

컴퓨팅 사고 신장을 위한 놀이중심 SW교육 교수학습 전략

한선관

경인교육대학교 컴퓨터교육과

요 약

본 연구는 소프트웨어 교육의 방법으로 놀이중심 SW교육 교수학습전략이 컴퓨팅 사고력 신장에 미치는 영향을 살펴보기 위한 것이다. 놀이중심 SW교육 교수학습전략은 주어진 컴퓨터 코드를 놀이 과정을 통하여 구체화하는 노력을 한 뒤 기능을 수정하고 새롭게 재구성하는 수업 전략이다. 우선 놀이중심 SW교육 교수학습전략을 적용한 교육 프로그램을 설계하여 초등학생들에게 투입하였다. 연구의 효과성을 검증하기 위하여 컴퓨팅 사고 개념과 컴퓨팅 사고 실습의 역량검사를 분석한 결과, 제안된 교수학습전략이 직접교수법보다 효과적이었으며 컴퓨팅 사고에 긍정적인 결과를 가져온 것으로 확인되었다.

키워드 : SW교육, 컴퓨팅 사고, 코딩교육, 놀이중심 SW교육 교수학습전략, 직접교수법

Play-based SW Education Teaching-Learning Strategy to Improve Computational Thinking

SeonKwan Han

Dept. of Computer Education, Gyeong-in National University of Education

ABSTRACT

This study investigates the effect of play-based software education instruction strategy for improving computational thinking as a method of software education. The play-based instruction strategy is a teaching-learning strategy in which the presented code is navigated through play, and the functions are revised and newly reconstructed. We designed an education program with the play-based instruction strategy and applied the program to elementary students. In order to verify the effectiveness of the research, we examined the competence of computing thinking concept and computing thinking practice. As a result, the computing thinking concept and computational thinking practice were that the proposed teaching-learning model is higher than the direct teaching method.

Keywords : Software Education, Computational Thinking, Coding Education, Play-based SW Education Instruction, Direct Method

1. 서론

정부는 SW교육의 활성화를 목적으로 2018년부터 중학교 교육과정에 정보 교과를 정규 과목으로 지정하고 2019년에는 초등학교 정규교육과정에 포함시킬 예정이다[10]. 이에 따라 SW교육을 위한 효과적인 정보 교육 내용과 교육방법이 개발되고 있다. SW교육의 목표가 창의적인 문제해결을 위한 컴퓨팅 사고(Computational Thinking)의 신장에 집중하면서 컴퓨팅 사고는 SW교육의 중심에 섰다.

컴퓨팅 사고에 대한 다양한 정의와 개념들이 발표되고 그와 관련된 다양한 연구들이 진행되었다[1][3][4][7][8][9][11]. 또한 컴퓨팅 사고를 신장시키기 위한 교수학습전략이 발표되면서 학교 현장에 적용되기 시작하였다. 대표적인 연구가 한국교육학술정보원에서 발표한 SW교육 교수학습모델이다[5]. 이 보고서에서는 MIT 미디어랩에서 제안한 창의컴퓨팅 교수학습전략을 바탕으로 학습자 중심의 수업전략을 5가지의 유형으로 제시하였다.

그러나 이 5가지 전략은 외국 연구 사례를 바탕으로 하는 이론적 수업모형으로 실제적으로 교육현장에 적용되어 컴퓨팅 사고의 신장에 효과가 있는지에 대한 검증은 부족한 상황이다. 특히 놀이중심 SW교육 교수학습전략과 같이 학생중심의 코딩 활동이 직접교수방법보다 컴퓨팅 사고에 효과적인지에 대한 연구는 찾아보기 힘들다.

따라서 본 연구에서는 코딩과 같이 상위 인지기능과 문제해결 역량을 필요로 하는 교육내용을 놀이중심 SW교육 교수학습전략으로 현장 수업에 효과적으로 적용 가능함에 대한 연구를 제시하고자 한다. 직접교수법과의 비교를 통하여 놀이중심 SW교육 교수학습전략이 컴퓨팅 사고의 신장에 긍정적인 효과가 있는지 밝혀본다.

2. 이론적 배경

2.1 SW교육과 컴퓨팅 사고

각 교과마다 추구하는 것들이 교과의 본질에 접근할 수 있는 사고력의 신장에 있는 만큼 SW교육도 창의적 문제해결을 위한 컴퓨팅 사고에 집중하고 있다. 시모어 페퍼트(S.

Papert)에 의해 제안된 컴퓨팅 사고력은 자넷 윙(J. Wing)의 연구에 의해 전 세계에 확산 되었다. 자넷 윙이 제시하는 컴퓨팅 사고력은 주어진 문제를 수립하고 해결방법을 설계(추상화)하여 컴퓨팅 시스템을 통해 효과적으로 수행되도록 표현(자동화)하는 사고 과정으로 정의하였다[3]. 컴퓨팅 사고력은 컴퓨터과학자 뿐만 아니라 누구나 가져야 하는 기초 소양(Literacy)이며 읽기, 쓰기, 셈하기와 함께 모두가 갖추어야 할 기본 능력이라고 제안하였다[4].

이후 여러 연구자들에 의해 컴퓨팅 사고력에 대한 정의와 개념들이 발표되고 있으나 명확한 하나의 개념으로 합의되지 못하고 아직까지 다양한 논의들이 오가고 있다[3][4][7][8][9][11].

국내에서는 한국과학창의재단에 의해 ‘컴퓨팅 시스템의 역량을 활용하여 해결하고자 하는 문제를 효과적이고 효율적으로 해결할 수 있는 절차적 사고능력’이라고 정의[8]하였으며 류미영과 한선관은 문제해결을 위해 컴퓨터 과학적인 개념과 원리를 이해하고 절차적인 해결 과정을 통해 컴퓨팅 역량을 효과적으로 수행하는 능력이라고 정의하였다[9].

국외에서는 자넷 윙이 제시한 추상화와 자동화를 구글이 Decomposition, Pattern Recognition, Abstraction, Algorithm, Automation의 5가지 단계로 구분하였다. CSTA와 ISTE에서는 자넷 윙과 구글의 개념을 더욱 확대하여 컴퓨팅 사고력을 Data Collection, Data Analysis, Data Representation, Problem Decomposition, Abstraction, Algorithms & Procedures, Automation, Simulation, Parallelization의 9가지 하위 개념으로 분류하였다.

2.2 놀이중심 SW교수학습전략

놀이중심 SW교육 교수학습전략은 학습자 중심의 수업방법이며 놀이의 개념에는 플레이(Play), 활용(Use), 탐색(Inquire) 등의 요소를 포함하고 있다. MIT Media Lab의 스크래치 사이트에서 제시한 창의컴퓨팅 교육의 기본 철학은 코딩의 Play를 통해 서로 만나고(meet), 공유하고(share) 배우는(learn) 경험을 하며 프로그래밍을 이해하게 된다[7].

또한, 한국교육학술정보원에서 발표한 SW교육 교수학습 모형 개발 연구는 컴퓨팅 사고의 향상을 위해 5가지 모형을 제시하였다[5]. 세부적인 내용은 <Table 1>과 같

다. 5가지 모형 중에서 재구성 모형인 UMC(Use, Modify, reCreate) 모형은 코딩 기능이 초보 수준의 학생들을 대상으로 하는 수업 전략이다. 초보자에게 컴퓨팅은 컴퓨터의 활용 능력과 기초 소양 지식 그리고 코딩의 경험 부족과 프로그래밍의 인지적 부담이 가중되어 가르치는 교사나 배우는 학생들 모두 쉽게 접근하기 어렵다.

<Table 1> CT Teaching&Learning Model by KERIS

CT Model	T-L Model	Approach
D-M-M	Direct method	Demonstration
U-M-C	Discovery learning	Reconstruction
D-D-D	Inquiry learning	Development
N-D-I-S	Project learning	Design
DPAA(P)	Problem solving learning	CT module

따라서 기존에 개발된 소스코드나 실행 모듈을 이용하여 간단한 놀이 또는 활용을 통해 코딩의 기초적인 지식을 경험하고 스스로 고쳐가며 새로운 코드를 구현해 내는 일련의 활동이다. 재구성중심 모형의 단계는 첫 번째로 배우고자 하는 핵심 개념을 놀이를 통해 학습자들이 탐색하고 교사에 의해 제시된 코드의 수정 단계를 통하여 코딩의 기능과 개념을 이해한다. 이후 자신의 아이디어를 추가하여 기존의 모듈을 수정하는 일련의 재구성 활동을 통해 학생들이 컴퓨팅 사고를 신장하게 된다. 구체적인 단계는 <Table 2>와 같다.

<Table 2> U-M-C Model for CT

Process	Strategy
Use (Play)	Manipulation, experience, play, utilization, navigation
Modify	Additional design, modification, extension, and supplementation
reCreate	Reconstruction, Implementation, Development, Output

이상 개발된 SW교육 교수학습전략의 분석 결과 새롭게 개발된 교수학습 전략의 접근을 통해 현장 교육에 적용 가능한 교육방법을 제시하였으며 이론중심보다는 실천적인 방법을 보여주었다. 한 가지 아쉬운 점은 실제 적용을 통한 구체적인 사례의 개발과 그에 따른 효과의 검증이 부족한 점이었다.

2.3 선행연구 분석

SW교육 교수학습과 관련된 선행연구를 살펴보면 MIT 대학의 미디어랩과 K. Brennan이 제안한 창의 컴퓨팅 수업 전략이다. 창의 컴퓨팅 가이드북에서 제시한 CT 요소 즉 개념(CT Concept), 실습(CT Practice), 관점(CT Perspective)을 신장시키기 위해 놀이와 게임을 적용한 디자인기반 학습(Design-based Learning)을 제시하였다[6]. DBL학습의 전략은 컴퓨팅 사고를 바탕으로 문제를 디자인하고 개인적으로 의미 있는 것을 산출하며 산출물을 타인과 공유하여 성찰의 과정으로 되돌아보기의 네 단계로 제시되어 있다[6].

이은경은 교육용 로봇을 이용한 코딩교육 교수학습전략을 제시하였다. CT 능력의 신장을 지원하기 위해 4 단계의 교수학습전략 즉, 과제 탐색-지원-과제 해결, 통합의 단계로 이루어졌다[2].

김수환과 한선관은 컴퓨팅 사고를 신장시키기 위한 디자인 기반 학습 전략을 제안하였으며 MIT미디어랩에서 제시한 모형을 한국형으로 수정하여 제시하고 있다. 학생 중심의 수업전략으로 디자인을 중심으로 하는 문제 해결 절차로 의미 있는 산출물을 만들며 긍정적인 학습 효과를 보여주었다[12].

전수진과 한선관은 예비교사를 대상으로 UMC 모형을 적용하여 그에 따른 컴퓨팅 사고 신장 능력을 분석하였다. 분석결과 Use단계와 Create 단계의 탐색과 자기주도적 개발과정이 CT능력 향상에 도움을 주고 있음을 보여 주었다[13]. 이 연구에서 제시하는 UMC 모형은 예비교사를 위한 수업의 전략으로 코딩의 문법적인 표준 프로토타입의 사례를 학습자들에게 적용하여 수정하고 새롭게 개발하는 전략을 사용하였다면 제안된 모형은 학습자가 초등학생임을 고려하여 Use 단계를 놀이 중심의 콘텐츠로 제시하여 학습자들의 흥미와 난이도를 조절하는 방법이 차이가 있다.

이러한 선행연구들을 분석하면 평가의 방법이 복잡하고 실천적 사례를 통한 경험적 측정 그리고 성인을 대상으로 진행된 연구 결과로 2018년부터 정규교과가 되는 정보 교육의 방법으로 우리나라 교육 환경에 맞는 적용 결과가 필요하다.

3. 연구 내용 및 방법

3.1. 연구 절차 및 내용

연구절차는 우선 연구 가설을 설정하고 그에 따른 놀이중심 SW교육 교수학습전략을 분석하였다. 놀이중심 SW교육 교수학습전략은 교육학술정보원에서 제시한 U-M-C 모형을 이용하였으며 초등학생 수준에 맞는 형태로 수정 보완하였다.

그리고 놀이중심 SW교육 교수학습전략을 적용하기 위한 교육프로그램을 개발하여 초등학생들에게 투입하였다. 이후 검사지와 프로젝트 산출물 평가를 통하여 결과를 분석하였다. 연구 내용이 되는 가설은 다음과 같다.

첫째, 놀이중심 SW교육 교수학습전략은 학생들의 컴퓨팅사고 신장에 도움을 주는가?

둘째, 놀이중심 SW교육 교수학습전략이 직접교수법보다 학생들의 컴퓨팅 사고 신장에 효과적인가?

3.2 교육 프로그램 설계

놀이중심 SW교육 교수학습전략을 적용하기 위한 교육프로그램 10주제(20차시)를 개발하였다. 개발된 교육내용은 직접교수법을 적용한 비교집단에게도 적용할 수 있는 주제로 선정하였다.

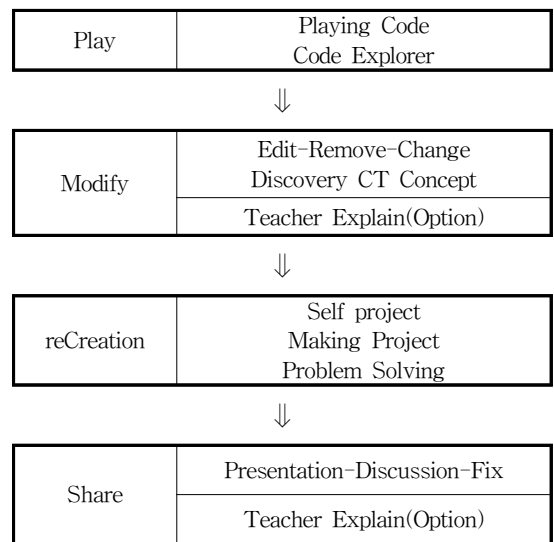
연구 참여집단이 코딩의 경험이 없기 때문에 스크래치의 기초부터 다룰 수 있는 내용으로 구성하였으며, <Table 3>처럼 문법을 고르게 학습할 수 있는 주제로 구성하였다. 놀이중심 SW교육 교수학습전략은 각각의 차시마다 해당 학습에 필요한 예제 파일을 제공하였다. 직접교수법의 경우 해당 차시에 대한 교사시범과 따라하기의 학습 절차와 수업내용을 마련하였다.

<Table 3> Coding Education Program

No	Subject	Contents
1	Coding Explorer	Scratch basic, Talking, Sprite, Moving, Rotating
2	Story Book	Sequence, Event Text, Loop, Story-telling
3	Dancing Show	Sprite, Sound, Animation Parallelism, Shape, Pause
4	Drawing Shape	Pen, Value, Angle Star, Rectangle, Triangle

No	Subject	Contents
5	Meeting Some	If condition, Racing Double Loop, Event
6	Breaking Brick	Design, Animation, Copy Condition, Loop, Broadcast
7	Fun Math	Calculate, Event, Module Operator, Logic, Compare
8	Multimedia	Video, Timer, Picture, Sound, Recording
9	Big Score	Game, Variable, List, Multi condition, x-y Axis
10	Remixing	Procedure, Function, Game Math theory

놀이중심 SW교육 교수학습전략은 한국교육학술정보원에서 개발한 교수학습모델[6]의 절차를 따랐다. 수정된 놀이중심 SW교육 교수학습전략의 절차는 (Fig. 1)과 같다.



(Fig. 1) UMC Learning Process

문법적인 이해와 컴퓨터과학의 내용을 전달하기 위해 Play 과정에서는 주어진 놀이형 코드를 이용하여 교사의 개입 없이 학습자 스스로 코드를 탐색하고 즐겨보는 활동을 하였고, Modify 과정에서 필요한 경우 교사가 참여하여 컴퓨터과학 및 코딩 관련 개념과 지식을 안내하였다. 또한 reCreation 과정이 끝나고 프로젝트 발표와 공유시간에 학습자들 스스로 문법적인 부분의 개념을 설명하고 오류를 상호 토론하여 수정하는 시간을 갖도록 하였다. 수업 종료전 교사의 주도하에 수업에서 요구하는 지식을 피드백하는 시간을 가졌다.

3.3 컴퓨팅 사고 신장 교육프로그램의 적용

연구대상은 경기도 소재의 E초등학교 5학년 학생 47명을 대상으로 하였다. 실험반 23명, 비교반 24명의 2개 반을 선정하였으며, 연구 참가자는 코딩교육의 경험이 없는 학생들로 구성되었다.

컴퓨팅 사고 신장을 위해 개발된 교육프로그램은 창의체험활동 시간을 활용하여 1주에 2시간씩 10주에 걸쳐 20시간을 실험반과 비교반에 적용하였다. 비교반과 실험반의 수업 격차를 줄이기 위해 동일한 한명의 교사가 2개 반을 다른 교수방법을 적용하여 지도하였다. 10주가 끝난 이후에 사후 검사를 실시하였다. (Fig. 2)는 연구에 참여한 학생들을 대상으로 코딩 수업을 실시한 장면이다. 스크래치를 이용하여 제시된 프로그램을 수정하며 새롭게 프로젝트를 구성하는 수업에 대한 사진을 보여준다.



(Fig. 2) Scenes of Coding Class

4. 연구결과 분석

4.1 검사 개요 및 검사지

본 연구에서 개발된 놀이중심 SW교육 교수학습전략이 컴퓨팅 사고 신장에 효과가 있는지 실험군과 비교군을 대상으로 대응표본 t-검정(Paired t-test)을 실시하였다. 분석도구로는 SPSS를 사용하였다.

두 집단의 동질성을 비교하기 위하여, 사전 검사로 수학교과와 학업성취도 검사를 실시하였다. 사전 동질성 검사로 수학교과를 선정한 이유는 컴퓨터과학의 지식과 코딩의 경험이 없는 학생들에게 바로 코딩 검사를 하기 어려웠고 컴퓨팅 사고와 밀접한 교과가 수학교과이기 때문이다. 이에 대한 근거는 미국 과학교사 협회에

서 제시한 STEAM교육 연구 분석(<http://ngss.nsta.org>)에서 제시한 것처럼 컴퓨팅 사고의 기반이 수학에 있다는 점을 바탕으로 하였다. 사전 동질성 검사 결과는 <Table 4>와 같다.

<Table 4> Pre-homogeneity t-test result

Group	N	Avg	SD	t	p
Experimental group	23	64.43	15.57	1.633	.892
Comparison group	24	61.82	17.38		

수학 학업성취도 사전검사에서 실험집단의 평균 점수는 64.43점이었고, 비교집단의 평균 점수는 61.82점으로 비슷한 점수를 보였으며 유의수준 5%에서 p값이 0.892로 .05보다 크므로 통계적으로 유의미한 차이가 없는 것으로 나타나 두 집단이 동질 집단임을 알 수 있었다.

놀이중심 SW교육 교수학습전략을 적용한 후 컴퓨팅 사고를 측정하기 위한 사후검사 도구는 Karen Brennan이 제시한 CT 프레임워크[7]의 CT 개념(Concept)과 CT 실습(Practice)의 기준을 이용하여 개발하였다.

<Table 5> CT Framework of Media Lab[7]

Category	Concept
CT Concept	Sequence, Loops, Parallelism, Events, Conditionals, Operators, Data
CT Practice	Experimenting and iterating Testing and debugging Reusing and remixing Abstracting and modularizing
CT Perspective	Expressing, Connecting, Questioning

<Table 5>에서 컴퓨팅 사고의 개념(CT Concept)은 시퀀스, 반복, 병렬처리, 이벤트, 조건, 연산, 데이터의 7가지 주제가 있다. 7가지 주제에 맞추어 총 14개의 스크래치 코딩 문법 문항을 개발하여 사후 검사 도구로 사용하였다. 14문항이 학습자들의 컴퓨팅 사고 개념을 적절하게 측정할 수 있는지에 대한 타당도는 SW교육전문가 6명을 대상으로 개발된 14문항을 각각 검사하였다. 문항의 타당도는 내용 타당도 검사로 실시했으며 각 검

사 문항에 대해 전문가들 사이에서 동의하는 정도 즉, CVI 지수(Content Validity Index)를 사용하여 타당도를 산출하였다. 전문가의 수가 6명 이상인 경우 CVI 값이 .78 이상이어야 타당하다고 보는데 각 문항들의 CVI 값이 최소 .89에서 최대 .99를 나타냈고 총 CVI값의 평균이 .92 이상으로 나타나 검사 문항의 타당도가 높다고 볼 수 있다.

그리고 컴퓨팅 사고의 실습(CT Practice)은 실험과 반복, 테스트링과 디버깅, 재사용과 재구성, 추상화와 모듈화의 4가지 영역으로 구성되어 있다. CT Practice의 검사는 학생들에게 주제가 있는 자유프로젝트(친구들을 놀라게 할 게임)를 구현하게 한 뒤 <Table 6>의 CT실습 평가지를 이용하였다. 평가지는 각 요소에 부합하는 각 기준의 점수를 20점 만점으로 전문가 3인(컴퓨터 교육 교수, 컴퓨터공학 박사, SW교육 선도학교 교사)이 각각 평가하여 평균점수를 산출하였다.

<Table 6> Practice Checklist(Score)

CT Practice Checklist(Score)						
Class	Student Name					
Category	Concept	5	4	3	2	1 Sum
CT Practice	Experimenting and iterating					
	Testing and debugging					
	Reusing and remixing					
	Abstracting and modularizing					
Uniqueness						Total 20

4.2 결과 분석

사후 검사 결과는 크게 2가지로 나뉜다. CT 프레임 워크에서 제시한 바와 같이 CT Concept 영역과 CT Practice영역으로 CT Concept 영역은 코딩 문법에 대한 검사문항으로 14개 문항으로 구성되어 100점 만점으로 환산된다. 그리고 CT Practice영역은 프로젝트 결과물 평가로 전문가 3인의 점수를 합하여 평균치로 나타낸다. <Table 7>은 CT Concept 영역의 점수를 이용하여 실험반과 비교반으로 t-검정한 것이다.

검사 결과 실험반 학생들의 사후 검사 평균 점수는 74.47이었고, 직접교수법을 적용한 비교반 학생들의 사

후 검사 평균 점수는 68.75로 나타났다. t-검증 결과 유의도는 유의수준 5% 수준에서 p값이 0.025로 0.05보다 작기 때문에 통계적으로 유의미한 차이가 있음을 볼 수 있다.

<Table 7> Post t-test result of CT Concept

Group	N	Avg	SD	t	p
Experimental group	23	74.47	16.41	2.740	.025*
Comparison group	24	68.75	19.08		

* : $p < .05$

따라서 놀이중심 SW교육 교수학습전략을 적용한 집단이 직접교수 모형에 따른 수업을 실시한 비교 집단보다 높은 학업성취도를 보이고 있음을 의미한다. 이러한 점으로 살펴 볼 때 놀이중심 SW교육 교수학습전략이 코딩 교육에서 특히 컴퓨팅 사고의 개념 이해에 효과적임을 알 수 있다.

CT Practice의 검사 결과는 학습자가 산출한 자유프로젝트를 전문가 3인이 20만점으로 각각 평가하여 평균 점수를 산출하였다. 학습자의 산출물 중 미완성으로 평가하기 어려웠던 6명의 작품은 제외하였다. <Table 8>은 CT Practice 영역의 점수를 이용하여 실험반과 비교반으로 t-검정한 것이다.

<Table 8> Post t-test result of CT Practice

Group	N	Avg	SD	t	p
Experimental group	20	16.89	3.07	2.110	.019*
Comparison group	21	14.13	4.03		

* : $p < .05$

검사 결과 실험반 학생들의 사후 CT Practice 검사 평균 점수는 16.89였고, 직접교수법을 적용한 비교반 학생들의 사후 CT Practice검사 평균 점수는 14.13으로 나타났다. t-검증 결과 유의도는 유의수준 5% 수준에서 p 값이 0.019로 0.05보다 작기 때문에 통계적으로 유의미한 차이가 있음을 볼 수 있다.

즉, 놀이중심 SW교육 교수학습전략의 실험집단이 직접교수 모형의 비교집단보다 우수한 프로젝트 작품을

만들었음을 의미한다. 이러한 점으로 미루어 볼 때, 놀이중심 SW교육 교수학습전략이 컴퓨팅 사고의 Practice 영역에서도 효과적임을 알 수 있다.

이로써 놀이중심 SW교육 교수학습전략이 직접교수 전략보다 CT Concept 영역과 CT Practice 영역에서 모두 효과적임을 보였으며 학생들의 컴퓨팅 사고의 신장에 긍정적인 영향을 미칠 수 있는 교수학습전략이라는 것을 보여주었다.

5. 결론

본 연구에서는 소프트웨어 교육의 새로운 교수학습모형으로 놀이중심 SW교육 교수학습전략이 컴퓨팅 사고의 신장에 미치는 영향을 살펴보았다.

연구 결과를 종합하여 보면 다음과 같다.

첫째, 놀이중심 SW교육 교수학습전략은 학생들의 컴퓨팅사고 신장에 많은 도움을 주고 있다. 특히 학생들이 제시된 놀이형 코드를 교사의 개입없이 자기 스스로 탐색하고 수정, 보완하면서 새로운 코드를 생성하는 과정을 거쳐 학습의 몰입을 강화하였다.

둘째, 놀이중심 SW교육 교수학습전략이 직접교수법보다 학생들의 컴퓨팅 사고의 신장에 효과적이었다. CT Concept(개념) 영역 검사에서 직접교수법보다 컴퓨터과학의 개념형성과 코딩의 문법 이해에 도움을 주고 있음을 분석하였고 CT Practice(실습) 영역의 검사 결과에서도 프로젝트 산출물이 직접교수법보다 창의적이고 참신한 결과를 보여주었다.

본 연구 결과에서 살펴보듯이 정보교육이 교사들 주도하의 코딩 문법 교육이나 컴퓨터 과학 지식의 암기 교과가 아니라 창의성을 이용하여 문제를 해결하고 자기주도적인 실습활동이 주요한 구성주의적 교과로 거듭나기 위해서는 본 연구에서 제시하는 놀이중심 SW교육 교수학습전략에 관한 이론적 기반과 함께 실천적 적용이 함께 이루어져야 할 것이다. 놀이중심 SW교육 교수학습전략 외에도 새롭게 제안되는 SW교육 교수학습전략에 대한 실제적인 연구와 적용이 향후 연구 주제로 제시된다.

참고문헌

- [1] Bundy, A. (2007). *Computational thinking is pervasive*. Journal of Scientific and Practical Computing, 1(2), 67-69.
- [2] E. K. Lee(2009), *Robot Programming Teaching and Learning Model to Enhance Computational Thinking Ability*, Korea National University of Education, Doctoral Thesis
- [3] J. M. Wing(2006). *Computational Thinking*. Communication of the ACM, 49(3), 33-35
- [4] J. M. Wing(2008). *Computational thinking and thinking about computing*. Philosophical transactions of the Royal Society A, 366(1), 3717-3725.
- [5] J. S. Kim, S. K. Han, S. H. Kim, S. W. Jung, J. M. Yang, E. D. Jang, J. N. Kim(2016), *Study on Development of Teaching and Learning Model for Software Education*, Korea Education Development Institute & KERIS Research Report CR 2015-35.
- [6] K. Brennan, C. Balch, M. Chung(2015). *Creative Computing Guide Book*, Harvard University
- [7] K. Brennan, M. Resnick(2012). *New frameworks for studying and assessing the development of computational thinking*. In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada. 2012(1), 1-25
- [8] Korea Foundation for the Advancement of Science & Creativity(2014). *A Basic Study for Introduction of Computational Thinking in Elementary and Secondary Schools*, Korea Foundation for the Advancement of Science & Creativity Report 2014.
- [9] M. Y. Ryu, S. K. Han(2015), *Development of Computational Thinking-based Educational Program for Software Education*, Journal of The Korean Association of Information Education, 19(1), 11-20.
- [10] Ministry of Education (2015). *Guidelines for Operating Software Education*, Korea Education and Research Information Service Report CR 2015-3.

- [11] National Research Council (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking*. The National Academics Press.
- [12] S. H. Kim, S. K. Han(2012). *Design-Based Learning for Computational Thinking*, Journal of The Korean Association of Information Education, 16(3), 319-326.
- [13] S. J. Jun, S. K. Han(2016), *Development of UMC Teaching and Learning Strategy for Computational Thinking*, Journal of The Korean Association of Information Education, 20(2), 109-117.

저자소개

한 선 관



1991 경인교육대학교(교육학
학사)
1995 인하대학교 교육대학원(컴퓨터교육학석사)
2001 인하대학교 전자계산공학과
(전산학 박사)
2002~현재 경인교육대학교 컴퓨터교육과 교수
관심분야 : 창의컴퓨팅 교육,
SW교육, 인공지능, ITS,
STEAM교육, 초등정보교육, 미래교육
e-mail : han@gin.ac.kr