

스크래치를 활용한 수학수업에서 도구화 수준과 컴퓨터 프로그래밍 자기효능감에 관한 연구 - 등식의 성질을 중심으로1) -

이현지²⁾ · 고상숙³⁾

본 연구는 4차 산업시대를 맞아 코딩을 포함한 교수·학습이 강조됨에 따라 코딩수업에서 수학 교수·학습의 효과를 파악하고자 수행되었다. 이를 위해 경기도 D 중학교 1학년 학생들 32명이 스크래치를 활용한 7차시 프로그램에 참여하여, 그들의 활동자료는 컴퓨터에 매 차시 저장되었고, 연구자는 관찰일지를 매주 작성하였다. 중학교 등식의 성질에서 이렇게 수집된 자료를 통해 연구자는 도구화 수준에 대해 개별 학생의 수준을 조사하고, 학생의 컴퓨터프로그래밍 자기효능감을 사전과 사후 검사를 통해 조사하였다. 학생들의 도구화는 다양하게 나타났는데 연구가 시작된 시점에는 도구화의 제 3 또는 4수준에 해당하는 학생이 9%에 불과하였지만 연구마무리 시점에는 80%를 상회하였다. 학생들의 컴퓨터프로그래밍에 대한 자기효능감은 유의한 수준으로 향상되었다.

주요용어 : 컴퓨팅 사고, 등식의 성질, 프로그래밍(코딩), 도구화, 도구발생

I. 서론

1. 연구의 필요성 및 목적

코딩(프로그래밍)을 통해 향상시킬 수 있는 계산적 사고(Computational Thinking, 이하 CT)⁴⁾는 수학적 내용을 논리적으로 전개하여 해결하는 알고리즘의 과정 또는 응용과정 등에서 수학과 관련된 역량들의 향상에 잠재성을 지닌다(Sengupta, Kinnebrew, Basu, Biswas & Clark, 2013; 이상구, 2015; 신동조, 고상숙, 2019). 즉, CT는 정보교과 뿐만 아니라 수학교과의 교수·학습 과정에서도 향상될 수 있다는 것이다. 또한, 2015 개정 수학과 교육과정의 핵심역량 중 공학적 도구나 교구를 활용하여 정보처리 능력을 향상시켜야 한다는 것은 수학교과 수업시간에 소프트웨어를 이용한 코딩수업(교육부, 2016)을 연계하여 운영하는 것이 현 수학과 교육과정에 부합하다는 것을 시사한다.

* MSC2010분류 : 97D40

1) 이 논문은 제1저자의 2018년 석사학위 논문 일부를 재구성한 것임.

2) 단국대학교 대학원생(hyunji@dankook.ac.kr), 제1저자

3) 단국대학교 교수(sangch@dankook.ac.kr), 교신저자

이미 국내에는 수학교육에서의 코딩교육에 관한 다양한 연구들이 이루어지고 있다(cf., 고영해, 안재호, 박남제, 2011; 유정호, 2015; 권점례, 이광상, 김성경, 2016; 송정범, 2017; 이서빈, 고상숙, 2018). 이들 코딩교육과 관련된 선행연구들을 살펴보면 여러 가지 컴퓨터 프로그래밍 언어를 활용하여 수학적 요소를 기반으로 프로그램을 개발하였다. 그러나 거의 대부분의 연구가 초등학생을 대상으로 하였으며, 중·고등학생을 대상으로 하는 연구는 많이 이루어지지 않고 있는 상황이다. 이는 초, 중, 고 전 과정에서 프로그래밍 초보자들이 블록 짓기와 같이 활용하기에 용이한 스크래치를 사용하는 연구들이 좀 더 활발히 이뤄질 필요가 있겠다.

한편, 2015 개정 교육과정에서는 중학교 1학년 수학에서 등식은 ‘등호를 사용하여 수나 식이 서로 같음을 나타낸 식’ 또는 ‘등호를 사용하여 나타낸 식’으로 정의된다(김원경 외, 2013). 초등학교에서 등호는 산술적 의미로서의 연산의 결과로 다루어지다가 중학교에 올라가서 대수적 의미로 발달한다. Carpenter, Franke, & Levi(2003)는 등식의 성질을 기반으로 하지 않은 대수적 이해는 가능하지 않으며 이 과정을 많은 학생이 어려워하기 때문에 다양한 경험을 제공해야 한다고 하였다. 따라서 본 연구에서는 스크래치를 활용하여 중학교 1학년을 대상으로 등식의 성질에서 프로그래밍 과정을 적용하여 그에 따라 나타나는 도구화 수준을 분석하였고, 컴퓨터 프로그래밍 자기효능감의 변화를 조사하였다. 이를 위해 본 연구에서는 탐구학습법을 바탕으로 하는 개발중심모델(D^3 모델)을 따라 수업에 참여하는 학생들을 대상으로 7차시에 해당하는 연구지도안을 구성하여 적용하였다.

II. 이론적 배경

1. 프로그래밍(코딩)이 가능한 스크래치

스크래치(Scratch)는 미국의 MIT(Massachusetts Institute of Technology) Media Lab의 Lifelong Kindergarten Group에서 개발한 교육용 프로그래밍 언어이며, 애니메이션이나 게임 등을 직접 만들 수 있다. 2007년 스크래치의 설계자인 Mitchel Resnick⁵⁾은 스크래치는 힙합 DJ가 새로운 사운드를 만들기 위해 레코드를 섞어서 스크래치 하는 방법에서 영감을 얻어 만들었고, 스크래치를 통해 사람들이 창조적인 방식으로 소리뿐만 아니라 모든 종류의 미디어를 함께 믹싱 할 수 있게 하는 것이라고 하였다. 또한, 홈페이지를 통해 무료로 다운로드하여 사용할 수 있다는 장점이 있다.

스크래치는 무대, 스프라이트, 블록, 스크립트 영역과 모양, 소리 편집 영역으로 구성되어있으며, 아이들이 가지고 노는 레고 블록 같이 모양을 맞추며 프로그래밍을 하도록 되어 있고 이 프로그래밍에 따라 스프라이트가 움직인다. 블록은 기본으로 동작, 형태, 소리, 이벤트, 제어, 감지, 연산, 변수, 나만의 블록으로 구성되어 있고, 음악, 펜, 비디오 감지, 텍스트 음성 변환(TTS), 번역 등의 확장 기능을 추가할 수 있다. 블록은 그룹별로 그 색이 모두 달라서 블록의 색깔만으로도 어느 블록 그룹에 속해 있는지 파악할 수 있다.

Resnick,(2019)이 스크래치의 장점을 기술하였는데 우리의 상황을 고려하여 다음처럼 재구성해볼 수 있다: 첫째, 다양한 연령대가 사용가능하다. 8~16세 아이들을 대상으로 만들어져서 16세 이상의 사람들은 더욱 쉽게 사용가능하다. 둘째, 놀이하는 것처럼 학습하는 프로그래밍 언어이다. 명령어를 암기하여 직접 코드를 입력하는 것이 아닌 아이들이 가지고 노는 레고처럼 블록을 쌓으며 놀이처럼 할 수

5) <http://news.bbc.co.uk/2/hi/technology/6647011.stm>

있으므로 재미있는 학습이 가능하다. 셋째, 병렬 실행 및 단계적 실행이 가능하므로 절차에 따른 논리적 사고를 하도록 돕는다. 넷째, 다양한 미디어 효과 사용이 가능하다. 스크래치 프로그램 내에서 제공하는 다양한 스프라이트, 배경, 효과음을 활용할 수 있다. 다섯째, 프로그램이 오픈된 자원으로 무료이고 완벽하게 한국어를 지원하고 있어서 어느 프로그래밍 언어보다 더욱 쉽게 프로그래밍할 수 있다. 여섯째, 공유 및 공동 작업이 가능하다. 스크래치 프로그램에서 프로젝트를 만들어 공유하고 자신이 만든 프로젝트를 다른 사람이 다운받아 사용할 수도 있다. 완성된 프로젝트뿐만 아니라 스크립트까지 공유가 가능해 프로그래밍 된 구조도 확인할 수 있다.

대부분의 사람들은 프로그래밍은 전문가만이 할 수 있는 복잡하고 어려운 것이라고 생각하는 경향이 있지만, 위에 열거한 장점들로 스크래치는 누구나 사용할 수 있는 도구이다. 특히 문법 위주의 교육보다는 프로그램의 구조를 익히는 것과 논리적인 문제에 초점을 맞추어 제작되었기 때문에 프로그래밍을 처음 해보는 사람에게 입문과정으로 활용할만하다.



[그림 II-1] 스크래치 화면 구성



[그림 II-1] 도구화 제 2수준/개발단계에서 블록잇기 일부

수학교과와 연계한 코딩수업은 코딩에 대한 교육을 도입하는데 있어서 다양한 장점이 있다. 코딩을 통해 함양할 수 있는 컴퓨터 사고력(또는 계산적 사고라고도 일컬음, CT)은 수학적 내용이 논리적으로 전개되고 해결되는 알고리즘의 과정, 수학적 내용을 다방면으로 적용하는 응용과정 등 수학과 관련된 모든 과정에서 필요한 역량이며(이상구, 2015), 다시 말하면 2015개정 정보과 교육과정에서 제시하고 있는 핵심역량 중 ‘계산적 사고력’이 정보과교과뿐만 아니라 수학교과의 교수·학습 과정에서도 신장될 수 있음을 의미한다.

이에 따라 국내에서 코딩을 연계한 수학교육에서의 다양한 연구가 이루어지고 있다. 고영해 외(2011)는 LOGO를 활용한 4차시의 프렉탈 기하이론 기반의 초등학생 대상 프로그램을 개발하여 초등학교 수학과 교육과정에 다루어지는 도형, 측정, 규칙성과 문제해결 영역을 프로그래밍을 통해 학습할 수 있도록 하였다. 유정호(2015)는 다양한 컴퓨터 프로그래밍 언어 중, C언어를 활용하여 초등학교 5학년 및 6학년을 대상으로 하는 10차시 분량의 수학중심 프로그래밍 프로그램을 개발하였으며, 창의적 체험활동 시간에 적용 가능하도록 하였다. 프로그램에 포함된 수학적 내용에는 쌓기 나무의 개수 구하기, 약수 구하기, 소수 구하기 등이 있다. 권점례 외(2016)는 초등학교 수학과를 중심으로 학교 교육에서 코딩 기반 소프트웨어 교육의 적용 가능성을 탐색하기 위한 연구를 진행하였고, 코딩교육을

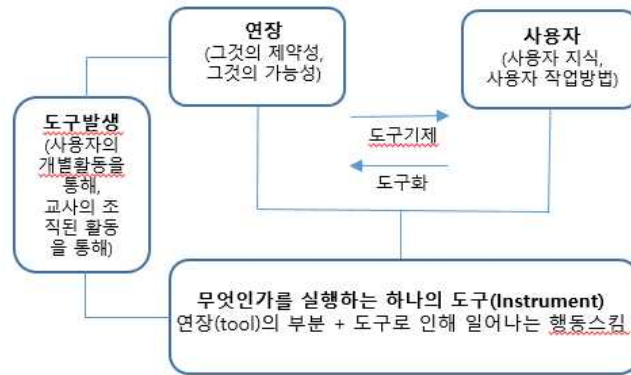
위한 4~6학년 군의 수학과 학습요소를 추출하여 스크래치와 엔트리와 같은 교육용 SW소프트웨어를 활용한 프로그램을 개발하였다. 강하람, 임채령, 조한혁(2021) 연구에서는 초 6학년·중 1학년 수학교과와 정보교과를 융합하는 코딩수학 교육과정과 이를 위한 최소 코딩게임 기반 교육방법에 대해 총 3년간 연구하였다. 제 1차 연도에는 공간좌표의 필요성에 따라 3차원 좌표의 수학적 개념을 포함하는 코딩환경으로 교육과정을, 제 2차 연도에는 명령어의 위계성에 따라 건물 요소별 다른 수준의 명령어를 도입하여 자기주도적 학습을 안내하고, 제 3차 연도에는 컴퓨팅 사고력 향상을 유도하는 최소 코딩게임 기반의 교수·학습 전략을 설계하여 컴퓨팅 사고력 진단을 위한 평가 및 피드백을 개발하였다. 이러한 자기주도적 학습 및 컴퓨팅 사고력 증진을 유도하는 최소 코딩게임 기반 교육방법과 코딩수학 교육과정은 수학-정보교과의 융합교육 연구와 실천에 의미가 있겠다. 하지만 코딩과 관련하여 본 연구에서 시도하고자 하는 등식의 성질에 대해서 과거 연구된 바가 없으며, ‘방정식의 조작은 등식의 이해를 기반으로 하기 때문에 등호를 이해하지 못하면 이후 대수 학습에 장애가 된다’(Carpenter et al., 2003)는 주장에 따라 등식의 성질에 대한 학생들의 이해를 도울 수 있는 다양한 연구가 이루어져야 한다.

한편 이러한 코딩을 포함한 수업에서의 정의적 측면을 조사한 연구로는 송정범(2017)은 수학교과와 삼각함수 내용과 코딩을 연계하여 중학교 3학년 학생들을 대상으로 포트리스 게임을 제작하는 6차시 프로그램을 개발하였고, 이를 통해 수학교과에서의 프로그래밍 학습이 수학교과에 대한 흥미와 가치 인식에 미치는 영향을 분석하였을 때 실험집단이 비교집단에 비해 유의미하게 향상되었음을 확인하였다. 이서빈, 고상숙(2018)은 파이썬을 이용한 총 7차시 함수 교수·학습에서 학생들의 컴퓨터프로그래밍 자기효능감을 조사하였는데 역시 유의미하게 향상되었음을 밝혔다. 또한 이러한 변화를 피하려면 공학도구에 대한 (예비)교사들의 인식도 시대에 맞게 꾸준히 재고될 필요가 있다(cf., 김소민, 2021; 고상숙, 박만구, 한혜숙, 2013).

2. 도구발생

도구의 발달에 대한 개인, 사회적 본질을 연구한 Trouche(2004)는 [그림 II-3]과 같은 결과를 도출하였다. 도구적으로 가치가 없는 가공물(artifact)의 잠재적인 기능(function)을 파악하여 특정한 목적(purpose)에 맞추어 변형(adapt)하여 사용하는 것을 의미하는 가공물의 도구화(instrumentalization), 그 도구로 인하여 발생하는 행동(instrumented action)을 통해 과제를 효율적(efficient)으로 처리하도록 하는 테크닉(technique)의 점진적 발달을 바탕으로 스키마(schema)의 발달이 이루어질 때, 연장⁶⁾(tool)이 도구기제(instrumentation)로 사용된다고 보았다(고상숙 외, 2015 재인용). 스킴(Scheme)은 Vergnaud(1996)의 정의에 따라 역동적인 기능의 실재(dynamic functional entity)이다. 스킴은 주어진 상황에서 주어진 상황에 대한 불변성을 지닌 행동 구성물이고 의도와 목표를 지닌다. 스킴 안에 내재된 지식을 조작적 불변성(operational invariant)이라고 한다. Trouche(2005)는 도구발생의 결과물이 이용 스킴(utilization scheme) 형태로 압축된다고 하였다. 다시 말하면 도구발생은 형성된 이용 스킴으로 구성된다는 것이다. 각 용어에 대해 정의를 하면 다음과 같다.

6) 역자 주: 본 연구에서 tool을 연장으로 번역한 것은 도구(instrument)와 구별하기 위함이다. 사용자에게 처음 연장으로 주어졌을 때 이를 자기의 행위 안에 사용하고자 인지하고 접근할 때 비로소 도구로 인식된다는 것임.



[그림 II-3] 도구발생과정(Trouche, 2004, 고상숙 외, 2015 재인용)

- 도구발생(instrument genesis): 주어진 문제를 해결하기 위해 도구를 활용한 결과물로서 개인의 개별 활동 또는 교사가 구성한 활동에서 도구기제와 도구화의 역동적 순환을 거쳐 이용 스킴으로 저장되는 것이다.
- 도구기제(instrumentation): 한세호, 장경운(2009)에서는 도구화된 행동스킴 (instrumented action scheme: IAS)으로 번역되어 사용되었고, 임현정, 고상숙 (2016)에서 도구화(instrumentalization)와 구별하여 도구기제로 번역하였다. 더 일찍이 Drijvers(2003)는 도구기제를 특정문제를 해결하기 위해 기술 공학적 연장을 활용하며 통일성 있는 유의미한 스킴으로 정의하였듯이, 도구가 가지는 특정한 절차를 따르는 것으로서 목적인 도구발생을 이끌어내기 위한 도구만의 적절한 과정으로 보았다.
- 도구화(instrumentalization): 사용자가 연장을 조정하는 것(Drijvers, 2003)을 의미한다. 도구발생을 위해 사용자가 자신의 지식과 활용방식에 따라 일련의 과정(process)을 연장에게 가하는 것으로 해석하였다. 본 연구에서는 관찰을 통해 사용자가 도구를 잘 사용하는 수준을 도구화 수준이라고 하였으며 도구발생은 도구화를 통해 이루는 목적이므로 학생이 올바른 과정으로 수행을 하였을 때 도구화와 구별없이 사용되기도 한다.

3. 자기효능감

교육목표 설정의 중요성이 인식되면서 좀 더 명확하고 구체적인 교육목표를 위한 연구가 이루어지게 되었고, 이에 따라 교육목표의 영역을 크게 지적영역, 정의적 영역, 기능적 영역으로 분류하였으며 (Krathwohl, Bloom, & Masia, 1964), 오늘날 우리가 알고 있는 인지적 특성과 정의적 특성은 이를 시작으로 정의되었다. 정의적 특성은 D. McLeod에 의해 좀 더 활성화되었다고 볼 수 있는데, 1989년 그의 연구에서는 이를 인지적 영역과 다른 영역으로 태도, 신념, 정서와 같은 감정이라고 하였다. 또한 McLeod(1992)는 정의적 특성을 태도와 신념, 감정, 자신감과 자기효능감, 자아개념 등으로 재정의하였으며 태도, 신념 그리고 감정이라는 세 가지 요소를 정의적 특성을 정의하는데 있어서 가장 중요한

하위요소라 하였다. 황정규(2002)는 정의적 특성의 하위요소를 흥미와 태도, 자아개념, 인성, 도덕성과 가치관, 동기, 협동심 등으로 제시하였으며, 정의적 특성에 대한 하위요소들은 상호 배타적인 것이 아니라 본질적으로 다변인복합체면서 각각의 고유성을 지니는 것이라 하였다. 이처럼 연구에 따라 정의적 특성에 대한 해석은 매우 다양한데, 좀 더 최근에는 자기효능감, 흥미, 불안, 태도, 신념, 동기 등과 같은 하위요소를 중심으로 측정도구의 개발, 또는 정의적 특성을 함양하는 프로그램에 대한 개발 등과 같이 학생을 위한 실질적인 연구가 다양하게 이루어지고 있다(cf., 남진영, 2015; 조혜정, 김인수, 2016; 이환철, 이현숙, 김형원, 이지혜, 고호경, 2017).

한편 정의적 특성의 구성요소 중에서 자기효능감을 Bandura(1986)는 어떤 결과를 요구하는 구체적인 상황에 대하여 자신이 성공적으로 그를 수행할 수 있다고 기대하는 개인의 신념이라 정의하였다. 또한 Bandura(1977)는 ‘성공의 경험’, ‘대리적 경험’, ‘언어적 설득’, ‘정서적 상태’로 네 가지의 자기효능감 요인이 있다고 하였으며, 자기효능감 차원으로는 ‘수중’, ‘강도’, ‘일반화’ 세 가지의 차원이 있다고 하였다. 성공 또는 성취의 경험과 대리적 경험의 정도는 직접적으로 자기효능감에 영향을 미친다. 언어적 설득은 피드백과 관련된 요소라 할 수 있으며, 정서적 상태에 따라 자신의 능력에 대한 평가를 하기 때문에 자기효능감의 하위요인에 포함된다. Salomon(1984)는 이러한 자기효능감이 높을수록 도전적인 과제에 긍정적인 자세를 보이고, 더 많은 노력을 통하여 상대적으로 높은 성취도와 성과를 보인다고 하였다. 이에 따라 자기효능감을 개인의 성과와 성취도를 예측하는 요인으로 고려해볼 수 있다(Linsley, Brass, & Thomas, 1995; Mitchell, Hopper, Daniels, George-Falvy, & James, 1994).

본 연구에서 연구한 컴퓨터 프로그래밍 자기효능감은 컴퓨터 프로그래밍 과정에서의 자기효능감을 의미한다. 우선 컴퓨터 프로그래밍이란 수식이나 작업을 컴퓨터에 알맞도록 정리해서 순서를 정하고 컴퓨터가 이해할 수 있는 명령 코드(code)로 고쳐 쓰는 작업을 총칭한 것으로, 명령 코드를 쓰는 작업 자체를 코딩(coding)⁷⁾이라고 한다. 여기서 코드란 기호의 체계로 볼 수 있으며 코딩이란 주어진 명령어를 특정한 컴퓨터 프로그래밍 언어를 활용하여 입력하는 것을 의미한다(이서빈, 고상숙, 2018). 따라서 앞서 Bandura(1986)의 관점에 따라 컴퓨터 프로그래밍 자기효능감을 ‘어떤 결과를 요구하는 구체적인 상황에 대하여 자신이 성공적으로 컴퓨터 프로그래밍을 할 수 있다고 기대하는 개인의 신념’으로 정의할 수 있으며, 언어, 자기효능감 요인, 그리고 자기효능감 차원에 의하여 구체화할 수 있다(김갑수, 2014).

Ⅲ. 연구방법

본 연구는 중학교 1학년 학생들을 대상으로 스크래치를 활용한 수학수업에서 인지적 측면과 정의적 측면의 교수·학습 효과를 검증하기 위해 매 차시 32명의 학생들의 도구화 수준을 조사하여 도구발생(등식의 성질 개념)과정을 이해하고자 하였고, 컴퓨터 프로그래밍 자기효능감은 단일집단 사전-사후검사(one group pretest-posttest design)로 대응표본 t-검정으로 분석하였다.

7) <http://terms.naver.com/entry.nhn?docId=1159441&cid=40942&categoryId=32837>

1. 연구대상

본 연구는 경기도 용인시 소재의 A 중학교의 자유학년제를 수행하는 학생들을 대상으로 진행하게 되었는데 개발된 수업의 학습을 희망하는 신청자를 모집하여 운영하였다. 수업모델 D^3 는 소프트웨어 활용을 수업에 적용할 때 학습자가 탐구(discovery)-설계(design)-개발(develop)을 거치게 되는데 먼저 스크래치의 원리를 탐구하고, 순서도를 설계하여, 그에 맞는 스프라이트를 작동하게 하는 과정을 통해 자신의 것을 개발해가는 것이다. 이에 대해선 다음 연구도구 단원에서 상세히 다룰 것이다. 작은 수의 학생집단으로 임의로 A반, B반 각각 16명의 두 개의 반으로 나누어 수업을 운영하였고 매주 월요일 연이은 시간대에 모두 동일한 수업을 적용하였다. 자유학년제 실시로 인하여 본 연구의 대상자들의 직전 학기 성취기준에 대한 수집이 불가능하였기 때문에 성취수준을 변인으로 두지 않았다. 또한, 학생들의 컴퓨터 프로그래밍 자기효능감에는 2명의 학생이 자신의 검사지를 제출하지 않고 귀가해버려서 30명의 학생의 자료가 수집되었다.

2. 연구절차

본 연구는 2019년 9월 2일부터 2019년 10월 28일까지 9월 30일, 10월 14일을 제외한 매주 월요일, 총 7주 동안 진행되었고 본 연구자가 직접 수업을 운영하였다. 본 연구를 진행하기에 앞서 문헌고찰을 통해 연구의 주제 설정을 명확히 하고 관련 선행연구를 조사하였으며, 컴퓨터 프로그래밍 자기효능감 검사지를 수집하여 수정 및 보완하였다. 이후 선행연구에서 제안한 코딩을 위한 교수·학습 모델을 참고하여 연구지도안을 구성하였고, 이에 필요한 활동지를 포함하였다. 연구 설계를 통하여 설정된 연구 집단을 대상으로 컴퓨터 프로그래밍 자기효능감을 위한 사전검사를 실시하고 7차시 분량의 수업을 적용하였으며, 수업 적용 이후에 사전검사와 동일한 문항으로 사후검사를 실시하였다. 한편 학생들의 인지적 사고 과정에서의 도구적 수준은 매 차시 학생들이 저장한 활동지 및 스크래치 프로그래밍 결과물을 바탕으로 분석하였다.

3. 연구도구

1) 코딩을 위한 수업모형(D^3 모델)

본 연구는 코딩을 포함한 수업의 효과를 위하여 개발한 수업모형에서 개발중심모델(D^3 모델)은 김진숙 외(2015)의 것을 적용하였다. 개발중심모델은 탐구학습법의 개념을 바탕으로 세 단계 즉 탐구(discovery), 설계(design), 그리고 개발(development)을 통해 본 연구에서 수학적 개념인 등식의 성질을 표현하는 양팔접시를 구현하는 과정을 포함하였다.

양팔접시 또는 양팔저울은 수학수업에서 등식의 성질을 쉽게 설명해주는 주요 교구 중에 하나이다. 본 연구에서는 학생들이 자신의 교구를 직접 구현해봄으로써 등식의 성질을 효과적으로 이해하도록 하는 것이다. 거의 대부분의 수업에서 교구가 주로 교사에 의해 주어지는 반면에 본 수업에서는 학생이 등식의 성질을 반영한 양팔접시를 직접 구현해보는 활동을 통해 조작적 사고를 강화하는 것이며 이는 컴퓨터 환경이 주어지므로 가능한 것이다. 이런 일련의 활동을 통해 등식의 성질에 대한 학생의 도구발생은 어떻게 나타나는지 그리고 학생의 컴퓨터 프로그래밍 자기효능감은 어떻게 변화하는지 조사하였다.

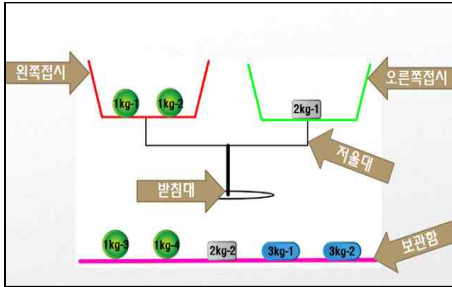
먼저 탐구(discovery)에서는 기능의 이해, 분석, 탐색, 구현하는 단계로 도전과 디버깅 과정을 통해 수정해가는 부분이다. 컴퓨팅 활동에는 재구성-진단-수정-평가라는 네 과정의 순환을 통해 양팔접시와 관련된 간단한 게임을 주제로 교사가 주요기능과 스크립트의 논리를 제안하고 피드백을 위해 학생들의 활동을 분석하여 메모한다. 특히 이 단계는 매 차시 비슷한 상황이 전개되므로 차시가 진행됨에 따라 익숙해져서 생략되는 경우가 발생가능하기도 한다.

<표 III-1> 연구 진행 절차

기간	진행 절차	세부 내용
2019.01. ~ 2019.08.	문헌 고찰	<ul style="list-style-type: none"> • 선행연구 조사 • 연구 문제 설정
	연구 설계	<ul style="list-style-type: none"> • 연구 대상자 설정 • 컴퓨터 프로그래밍 자기효능감 검사지 수정 및 보완 • 스크래치를 활용한 수학교과 코딩수업 모형 설계 • 스크래치를 활용한 수학교과 코딩수업 교수·학습 지도안 구성 • 스크래치를 활용한 수학교과 코딩수업 활동지 제작
2019.09.02	연구 수행	<ul style="list-style-type: none"> • 컴퓨터 프로그래밍 자기효능감 사전검사
2019.09. ~ 2019.10.		<ul style="list-style-type: none"> • 수업 적용
2019.10.28		<ul style="list-style-type: none"> • 컴퓨터 프로그래밍 자기효능감 사후검사
2019.09. ~ 2019.12.	연구 분석	<ul style="list-style-type: none"> • 자료 정리 및 분석 • 결과 분석 • 논문 작성 마무리

설계(design)에서는 알고리즘을 설계 및 계획하는 단계로 옆 학생과 공유를 통해 아이디어확장하며 설계사고(design thinking)을 강조하게 되는데 정의-창의설계-프로토타입-공유 및 평가를 통해 진행한다. 프로그래밍 언어로 구현하기 전에 즉 등식의 성질을 구현하기 위해 양팔접시에 사용되는 알고리즘을 세부적으로 생각해보도록 유도한다. 순서도, 의사(pseudo) 코드, 그리고 설계학습지 등을 활용할 수 있다. 코딩과정을 통해 [그림 III-1]에서처럼 저울대가 평형을 유지하기 위해 구현하는 방법을 터득한다. 즉 저울대의 평형은 기준점이 0도가 아닌 90도 시계반대방향으로 회전했을 때 가능하다는 스프라이트의 원리를 스크립트에서 입력하고 이를 반영한 것이 스프라이트에서 작동되는 것이다. 학생은 이런 스크래치의 특성에 따라 양팔접시가 구동할 수 있는 방법을 모색하게 되므로 등식의 성질이라는 도구발생을 위해 스크래치가 갖는 도구기제로써 기능을 이해하게 된다. 이를 통해 다음 단계인 개발에서 도구를 통제할 수 있는 도구화가 활발해진다.

개발(development)에서는 구현, 공유, 개발, 산출하는 단계로서 배운 것을 토대로 새로운 산출물 생산하기 위해 스크래치를 활용하여 프로그래밍에 의한 블록 잇기를 통해 구현한다. 자신의 작품을 발표하고 피드백을 받는다. 역시 디버깅 과정을 거치며 개발과 구현을 반복한다.



[그림 III-1] 스프라이트 이름 설정하기

<표 III-2> 연구지도안의 차시별 활동내용

차시	활동내용
1	• 스크래치의 기초 학습
2	• 스크래치로 접시저울 구현하기(스프라이트 구성)
3	• 스크래치로 접시저울 구현하기(접시저울의 저울대 코딩)
4	• 스크래치로 접시저울 구현하기(접시저울의 추 코딩)
5	• 스크래치로 접시저울 구현하기(접시저울의 접시 코딩)
6	• 스크래치로 접시저울 구현하기(접시저울의 스프라이트 연결 코딩) • 등식의 성질 이해
7	• 등식의 성질 실생활 문제 코딩 발표

2) 연구지도안 예시(제 3차시)

제 1~2차시는 프로그램에 익숙하기 위한 기초학습으로써 학생들이 자유롭게 접시를 구현해보는 과정 중 일차적인 스프라이트를 구성하여 앞으로 사용할 대부분의 블록들을 경험하는 시간을 가졌다. 제 3차시부터는 접시저울에 움직임을 주기 위한 코딩이 본격적으로 시작되었다.

<표 III-3> 연구지도안

차시명	스크래치로 접시저울 구현하기(저울대 코딩)		
학습목표	<ul style="list-style-type: none"> • 스크래치를 활용하여 접시저울을 구현해 낼 수 있다. • 스크래치를 활용하여 접시저울을 구현함을 통해 등식의 성질을 이해할 수 있다. 		
흐름	교수-학습 내용	활동	
도입	• 전시학습 확인: 스크래치에서 접시저울 모양 나타내기	서로 비교하며 참고	
전개	탐구	• 접시저울의 저울대 부분은 어떻게 움직이는지 이야기하기	각자 자신이 생각하는 것 발표
	설계	• 접시저울의 저울대 부분이 어떻게 움직여야 하는지 순서도 작성하기	활동지에 작성
	개발	• 스크래치에서 저울대 스프라이트 코딩하기	서로 비교하고 참고
정리	<ul style="list-style-type: none"> • 차시 마무리: 학습내용 정리 • 차시 예고: 접시저울의 접시 코딩 	자신 컴퓨터에 저장	

3) CT 과정에서의 도구발생 수준

본 연구에서는 코딩수업이 접시저울을 구현하고 도구화가 시작되는 3차시부터 7차시까지 총 5차시 동안 진행한 학습 내용 및 자료인 활동지와 스크래치 결과물을 활용하여 인지적 사고 과정에서의 도구화 수준을 알아보려 하였다. 학생들의 도구화 수준을 0수준부터 4수준으로 분류하였으며 그 틀은 고상숙 외 (2015)에서 총체적 체점기준표에 대한 수준별 구성을 참고하여 수학교육전문가 2인과 박사 후 연구원생의 도움을 받아 아래의 <표 III-4>와 같이 작성되었다. 본 수업은 탐구, 설계, 그리고 개발의 단계로 수업이 진행되므로, 학생들의 활동지와 스크래치 결과물은 <표 III-5>와 같이 정리하여 분류하였으며, 그것을 <표 III-6>와 같이 수준 및 유형별로 구성하였다.

<표 III-4> 도구발생에서 도구화 수준

수준	수준 내용
0	• 순서도를 전혀 그릴 줄 모르고 스크래치를 전혀 다룰 줄 모르는 수준
1	• 나타내고자 하는 것이 어떻게 움직여야 하는지 시도는 하나 순서도를 그리는데 <u>오류가 많으며</u> 그것을 스크래치로 코딩하여 구현하는데 잘못된 수준
2	• 순서도를 그리는데 <u>약간의 오류가 있을 수 있으며</u> 순서도가 주어지면 그것을 스크래치로 코딩하여 구현을 시도하나 큰 진전이 없는 경우 (오류가 있음).
3	• 나타내고자 하는 것의 순서도를 잘 그리고 그것을 스크래치로 코딩하여 구현할 수 있으며 프로그램에 오류가 있는 경우 <u>도움을 받아</u> 디버깅할 수 있는 수준 (약간의 오류가 있음).
4	• 나타내고자 하는 것의 순서도를 잘 그리고 그것을 스크래치로 코딩하여 구현할 수 있으며 프로그램에 오류가 있는 경우도 <u>찾아서 스스로</u> 디버깅할 수 있는 수준

<표 III-5> 학생들의 활동지 및 스크래치 결과물 분류 코드

단계	내용	코드	
탐구 (discovery)	필요한 모든 내용을 작성한 경우	OK	O
	작성을 하였으나 내용이 부족한 경우	Lack	L
	작성을 하지 않은 경우	X	X
설계 (design)	전체 순서도를 다 그린 경우	OK	O
	순서도의 내용은 잘 그렸으나 기호가 틀린 경우	Symbol	SY
	순서도를 간략화 하여 그린 경우	Simple	SI
	순서도의 내용이 부족한 경우	Lack	L
	순서도를 작성하지 않은 경우	X	X

스크래치를 활용한 수학교과 코딩 수업 적용이 컴퓨터 프로그래밍 자기효능감에 관한 연구

개발 (development)	순서도에 맞게 스크래치로 코딩을 잘 한 경우	OK	O
	스크래치로 코딩을 하는 과정에서 한 가지 실수가 있는 경우	Error	E
	스크래치로 코딩을 하는 과정에서 두 가지 이상의 실수가 있는 경우	Error Error	EE
	스크래치로 코딩을 하지 않은 경우	X	X

<표 III-6> 도구발생수준별 유형 코드

수준	코드
0	XXX, LXX, OXX
1	XXEE, XLX, XLEE, XSIX, XSIEE, XSYX, XSYEE, XOX, XOEE, LXEE, LLX, LLEE,
2	LSIX, LSIEE, LSYX, LSYEE, LOX, LOEE, OXEE, OLY, OLEE, OSIX, OSIEE, OSYX, OSYEE, OOX, OOEE, XXE, XXO, XLE, XLO, XSYO, XSYE, LXE, LXO, LLE, LLO, LSYE, LSYO, OXE, OXO, OLE, OLO, OSYE, OSYO
3	XSIE, XSIO, XOEE, LSIE, LOE, XOO, OSIE, OOE
4	LSIO, LOO, OSIO, OOO

4) 컴퓨터 프로그래밍 자기효능감

스크래치를 활용한 수학교과 코딩 수업 적용이 컴퓨터 프로그래밍 자기효능감에 유의미한 영향을 미치는지 확인하기 위하여 자기효능감 검사를 실시하였다. 본 연구에서는 컴퓨터 프로그래밍을 처음 접하는 중학교 1학년 학생을 위해 김갑수(2014)를 수정하여 사용한 이서빈, 고상숙 (2018)의 검사지를 사용하였다. 그 이유는 이 선행연구의 연구대상과 본 연구의 대상자는 모두 중학생들이기 때문이다. 검사지를 ‘언어’, ‘효능감 요인’, ‘효능감 차원’의 3가지 하위영역에 따른 총 30개의 문항으로 구성하고 5단계 Likert 척도를 적용하였다.

<표 III-7> 자기효능감 검사지 구성요소

하위 영역	문항번호(*은 부정문항)	문항 수
언어	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10
효능감 요인	11, 12, 13, 14, 15, 16, 17, 18, 19, 20	10
효능감	21, 22, 23, 24, 25, 26*, 27, 28,	10

<표 III-8> 자기효능감 검사지 신뢰도

하위 영역	Cronbach α		문항 수
	사전검사	사후검사	
언어	0.915	0.966	10
효능감 요인	0.924	0.891	10
효능감	0.752	0.812	10

차원	29*, 30	
Total		30

차원			
Total	0.952	0.956	30

IV. 연구결과

1. 도구화 수준 변화

학생들의 학습과정에서 수집된 산출물에 대해 제 III 장에서 제시한 <표 III-4>에 따라 제 0수준부터 제 4수준까지의 도구발생의 수준을 분석하였고 그 결과 다량의 연구결과물을 산출하였다. 스크래치에 익숙해지는 제 1, 2차시의 준비과정을 통해 등식의 성질에 관해 접시저울의 저울대 코딩이 3차시부터 본격적으로 이루어졌다. 특히 참여한 학생들에 대한 전수조사가 가능했던 것은 학생들이 매차시 자신의 컴퓨터에 활동한 자료를 저장하였기에 연구자는 그들을 매주 수집 가능했기 때문이다. 또한 본 장에서는 각 학생의 순서도와 블록 잇기가 대용량의 이미지 파일들이어서 공간상의 이유로 인해 부록에 예시자료들을 수록하였다(부록 참고).

1) 제 3차시 학생들 변화

학생들이 작성한 활동지와 스크래치 결과물을 분석한 결과 학생들의 CT 과정에서 도구발생 수준은 아래의 <표 IV-1>과 같다.

<표 IV-1> 제 3차시에서 수준분포

수준	0수준	1수준	2수준	3수준	4수준
인원	0명	7명	22명	1명	2명
(%)	(0)	(21.9)	(68.8)	(3.1)	(6.3)

제 3차시에 CT 과정에서 도구발생 수준이 1 수준인 학생들은 총 세 가지 유형(XOEE, LLEE, LXEE)이 나타났으며 가장 많은 인원이 나타난 유형코드는 XOEE로 본인이 나타내고자하는 것이 어떻게 움직여야하는지는 시도는 하나 그것을 순서도로 나타내는 것을 어려워하며 그것을 스크래치로 코딩하는 과정에 많은 오류가 있는 유형이다. 제 2수준인 학생들은 총 22명으로 다섯 가지 유형(OSYO, OLO, OLE, OXO, LLO)이 나타났고 가장 많은 인원이 나타난 유형코드는 OLO(11명)로 본인이 나타내고자하는 것이 어떻게 움직여야 하는지 알고 있으며 그것을 스크래치로 코딩하는 것은 해내지만 순서도를 그리는 것을 어려워하는 유형이다. 제 3수준인 학생은 1명으로 LOE 유형이 나타났으며 이것은 본인이 나타내고자하는 것을 순서도로 나타낼 수 있으며 스크래치로 코딩을 잘하지만 약간의 오류만이 있는 유형이다. 제 4수준인 학생들은 총 두 가지 유형(OOO, LOO)이 나타났으며 이 학생들은 모두 본인이 나타내고자하는 것을 순서도로 잘 나타내었으며, 그것을 스크래치로 코딩을 잘하였다. 제 3 또는 4수준에 도달한 학생은 본 연구에서 처음 시도하는 활동인데도 스크래치의 기능들을 잘 이해한 학생들로 잘 적응하고 있음을 보여주었는데 스크래치가 지니는 도구기제를 잘 이해하고 있는 것으로서 이 도구에 대한 선행적 경험이 많은 도움을 주었다고 볼 수 있다.

2) 제 4차시 학생들 변화

학생들이 작성한 활동지와 스크래치 결과물을 분석한 결과 학생들의 CT 과정에서 도구발생 수준은 아래의 <표 IV-2>과 같다.

<표 IV-2> 제 4차시에서 수준분포

수준	0수준	1수준	2수준	3수준	4수준
인원	1명	3명	21명	2명	5명
(%)	(3.1)	(9.4)	(65.6)	(6.3)	(15.6)

제 4차시에 CT 과정에서 도구화 수준이 0수준인 학생은 1명으로 XXX 유형이 나타났으며 이것은 본인이 나타내고자하는 것이 어떻게 움직여야 하는지 몰라서 순서도도 작성하지 못하고 스크래치로 코딩도 하지 못하는 유형이다. 도구화 수준이 1수준인 학생들은 총 두 가지 유형(LLEE, XOOE)이 나타났다.

LLEE(2명)은 자신이 나타내고자 하는 것의 순서도를 작성하는 것에 어려움이 있었으며 그것을 스크래치로 코딩하는 과정에 많은 오류가 나타났다. 나머지 1명은 나타내고자 하는 것의 순서도는 잘 그러나 코딩은 전혀 하지 못하였다. 이런 경우는 특이한 경우로 블록 잇기에 더 많은 지도가 필요하다. 도구화 수준이 제 2수준인 학생들은 총 다섯 가지 유형(OSYO, OSYE, LSYO, LSYE, LLO)이 나타났으며 가장 많은 인원이 나타난 유형코드는 OSYO(9명)로 본인이 나타내고자 하는 것이 어떻게 움직여야 하는지 잘 알고 그것을 스크래치로 코딩하는 것은 잘 하나 순서도를 작성하는데 있어서 기본적인 오류를 범한 유형이다. 여기서 순서도를 작성할 때 기본적인 오류는 시작이나 끝을 작성하지 않은 경우, 화살표의 방향을 틀린 경우, 순서도의 기호 모양이 틀린 경우, 화살표는 있으나 YES 또는 NO를 작성하지 않은 경우 등이 있다. 도구화 수준이 제 3수준인 학생들은 총 두 가지 유형(XOO, LOE)이 나타났다. 두 학생 모두 본인이 나타내고자 하는 것이 어떻게 움직여야 하는지 설명은 잘 하지 못하였지만 순서도는 보완하는 과정을 통해 잘 그려내었다. 도구발생을 위한 도구화수준이 제 4수준인 학생들은 모두 5명으로 학생들 모두 OOO 유형이 나타났으며 이것은 본인이 나타내고자 하는 것이 어떻게 움직여야 하는지 알고 그것을 순서도로 잘 표현해 냈으며 그것을 스크래치로 코딩을 잘한 유형이다. 본 차시는 지난 차시의 확장으로 도구기제라 하는 스크래치의 특성을 인지하며 수행과정에서 학생들이 어느 정도 내면화를 이루어가고 있음을 의미한다.

3) 제 5차시 학생들 변화

학생들이 작성한 활동지와 스크래치 결과물을 분석한 결과 학생들의 CT 과정에서 도구화 수준은 아래의 <표 IV-3>과 같다.

<표 IV-3> 제 5차시에서 수준분포

수준	0수준	1수준	2수준	3수준	4수준
인원	1명	9명	17명	2명	3명
(%)	(3.1)	(28.1)	(53.1)	(6.3)	(9.4)

제 5차시에 CT 과정에서 도구화 수준이 0수준인 학생은 유형(LXX)이 1명으로 나타났으며 학생 스스로 나타내고자하는 것이 어떻게 움직이어야 하는지 시도는 하나 그것을 순서도로 나타내고 스크래치로 코딩하는 것을 전혀 할 줄 모르는 유형이다. 도구화 수준이 제 1수준인 학생

들은 총 다섯 가지 유형(LXEE, XSYEE, XOX, LLEE, XLEE)이 나타났으며 가장 많은 인원이 나타난 유형 코드는 LXEE(3명)이다. LXEE는 본인이 나타내고자 하는 것이 어떻게 움직여야 하는지 시도하나 그것을 순서도로 나타내는 것과 그것을 스크래치로 코딩하는 것을 어려워하는 유형으로 차이점은 순서도를 제시하는 경우와 제시하지 않은 경우로 나타났다. 도구화 수준이 제 2수준인 학생들은 총 여섯 가지 유형(LSYO, LSYE, LLO, LLE, XSYE, XXO)이 나타났으며 가장 많은 인원이 나타난 유형코드는 LSYO(7명)로 본인이 나타내고자 하는 것이 어떻게 움직여야 하는지 자세히 알지는 못하여 순서도

를 그리는데 작은 실수가 있지만 나타내고자 하는 것을 스크래치로 코딩하는 것은 잘하는 유형이다. 이런 학생은 순서도가 굳이 필요하다고 생각하지 않은 학생이다. 도구화 수준이 제 3수준인 학생들은 총 두 가지 유형(LOE, LSIE)이 각 한 명씩 나타났다. 두 학생 모두 본인이 나타내고자 하는 것을 순서도로 잘 표현해냈지만 그것을 스크래치로 코딩하는 부분에서는 한 가지 씩의 오류가 나타났다. 이 두 학생 중 한 명은 순서도를 정석대로 그려낸 반면 다른 한 명의 학생은 계속 반복적으로 나타나는 것을 간략화(수정)하여 순서도를 그려내었다. 도구화 수준이 4수준인 학생들은 모두 3명으로 각 학생이 한 가지씩 유형을, 총 세 가지(OOO, LOO, LSIO)가 나타났다. 이 세 학생은 약간의 차이를 보였지만 모두 본인이 나타내고자 하는 것의 순서도를 잘 그려냈으며 그것을 스크래치로 코딩하는 것도 잘 해내었다.

4) 제 6차시 학생들 변화

학생들이 작성한 활동지와 스크래치 결과물을 분석한 결과 학생들의 CT 과정에서 도구화 수준은 아래의 <표 IV-4>와 같다.

<표 IV-4> 제 6차시에서 수준분포

수준	0수준	1수준	2수준	3수준	4수준
인원 (%)	2명 (6.3)	3명 (9.4)	16명 (50)	8명 (25)	3명 (9.4)

제 6차시에 CT 과정에서 도구화 수준이 제 0수준인 한 가지 유형(XXX)이 나타났으며 이것은 본인이 나타내고자 하는 것이 어떻게 움직여야 하는지 몰라서 순서도를 작성하지 못하고 스크래치로 코딩도 하지 못하는 유형이다. 도구화 수준이 제 1수준인 학생들은 총 두 가지 유형(XOEE, LXEE)가 나타났다. LXEE(2명)으로 순서도를 작성하는 과정에서는 기본적인 오류가 있었으며 스크래치로 코딩하는 과정에서도 오류가 발생하였고 XOEE(1명)은 본인이 나타내고자 하는 것을 순서도로 잘 작성하였으나 스크래치로 코딩하는 과정에서 많은 오류가 발생하였다. 도구화 수준이 제 2수준인 학생들은 총 세 가지 유형(OSYO, OSYE, XXE)이 나타났으며 가장 많은 인원이 나타난 유형코드는 OSYO(8명)은 나타내고자 하는 것이 어떻게 움직여야 하는지 그리고 그것을 스크래치로 코딩하는 것도 잘 하나 순서도를 그리는데 있어서 기본적인 오류를 범한 유형이다. 도구화 수준이 제 3수준인 학생들은 총 다섯 가지 유형(OOE, OSIE, LOE, LSIE, XOE)이 나타났으며 가장 많은 인원이 나타난 유형코드는 OOE(4명)으로 본인이 나타내고자 하는 것이 어떻게 움직여야 하는지 알고 그것을 간략화 하여 순서도로 나타냈으며 그것을 스크래치로 코딩할 때 한가지씩의 오류가 있는 유형이다. 도구화 수준이 제 4수준인 학생들은 모두 OOO유형(3명)이 나타났으며 이것은 학생이 나타내고자 하는 것이 어떻게 움직여야 하는지 알고 그것을 순서도로 잘 표현해 냈으며 그것을 스크래치로 코딩을 잘한 유형이다. 특히 지난 제 5차시와 비교해보면 제 3수준의 학생 수가 증가하였음을 알 수 있다.

5) 제 7차시 학생들 변화

본 차시는 종합적으로 정리하는 과정으로 학생들이 작성한 활동지와 스크래치 결과물을 분석한 결과 학생들의 CT 과정에서 도구화 수준은 아래의 <표 IV-5>과 같다.

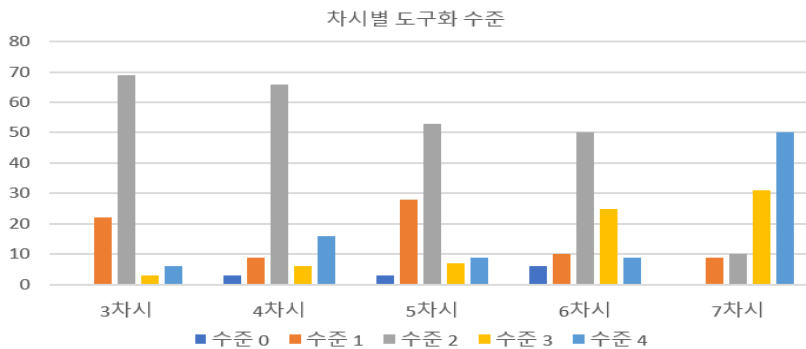
<표 IV-5> 제 7차시에서 수준분포

수준	0수준	1수준	2수준	3수준	4수준
인원 (%)	0명 (0)	3명 (9.4)	3명 (9.4)	10명 (31.3)	16명 (50)

도구화 수준이 제 1수준인 학생들은 총 두 가지 유형(XOEE, XSYEE)가 나타났다. XOEE(2명)는 학생이 나타내고자 하는 것을 순서도로 그리나 스크래치로 코딩하는 과정에서 두 가지 이상의 오류가 발생하였고,

XSYYE(1명)으로 순서도를 작성하는 과정에서 기호에 대한 오류와 스크래치로 코딩하는 과정에서도 오류가 발생하였다. 도구화 수준이 제 2수준인 학생들은 총 두 가지 유형(OSYEE, OLEE)이 나타났으며 가장 많은 인원이 나타난 유형코드는 OSYEE(2명)은 나타내고자 하는 것이 어떻게 움직여야 하는지 알지만 순서도에서 몇 개의 기호가 틀리고 스크래치를 완성하는 데 오류를 범한 유형이다. 도구화 수준이 제 3수준인 학생들은 총 네 가지 유형(OSIE, XOE, XSIO, LOE)으로 나타났는데 익숙해짐에 따라 첫 단계를 생략하거나 부족하지만 바로 순서도 작성을 시도한 경우도 포함된다. 가장 많은 인원이 나타난 유형코드는 OSIE(6명)으로 본인이 나타내고자 하는 것이 어떻게 움직여야 하는지 알고 그것을 간략화 하여 순서도로 나타냈으며 그것을 스크래치로 코딩할 때 한가지씩의 오류가 있는 유형이다. 도구화 수준이 제 4수준인 학생들은 4가지 유형(OSIO, LSIO, LOO, OOO)이 나타났으며 OSIO(7명)으로 이것은 학생이 나타내고자 하는 것이 어떻게 움직여야 하는지 모르므로 그것을 순서도로 간략하게 표현하고 그것을 스크래치로 코딩을 잘한 유형이다. 특히 7차시에는 그동안 배웠던 것을 활용하는 과정으로 학생들 대부분 제 3수준과 제 4수준의 학생이 전체의 80%이상으로 초기에 비해 크게 향상되었음을 알 수 있다. 이 결과는 지난 제 6차시 과정에서 시간부족으로 인해 재인지 과정에 어려움이 있었는데 이를 보완하는 과정을 통해 스크래치 활용에 의한 등식의 성질을 잘 마무리한 것으로 보인다.

종합적으로 이들을 하나의 그래프로 차시별 수준을 요약해보면 다음 그림과 같다. 학생들이 연구시작 점에서 1수준에서 2수준에 속한 수가 컸던 것을 보면 스크래치라는 도구가 사용하기에 그리 어렵지 않음을 나타내는 것이며 차시가 진행될수록 스크래치의 도구기제에 대한 이해도가 필요하고 이에 따른 순서도 작성이 병행하여 이루어지면서도 학생의 수준이 2수준에서 3 또는 4수준으로 이동하고 있음을 보여주고 있다. 제 7차시는 마무리하는 시간으로서 실생활에 적용하는 문제에서 동료학생들이 발표하는 시간이 주어졌는데 이 때 자신의 것도 수정하는 과정을 통해 도약적인 수준이동을 보이고 있다. 좀 더 연구기간을 확장하였다면 이런 상위수준으로의 향상이 안정적으로 이루어졌을 것으로 예측된다.



[그림 IV-1] 학생들의 도구화 수준변화

2. 학생들의 컴퓨터프로그래밍 자기효능감

실험반의 집단 내 변화 분석을 위해 SPSS 25.0 프로그램을 이용하여 수집된 중학교 1학년 학생들의 사전-사후 자료에 대한 대응표본 t-검정을 실시하였으며 그 결과는 <표 IV-7>와 같다. 대응표본 t-검

정 결과에 의하면 중학교 1학년 학생들의 컴퓨터 프로그래밍 자기효능감은 전체에 대하여 통계적으로 유의미한 차이가 있는 것으로 나타났으며($p=0.012$), 각 하위영역에서는 ‘효능감 차원($p=0.385$)’을 제외한 나머지 다른 하위영역에 대하여 $p<0.0167(=0.05/3)$ 보다 낮으므로 통계적으로 유의미한 향상이 있음을 나타냈다(언어 $p=0.008$; 효능감 요인 $p=0.002$). 즉, 본 연구에서 개발한 스크래치를 활용한 수학교과 코딩수업이 중학교 1학년 학생들의 컴퓨터 프로그래밍 자기효능감을 향상시키는데 긍정적인 역할을 하였으며, 특히 컴퓨터 프로그래밍의 이해와 활용을 바탕으로 하는 언어적 측면과 경험을 바탕으로 하는 효능감 요인의 측면을 향상시키는 데 유의미한 효과가 있었음을 나타낸 것이다.

<표 IV-6> 컴퓨터 프로그래밍 자기효능감에 대한 사전-사후검사 평균 및 표준편차

하위 영역	기준	N	평균	표준 편차
언어	사전	30	2.80	0.91
	사후	30	3.54	0.94
효능 감 요인	사전	30	2.78	0.99
	사후	30	3.53	0.83
효능 감 차원	사전	30	3.22	0.70
	사후	30	3.39	0.67
총계	사전	30	2.93	0.81
	사후	30	3.49	0.74

<표 IV-7> 컴퓨터 프로그래밍 자기효능감에 대한 사전-사후검사의 대응표본 t-검정 결과

하위 영역	대응차					t	자유도	p 유의 확률 (양 쪽)
	평균	표준 편차	표준 오차 평균	차이의 95% 신뢰구간				
				하한	상한			
언어	-0.74	1.418	0.259	-1.269	-0.211	-2.859	29	0.008
효능감 요인	-0.76	1.195	0.218	-1.203	-0.310	-3.467	29	0.002
효능감 차원	-0.17	1.057	0.193	-0.565	0.225	-0.881	29	0.385
총계	-0.56	1.133	0.207	-0.979	-0.133	-2.686	29	0.012

V. 결론 및 시사점

1. 결론

본 연구는 스크래치를 활용한 수학교과 수업을 개발하고 이를 적용하여 학생의 도구화 수준과 컴퓨터 프로그래밍 자기효능감에 대해 조사하였다. 본 연구에서 개발중심모델(D^3 모델) 수업모형에 따른 수업은 자유학년제를 실시하고 있는 중학교 1학년 수업시간에 제공되었는데 이 과정에서 나타나는 각 학생들의 도구화수준의 변화를 매 차시 학생이 저장한 파일을 수집하여 분석하였고, 학생들의 자기효능감은

컴퓨터 프로그래밍 자기효능감 검사지를 활용하여 실험처치 사전 및 사후 자료를 통해 대응표본 t-검정으로 분석하였다. 본 연구의 결과를 바탕으로 다음과 같은 결론을 도출하였다.

첫째, 스크래치를 활용했을 때 학생들의 도구화 수준은 [그림 IV-1]에 나타나는 것처럼 매우 다양하게 나타났다. 좀 더 자세히 살펴보면 도구화가 이뤄지기 시작한 3차시에서는 제 3수준과 제 4수준에 해당하는 학생이 9%에 불과하였는데 수업이 완료되는 시점인 7차시에서 이들 수준에 해당하는 학생들은 80%이상인 것으로 나타났다. 또한 스크래치를 다뤄본 경험이 있었던 두 학생들은 늘 수업내용에 좋은 아이디어나 연구자의 질문에 능동적으로 답을 하였고 제 4수준의 결과물을 곧잘 제시하였지만 그렇지 못한 학생은 시간이 지나면서 향상되는 수준이 서서히 나타났다. 즉 도구화는 도구기제를 활용하여 이루어지기 때문이다. 예를 들어 양팔접시의 저울대가 평형을 유지하기 위해서는 기본적으로 0도가 아닌 90도 좌회전 시에 이루어진다는 스크립트에 의한 스프라이트의 작동원리를 이해해야 하고, 여기에 좌우접시에 올려놓는 추의 평형을 염두에 두기 위해서는 알고리즘의 구성에 대한 이해가 필요하다. 이 도구기제 측면(도구가 가지고 있는 특징)을 이용해서 도구화를 이루고 여기에 순서도로 표현해보는 것은 학생들의 사고과정을 논리적으로 반성하는 계기가 되며 스크래치로 작동하는 과정에서 이를 재확인한다. 이런 공학 도구를 사용하면 위 과정이 순환적 뿐만 아니라 역동적으로 이루어지므로 재방문(revisit)이 활성화된다. 따라서 도구에 대한 선수 경험을 지닌 학생들은 이런 도구기제 측면이 쉽게 이해되었다고 보이며 대부분의 학생들은 시간이 갈수록 도구기제를 바탕으로 도구화 수준이 향상되어 등식의 성질에 대한 행동 스킴이 점차적으로 강화되었다고 하겠다. 다만 이러한 수준 향상은 이 학생들이 중학교 1학년이므로 앞으로 더 많은 기회를 통해 잘 이루어나갈 수 있을 것으로 예측된다.

특히 본 연구에서 학생 각자가 컴퓨터 모니터에서 자신이 사용할 교구(양팔저울)를 직접 제작해보는 과정과 나아가 저울대의 평형을 유지해야하는 부분에서 등식의 성질을 파악하게 되는 활동들은 단순히 양팔저울이 교구로 주어진 환경에서의 수업과 매우 다른 측면이다. 왜냐하면 이미 마련된 양팔저울을 통해 등식의 성질을 학습하는 전통적 수업에서는 양팔저울의 구성원리를 생각해 볼 필요가 없기 때문이다. 바로 이런 측면은 본 연구에서처럼 순서도를 바탕으로 스크래치의 코딩에 의해 이루어지므로 이런 CT의 논리적 추론에 의해 블록 잇기가 더욱 단순화되는 과정이 발생가능하게 된다. 즉, 등식의 성질이란 수학적 개념이 CT와 융합되는 과정으로 수학적 추상화와 CT에서의 추상화가 교집합으로 작용하는 것이라 할 수 있겠다(신동조 외, 2019 참고).

둘째, 본 연구에서 개발한 스크래치를 활용한 중학교 1학년 수학교과 코딩수업은 학습자인 중학교 1학년 학생들의 컴퓨터 프로그래밍 자기효능감에 긍정적인 영향을 미친 것으로 나타났으며 ‘언어’, ‘효능감 요인’의 하위영역에 대하여 유의미한 차이가 있었다. 특히 컴퓨터 프로그래밍에 대한 이해와 적용, 활용과 같은 언어적 능력이 크게 함양되었다. 또한 본 연구에서 개발한 수업을 통해 컴퓨터 프로그래밍을 직접적으로 경험하면서 컴퓨터 프로그래밍에 대한 가치와 실용성을 알게 되었으며, 이를 바탕으로 효능감 요인에 따른 컴퓨터 프로그래밍 자기효능감이 향상되었다고 볼 수 있다. 다만 효능감 차원에 해당되는 문항에는 “나는 컴퓨터 프로그래밍 언어를 학습할 때 배우지 않은 것도 알고 싶다” 또는 “나는 학습한 컴퓨터 프로그래밍 언어의 기능을 높은 수준으로 확장하고 싶은 마음이 든다.”와 같은 문항에서의 높은 긍정적인 변화가 가능하려면 도구가 가지는 특정한 절차를 따르는 것으로써 스크래치가 갖는 특성(도구기제 측면)을 잘 이해할 뿐만 아니라 오락을 즐기듯 이런 도구를 자기 몸의 분신처럼 즐기면서 하고자 할 때(도구화 측면) 도구발생이 견고히 이루어지며 나아가 더 적극적으로

로 활용하고 싶은 학습의욕이 증만해지는 것이 필요한데 총 7차시의 수업을 통해 이루어지기는 무리가 있어 보인다.

2. 시사점

본 연구는 수학교육에서 방정식의 문제를 해결함에 있어 단순히 지필로만 해결하는 것이 아니라 공학도구, 스크래치를 사용해서 CT과정을 활성화하여 방정식의 해결과정을 시각화하였다. 중학교 과정에서 CT(컴퓨팅 사고)에 대한 몇 안 되는 연구로서 다음과 같은 시사점을 얻을 수 있었다.

첫째, 디버깅 과정의 중요성이 재인식될 필요가 있다. 기술공학을 활용할 때 일반적으로 정확한 답을 구할 수 있으니 쉽게 접근이 가능하다고 생각한다. 그러나 프로그래밍(코딩)과정에서는 더 나은 코딩과정이 얼마든지 있을 수 있으며, 자신의 코딩이 완벽하지 않은 경우가 발생할 수 있음을 인지하고 언제든지 이를 수정하는 과정인 디버깅 과정을 거치게 된다. 이러한 디버깅 과정은 도구발생에 기여하는 도구기제에 의한 요인일 수도 있다. 앞서 언급되었듯이 본 연구에서는 스크래치가 가지고 있는 도구기제, 예를 들면 양팔접시의 평형을 유지하기 위해서는 90도 좌회전에 기초를 둔다는 것이며 이와 더불어 사용자의 도구화에 기여하는 수학적 알고리즘 지식의 이해정도에 따른다. 다시 말해 서로 연결 가능한 블록들을 붙이는 과정은 사용자가 생각하고 그려내는 순서도에 따라 위계성에 맞게 연결하고 그럼으로써 재방문이 사이사이에 필요에 따라 이루어지게 된다. 순서도가 복잡할수록 연결되는 블록의 길이도 길어진다. 이 때 좀 더 잘하는 친구의 단순화된 블록들을 참고하며 자신의 것을 수정하는 과정도 디버깅 과정에 포함된다. 이 디버깅 과정은 코딩을 완성하는 매우 중요한 절차이고 수학적 알고리즘의 적용성이 능률적으로 이루어지도록 돕는다. 최단거리로 이를 완성하게 되면 높은 수준의 도구발생이 이루어진 것인데 바로 도구기제와 도구화가 서로 상호보완적으로 교류하여 진행되는 것이다. 학생은 도구의 유용한 방향성을 유지하게 되고 수학적 알고리즘에 의해 코딩과정의 단순화도 가능하다. 이런 코딩과정에서 창의, 융합적 역량과 협업 및 팀워크 역량이 육성될 수 있다. 본 연구에서는 이런 부분을 연구문제에 포함하지 않았으므로 후속연구에서는 정성적 분석을 통한 사고과정을 구체적으로 규명하길 기대한다. 특히 선수 경험을 지닌 두 명의 학생이 본 연구에서 늘 높은 수준을 나타냈던 것을 보면 일찍 스크래치라는 도구에 노출되는 것이 위에 열거한 경험들에 의해 도움이 되었다.

둘째, 프로그래밍 언어를 활용한 수학 수업을 할 때에는 메타인지 이동이 일어나지 않도록 주의해야 한다. 본 지도안에서는 등식의 성질을 시각화하는데 등호를 형상화한 양팔 접시의 균형(balance)을 코딩하는 과정이 중요하게 사용되었고, 그러한 양팔접시를 통해 방정식의 문제를 해결하도록 하는 것이 도구발생의 목적이었다. 그러나 코딩과정을 명확히 해주는 순서도를 등한시하고 블록들의 연결만을 초점을 두면 메타인지의 이동이 일어날 수 있다. 이런 경우를 예방하기 위해서는 활동지의 역할이 중요하다. 활동지에서 문제를 해결하는 과정을 안내하는 질문들이 학생들의 진행방향을 견지하도록 구성되어야 한다.

셋째, 스크래치를 활용한 수학교과 코딩 수업이 초등학교뿐만 아니라 중등학교로 확장하고 이에 대한 연구가 활발히 이루어져야 한다. 예를 들어 큰 수를 다루는 지수함수나 로그함수 등을 이용하는 복잡한 실생활 문제들로도 확장할 필요가 있다.

참고 문헌

- 강하람, 임채령, 조한혁 (2021). 수학교과와 정보교과를 융합하는 코딩수학 교육과정 및 교육방법 연구. **수학교육**, 60(4), 467-491.
- 고상숙, 고호경, 박만구, 한혜숙, 홍예윤 (2015). **수학교육평가론**. 서울: 경문사.
- 고상숙, 박만구, 한혜숙 (2013). 교구 및 공학도구를 활용한 수학적 과정중심 평가에 관한 교사들의 인식. **한국학교수학회논문집**, 16(4), 675-694.
- 고영해, 안재호, 박남제 (2011). LOGO 교육용 프로그래밍 언어를 이용한 프랙탈 기하이론 기반의 초등학교 컴퓨터교육 지도 방안. **한국정보기술학회논문지**, 9(8), 151-163.
- 교육부 (2016). **초중고교 코딩교육 보도 관련**. 교육부 보도자료 (2016. 10. 06.)
- 권점례, 이광상, 김성경 (2016). **지능정보사회 대비 학교 교육에서 코딩기반 소프트웨어 교육의 적용 가능성 탐색**. 한국교육과정평가원 연구보고 RRC 2016-9.
- 김갑수 (2014). 컴퓨터 프로그래밍 언어 교육에서 자기 효능감 척도 개발 및 적용. **정보교육학회논문지**, 18(1), 111-120.
- 김소민 (2021). 이차곡선 수업에서 공학도구 사용과 수작업 교구 활동에 대한 예비 수학교사들의 인식. **한국학교수학회논문집**, 24(1), 151-172.
- 김원경, 조민식, 방금성, 김수미, 배수경, 오혜정 외 (2013). **중학교 수학 1. 비상교육**.
- 김진숙 (2015). **2015년 교육정책네트워크 교육현장지원연구: SW교육 교수학습모델 개발 연구**. 연구보고 CR 2015-35. 한국교육개발원 .
- 남진영 (2015). 수학과 국가교육과정의 정의적 영역 목표 고찰. **한국초등수학교육학회지**, 19(2), 159-178.
- 송정범 (2017). 스크래치 활용 게임 프로그래밍 학습이 수학교과 흥미와 가치인식에 미치는 영향. **정보교육학회논문지**, 21(2), 199-208.
- 신동조, 고상숙 (2019). 수학교육에서 계산적 사고(Computational Thinking)의 의미 및 연구동향 탐색. **수학교육**, 58(4), 483-505.
- 유정호 (2015). **수학 내용 기반의 코딩 교육 프로그램 개발**. 서울교육대학교 석사학위 논문.
- 이상구 (2015). **수학과 컴퓨팅 사고력(CT)**. 대한수학교육학회 2015년 연보: 수학교육에서 공학적 도구(pp.368-384). 서울: 경문사.
- 이서빈, 고상숙 (2018). 파이썬을 활용한 수학교과 코딩수업(DM³)의 효과. **수학교육학연구**, 28(4), 479-499.
- 이환철, 김형원, 이지혜, 이현숙, 고호경 (2017). 수학학습 정의적 영역 검사 도구 개발 연구. **학교수학**, 19(2), 267-287.
- 임현정, 고상숙 (2016). GeoGebra 를 활용한 반힐레 기하교수법에서 도구화에 관한 연구. **수학교육논문집**, 30(4), 435-452.
- 조혜정, 김인수 (2016). 수학 학습에서의 정의적 영역에 관한 국내 연구 동향 분석. **수학교육논문집**, 30(1), 67-83.
- 한세호, 장경윤 (2009). 고등학교 수학 문제해결에서 CAS의 도구발생. **학교수학**, 11(3), 527-546.
- 황정규 (2002). **학교학습과 교육평가**. 서울: 교육과학사.
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2), 191-215.

- Bandura, A. (1986). *Social foundation of thought and action: A social cognitive theory* Prentice Hall. Englewood Cliffs, NJ.
- Carpenter, T. P., Franke, M. L. & Levi, S. (2003). *Thinking mathematically: Integrating arithmetic and algebra in elementary school*. Heinemann.
- Drijvers, P. M. (2003). Algebra on screen, on paper and in the mind. In J. T. Fey, A. Couco, C. Kieran, L. McMullin, & R. M. Zbiek(Eds.), *Computer Algebra System in Secondary School Mathematics Education*(pp. 241-267). New York: Springer.
- Krathwohl, D. R., Bloom, B. S., & Masia, B. B. (1964). *Taxonomy of Educational Objectives: The Classification of Educational Goals, Handbook 2: Affective Domain*. New York: McKay.
- Lindsley, D. H., Brass, D. J., Thomas, J. B. (1995). Efficacy- performance spirals: A multilevel perspective. *Academy of Management Review*, 20(3), 645-678.
- McLeod, D. B. (1989). Beliefs, Attitudes, and Emotions: New Views of Affect in Mathematics Education. In D. B. McLeod, & V. M. Adam(Eds.), *Affect and Mathematics Problem Solving: A New Perspective*(pp. 245-258). New York: Springer Verlag.
- McLeod, D. B. (1992). Research on affect in mathematics education: A re-conceptualization. In D. A. Grouws (Ed.), *Handbook Research on Mathematics Teaching and Learning* (pp. 575-596). New York: Macmillan.
- Mitchell, T. R., Hopper, H., Daniels, D., George-Falvy, J., James, L. R. (1994). Predicting self-efficacy and performance during skill acquisition. *Journal of Applied Psychology*, 79(4), 506-517.
- Resnick, M. (2019). *The Next Generation of Scratch Teaches More Than Coding*. EdSurge . (<https://www.edsurge.com/news/2019-01-03-mitch-resnick-the-next-generation-of-scratch-teaches-more-than-coding>).
- Salomon, G. (1984). Television is "easy" and print is "though": The differential investment of mental effort in learning as a function of perceptions and attributions. *Journal of Educational Psychology*, 76(4), 647-658.
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). *Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework*. *Education and Information Technologies*, 18.
- Trouche, L. (2004). Managing the complexity of human/machine interaction in a computerized learning environments : Guiding students' command process through instrumental orchestrations. *International Journal of Computers for Mathematical Learning*, 9, 281-307.
- Trouche, L. (2005). *Calculators in Mathematics Education: A Rapid Evolution of Tools with Differential Effects*. New York: Springer.
- Vergnaud, G. (1996). The Theory of Conceptual Fields. In L. Stette, P. Nesher, P. Cobb, G. A. Goldin, & B. Greer (Eds.), *Theories of Mathematical Learning*(pp. 219-240). Mahwah, NJ: Lawrence Erlbaum Associates.

<부록>

수업 6차시 제 3수준		
<p>✓ 저물대 왼쪽 ↓</p> <p>✓ 오른쪽 ↑</p> <p>✓ 추가 오른쪽, 즉 올라감.</p> <p>✓ 저물대 오른쪽 ↓</p> <p>✓ 왼쪽 ↑</p> <p>✓ 추가 왼쪽, 즉 올라감.</p>	<p>✓ 왼쪽 경사가 불리해야 한다</p> <p>✓ 추가 오른쪽 경사 불리하면 추가 불리해진다</p> <p>✓ 왼쪽 경사가 불리해진다</p> <p>✓ 특정 왼쪽 경사에 이르렀을 추가 불리하게 한다.</p>	
[그림 1] 탐구단계 O 예시	[그림 2] 탐구단계 L 예시	[그림 3] 설계단계 O 예시

[그림 4] 탐구단계 SI 예시	[그림 5] 개발단계 E 예시

수업 6차시 제 4수준		
<p>✓ 저물대가 오른쪽 경사면 왼쪽 경사가 불리해진다</p> <p>✓ " 추가 오른쪽 경사에 있으면 추가 불리해진다</p> <p>✓ 저물대가 오른쪽 경사면 오른쪽 경사가 불리해진다</p> <p>✓ " 추가 왼쪽 경사에 있으면 추가 불리해진다</p>		
[그림 6] 탐구단계의 O 예시	[그림 7] 설계단계의 O 예시	[그림 8] 개발단계의 O 예시

A Study on Instrumentalization Levels and Computer Programming Self-efficacy in a Mathematics Classroom Using Scratch: Focused on the Property of Equality

Lee, Hyun Ji²⁾ · Choi-Koh, Sang Sook³⁾

Abstract

The study investigated students' instrumentalization levels and computer programming self-efficacy in mathematics classrooms while using Scratches, to understand the properties of equality. 32 of 7th-grade students from D middle school in Gyeonggi-do participated in the program consisting of 7 lesson units. To investigate individual students' levels of instrumentalization, each worksheet they worked on using Scratches was saved into computers after each lesson. Questionnaires measured self-efficacy regarding computer programming at the study's beginning and the end. The level of students' instrumentalization was revealed to be variously from level 0 to 4. In the beginning, 9% of students corresponded to level 3 or 4, but more than 80% of students reached level 3 or above at the end. In addition, computer programming self-efficacy was improved significantly.

Key words : Computational thinking, Property of Equality, Programming(coding), Instrumentalization, Instrument Genesis

Received November 04, 2022

Revised December 09, 2022

Accepted December 11, 2022

* 2010 Mathematics Subject Classification : 97D40

2) Graduate School of Education, Dankook University (hyunji@dankook.ac.kr)

3) Dankook University (sangch@dankook.ac.kr), Corresponding Author