

수학교과와 정보교과를 융합하는 코딩수학 교육과정 및 교육방법 연구

강하람(서울대학교 대학원, 학생) · 임채령(서울대학교 대학원, 학생) · 조한혁(서울대학교, 교수)[†]
†교신저자

A study on coding mathematics curriculum and teaching methods that converges school mathematics and school informatics

Kang, Ha Ram(Seoul National University, Graduate School, haram926@snu.ac.kr)

Lim Chae Lyeong(Seoul National University, Graduate School, chae1004@snu.ac.kr)

Cho, Han Hyuk(Seoul National University, hanchu@snu.ac.kr)[†]

†Corresponding Author

초록

본 연구는 초·중학교 수학교과와 정보교과를 융합하는 코딩수학 교육과정과 이를 위한 최소 코딩게임 기반 교육방법에 대한 연구이다. 지난 3년간 코딩수학 교육과정과 효과적인 교육방법을 초6학년과 중1학년 학생을 대상으로 연구하였다. 1차년도 연구결과, 공간좌표의 필요성에 따라 3차원 좌표의 수학적 개념을 포함하는 코딩환경으로 교육과정을 수정하였다. 2차년도 연구결과, 명령어의 위계성에 따라 건물 요소별 다른 수준의 명령어를 도입하여 자기주도적 학습이 가능하도록 개선하였다. 3차년도 연구 결과, 컴퓨팅 사고력 향상을 유도하는 최소 코딩게임 기반의 교수·학습 전략을 설계하고, 컴퓨팅 사고력 진단을 위한 평가 및 피드백을 개발하였다. 자기주도적 학습 및 컴퓨팅 사고력 증진을 유도하는 최소 코딩게임 기반 교육방법과 코딩수학 교육과정은 수학-정보교과의 융합교육 연구와 실천에 의미가 있다.

Abstract

This study is a study on the coding mathematics curriculum that converges elementary and middle school mathematics and information subjects and a minimum coding game-based education method for this. For the past 3 years, the coding mathematics curriculum and educational methods to effectively operate the curriculum were studied by applying them to 6th graders of elementary school and 1st graders of middle school. As a result of the first year of research, the coding mathematics curriculum was modified to a coding environment including the mathematical concept of a three-dimensional coordinate space, and the three-dimensional object was improved to be output as a real 3D print. As a result of the 2nd year study, it was improved so that even low-level students can build buildings by introducing different level commands for each component of the building so that self-directed learning is possible. As a result of the 3rd year study, a teaching-learning strategy based on a minimal coding game was designed to induce an increase in the level of computational thinking, and evaluation and feedback for diagnosing computational thinking were developed. Educational methods to promote self-directed learning and computing thinking ability, and researched coding mathematics curriculum are meaningful for the research and practice of the convergence education of school mathematics and informatics.

* 주요어 : 컴퓨팅 사고력, 코딩수학, 3차원 코딩환경, 최소 코딩게임, 수준 상승, 피드백

* **Key words** : computational thinking, coding mathematics, 3D coding environment, minimum coding game, level up, feedback

* **Address**: Department of Mathematics Education, Seoul National University, Seoul, Korea

* **2000 Mathematics Subject Classification** : 97U50

* **Received**: August 9, 2021 **Revised**: August 14, 2021 **Accepted**: August 29, 2021

I. 서론

4차 산업혁명 시대가 도래함에 따라 전 세계적으로 SW(소프트웨어)의 중요성이 대두되고 있다. 이에 세계의 여러 나라는 소프트웨어 교육을 교육과정으로 도입하며 단순 코딩 능력을 넘어 컴퓨팅 사고력 교육에 힘쓰고 있다(Kwon, 2018). 이러한 추세에 맞추어 우리나라에서도 2018년부터 학생이 소프트웨어에 대한 기초소양을 충실히 갖추어 나갈 수 있도록 소프트웨어 교육 중심의 중학교 정보교과를 필수 과목으로 지정하였으며, 2019년부터 초등학교는 ‘실과’ 과목의 단원으로 코딩교육을 실시하고 있다(Ministry of Education, 2015a). 2015 개정 중학교 정보과 교육과정에는 ‘문제해결과 프로그래밍’ 영역 내용 체계의 핵심 개념으로 알고리즘이 있으며, 프로그래밍 단원의 내용 요소에 변수와 연산, 제어 구조가 포함되어 있다. 정보교과의 내용 요소에 있는 변수는 중학교 수학과 교육과정의 ‘함수’ 영역의 학습 요소에서도 좌표, 순서쌍, 좌표평면, 함수와 함께 포함된 용어로 코딩과 수학에서 모두 중요한 개념이다.

또, Ministry of Education(2020)은 정보 지능 기술을 활용할 수 있도록 인공지능을 학교 교육에 도입하기로 하면서 2021년 2학기부터 고등학교 수학과 선택과목으로 ‘인공지능 수학’ 과목을 도입하였다. 이 과목의 ‘인공지능과 수학’ 영역에서는 제6차 수학과 교육과정의 학습 요소였던 ‘알고리즘과 순서도’ 중 순서도가 부활하였으며, 알고리즘은 또한 중학교 정보과 내용에서도 핵심 개념으로 등장한다. 나아가 ‘인공지능과 수학’에서는 일반화된 지식으로 수학이 코딩을 이용하는 인공지능 기술 전반에 활용되고 있음을 설명하고 있고, 성취기준에서는 ‘인공지능에 수학이 활용되는 다양한 예를 찾을 수 있다.’가 포함된 등 정보교과의 코딩교육에서 수학교과의 다양한 수학적 개념이 사용되고 있음이 초·중·고등학교 전반에 걸쳐 나타나고 있다.

4차 산업혁명 시대에 소프트웨어 교육의 중요성과 더불어 Ministry of Education(2015a)는 2015 개정 교육과정의 총론에서 추구하는 미래의 인간상으로 ‘창의·융합형 인재’를 강조하였으며, 교육부 장관은 2015 개정 교육과정의 개정 배경으로 ‘창조경제 사회에서는 유연하고 창의적인 사고력, 서로 다른 지식을 융합·활용할 수 있는 창의

융합형 인재 양성’을 말하였다. 이처럼 창의·융합형 인재 양성을 위해서는 현재 교육과정에 서로 다른 지식을 융합하는 교육과정 개발의 필요성이 강조되고 있으며, 본 연구팀은 앞서 언급한 정보교과와 수학교과의 밀접한 관련성을 바탕으로 이를 융합하는 코딩수학 교육과정 개발 연구와 교육방법에 관심을 두고 연구하게 되었다.

먼저, 교육과정의 의미는 다양한 방식으로 규정되고 있는데, 가르치고자 의도하는 교육 내용을 지칭하는 전통적인 방식이 있는가 하면, 수업과 평가 등 교육 내용을 가르치는 전체적 과정을 포괄적으로 지칭하기도 한다(Yang, 2002). 본 연구에서 사용하는 교육과정은 교육 내용뿐만 아니라 수업에서 일어나는 상호작용, 수업 후 평가, 평가에 대한 피드백 등 교실에서 일어나는 전체 과정을 포함하는 것을 의미한다.

다음으로, 교육방법은 교육의 목적을 달성하기 위한 다양한 교수학습 상황을 고려하는 실질적인 방법을 의미한다(Chang, Koh, Park, 2013). 교육방법은 학습내용과 학습자, 학습 환경에 따라 다양하게 조직될 수 있는데, 본 연구에서 사용하는 교육방법은 코딩수학 교육과정의 목적을 효과적으로 달성하기 위해 학습 내용 및 학습 과제를 제시하는 방법에 관한 것으로 제한적인 의미로 사용하고 있다.

교육방법은 크게 전통적 방법과 구성적 방법으로 이루어져 있는데, 이 중 구성적 교육방법은 체험 중심의 교육 활동을 가능하게 하고 학습자의 자발성에 기초한 교육 활동을 제공함으로써 전통적 교육방법의 문제점을 보완하는 교육방법으로 많은 학자에 의해 제안되어 왔다(Hur, 2008). 또한, 구성적 교육방법의 토대가 되는 구성주의 이론은 급격한 사회 변화에 대응하기 위해 필요한 ‘자기주도적 학습’의 인식론적 토대를 제공해주고 있다(Yang, 2002). 이에 본 연구에서는 구성주의 이론을 바탕으로 교육과정 및 교육방법을 개발하였다. Tyler는 이러한 교육과정이 합리적으로 연구·개발되기 위해서는 목적 설정, 내용 선정 내용 조직, 평가라는 순으로 연구·개발이 이루어져야 함을 주장했다(as cited in Kim, 2002).

그렇다면 정보교과와 수학교과를 융합하는 코딩수학 교육과정의 교육목표는 무엇이어야 할까? Kim 외(2017)는 한국, 미국, 영국, 필란드 등에서 코딩교육을 통해 수학 실력의 향상이 보고된 점을 바탕으로 코딩교육과 수

학교교육이 밀접한 관련이 있다고 하였다. 글로벌 SW교육 포럼에서 BCS(영국컴퓨터협회) 교육총괄은 5세부터 코딩 교육을 실시한 결과 전체적인 아이들의 학습태도와 수학 과목 향상에 변화를 가져온 교육적 효과에 대해 소개하였다. 이러한 코딩교육을 통해 학습자가 길러야 할 핵심 역량은 고등 사고력과 관련되며, 이는 컴퓨팅 사고력(Computational Thinking)¹⁾과 연결되어 2015 개정 교육과정에서 코딩교육은 컴퓨팅 사고력을 함양하기 위한 내용과 방법을 주요 골자로 하고 있다(Cho & Cho, 2017). Wing(2006)은 컴퓨터 과학은 공식적인 기초가 수학에 있다는 점을 고려할 때 본질적으로 수학적 사고에 기초한다고 하였으며, Shin(2019)는 컴퓨팅 사고력과 수학적 사고력이 대체로 문제해결을 지향한다는 점과 문제해결 방식에서 유사한 사고 과정이라고 주장하였다. 이는 결국 컴퓨팅 사고력의 함양을 위해서는 수학적 사고력이 요구됨을 의미한다.

이에 관한 연구로 Sim, Park(2019)는 수학 학습과 평가를 통합한 컴퓨팅 사고력 기반의 수업은 초등학교의 메타인지와 수학 학습성취도의 향상에 긍정적인 영향을 미치는 것을 확인하였다. 또한 Ko(2018)에서 코딩교육은 시행착오를 겪으며 디버깅(debugging)하는 과정을 통해 올바른 알고리즘 수립 전략을 습득하고 문제해결력을 증진시킬 수 있고 창의력과 자기주도적 학습이 가능하다고 하였다. 앞의 선행연구에서 코딩교육이 컴퓨팅 사고력 및 수학적 사고력 신장에 유의미한 효과를 보였으므로 수학교과와 정보교과를 융합한 코딩수학 교육과정을 구체적으로 개발하고 적용할 때 융합 교육과정 연구와 실천에 컴퓨팅 사고력 역량의 증진이 핵심 요소가 될 수 있다.

그러나 단순히 수학 연습문제 또는 알고리즘을 코딩 명령으로 접근하며 수학과 코딩을 물리적으로 결합하는 교육과정은 큰 의미가 없다. 예를 들어, 구구단과 같은 수학 알고리즘을 코딩 명령으로 프로그래밍하는 단순한 문제풀이 기반의 교육과정은 학생의 동기 유발 및 자기주도적 학습을 유도하기 어렵고, 무엇보다도 추상화와 자동화로 대표되는 컴퓨팅 사고력 역량의 신장과 이에 대한

평가 및 적절한 교육적 피드백을 주기도 어렵다.

이에 본 연구팀은 컴퓨팅 사고력을 신장시키고 학생의 동기 유발 및 자기주도적 학습을 가능하게 하는 코딩수학 교육과정을 개발하고 이를 초등학교 및 중학교 학생들을 대상으로 적용하며 연구를 진행하였다. 개발한 코딩수학 교육과정은 온라인 교육환경에서도 학생이 동기를 유지하며 자기주도적으로 학습할 수 있도록 3차원 메인 크래프트 게임환경과 비슷한 터틀크래프트(Turtlecraft) 3차원 코딩환경을 도입하여 그 안에서 건축물을 제작한다. Papert(1972)의 구성주의(constructionism) 철학에서는 학생들이 조형물을 만드는 과정을 통한 학습(Learning by making)을 강조하는데, 본 연구에서는 3차원 코딩환경에서 건축물을 만드는 과정을 통해서 자연스럽게 명령에 대한 학습이 이루어지는 Learning by Making 구성주의 교육전략을 기반으로 한다. 이때 구성주의는 Piaget의 구성주의(constructivism)가 아닌 Papert의 구성주의를 의미한다. Piaget의 구성주의는 개별적 지식의 구성에 초점을 둔다면, Papert의 구성주의는 도구를 통해 지식을 구성하고 학생에게 의미 있는 결과물을 도출하도록 하며, 도구를 매개로 하여 무형의 지식을 유형의 조형물로 만들게 유도한다는 점에서 본 연구의 교육전략과 유사하다.

본 연구는 수학교과와 정보교과를 융합하는 코딩수학 교육과정과 최소 코딩게임 기반의 교육방법의 설계와 개발에 대해 논한다. 먼저, 코딩수학 융합 교육과정의 목표는 컴퓨팅 사고력의 신장임을 원칙으로 삼는다. 이를 위하여 수학교과와 정보교과의 공통점을 찾고, 수학교과의 수학적 표현에 기반하는 알고리즘과 정보교과의 프로그래밍 기반 알고리즘을 융합하여 최소 코딩게임 교수·학습 전략을 설계하였다. 최소 코딩게임 기반의 평가 문항을 통해 학생 스스로 자신의 수준을 파악할 수 있도록 하고, 컴퓨팅 사고력 역량의 수준 상승에 도움이 되도록 피드백을 제안한다. 본 연구를 통해 수학교과와 정보교과의 융합 교육 연구에 교육적 시사점을 제공하고, 향후 다양한 온라인 융합 교육과 중학교 자유학년제 수업 연구와 발전에 이바지하고자 한다.

1) Computational Thinking은 '컴퓨팅 사고력', '컴퓨터적 사고', '컴퓨팅적 사고' 등으로 다양하게 번역되고 있으나, 과학교육 분야에서는 Ministry of Education과 한국과학창의재단을 중심으로 '컴퓨팅 사고력'으로 번역하고 있으므로, 본 연구에서도 '컴퓨팅 사고력'을 사용하도록 하였다.

II. 이론적 배경

1. 컴퓨팅 사고력

1) 컴퓨팅 사고력의 의미 및 구성요소

Papert(1980)는 코딩환경이 학생들의 강력한 아이디어를 개발하는 도구를 제공하는 학습 환경이라고 주장하며, ‘가자’와 ‘돌자’의 몸동작에 기반한 거북 명령 프로그래밍 언어 LOGO를 개발하였다(LOGO 언어의 마우스 버전이 바로 스크래치 언어이다). MIT 대학의 수학자였던 Papert는 컴퓨팅 사고력(CT) 용어를 처음으로 도입하였으며, 컴퓨팅 사고력은 정보사회에 요구되는 새로운 능력으로 인간의 사고와 컴퓨터의 능력을 조합하는 의미에서 강조되고 있다고 하였다(as cited in Hwang, 2016). 이에 우리나라에서도 2018년부터 컴퓨팅 사고력을 기를 수 있도록 소프트웨어 교육 중심 교육을 강조하여 앞서 언급했듯이, 중학교에서 ‘정보교과’를 필수 과목으로 지정하였으며, 고등학교에서 ‘인공지능 수학’을 선택 과목으로 도입하였다.

Wing에 따르면 컴퓨팅 사고력은 “컴퓨터 과학자처럼 사고하는 것”(Wing, 2006; 2017; Wing & Stanzone, 2016)으로 “컴퓨터 과학의 근본적인 개념을 사용하여 문제를 해결하고 시스템을 설계하며 인간의 행동을 이해하는 것”(Wing, 2006; 2008)을 의미한다(as cited in Shin, 2019). 또한, Wing은 컴퓨팅 사고력을 “컴퓨터(인간 또는 기계)가 효과적으로 수행할 수 있는 방식으로 문제를 공식화(formulating)하고 해법을 표현하는 것(expressing)과 관련된 사고과정”(Wing, 2017, p. 8)으로 정의하였다. 이때 ‘효과적’이란 컴퓨터를 통해 ‘계산 가능한(computable)’을 의미하고, ‘표현하는 것’이란 해법을 다른 사람이나 기계와 의사소통하기 위해 언어적 표상을 만드는 것을 의미한다(as cited in Shin, 2019).

International Society for Technology in Education & Computer Science Teachers Association(ISTE & CSTA, 2011)는 컴퓨팅 사고력을 여러 분야에서 문제를 해결하는데 필요한 능력으로 보고, 컴퓨팅 사고력의 구성요소를 데이터 수집·분석·표현, 문제 분해, 추상화, 자동화, 알고리즘 및 절차, 검증, 일반화 등으로 제시하였다. 구성요소 각각의 정의를 정리하면 [Table 1]과 같다.

[Table 1] Definition of Components of Computational Thinking

Components	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
Automation	Having computers or machines do repetitive or tedious tasks
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) 컴퓨팅 사고력의 추상화와 자동화 역량

Wing(2006)은 컴퓨팅 사고력의 중요한 핵심 요소로 추상화(abstraction)와 자동화(automation)를 제시하였으며, 그중 추상화를 강조하였다. 또한, 컴퓨팅 사고력의 구성요소 중 Chang(2017)은 컴퓨팅 사고력과 수학교육의 공통요소로 알고리즘, 추상화, 분해, 패턴 인식을, Yu(2017)는 컴퓨팅 사고력과 수학적 사고의 공통요소로 알고리즘과 절차, 추상화, 문제 분해를 추출하였으며, Nam & Kim(2011)은 컴퓨팅 사고력과 수학적 사고의 공통요소로 알고리즘적 사고, 논리적 사고, 비판적 사고, 재귀적 사고를 추출하였다. 2015 개정 중학교 정보교과에서는 추상화를 핵심 개념으로 다루고 있고 제7차 수학과 교육과정에서는 수학적 지식의 특성 요인 중 하나로 추상화를 제시하였다. National Research Council(NRC, 2010)에 따르면, 컴퓨팅 사고력과 수학적 사고는 추상화를 통해 단순화된 모델에 관한 추론을 기반으로 한다는 점에서 공통점을 가지며, 고유의 언어(프로그래밍 언어와 수학기호)를 통해 대상을 표현한다는 점에서 비슷하다고 하였다. 이처럼 많은 선행연구에서 컴퓨팅 사고력과 수학적 사고의 공통요소 중 하나로 추상화 과정을 제시하고 있다.

ISTE & CSTA(2011)는 컴퓨팅 사고력의 중요한 특성인 추상화 과정을 주요 아이디어를 정의하기 위한 복잡성을 줄이는 과정이라고 말하였다. 구체적으로 중학교 정보 교과서(Jeong et al., 2020)에서는 [Fig. 1]과 같이 사용한 블록의 개수를 11개에서 6개로 줄이는 과정이 있는데, 반복되는 부분을 주요 아이디어로 정의하여 복잡성을 줄이는 과정이므로 바로 추상화 과정이라고 할 수 있다.



[Fig. 1] Abstraction and Automation in Middle School Information subject

Wing은 추상화는 인간이 할 수 있는 정신적인 도구라고 말하였으며, 추상화된 개념은 컴퓨터에 의해 자동화되어 반복적으로 수행될 수 있다고 하였다(as cited in Han, 2018). 또한, Wing(2008)은 컴퓨팅 사고력의 핵심은 추상화된 개념을 자동화의 도구를 이용하여 문제해결을 수행하는 것이라고 하였는데, 이처럼 컴퓨팅 사고력의 구성요소에서 추상화와 자동화는 따로 분리하여 생각하는 것이 아니라 함께 일어나는 과정임을 알 수 있다. ISTE & CSTA(2011)는 컴퓨팅 사고력의 자동화 과정을 컴퓨터나 기계가 반복적인 작업을 수행하는 과정이라고 정의하였다. 위의 [Fig. 1]에서 이동 방향으로 100만큼 움직이고 방향을 90°만큼 회전하는 명령을 주요 아이디어로 정의하는 추상화 과정을 거치면 이를 컴퓨터가 기계적으로 4번 반복하는 자동화 과정을 거친다는 점에서 추상화와 자동화는 연결되어 있다. 이처럼 중학교의 코딩교육에서는 추상화와 자동화가 컴퓨팅 사고력을 신장시키는 중요한 특성이라고 할 수 있다.

2. 터틀크래프트

1) 마인크래프트와 텍스트 기반 터틀크래프트

앞의 [Fig. 1]과 같은 추상화 및 자동화 예는 거북의 몸동작에 기반한 Papert의 LOGO 언어 거북 명령을 스크래치 언어로 코딩한 예이다. 교육용 프로그래밍 언어는 교육을 목적으로 학습자의 인지 발달 단계를 고려하여 개발된 프로그래밍 언어들에 의미를 둔다(Ju, 2021). Papert는 이러한 교육용 프로그래밍 언어가 갖추어야 할 요소로 ‘low floor’(시작하기가 쉽고), ‘high ceiling’(점점 복잡한 고수준의 프로젝트로도 접근할 수 있으며), ‘wide wall’(여러 관심사와 학습 스타일을 가진 사람들이 모두 참여할 수 있음)을 강조하였다(as cited in Park, 2018).

교육용 프로그래밍 언어는 블록 기반 프로그래밍 언어와 텍스트 기반 프로그래밍 언어로 나누어지는데, 특징을 간략히 정리하면 [Table 2]와 같다. 중학교 이상에서는 파이썬(Python)과 같은 텍스트 기반 언어가 점점 많은 나라에서 사용되고 있으며, 우리나라에서는 스크래치(Scratch)와 엔트리(Entry)와 같은 블록 기반 언어가 많이 사용되고 있다.

[Table 2] Block-based programming languages and text-based programming languages

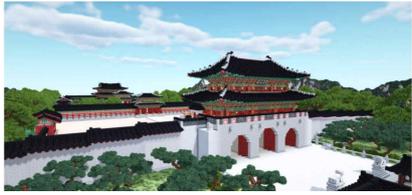
Type	Characteristic
Block-based programming languages	<ul style="list-style-type: none"> Implementation of building methods like Lego blocks instead of text commands Mainly used for the development of stories, games, animations, digital art, etc Example languages: Scratch, Entry, Algeomath, App Inventor, Kodu, Alice, etc
Text-based programming languages	<ul style="list-style-type: none"> Extract and apply only content for educational use in traditional programming languages Algorithms, procedural thinking, abstraction, reuse, debugging, modularization, etc Example languages: Python, JavaScript, PHP, BASIC, etc

블록 기반 언어는 유아 및 초·중등 학생을 위한 교육용으로 또한 텍스트 기반 언어는 중·고등학생 이상에서 문제해결용으로 많이 사용한다. 텍스트 기반 언어는 고급 언어로서 알고리즘, 절차적 사고, 추상화 등 수준 높은 컴퓨팅 사고력을 요구한다. 따라서 유아 및 초·중등 학생이

명령을 키보드로 입력해야 하는 텍스트 기반 언어를 어려워한다. 그러나 블록 코딩을 먼저 접하고 텍스트 코딩을 하는 경우 블록 이름을 영어 단어로 유추하며 명령을 찾을 수도 있어 텍스트 코딩에 도움을 줄 수 있다(Ann, 2018). 앞의 [Fig. 1]에서 사용된 스크래치 언어의 블록 코딩에서는 추상화 및 자동화를 블록을 연결하여 나타내는데, 블록의 형태가 규격화되어 있어 학생이 자유롭게 코딩하는 것에 제한이 있다. 반면에 텍스트 코딩은 추상화 및 자동화 과정을 규격화하지 않고 명령을 다양하게 사용할 수 있어 블록 코딩과 비교해 학생이 코딩하는 데 유연성이 더 높다는 장점이 있다.

터틀크래프트는 텍스트 기반 프로그래밍 언어로 자바스크립트(Javascript) 언어의 모든 명령이 여기에서 작동된다. 또한, [Table 3]과 같이 세계적으로 인기를 끌고 있는 마인크래프트(Minecraft) 게임과 유사한 큐브 기반의 구성 활동과 코딩을 지원하며, 구성된 산출물을 3D VR 게임²⁾환경과 3D 프린터 OBJ 파일 등과 연결할 수 있다. 이러한 특성에 기반하여 Kim(2017)은 터틀크래프트를 4차 산업혁명에 대비하는 ‘게임형 수학 학습 서비스’의 예로 소개하였다. 터틀크래프트는 다른 소프트웨어 교구와는 다르게 마인크래프트를 많이 접한 초등학생에게 매우 익숙한 환경일 수 있으며, 게임을 하듯이 접근할 수 있기에 학생에게 흥미를 불러일으키며 온라인 환경에서도 자기주도적 학습을 유도할 수 있다.

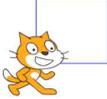
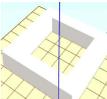
[Table 3] Minecraft game and TurtleCraft coding

Program	Screen example
Minecraft	
TurtleCraft	

²⁾ 3D 프린팅과 VR도 가능한 터틀크래프트 환경 : hanaone.com

2) 코딩수학 학습을 위한 터틀크래프트 코딩 환경 코딩 프로그램을 블록 코딩과 텍스트 코딩 중 해당하는 유형과 코드의 실행 결과가 2차원 평면과 3차원 공간 중 해당하는 유형을 정리하면 [Table 4]와 같다.

[Table 4] Comparison between Scratch, Algeomath, TurtleCraft and Python

Program	Characteristic	
	Coding command	Coding result
Scratch		 <ul style="list-style-type: none"> • Block coding • 2-dimensional plane
Algeomath		 <ul style="list-style-type: none"> • Block coding • 3-dimensional space
Turtle-craft	<pre>1 repeat 4{ 2 doit(5s) 3 doit(L) 4 }</pre>	 <ul style="list-style-type: none"> • Text coding • 3-dimensional space
Python	<pre>1 import turtle 2 3 t=turtle.Turtle() 4 t.shape("turtle") 5 6 for i in range(4): 7 t.forward(100) 8 t.left(90)</pre>	 <ul style="list-style-type: none"> • Text coding • 2-dimensional plane

터틀크래프트는 저학년의 스크래치 및 알지오매스(Algeomath)와 고학년의 자바스크립트 및 파이썬 언어를 중간에서 연결할 수 있는 교육적인 특징을 갖고 있다.

첫째, 터틀크래프트는 저학년이 사용하는 블록 기반의 스크래치 언어와 고학년이 사용하는 자바스크립트 및 파이썬과 같은 텍스트 기반의 고급언어를 연결할 수 있다. 고등학교 정보과 교육과정(Ministry of Education, 2015b)에서는 ‘[12정보04-01] 텍스트 기반 프로그래밍 언어의

개발 환경 및 특성을 이해한다'와 같이 텍스트 기반 프로그래밍 언어를 성취기준으로 삼고 있다. Ann(2018)은 텍스트 기반 프로그래밍의 강력함과 유연성을 대체할 수 없다며 고등학교 이상에서의 텍스트 기반 언어사용을 정당화하였다. 터틀크래프트는 프로그래밍 언어를 단순화시킨 스크래치의 블록 코딩을 고급 프로그래밍 언어인 자바스크립트 등의 텍스트 코딩으로 연결할 수 있다. 블록 코딩에 의한 명령과 대비하여 터틀크래프트에서 작성한 텍스트 기반의 명령은 카카오(Kakao) 등의 SNS를 통해 쉽게 코드를 전달할 수 있으며, 텍스트를 변형시키며 명령의 오류를 발견해내며 또한 명령을 여러 방향으로 개선하며 학습할 수 있도록 강력한 융통성을 제공한다.

둘째, 터틀크래프트에서는 자바스크립트 언어의 모든 명령이 작동되며, 스크래치와 파이썬 등에서 사용하는 거북 명령 기반의 LOGO 코딩도 유사하게 실행할 수 있다. [Table 4]을 보면 파이썬에서는 'for i range(4)'를 이용하여 명령을 4번 반복하고 있으며, 스크래치에서는 'repeat 4' 블록을 사용하여 해당 횟수만큼 repeat 안의 명령을 4번 반복하고 있고, 터틀크래프트에서도 스크래치와 마찬가지로 'repeat 4'를 이용하여 반복 명령을 표현할 수 있다. 터틀크래프트에서는 'repeat 4' 대신에 'doit((4){5sL})' 또는 'X='5sL' ; doit(4X)' 코드와 같은 수학적 코딩 표현으로 코드를 점점 짧게 표현할 수 있다. 'doit((4){5sL})'은 중괄호 { } 안에 있는 5sL을 4번 반복하며 쌓기나무를 만들라는 것이고 5s는 s를 5번 만들라는 수학적 표현이다. 'X='5sL' ; doit(4X)'은 문자 X에 명령 '5sL'을 저장하고, 4X 즉 4번 반복하라는 것이다. 터틀크래프트에서는 이처럼 추상적인 수학적 표현을 많이 사용할수록 점점 짧은 코드를 만들 수 있도록 설계되어 있다. 본 연구에서는 이러한 터틀크래프트의 특성을 기반으로 하는 최소 코딩게임 교수·학습 전략을 컴퓨팅 사고력 역량의 신장 및 평가에 핵심적으로 활용하였다.

또한, 터틀크래프트에서는 자바스크립트 언어의 모든 명령이 실행되며 특히 입문용으로 많이 쓰이는 스크래치와 파이썬에서 작동하는 거북코딩을 통해 교육적인 연결성을 유도하는 특징을 갖고 있으며, 또한 독자적인 여러 교육적인 특징들을 갖고 있다. 그 특징은 다음과 같다.

첫째, 일반적인 코딩 프로그램들은 실행 결과가 평면에서 이루어지지만, 터틀크래프트는 실행 결과가 3차원 공

간에서 이루어진다. [Table 4]에서 네 프로그램의 코드가 모두 정사각형을 만드는 명령인데, 스크래치와 파이썬은 2차원 평면에서 정사각형을 만든 반면에 터틀크래프트와 알지오매스는 3차원 공간에서 정사각형 모양의 입체 구조물을 만들고 있다. 이처럼 터틀크래프트는 코딩을 통해 3차원 입체물을 구성하며 공간적 추론 능력 및 컴퓨팅 사고력을 기를 수 있다는 장점이 있다(Chung, 2020). 또한 터틀크래프트는 명령을 실행한 결과가 3차원 좌표공간에서 입체물로 나타나므로 학생이 코딩 학습을 하면서 동시에 3차원 공간좌표라는 수학적 개념학습은 물론 만들어진 입체물을 3D 프린터와 VR 등으로 연결할 수 있기에 학생에게 동기와 흥미를 일으킬 수 있다.

둘째, 터틀크래프트는 텍스트 명령을 유연하게 고쳐가며 실행시킬 수 있으며 실행을 통해 피드백을 즉각적으로 얻을 수 있다. 스크래치와 알지오매스 같은 경우 블록을 사용하기 때문에 명령이 구체화되어 있어 명령을 유연하게 사용하는 데 어려움이 있고 블록의 개수로 인해 명령이 복잡해질수록 명령이 상당히 길어질 수 있다. 그러나 터틀크래프트는 텍스트 코딩이기 때문에 명령이 구체화되어 있지 않아 유연하게 사용하는 것이 가능하며 명령이 간결하고 짧다. 또한 스크래치는 명령이 실행되지 않는 경우 어느 부분에서 명령이 잘못되었는지 파악하기 어렵다. 터틀크래프트는 만들고자 하는 입체 구조물의 명령을 실행했을 때, 잘못 만들어진 부분이 시각적으로 표현되므로 학생에게 즉각 피드백이 유용하다.

이에 본 연구에서는 이러한 터틀크래프트의 특징을 활용하여 3차원 입체 구조물인 경회루를 만드는 코딩수학 교육과정을 설계하고, 또한 초·중등 학생들의 추상화 및 자동화 기반의 컴퓨팅 사고력 역량의 학습과 평가를 위해 최소 코딩게임 교수·학습 전략을 제안한다.

3. 최소 코딩게임 교수·학습 전략

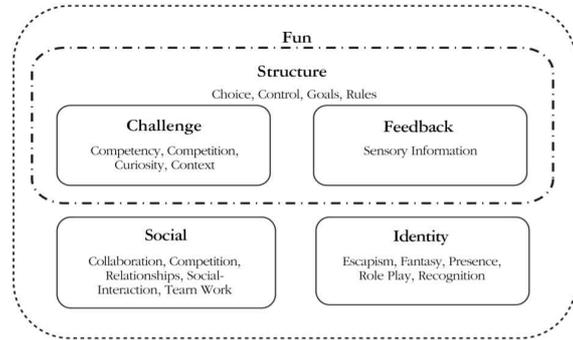
1) 최소 코딩게임 교수·학습 전략의 의미

코딩교육의 핵심 목표는 컴퓨팅 사고력을 신장시키는 것이며, 컴퓨팅 사고력의 주요한 특성은 추상화 및 자동화 과정이다. 최소 코딩게임 교수·학습 전략은 앞서 [Fig. 3]에서 블록의 개수를 줄이는 과정인 추상화 및 자동화 과정을 학생이 효과적으로 학습하기 위해 도입한 교수·학습 전략이다. 최소 코딩게임(Minimum coding game)은

여러 선행연구에서 효과적인 교수·학습 전략으로 사용되었으며, 최소 표현 게임, 최소 코드 게임 등의 용어로 불리고 있다. Jeong(2015)은 ‘최소 표현 게임’을 학생들에게 패턴을 발견하고 표현하는 것을 유도하는 활동으로 정의하였으며, 최소 표현 게임을 통해 학생이 대상을 추상화한다고 설명하였다. Jeong, Cho(2020)는 ‘최소 코드 게임’을 가장 적은 코드를 사용하여 대상을 구성하는 게임이라고 정의하였으며, 산술적 코딩에서 대수적 코딩으로 발전되는 교육 내용을 설계하고 대수적 코딩으로 나아가도록 하는 최소코드 지도전략을 제안하였다. 이와 관련하여 본 연구에서 언급하는 최소 코딩게임 교수·학습 전략은 다음과 같은 두 가지 특징을 통해 효과적인 지도전략으로 제안한다.

첫째, 명령을 가장 짧게 표현하라는 것이다. 예를 들어, 앞의 [Fig. 1]에서는 스크래치 프로그래밍 언어를 사용하여 정사각형을 만들 때, 블록의 개수를 11개에서 6개로 줄이고 있다. 최소 코딩게임을 활용하면 블록의 개수를 줄이는 것뿐만 아니라 ‘최소’로 블록의 개수를 사용하자는 조건을 붙여 문제를 제시한다. 마찬가지로 터틀크래프트에서도 특정 입체 구조물을 제시하고, 이 입체 구조물을 만드는 명령을 ‘가장 짧게’ 코딩하도록 문제를 제시한다. 학생들은 최대한 짧게 코딩하고 서로의 응답을 공유하며 스스로 효율적 코드를 발견하게 한다. 학생들은 공유된 코드의 효율성을 파악하며 자신의 코드를 반성하게 된다 (Jeong, 2020).

둘째, 문제를 게임 형태로 제시하는 것이다. 게임이라는 동기유발 전략을 활용하여 교육목표에 도달하고자 하는 학습인 게임 기반 학습은 학습 목표 달성을 위해 게임이라는 형태를 빌린다(Park, 2018). 즉, 게임 형태로 문제를 제시하는 것은 학생이 명령을 습득하는데, 게임을 활용함으로써 효과적으로 도움을 얻는 방법이라고 할 수 있다. 또, Korea Creative Content Agency(KCCA, 2013)은 [Fig. 2]와 같이 학습자를 위한 게임의 핵심 요인을 제시하였는데, 도전, 피드백이 있을 때 학습자를 위한 구조가 되어 학습자가 즐거움을 느낄 수 있다고 하였다. 따라서 최소 코딩게임을 통해 게임 형태로 학습을 진행하면서 학생들은 도전 의식을 느끼며 서로의 응답을 통해 피드백을 받을 수 있다.



[Fig. 2] Key Factors in Games for Learners

2) 최소 코딩게임 교수·학습 전략의 예시

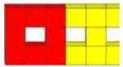
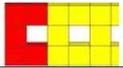
최소 코딩게임은 ‘가장 짧게’ 명령을 코딩하는 게임 형태의 문제이다. 학생에게 입체 구조물을 제시하고 학생이 그 구조물을 만드는 명령을 코딩하였을 때, 가장 짧게 코딩하였는지 판단하기 위해서는 판단 기준이 필요하다. Jeong, Cho(2020)는 최소코드 게임의 예시로 [Fig. 3]와 같은 입체 구조물을 제시하며, 판단 기준으로 코드의 개수를 두고 코드의 개수는 글자의 개수로 세도록 하였다. 이때 명령을 짧게 코딩하기 위해 사용한 ‘=’, ‘do’, ‘l’, ‘j’와 따옴표의 개수는 세지 않는다. 짧게 코딩하기 위해 사용한 명령에 관한 내용은 [Table 9]에서 다룬다.

1. 코드의 개수가 가장 짧으면 승리
2. 코드의 개수는 글자의 개수
3. ‘=’, ‘do’, ‘l’, ‘j’와 따옴표의 개수는 세지 않음

[Fig. 3] Example of minimum coding game

Jeong, Cho(2020)에 의하면 [Fig. 3]에서 제시한 입체 구조물에 대한 학생들의 응답을 대표적으로 [Table 5]로 나타낼 수 있다. 학생이 명령을 가장 짧게 코딩하기 위해서는 추상화가 필수적이다. 추상화는 주요 아이디어인 반복되는 부분을 정의하여 복잡성을 줄이는 과정이므로 [Table 5]에서 반복되는 부분을 어떻게 인지하느냐에 따라 코드의 개수가 달라지게 된다. Jeong, Cho(2020)이 제시한 최소코드 게임은 터틀말(Turtlmal)에서 적용되는 것으로 본 연구에서는 터틀말에 3차원 좌표공간을 추가한 터틀크래프트를 활용한 최소 코딩게임을 제안한다.

[Table 5] Examples of student answer

Type	Code	The number of codes
	X = ' s [2u 2s 2d] s ' do 5X	11
	X = ' 2s [2u 2t] ' do s [2u] 5X	12

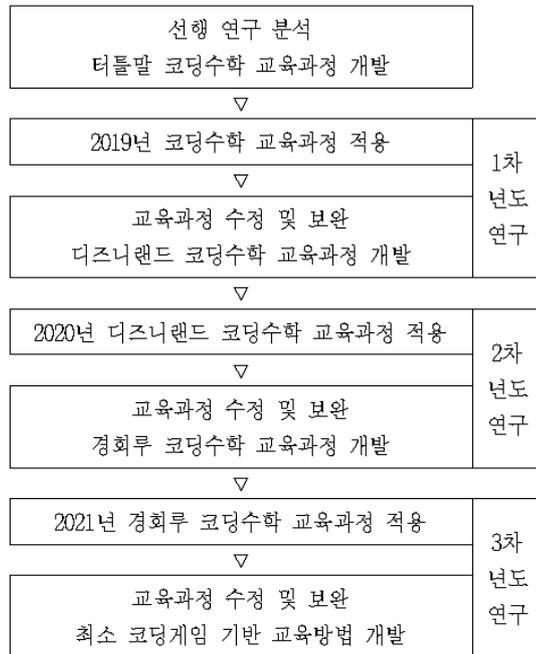
III. 교육과정 및 교육방법 연구 개발 방법

1. 연구 개발의 개관

코딩수학 교육과정 및 교육방법을 개발하는 본 연구의 과정은 크게 1차년도 연구, 2차년도 연구와 3차년도 연구로 나눌 수 있다. 먼저 1차년도 연구는 2차년도 연구를 위한 선행연구 분석 단계로서, 터틀말을 이용한 코딩수학 수업에 관한 선행연구를 바탕으로 ‘터틀말 코딩수학 교육과정’을 개발하였다. 이를 본 연구자가 2019년 중학생을 대상으로 적용한 결과 나타나는 한계점을 토대로 수학교과와 정보교과를 융합하여 3차원 코딩환경에서 3D 건축물을 제작하고 3D 프린트로 출력할 수 있는 ‘디즈니랜드 코딩수학 교육과정’을 개발하였다.

다음으로 2차년도 연구는 교육과정 개발 단계로서, 개발한 디즈니랜드 코딩수학 교육과정을 보완하여 ‘경희루 코딩수학 교육과정’으로 수정하였다. 구체적으로 개발한 디즈니랜드 코딩수학 교육과정을 본 연구자가 2020년 초등학교 및 중학생을 대상으로 적용하여 그 결과를 분석하였다. 분석 시 나타나는 보완점을 바탕으로 학생의 흥미를 유발하여 자기주도적 학습 및 수준별 수업을 가능하게 하는 ‘경희루 코딩수학 교육과정’을 개발하였다.

마지막으로 3차년도 연구는 교육방법 개발 단계로서, 개발한 경희루 코딩수학 교육과정을 적용하고 효과적인 교육과정 운영을 위한 ‘최소 코딩게임 교수·학습 전략’을 설계하였다. 본 연구자가 2021년 초등학교를 대상으로 적용하며 코로나19로 인한 온라인 교육환경에서도 학생의 동기유발 및 수준 상승을 유도할 수 있도록 경희루 코딩수학 교육과정과 연계되는 ‘최소 코딩게임 기반의 교육방법’을 개발하였다. 개발 절차를 정리하면 [Fig. 4]와 같다.



[Fig. 4] Research procedure

2. 연구 개발의 과정

1) 1차년도 연구

1차년도 연구에서는 터틀말을 이용한 코딩수학 수업 및 평가에 관한 선행 연구에서 제안한 지도방안과 본 연구자가 2018년 중학교 자유학년제에 터틀말을 활용하는 코딩수학 수업을 한 경험을 바탕으로 중학교 수학교과와 정보교과를 융합하는 터틀말 코딩수학 교육과정을 개발하였다. 이를 2차년도 연구를 위해 2019년 G중학교의 자유학년제 수업에 적용하고 수업 적용 결과 터틀말을 활용한 코딩수학 수업의 한계 및 보완점을 분석하였다. 또한, 터틀말을 활용하는 기존의 코딩수학 수업의 한계점을 극복하기 위해서 터틀말에 3차원 좌표공간을 추가한 터틀크래프트를 활용하는 ‘디즈니 코딩수학 교육과정’으로 수정·보완하였다.

2) 2차년도 연구

2차년도 연구에서는 1차년도 연구에서 개발한 터틀크래프트를 활용하는 디즈니랜드 코딩수학 교육과정을 2020년 S대와 H시가 함께 하는 창의인재육성 코딩멘토링 프

로그래를 통해 초등학교 6학년 학생 60명을 대상으로 적용하였고, G중학교의 자유학년제 수업에서 중학교 1학년 학생 40명을 대상으로 적용하였다. 자유학년제에서는 주제선택 활동 수업이 일반적으로 한 프로그램당 일주일에 2시간으로 진행되며 총 9차시, 17시간(1차시는 1시간)으로 운영되고 있다(Ministry of Education, 2015d). 따라서 본 연구에서는 자유학년제 주제선택 활동 수업 운영 시수에 맞춰 1차시당 2시간의 수업 분량으로 총 8차시, 16시간에 걸친 교육과정으로 구성하였다.

디즈니랜드 코딩수학 교육과정은 총 8차시로 매 차시마다 4~6개의 과제를 제시하였다. 1개의 과제에 각 1점씩을 부여하여 총 과제 개수는 40개로 최종 과제 점수는 40점이 된다. 과제는 객관식과 주관식이 혼합되며 있으며, 정답이면 1점, 오답이면 0.5점, 미제출이면 0점으로 부여하였다. 8차시 최종 과제는 ‘나만의 건축물 만들기’로 최종 과제 점수에는 포함되지 않는 과제이며, ‘나만의 건축물 만들기’에 대한 학생 답안을 코드, 명령어 코딩 과정, 명령의 효율성에 해당하는 세 가지 문항으로 구성하였다.

학생들의 과제 제출 답안은 코딩수학 홈페이지³⁾의 과제 게시판을 통해 수집하였다. 학생들이 각 과제에 대한 답안을 저장하면 과제별로 학생들의 답안이 저장되어 교사가 볼 수 있도록 설정하였다. 학생들의 답안을 바탕으로 초등학교 6학년 학생 60명의 최종 과제 점수의 분포를 양적으로 분석하였으며, 8차시 최종 과제 ‘나만의 건축물 만들기’에 대한 학생 답안을 질적으로 분석하고 보완점을 도출하고자 하였다. 2가지 자료인 양적 분석과 질적 분석을 통해 디즈니랜드 코딩수학 교육과정의 보완점을 파악하고 수정하여 최종적으로 ‘경희루 코딩수학 교육과정’을 개발하였다. 본 연구에서는 학생 답안의 분석 과정을 통해 개발한 경희루 코딩수학 교육과정을 제안하고 그 설계 원칙에 대해 논하고자 한다.

3) 3차년도 연구

3차년도 연구에서는 2차년도 연구에서 개발한 터틀크래프트를 활용하는 경희루 코딩수학 교육과정을 2021년 S대 사범대 부설 H협력센터 초등코딩 프로그램을 통해 초등학교 6학년 학생 40명을 대상으로 적용하였다. 초등코딩 프로그램은 1차시당 3시간씩 총 10차시로 운영하였

다. 3시간에는 과제에 대한 피드백 제공 시간도 포함되는 것으로 실제 수업 시간은 1차시당 1시간 30분씩으로 총 10차시, 15시간으로 운영하였다.

경희루 코딩수학 교육과정은 [Fig. 13]과 같이 총 6단원으로 구성하였으며, 매 단원마다 10개의 과제를 제시하였다. 10개의 과제에 각 1점씩을 부여하여 총 과제 개수는 60개로 최종 과제 점수는 60점이 된다. 과제는 객관식으로만 이루어져 있으며, 정답이면 1점, 오답이면 0점으로 부여하였다. 학생들의 과제 제출 답안은 경희루 코딩수학 홈페이지⁴⁾의 단원별 게시판을 통해 수집하였다. 학생들이 각 단원에 대한 답안을 저장하면 답안이 엑셀로 저장되어 교사가 볼 수 있도록 설정하였다. 초등학교 6학년 학생 40명의 최종 과제 점수의 분포를 양적으로 분석하고 그 결과를 통해 경희루 코딩수학 교육과정을 효과적으로 운영하기 위한 ‘최소 코딩게임 기반 교수방법’을 설계하였다. 본 연구에서는 설계한 최소 코딩게임 기반 교육방법을 제안하고 그 설계 원칙에 대해 논하고자 한다.

IV. 연구 결과 및 논의

1. 1차년도 연구

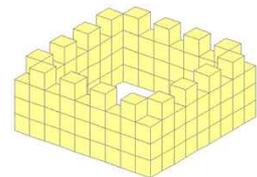
1) 터틀말 코딩수학 교육과정의 적용 및 분석

Cho, Cho(2017)는 터틀말을 이용하여 쌓기나무 학습의 평가 및 지도방안을 제안하였다. 터틀말은 터틀크래프트와 유사하게 공간에서 입체 구조물을 만드는 프로그래밍 언어이다. [Fig. 5]는 터틀말을 이용하여 쌓기나무 성을 표현하고 있는 지도방안의 예시이다.

1. 오른쪽의 쌓기나무 성을 표현할 수 있는 실행식을 써보세요.

2. 실행식을 줄일 수 있는 방법에는 무엇이 있을까요? 방법을 써보고 실행식을 압축해보세요.

3. 쌓기나무의 한 모서리의 길이가 1이라고 했을 때, 가로 길이가 20인 쌓기나무 성의 실행식을 써보세요.



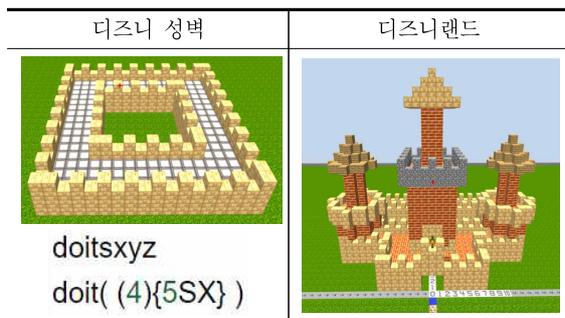
[Fig. 5] Example of coding mathematics in TurtleMal

3) 코딩수학 홈페이지 주소 : snucode.org

4) 경희루 코딩수학 홈페이지 주소 : snucodemath.pythonanywhere.com

터틀말을 활용한 코딩수학 교육과정은 다음과 같은 한계가 있었다. 첫째, 앞서 다룬 터틀크래프트는 3차원 좌표 공간이지만 터틀말은 공간이기만 할 뿐 3차원 좌표가 없기 때문에 수학교과에 있는 좌표, 순서쌍과 같은 수학적 개념을 학습하기에 제한이 있었다. 둘째, 교육과정에서 매차시 학습 목표가 새로운 명령에 대한 습득으로 최종적으로 교육과정이 끝난 후 학생들에게 남는 결과물이 없어 학생들의 흥미를 유발하기 어려웠고 공간에서 만든 입체물을 실제 공간에서 시각적으로 확인하지 못하고 컴퓨터 화면에서 평면으로 남는 한계가 있었다.

이전 선행연구에 터틀크래프트로 중학교 자유학기제 수업에 적용한 결과 자기주도적 학습 및 학습자의 자발적 참여를 유도할 수 있었다는 연구(Kim, Seo & Cho, 2018)와 터틀크래프트를 중학교 자유학기제 활동에 적용한 후 흥미를 유발하고 수학 및 코딩의 정의적 영역을 개선하는데 긍정적 효과가 있었다는 연구(Choi, 2020)를 근거로 터틀말의 제한점을 보완한 터틀크래프트를 활용하는 것으로 수정하였다. 즉, 터틀말을 활용한 코딩수학 교육과정의 제한점을 보완하여 수학교과와 정보교과를 융합할 수 있도록 터틀크래프트를 활용한 ‘디즈니 코딩수학 교육과정’을 설계하였다. 디즈니 코딩수학 교육과정에서는 3차원 좌표공간에서 건축물을 표현하는 활동으로 수학교과의 좌표, 순서쌍 등 수학적 개념을 학습할 수 있을 뿐만 아니라 제작한 건축물을 3D 프린트로 출력하여 실제 공간에서 확인할 수 있도록 하여 학생들의 흥미를 유발할 수 있다. 다음은 [Fig. 6]은 [Fig. 5]의 터틀말에서만 성벽을 터틀크래프트에서 성벽 건축물로 3차원 좌표공간에 표현한 것이다.



[Fig. 6] Disneyland in TurtleCraft

2) 디즈니랜드 코딩수학 교육과정 설계

앞서 서론에서 언급했듯이, 수학과 정보의 관련성, 창의·융합형 인재라는 미래 인재상에 근거하여 수학교과와 정보교과를 융합한 교육과정의 필요성이 나타나고 있다. 중학교 수학에서는 학습 요소에 좌표, 순서쌍, 좌표평면 등이 포함되어 있는데, 터틀크래프트 3차원 코딩환경은 스크래치 등 코딩환경과 달리 이러한 수학적 개념을 기반으로 코딩 학습을 할 수 있다. 특히, 2차원 좌표평면이 아니라 3차원 좌표공간에서 명령을 다루어 좌표공간까지도 학습할 수 있으며, 명령 실행 결과가 즉각적으로 시각화되어 나타나므로 공간좌표에 대한 학생의 이해를 높일 수 있다. 이번 장에서는 터틀크래프트 3차원 코딩환경을 기반으로 한 디즈니랜드 코딩수학 교육과정에서 수학교과와 정보교과를 융합하여 설계하는 과정을 논의하겠다.

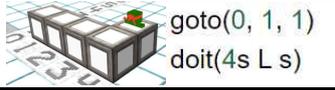
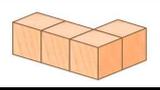
(1) 초등학교 수학과 수학적 코딩표현

2015 개정 교육과정에서 초등학교 수학과 목표 중 하나로 ‘수학적으로 추론하고 의사소통하며, 창의·융합적 사고와 정보 처리 능력을 바탕으로 생활 주변 현상을 수학적으로 이해하고 문제를 합리적이고 창의적으로 해결한다.’를 제시하고 있다(Ministry of Education, 2015c). 이처럼 초등학교에서도 창의·융합적 사고와 정보 처리 능력을 강조함을 알 수 있으며, 코딩수학 교육과정의 필요성이 제시된다. 또한, Ministry of Education(2015)는 초등학교 수학교과에서 한 도형을 다루는 경험으로부터 비롯되는 공간 감각은 수학적 소양을 기르는 데 도움이 된다고 하였으며, 이에 ‘[2수02-02] 쌓기나무를 이용하여 여러 가지 입체도형의 모양을 만들고, 그 모양에 대해 위치나 방향을 이용하여 말할 수 있다’를 성취기준으로 삼고 있다.

초등학교 수학 내용인 쌓기나무에서는 정육면체를 하나씩 쌓으며 입체 구조물을 만든다. 마찬가지로 터틀크래프트에서도 [Table 6]과 같이 정육면체를 doit 명령으로 하나씩 만들며 입체 구조물을 만든다. [Table 6]에서 goto(0, 1, 1)은 거북이가 좌표평면에서 순서쌍 (0, 1, 1) 위치로 이동하는 것을 말하며, doit(4s L s)에서 s는 step의 약자로 앞으로 1칸 가서 쌓기나무를 하나 만드는 것, L은 Left의 약자로 거북이가 왼쪽으로 90도 회전하며 방향을 바꾸는 것을 말한다. 앞으로 4칸 가면서 쌓기나무를 4개 쌓는 것은 중학교 수학에서 나오는 계수를 이용하여

4s로 나타낼 수 있다. 이처럼 터틀크래프트에서는 정보교과의 코딩에 초등학교 수학교과의 도형 영역의 쌓기나무와 중학교 수학교과의 대수 영역의 계수, 함수 영역의 순서쌍을 융합하고 있다.

[Table 6] Comparison of TurtleCraft coding and elementary school Mathematics subject

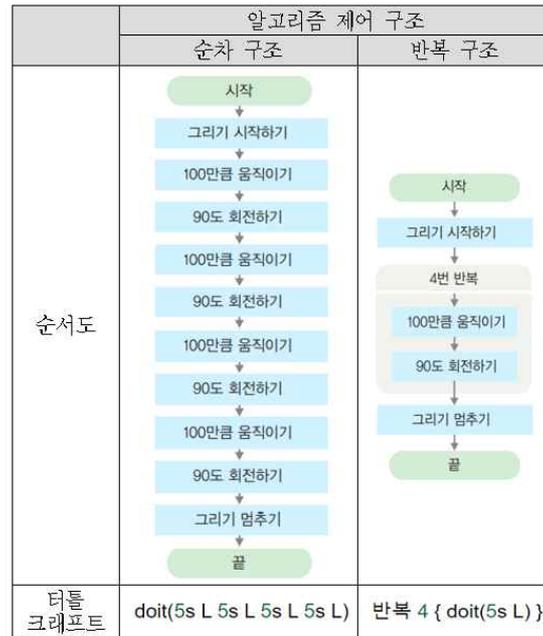
TurtleCraft coding	Elementary school math
	

1차년도 연구의 터틀말 교육과정을 중학교 학생들에게 적용하였을 때, 중학교 학생들은 초등학교에서 이미 쌓기나무를 접하였기 때문에 쌓기나무를 코딩으로 쉽게 쌓을 수 있었다. 디즈니랜드 코딩수학 교육과정은 터틀크래프트 3차원 코딩환경에서 초등학교 쌓기나무를 수식과 문자를 포함하는 goto와 doit 명령으로 쌓을 수 있도록 설계하였다. 또한, [Table 3]에서 볼 수 있듯이, 코딩환경을 마인크래프트 게임과 유사하게 함으로써 초등학교 6학년 학생도 수학적 표현을 게임을 위한 놀이 언어로 받아들일 수 있도록 설계하였다. goto(0, 1, 1)와 같은 좌표와 doit(4s)의 '4s'와 같은 표현은 학생이 중학교 수학에서 만나게 될 수학적 표현이고 s를 4번 반복해서 실행하는 관점에서 중학교 정보교과의 알고리즘 반복 구조의 수학적 표현이 된다. 이렇게 수학적 표현으로 3차원 입체물을 만들며 수학과 코딩을 융합하는 것을 디즈니랜드 코딩수학 교육과정의 출발점으로 잡았다. 다음에서는 디즈니랜드 코딩수학 교육과정에서 사용하는 터틀크래프트 코딩환경이 정보교과의 알고리즘 제어구조와 수학적 표현을 융합하는 과정을 논의하겠다.

(2) 정보교과 알고리즘에 대한 수학적 코딩표현

정보교과에는 [Fig. 7]과 같이 알고리즘 및 제어 구조가 포함되어 있다. 알고리즘이란 문제를 해결하는 방법이나 절차를 순서에 따라 논리적으로 설명하거나 표현한 것을 말하며, 이러한 알고리즘을 순서도로 나타낼 수 있다. 알고리즘 제어 구조에는 순차·반복·선택 구조가 있는데, 순차 구조는 시작부터 끝까지 제시된 순서에 따라 차례대로 처리하는 경우이며, 반복 구조는 주어진 조건을

만족할 때까지 특정 부분을 반복하여 처리하는 경우이다 (Jeong et al., 2020). 특히, 반복 구조는 순차 구조에서 반복되는 부분을 간략히 줄일 때 사용한다.



[Fig. 7] Sequential and Iterative Structure of Algorithm Control Structure

[Fig. 7]의 순서도 표현은 100만큼 앞으로 가고 90도 회전하는 행동을 4번 반복하여 2차원 평면에 정사각형을 그리는 활동을 알고리즘의 순차 구조와 반복 구조로 각각 표현한 것이다. 터틀크래프트 코딩환경에서 앞으로 5칸 가는 '5s'와 왼쪽으로 90도 회전하는 'L'을 4번 사용하여 3차원 공간에서 쌓기나무로 정사각형을 표현하는 것은 순차 구조에 해당하며 여기서 4번 반복되는 부분을 '반복 4'를 이용하여 순차 구조를 줄인 것이 반복 구조에 해당한다. 이처럼 터틀크래프트 코딩환경에서는 이러한 순차 구조와 반복 구조를 '반복 4 { doit(5s L) }'와 같이 문자와 수식이 들어간 수학적 표현으로 코딩할 수 있다. '5s'는 s를 5번 쓴 것으로 수학의 계수를 의미하며, '4 { }'는 수학에서의 괄호를 사용하여 괄호 안의 명령어를 4번 반복하는 것을 의미한다.

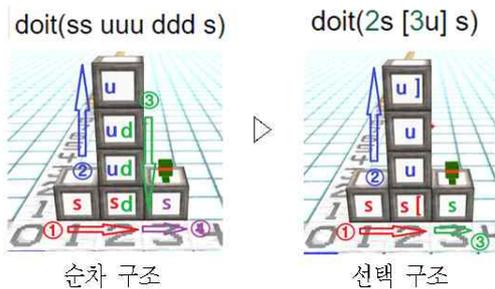
알고리즘 제어 구조의 선택 구조는 주어진 조건에 따

라 특정 부분을 선택적으로 처리하는 경우로 정보교과서에서는 [Fig. 8]처럼 세노와 달세노에 비유하여 설명하기도 한다. 터틀크래프트에서는 세노에 해당하는 꺾쇠 명령 '['와 달세노에 해당하는 꺾쇠 명령 ']'가 있다. '['는 거북이의 현재 위치를 기억하는 명령이고 ']'는 거북이가 기억하는 위치로 돌아가는 명령으로 꺾쇠 명령을 이용하여 선택 구조를 표현할 수 있다.

선택 구조			
정보 교과서		터틀크래프트	
에서 세노(%)는 반드시 뒤쪽에 달세노(D.S)가 나옵니다. D.S를 만나면 %표시가 있는 곳으로 돌아가 거기서부터 연주하라는 뜻입니다.	%	[현재 위치를 기억하는 명령
	D.S]	기억한 위치로 돌아가는 명령

[Fig. 8] Selective Structure of Algorithm Control Structure

[Fig. 9]는 꺾쇠 명령을 이용하여 순차 구조를 선택 구조로 짧게 표현하는 과정이며, 이는 컴퓨팅 사고력의 추상화에 해당한다. 이러한 추상화 및 자동화 과정은 3차년도 연구의 [Table 9]에서 자세히 다루겠다.



[Fig. 9] Selective Structure of TurtleCraft

이처럼 터틀크래프트 코딩환경을 통해 수학적 표현과 수학적 개념이 포함된 텍스트 코딩으로 알고리즘 제어 구조의 순차·반복·선택 구조를 나타낼 수 있으며 컴퓨팅 사고력의 추상화 과정이 포함되어 있다. 다음 장에서는 이러한 1차년도 연구의 결과로 본 연구의 핵심 교육과정인 수학교과와 정보교과를 융합한 경희루 코딩수학 교육 과정에 대한 연구를 논하겠다.

2. 2차년도 연구

1) 디즈니랜드 코딩수학 교육과정 적용 및 분석

1차년도 연구를 통해 수정·보완한 디즈니랜드 코딩수학 교육과정을 터틀크래프트 3차원 좌표공간에서 디즈니랜드를 코딩하는 학습활동이다. 개발한 디즈니랜드 코딩수학 교육과정을 실제 수업에 적용하고, 적용 결과를 토대로 3차년도 연구를 위한 새로운 경희루 코딩수학 교육과정으로 수정·보완하는 과정을 기술한다.

[Table 7]은 코딩수학 교육과정이 모두 종료된 후 총 과제 점수 40점에 대한 학생들의 점수 분포표이다. 31~40점에 해당하는 학생의 비율은 68.3%로 대부분을 차지하는데, 11~20점과 1~10점과 0점에 해당하는 학생의 비율은 각각 8.3%, 15%, 1.7%로 20점에 못 미치는 학생의 비율이 총 25%, 총 학생의 4분의 1에 해당한다. 이는 코딩수학 교육과정의 학습 내용을 제대로 이해하지 못한 학생이 상당히 많다는 것을 의미한다.

[Table 7] Student Score Distribution Table

Score	Students	Percent(%)
31 ~ 40	41	68.3
21 ~ 30	4	6.7
11 ~ 20	5	8.3
1 ~ 10	9	15
0	1	1.7
Total	60	100

이러한 결과가 나온 이유는 코딩교육이 위계성을 가진다는 점인데, 실제로 2015 개정 정보 교과 교육과정의 내용 체계는 네 영역으로 구성되어 있으며 위계성을 가지고 있고, 중학교와 고등학교에도 학교급간 연계성을 가지고 있다(Choi, Lee, Kim & Lee, 2015). 서론에서 언급했듯이, 코딩교육은 수학교육과 관련이 있고(Kim et al., 2017), 수학교과의 중요한 특징은 위계적 성격을 가지므로(Shin, 2017) 코딩교육도 수학교육과 마찬가지로 위계성이 있다. 이렇게 코딩교육이 위계성을 가지면 수학 학습처럼 초반 학습 내용을 제대로 습득하지 못하면 다음 내용도 이해하기 어렵게 된다. 즉, 코딩교육의 위계성으로 인해 이전 명령을 이해하지 못하면 다음 수준의 명령도 이해하기 어렵다. 본 연구에서는 이러한 한계점을 보완하기 위해 낮은 수준의 명령으로도 최종적으로 건축물을 제작할 수 있도록 코딩수학 교육과정을 개선하였다.

한편, 디즈니랜드 코딩수학 8차시 수업에서는 최종 과제인 ‘나만의 건축물 만들기’를 제시하면서 건축물의 명령을 만드는 과정과 명령의 효율성을 파악하기 위하여 [Fig. 10]과 같은 3가지 문항으로 구성하여 제시하였다.

-
- [1번]** 건축물을 만드는 코딩명령어
- 실행했을 때 건축물이 나오도록 아래 노란색 명령어창에 코딩명령어 전체를 적으세요.
-
- [2번]** 건축물의 주제 및 스토리텔링
- 아래 노란색 명령어창에 건축물의 주제 또는 제목을 적으세요.
 - 스토리텔링 : 건축물의 코딩명령어를 만드는 과정, 건축물의 특징이나 만든 이유를 설명하세요.
-
- [3번]** 내 명령어의 장점 및 효율성
- 1번에서 내가 만든 코딩명령어의 장점 및 효율성 등을 적으세요.
-

[Fig. 10] The final task, ‘Creating your own building’

[학생 A]

- cube 명령어 : 원하는 곳에 블록을 만들 수 있지만 하나하나 입력해야 하기 때문에 여러 가지 블록을 쌓는 것이 비효율적이다.
- doit 명령어 : 여러 가지 블록을 쌓을 수 있지만, 실행 버튼을 눌러야 한다.
- 치환 명령어 : 복잡한 doit 명령어를 쉽고 간편하게 쓸 수 있어 효율적이다.
- 집합 명령어 : 여러 가지 도형들을 간편하게 만들 수 있어 매우 효율적이다.
- 반복 명령어 : 반복되는 명령어들을 여러 번 쓰지 않도록 해 주어 명령어를 줄여줘 효율적이다.

[학생 B]

일단, 한옥과 풍선을 표현한 집합 명령어는 일반 doit 명령어보다 더욱 빠르고 명령어를 짧게 할 수 있다. 그리고 창문, 나무, 기둥을 만들 때 사용한 반복 명령어는 일일이 하나하나 같은 규칙이 있는 명령어를 짧고 간단하게 나타낼 수 있어서 더욱 편하게 조형물을 만들 수 있었다.

[Fig. 11] Examples of Student answer about ‘Creating your own building’

학생들이 만든 건축물을 분석한 결과 건축물의 특정한 구성요소에 특정한 명령을 사용한다는 점을 발견하였다. 위의 [Fig. 11]은 ‘나만의 건축물 만들기’의 3가지 문항 중 [3번] 문항에 대한 학생 2명의 답안이다. 학생 A는 doit 명령어가 여러 가지 모양의 쌓기나무를 쌓을 수 있다고 하였고, 치환·집합·반복 명령이 명령을 짧게 줄여주어 효율적이라고 하였다. 특히, 집합 명령이 여러 가지 ‘도형’을

간단히 쓸 수 있다고 표현하였는데, 이는 학생 B도 공통으로 등장하는 답안이다. 또, 학생 A는 반복 명령이 반복되는 부분을 여러 번 쓰지 않아 효율적이라고 하였는데, 학생 B는 그 구조물이 나무, 기둥 등 반복되는 구조물일 때 더욱 편하다고 답변하였다. 이처럼 학생들은 구조물의 특성에 따라 더 효율적인 명령어가 있다고 답변하였다.

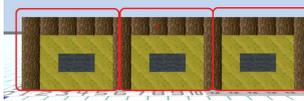
<doit 및 반복 명령어>



<집합 명령어>

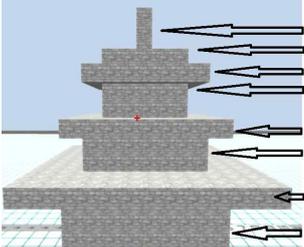


<doit 및 반복 명령어>



```
반복 3 {doit(s 3u 5s 3d ;2)
doit(4t 2u 3s d ;11)
doit(2t ;42)
doit(d 2s ;11) }
```

<집합 명령어>



```
집합(정(0,0,z,0) && 12<z && z<16 :9)
집합(정(0,0,z,2) && z==12 :9)
집합(정(0,0,z,3) && z==11 :9)
집합(정(0,0,z,2) && 8<z && z<11 :9)
집합(정(0,0,z,4) && z==8 :9)
집합(정(0,0,z,3) && 5<z && z<8 :9)
집합(정(0,0,z,6) && z==5 :9)
집합(정(0,0,z,4) && 1<z && z<5 :9)
```

[Fig. 12] doit command and set command

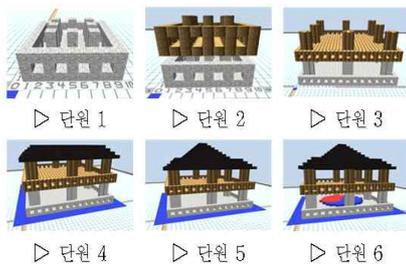
위의 [Fig. 12]는 ‘나만의 건축물 만들기’ 문항 중 [1번] 문항에 대한 답안을 실제로 실행한 결과이다. [Fig. 12]의 왼쪽에 표시한 창문, 기둥 등 특정 부분이 반복될 때는 doit 명령어로 특정 구조물을 만들고 그 구조물을 반복 명령어로 반복하면 명령이 짧아져 효율적인데, 이는 학생 B의 의견과 일치한다. 또한, [Fig. 12]의 오른쪽에 표시한 답과 같이 ‘도형’을 층별로 쌓아 올릴 때는 doit보다 집합 명령어를 사용하면 훨씬 효율적인데, 이는 학생 A의 의견과 일치한다. 이에 본 연구에서는 기둥, 난간, 바닥, 지붕 등 구성요소마다 효율적인 명령어가 있다는 점을 보완하여 최종적으로 경회루를 제작하는 ‘경회루 코딩수학 교육과정’을 개발하였다. 다음에서는 디즈니랜드 코딩수학 과정을 수정·보완하여 개발한 경회루 코딩수학 교육과정을 제안하고 그 설계 원칙에 대해 논하고자 한다.

2) 경회루 코딩수학 교육과정 설계 및 개발

(1) 자기주도 학습을 위한 설계 원칙

융합 인재 교육(STEAM 교육)은 과학(Science), 기술(Technology), 공학(Engineering), 인문·예술(Arts), 수학(Mathematics)의 머리글자를 합하여 만든 용어로 우리나라는 2011년부터 사용되었다. 경회루 코딩수학 교육과정은 정보교과의 코딩에 수학교과를 융합하고 학생이 명령어로 건축물을 디자인한다는 점에서 융합 인재 교육을 위한 것이라 할 수 있다. 이러한 융합 인재 교육은 초·중등학교 수준에서 국가 경쟁력 향상을 위한 창조적이고 융합적인 인재의 양성을 목표로 한다(Ministry of Education, Science and Technology [MEST], 2010). Kim, Hong, Cho, Im(2013)에 의하면 융합 인재 교육 이후 전반적으로 학생의 자기주도적 학습 능력이 향상되었다. 학교급별로 차이가 나타났는데, 초등학생은 주로 1년 차부터 중학생은 2년 차부터 긍정적인 변화가 나타난 반면 고등학생은 변화가 나타나지 않았다.

따라서 본 연구에서는 초등학생과 중학생의 긍정적인 변화가 나타났다는 점에 근거하여 초등학생과 자유학년제 수업에서 중학생들이 자기주도적 학습 능력을 향상할 수 있도록 수학교과와 정보교과를 융합하되, 다음과 같이 단계를 따라 경회루를 구성하는 Learning by Making 구성주의를 기반으로 설계하였다. 구체적인 코딩수학 교육과정의 설계 원칙은 다음과 같다.



[Fig. 13] Gyeonghoeru Coding Mathematics Curriculum

첫째, 학생의 수준에 맞는 명령어를 쓰도록 개별 맞춤형으로 교육과정을 구성하며, 컴퓨팅 사고력을 신장할 수 있도록 수준이 다른 명령어를 통해 학생의 수준 상승을 유도해야 한다. [Fig. 13]과 같이 터블크래프트 3차원 코

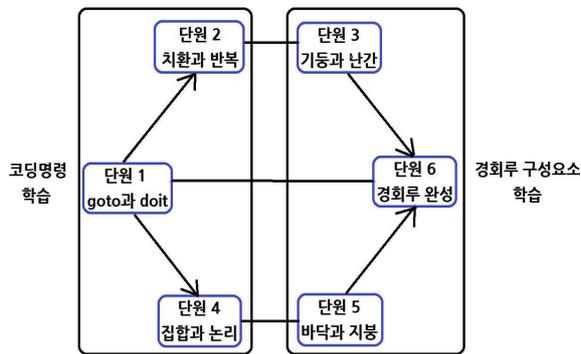
딩환경에서 경회루의 구성요소들 하나씩 만드는 과정을 따라 교육과정을 6단원으로 설계하였다. 디즈니랜드 코딩수학 교육과정은 명령의 위계성에 따른 누적 학습으로 이전 수업의 명령어를 이해하지 못하면 다음 수업의 명령어를 이해하기 어려운 구성이었다. 그러나 경회루 코딩수학 교육과정은 다양한 수준의 학생이 모두 경회루를 제작할 수 있도록 수준이 낮은 1단원부터 수준이 가장 높은 6단원까지 수준이 다른 명령어들로 구성되어 있다. 수준이 낮은 학생들도 1단원에서 배우는 명령어를 이용하여 2단원부터 6단원까지의 경회루의 구성요소들을 모두 만들 수 있으며, 학생들은 6단원에 걸쳐 수준이 더 높은 효율적인 명령어를 이용하여 경회루의 구성요소를 하나씩 만들어나가며 최종적으로는 경회루를 완성한다. 여기서 학생이 수준이 높은 효율적인 명령어를 쓴다는 것은 추상화 및 자동화 과정을 거치는 것을 의미하므로 경회루 코딩수학 교육과정을 통해 학생의 컴퓨팅 사고력의 신장을 유도함을 알 수 있다. 또한, 수준이 낮은 1단원에서 배우는 명령어로도 경회루의 구성요소를 만들 수 있다는 것은 학생 개인의 수준에 맞는 맞춤형 학습을 제공함을 의미한다. 이러한 맞춤형 학습은 학습시간과 자기주도적 학습능력을 매개로하여 수학에 대한 태도와 수학학업성취도에 간접 영향을 미치고 있었으며, 수학에 대한 태도에는 직접 효과도 있는 것으로 나타난다(Park, Lim, Kim, Lee & Kim, 2020).



[Fig. 14] 3D printing and snow ball

교육과정 마지막 차시에는 최종적으로 완성한 경회루를 [Fig. 14]과 같이 3D 프린팅한 후 프린팅한 경회루를 스노우볼로 만들 수 있다. 교육과정이 이루어지기 전에 학생에게 스노우볼에 대한 안내를 하여 3D 프린터와 연결을 함으로써 학생의 학습 동기를 유발할 수 있다. 학습에 대한 참여 동기와 적극성의 증진은 자기주도적 학습할 수 있도록 하고 이는 학업성취도의 향상까지 연결될 수 있다(Yang, Cho, 2009).

둘째, 경회루의 구성요소를 하나씩 만드는 문제를 통해 문제 안에 코딩 명령이 숨어 있도록 잠재적 교육과정 (Hidden Curriculum)으로 제시하는 전략을 쓴다. 많은 코딩 교육과정이 프로그래밍 언어를 학습하는 것으로 코딩 명령 중심으로 수업이 구성되어 있다. 중학교 정보과 성취기준에서 ‘프로그래밍’ 단원의 학습 요소를 살펴보면 프로그래밍 환경, 프로그래밍 과정, 입력, 처리, 출력, 변수, 산술연산, 비교연산, 논리연산, 순차 구조, 선택 구조, 반복 구조 등으로 명령 위주로 학습 요소가 구성되어 있음을 확인할 수 있다(Ministry of Education, 2015b). 명령 중심의 수업을 진행하게 되면 학생이 코딩 수업에 흥미를 느끼지 못하고 학습 동기가 일어나지 않을 수 있다. [Fig. 15]처럼 경회루 코딩수학 교육과정은 명령문 중심이 아니라 단원 1, 2, 4에서 경회루의 구성요소에 따라 명령을 도입하고, 단원 3, 5, 6에서 직접 경회루 구성요소를 하나씩 만드는 Learning by Making 활동과 문제 속에 명령이 숨어 있도록 잠재적 교육과정으로 구성하였다.



[Fig. 15] Structure of Gyeonghoeru Coding Mathematics Curriculum

(2) 컴퓨팅 사고력 수준 상승을 유도하는 교육과정

코딩교육은 컴퓨팅 사고력을 신장시키는 것이 목표이며, 컴퓨팅 사고력의 핵심 구성요소인 추상화와 자동화가 일어나는 명령을 학생이 사용하도록 유도하는 것이 바로 학생의 컴퓨팅 사고력을 신장시키는 것이다. 명령이 추상화 및 자동화 과정을 거치게 되면 명령이 더 짧아지고 더 많은 수학적 개념이 요구되므로 짧은 명령이 수준이 더 높다고 할 수 있다. 결국 수준이 높은 명령이 컴퓨팅

사고력을 신장시키는 명령이며 경회루 코딩수학 교육과정을 통해 학생이 수준이 낮은 명령에서 수준이 높은 명령을 사용하도록 학생에게서 수준상승이 일어나야 한다.

경회루 코딩수학 교육과정에서 경회루의 구성요소를 쌓아 나가며 수준 상승을 유도할 수 있는 이유는 구성요소 별로 각기 수준이 다른 명령을 도입할 수 있기 때문이다. [Fig. 15]에서 볼 수 있듯이 수준이 가장 낮은 1단원에서는 경회루의 기둥, 난간, 바닥, 지붕을 goto, doit 명령만 이용해서 모두 제작할 수 있도록 하였다. 1단원보다 수준이 높은 2, 3단원에서는 경회루의 기둥과 난간을 제작할 때 치환과 반복 명령을 이용하여 짧게 줄일 수 있도록 하였으며, 2, 3단원보다 수준이 높은 4, 5단원에서는 경회루의 바닥과 지붕을 제작할 때 집합 명령을 이용하여 더 짧게 줄일 수 있도록 하였다. 최종적으로 6단원에서는 학생은 직접 자신의 수준에 맞는 명령을 이용하여 자기주도적으로 경회루를 제작한다. 이처럼 경회루 코딩수학 교육과정에서는 학생이 자신의 수준에 맞는 명령을 스스로 조작하면서 자기주도적 학습을 가능하게 할 뿐만 아니라 경회루의 구성요소들을 레고 블럭을 조립하듯 하나씩 만들어감으로써 학습이 진행될 때마다 성취감을 느낄 수 있으며, 구성요소에 따라 수준이 다른 명령을 도입하여 수준 상승을 유도할 수 있다.

본 연구에서 주목하는 컴퓨팅 사고력 역량의 핵심 요소는 추상화와 자동화 역량이다. 추상화된 명령이 사용될수록 명령의 길이가 짧아지고, 또한 알고리즘을 자동화하면 더욱 짧은 명령으로 코드를 만들 수 있다. 터틀크래프트 3차원 코딩환경에서는 수학적 표현과 수준이 다른 여러 명령을 통해 구성물을 만들 수 있으며, 복잡도가 높은 수학적 표현과 코딩 명령을 통해 명령의 길이를 줄일 수 있다. 이에 다음 장에서는 학생의 컴퓨팅 사고력의 신장과 수준 상승을 끌어낼 수 있는 효과적인 교수·학습 전략으로 최소 코딩게임 교수·학습 전략을 논의하겠다.

3. 3차년도 연구

1) 경회루 코딩수학 교육과정 적용 및 분석

2차년도 연구를 통해 디즈니랜드 코딩수학 교육과정을 수정·보완한 경회루 코딩수학 교육과정을 터틀크래프트 3차원 좌표공간에서 경회루를 코딩하는 학습활동이다. 2차년도 연구에서 수준이 낮은 학생이 초반에 명령어를 이

해하지 못하면 그 이후 명령어도 학습하기 어렵다는 점을 보완하여 낮은 수준에서도 최종 건축물을 제작할 수 있게 수준별 수업으로 개선하였다. 또한, 건축물의 구성요소를 제작할 때 구성요소의 특징에 따라 특정 명령을 사용하는 것이 효율적이라는 점을 반영하여 경회루의 구성요소에 따라 수준이 다른 명령을 도입할 수 있도록 하였다. 본 장에서는 개발한 경회루 코딩수학 교육과정을 실제 수업에 적용하고, 적용한 결과를 토대로 코딩수학 교육과정을 2차로 수정·보완하는 과정을 기술하였다.

[Table 8] Student Score Distribution Table (2)

Score	Students	Percent(%)
50 ~ 60	4	10
41 ~ 50	17	42.5
31 ~ 40	9	22.5
21 ~ 30	3	7.5
11 ~ 20	5	12.5
1 ~ 10	2	5
0	0	0
Total	40	100

[Table 8]은 경회루 코딩수학 교육과정이 모두 종료된 후 총 과제 점수 60점에 대한 학생들의 점수 분포표이다. 41~50점에 해당하는 학생의 비율은 42.5%, 31~40점에 해당하는 학생의 비율은 22.5%로 31~50점에 해당하는 학생의 비율이 대부분을 차지한다. 21~30점과 11~20점과 1~10점에 해당하는 학생의 비율은 각각 7.5%, 12.5%, 5%로 디즈니랜드 코딩수학 교육과정에서 11~20점과 1~10점과 0점에 해당하는 하위권 비율인 8.3%, 15%, 1.7%보다 개선된 것을 확인할 수 있다. 그러나 51~60점과 41~50점과 31~40점에 해당하는 학생의 비율은 10%, 42.5%, 22.5%로 디즈니랜드 코딩수학 교육과정에서 31~40점과 21~30점에 해당하는 상위권 비율인 68.3%, 6.7%보다 최상권 비율이 줄어든 것을 확인할 수 있다. 이는 수준이 낮은 명령어라도 학생들이 최종 건축물인 경회루를 제작할 수 있으므로 학생들에게 더 높은 수준으로 상승해야 할 동기가 적어 발생한 현상이라고 해석할 수 있다. 실제로, 학생들은 코딩수업 시간에 수준이 낮은 명령어로 경회루의 구성요소를 전부 만들 수 있다는 점에서 더 수준이 높은 명령을 받아들이는 것에 크게 흥미를 느끼지 못했다.

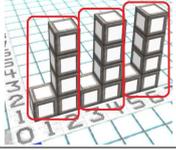
따라서 본 연구에서는 이러한 한계점을 보완하기 위해 수준이 낮은 명령에서 수준이 높은 명령으로 수준 상승을 유도하면서 학생들의 동기를 유발할 수 있는 교수·학습 전략인 최소 코딩게임 교수·학습 전략을 개발하였다.

2) 최소 코딩게임 기반 교육방법 설계

(1) 최소 코딩게임 교수·학습 전략의 설계 원칙

본 연구에서는 컴퓨팅 사고력의 신장, 즉 수준 상승이 일어나기 위한 효과적인 교수·학습 전략으로 최소 코딩게임 교수·학습 전략을 제안한다. 최소 코딩게임 교수·학습 전략을 통해 학생의 컴퓨팅 사고력의 신장을 유도하는 과정은 다음과 같다.

[Table 9] Abstraction and Automation of TurtleCraft

	Coding command	Coding result
[A]	goto(0, 1, 1) doit(2s 3u 3d) doit(2s 3u 3d) doit(2s 3u 3d)	
▽ Abstraction		
[B]	goto(0, 1, 1) X='2s 3u 3d' doit(3X)	
▽ Automation		
[C]	goto(0, 1, 1) X='2s 3u 3d' repeat 4 { doit(3X L) }	

첫째, 하나의 입체 구조물을 만들 때 추상화 및 자동화를 거쳐 컴퓨팅 사고력이 향상되도록 수준이 다른 명령을 사용하며, 명령의 길이를 가장 짧게 사용하도록 게임 형태로 문제를 제시한다. 터틀크래프트 3차원 코딩환경은 수준이 다른 명령을 사용하는 과정에서 추상화와 자동화가 나타난다. [Table 9]의 [A]에서 doit 명령에서 반복되는 부분이 주요 아이디어이므로 반복되는 부분을 치환문자 X를 사용하여 치환한 후 X가 의미하는 기둥이 3개 반복되는 것을 3X로 짧게 줄여 [B]로 복잡성을 줄이는 과정은 추상화다. 또한 [Table 9]의 [B]에서 기둥 3개와

왼쪽으로 회전하는 것을 '3X L'로 추상화한 명령을 4번 반복하여 [C]로 컴퓨터가 반복적인 작업을 수행하는 과정은 자동화다. 이처럼 [A], [B], [C]는 추상화 및 자동화를 거쳐 점차 수준이 상승하는 명령이라고 할 수 있다.

[Table 10] Minimal Coding Game

	Student answer	The number of codes
[A]	doit(2s 3u 3d) doit(2s 3u 3d) doit(2s 3u 3d)	18
[B]	X='2s [3u]' doit(3X)	7

최소 코딩게임 형태로 제시한다는 것은 학생에게 경회루의 기동을 만드는 명령을 코딩하는 활동뿐만 아니라 학생에게 글자를 가장 최소로 사용하여 '가장 짧게' 쓰도록 조건을 추가하는 것을 말한다. 코딩을 최소로 '가장 짧게' 쓰는 판단 기준으로 코드의 개수를 두고 코드의 개수는 글자의 개수로 세도록 하였다. 이때 학생이 사용한 명령의 수준에 따라 학생이 글자를 사용하는 개수가 다르게 된다. 최소 코딩게임 교수·학습 전략의 목적은 학생의 수준 상승을 끌어내는 것이고 수준 상승은 추상화 및 자동화를 거쳐 명령을 짧게 쓸수록 일어나게 된다. 따라서 명령을 짧게 코딩하기 위해 사용하는 글자는 개수로 세지 않는다. 즉, 치환하기 위해 사용하는 '='와 따옴표, 거북이의 이동 경로를 줄이기 위해 사용하는 꺾쇠 명령인 '['와 ']', 거북이가 움직이도록 명령하는 'doit'와 괄호의 개수는 세지 않는다.

[Table 10]은 [Table 9]의 [A]인 경회루의 기동을 최소 코딩게임 형태로 제시했을 때, 학생의 답안 예시이다. [Table 10]의 [A]에서 해당 입체 구조물을 치환문자를 사용하지 않고 순차 구조로 명령을 쓰기 위해 사용되는 글자의 개수는 'doit'과 괄호의 개수를 제외하고 18개이다. 또한 [Table 10]의 [B]에서 반복되는 부분을 발견하고 치환문자와 꺾쇠 명령을 사용하여 반복 구조와 선택 구조로 명령을 완성할 때의 횟수는 치환하기 위해 사용하는 '='와 따옴표, 거북이의 이동 경로를 줄이기 위해 사용하는 '['와 ']', [A]와 마찬가지로 'doit'과 괄호의 개수를 제외하고 7개가 된다. [B]로 명령을 코딩한 학생이 [A]보다 더 짧게 썼으므로 코딩게임에서 승리하게 되며, 학생들은

명령을 가장 짧게 줄이기 위해 추상화 및 자동화를 거치며 컴퓨팅 사고력의 신장, 즉 수준 상승이 일어나게 된다.

둘째, 최소 코딩게임 문제를 통해 학생의 학습 동기와 도전 의식 및 성취감을 불러일으키며, 서로의 응답을 확인하며 가장 효율적인 명령에 대한 피드백을 제공받는다. 단순히 추상화 및 자동화 과정만 설명하는 것이 아니라 학생에게 글자의 개수를 '가장 적게' 사용한다는 조건을 게임 형태로 제시함으로써 도전 의식을 불러일으킨다. 또한 학생은 반복되는 부분을 발견하면서 명령을 짧게 줄일 수 있다는 문제의식을 느끼고 명령을 짧게 줄이면서 성취감을 느끼게 된다. 최소 코딩게임 교수·학습 전략은 '가장 짧게'라는 조건이 있기 때문에 짧게 줄인 명령이 실제 가장 짧은지 생각해보며 학습 동기를 지속해서 유발할 수 있다. 최소 코딩게임 문제를 푸는 활동이 끝난 후에는 학생들이 나타난 명령의 개수를 정확히 셀 수 있으므로 직관적으로 개수를 파악하며 서로의 응답을 확인하며 효율적인 명령을 파악할 수 있다. 이처럼 최소 코딩게임 교수·학습 전략은 학생이 자신의 수준에 머물러 있게 하는 것이 아니라 수준이 다른 명령을 다루며 학생이 명령을 짧게 쓰도록 동기부여를 하여 학생의 수준 상승을 끌어내고 컴퓨팅 사고력을 신장시킬 수 있다.

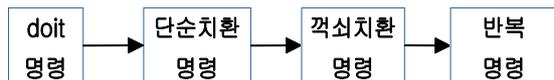
(2) 컴퓨팅 사고력을 진단하는 평가 및 피드백

소프트웨어 교육이 강조됨에 따라 컴퓨팅 사고력 역량의 수준과 신장을 위한 연구가 활발한데, Lee, Jang(2018)에 따르면 컴퓨팅 사고력 평가를 위한 기존 도구를 분석해본 결과 컴퓨팅 사고력 기반의 문제해결 과정을 측정하기 위한 능력 요소의 타당성이 부족하였으며, 문항 특성의 변별도는 양호하였으나 난이도는 쉬운 문항의 비율이 높아 학생의 역량을 정확하게 측정하기에는 부족한 것으로 나타났다. 또한, 기존에 개발된 컴퓨팅 사고력 검사 도구는 컴퓨팅 사고력 역량에 대한 평가라기보다는 내용과 기능에 대한 평가에 가까운 것이 많았다(Lee & Jang, 2018). 따라서 컴퓨팅 사고력의 진단 및 수준 상승을 유도할 수 있는 평가 문항을 개발할 필요성이 있다.

이에 본 연구에서는 기존 검사 도구의 한계를 보완하여 컴퓨팅 사고력 중에서도 추상화 및 자동화에 초점을 두고 컴퓨팅 사고력을 진단하고 또한 학생들의 수준 상승을 유도할 수 있는 평가 문항을 제안한다. 구체적으로

명령을 짧게 줄이는 추상화 과정을 거쳐 추상화된 명령을 반복하는 자동화 과정을 [Table 9]에서 언급한 치환 명령과 [Fig. 8]에서 언급한 격쇠 명령 적용한 평가 문항 예시를 논한다. 또한 본 연구에서는 교육과정과 평가 문항을 연계하도록 교육과정에서 학생의 컴퓨팅 사고력을 신장시키기 위해 활용한 최소 코딩게임 교수·학습 전략을 평가 문항에도 적용하였다. 교육과정에서 경회루의 구성요소를 만드는 명령의 개수를 세는 과정을 통해 최소 코딩게임 교수·학습 전략을 적용하였다면, 평가 문항에서는 경회루의 구성요소를 만드는 명령을 제시하고 그 명령을 짧게 바꾼 알맞은 명령을 고르는 5지 선다형의 객관식 문항으로 출제하였다. 주어진 구성요소를 만드는 명령을 문항에서 제시하고 제시된 명령을 올바르게 짧게 줄여 쓴 명령을 고른다는 점에서 최소 코딩게임 교수·학습 전략의 객관식 유형이라고 할 수 있다.

앞서 말했듯이, 수학교과와 같이 명령어에도 위계성이 존재하므로 명령의 수준에 따른 학습 계통도를 작성하여 학생이 명령의 수준에 따라 순차적으로 학습하고 평가할 수 있도록 제작하였다. [Fig. 16]은 경회루의 구성요소인 기둥과 터틀크래프트 안에서 사용되는 명령을 조합하여 학생이 학습하는 명령에 대한 위계를 나타낸 학습 계통도이다. 격쇠치환 명령을 학습하기 전에 단순치환 명령에 대한 학습이 선행되어야 하고 단순치환 명령을 학습하기 전에는 doit 명령에 대한 학습이 선행되어야 한다. 이러한 명령 간에 위계가 있으므로 이를 평가 문항을 제작할 때 이를 반영하여 제작하였다.



[Fig. 16] Learning tree of TurtleCraft

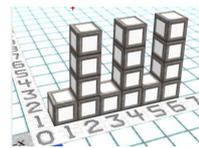
학습 계통도에 따른 학생의 학습 과정을 살펴보면 먼저 학생은 doit 명령을 이용하여 기둥을 만들고 [Table 9]와 같이 doit 명령을 단순치환 명령을 통해 짧게 줄이는 과정을 학습하며 치환한 명령을 더 짧게 줄이기 위해서 격쇠를 도입하여 격쇠치환 명령을 학습한다. 여기서 말하는 단순치환 명령이란, doit 명령에 반복되는 부분을 치환 문자 X를 사용하여 치환하는 것을 의미하며, 격쇠치환이란, 단순치환 명령어에 거북이의 이동 경로를 줄이기 위

한 격쇠 명령 '['와 ']'를 사용한 것을 의미한다. 이 과정은 컴퓨팅 사고력의 추상화 과정을 의미하며 점차 짧게 줄이는 과정을 통해 학생의 수준 상승이 일어나게 된다. 다음으로 격쇠치환 명령으로 명령을 짧게 줄이고 나면 하나의 기둥을 여러 번 반복하여 사용하는 반복 명령을 학습하고 반복 명령을 이용하여 기둥을 여러 개 만들 수 있게 된다. 이 과정은 격쇠치환 명령으로 추상화 과정을 거친 명령을 여러 번 반복하는 것으로 컴퓨팅 사고력의 자동화 과정에 해당한다.

[Fig. 17]은 컴퓨팅 사고력의 수준을 진단하기 위해 추상화 및 자동화를 거치는 단순치환 명령과 격쇠치환 명령의 학습 여부를 진단하기 위한 문항 예시이다. [Fig. 17]에 제시한 문항에 대한 학생의 응답을 통해 학생의 단순치환 명령과 격쇠치환 명령의 숙달 정도를 진단할 수 있다. 추상화와 자동화는 컴퓨팅 사고력의 중요한 특성이기 때문에 해당 명령의 숙달은 컴퓨팅 사고력의 함양과 관련이 있다고 할 수 있다. 본 연구에서 제안하는 평가 문항은 이러한 명령들 사이의 위계성을 고려하여 제작되었으며, 문항 제작의 원리는 다음과 같다.

Question 52

다음은 경회루의 기둥과 이를 만드는 명령어입니다. 제시된 명령어를 짧게 줄여 쓴 명령어를 2개 고르세요.



doit(2s 3u 3d)
doit(2s 3u 3d)
doit(2s 3u 3d)

- ① X='2s 4u 3d' ② X='2s 4u 4d' ③ X='2s 3u 3d'
- doit(3X) doit(3X) doit(3X)
- ④ X='2s [3d]' ⑤ X='2s [3u]'
- doit(3X) doit(3X)

[Fig. 17] Examples of evaluation question

첫째, 각 문항의 선택지에는 두 가지 이상의 서로 다른 수준의 명령을 제시한다. 출제 의도는 학생이 사용하는 명령에 따른 수준을 파악하고 수준에 따른 피드백을 주기 위함이다. 같은 수준의 명령을 제시하는 것보다 두 가

지 이상의 서로 다른 수준의 명령을 제시할 때 출제 의도를 효과적으로 적용할 수 있다. [Fig. 17]은 이에 대한 평가 문항 예시로 터틀크래프트에서 ‘치환 명령’과 ‘꺾쇠 치환 명령’으로 경회루의 기둥을 만드는 명령을 코딩할 수 있는지 숙달 여부를 확인하는 문제다. ①, ②, ③은 ‘단순치환 명령’을 묻는 보기이며, ④와 ⑤는 ‘꺾쇠치환 명령’을 묻는 보기로 앞의 [Table 9]처럼 단순치환 명령보다 꺾쇠치환 명령이 더 짧으므로 수준이 높은 명령이다.

[Fig. 17]의 정답은 ③과 ⑤인데, ⑤가 ③보다 높은 수준이라고 할 수 있다. 그 이유는 명령이 더 짧아 명령을 코딩하는 데 사용되는 글자의 개수가 더 적기 때문이다. 앞에서 언급한 최소 코딩게임 교수·학습 전략에서 짧게 쓰는 것에 대한 판단 기준인 글자의 개수를 세는 방법에 따르면 ③의 명령에 해당하는 글자의 개수는 9개이고 ⑤의 명령에 해당하는 글자의 개수는 7개이기 때문에 ⑤의 명령이 ③의 명령보다 수준 높은 명령이라고 할 수 있다.

둘째, 평가 문항의 유형은 5지 선다형으로 제작하며 수준을 파악하고 수준 상승을 유도하기 위해 낮은 수준의 명령부터 높은 수준의 명령까지 차례대로 나열한다. 이는 학습 목표에 따른 명령의 학습 수준을 단계별로 나누어 각 단계에 대한 학생의 학습 목표 숙달 정도가 어느 위치에 있는지 진단하기 위함이다. 터틀크래프트 안에서는 같은 조형물이라도 다양한 명령을 혼합하여 만들 수 있다는 점에서 단답형이나 서술형의 형태로 문항을 출제할 경우, 학생의 답안이 지나치게 다양해져 학생 수준의 진단이 어려워질 수 있다. [Fig. 17]에서 학생은 명령을 나열된 순서대로 분석하면서 치환 명령을 사용해서 doit 명령을 짧게 줄이는 방법에 대해 생각해보게 되고 그 이후에 치환 명령을 다시 꺾쇠 명령을 사용해서 짧게 줄이는 방법에 대해 생각해보게 된다. 이러한 사고 과정은 학생이 점점 명령을 짧게 쓸 수 있다는 것을 깨닫게 되며 이는 컴퓨팅 사고력의 추상화 및 자동화에 해당한다.

셋째, 평가 문항은 5개의 보기 중 정답을 1개가 아닌 2개로 구성하였다. 정답이 1개가 아닌 복수 정답으로 구성하면 학생은 첫 번째 정답을 찾았더라도 다음의 정답을 찾기 위해서 보기를 더 분석해야 하므로 학생의 수준 상승을 유도할 수 있다. 또한 정답이 되는 두 가지 명령은 서로 다른 수준이기 때문에 학생이 선택한 정답의 명령에 기반하여 학생의 수준을 진단할 수 있다.

[Fig. 17]에서 ③, ⑤를 둘 다 맞은 경우, 치환과 꺾쇠 명령을 모두 이해하고 있으며, ③, ⑤를 둘 다 틀린 경우 치환과 꺾쇠 명령을 모두 이해하지 못한 것으로 볼 수 있다. ③은 맞고 ⑤를 틀린 경우 ③보다 ⑤가 명령의 수준이 더 높으므로 ③의 치환 명령은 이해하나 ⑤의 꺾쇠 명령은 이해하지 못한 것이다. 이처럼 경회루의 구성요소의 명령에 대한 문항을 푸는 과정을 통해 학생의 수준을 진단하는 것은 수준에 따른 피드백을 가능하게 한다.

‘학교 수학을 위한 원리와 기준’에서 평가는 학습을 지원하는 데 초점이 맞추어져 있어야 하며 교사와 학생 모두에게 유용한 정보를 제공해야 함을 평가 원리로써 말하고 있으며, 우리나라 2015 개정 수학과 교육과정도 이와 맥락을 같이한다(Shin, 2017). 또한 학습에 대한 평가는 필연적이며, 평가를 통해서 학생은 자신의 수준을 파악할 수 있고 다음 수준의 학습으로 진입할 수 있는지에 대한 여부를 판단할 수 있다. 따라서 교사가 평가 문항을 통해 학생의 수준을 진단했다면 학생에게 앞으로 나아갈 학습 방향에 대한 지도와 학습 목표에 대한 명확한 제시가 담긴 피드백을 제공해야 한다. Ministry of Education, Korea Foundation for the Advancement of Science & Creativity(2018)에 의하면 교사는 평가를 통해 얻은 결과에 기초하여 학생에게 즉각적이고 실제적인 피드백을 가능한 한 많이 제공함으로써 평가가 학생의 학습 향상에 이바지하도록 해야 한다. 학생은 평가 후에 본인의 학습 상황을 교사의 피드백에 의해 파악하게 되므로 교사는 평가 후에 학생에게 우수한 점과 부족한 점, 나아갈 학습 방향을 알려주는 적절한 피드백을 제공할 필요가 있다.

[Table 11] Contents of Feedback

Contents of Feedback	Detail
<ul style="list-style-type: none"> Where is your goal? Learning goal 	<ul style="list-style-type: none"> Clear presentation of learning goal One way to present the best answers and the best practices
<ul style="list-style-type: none"> What is the student's level?? Evaluation results 	<ul style="list-style-type: none"> Objective information about evaluation results Compare your learning goals with student's level
<ul style="list-style-type: none"> How can be learning goal reached? Improvements 	<ul style="list-style-type: none"> Information on what to do more Efficient Learning Strategies and Pathways

Kim, Park, Park, Jung(2014)에 의하면 앞의 [Table 11]과 같이 학생에게 제공되는 피드백에는 현재 학습 목표가 무엇인지, 학습 후에 학생이 참여한 평가를 통해서 확인되는 학생의 학습 수준은 어느 정도인지, 목표한 학습 수준에 도달하지 못했다면 어떤 방법으로 학습하여 목표한 학습 수준에 도달할 수 있는지에 관한 내용이 포함되어야 한다. 이에 본 연구에서는 코딩수학 교육과정을 단원별로 진행하면서 학생의 현재 수준을 진단하는 평가 문항에 대한 응답을 통해 수준 상승을 유도할 수 있는 교사의 피드백을 제안한다. 따라서 앞서 최소 코딩게임 교수·학습 전략과 관련하여 논한 평가 문항의 예시에 대해 학생의 수준 상승을 유도하는 피드백 예시를 논하겠다. 최소 코딩게임 교수·학습 전략이 평가 문항과 연계되기 위해서는 코딩수학 교육과정에서 학생이 단원별로 명령을 학습하고 각 단원에서 학습한 내용과 관련된 평가 문항을 풀어야 한다. 따라서 평가 문항도 교육과정과 마찬가지로 1단원부터 6단원까지 나누어 구성하였다. 학생이 평가 문항을 풀기 위해서는 각 단원에 설정된 구성요소에 따른 명령을 학습해야 하므로 코딩수학 교육과정의

단원별 학습 목표의 숙달이 필수적이며, 이에 따라 학습 목표의 숙달 정도를 통해 학생의 수준을 파악하며 수준에 따른 알맞은 피드백을 제공하게 된다.

앞서 [Fig. 16]에 따르면 단순치환 명령을 학습이 완료되어야 다음 학습 목표인 꺾쇠치환 명령을 학습할 수 있다. 단순치환 명령과 꺾쇠치환 명령을 모두 틀린 학생의 경우, 치환 명령에 대한 학습이 필요하다는 것과 함께 치환 명령의 이전 학습에서 어떤 명령을 학습하지 못했는지 확인할 필요가 있다. 따라서 단순치환 명령의 전 단계인 doit 명령에 대한 학습 여부도 함께 확인해야 한다. 단순치환 명령은 맞고 꺾쇠치환 명령은 틀린 학생의 경우, doit 명령과 치환 명령에 대한 학습이 이루어졌으나 꺾쇠치환 명령은 학습이 필요하다는 것을 알 수 있다. [Fig. 17]에서 모두 정답을 선택한 학생의 경우, 치환 명령과 꺾쇠치환 명령을 모두 알고 있으므로 다음 학습 목표인 반복 명령에 대한 학습을 진행하게 된다.

한편, 학생의 성장을 위한 피드백은 ‘결과’보다는 ‘과정’에 초점을 두어야 한다. 따라서 학습 목표의 숙달이 미흡한 학생이 어떤 부분을 학습하여 앞으로의 학습에 성장

[나의 현재 학습 정도]

명칭	성취수준
계수	계수를 이용하여 명령을 줄일 수 있다.
doit 명령	거북이를 앞과 위, 아래로 움직이며 쌓기나무를 쌓을 수 있다.
단순치환 명령	기동 한 줄에 해당하는 명령이 반복되는 것을 이해하고, 이를 치환하여 명령을 줄일 수 있다.

경회루의 구성요소 중 하나인 기동을 doit 명령을 이용해서 만들 수 있고, 만들어진 명령을 계수를 이용해서 줄일 수 있습니다. 줄인 명령어를 보고 기동 하나를 만드는 명령어가 반복된다는 것을 알고 치환 명령어를 이용하여 줄일 수 있습니다.

[학습해야 하는 명령어]

명칭	성취수준
꺾쇠치환 명령	치환된 명령 안에서 u와 d가 연속으로 사용되면 거북이의 이동경로가 중복된다는 것을 이해하고, 이를 꺾쇠를 이용해서 명령을 줄일 수 있다.

제시된 기동 명령을 보고 꺾쇠치환 명령을 이용해서 바꾸는 과정에 대한 학습이 필요합니다. 꺾쇠를 열고 닫았을 때 거북이가 어느 위치로 돌아가게 되는지 다시 공부해봅시다. 아래의 링크로 가서 꺾쇠치환 명령을 복습하세요.
<https://www.youtube.com/watch?v=oHFAPj5d0N8>

[Fig. 18] Examples of Feedback

을 이룰 수 있는지 개선점을 나열한 피드백보다는 가장 중요하고 학습의 성장에 빠르게 영향을 줄 수 있는 학습의 방향을 우선 제시하는 것이 효율적이다. 학습이 부진한 학생뿐만 아니라 우수한 학생에게도 피드백이 필요하다는 것이다. 학습 목표를 충분히 숙달한 학생에게는 현재 학습 목표의 도달 정도가 어느 정도인지, 다음의 학습을 수행하기 위해서 어떤 학습계획을 세워야 하는지 방향의 제시가 포함된 피드백의 제공이 필요하다. 한국교육과정평가원에 따르면 이러한 피드백은 학생이 평가 도구 또는 수행 과제에 대해 기억하고 있을 때 제공하는 것이 좋으므로 평가한 지 2주 이상 지난 후에 주어지는 피드백은 효과가 줄어들게 된다. 따라서 학생이 성장을 위해 학습할 시간이 남아있을 때 제공하는 것이 바람직하다.

본 연구에서는 학생에게 과제가 제시되는 시점부터 과제를 제출한 학생이 피드백을 제공받는 것까지 걸리는 시간을 최대 7일로 하여 [Fig. 18]과 같은 피드백 양식을 학생에게 제공할 것을 제안한다. 피드백에는 다음과 같은 내용이 들어있다. 피드백에 현재 학생이 학습한 명령은 치환 명령까지 학습이 되었으며, 학습을 통해서 경회루의 기둥을 어떤 수준으로 제작할 수 있는지에 관한 내용이 담겨있다. 또한 학습이 필요한 명령은 격쇠치환 명령이며, 학습을 통해서 학생이 경회루의 기둥을 만들 때 어떤 도움을 받을 수 있는지가 나타나 있다. 이러한 학습이 필요한 명령을 학생이 학습하게 하려면 격쇠치환 명령에 대한 강의 영상 링크로 제공한다. 마지막으로 학생이 강의 영상을 듣고 난 후 격쇠치환 명령에 대한 학습을 재진단할 수 있는 보충 문제를 [Fig. 19]와 같이 제공한다.

Question 53

다음은 경회루의 기둥과 이를 만드는 명령어입니다. 제시된 명령어를 **짧게 줄여** 쓴 명령어를 고르세요.



① doit([2s] 3u) ② doit(2s [3u])
 ③ doit([2s] 3d) ④ doit(2s [3d])

[Fig. 19] Examples of Supplemental evaluation question

V. 결론 및 제언

코드를 효율적으로 작성하는 과정에는 많은 수학적 지식과 역량이 쓰이며, 또한 코딩은 수학적 과정을 구체화하며 익힐 수 있는 좋은 활동이다(Kim et al., 2018). 이러한 이유로 코딩과 수학은 서로에게 도움이 되며, 특히 고등학교 인공지능수학 및 중학교 자유학년제 융합교육 등과 관련하여 코딩과 수학이 적절히 융합된 코딩수학을 연구하고 개발할 필요성이 있다. 이러한 본 연구의 필요성에 따라 개발된 경회루 코딩수학 교육과정과 교육방법 그리고 현장에 적용한 연구의 결과는 다음과 같다.

첫째, 수학교과와 정보교과를 융합한 코딩수학 교육과정 및 효과적인 교육방법을 통해 학생은 4차 산업혁명 시대에 필요한 미래 역량인 컴퓨팅 사고력을 함양할 수 있다. 경회루 코딩수학 교육과정은 터틀말을 이용한 쌓기나무 학습의 평가 및 지도방안에 관한 선행연구(Cho, Cho, 2017)를 바탕으로 한 코딩교육을 3차원 코딩환경 터틀크래프트를 활용하여 좌표, 순서쌍, 좌표평면이라는 수학적 개념을 코딩과 융합하여 개발하였다. 이러한 경회루 코딩수학 교육과정은 같은 조형물이라도 다양한 명령을 이용하여 제작할 수 있다는 점을 이용하여 학생이 숙달한 명령의 정도와 관계없이 학습에 참여할 수 있어 학습에 대한 접근성이 낮다는 장점이 있다. 또한, 이는 누구나 습득한 명령을 이용하여 터틀크래프트 안에서 유형의 아이디어를 만들어 낼 수 있다는 것으로 연결되어 교사에게 학생이 작성한 명령을 보고 학생의 학습 수준에 대한 진단을 쉽게 한다. 명령을 줄이는 과정에서 컴퓨팅 사고력의 주요한 특징인 추상화 및 자동화가 내포되어 있으며, 추상화 및 자동화를 거친 명령은 계수, 치환, 집합 등 수학적 표현을 포함하기 때문에 수학교과와 정보교과의 융합이라 할 수 있으며 융합 교육의 실천에 의미가 있다.

Jeong, Cho(2020)는 가장 적은 코드를 사용하여 조형물을 만드는 ‘최소코드 게임’을 이용한 지도전략을 연구하였는데, 이는 컴퓨팅 사고력의 신장, 수준 상승을 유도하며, 학생 답안으로 다양한 수준을 관찰할 수 있음을 보였다. 본 연구는 선행연구를 통해 최소 코딩게임을 이용하여 학생들이 학습에 대한 수준 상승을 유도할 수 있는 최소 코딩게임 교수·학습 전략을 제안하였다. 나아가 이 전략을 이용한 평가 문항의 제작은 학생의 학습 수준을

관찰할 수 있게 하여 학생의 수준 상승 여부를 파악할 수 있게 한다. 학생은 경회루 코딩수학 교육과정에서 구성요소에 따라 수준이 다른 명령을 다루며, 수준이 높은 명령, 즉 더 짧은 명령일수록 컴퓨팅 사고력의 추상화 및 자동화 과정이 이루어진다. 따라서 최소 코딩게임 교수·학습 전략에 의해 학생은 수준이 낮은 명령에서 수준이 높은 명령을 사용하는 수준 상승이 일어나게 되며, 이 과정에서 컴퓨팅 사고력을 신장시킬 수 있다.

둘째, 본 연구에서 제안하는 교육과정과 평가는 모두 앞서 소개한 코딩수학 홈페이지를 이용한 온라인 시스템을 기반으로 한다. 코로나19로 인한 학생들의 교육격차를 줄이기 위해서 온라인 교육환경이 효과적으로 갖춰질 필요가 있다. 본 연구는 학습부터 평가와 피드백까지 온라인 시스템을 통해 진행하는 교육과정을 제안하기 때문에 장소와 시간에 구애받지 않고 학습에 참여하기를 원하는 학생은 누구나 코딩수학 홈페이지를 이용할 수 있도록 개방하고자 한다. 이는 장기적으로 많은 학생의 답안을 데이터로 얻을 수 있다는 장점으로 연결되기 때문에 추후에 보완하거나 수정할 때 학생의 답안을 바탕으로 보완점 및 개선점을 파악하는 데 도움이 될 수 있다.

셋째, 터틀크래프트 3차원 코딩환경을 활용한 코딩수학 교육과정은 초등학교의 수학교과, 중학교의 정보 및 수학교과와 연결된다. 구체적으로 터틀크래프트 3차원 코딩환경은 초등학교 6학년 수학교과에 등장하는 쌓기나무와 중학교 1학년 수학교과에 등장하는 좌표, 순서쌍, 좌표평면과 연결되고 있으며, 중학교 1학년 수학교과에 등장하는 변수는 중학교의 정보교과에 등장하는 변수와 연결되고 있다. 이는 중학교 교육과정인 자유학년제 수업에서 경회루 코딩수학 교육과정 및 최소 코딩게임 기반 교육방법을 운영하기에 적합함을 알 수 있으며 이는 추후에 중학교 자유학년제 수업 연구에 시사점을 제공한다.

또한, 경회루 코딩수학 교육과정에서 사용하는 3차원 코딩환경인 터틀크래프트는 교육용 프로그래밍 언어 중 고급언어인 자바스크립트 및 파이썬과 연결된다. 파이썬은 텍스트 기반 프로그래밍 언어로 고등학교의 정보교과에서 다루며, 고등학교 수학과 교육과정 중 인공지능수학과도 연결이 가능하다. 수학교과와 정보교과를 융합한 교육과정은 고등학교 교육과정에서 다루는 프로그래밍 언어를 초등학교와 중학교에서 터틀크래프트를 통해 미

리 익숙해질 수 있다는 장점이 있으므로 이에 대한 융합교육의 후속 연구가 필요하다.

한편, 본 연구는 2차년도 연구 및 3차년도 연구를 바탕으로 교육과정과 교육방법을 제안하고 있는데, 개발한 코딩수학 교육과정, 최소 코딩게임 교수·학습 전략, 평가 및 피드백을 많은 학생에게 적용하지 않았다는 한계가 있다. 따라서 이는 현실과의 격차가 발생할 것으로 예상되기 때문에 더 많은 학생에게 적용한 다음 발생하는 문제를 보완·수정할 방법을 구상할 필요가 있다. 예를 들어, 본 연구에서는 수준이 가장 낮은 doit 명령어와 바로 이어지는 치환 명령이 격차 명령보다 수준이 낮은 명령이라 판단했으나, 학생에 따라서는 격차 명령이 더 쉽게 받아들여질 수 있다. 이러한 가정은 명령의 수준 및 위계를 다시 설정해야 하는 문제가 발생한다. 따라서 학생의 응답에 관한 빅데이터를 분석해보고 명령의 수준 및 위계를 구체적으로 설정할 필요성이 있다. 따라서 본 연구에서 제안하는 교육과정과 교수·학습 전략, 평가 및 피드백을 현장에 더 적용해보고 이에 발생하는 문제를 보완한 교육과정 및 평가 연구가 계속될 수 있기를 기대해 본다.

참 고 문 헌

- An, C. H., Lee, K. B. & Moon, S. J. (2018). Programming learning method for beginner based on Entry Block-based/Text-based coding. *The Society of Convergence Knowledge Transactions*, 6(1), 127-134.
- Chang, S. Y., Koh, E. H. & Park, I. W. (2013). Tendency and Task of Educational Methodology Research. *Research of educational methodology*, 25(3), 603-621.
- Chang, K. Y. (2017). A Feasibility Study on Integrating Computational Thinking into School Mathematics. *School Mathematics*, 19(3), 553-570.
- Cho, A. R. & Cho, H. H. (2017). A Study on Evaluation and Teaching of Building Block Curriculum Based on Executable expression. *Journal of Learner-Centered Curriculum and Instruction*, 17(18), 513-536.
- Choi, I. Y. (2020). Development and Application of a 3D Coding Environment for Mathematics Learning. *Journal of Educational Research in Mathematics*, 30(2), 199-225.

- Choi, J. W., Lee, E. K., Kim, K. H. & Lee, Y. J. (2015). Suggestios for Assessment of Students' Computational Thinking in 2015 Informatics Curriculum. *The Korean Association of Computer Education Conference Papers, 19(2)*, 9-12.
- Chung, I. W. (2020). *A Study on Mathematical Design Activities Using 3D Coordinate System-Based Coding Environment : focusing on mathematical learning by different code expressions*. Master's thesis. Seoul National University, Seoul.
- Han, C. G. (2018). *Computational Thinking*. Gyeonggi: Bae-Um Teo.
- Hur, Y. J. (2008). Inquiry on possibility of integration of traditional and constructional education method : On Whitehead's 'Rhythm of Education'. *The Korean Journal of Educational Methodology Studies, 20(2)*, 91-107.
- Hwang, Y. H., Mun, K. J. & Park, Y. B. (2016). Study of Perception on Programming and Computational Thinking and Attitude toward Science Learning of High School Students through Software Inquiry Activity: Focus on using Scratch and physical computing materials. *Journal of the Korean Association for Science Education, 36(2)*, 325-335.
- International Society for Technology in Education (ISTE) & Computer Science Teachers Association (CSTA). (2011). *Computational Thinking in K-12 Education teacher resource 2nd Ed*. Retrieved from Coding Education Command: Focusing on Algebra <https://www.iste.org/explore/computational-thinking/computational-thinking-all>
- Jeong, J. H. (2015). *Mathematics Education of Pattern Generalization by Computational Thinking Game*. Master's thesis. Seoul National University, Seoul.
- Jeong, J. H., Cho, H. H. (2020). Mathematising of Education. *Journal of Educational Research in Mathematics, 30(1)*, 131-151.
- Jeong, Y. S., Kang, S. C., Kim, M. G., Seo, M. R., Son, Y. K., Song, S. L., ..., Lim, J. S. (2020). *Middle school information textbook*. Seoul: Ssi-Ma Seu.
- Ju, M. J. (2021). *Development of Teaching and Learning Materials Using AlgeoMath Block Coding for Theme Selection Activity of Free Semester System*. Master's thesis. Korea National University of Education, Chungbuk.
- Kim, G. J. & Han, J. S. (2017). 4th Industrial Revolution and Mathematics Education. *Journal of the Korean Multimedia Society, 21(4)*, 33-38.
- Kim, H. J., Hong, O. S., Cho, H. S. & Im, S. M. (2013). An Analysis of Change on Science Interest and Self-directed Learning through STEAM Educational period. *Journal of Learner-Centered Curriculum and Instruction, 13(3)*, 269-289.
- Kim, H. K., Park, S. W., Park, J. L. & Jung, Y. J. (2014). *How do we evaluate students' creativity and character? Guidebook, PIM 2014-7*. Seoul: Korea Institute for Curriculum and Education.
- Kim, J. C. (2002). A Critical Examination of the Korean National Curriculum Research and Development System. *The journal of Curriculum Studies, 20(3)*, 77-97.
- Kim, N. R., Seo, Y. H., Cho, H. H. (2018). Coding Mathematics Contents and Environment Design - Focusing on mathematization and computational thinking. *Journal of Learner-Centered Curriculum and Instruction, 18(4)*, 647-673.
- Kim, Y. O., Ko, S. E., Kim, Y. R., Park, H. S., Lee, S. O., Jang, J. O. & Cho, D. S. (2017). *A study on the content of mathematics learning suitable for the future talent- Research report*. Korea Foundation for the Advancement of Science and Creativity. Retrieved from <http://kms.kofac.re.kr/openapi/kofac.html>
- Ko, J. K. (2018). *A Study on the Development of Primary Software Education Program using 3D modelling*. Master's thesis. Seoul National University of Education, Seoul.
- Korea Creative Content Agency (KCCA). (2013). *Functional game status and Research on activation methods - KOCCA Research Report 13-11*. Retrieved from <https://www.kocca.kr/cop/bbs/list/B0000147.do?menuNo=201825>
- Kwon, O. N. (2018). The 4th Industrial Revolution and Future Mathematics Education. *education plaza, 67*, 10-13.
- Lee, J. H. & Jang, J. H. (2018). Exploration for Developing Assessment Tools for Computational Thinking. *Journal of Creative Information Culture, 4(3)*, 273-283.
- Ministry of Education, Science and Technology (MEST). (2010). *Business Report of MEST for 2011*. Retrieved from <https://www.moe.go.kr/boardCnts/view.do?boardID=72713&lev=0&statusYN=W&s=moe&m=03&opType=N&boardSeq=80770>
- Ministry of Education. (2015a). *Confirmation and presentation of 2015 revised curriculum*. Press Release (2015.09.23.). Sejong: Ministry of Education.

- Ministry of Education. (2015b). *Department of Information curriculum. Notice No. 2015-74 [a separate book 10]*. Sejong: Ministry of Education.
- Ministry of Education. (2015c). *Department of Mathematics curriculum. Notice No. 2015-74 [a separate book 8]*. Sejong: Ministry of Education.
- Ministry of Education (2015d). *Middle School Free-Semester System Implementation Plan*. Sejong: Ministry of Education.
- Ministry of Education. (2020). *Artificial intelligence, into school!*. Press Release (2020.09.11.). Sejong: Ministry of Education.
- Ministry of Education & Korea Foundation for the Advancement of Science & Creativity. (2018). *Mathematics Course-Centered Evaluation Training Materials -Middle School-*.
- Nam, C. M. & Kim, C. W. (2011). An Analysis of Teaching and Learning Activities in Elementary Mathematics Based on Computational Thinking. *Educational Science Research*, 13(2), 325-334.
- National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. Washington, DC: National Academies Press.
- Papert, S. (1972). Teaching children to be mathematicians versus teaching about mathematics. *International journal of mathematical education in science and technology*, 3(3), 249-262.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Park, M. G., Lim, H. J., Kim, J. Y., Lee, K. H. & Kim, M. K. (2020). The effects on the personalized learning platform with machine learning recommendation modules: Focused on learning time, self-directed learning ability, attitudes toward mathematics, and mathematics achievement. *The mathematical education*, 59(4), 373-387.
- Park, N. S. (2018). *Identifying Impact Factors on Computational Thinking in the Game-based Preschooler Software Education*. Doctoral Dissertation. Ewha Womans University of Education, Seoul.
- Shin, D. J. & Go, S. S. (2019). A study on investigation about the meaning and the research trend of computational thinking(CT) in mathematics education. *The mathematical education*, 58(4), 483-505.
- Shin, S. R. (2017). *A Study of Cognitive Diagnostic Assessment of Computerized Multistage Testing : Focusing on a Sequence of Numbers*. Doctoral Dissertation. Seoul National University, Seoul.
- Sim, H. S. & Park, M. G. (2019). The Effects of Computational Thinking-based Instruction Integrating of Mathematics Learning and Assessment on Metacognition and Mathematical Academic Achievements of Elementary School Students. *Education of Primary School Mathematics*, 22(4), 239-259.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Royal Society of London Philosophical Transactions A*, 366(1881), 3717-3726.
- Yang, A. K., Cho, H. J. (2009). An Analysis on the Influence of Self-Regulated Learning upon Academic Achievement. *Journal of Educational Research & Implementation*, 8(3), 61-82.
- Yang, M. K. (2002). Exploring the characteristics of curriculum : a Constructivistic point of view. *The Journal of Curriculum Studies*, 20(1), 1-16.
- Yu, G. Y. (2017). *Development and Application of a Computational Thinking Program in Elementary Mathematics*. Master's thesis. Seoul National University of Education, Seoul.