of data, finding patterns, and drawing conclusions
Data Representation	Depicting and organizing data in appropriate graphs, charts, words, or images
Problem Decomposition	Breaking down tasks into smaller manageable parts
Abstraction	Reducing complexity to define main idea
Algorithms & Procedures	Series of ordered steps taken to solve a problem or achieve some end
Simulation	Representation or model of a process. Simulation also involves running experiments using models
Parallelization	Organize resources to simultaneously carry out tasks to reach a common goal

2.3 수행시간

알고리즘의 수행시간은 ‘시간복잡도’와 유사한 의미를 지니며, 알고리즘이 어떤 문제를 해결하는데 걸리는 시간으로 정의된다. 시간복잡도와 수행 시간은 비례하는데, 시간 복잡도 수치가 작을수록, 알고리즘의 수행시간이 짧고 효율적인 알고리즘이 된다. 동일한 문제를 해결하고 같은 결과를 내는 경우라도 어떤 알고리즘을 사용하는가에 따라 문제 해결까지 걸리는 시간이 달라질 수 있고, 이는 효율적인 문제해결과 직결된다.

알고리즘의 수행시간을 분석함에 따라 알고리즘의 작동 원리를 이해하고 문제를 해결하기 위해 적합하고 효율적인 알고리즘을 선택하는 능력이 향상된다. 알고리즘의 수행시간을 분석하고 문제해결에 적절한 알고리즘을 선택, 적용하는 교육을 통해 효율적으로 문제를 해결하는 컴퓨팅사고력을 기를 수 있다.

2.4 초등학교 3학년 학생의 발달단계

피아제의 인지발달이론에 따르면 초등학생의 발달단계는 구체적조작기(7~11세)와 형식적조작기(11세 이후)로 구분된다. 이 중 3학년 학생은 구체적조작기에 해당

된다. 구체적 조작기는 사고를 하는데 있어 보고나 만질 수 있는 구체적 사물을 통해 추상화가 가능해진다. 사물을 조작하고 학습하는 능력이 생긴 구체적조작기의 아동들에게 사물과 놀이를 통해 학습하는 언플러그드는 최적의 교육방식이다.

또한 구체적조작기에는 인과관계에 대한 일련의 추리를 할 수 있게 된다. 이는 논리적인 사고를 필요로 하는 알고리즘 교육이 적합함을 의미한다. 뿐만아니라 이 시기의 아동은 구체적 사물을 통해 추상화가 가능해지는데, 이는 구체적조작물을 이용하면 초등학교도 검색알고리즘과 같은 컴퓨터과학의 요소를 교육받고 알고리즘 문제해결의 아이디어를 제시할 수 있다는 것을 의미한다.[5]

2.5 선행연구분석

그동안 다양한 연구에서 언플러그드 교육에 대한 다양한 논의가 진행되어 왔다. 특히 교육 대상에 따른 언플러그드 학습의 효과에 대한 연구가 많이 이루어졌다.

전영욱(2018)은 협동학습 중심의 언플러그드 컴퓨터 과학 활동이 초등학교 고학년 학생들의 컴퓨팅 사고력에 미치는 효과를 분석하였고, 초등학교 학생들에게 구체적 조작활동이 포함된 언플러그드 활동이 컴퓨팅 사고력 향상에 효과적임을 검증하였다.[5]

박영선(2012)는 언플러그드를 활용한 컴퓨터과학교육의 연구 동향을 파악하였고, 언플러그드 교육 방법에 대한 연구가 초등학교를 대상으로 가장 많이 연구되고, 초등학교 컴퓨터과학 교육에서 가장 많이 활용되고 있다는 것을 검증하였다.

김정일(2012)은 특수학교 학생 23명을 대상으로 언플러그드 교육 4차시 수업을 실시하였다. 연구 결과 언플러그드를 실시한 집단이 강의식 수업을 실시한 통제집단보다 학업성취도 측면에서 효과가 있다는 것을 검증하였다.

한선관(2010)이 초등학교 영재 학생들을 대상으로 언플러그드 컴퓨터 수업을 진행하다. 알고리즘의 이해와 EPL 프로그래밍을 작성하면서 원리를 파악하도록 설계하였으며, 이를 적용하여 영재학생들이 언플러그드에 높은 학업성취도를 보인다는 결과를 검증하였다.

위에서 살펴본 것처럼 언플러그드 교육에 대한 다양한 논의가 진행되지만, 여러 연구결과들은 언플러그드

는 교육 방법과 교육대상에만 초점을 맞추었을 뿐 교육 내용에 초점을 맞춘 연구는 미비하다. 따라서 본 연구는 언플러그드라는 교육 방법을 사용하고, 교육 내용은 알고리즘과 그 수행시간을 분석하는데 초점을 맞추었다. 알고리즘별 수행시간을 분석하고 문제에 알맞은 알고리즘을 선택, 적용하는 능력이 컴퓨팅 사고력에 큰 영향을 미칠 것이라는 가설을 세웠고, 본 연구에서 가설을 검증해보기 위해 교육프로그램을 만들고 적용하여 그 효과를 검증하였다.

2.6 요구분석

컴퓨팅사고력 신장을 위한 교육프로그램에 대하여 다음과 같은 내용의 요구 분석을 실시하였다.

- 소프트웨어 교육 경험
- 소프트웨어 교육 방법
- 소프트웨어 교육의 필요성
- 알고리즘의 수행시간 분석

요구분석은 ○○대학교에서 실시하는 컴퓨터교실(교육기부 프로그램)에 지원한 초등학교를 포함하여, 도내 초등학교 3학년 164명의 학생들과 도내 초등 현직교사 66명을 대상으로 실시하였다.

SW 교육 경험에 대한 조사 결과는 교사는 56.1%이지만 3학년 학생들은 26.7%로 소프트웨어교육을 경험한 학생이 매우 적은 것으로 나타났다.

<Table 3> Software education experience

	○	×
teacher	37(56.1%)	29(43.9%)
student	31(26.7%)	85(73.3%)

<Table 4>에 의하면 많은 교사들이 소프트웨어 교육 방법으로 언플러그드 교육 방법을 선호하는 것으로 나타났다. <Table 5>에 의하면 학생들은 EPL에 비해 언플러그드 활동을 경험해본 학생이 적은 것으로 나타났다. 또한 교육의 내용적인 측면에서 <Table 6>의 결과를 참고하여 효율적인 프로그래밍을 위한 알고리즘의 수행시간 분석을 교육내용으로 선정하였다.

<Table 4> Preferred Software education Method

	Unplugged	EPL	Physical Computing
experienced	15(40.5%)	17(45.9%)	5(13.5%)
Unexperienced	20(69%)	4(13.8%)	5(17.2%)

<Table 5> Expreience in Software education method

	Unplugged	EPL	Physical Computing
student	11(35%)	13(42%)	7(23%)

<Table 6> The need for algorithm execution time analysis for efficient programming

	need	normal	no need
teacher	49(74.2%)	14(21.2%)	3(4.5%)

도내 초등학교 3학년 120명의 학생들과 도내 초등 현직교사 70명을 대상으로 실시한 요구분석의 결과 교육방법은 언플러그드 교육방식으로 선정하였고, 교육 내용은 알고리즘의 수행시간을 분석하는 것에 초점을 맞추었다.

3. 수행시간 분석 중심의 언플러그드 교육

3.1 교육 방향

본 연구에서는 ADDIE 모형에 따라 교육 프로그램을 개발하였다.

단계	내용
Analysis (분석)	학습자 분석 요구 분석
Design (설계)	교수전략과 매체 선정 - 언플러그드 활동 학습 - 언플러그드 평가도구 설계 - 학습자 컴퓨팅사고력 검사지
Development (개발)	교수·학습 과정안(36차시) 학생 활동지(36차시)
Implementation (실행)	수업 실시
Evaluation (평가)	CT검사지

요구분석 결과 대부분의 학습자들은 소프트웨어 교육을 처음 접하는 학생들임을 고려하여 교육 방향을 설정하였다. 먼저 학생들이 흥미를 느낄만한 언플러그드 활동으로 알고리즘의 개념을 익히고 자연스럽게 수행시간의 개념까지 익히도록 수업교재를 설계하였다.

학생들의 학습내면화를 위해 우리 주변에서 쉽게 접할 수 있는 문제들로 알고리즘을 학습한다. 다양한 알고리즘 중 탐색알고리즘, 역추적알고리즘을 중점적으로 학습하고 각각의 알고리즘 수행시간을 분석한다. 탐색 알고리즘의 종류에도 순차탐색, 이진탐색 등 다양한 탐색 알고리즘이 있다. 어떤 탐색알고리즘을 선택하는가에 따라 문제해결의 시간이 달라진다는 점을 학생들 스스로 발견할 수 있도록 교육방향을 설정하였다.

3.2 교육 내용

교육 내용은<Table 7>과 같다.

<Table 7> Unplugged education focused on performance time analysis

Hour	Learning theme
1-11	Algorithm & Execution time analysis -Changing seat activity -Tower of Hanoi
	Input of pre-test
	Search Algorithm & Execution time analysis -Binary search algorithm -Sequential search algorithm
12-17	
18-23	Backtracking Algorithm & Execution time analysis
24-29	Organize learning contents Creating a project plan for individual project
30-36	Present project work Input of post-test

평가 도구로는 수업 활동의 효과를 검증하기 위해 김병수(2014)의 연구에서 개발한 계산적 인지력 검사(Computational Cognition Test) A, B형을 선정하였다.[8]

3.3 교육 방법

본 연구에서는 ○○대학교에서 진행된 교육기부 프로그램의 지원자 48명을 대상으로 6일 동안 1일 6차시씩 36차시를 진행하였다. 48명의 학생을 두 집단으로 나누고 한 집단은 수행시간 분석에 초점을 맞춘 언플러그드 교육을 적용하였고, 다른 집단에는 보드게임을 활용한 교육을 하였다.

언플러그드 활동을 통해 알고리즘과 수행시간의 개념을 익힌 후 각각의 알고리즘에 대한 수행시간을 분석해보았다. 다양한 실생활의 문제를 제시하고, 해결해야 하는 문제에 따라 다양한 알고리즘을 적용해보고 알고리즘별로 수행시간에 차이가 있다는 점을 학습하였다. 6차시의 수업을 마친 후에는 매일 실생활과 접목한 과제를 제시하여 가정에서 복습할 수 있도록 하였다. 연필코딩을 통해 수행시간의 개념을 알아보는 차시의 교육 자료를 살펴보면 다음과 같다.

수행시간이란?

➡ **해당책이** 중요한 일이 있었던 날과 평범한 날의 알고리즘에는 어떤 차이가 있었나요?

이렇게 양치를 하는데 까지 걸리는 단계에 차이가 있습니다.
단계에 따라 물론 걸리는 시간에도 차이가 있겠죠? 단계를 모두 해결하기까지 걸리는 시간을 우리는 **수행시간**이라고 합니다.

! 수행시간

문제를 해결하기 위해 **실행**해야하는 단계의 길이
문제를 해결하기 **까지** 걸리는 시간

해당책이 이를 닦을 때 평범한 날과 중요한 날 중 어떤 날의 수행시간이 더 길었나요?

	앞으로가기
	오른쪽으로 돌기
	왼쪽으로 돌기

~알고리즘 명령표~

그 중 빨간 길로 통과 하는 방법을 알고리즘으로 표현해보겠습니다.

해당책이의 빨간 통과길은 이렇게 표현할 수가 있겠네요. 총 8번의 알고리즘 단계를 거치네요!
이와 같은 방법으로 다른 길로 통과하는 알고리즘을 여러분들이 직접 표현해보나요?

<노란색 길>

<초록색 길>

➡ **학교가는** 길을 알고리즘으로 생각해봅시다.

학교가는 방법과 단계 = 알고리즘

명령어의 **갯수**, 학교에 도착하기까지 걸리는 시간 = ()

이렇게 '학교를 간다'라는 똑같은 과제를 수행하는데도 학교를 가는 방법마다 명령의 갯수가 다릅니다. 같은 문제를 수행하더라도 방법에 따라 명령의 갯수와 수행시간이 달라질 수 있다는 것이죠.

정리!

**같은 문제를 해결하더라도
방법과 명령에 따라 수행시간이 달라집니다.**

4. 적용 결과 및 분석

2개의 동일집단으로 나눈 후, 한 집단에는 언플러그드를 활용한 수행시간 분석 교육을 하였고, 나머지 집단에는 보드게임을 진행하였다. 36시간의 교육을 마친 후, 컴퓨팅 사고력 검사를 실시하였으며 검사도구로는 김병수(2014)의 연구에서 개발한 계산적 인지력 검사(Computational Cognition Test) A, B형을 선정하였다.[8]

4.1. 교육 프로그램 효과 검증

수행시간 분석 중심의 언플러그드 교육이 3학년 학생들의 컴퓨팅 사고력에 어떤 영향을 주는지 분석하기 위한 검증을 실시하였다.

4.2. 컴퓨팅 사고력 사전검사

교육프로그램 적용 전 실험집단과 비교집단의 컴퓨팅 사고력 하위요소 각각의 점수에서 차이가 존재하는지 알아보기 위해 독립표본 t검정을 실시하였다. 그 결과는 <table 8>과 같다.

<Table 8> Pre-test result

Subscales	Group	Descriptive Statistics			t	p
		n	M	SD		
Sequential structure	Experimental	23	1.39	.722	.307	.760
	comparison	24	1.33	.565		
Algorithm	Experimental	23	.35	.647	-.162	.872
	comparison	24	.38	.495		
Conditional branch	Experimental	23	2.04	.878	.737	.465
	comparison	24	1.88	.680		
Variable	Experimental	23	.61	.722	-.082	.935
	comparison	24	.63	.647		
Function	Experimental	23	1.65	1.584	-.031	.976
	comparison	24	1.67	1.633		
Recursion	Experimental	23	.22	.518	-1.154	.254
	comparison	24	.24	.654		
Total	Experimental	23	6.09	3.765	-.079	.937
	comparison	24	6.17	3.102		

실험집단과 비교집단을 비교하였을 때 6개의 하위영역 중 순차구조, 조건분기, 함수 영역은 실험집단의 평균점수가 높았고 알고리즘, 변수, 재귀영역은 비교집단의 평균점수가 더 높았으나, 통계적으로 유의한 차이는 없었다. 전체 합계점수 또한 두 집단이 유의한 차이가 없어 두 집단이 동질집단임이 확인되었다.

4.3. 컴퓨팅 사고력 사후검사

두 집단에 교육프로그램을 적용한 후 실험집단과 비교집단의 컴퓨팅 사고력 각각의 하위요소들의 변화가 두 집단 간 통계적으로 유의한 차이를 보이는지 알아보기 위해 독립표본 t검정을 실시하였다. 그 결과는 <table 9>과 같다.

<Table 9> Post-test result

Subscales	Group	Descriptive Statistics			t	p
		n	M	SD		
Sequential structure	Experimental	23	1.70	.559	2.210	.032*
	comparison	24	1.33	.565		
Algorithm	Experimental	23	.35	.647	-.162	.872
	comparison	24	.38	.495		
Conditional branch	Experimental	23	2.52	.665	3.295	.002*
	comparison	24	1.88	.680		
Variable	Experimental	23	1.26	.619	-1.153	.879
	comparison	24	1.29	.751		
Function	Experimental	23	2.39	1.901	-.122	.903
	comparison	24	2.46	1.865		
Recursion	Experimental	23	.48	.790	-1.024	.311
	comparison	24	.71	.751		
Total	Experimental	23	8.87	3.958	.885	.381
	comparison	24	7.92	3.412		

*p<.05

교육프로그램 적용 후 실험집단과 비교집단의 합계 평균 점수는 각각 8.87(±3.958)과 7.98(±3.412)로 실험집단의 점수가 높게 나타났다. 그 하위영역 중에서도 특히 순차구조 영역에서 실험집단과 비교집단의 점수가 각각 1.70(±0.559)과 1.33(±0.565)으로 실험집단이 높았다. 또한 조건분기 영역에서 실험집단과 비교집단의 점수가 각각 2.52(±0.665)과 1.88(±0.680)로 실험집단의 점수가 높게 나왔으며, 순차구조, 조건분기 두 영역 모두 통계적으로 유의미한 차이를 보였다.

따라서 본 연구에서 개발된 수행시간 분석 중심의 언플러그드 교육프로그램이 보드게임을 활용한 소프트웨어교육에 비해 순차구조, 조건분기 그리고 전체적인 컴퓨팅사고력에 긍정적인 영향을 주는 것으로 확인되었다.

5. 결론 및 제언

본 연구에서는 초등학교 3학년 학생들에게 수행시간 분석에 초점을 맞춘 언플러그드 교육을 실시하였고, 이 교육이 학생들의 컴퓨팅사고력에 어떤 영향을 미치는지 알아보았다. 방학 기간 중 6일 간의 집중 교육을 실시한 후 보드게임 교육을 받은 통제집단과 본 연구에서 개발한 교육프로그램을 적용한 집단을 비교분석하였다. 계산적 사고력과 창의성 사전·사후 검사를 실시한 결과

본 연구에서 개발한 교육 프로그램은 초등학생의 컴퓨팅사고력 향상에 효과적인 것으로 나타났다.

다만, 본연구의 집단은 상관연구에 필요한 30명을 전부 확보하지 못하였다. 연구 결과를 일반화 하는 데에는 한계가 있다. 또한 교육기부 프로그램을 지원한 학생들을 대상으로 연구를 진행하여 연구 대상자가 SW교육에 관심이 높은 학생들이 모집되었을 가능성이 있다. 추후의 연구에서는 30명의 연구 대상을 확보하고, 모집 대상들의 SW교육에 대한 관심도를 분석하여 연구결과를 일반화할 필요가 있다.

참고문헌

- [1] Na Jung Eun (2015), Computational Thinking Curriculum Development, *Korean Association of general Education conference, 2015*(11), 161-166.
- [2] Yongmin Kim·Jonghoom Kim (2017) / Effect of data visualization using scratch on improvement of creativity of preliminary coding instructors.
- [3] Kim Soo-Jin (2018) / Design and Effectiveness Evaluation of Curriculum for Programming Education in Lower Grade of Primary Schools.
- [4] Park Namsu (2018) / Identifying Impact Factors on Computational Thinking in the Game-based Preschooler software Education.
- [5] YoungUk Jeon·Byong-Rae Han (2018)/ The Effect of Unplugged Cooperative Learning Activies of the Cooperating Study Methods on the Improvement of Computational Thinking in the upper grades Elementary Students.
- [6] Koo Young Eun (2015) / An Effect of Unplugged Education based on Play Learning for Lower Grade in Elementary School.
- [7] Byoungrae Han (2013)/ The Research of Unplugged Computing Method for Computational Thinking in Elementary Informatics Education.
- [8] Byengsu Kim (2014) / Programming Education Program based on PPS to Improve Computational Thinking Ability.
- [9] Bell, T., Witten I., and Fellows, M.(2015). CS Unplugged : An enrichment and ex-tensionprogramme for primary-aged students.
- [10] Ministry of Education. (2015:2)
- [11] Minjin Jang(2017) / The effect of Unplugged Learning on Elementary School Student's Talent: Focus on Logical Thinking and Creative Propensity.
- [12] ISTE & CSTA(2011). Computational Thinking Leadersip toolkit 1st edition.
- [13] Wing, J. M. (2008). "Computational thinking and thinking about computing." *Philosophical Transactions of the Royal Society, vol. 366* July 2008. pp. 2717-3725.
- [14] Young-Jun Lee, Eun-Kyoung Lee(2008). The Nature and perspective of Informatics Education. *The Journal of Korean association of computer education 11*(3). pp. 1-11.

저자소개



김 정 아

2004 제주교육대학교 초등컴퓨터
교육(교육학사)
2008 제주대학교 교육대학원 컴퓨
터교육전공(교육학석사)
2012 제주대학교 일반대학원 컴퓨
터교육전공(박사과정 수료)
2019~현재 김녕초등학교 교사
관심분야 : 소프트웨어교육, 영재
교육
E-Mail : vadang@korea.kr



김 태 훈

2003 제주교육대학교 초등컴퓨터
교육(교육학사)
2010 제주대학교 교육대학원 컴퓨
터교육전공(교육학석사)
2015 제주대학교 일반대학원 컴퓨
터교육전공(교육학박사)
2019~현재 도남초등학교 교사
관심분야 : 컴퓨팅사고력, 컴퓨팅
융합교육
E-Mail : gtranu@gmail.com



오 민 우

2015 제주대학교 교육대학
초등컴퓨터교육(교육학학
사)
2015~현재 제주대학교 교육대학
원 컴퓨터교육전공 석사과정
2015~현재 서귀북초등학교 교사
관심분야 : SW교육, 언플러그드
E-Mail : minwoo4286@naver.com



김 종 훈

1998 홍익대학교 전자계산학과(이
학박사)
1998~1999 ETRI Post-Doc.
1999~현재 제주대학교 초등컴퓨
터교육전공 교수
관심분야 : 컴퓨터교육
E-Mail : jkim0858@jejunu.ac.kr



양 영 훈

2005. 2. 제주교육대학교 학사
2011. 2. 제주교육대학교 석사
2005~ 초등교사
현재 노형초등학교 교사
현재 제주대학교 컴퓨터교육 박사
과정
관심분야 : SW교육, 알고리즘
E-Mail : atriple19816@hanmail.net