



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

교육학 석사학위논문

초등학생의 창의적 문제해결력 향상을
위한 소프트웨어 교육 프로그램
효과 연구



2019년 8월

부경대학교 교육대학원

교육컨설팅전공

윤혜정

교육학석사학위논문

초등학생의 창의적 문제해결력 향상을
위한 소프트웨어 교육 프로그램
효과 연구

지도교수 허 군

이 논문을 교육학석사 학위논문으로 제출함.



2019년 8월

부경대학교 교육대학원

교육컨설팅전공

윤혜정

윤희정의 교육학석사 학위논문을 인준함.

2019년 8월 23일



주 심 교육학박사 원 호 현 (인)

위 원 교육학박사 조 진 호 (인)

위 원 교육학박사 허 균 (인)

목 차

Abstract	v
I. 서론	1
1. 연구의 목적 및 필요성	1
2. 연구문제	5
3. 용어의 정리	5
4. 연구의 제한점	7
II. 이론적 배경	8
1. 창의적 문제 해결력	8
2. 소프트웨어 교육	13
3. 선행 연구	24
III. 연구방법	29
1. 연구대상	29
2. 실험 설계 및 분석 방법	30
3. 검사도구	31
4. 연구절차 및 자료처리	32
IV. 연구결과	33
1. 소프트웨어 교육 프로그램 구성	33

2. 소프트웨어 교육 프로그램의 효과 검증	43
V. 논의 및 결론	46
1. 논의	46
2. 결론 및 제언	49
참고문헌	52
부록	
· 창의적 문제 해결력 검사지	58
국문초록	61



표 목 차

<표 I-1> 주요국가 소프트웨어 인재양성 사례	2
<표 I-2> 2015 교육과정 개편 내용	3
<표 II-1> 창의적 문제해결력 구성요소와 의미	12
<표 II-2> 초등학생의 소프트웨어 교육 목표	14
<표 II-3> 초등학생의 성취기준	14
<표 II-4> 초등 실과 - 내용 요소와 성취기준	16
<표 II-5> 소프트웨어 교육 교수학습 모델	18
<표 II-6> 시연중심모델(DMM) 교수학습 절차	19
<표 II-7> 재구성중심모델(UMC) 교수학습 절차	20
<표 II-8> 초등학생을 위한 교육용 프로그래밍 언어	21
<표 II-9> 엔트리와 스크래치의 특징	22
<표 II-10> 엔트리와 스크래치의 기능 차이	22
<표 III-1> 연구대상	29
<표 III-2> 실험 설계	30
<표 III-3> 하위요소별 문항번호	31
<표 III-4> 연구의 진행 절차	32
<표 IV-1> 시연중심모델 교수학습 과정안 예시	35
<표 IV-2> 재구성중심모델 교수학습 과정안 예시	36
<표 IV-3> 교수-학습 지도계획	37
<표 IV-4> 차시별 학습 목표와 학습 내용 (기초, 이해)	38
<표 IV-5> 차시별 학습 목표와 학습 내용 (발전, 응용)	39
<표 IV-6> 11~12차시 상세 지도안 (엔트리 기준, 2차시 연강)	40

<표 IV-7> 창의적 문제해결력 사전·사후 검사	43
<표 IV-8> 성별의 창의적 문제해결력 차이	44
<표 IV-9> 프로그램 도구에 따른 창의적 문제해결력 차이	45



Effect of Software Education Program to Improve Creative Problem Solving
in Elementary School Student

Hye Jung Yun

Education consulting
Graduate School, Pukyong National University

Abstract

The purpose of this study is to propose that software education programs are used to improve creative problem solving of elementary school students by analyzing their effectiveness.

For this purpose, the theory of creative problem solving and software education program were reviewed. Based on this, the following research questions were chosen.

First, how should the 'Software education program for improving creative problem solving' of elementary school students be organized and conducted?

Second, does participation in software education programs affect creative problem-solving?

Third, is there any gender difference to creative problem-solving after software education?

Fourth, is there any difference in creative problem-solving in the educational programming language (scratch and entry)?

In this study, software education was conducted from October 2018 to February 2019 for two classes of G Elementary School in Gijang-gun, Busan. I used a different educational programming language (Scratch and Entry) in each class. To measure creative problem solving, a pre- and post examination form, developed by the MI team of Seoul National University Psychological Research Center in 2004, was used.

Verify the effectiveness, we investigated the pre-post changes in creative problem solving, the differences between sexes, and the differences in the educational programming languages (scratch and entry). The statistical processing of the data was carried out by verifying independent sample t-Test and paired sample t-Test as a pre-post Test.

The following is the summary of this study results.

First, significant differences emerged in the divergent thinking area, a sub-area of creative problem solving, after software education.

Second, there were significant gender differences in creative problem-solving.

Third, programming languages difference indicated that using the Entry in the class had a more positive effect on the motivational factors, but there were no significant differences in other specific thinking, divergent thinking or critical thinking.

By analyzing the effectiveness of the software education program using Scratch and Entry, the representative program of block coding, we can see that software education works in the divergent thinking area and that the effectiveness varies in the motivational elements depending on the programming tool. Therefore, before educating the software program, the curriculum should be organized after sufficiently discussing the ways to enhance students' creative problem-solving.

The results of this study show that software education programs are useful as a tool for improving creative problem solving.

I. 서론

1. 연구의 목적 및 필요성

1784년 영국에서 시작된 증기기관을 이용한 ‘기계적 혁명’은 1차 산업혁명, 1870년 전기를 이용한 대량생산의 시작인 2차 산업혁명, 컴퓨터와 인터넷의 정보화 시대로 디지털 혁명인 3차 산업혁명이 있었다면 이제는 로봇, 인공지능을 통해 실제와 가상이 통합되는 시대인 4차 산업혁명이 시작되었다. 4차 산업혁명이 시작되면서 우리 주변의 모든 사물에 인터넷과 컴퓨터가 연결되는 사물인터넷(IoT) 환경으로 변화되어 가고 있다. 이러한 변화는 컴퓨터와 소프트웨어에 대한 이해가 필수인 시대가 도래됨을 알 수 있다.

새로운 시대에 알맞은 인재를 양성하기 위해 세계 각국에서는 소프트웨어 교육의 다양한 교육적 제도와 환경을 준비하고 있다. 이스라엘은 1994년부터 정규교과에 포함하여 운영하고 있으며, 미국은 2016년 1월 신년 국정 연설에서 오바마 대통령이 “초중등 모든 학생을 위한 컴퓨터과학 교육 정책을 추진하겠다.”고 발표하였다. 영국은 2014년 9월부터 초·중등학교에서 소프트웨어 교육을 필수로 편성하여 운영하고, 프랑스도 2016년 9월부터 소프트웨어 교육을 정규과목으로 편성하였다.

국내에서는 미래창조과학부가 글로벌 창의융합형 인재를 길러내는 방향으로 ‘소프트웨어 중심 사회를 위한 인재 양성 추진 계획’을 발표하였다. 미래창조과학부에서 말하는 소프트웨어 중심 사회란 소프트웨어가 혁신의 성장, 가치 창출의 중심이 되고 개인, 기업, 국가의 경쟁력이 좌우하는 사

회이며, 급변하는 소프트웨어 중심 사회에서 미래형 창의인재 양성 전략의 추진과제는 초·중등 SW 교육의 활성화이다(미래창조과학부, 2014; 교육부, 2015).

<표 1-1> 주요국가 소프트웨어 인재양성 사례

국가	년도	주요 내용
이스라엘	1994년 2011년	SW 정규과목, 고등학생 이과 3년간 SW 수업 중학교 컴퓨터과학 과정 개발 및 운영
중국	2001년 2003년 2005년	초등학교 정보통신 과목 필수 지정 중학교 정보통신 과목 필수 지정 고등학교 정보통신 과목 필수 지정
미국	2014년 2015년 2016년	code.org 가입자 3,700만 명 30개 교육청에서 정보과학을 졸업학점 인정 과목 지정 AP 코스 'Computational Thinking' 과목 실시 결정
일본	2009년 2011년 2012년	소프트웨어 교육 강화 - 유치원 필수과목 지정 초등학교 필수과목 지정 중고등학교 '정보' 필수과목으로 지정
인도	2010년	초중등학교 '컴퓨터과학' 필수 지정 교육
영국	2014년	컴퓨터 과목을 5~16세 필수과목으로 지정 초중고교 매주 1시간 이상 컴퓨터 과목 교육
프랑스	2016년	중학교 정규 과목

출처 : 미래창조과학부, 2015 재구성

2015 개정 교육과정의 경우 초·중등학교의 정규 교육과정에 소프트웨어 교육을 필수로 이수할 수 있도록 2019년도부터 초등학교 5, 6학년 실과시간에 17시간 이상, 중학교는 2018년도부터 정보 교과 34시간을 학생들이 필수적으로 이수하도록 소프트웨어 '코딩교육'을 의무화하였다. 개편된 내용은 초등학교는 실과과목의 ICT 활용 단원을 소프트웨어 기초소양 단원

으로 구성하여 17시간 이상 교육하고, 중학교에서는 ‘과학/기술·가정/정보’ 교과를 새로이 신설하고, ‘정보’ 과목을 개편하여 소프트웨어 교육 중심의 필수 과정으로 34시간 이상 교육하며, 고등학교에서는 심화 선택과목의 정보 교과가 일반선택으로 바뀌었다(교육부, 2015).

<표 1-2> 2015 교육과정 개편 내용

구분	현행	개편안	주요 개편 방향
초등학교 (19년~)	실과 ICT 단원 12시간	실과 SW 교육 17시간 이상	문제해결 과정, 알고리즘, 프로그래밍 체험 정보윤리 의식 함양
중학교 (18년~)	정보 선택	정보 34시간 필수	컴퓨팅사고 기반 문제해결 실시 간단한 알고리즘, 프로그래밍 개발
고등학교 (18년~)	정보 심화 선택	정보 일반 선택	다양한 분야와 융합하여 알고리즘 프로그램 설계

출처 : 교육부, 2015

개정된 교육과정으로 인하여 코딩교육과 창의적 문제 해결력, 컴퓨팅 사고력에 대한 연구가 진행되고 있으며, 교육부에서는 소프트웨어 선도 학교, 연구학교 등을 지정하여 소프트웨어 교육 프로그램 개발과 적용을 통해 교수·학습 방법에 대한 연구, 효과성 검증이 이루어지고 있다. 박정미(2013)는 이미 많은 연구를 통하여 프로그래밍 과정이 문제해결력과 논리적 사고력, 창의력 신장 등에 있어서 교육적 효과성이 입증되었기에 프로그래밍 교육 자체만으로도 큰 교육적 의미가 있으며, 이러한 창의적인 프로그래밍 학습이 단순히 학습자들을 프로그래머로 만드는 것이 아니라 계산적 사고자로 발전시키는 것이라고 하였다. 계산적 사고라는 단어는 Wing(2006)에

의해 처음으로 소개되었으며 국내에서는 ‘컴퓨터적 사고’, ‘정보 과학적 사고’라고 번역하기도 한다. 또한, 소프트웨어 교육을 통해 학생들이 소프트웨어를 이해하기 위하여 스스로 프로그램을 작성하고 즉각적으로 실행 결과를 알아보면서 오류 수정을 통해 학생들의 문제해결력과 창의력을 향상할 수 있다고 한다(이은경, 2013). 김은탁(2018)은 교육용 프로그래밍 언어에서 그래픽 기반 언어인 스크래치의 경우 학생들이 자유롭게 가지고 놀 수 있을 정도로 호기심과 흥미를 유발하기에 다른 교과 내용 요소를 접목해 문제 해결 형식의 통합 수업을 한다면 자연스럽게 몰입을 경험하면서 문제해결력 향상에도 도움이 될 것이라고 하였다.

본 연구는 창의적 문제해결력을 위한 소프트웨어 교육 프로그램을 2015 개정 교육과정의 성취기준에 맞추어 구성하여 6학년을 대상으로 교육용 프로그래밍 언어 중에서 현재 가장 많이 사용되는 스크래치와 엔트리로 하여 소프트웨어 교육을 진행한 뒤 초등학생들의 창의적 문제 해결력 변화에 대해 알아보고자 한다. 나아가 향후 초등학생들의 창의적 문제 해결력 향상을 위한 소프트웨어 교육 프로그램 개발 및 교수·학습 방법 개선에 기여할 수 있기를 바란다.

2. 연구 문제

본 연구의 연구 문제는 다음과 같다.

첫째, 초등학생의 ‘창의적 문제해결력 향상을 위한 소프트웨어 교육 프로그램’은 어떻게 구성되고 진행되어야 하는가?

둘째, 소프트웨어 교육 프로그램의 참여는 창의적 문제해결력에 영향을 미치는가?

셋째, 소프트웨어 교육 프로그램은 성별에 따라 창의적 문제해결력의 차이가 있는가?

넷째, 교육용 프로그래밍 언어(스크래치와 엔트리)에 따른 창의적 문제해결력에 차이가 있는가?

3. 용어의 정의

가. 소프트웨어 교육

소프트웨어 교육은 정보과학 교육, 컴퓨터과학 교육, 알고리즘 교육, 프로그래밍 교육, 코딩 교육 등과 같은 용어로 혼용되어 정의되고 있다.

본 연구에서는 소프트웨어 교육이란 한글·엑셀 등 컴퓨터를 단순히 활

용하는 방법을 익히는 교육이 아니라 다양한 문제의 해결방법을 찾기 위해 ‘컴퓨터’를 기반으로 자료를 수집·분석하고, 문제의 효율적 해결 과정 등을 창조하는 일련의 사고력 교육으로 정의하고자 한다(김우찬, 2016).

나. 코딩

코딩이란 컴퓨터 프로그래밍과 같은 의미의 단어로 사람이 필요로 하는 프로그램을 만들기 위하여 C언어, 파이썬, 자바, 델파이 등의 컴퓨터 프로그래밍 언어를 이용해서 코드를 작성하는 것이다(EBS 스토리, 2017).

다. 교육용 프로그래밍 언어

교육용 프로그래밍 언어(Educational Programming Language)는 학생들이 프로그래밍 언어를 좀 더 쉽게 배울 수 있도록 교육용으로 만들어진 프로그래밍 언어이다. 교육용 프로그래밍 언어는 알고리즘적인 사고와 문제 해결력을 키우기 위한 학습의 목적을 가지므로 프로그래밍 언어 자체가 단순화되어야 프로그래밍 교육에 쉽게 활용되고, 궁극적인 교육의 목표를 달성할 수 있다(박주연, 2015).

소프트웨어 교육을 위해 다양하게 개발된 교육용 프로그래밍 언어 중에서 2007년에 인텔의 지원을 받아 미국 MIT 공대 미디어연구실과 UCLA의 연구원들이 개발된 스크래치(Scratch)는 레고처럼 블록 형태의 명령어들을 연결하는 블록 기반 프로그램 언어이다. 엔트리교육연구소에서 개발한 엔트리(Entry)도 스크래치와 같은 블록 기반 프로그램 언어로 초등학생들에게 적합한 환경과 다양한 콘텐츠를 제공하고 있다(박영순, 2016). 스크래치, 엔트리와 같은 블록 기반 프로그램은 텍스트에 대한 부담을 덜어줄 뿐만 아니라 다양한 멀티미디어 환경을 제공하여 웹을 통한 공유기능까지 있어

교육용 프로그래밍 언어로 많이 활용되고 있다.

라. 창의적 문제 해결력

창의적 문제해결 학습이라는 용어는 곧 창의적 문제해결력을 높이기 위한 학습 과정이라는 의미를 내포하며 비구조화된 문제 상황을 파악하여 새로운 산출물이나 해결책을 만들어내는 능력으로 지식과 기능을 바탕으로 확산적 사고와 논리·비판적 사고가 상호 작용하는 사고 과정을 창의적 문제해결(Creative Problem Solving)이라고 정의한다(황지현, 2010).

창의적 문제해결력이란 제시되는 비구조화된 문제를 정확하게 파악하여 문제를 해결하는 과정에서 여러 요인이 복합적이며 역동적으로 상호작용하는 사고 과정을 통하여 독창적인 방법 또는 해결방안을 만들어 내는 능력을 말한다(김우찬, 2016; 김효성, 2016).

4. 연구의 제한점

본 연구의 제한점은 다음 내용과 같다.

첫째, 소프트웨어 교육 프로그램에 참여하는 교육 대상은 부산시 기장군 G초등학교 6학년 2개 반으로 한정하였다. 그러므로 본 연구 결과를 일반화시키는 데는 한계가 있다.

둘째, 교육용 프로그래밍 언어 중에서 스크래치와 엔트리로 한정하여 효과연구를 하였기에 모든 교육용 프로그래밍 언어를 창의적인 문제해결력 향상을 위한 것으로 일반화하여 해석하기에는 한계가 있다.

셋째, 학습자의 학습 발달 단계와 여러 환경적 요인에 따라 연구의 결과가 차이날 수 있다.

II. 이론적 배경

1. 창의적 문제 해결력

가. 개념

일반적인 문제해결과 창의적인 문제해결의 차이는 문제해결의 과정에서 창의성 요인들이 발휘되는 것이라고 한다.

학자마다 창의성의 정의에 대한 개념과 의견이 다르다. Getzels & Jackson(1962)은 “창의성은 기존의 아이디어, 산출물, 기술, 접근 방법을 새로운 방식으로 결합하는 능력과 새롭고 유용한 산출물, 확산적이고 풍부한 사고 과정, 고양되는 내재적인 주관적 경험이다”라고 설명하였다. 창의성이란 매우 복잡적이고 다면적이므로 학자마다 정의가 다르면서도 독창적이며 유용한 산출물을 만들어 내는 것이라는 의견은 같다.

Gagne(1985)는 문제해결을 “학습한 규칙을 응용하여 새로운 문제들에 대한 해결책을 발견하는 것”이라 정의하고, Mayer(1999)는 “해결방법이 명확히 보이지 않을 때 하나의 목적을 성취하려는 인지 과정”이라고 정의하였다.

학자들이 정의하는 창의적 문제해결력은 다음과 같다. 현실에 존재하는 둘 이상의 아이디어를 모두 합하고 다시 정리하고 적용하는 과정이다 (Dewey, 1910). 학생들의 실생활에서 발생하는 여러 가지 문제를 해결하기 위하여 문제 특성을 분석하고, 요소 간의 상관관계를 논리적으로 파악하여

해결책을 계획하며 분석하는 과정에서 여러 친구와 협동하여 아이디어를 공유하여 다양한 해결방법을 발견하게 되면서 문제해결력이 향상되고 더불어 팀별 과제 수행으로 사회적으로 필요한 소통의 능력도 키울 수 있을 것이다(김혜진, 2016). 창의적 문제해결력을 Treffinger & Isaksen (1985)는 “문제를 이해하고 아이디어 산출한 뒤 행동계획 및 실행의 3단계를 수행하면서 확산적 사고 및 수렴적 사고가 작용하여 창의적이고 생산적인 사고가 일어나는 문제해결의 과정에서 요구되는 능력”이라고 하였다. 조석희, 한국교육개발원(2001)에서는 “창의적 문제해결력이란 문제해결 과정에서 다양한 요인이 복합적이며 역동적으로 상호작용하여 문제 해결에 유용하며 독창적인 산출물 혹은 해결책을 만들어 내는 것”으로 정의하였다. 여기서 다양한 요인이란 창의적 문제해결력 구성요소의 6가지 영역으로 일반적인 영역의 지식-기능기반, 특정한 영역의 지식-기능기반, 확산적 사고, 비판적 사고, 동기적 요소를 말한다. 문제를 해결하는 과정에서 다양한 요인이 역동적이고 복합적으로 상호작용하여 해결방안이나 산출물을 만들어 내는 능력으로 정의하면서 다양한 연구를 통해 학습 내용, 환경, 학습 과정의 상호작용이 창의적 문제 해결력에 중요함을 강조하였다(조석희, 장영숙, 정태희, 2003; 배영권, 2006). 일상생활에서 복잡하고 애매모호한 상황 같은 문제를 해결하기 위하여 지식과 기능을 바탕으로 논리·비판적 사고와 확산적 사고와 같은 고등사고력을 이용하여 문제 상황을 파악하고 풀어나가는 모든 과정에 독창적이고 창의적인 사고를 지원하는 능력과 유용한 산출물을 만들어 내는 것을 창의적 문제해결이라고도 하였다(황지현, 2010; 김우찬, 2016).

창의적 문제해결력을 향상시키기 위해 교사는 학생들에게 창의적 문제해결 향상에 적합한 구조화되지 않은 문제를 제시하고 다양한 사고능력을 발휘하고 향상시킬 수 있는 학습과정과 내용을 함께 제공해 주어야 한다. 더

불어 학생들의 내재적 동기화를 강화하고, 자기 효능감을 높여주는 환경도 조성하여 동기적 요소를 높여야 한다(김효성, 2016).

나. 구성 요소

한국교육개발원에서 편찬한 연구보고서(2002)에서는 창의적 문제해결력의 구성요소를 다음과 같이 다섯 가지로 구분하고 있다(조석희, 한국교육개발원, 2002).

첫째, 일반적 영역의 지식과 기능이다. 지식은 주어진 문제와 직접적인 관계가 없는 일반적인 지식이고, 기능은 문제해결에 도움이 되는 일반적 사고 전략, 기억전략, 메타인지 전략, 문제해결 전략 등이 있다. 이 요소는 모든 교과 영역에 걸쳐서 학습자가 창의적 문제해결을 할 때 활용하는 지식과 기능이다.

둘째, 특정 영역의 지식과 기능기반이다. 문제해결에 직접적인 관계가 있는 영역의 지식과 기능기반으로 창의적 문제해결력을 위해 특정 영역에서 새롭고 뛰어난 아이디어를 만들어내야 하므로 필요한 요소이다. 반드시 학습할 것과 학습해두면 좋을 것을 구분하여 학습의 우선순위를 정해야 한다. 따라서, 교사는 풍부하고 흥미로운 학습 환경을 제공하여 학습자들에게 강한 학습 동기를 불러일으키도록 하여 특정 영역의 지식과 기능이 발휘되도록 하여야 한다.

셋째, 확산적 사고이다. 주어진 문제 해결에 있어 다양하고 새로운 대안들을 찾아내려는 노력으로 갈등, 긴장, 모순, 호기심, 아이디어, 애매모호함에 대한 개방성, 모험하기, 상상과 유머, 결정적 요소 찾아서 건설적인 해결책 찾아내기 등이 있다. 교사나 학부모는 일상생활에서 접할 수 있는 구조화되지 않은 문제, 개방적인 문제를 제공하여 학습자가 이에 대한 최적의 문제해결을 할 수 있도록 격려하고, 학습자들에게 확산적사고 전략을

자유롭게 발휘하는 안정된 학습 환경을 조성해 주어야 한다. 확산적 사고 전략으로는 브레인스토밍, 형태학적 분석법, 속성열거법, 체크리스트(스캔 퍼), 시네틱스, 마인드맵 등이 있다.

넷째, 비판적 사고이다. 정은영(2008)은 “비판적 사고를 신장하려면 옳고 그름을 찾아내고, 상황, 사물, 인과관계를 따질 기회를 자주 제공해야 한다.”고 하였다. 의문시에 문제를 찾고 탐색하는 행동, 문제해결을 위한 사고 등을 격려해주고 긍정적으로 반응하며 학습자가 문제해결을 하기 위한 시간을 충분히 주어야 한다고 하였다. 학습자가 문제해결을 하도록 돕기 옳고 그름을 판단하기 전에 다음 단계의 사고를 할 수 있도록 유도하는 질의를 하는 것이 더 깊이 있는 사고를 끌어낼 수 있다. 이러한 비판적 사고력을 향상하기 위한 사고 전략에는 PMI, 쌍비교분석법, 역브레인스토밍, 하이라이팅기법, 평가행렬법 등이 있다.

다섯째, 동기적 요소이다. 동기는 여러 가지 지식과 기능의 습득을 위한 원동력을 제공한다. 이는 창의적 문제 해결자의 자기 효능감과 내재적 동기화를 말한다. 동기는 환경의 영향을 많이 받기 때문에 학습자에게 적절한 학습 환경을 제공해야 하며 학습자의 수준에 맞게 학습하도록 하여 학습자의 흥미와 동기가 높아지도록 해야 한다.

이러한 여러 요소들이 서로 역동적으로 상호작용할 때 창의적으로 문제를 해결할 수 있다. 그래서 본 연구에서는 소프트웨어 교육의 효과 분석으로 한국교육개발원에서 편찬한 창의력 문제해결력 검사 도구를 활용하기로 하였다.

위의 창의적 문제해결력의 구성요소와 의미는 다음과 같이 요약하였다 (조석희, 한국교육개발원, 2001; 황지현, 2010; 김혜진, 2016; 김우찬, 2016).

<표 II-1> 창의적 문제해결력 구성요소와 의미

구성요소	의미
일반 영역의 지식, 기능	모든 영역에 걸쳐 영향을 주는 여러 가지 다양한 상황에서 적용되는 지식과 기능
특정 영역의 지식, 기능	그 영역에서 문제해결과 직접적인 관련 있는 원리, 개념, 기본적 아이디어를 포함하는 지식과 이들의 관계를 탐구하는 양식을 모두 포함하는 기능 기반
확산적 사고	문제 발견하여 정의하고, 독창적 해결책 또는 대안을 마련하는데 기여하는 호기심, 사고기능, 여러 가지 아이디어, 애매모호함에 대한 개방성 등
논리-비판적 사고	문제 명료화하고, 가능한 방안들의 타당성 평가함으로써 문제 해결책을 구체화하는 데 필요로 하는 사고기능으로 수렴적 사고라고도 한다.
동기적 요소	정의적 요소 또는 개인이 가진 성격을 포함하는 개념으로 내재적 동기화, 자기 효능감, 자기조절요소 포함

출처 : 조석희, 한국교육개발원, 2001

2. 소프트웨어 교육

컴퓨터가 어떤 기능을 수행할 수 있도록 지시하는 명령어들을 순서에 따라 차례로 나열해 놓은 것을 프로그램이라 한다. 이러한 명령어를 이용하여 프로그램을 작성하는 것을 프로그래밍이라고 한다(김대수, 2017).

교육부가 2015 개정 교육과정에서 “소프트웨어 교육이란 소프트웨어의 기본 개념과 원리 및 기술의 이해를 바탕으로 실생활 및 다양한 학문 분야의 문제를 창의적이고 효율적으로 해결하는 컴퓨팅 사고력 함양을 위한 교육”으로 제시하였으며, 전용주(2017)에서는 “소프트웨어의 기본 개념과 원리 및 기술의 이해를 바탕으로 실생활 및 다양한 학문 분야의 문제를 창의적이고 효율적으로 해결하는 역량과 태도를 함양하는 교육”이라는 의미로 재정의하였다.

가. 2015 개정 교육과정의 소프트웨어 교육 이해

세계적으로 SW 교육이 강화되고 있는 가운데 우리나라는 컴퓨팅 사고력을 기반으로 문제해결역량을 함양하는 SW 교육을 강화하기 위해 ‘소프트웨어 운영지침’과 ‘2015 개정 교육과정 실과/정보과 각론’이 발표되었다(교육부, 2015).

소프트웨어교육 운영지침에서 초등학생의 소프트웨어 교육 목표와 초등학생의 성취 기준은 다음의 표와 같다.

<표 II-2> 초등학생의 소프트웨어 교육 목표

영역	교육목표	내용 요소
생활과 소프트웨어	소프트웨어가 가져온 생활의 변화 알기 정보사회에 필요한 건전한 태도와 의식을 가진다.	나와 소프트웨어 - 소프트웨어와 생활 변화
		정보 윤리 - 사이버 공간에서 예절 - 인터넷 중독, 예방 - 개인정보 보호 - 저작권 보호
알고리즘과 프로그래밍	알고리즘과 프로그래밍을 체험하여 실생활의 다양한 문제를 컴퓨팅 사고로 이해할 수 있다.	문제 해결 과정 체험 - 문제 이해와 구조화 - 문제 해결의 방법 탐색
		알고리즘의 체험 - 알고리즘의 개념과 체험
		프로그래밍의 체험 - 프로그래밍 이해와 체험

출처 : 교육부, 2015

<표 II-3> 초등학생의 성취기준

영역	중영역	내용 요소	성취기준
생활과 소프트웨어	나와 소프트웨어	소프트웨어와 생활 변화	소프트웨어로 인한 생활 모습의 변화를 설명할 수 있다.
	정보윤리	사이버 공간 예절	사이버 공간에서 지켜야 하는 예절을 알고 실천할 수 있다.
		인터넷 중독과 예방	게임과 인터넷 중독의 문제점을 알고 예방 방법에 대해 설명할 수 있다.
		개인 정보 보호	개인 정보가 중요한 이유, 지킬 방법 찾아 발표할 수 있다.
		저작권 보호	생활 속에서 저작권 보호를 받는 것은 어떤 것인지 찾아보고, 보호하는 방법 설명할 수 있다.

영역	중영역	내용 요소	성취기준
알고리즘과 프로그래밍	문제해결과정 의 체험	문제의 이해와 구조화	주어진 문제를 이해할 수 있다.
			주어진 문제를 단순화할 수 있다.
		문제 해결 방법 탐색	문제 해결 방법을 순서에 따라 설명 할 수 있다.
			제안한 문제 해결 방법의 문제점, 개 선된 방법에 대해 설명할 수 있다.
	알고리즘의 체험	알고리즘의 개념	알고리즘 개념을 이해할 수 있다.
		알고리즘의 체험	순차, 반복, 선택 구조를 이용하여 문제해결 절차를 그림, 기호 이용하 여 표현할 수 있다.
			간단한 알고리즘(탐색, 정렬)을 체험 활동을 통해 이해할 수 있다.
	프로그래밍의 체험	프로그래밍의 이해	프로그래밍 언어의 기본적인 요소를 알 수 있다.
		프로그래밍의 체험	제시된 프로그램을 동일하게 만들 수 있다.
			제시된 프로그램 수정하여 자기만의 프로그램으로 만들 수 있다.
			내가 생각한 간단한 프로그램을 만 들 수 있다.

출처 : 교육부, 2015

나. 2015 개정 실과/정보과 교육과정

2015 개정 교육과정에서 소프트웨어 교육은 초등학교 실과 교과에서 17시간, 중학교에서는 정보 교과가 필수로 개편되어 34시간으로 개편되었다.

2015 개정 교육과정에 따르면 초등학교에서는 기존 ICT 활용 중심의 정보 단원에서 SW 기초 소양 중심의 대단원으로 구성하여, 문제해결 과정과 알고리즘 및 프로그래밍 체험을 중심으로 저작권 보호 등 정보윤리 의식 함양을 포함하는 내용으로 교육하기로 하였다(교육부, 2015).

<표 II-4> 초등 실과 - 내용 요소와 성취기준

내용 요소	성취기준	성취기준해설
소프트웨어의 이해	[6설04-07] 소프트웨어가 적용된 사례 찾아보고 우리 생활에 미치는 영향을 이해한다.	컴퓨터뿐만 아니라 우리 생활 주변에 모든 가전제품에 사용된 소프트웨어를 탐색해 보고 우리 생활에 미치는 영향을 이해한다.
절차적 문제해결	[6설04-08] 절차적 사고에 의한 문제의 해결 순서를 생각하고 적용한다.	절차적 사고의 의미를 이해하고 일상생활에서 사례를 찾아보고 절차적 사고 과정을 통해 문제 해결에 적용한다.
프로그래밍 요소와 구조	[6설04-09] 프로그래밍 도구 사용하여 기초적인 프로그래밍 과정을 체험한다.	블록형 교육용 프로그래밍 도구를 활용하여 프로그래밍 과정을 체험하고 간단한 프로그램을 만들어 본다.
	[6설04-10] 자료 입력하고 필요한 처리 수행 후 결과를 출력하는 단순 프로그램을 설계한다.	수치를 입력하여 계산의 결과를 출력하거나, 여러 개의 문자열을 입력하여 문자열을 서로 연결한 결과를 출력하는 프로그램을 만들어 봄으로써, 소프트웨어의 입력, 처리, 출력 과정을 이해한다.
	[6설04-11] 문제 해결하는 프로그램을 만드는 과정에서 순차, 선택, 반복 등의 구조를 이해한다.	차례대로 수행되는 명령문은 '순차', 조건에 따라 선택적으로 명령을 수행하는 것은 '선택', 주어진 명령문에 맞게 여러 번 반복하는 '반복'이 프로그램의 3가지 구조이다. 이러한 구조를 프로그램을 만드는 과정에서 이해한다.

출처 : 교육부, 2015

2015 개정 실과 교육과정의 소프트웨어교육과 관련된 성취기준에서 학습 요소, 교수·학습 방법 및 유의사항을 다음과 같이 제시하고 있다.

첫째, 응용 소프트웨어 사용법이나 프로그래밍 언어의 문법 학습을 최소화하고, 문제해결에 필요한 프로그래밍을 통한 컴퓨팅 사고력 신장에 초점을 맞춘다.

둘째, 절차적 사고를 적용할 수 있는 일상생활 속의 사례들을 찾아보고, 놀이 중심의 신체 활동, 퍼즐 등의 다양한 활동을 통해 절차적인 문제해결 과정을 이해하도록 한다.

셋째, 실생활 속에서 일어나는 문제 상황을 중심으로 학생들이 컴퓨팅 사고를 활용할 수 있도록 지도한다.

넷째, 컴퓨터를 활용한 활동 이외에도 컴퓨터 없이 문제를 해결할 방법과 절차를 이해할 수 있도록 지도한다.

다섯째, 언플러그드 활동 시 학습과 놀이가 동시에 이루어질 수 있도록 시간과 내용을 적절히 구성하여 지도한다.

여섯째, 컴퓨팅 사고는 소프트웨어교육에 국한되는 것이 아니므로 국어, 사회, 수학, 과학 등 다양한 교과에서도 반영하여 지도한다(교육부, 2015).

다. 소프트웨어 교육 교수학습모델

한국교육개발원, 한국교육학술정보원(2015)에서 개발한 ‘SW 교육 교수학습 모델 개발 연구’에서 제안된 5가지의 SW 교육 교수학습모델을 본 연구의 기반으로 활용하였다.

<표 II-5> 소프트웨어 교육 교수학습 모델

구 분	절차	설명
시연중심모델 (DMM 모델)	시연 (Demonstration)	교사의 설명과 시범, 표준모델 제시
	모방 (Modeling)	학생 모방하기, 질문과 대답
	제작 (Making)	단계적이고 독립적 연습과 반복활동을 통한 기능 습득
재구성 중심모델 (UMC 모델)	놀이(Use)	학습자 체험 활동, 관찰과 탐색
	수정(Modify)	교사가 의도적으로 모듈 및 알고리즘을 변형하여 제시
	재구성(reCreate)	놀이, 수정 활동을 확장하여 자기만의 프로그램 설계/제작
디자인 중심모델 (NDIS 모델)	요구분석 (Needs)	주어진 문제에 대한 고찰과 사용자 중심의 요구 분석
	디자인(Design)	분해와 패턴 찾기, 알고리즘의 설계
	구현 (Implementation)	프로그래밍, 피지컬 컴퓨팅으로 산출물 구현
	공유 (Share)	산출물의 공유와 피드백을 통해 자기성찰
CT요소 중심모델 (DPAA(P) 모델)	분해 (Decomposition)	컴퓨터가 해결 가능한 단위로 문제 분해
	패턴인식(Pattern Recognition)	반복되는 일정한 경향 및 규칙의 탐색
	추상화(Abstraction)	문제 단순화와 패턴인식으로 발견한 원리를 공식화
	알고리즘(Algorithm)	추상화된 핵심적인 원리를 절차적으로 구성
	프로그래밍 (Programming)	컴퓨터가 이해할 수 있는 언어로 구현/실행

출처 : 한국교육개발원, 한국교육학술정보원, 2015

위의 모델에서 본 연구는 초등학생의 인지발달 수준과 코딩교육의 초기 임을 고려하여 DMM모델과 UMC모델을 병행하며 수업하기로 하였다.

DMM모델은 직접교수법으로 Beriner & Rosenshine(1977)이 개발한 부진아 프로그램에서 처음 사용되었다. 이 직접교수법은 기본적 분할정복방식으로 전체를 여러 부분으로 나누어 각각의 부분들을 차례대로 학습하면서 전체를 이해할 수 있다고 가정한다. 실제적인 학습을 강조하여 학습 목표에 따른 과제를 할당하고, 학생 스스로 책임을 부여하고 그것을 완수하게 하는 수업 방식으로 진행한다. 교사가 모델이 되고 학생들이 교사를 통해 모방 학습하게 하려는 것이다. 교사 중심방법이지만 교수자의 수업 방법에 따라 학습자 중심도 가능하다(한국교육개발원, 한국교육학술정보원, 2015).

<표 II-6> 시연중심모델(DMM) 교수학습 절차

단계명	학습방법	세부 단계 설명
Demonstration (시연)	시범보이기 설명	가르치려고 하는 핵심전략, 기능을 교사가 시연하거나 설명을 통해 학생들에게 소개한다.
Modeling (모방)	따라하기 질문과 답변	교사가 시연한 내용을 학생들이 따라 실습한다. 실습 과정에서 질문과 답변을 통해 학습자들이 시연을 모방한다.
Making (제작)	만들기 반복 활동	시연, 모방 단계에서 배운 학습 내용을 토대로 학생들이 스스로 만들기 반복 진행하되 단계적이고 전체적인 활동을 학습자들이 전개한다.

재구성중심모델(UMC 모델)은 발견학습법에서 사용하는 여러 가지 사례를 중심으로 핵심 개념, 원리를 발견하고 제시된 사례의 수정, 재구성을 통하여 컴퓨팅 사고를 이끄는 모델이다. 동기유발로 놀이를 통해서 배우고자 하는 학습 모듈을 학생 스스로 탐색하고 사전에 준비된 모듈의 수정과정을 통하여 기능과 개념을 이해한다. 놀이 및 수정 활동과 연계된 재구성 활동으로 컴퓨팅 사고의 전반적인 능력을 이해하게 된다(한국교육개발원, 한국교육학술정보원, 2015).

재구성중심모델에는 놀이를 통한 수정 활동과 학습자의 능동적인 재구성 활동을 통해 새로운 프로젝트를 생성하는 MIT 미디어랩에서 주로 진행했던 Creative Computing 활동의 3가지 학습요소가 나타나 있다(Brennen, Balch & Chung, 2014).

<표 II-7> 재구성중심모델(UMC) 교수학습 절차

단계명	학습방법	세부 단계 설명
Use (놀이)	탐색, 놀이, 체험, 조작, 활용	학습 내용의 프로젝트 시연 또는 조작하면서 프로젝트를 이해하는 단계이다. 즉, 결과물을 먼저 가지고 놀아보게 하여 친숙해지도록 한다. 또는 직접 교수법으로 교사 시범에 따라 간단한 프로젝트를 제작하면서 작동해 보도록 한다.
Modify (수정)	추가 설계, 보완, 수정, 확장	간단히 제공된 프로젝트에 새로운 아이디어 추가 또는 내용 확장하여 설계한다. (새로운 스프라이트 추가 및 수정, 변수의 추가, 스테이지 확장 등)
reCreate (재구성)	재구성, 구현, 개발, 산출	학습한 기능, 내용 활용하여 재구성하고 확장된 프로그램을 설계하여 제작해본다.

라. 교육용 프로그래밍 언어

프로그래밍은 소프트웨어 교육의 핵심이 되는 교육으로 교육용 프로그래밍 언어는 일반적인 프로그래밍 언어에 비해 간단한 문법과 사용법을 갖추어 누구나 쉽게 프로그래밍을 하고 빠르게 결과를 확인할 수 있어야 한다.

초등학교의 소프트웨어 교육은 전문적 프로그래머를 기르기 위한 교육이 아니기에 이해하기 쉽고 즐겁게 교육을 받을 수 있도록 교육용 프로그래밍 언어를 활용하여 교육해야 한다. 이러한 교육용 프로그램 언어를 EPL이라고 한다(안경미, 2010).

EPL은 컴퓨터 사고를 이끄는 데 다른 어떤 방법보다 가장 훌륭한 도구로 현재 세계적으로 40여 종의 교육용 프로그래밍 언어가 개발되어 사용되고 있으며, 그중에서 초등학생들이 가장 활발하게 사용하고 있는 엔트리와 스크래치의 특징은 <표 II-9>와 같다(한선관, 류미영, 2016).

<표 II-8> 초등학생을 위한 교육용 프로그래밍 언어

이름	소개
엔트리 play_entry.com	엔트리연구소에서 개발된 블록형 프로그래밍 도구로 프로그래밍 교육에 특화되어 있고, 다양한 디바이스와 웹에서 활용 가능
스크래치 scratch.mit.edu	MIT에서 제작한 블록형 프로그래밍 도구로 세계적으로 가장 많이 사용되고 있으며, 다양한 운영체제에 지원
코듀 www.kodugamelab.com	마이크로소프트에서 만든 3차원 그래픽을 이용한 게임 제작 도구
앨리스 www.alice.org	게임과 애니메이션을 쉽게 만들 수 있는 3D 프로그래밍 환경

<표 II-9> 엔트리와 스크래치의 특징

언어	명령어	상용/공용	사용환경	다국어	개발국
엔트리	비주얼 블록	공용	웹 기반	한국어	국내
스크래치	비주얼 블록	공용	윈도즈/맥	다국어 한국어	국외

엔트리는 카이스트 출신의 학생들이 주축이 되어 만들었으며, 엔트리교육연구소가 2014년 10월 네이버에 인수되어 이후 비영리로 사업목적을 전환하여 운영하고 있다(이민영, 2016).

스크래치는 미국 매사추세츠공과대학(MIT)의 라이프롱킨더가든그룹(Lifelong Kindergarten Group) 미디어연구실(Media Lab)에서 운영하는 EPL이다. 2003년 개발 프로젝트로 출발하여, 세계 150여 국가에 40여개의 언어로 공개되었다. 미국과학재단, 스크래치재단, 구글, 레고재단, 인텔, 카툰네트워크, 레만재단, 맥아더재단에서 지원을 받았다(함성진, 2011).

<표 II-10> 엔트리와 스크래치의 기능 차이

구분	엔트리 1.5	스크래치 2.0
기본 객체	엔트리봇	cat
객체	오브젝트	무대와 스프라이트
배경	장면에 따른 배경 오브젝트 있음 /배경 위치 이동 가능	있음/ 장면에 따른 배경 이동 없음
장면 크기	480*270	480*360
자동초기화	가능	불가능
글상자	가능	불가능
하드웨어 연동	가능	가능(API 3.0 가능)

구분	엔트리 1.5	스크래치 2.0
방향	방향과 이동방향	방향으로 통일
변수	자료	데이터
호출	신호보내기	방송하기
함수	모든 오브젝트에서 가능	추가 블록을 활용하여 생성한 오브젝트에서만 사용 가능
다시 시작	처음부터 다시 실행하기 블록을 사용함	없음
스마트폰 연동	안드로이드, 아이폰 가능	불가능
학습 모드	가능	불가능
학급 기능	가능	3.0 가능
강의 기능	가능	불가능
텍스트언어 연계	엔트리파이썬 가능	불가능
작품 공유	가능	가능
코드 보기	가능	가능
토론/게시판	가능	가능
개인 저장소	불가능	가능
작품 복구	가능	불가능
작품 출력	가능	불가능
비디오 감지	불가능	가능
소리 편집	불가능	가능
조건 반복	~까지, ~인 동안	~까지
색에 닿았는가?	불가능	가능
한글 입력	가능	3.0 가능

3. 선행 연구

김인호(2019)는 창의적 문제해결력과 소프트웨어 관련 진로지향도에 대해 효과성 검토에서 창의적 문제해결력 향상에 긍정적인 영향을 미치고, 남학생이 여학생보다 창의적 문제해결력 향상 정도가 더 높았다. 이를 통해 모든 학생에게 효과적인 소프트웨어 교육이 되도록 설계되어 적용되어야 할 것이라고 하였다.

김은탁(2018)은 구미시 지역의 5학년 학생 66명을 대상으로 문제 해결력과 학습 몰입에 대한 변화를 알아보았다. 양적 연구를 통해 스크래치를 활용한 교과 통합형 수업을 한 집단에서 문제 해결력과 학습 몰입 향상에 유의미한 변화가 있었지만, 문제 해결력의 하위 영역인 ‘문제의 해결책 고안’ 영역에서는 집단 간의 유의미한 차이가 없었다. 스크래치 프로그래밍 언어는 그래픽 기반 언어로 블록 조립의 직관적인 인터페이스가 몰입에 긍정적 영향을 끼치는 것으로 보인다. 여러 교과와 배경지식을 동시에 활용하는 통합적 사고 과정에서 문제 해결 학습의 반복을 이루어져 문제 해결력 향상에 도움이 되었다는 것을 확인하였다.

이규완(2018)은 코딩 프로그램을 적용한 수업이 초등학교 학생의 컴퓨팅 사고력 변화에 미치는 영향 연구를 하였다. 코딩교육 프로그램을 적용한 수업이 컴퓨팅 사고력의 알고리즘과 절차에는 유의미한 차이가 있으나 자료 분석과 추상화 사고 능력에는 유의미한 차이가 없었다. 이를 통해 알고리즘과 절차 사고 능력 향상에는 코딩교육 프로그램이 일반 교과 수업보다 효과적임을 알 수 있으며, 전체적인 컴퓨팅 사고력 향상을 위해서는 다양한 프로그램 개발과 적용이 필요하다고 하였다.

이신현(2018)은 경기도 P시의 P초등학교 5학년 2개 반 35명을 대상으로

재구성한 Keller의 코스 흥미 검사와 진로성숙도 검사지를 통해 학습 흥미도와 진로 인식에 대해 독립표본/대응표본으로 변화를 확인하였다. 그 결과를 살펴보면 집단 간의 학습 흥미도는 유의미한 차이를 보였다. 집단 안에서도 학습흥미도 향상 정도는 실험집단이 유의미한 결과를 보였다. 이에 소프트웨어 교육이 수업흥미도 향상에 유의미한 영향을 미쳤을 것으로 판단하였다. 소프트웨어 교육이 학생들의 진로인식 수준을 향상시키는 데는 유의미한 결과를 미치지지는 않지만 교육 후 학생들의 진로인식 향상에 긍정적인 영향을 미치고 있음을 알 수 있었다.

진세언(2018)은 초등학교 5, 6학년생을 대상으로 만족도 조사와 컴퓨팅사고력에 대해 추상화, 알고리즘과 절차화를 비교하니 그 능력이 향상되었고 학생들의 수준에 맞춘 교육과정을 운영한 결과 흥미와 내적 학습 동기를 느끼는 것을 알 수 있었다. 따라서, 학생들의 발달 과정에 맞추어 체계적인 프로그램 교육의 체계가 보완되어야 할 것이다.

강오한(2017)은 00대학교 과학영재교육원의 초등심화 수학, 정보의 30명을 대상으로 수행한 결과에 따르면, 협동학습을 수행한 학생들의 사회성이 향상된 것으로 나타났다. 스크래치의 리믹스 기능을 활용한 프로그래밍 학습이 통계적으로 유의미하며, 실험집단의 협동심, 사교성, 자주성에서 긍정적인 효과가 있는 것으로 나타났다.

송정범(2017)은 00지역 중학교 3학년 학생 56명을 대상으로 수학교과 속의 개념을 활용한 게임 제작 활동(스크래치의 포트리스 게임)이 현행 실생활 문제 중심 수학학습 활동과 수학교과의 흥미와 가치인식 수준에 미치는 효과를 확인해보니 수학교과 흥미, 가치인식 수준 모두 향상되었고 특히, 수학교과 가치 인식 수준의 신장 정도가 더 큰 효과를 보였다.

SW코딩교육은 초등과학영재학생과 일반학생 모두 진로성숙도, 사회적 기여의식 향상에 긍정적인 영향을 미치며, SW코딩관련 대회에 참가한 경

험은 학생의 사회적 기여의식 향상에 긍정적임을 알 수 있었다. SW코딩교육을 받은 학생의 진로성숙도 향상과 사회적 기여의식향상은 서로 정적인 상관관계를 갖는 것도 확인되었다(김성환, 2017).

블록형 프로그래밍 교육은 컴퓨터 사고력 향상, 창의성 신장에 도움이 된다는 것을 확인할 수 있다(임수영, 2017; 김민주, 2017).

EPL(교육용 프로그래밍 언어) 교육보다는 피지컬 컴퓨팅 교육이 만족도와 사고력, 문제해결력의 향상 측면에서 더 효과적이다(김재휘, 김동호, 2016).

이민영(2016)의 블록형 프로그램 실험연구에서 살펴보면 엔트리와 스크래치 모두 흥미는 높아짐을 알 수 있었으나, 두 프로그램의 차이는 발견하지 못하였다.

중학생의 정보과학 창의적 성향에 미치는 영향과 논리적 사고력 향상에 스크래치 프로그래밍 교육은 긍정적인 효과가 있다(김기열, 2016; 김진우, 2015; 김경규, 이종연, 2016).

홍정미(2016)은 코딩경험이 없는 3, 4학년 동아리 반 20명을 대상으로 SW인식을 분석한 ICT소양검사에서 정보처리과정 이해, 효율적인 작업을 위한 작업순서 예측, 문제해결 전략세우기 등 3개의 항목에서 유의미한 결과를 얻었다. ‘조건과 반복을 사용한 문제해결능력’에서는 사전검사보다 사후검사에서 평균이 향상되었음에도 불구하고 유의미한 결과를 얻지 못하고, 이것은 기본적으로 ‘조건’과 ‘반복’에 대한 이해가 있었기에 향상도가 작았다. SW에 대한 인식 변화에서는 전반적으로 긍정적인 부분을 보였다.

Simon Rose(2016)는 ‘Bricolage Programming and Problem Solving Ability in Young Children: An Exploratory Study.’에서 Scratch와 같은 시각적 프로그래밍 환경이 학교에서 해결하고 계산하는 사고 기술 문제를 가르치는 데 많이 사용되고 있으나 학문적 연구는 시각적 프로그래밍이 컴

퓨터적 맥락에서 문제 해결에 미치는 영향으로 나누어진다. 분석에 따르면 게임 참가자들의 상대적인 발전에 따라 달라질 수 있다.

Kalelioglu & Gülbahar(2014)는 ‘The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners’ Perspective.’에서 컴퓨터 프로그래밍은 논리적 추론과 더불어 문제 해결 기술 개발을 위한 중요한 역량으로 인식된다. 초등학교 5학년 학생들의 문제 해결 능력에 대한 Scratch 프로그래밍의 영향을 조사하기 위한 탐색적 노력을 했다. Scratch 플랫폼에서 프로그래밍을 하더라도 초등학교 학생들의 문제 해결 능력에 큰 차이가 없었다. “그들의 문제 해결 능력에 대한 자신감”의 요소 평균에는 현저하지 않은 증가만 있을 뿐, 초등학교 학생들의 생각이 고려되었을 때, 모든 학생이 프로그래밍을 좋아하고 그들의 프로그래밍을 개선하기를 원한다고 분명히 말할 수 있다. 대부분의 학생이 Scratch 플랫폼을 사용하기 쉽다고 생각했다.

Calder, Nigel(2010)은 ‘Using Scratch: An Integrated Problem-solving Approach to Mathematical Thinking.’에서 이 논문은 아이들이 Scratch라고 하는 대화형 프로그래밍을 사용하여 수학 사고를 어떻게 발전시키는지에 중점을 둔다. 6학년 학생들과 함께 한 파일럿 연구 프로젝트는 1학년 파트너를 대상으로 한 수학 게임을 설정하게 되어 있다. Scratch 프로그래밍은 아이들의 각도와 측정에 대한 지식을 향상시키고 수학적 사고를 용이하게 하며 스크래치(Scratch)를 사용하는 또 다른 작업을 제시한다.

Funke, Geldreich & Hubwieser(2017)는 ‘Analysis of scratch projects of an introductory programming course for primary school students’에서 컴퓨터 과학은 초등학교나 유치원과 같은 초기 1차 수준의 젊은 학생들에게 적합한 교수 방법과 내용에 대한 통찰력을 얻는 것이 점점 더 중요해지고 있다. 유아교육 수준에 점차적으로 진입하고 있음을 조사하기 위해 4학년

학생들을 대상으로 3일간의 특별 입문 프로그래밍 과정을 고안했다. 초등 4학년(여자 26명, 남자 32명)의 어린이들이 2016년 5월부터 8월까지 이 과정에 참여했으며 이 과정에서 아이들은 127개의 스크래치 프로젝트를 개발했다. 학생들은 스토리, 애니메이션 및 게임의 세 가지 유형의 프로그램을 만들었으며, 게임을 프로그래밍 한 학생들의 이해 수준은 대부분 향상되었다.



Ⅲ. 연구방법

1. 연구대상

초등학생들의 창의성 향상을 위한 소프트웨어 교육으로 교육용 프로그래밍 언어(EPL) 중에서 가장 많이 수업에 활용되는 스크래치와 엔트리를 활용하여 소프트웨어 교육을 진행하였다.

부산 기장군 G초등학교 6학년 2개 반을 대상으로 반별로 나누어서 3반은 스크래치, 4반은 엔트리 교육을 계획하여 2학기(2018. 10. ~ 2019. 2)에 진행하였다. 교육을 진행하기 전 2회기에 걸쳐 사전검사를 하여 동일한 집단임을 확인하였다.

총 46명을 소프트웨어 교육을 진행하기 전, 후에 사전·사후 검사를 하였다. 불성실하게 응답한 설문지와 설문 당일 결석한 학생들의 설문지를 포함한 10부를 제외하고 36명을 대상으로 소프트웨어 교육 프로그램 효과 분석을 하였다.

<표 III-1> 연구대상

집단	학반	인원	처치내용
실험집단A	6학년 3반	19명	스크래치 교육
실험집단B	6학년 4반	17명	엔트리 교육

2. 실험 설계 및 분석 방법

소프트웨어 교육용 프로그램의 학습 내용은 소프트웨어 중심 사회, 엔트리 연구소, 에듀넷 등의 교육 사이트에서 개발된 교사용 지침서, 교육 서적을 만드는 출판사에서 판매되고 있는 도서 등을 참고하여 초등학생들이 흥미를 갖고 재미있게 수업에 참여할 수 있을 내용을 재구성하고, 시연중심 모델(DMM)과 재구성중심모델(UMC)을 병행하며 수업할 수 있도록 재편집하였다.

효과 검증으로 체험 전과 후의 창의적 문제해결력의 변화, 성별에 따른 차이, 교육용 프로그래밍 언어(블록코딩으로 스크래치와 엔트리 적용)에 따른 차이를 알아보았다. 방법으로는 독립표본 t 검정과 사전·사후 검증으로 대응표본 t 검정으로 분석하였다.

본 연구의 독립변인은 소프트웨어 교육 프로그램이며, 종속변인은 창의적 문제해결력이다. 학교 교육일정과 조율하여 실험집단 A, B에게 수업 전 사전검사를 하고, 소프트웨어 교육 프로그램을 40분씩 14차시 진행한 뒤 사전검사와 동일한 설문지를 사후검사에도 사용하였다.

<표 III-2> 실험 설계

구분	사전검사	소프트웨어 교육 프로그램	사후검사
실험집단A	O_1	X_1	O_2
실험집단B	O_3	X_2	O_4

3. 검사도구

본 연구에서는 창의적 문제해결력에 대한 측정을 위해 사전·사후 검사지로 2004년 서울대학교 심리연구실 MI팀에서 개발한 검사지를 사용하였다. 이 검사지는 2001년 한국 교육개발원의 “간편 창의적 문제해결력 검사 개발 연구 I”(연구책임자 조석희)를 기반으로 작성되었다. 모든 문항은 5점 균형 척도인 Likert 척도를 사용하여 숫자가 클수록 창의적 문제해결력이 높다는 것을 나타낸다. 창의적 문제해결력의 하위변인으로 특정한 영역의 지식과 사고, 기능과 기술의 이해와 숙달 여부, 확산적인 사고, 비판적·논리적 사고, 동기적 요소로 구성되어 총 문항 수는 20문항이다. 신뢰도 분석을 하여 크론바하 알파 계수(Cronbach's α)를 산출한 결과 하위변인 모두 0.7(평균 .78) 이상으로 높게 나타나, 본 연구의 주요 변인들의 신뢰도는 양호한 것으로 판단된다.

<표 III-3> 하위요소별 문항번호

하위변인	문항번호	문항수	Cronbach's α
특정한 영역의 지식과 사고, 기능과 기술의 이해 및 숙달 여부	1, 2, 3, 4, 5	5	.78
확산적 사고	6, 7, 8, 9, 10	5	.88
비판적, 논리적 사고	11, 12, 13, 14, 15	5	.83
동기적 요소	16, 17, 18, 19, 20	5	.85

4. 연구절차 및 자료처리

본 연구의 내용은 크게 다섯 부분으로 나누어 1장 서론, 2장 이론적 배경, 3장 연구방법 및 적용, 4장에서는 프로그램 적용 후의 연구결과를 5장은 논의 및 결론으로 구성하였다. 본 연구는 2018년 10월부터 2019년 2월까지 40분씩 14차시로 진행하였다. 소프트웨어 교육은 본 연구자가 담당하였으며 통계프로그램으로 SPSS v.25를 사용하여 분석하였다. 연구의 실제 진행 절차는 다음과 같다.

<표 III-4> 연구의 진행 절차

연구 절차단계	연구 세부내용
선행연구 탐색	창의적 문제해결력에 대한 선행연구 고찰 소프트웨어 교육 프로그램을 활용한 선행연구 탐색
소프트웨어 교육 설계 및 내용	소프트웨어 교육 프로그램 설계 및 내용 측정도구 구성 설문지 작성
사전검사 프로그램 실시 사후검사	연구대상 : 초등학교 6학년 2개 반 실험기간 : 2018년 10월 ~ 2019년 2월 (14차시) 측정도구 : 창의적 문제해결력 (사전, 사후 검사) 실험도구 : 소프트웨어 교육 프로그램
자료분석 연구결과	SPSS v.25로 통계처리하여 분석, 결과 도출

IV. 연구결과

1. 소프트웨어 교육 프로그램 구성

가. 소프트웨어 교육 프로그램 설계

창의적 문제해결력 향상을 위한 소프트웨어 교육 프로그램은 2015 개정 교육과정 분석을 토대로 설계 방향을 설정한 뒤 교육 내용과 교수학습 모형을 적용하여 프로그램 지도안을 작성하였다.

한국교육학술정보원(2018)이 2018년에 발간한 연수교재를 살펴보면 학교 교육에 적용하는 교육과정은 “교육 목표와 내용, 방법, 평가를 체계적으로 조직한 교육 계획”으로 정의하였다. 이러한 전체를 다른 말로 수업디자인이라고도 한다. 현재 요구되는 수업디자인(교육과정)은 성취기준 중심으로 핵심역량, 교과역량에 맞게 학생참여 중심, 과정 중심, 컴퓨팅 사고력 반영 수업의 목표로 설정하고 17차시 이상 교육과정을 구성하되 지역, 학생 수준 등의 실태에 따른 교육과정을 재구성하고, 컴퓨팅 사고력 중심의 교수학습방법을 적용하고, 학습의 과정으로 학생 평가를 고려하여 수업활동이 곧 평가가 되도록 해야 한다고 한다(한국교육학술정보원, 2018; 한국창의재단, 2017).

이에 본 연구자는 창의적 문제해결력 향상을 위해 교육용 프로그래밍 언어는 이해하기 쉽고, 배우기 쉬운 블록 방식의 교육용 언어인 스크래치와 엔트리를 학습도구로 선정하였다.

창의적 문제해결력 향상을 위한 프로그램 설계 방향은 다음과 같다.

첫째, 교육과정 분석을 통해 2015년 개정 교육과정 실과의 내용과 성취기준이 포함되도록 한다.

둘째, 수업대상자(여기서는 연구대상)의 소프트웨어 경험, 관심 등의 실태를 확인한 뒤 교육 내용에 반영하여 교육과정을 재구성하도록 한다.

셋째, 창의적 문제해결력 향상을 위해 교육모형과 학습전략을 적용하도록 한다.

넷째, 소프트웨어의 중점 요소인 순차, 반복, 조건 등을 포함하여 경험하도록 한다.

다섯째, 학생들이 학습이 쉽고 재미있다는 것을 느낄 수 있도록 구성하여 학습에 자발적 참여가 이루어지도록 한다.

나. 창의적 문제해결력을 위한 교수학습전략

본 연구에서의 교수학습모델로 ‘소프트웨어교육 교수학습 모델 개발 연구’(한국교육개발원, 한국교육학술정보원, 2015)에서 제안된 5가지 소프트웨어교육 교수학습 모델로 시연중심모델, 재구성중심모델, 디자인중심모델, CT요소 중심모델이 있다. 이 중에서 시연중심모델(DMM모델)과 재구성중심모델(UMC모델)을 기반으로 2015 개정 교육과정에서는 소프트웨어의 기초 소양과 관련된 교육 내용 체계의 기술시스템 영역인 소프트웨어 이해, 절차적인 문제해결, 프로그래밍의 요소와 구조를 바탕으로 수업내용을 구성하였다.

시연중심모델은 교사의 설명과 시연을 보여주면 학생들이 따라 작성, 모방을 하고 학습전략의 질의, 응답을 통해 창의적인 활동이 이루어지도록 유도하면서 단계적으로 제작하는 방식의 교수학습모델이다. 재구성중심모델은 학습자들이 놀이와 체험을 함으로써 동기유발이 되도록 하고 이 놀이

와 체험의 관찰, 탐색을 통해 교사의 의도적으로 알고리즘 등을 변형하여 제시하면 학습자들이 스스로 수정활동을 통해 자신만의 프로그램으로 설계하는 교수학습모형이다. 이러한 과정을 통해 컴퓨팅 사고의 전반적인 과정을 이해하게 된다(한국교육개발원, 한국교육학술정보원, 2015).

기장군의 G초등학생들은 코딩경험이 거의 없으며, 경험이 있는 학생들은 기초단계로 확인되어 시연중심모델과 재구성중심모델을 병행하는 교수-학습 과정으로 구성하였다.

<표 IV-1> 시연중심모델 교수학습 과정안 예시

학습 단계	교수-학습 활동	시간	유의점 및 자료
시연	동기유발 (문제 인식) 학습 목표 제시 학습 활동 안내	5	핵심전략을 교사가 설명 시범보이기
모방	활동1 (설명하기) 활동2 (시범보이기, 따라하기)	5 10	교사의 시연을 모방을 통해 학습자들이
제작	활동3 (만들기) 추가활동 제시 및 정리하기 (반복하기)	10 10	직접 제작 단계적 반복

시연중심모델의 교수학습 절차에 따라 교사는 학습활동에 필요한 명령어 블록을 학습자에게 설명한 뒤, 교사의 시연에 따라 모방하여 작품을 만들어 가도록 하는 진행하는 교수모델이다. 이러한 학습에서의 학습전략으로는 단순 모방이 되지 않도록 교사가 무조건 시범을 보일 것이 아니라 학습

자들의 창의적 문제해결력을 유도하도록 ‘이런 경우에는 어떤 것을 하면 될까요?’ 등의 발문을 사용하고 2인 1조로 서로의 아이디어를 공유하여 학생들이 참여하는 학생참여중심이 되도록 한다. 수업의 시작은 교사가 주도하다가 본시 수업으로 진행되면서 자연스럽게 학생들의 참여로 학생이 주도하도록 한다.

<표 IV-2> 재구성중심모델 교수학습 과정안 예시

학습 단계	교수-학습 활동	시간	유의점 및 자료
놀이	동기유발 (문제 인식) 학습 목표 제시 학습 활동 안내	5	프로젝트를 체험하여 탐색, 관찰 후 아이디어 추가, 내용 확장하여 설계, 제작
	활동1 (탐색하기)	10	
수정	활동2 (확장하기)	10	
재구성	활동3 (구현, 개발, 산출) 정리 (공유하기)	10 5	

위 학습에서 학습전략으로 같은 프로젝트를 체험하여 재미를 통해 새로운 아이디어가 산출되도록 발표를 시키고 자신이 말한 것을 구현하기 위하여 프로그램에서 필요한 요소를 찾아 설계, 제작하도록 유도하도록 한다. 완성된 작품들은 온라인 공유하여 다른 친구들과의 작품을 살펴서 다양한 아이디어를 서로 탐색할 수 있도록 한다. 이러한 학습 과정을 통해 학생들의 창의적 문제해결을 체험하고 컴퓨팅 사고의 과정을 이해하게 된다.

다. 교수학습 지도계획 - 교육내용 선정 및 내용

소프트웨어 교육을 체험한 후 각 프로그램에 대한 차이를 확인하기 위해 동일한 주제를 선택하여 스크래치와 엔트리 학습을 진행하였다.

6학년들의 창의적 체험활동 및 다른 교육과정으로 인하여 소프트웨어교육은 14차시로 수업을 진행하였다. 교수-학습 지도계획은 다음과 같다.

<표 IV-3> 교수-학습 지도계획

구분	차시	학습주제	소프트웨어 학습 영역
기초	1	소프트웨어 세상	소프트웨어의 이해
	2	블록 코딩미션	소프트웨어 알고리즘
이해	3	움직이는 나비	순차, 반복, 조건, 신호
	4	통통통 튀어요	순차, 반복, 조건, 변수, 난수
	5	통닭을 먹고 싶어요	순차, 반복, 조건, 복제, 변수, 난수
	6	나만의 계산기	변수, 신호, 묻고 답하기, 문자열 합치기
발전	7-8	두더지 잡기	순차, 반복, 조건, 변수, 난수, 신호
	9	물풍선 터트리기	순차, 반복, 조건, 변수, 난수
	10	신나는 물총 놀이	순차, 반복, 조건, 변수, 난수, 복제
응용	11-12	하늘에서 음식이 내려와요!	순차, 반복, 조건, 변수, 난수, 복제
	13-14	창작 미션	순차, 반복, 조건

라. 차시별 학습 목표와 학습 내용

<표 IV-4> 차시별 학습 목표와 학습 내용 (기초, 이해)

차시	구분	학습 목표 및 학습 내용
1	학습 목표	창의적 문제 해결력 사전 검사 소프트웨어 세상의 다양한 용어를 설명할 수 있다.
	학습 내용	소프트웨어, 4차 혁명, IoT, AI, 자율 주행 자동차 등의 용어를 이해하고 우리 생활 속의 소프트웨어를 찾아보기
2	학습 목표	다양한 코딩 학습 사이트에서 미션을 해결할 수 있다.
	학습 내용	code.org에서 미로학습으로 순차, 반복 의미 알아보기 lightbot.com(코딩기반의 퍼즐게임)에서 알고리즘 이해하기
3	학습 목표	움직이는 나비와 따라다니는 나비를 만들 수 있다.
	학습 내용	EPL의 구성요소에 대해 알아보기 오브젝트(스프라이트)가 마우스 포인터 따라다니도록 하기 다른 오브젝트는 처음 오브젝트를 따라다니도록 하기 * 확장 - 나비에 닿았을 때 ‘잡았다’라고 말하기 * 확장 - 키보드로 움직여보기
4	학습 목표	바운스 볼 게임을 체험하고 모방하여 만들 수 있다.
	학습 내용	EPL사이트에 공유된 ‘바운스 볼 게임’ 검색하고 체험하기 게임을 만들 때 필요한 블록들을 자유롭게 말해보기 이동판이 마우스의 좌우만 따라다니기(x좌표) 공이 자유롭게 움직이도록 하기 이동판에 닿으면 회전하면서 대각선 방향으로 튕기기 * 확장 1 - 바닥에 닿으면 게임 종료하기 * 확장 2 - 점수변수와 생명변수 적용하기
5	학습 목표	통닭을 먹고 싶어요.
	학습 내용	먹고 싶은 음식을 추가하고 자유롭게 나타나고 사라지기 좌표, 복제, 변수, 난수의 개념을 이해하기 * 확장 - 미세먼지 제거하는 프로그램 만들기
6	학습 목표	나만의 계산기를 만들 수 있다.
	학습 내용	문고 대답하기 기능을 사용하여 질문과 답을 표현하기 변수 2개를 사용하여 사칙연산을 계산하는 계산기 만들기 * 확장 - 신호를 사용하여 계산기 버튼을 누르면 답하기

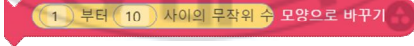
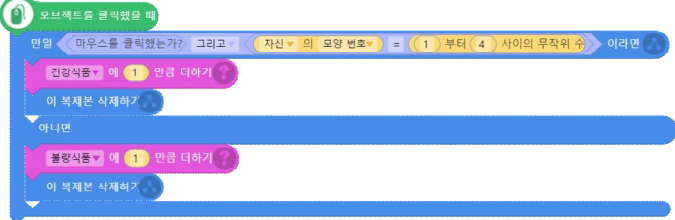
<표 IV-5> 차시별 학습 목표와 학습 내용 (발전, 응용)

차시	구분	학습 목표 및 학습 내용
7-8	학습 목표	고전게임의 대표작 두더지게임을 만들 수 있다.
	학습 내용	망치가 때리는 모양을 표현하기 두더지가 숨었다가 나타나도록 표현하기 망치에 닿으면 두더지의 모양 바꾸기 * 확장 1 - 점수변수를 사용하여 두더지가 망치에 맞을 때마다 점수 올리기 * 확장 2 - 게임의 표지를 만들고 장면 바꾸기
9	학습 목표	물풍선을 터트리는 게임을 만들 수 있다.
	학습 내용	물풍선이 자유롭게 움직이면서 나타났다가 사라지기 풍선을 선택하면 풍선이 터지게 하기 점수변수를 사용하여 풍선 터질 때마다 점수 올리기 * 확장 - 타이머 60초를 정해서 60초가 지나면 게임 종료하기
10	학습 목표	신나는 물총 놀이 게임을 만들 수 있다.
	학습 내용	키보드를 누르면 움직이는 물총을 만들기 오른쪽에서 왼쪽으로 오브젝트(스프라이트)가 움직이도록 한다. 물총게임에 필요한 아이템을 6개 정도 정해서 무작위로 아이템이 바뀌면서 나타나도록 한다. * 확장 - 점수 올리기
11-12	학습 목표	하늘에 음식이 내려오는 게임을 만들 수 있다. (물총게임 발전)
	학습 내용	하늘에서 자신이 좋아하는 음식이 내려오기 오브젝트를 활용하거나 모양을 이용하여 여러 음식이 무작위로 나오게 하기 건강식품을 선택하면 점수가 올라가고 불량식품을 선택하면 점수가 내려가게 하기 * 확장 - 두 개의 상자를 만들어서 건강, 불량이 정해지면서 음식이 들어가도록 하기(예: 쓰레기 분리수거)
13-14	학습 목표	창의적 활동 - 자기만의 작품 만들기 창의적 문제 해결력 사후 검사
	학습 내용	각자의 아이디어를 내고 스스로 작품 만들기 순차, 반복, 조건을 꼭 사용하기 * 주제를 정하지 못한 학생들에게는 미로를 제안하여 자신만의 미로를 만들고 점수를 올리는 것, 내리는 것 아이템의 의견을 주기도 한다.

차시별 학습 목표와 학습 내용에는 기본활동에서 확장활동까지 미리 계획하여 학생들의 수준별 학습이 되어야 한다. 차시별 학습계획 중에서 11-12차시의 교수학습 지도안은 <표 IV-6>과 같다.

<표 IV-6> 11~12차시 상세 지도안 (엔트리 기준, 2차시 연강)

일시	년 월 일	차시	11/14
주제	하늘에서 음식이 내려와요!	수업모델	DMM
학습 목표	복제기능을 사용하여 음식이 내려오게 할 수 있다.		
성취 기준	[6살04-09] [6살05-10] [6살04-11]		
알고리즘 요소	순차, 반복, 조건 (변수, 난수, 복제)		
단계	학습 요소	교수학습활동	자료□ 및 유의점※
도입 10분	동기 유발	▶ 동기유발하기 ‘하늘에서 음식이 내린다면’ 애니메이션 보여주기 하늘에서 음식이 내려오는 상황을 코딩으로 어떻게 표현해야 하는지 말해보기	※ 흥미를 이끌 수 있도록 동영상 을 보여준다.
	학습 문제 확인	▶ 학습 문제 확인하기 하늘에서 음식이 내려오는 게임을 만들어보자.	
전개 30분	학습 활동	▶ 학습활동 안내 [활동 1] 음식 오브젝트 추가하고 음식 오브젝트의 모양에 여러 오브젝트를 넣기 - 음식 오브젝트 추가하기 (예: 도넛 등) - 자신의 모양이 사라지고 복제본을 만들기 - 특정 위치를 잡아서 시작해도 된다. 	※ 제시된 블록이 정답이 아니라는 것을 상기시킨다.

단계	학습 요소	교수학습활동	자료(□) 및 유의점(※)
전개	학습 활동	[활동 2] 위에서 아래로 떨어지도록 하기 - 오브젝트의 X, Y 좌표를 확인하기 - ‘복제본이 처음 생성되었을 때’ 모양이 보이고 아래(Y)로 내려가기 - 내려가는 속도를 무작위 수를 사용하여 자유롭게 하기 - ‘아래쪽 벽에 닿으면’ 다시 위쪽으로 올라가도록 하되 X좌표의 위치를 자유롭게 하기  [활동 3] - 10종류 이상의 음식이 무작위로 선택되어서 나오게 하기 - 오브젝트를 여러 개 만드는 방법과 한 오브젝트에서 모양을 바꾸게 하는 방법이 있다. 여기서는 한 오브젝트에서 모양을 바꾸게 하자. - 아래 블록을 어디에 넣어야 하는지 답은 알려주지 않고 각자가 체험하면서 찾을 수 있도록 한다.  [활동 4] - 건강식품, 불량식품 변수를 만든다. - 시작 값은 0으로 하고 건강식품을 클릭하면 점수가 올라가고 불량식품을 선택하면 점수가 내려가게 한다. 	※ 제시된 블록이 정답이 아니라는 것을 상기시킨다.

단계	학습 요소	교수학습활동	자료(□) 및 유의점(※)
20분	확장 활동	[확장 활동] - 게임 시간을 설정해본다. - 쓰레기 분리수거하는 프로그램을 보여주고 ‘음식이 내려오 면’을 발전시켜 본다.	
정리 및 평가 20분	학습 내용 정리	[결과물 확인] - 공유로 게시한 뒤 팀별로 발표를 한다. - 자신의 작품과 다른 팀 작품을 보면서 느낀 점 말하기 - 다음 차시에 대한 예고를 한다. (13, 14차시는 창의 활동으로 자기가 만들고 싶은 작품에 대 한 고민을 해보라고 전한다.)	※ 친구들 의 아이디어 에서 또 다 른 아이디어 를 발견할 수도 있다.
	학습 평가	자신이 만든 작품을 기준으로 평가한다.	
		1. 복제기능을 사용하여 음식이 내려오게 할 수 있다.	
		2. 순차, 조건, 반복을 이해하고 다룰 수 있다.	
		3. 만든 프로그램에서 창의적 활동이 가능하다.	
		4. 자기 주도적으로 수업에 참여하였다.	

11~12차시는 ‘하늘에서 음식이 내린다면’이라는 애니메이션을 보면서 작품을 코딩에 적용하기 위해 어떤 블록들이 필요한지 말하도록 한다. 하늘에서 음식이 떨어지는 프로그램을 만든 뒤 음식을 클릭하면 사라지게 하는 게임을 만드는 것으로 다양한 생각들을 나누면서 진행한다. 수업에 쓰이는 블록을 제시하지만 같은 블록으로 만들 필요는 없다는 것을 알려주어 수업의 전체가 창의적으로 진행될 수 있도록 한다. 11차시에서는 시연중심모델로 수업을 진행하다가 12차시에서는 재구성중심모델로 각자의 아이디어를 수용하면서 재구성되어 자신만의 프로그램으로 만들 수 있도록 지도한다.

2. 소프트웨어 교육 프로그램의 효과 검증

가. 창의적 문제해결력의 사전·사후 변화

소프트웨어 교육 프로그램이 창의적 문제해결력에 유의미한 차이를 보이는지 검증하고자 대응표본 t 검정을 사용하였다. 그 결과 소프트웨어 교육 프로그램은 창의적 문제해결력의 하위영역인 확산적 사고 영역에서 유의미한 차이($t = -2.47, p < .05$)를 보이는 것으로 나타났다.

<표 IV-7> 창의적 문제해결력 사전·사후 검사

영역		인원	평균	표준화 편차	표준오차 평균	t
사고	사전	36	3.04	.78	.13	-1.61
	사후	36	3.25	.84	.14	
확산	사전	36	3.11	.78	.13	-2.47*
	사후	36	3.46	.95	.16	
비판	사전	36	4.06	.66	.11	-.37
	사후	36	4.09	.75	.13	
동기	사전	36	3.54	.73	.12	-1.54
	사후	36	3.73	.90	.15	

* $p < .05$

나. 성별에 따른 창의적 문제해결력 차이

소프트웨어 교육 체험 후 창의적 문제해결력이 성별에 따른 차이가 있는지 검증하고자 독립표본 t 검정으로 분석한 결과는 <표 IV-8>과 같다. 총 4가지 하위영역 중에서 확산적 사고 영역에서 성별에 따른 유의미한 차이 ($t = 2.09, p < .05$)를 보이며 남학생이 3.73 여학생은 3.11로 남학생의 확산적 사고 영역이 여학생보다 높음을 알 수 있다.

<표 IV-8> 성별의 창의적 문제해결력 차이

	성별	인원	평균	표준화 편차	표준오차 평균	t
사고	남	20	3.46	.93	.21	1.72
	여	16	2.99	.65	.16	
확산	남	20	3.73	1.04	.23	2.09*
	여	16	3.11	.73	.18	
비판	남	20	4.27	.65	.14	1.60
	여	16	3.88	.84	.21	
동기	남	20	3.85	.99	.22	.91
	여	16	3.56	.79	.20	

* $p < .05$

다. 프로그램 도구에 따른 창의적 문제해결력의 차이

교육용 프로그래밍 언어에 따른 창의적 문제해결력의 차이를 검정하고자 독립표본 t 검정을 한 결과는 <표 IV-9>와 같다. 문제해결력의 하위 영역인 동기적 요소에서 유의미한 차이($t=-2.10, p<.05$)를 보였고, 스크래치는 3.4, 엔트리는 4.0으로 교육용 프로그래밍 언어 중에서 스크래치보다 엔트리가 더 긍정적인 영향으로 확인되었다.

<표 IV-9> 프로그램 도구에 따른 창의적 문제해결력 차이

	도구	인원	평균	표준화 편차	표준오차 평균	t
사고	스크래치	19	3.15	.78	.18	-.77
	엔트리	17	3.36	.92	.22	
확산	스크래치	19	3.36	.87	.20	-.64
	엔트리	17	3.56	1.06	.26	
비판	스크래치	19	4.11	.68	.16	.09
	엔트리	17	4.08	.85	.21	
동기	스크래치	19	3.44	.77	.18	-2.10*
	엔트리	17	4.05	.96	.23	

* $p < .05$

V. 논의 및 결론

1. 논의

가. 소프트웨어 교육 프로그램 설계 방향

2015 개정 교육과정과 소프트웨어교육 운영지침에서 초등학교의 소프트웨어 교육 목표와 초등학교의 성취 기준을 준거로 하고, 교수학습 모델은 선행연구 분석을 통해 적합한 모델을 선정하였다. 과거 정보 교과에 제안된 많은 교수학습모델에는 다양한 모델들이 있고 현재에도 새로운 모델이 연구되고 있다는 것은 정보교과 교수학습 모델 또는 방법이 정해져 있기보다는 학습자, 환경 등의 실태에 따라 다르게 구성할 수 있음을 시사한다.

‘2018년 소프트웨어 선도 교육 연수 교재’에 따르면 실과 교과의 소프트웨어 단원에서 제시한 주제와 학습 목표들이 학교마다 처한 환경, 지역적 특성, 학생들의 사전 경험이 다르기에 교실 수업에 적용하는 데는 한계점이 있기에 교과의 성취기준과 교육요소, 활동의 흐름은 유지하면서 수업의 주제를 재구성할 필요성을 제시하였고, 학생들의 수준에 따라 수준별 학습도 제공하고 있다(교육부, 2018).

창의적 문제해결력 향상을 위한 소프트웨어 교육 프로그램 설계에 있어서 교육과정, 성취기준, 지역성, 교수학습모델 등을 모두 고려하여 교수-학습 내용을 재구성하였다.

나. 창의적 문제해결력의 사전·사후 변화

소프트웨어 교육 프로그램에 참여한 6학년 2개 반의 사전·사후 변화를 확인하니 창의적 문제해결력의 확산적 사고 영역에서 유의미한 차이가 있음이 나타났다. 블록형 프로그래밍 교육은 컴퓨터 사고력 향상, 창의성 신장에 도움이 된다(임수영, 2017; 김민주, 2017)는 사실을 확인하였고 연구대상을 세 집단으로 나누어서 각각 다른 소프트웨어 교육 프로그램을 적용했음에도 세 집단 모두가 확산적 사고력이 향상되었고(김우찬, 2016), 스크래치 프로그래밍 교육은 창의적 성향에 미치는 영향과 논리적 사고력 향상에 긍정적 효과가 있다(김기열, 2016; 김진우, 2015; 김경규, 이종연, 2016)는 연구 결과가 본 연구 결과와 일치한다.

다. 성별에 따른 창의적 문제해결력의 차이

소프트웨어 교육 프로그램 활동이 창의적 문제해결력의 하위영역인 확산적 사고영역에서 성별에 따라 유의미한 차이를 보였고 남학생의 평균이 더 높은 것으로 나타났다. 다른 영역에서는 큰 차이는 없었지만 대부분 남학생의 평균이 높음을 알 수 있다. 김인호(2019)의 연구에서도 남학생이 창의적 문제해결력 향상의 정도가 더 높았으며 하위영역으로는 자기 확신 및 독립성, 비판-논리적 사고에서 향상 효과가 더 크다는 것을 알 수 있다. 손경진(2018)의 로봇 활용 SW 교육이 창의적 문제해결력에 영향을 미치는 영향에서는 대부분의 영역에서 실험집단의 점수가 높게 나타났으며, 그중에서도 남학생이 좀 더 향상된 결과가 나왔다. 이와 같은 결과는 모든 학생에게 창의적 문제해결력에 긍정적인 효과가 있으며, 특정 영역에서는 남학생에게 더 효과가 있다는 선행 연구 결과를 지지하고 있다.

라. 프로그램 도구에 따른 창의적 문제해결력의 차이

소프트웨어 교육용 프로그램은 다양하게 개발되어 초등학생들에게 배우기 쉽고 학습효과가 높은 블록형 프로그래밍 언어로 스크래치와 엔트리를 선택하여 각각 다르게 처치하여 그 결과를 확인해보니, 동기적 요소에서 유의미한 차이가 있으며 엔트리가 더 효과가 높았다. 동기적 요소는 학습을 지속해서 하고 싶은 학습흥미, 몰입, 학습 목표에 달성하고자 하는 의지 등의 영역으로 스크래치보다 엔트리가 높다는 것은 국내에서 만들어진 프로그램으로 좀 더 흥미를 유발하였음을 알 수 있었다. 이민영(2016)의 연구에서 엔트리와 스크래치를 학습한 집단 모두 논리적 사고력이 향상되었으며, 엔트리는 확률 조합 논리에서 향상되었고 스크래치는 모든 영역의 논리가 고르게 향상되었다. 이에 “논리적 사고력 향상을 위해 고른 발달에는 스크래치가 적합하다”라고 하였으나 두 프로그램 모두 논리적 사고력 향상에는 도움이 된다. 임지빈, 이태민, 채정민, 정영희(2019)에서 컴퓨터 사고력 핵심능력에 따른 엔트리와 스크래치의 기능을 비교한 결과를 살펴보면, 모두 엔트리가 핵심능력에 대한 기능들이 더 많아 컴퓨팅 사고력의 다양한 형태로 핵심능력을 키울 수 있어 엔트리가 컴퓨팅 사고력을 증진하는 데 더욱 도움이 된다고 한다. 이러한 연구 결과가 상이한 것은 창의적 문제해결력, 컴퓨팅 사고력, 논리적 사고력 등의 차이 분석 방향에 따라 달라지는 것으로 판단된다. 따라서, 교육과정을 개설할 때 수업의 목적을 살펴서 프로그램을 선별할 필요가 있음을 시사한다.

2. 결론 및 제언

본 연구의 목적은 소프트웨어 교육 프로그램이 초등학생에게 창의적 문제 해결력 향상에 효과가 있는지를 분석하여 창의적 문제해결력 향상을 위한 소프트웨어 교육 프로그램을 활용하도록 제안하고자 하는 것이다.

이를 위해 2015 개정 교육과정과 실과 교과 내에서 소프트웨어 교육 성취기준을 살펴 초등학생들에게 적합한 교육용 프로그래밍 언어를 활용하여 소프트웨어 교육 프로그램을 설계하였다. 부산광역시 G초등학교 6학년 2개 반을 연구대상으로 교육용 프로그래밍 언어를 스크래치, 엔트리로 다르게 선정하여 2018년 10월부터 2019년 2월까지 소프트웨어 교육 프로그램을 14차시 진행하였다.

본 연구를 통해 초등학생의 ‘창의적 문제해결력 향상을 위한 소프트웨어 교육 프로그램’은 어떻게 구성되고 진행되어야 하며, 소프트웨어 교육 프로그램의 참여가 창의적 문제해결력에 미치는 영향, 창의적 문제해결력에 성별이 미치는 영향, 교육용 프로그래밍 언어가 다른 경우 창의적 문제해결력에 미치는 영향을 알아보하고자 하였다. 이러한 소프트웨어 교육 프로그램의 효과를 검증하기 위하여 소프트웨어 교육 전, 후에 창의적 문제해결력 검사를 하고, 통계 분석하여 종합한 결론은 다음과 같다.

첫째, 교육과정을 준수하고 실과의 성취기준을 지키며 학습의 효과를 위한 수업전략을 적절히 사용하여 학생들의 학습동기를 일으키는 수업이 되어야 한다. 교사중심수업에서 학생중심수업으로 진행될 수 있도록 교사가 수업을 진행할 때 프로그램을 무조건 모방하여 기계처럼 따라하지 않도록 수업전략을 구성해야 한다.

둘째, 소프트웨어 교육용 프로그램 활동은 창의적 문제해결력을 향상시

키는 데 효과가 있었다. 소프트웨어 교육 후 창의적 문제 해결력의 하위영역인 확산적 사고 영역에서 유의미한 차이가 나타났다. 이는 초등학생의 창의적 문제 해결력 향상에 도움이 된다는 선행연구(김우찬, 2016; 김민주, 2017; 임수영, 2017)를 지지한다. 따라서 교육용 프로그래밍 언어를 활용한 소프트웨어 교육 프로그램은 창의적 문제해결력 향상에 긍정적인 효과가 있다.

셋째, 소프트웨어 교육 프로그램은 남학생의 확산적 사고 영역 향상에 효과가 있었다. 소프트웨어 교육 후 성별의 창의적 문제해결력의 하위영역인 확산적 사고 영역에서 유의미한 차이가 나타났다. 다른 영역에서는 미미하지만, 여학생보다 평균이 높음을 알 수 있다. 남학생들이 컴퓨터와 기계에 대한 호기심이 많음에 따라 컴퓨터를 활용한 소프트웨어 교육 프로그램이 학습몰입으로 인하여 창의적 문제해결력에 변화가 있었다고 할 수 있으며 성별에 다른 차이를 분석한 선행연구(김인호, 2019; 손경진, 2018)에서 모두 남학생이 높은 것을 확인하였다. 이러한 결과는 소프트웨어 교육 프로그램이 여학생보다는 남학생에게 좀 더 효과가 있다고 하겠다.

넷째, 창의적 문제해결력 향상에는 엔트리를 활용한 교육 프로그램이 동기적 요소에서 효과가 있었다. 소프트웨어 교육 후 프로그램에 따른 차이는 엔트리를 활용한 수업에서 동기적 요소가 좀 더 긍정적인 영향을 미치는 것으로 나타나지만 다른 특정적 사고, 확산적 사고, 비판적 사고에서는 유의미한 차이가 없었다. 블록코딩의 대표 프로그램인 스크래치와 엔트리를 활용한 프로그램 개발을 한 후 적용하여 효과를 분석해보니 소프트웨어 교육이 확산적 사고 영역에서 효과가 있고, 프로그램 도구에 따라 동기적 요소에서 효과가 달리 나타남을 알 수 있었다. 스크래치와 엔트리를 비교한 선행연구를 보면 논리사고력 효과 검증(이민영, 2016)에서는 스크래치가 전 영역에 고른 효과가 있었고, 컴퓨팅 사고력 증진 분석(임지빈, 이태민,

채정민, 정영희, 2019)에서는 엔트리가 더 많은 기능이 있으므로 효과가 있다고 하였다. 이런 상이한 결과는 스크래치와 엔트리는 둘 다 교육적 효과가 있으며 프로그램 분석의 방향에 따라 효과가 다를 수 있다.

본 연구의 결과를 바탕으로 후속 연구를 위한 제언은 다음과 같다.

첫째, 소프트웨어 교육 프로그램의 효과 영역을 창의적 문제해결력의 한 영역에 국한하지 말고 컴퓨팅 사고, 학습몰입, 협동 학습, 진로 인식 등의 다양한 영역으로 나누어서 추가적인 연구가 필요하다.

둘째, 소프트웨어 교육 프로그램의 효과 검증을 위해 다양한 지역과 보다 많은 학생들을 설정하고 실험기간을 좀 더 늘려서 지속적 효과까지 검증해 볼 필요가 있다.

셋째, 검사지의 응답방식이 학생들의 판단에 의한 자가응답으로 통계처리를 하였기에 일반화의 한계가 있을 수 있어 연구결과가 실제와 다소 차이가 날 수 있을 것이다. 따라서 검사지를 보완할 수 있는 관찰지, PMI, 평가 등을 연구 개발할 필요가 있다.

넷째, 프로그램의 효과를 분석하기 위해 본 연구는 양적 측정에 의존하여 결과가 산출되었기에 다양한 질적 연구가 병행되어야 한다.

다섯째, 본 프로그램을 적용한 수업시간 이외의 창의적 문제해결력에 영향을 미칠 수 있는 여러 가지 상황과 여건에 대해 고려하지 않았다.

이번 연구 결과가 창의적 문제해결력 향상에 소프트웨어 교육 프로그램이 효과가 있음을 모두 대표하기에는 한계점이 있다는 제한점을 보완하고 초등학생의 창의적 문제해결력 향상을 위한 다양한 소프트웨어 교육 프로그램 개발 후속 연구에 도움을 줄 수 있을 것으로 기대된다.

참 고 문 헌

- 강오한(2017). 프로그래밍 학습이 초등생의 사회성에 미치는 영향. **한국컴퓨터교육학회 학술발표대회논문집**, 21(2), 185-188.
- 교육부(2014). **소프트웨어교육 활성화 방안**. 서울: 교육부.
- 교육부(2015). **SW중심사회를 위한 인재양성 추진계획 발표**. 보도자료.
- 교육부(2015). **초중등 소프트웨어교육 운영지침**. 서울: 교육부.
- 교육부(2016). **초등 SW 교육 선도교원 연수자료**. 서울: 교육부.
- 김경규, 이종연(2016). 컴퓨팅 사고력 기반 프로그래밍 학습의 효과성 분석. **한국컴퓨터교육학회 논문지**, 19(1), 27-29.
- 김기열(2016). 스크래치 프로그래밍 교육이 중학생의 정보과학 창의적 성향과 기술적 문제해결 성향에 미치는 영향. **대한공업교육학회지**, 41(2), 119-133.
- 김대수(2017). **소프트웨어와 컴퓨팅 사고**. 경기: 생능출판사.
- 김민주(2017). 스크래치 프로그래밍을 활용한 정보교육이 초등학생에게 미치는 영향. **경인교육대학교 석사학위논문**.
- 김성환(2017). SW코딩교육이 초등과학영재와 일반 학생의 진로성숙도, 사회적 기여의식에 미치는 영향. **경인교육대학교 석사학위논문**.
- 김우찬(2016). 소프트웨어교육이 초등학교 5학년 학생들의 논리적 사고력과 창의적 문제해결력에 미치는 효과. **경희대학교 석사학위논문**.
- 김은탁(2018). 스크래치 활용 교과 통합형 수업이 초등학생의 문제 해결력 및 학습 몰입에 미치는 영향. **한국교원대학교 석사학위논문**.
- 김인호(2019). 소프트웨어 교육이 초등학생의 창의적 문제해결력 및 SW 관련 진로지향도에 미치는 영향. **아주대학교 석사학위논문**.

- 김재휘, 김동호(2016). 컴퓨팅 사고력 향상을 위한 초등 피지컬 컴퓨팅 교육과정 개발. **한국정보교육연구**, 20(1), 69-82.
- 김진숙, 한선관, 김수환, 정순원, 양재명, 장의덕, 김정남(2015). **SW 교육 교수학습 모델 개발 연구**. 수탁연구 CR 2015-35.
- 김진우(2015). **스크래치를 활용한 프로그래밍 학습이 논리적 사고력에 미치는 영향**. 제주대학교 석사학위논문.
- 김태훈(2015). **컴퓨팅 사고력 신장을 위한 프로그래밍 중심 STEAM 교육 프로그램**. 제주대학교 박사학위논문.
- 김혜진(2016). **아두이노를 이용한 피지컬 컴퓨팅 기반의 프로그래밍 교육이 중학생의 창의적 문제해결력에 미치는 영향**. 한국교원대학교 석사학위논문.
- 김효성(2016). **창의적 문제해결력 신장을 위한 알고리즘 설계 중심의 프로그래밍 교육프로그램 개발 및 적용**. 서울교육대학교 석사학위논문.
- 미래창조과학부(2014). 소프트웨어 중심사회. <http://www.software.kr/>
- 미래창조과학부(2015). 소프트웨어 중심대학 추진계획. <https://msit.go.kr/>
- 박영순(2016). **엔트리를 활용한 초등학교 SW 교육 모델의 설계 및 적용**. 대구교육대학교 석사학위논문.
- 박정미(2013). **스크래치를 활용한 STEAM 교육이 초등 정보영재의 논리적 사고력에 미치는 영향**. 안동대학교 석사학위논문.
- 박주연(2015). **Scratch 프로그래밍 수업에서 학습자 특성, 학습몰입, 학습효과의 구조적 관계 규명**. 이화여자대학교 박사학위논문.
- 배영권(2006). **창의적 문제해결력 신장을 위한 유비쿼터스 환경의 로봇 프로그래밍 교육 모형**. 한국교원대학교 박사학위논문.
- 서울대심리연구팀(2004). **창의적 문제해결력 검사도구**.

- 손경진(2018). 내러티브 아트 활동 기반 로봇활용 SW 교육 효과성 비교. 청주교육대학교 석사학위논문.
- 송정범(2017). 스크래치 활용 게임 프로그래밍 학습이 수학교과 흥미와 가치인식에 미치는 영향. 정보교육학회논문지, 21(2), 199-208.
- 안경미(2010). 스크래치 프로그래밍 교육이 초등학생의 학습 몰입과 프로그래밍 능력에 미치는 효과. 경인교육대학교 석사학위논문.
- 엔트리연구소(2015). 소프트웨어 교육 활성화를 위한 엔트리 교육과정 개발 방안 연구. 수탁연구 CR 2015-01.
- 이규완(2018). 코딩교육이 초등학생 컴퓨팅 사고력에 미치는 영향. 청주교육대학교 석사학위논문.
- 이민영(2016). 엔트리와 스크래치를 활용한 초등학생의 논리적 사고력 신장에 관한 연구. 서울교육대학교 석사학위논문.
- 이신현(2018). 컴퓨팅사고력 기반 소프트웨어교육이 초등학교 학생들의 학습흥미와 진로인식에 미치는 영향. 한국교원대학교 석사학위논문.
- 이은경(2013). 계산적 사고 향상을 위한 창의적 스크래치 프로그래밍 학습. 컴퓨터교육학회논문지, 16(1), 1-9.
- 이은경(2013). 컴퓨터 교육의 전망과 과제:계산적 사고 및 협력적 문제 해결 능력 향상을 위한 융합적 접근. 한국컴퓨터정보학회 학술발표논문집, 21(2), 203-206.
- 임수영(2017). 블록형 코딩 언어 학습을 통한 컴퓨팅 사고력 향상의 분석. 경북대학교 산업대학원 석사학위논문.
- 임지빈, 이태민, 채정민, 정영희(2019). 컴퓨터 사고력 핵심능력에 따른 엔트리와 스크래치간 기능 비교. 한국컴퓨터교육학회 학술발표대회 논문집, 23(1), 141-144.

- 전용수(2018). **초등학교 소프트웨어 교육의 성공요인에 관한 실증적 연구**. 숭실대학교 박사학위논문.
- 전용주(2017). **새로운 교육과정의 소프트웨어 교육을 위한 컴퓨팅 사고력 기반 창의적 문제해결 수업 모형의 개발 및 적용**. 한국교원대학교 박사학위논문.
- 정은영(2008). **Squeak Etoys 기반 정보 교육이 초등학생의 창의적 문제해결력에 미치는 영향**. 한국교원대학교 석사학위논문.
- 조석희(2001). 교과를 통한 창의적 문제해결력 교육방법 모색: 문제중심학습. **한국교육**, 28(2), 205-227.
- 조석희, 한국교육개발원(2001). **간편 창의적 문제해결력 개발 연구(I)**. 서울: 한국교육개발원.
- 조석희, 한국교육개발원(2002). **간편 창의적 문제해결력 개발 연구(II)**. 서울: 한국교육개발원.
- 조석희, 장영숙, 정태희(2003). 영재판별을 위한 간편 창의적 문제해결력 검사 개발을 위한 기초 연구. **한국교육**, 30(1), 259-296.
- 진세언(2018). **블록코딩 교육을 통한 학습자의 컴퓨팅사고력 변화 분석**. 공주교육대학교 석사학위논문.
- 커넥트재단 소프트웨어야 놀자(2016). <https://www.playsw.or.kr/main>
- 한국교육개발원, 한국교육학술정보원(2015). **SW 교육 교수학습 모델 개발 연구**. 서울: 한국교육개발원.
- 한국교육학술정보원(2018). **2018년 소프트웨어(SW)교육 선도교원 연수교**. 서울: 한국교육학술정보원.
- 한선관, 류미영(2016). **컴퓨팅 사고력을 위한 소프트웨어 교육**. 경기: 생능출판사.
- 함성진(2011). **프로그래밍 단원 도입을 위한 초등학교 컴퓨터 교육과정**

설계에 관한 연구. 청주교육대학교 석사학위논문.

홍정미(2015). 스크래치를 활용한 SW융합교육프로그램이 초등학생의 ICT소양과 SW인식에 미치는 영향. 경인교육대학교 석사학위 논문.

황지현(2010). 창의적 문제해결력 신장을 위한 가상현실 학습공간의 설계. 한국교원대학교 석사학위논문.

Berliner, D. C. & Rosenshine, B. (1977). The acquisition of knowledge in the classroom, In R. C. Anderson et al., *Schooling and the Acquisition of Knowledge*. Hillsdale, NJ: Lawrence Erlbaum Associate.

Brennen, K., Balch, C., & Chung, M. (2014). *Creative computing guide book*, Retrieved Form <http://scratched.gse.harvard.edu/guide>

Calder, Nigel. (2010). Using Scratch : An Integrated Problem-solving Approach to Mathematical Thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14.

Dewey, J. (1910). *How We Think*. Lexington, MA: D.C. Heath and Company

Funke, A., Geldreich, K. & Hubwieser, P.(2017). Analysis of scratch projects of an introductory programming course for primary school students. *2017 IEEE Global Engineering Education Conference (EDUCON)* : 2017 IEEE. 1229-1236 Apr, 2017

Gagne, R. M. (1985). *the conditions of learning(4th ed)*. NY : Holt Rinehart & Windion.

Getzels, J. Z. & Jackson, P. W.(1962). *Creativity and Intelligence*. NY : Wiley.

- Guilford, J. P. (1959). *Creativity and it's cultivation*. NY : Harper and Row.
- Kalelioglu, F. & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills : A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33-50.
- Mayer, R. E. (1999). *Problem solving*. In M, A, Runco S. R. Pritzker(eds.), *Encyclopedia of creativity*, 2. San Diego, CA: Academic Press.
- Simon Rose. (2016). Bricolage Programming and Problem Solving Ability in Young Children: An Exploratory Study. *Proceedings of the European Conference on Games Based Learning*; 2016(1), 914-921.
- Sternberg, R. J. & Lubart, T. I. (1995). *Defying the croed : Cultivating creativity in a culture of conformity*. NY : Free Press.
- Treffinger, D. J. & Isaksen, S. G. (1985). *Creative Problem Solving: The basic course*. Buffalo : Bearly Limited.
- Wing. J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.

<부록> 창의적 문제 해결력 검사지

날짜		시간			
학반	6학년 () 반 () 번	성별		이름	

안녕하세요?

본 설문지는 여러분들이 수업 시간에 이루어지는 학습을 어떻게 생각하며 참여하고 있는지 알아보고자 하는 것입니다.

모든 질문에 대한 실제 정답은 없으며, 학생들의 순위를 매기거나 비교하는 것이 아니므로 평소 느끼고 생각한 대로 솔직하고 성실하게 응답하면 됩니다. 또한, 설문지의 응답 결과는 본 연구를 위한 자료로만 활용되며, 설문 결과는 다른 사람에게는 절대 공개가 되지 않습니다.

※ 응답요령

1. 각각의 질문에 대해 반드시 하나만 √표 해 주세요.
2. 모든 질문에 대하여 성실한 답변을 부탁드립니다.

★ 해당하는 번호에 √표 해 주세요.

1. 코딩(스크래치, 엔트리 등)을 배운 적이 있나요?
① 예 ② 아니오
2. 있다면 어떤 것을 배웠나요?
① 스크래치 ② 엔트리 ③ 피지컬컴퓨팅 ④ 기타()
3. 배우지 않았다면 배우고 싶은 코딩은 무엇인가요?
① 스크래치 ② 엔트리 ③ 피지컬컴퓨팅 ④ 기타()

다음 문항을 잘 읽고 해당되는 부분에 √ 표 하시오.

1. 특정한 영역의 지식, 사고기능, 기술의 이해와 숙달 여부

문 항	질문	전혀 그렇 지않 다	그렇 지않 다	보통 이다	그렇 다	매우 그렇 다
1	나는 수업 시간 중에 호기심이 많아서 계속 질문한다.	①	②	③	④	⑤
2	나는 제시된 문제에 대해 다양한 답을 찾아내며, 가끔 독특한 답을 제시한다.	①	②	③	④	⑤
3	나는 수업 시간에 생각을 자유롭게 표현하며, 가끔 다른 사람들과 의견이 맞지 않더라도 나의 주장을 굽히지 않는다.	①	②	③	④	⑤
4	나는 평소 유머가 풍부한 편이며, 남이 웃을 것 같지 않은 상황에서도 남들을 곧 잘 웃긴다.	①	②	③	④	⑤
5	나는 수업 시간에 머를 쓰는 활동을 좋아한다.	①	②	③	④	⑤

2. 확산적 사고

문 항	질문	전혀 그렇 지않 다	그렇 지않 다	보통 이다	그렇 다	매우 그렇 다
6	나는 새롭고 남과 다른 생각을 말할 수 있다.	①	②	③	④	⑤
7	나는 이미 알려져 있는 것과는 다른 새로운 방법으로 문제를 해결할 수 있다.	①	②	③	④	⑤
8	내가 생각한 아이디어는 다른 친구들이 생각한 아이디어와는 많이 다르고 새롭다.	①	②	③	④	⑤
9	나는 다양하고 풍부한 아이디어로 문제를 풀 수 있다.	①	②	③	④	⑤
10	나는 서로 상관없어 보이는 것을 잘 연결짓는다.	①	②	③	④	⑤

3. 비판적, 논리적 사고

문 항	질문	전혀 그렇 지않 다	그렇 지않 다	보통 이다	그렇 다	매우 그렇 다
11	나는 현실과 상상을 구별할 수 있다.	①	②	③	④	⑤
12	나는 나의 아이디어를 생각한 대로 잘 정리하고 설명할 수 있다.	①	②	③	④	⑤
13	나는 남이 하는 이야기가 진실인지 거짓인지 구별할 줄 안다.	①	②	③	④	⑤
14	나는 친구들과 함께 나눈 다양한 정보를 가지고 스스로 문제 해결을 할 수 있다.	①	②	③	④	⑤
15	나는 주어진 문제와 관련 있는 정보를 찾아낼 수 있다.	①	②	③	④	⑤

4. 동기적 요소

문 항	질문	전혀 그렇 지않 다	그렇 지않 다	보통 이다	그렇 다	매우 그렇 다
16	나는 힘들고 어려운 것도 쉽게 포기하지 않고 끝까지 하려고 한다.	①	②	③	④	⑤
17	나는 수업시간에 배운 과목의 다른 내용(주제)에 대해서도 공부를 더하고 싶다.	①	②	③	④	⑤
18	나는 수업 시간의 학습 내용이 재미있다.	①	②	③	④	⑤
19	나는 목표를 이루지 못했다는 생각이 되면 목표를 이루기 위해 더욱 노력한다.	①	②	③	④	⑤
20	나는 목표를 이루었다고 생각이 되면 다음 단계의 목표를 정한다.	①	②	③	④	⑤

국문초록

본 연구의 목적은 소프트웨어 교육 프로그램이 초등학생에게 창의적 문제 해결력 향상에 효과가 있는지를 분석하여 창의적 문제해결력 향상을 위한 소프트웨어 교육 프로그램을 활용하도록 제안하고자 하는 것이다.

이러한 목적에서 창의적 문제해결력과 소프트웨어 교육 프로그램에 대한 이론을 고찰하고 이를 바탕으로 다음과 같은 연구 문제를 설정하였다.

첫째, 초등학생의 '창의적 문제해결력 향상을 위한 소프트웨어 교육 프로그램'은 어떻게 구성되고 진행되어야 하는가?

둘째, 소프트웨어 교육 프로그램의 참여는 창의적 문제해결력에 영향을 미치는가?

셋째, 소프트웨어 교육 프로그램은 성별에 따라 창의적 문제해결력의 차이가 있는가?

넷째, 교육용 프로그래밍 언어(스크래치와 엔트리)에 따른 창의적 문제해결력에 차이가 있는가?

본 연구는 부산 기장군 G초등학교 6학년 2개 반을 대상으로 2018년 10월부터 2019년 2월까지 소프트웨어 교육을 진행하였다. 반별로 다른 교육용 프로그래밍 언어(스크래치와 엔트리)를 사용하였다. 창의적 문제해결력 측정을 위해 사전·사후 검사로 2004년에 서울대학교 심리연구실 MI팀에서 개발한 창의적 문제해결력 검사 도구를 사용하였다.

효과 검증으로 창의적 문제해결력의 사전·사후 변화, 성별에 따른 차이, 교육용 프로그래밍 언어(스크래치와 엔트리)에 따른 차이를 알아보았다. 자료의 통계처리는 독립표본 t 검정과 사전·사후 검증의 대응표본 t 검정으로 분석하였다.

본 연구의 결과를 요약하면 다음과 같다.

첫째, 소프트웨어 교육 후 창의적 문제 해결력의 하위영역인 확산적 사고 영역에서 유의미한 차이가 나타났다.

둘째, 소프트웨어 교육 후 성별의 창의적 문제해결력에서도 확산적 사고 영역에서 유의미한 차이가 나타났다.

셋째, 교육용 프로그래밍 언어에 따른 차이는 엔트리를 활용한 수업이 동기적 요소가 좀 더 긍정적인 영향을 미치는 것으로 나타나지만 다른 특정적 사고, 확산적 사고, 비판적 사고에서는 유의미한 차이가 없었다.

블록코딩의 대표 프로그램인 스크래치와 엔트리를 활용한 소프트웨어 교육 프로그램의 효과를 분석해보니 소프트웨어 교육이 확산적 사고 영역에서 효과가 있고, 교육용 프로그래밍 언어에 따라 동기적 요소에서 효과가 다르게 나타남을 알 수 있었다. 그러므로 소프트웨어 프로그램을 교육하기 이전에 학생들의 창의적 문제해결력을 향상시킬 수 있는 방향에 대하여 충분한 논의를 거쳐서 교육과정을 편성해야 할 것이다.

본 연구의 결과에서 소프트웨어 교육 프로그램은 창의적 문제 해결력의 향상을 위한 한 도구로서 유용함을 알 수 있었다.