

# 컴퓨팅 사고 프레임워크 기반 방과후 스크래치 팀프로젝트 경험의 분석

최형신\* · 정인기\* · 소효정\*\*

춘천교육대학교 컴퓨터교육과\* · 포항공과대학교 창의IT융합공학과\*\*

## 요 약

본 연구의 목적은 방과후 소프트웨어 교육으로 진행된 스크래치 수업에서 학생들의 팀 프로젝트 경험을 분석하고 향후 방과후 수업의 방향을 설정하는 것이다. 이를 위해 한 학기 동안 초등학교 방과후 수업으로 12주차 스크래치 수업을 설계하고 2인 1팀으로 팀을 구성하여 스크래치로 프로젝트를 기획 및 구현해서 발표하는 과정을 진행하였다. 본 연구에서는 3팀을 선정하여 산출물 기반 인터뷰(Artifact-based Interview)를 수행하고 방과후 소프트웨어 수업에서 학생들의 경험을 컴퓨팅 사고(Computational Thinking) 개념, 수행, 관점에 초점을 둔 프레임워크에 기반하여 질적으로 분석하였다. 본 연구는 연구와 실천이라는 두 가지 측면에서 의의를 가진다. 연구 측면에서는 컴퓨팅 사고 프레임워크를 소프트웨어 교육 평가에 활용할 수 있는 방안을 제안하였다는 점이며, 실천적 측면에서는 도출된 학생들의 경험의 질적 분석 자료가 향후 초등 소프트웨어 교육의 내용 및 평가 방법 설계에 시사점을 제공하고 있다는 점이다.

키워드 : 방과후 프로그램, 소프트웨어 교육, 컴퓨팅 사고, 산출물 기반 인터뷰, 스크래치

## Computational Thinking Framework-based Analysis of Afterschool Scratch Team Project Experiences

Hyungshin Choi\* · Inkee Jeong\* · Hyejeong So\*\*

Dept. of Computer Education, ChunCheon National University of Education\* ·

Dept. of Creative IT Engineering, Pohang University of Science & Technology\*\*

## ABSTRACT

This study aims to provide suggestions for software education in an afterschool program, deriving from the analysis of student experiences of working on Scratch team projects. This study reports on the implementation of the 12 week afterschool software education program in an elementary school, where students worked in pairs to learn Scratch programming from ideation to design and presentation. For an in-depth study of student-generated artifacts, we selected three groups' Scratch projects and conducted artifact-based interviews to unpack student experiences working on Scratch projects as a group. Adopting the computational thinking framework as an over-

교신저자 : 정인기(춘천교육대학교 컴퓨터교육과)

논문투고 : 2014-11-04

논문심사 : 2014-11-28

심사완료 : 2014-12-16

arching analytical lens, we focused on examining student experiences from three dimensions of computational thinking (CT), namely, CT concepts, CT practices, and CT perspectives. The present study provides both theoretical and practical implications. Firstly, we demonstrate the feasibility of applying the CT framework for assessing student-generated artifacts in design-oriented software education. We also believe that this study provides important suggestions to future software education programs adopting CT as an overarching design and assessment framework.

Keywords : Afterschool Program, Software Education, Computational Thinking, Artifact-based Interview, Scratch

## 1. 서론

2014년 정부가 공교육 소프트웨어 의무 교육 정책을 발표하면서 소프트웨어 교육에 대한 관심이 높아지고 있다[15]. 지금까지 초등 공교육에서 근본적이고 체계적인 소프트웨어 교육은 부족한 실정이었다고, 그나마도 컴퓨터 활용 교육에 치중되어 온 것이 현실이다. 따라서 지금까지의 컴퓨터 교육이 향후 창조경제의 주력이 되고 있는 소프트웨어를 창의적으로 만들어낼 수 있는 인재의 역량 개발에는 제대로 대응하지 못하고 있다는 인식과 비판이 있다. 이렇게 시대적으로 소프트웨어 교육의 필요성에는 공감대가 형성되어 가고 있으나 구체적 교육 방향과 체계적 접근 방법론에 대해서는 다양한 목소리가 나오고 있다.

소프트웨어 교육의 방향은 그 거시적 목표 설정에 좌우되며 그 결과 또한 목표의 성취로 얻어지게 될 것이다. 이런 맥락에서 소프트웨어 교육의 주요 목표로 제안되고 있는 것이 컴퓨팅 사고(Computational Thinking, 이하 CT) 역량 개발이다[11]. Papert에 의해 처음 설명된 CT는 카네기 멜론 대학교 Wing의 논문에 의해 크게 부각되었다[12][14]. Wing은 ‘Computational Thinking은 컴퓨터과학의 근본적인 개념들을 적용한 문제 해결, 시스템 디자인, 인간 행동의 이해에 관한 것이다.’[14]라고 하였다. Kafai와 Burke는 최근 K-12 교육에서 CT 강화 현상은 프로그램 교육 패러다임의 변화를 보여주는 “Social Turn”으로서, 구체적으로는 (1) 코드 작성에서 어플리케이션 개발, (2) 처음부터 개발하는 것에서 다른 사람의 작업을 리믹스하는 것, (3) 틀 개발에서 커뮤니티 촉진으로, 즉 “참여”를 중심으로 한 문화적 변화가 소프트웨어 교육에서 일어나고 있다고

주장한다[6].

CT의 중요성이 부각되면서 이를 교육하고 평가하는 방법에 대한 연구도 증가해 왔다. 그 예로서 Brennan과 Resnick은 CT 역량을 “과정”에 초점을 두어 평가할 수 있는 CT 프레임워크를 제안하였으며, 이 프레임워크를 기반으로 평가를 수행하는 연구가 증가하고 있다[1]. 그러나 실제 연구 내용을 살펴보면 CT 개념의 습득을 양적으로 평가하는 것으로 제한되는 경우가 많아 아직 CT 프레임워크가 포괄적이며 질적 연구 방법으로 활용된 연구는 부족한 실정이다.

따라서 본 연구에서는 방과후 소프트웨어 교육으로 진행된 스크래치[13] 수업에서 학생들의 팀 프로젝트를 CT 프레임워크를 기반으로 질적 분석하여 학생들의 CT 관련 경험을 총체적으로 평가하고자 하였다. 한 학기 동안 초등학교 방과후 수업으로 스크래치 수업을 설계하였으며, 학생들은 팀을 구성하여 스크래치로 프로젝트를 기획 및 구현해서 발표하는 과정을 진행하였다. 본 연구에서는 방법론적으로 산출물 기반 인터뷰(Artifact-based Interview)를 수행하여 방과후 소프트웨어 수업에서 학생들의 경험을 CT 프레임워크에 기반하여 분석하고 이를 통해 향후 방과후 소프트웨어 교육의 방향에 대한 시사점을 도출하고자 하였다.

## 2. 이론적 배경

### 2.1 컴퓨팅 사고(Computational Thinking) 프레임워크

Brennan과 Resnick은 CT 역량을 과정에 초점을 두

며 다각적 측면으로 평가할 수 있는 CT 프레임워크를 제안하였다. CT 프레임워크는 세 개의 차원, 즉 CT 개념, CT 수행, CT 관점으로 구성된다[1].

첫째, CT 개념(concepts)은 순차 프로세싱, 반복 프로세싱, 병행 프로세싱, 이벤트, 조건 프로세싱, 연산, 데이터의 7 개의 개념을 가진다. 이들 개념은 많은 프로그래밍 언어들에 공통적으로 포함되는 내용이며 비프로그래밍 맥락에 전이되는 개념이기도 하다.

둘째, CT 수행(practices)은 창의적 미디어 프로젝트에서 CT 개념 구성의 과정에 초점을 두며 누적 및 반복, 테스트 및 디버깅, 재사용 및 리믹싱, 추상화와 모듈화를 포함한다. CT 수행 역량은 사고의 과정과 학습의 과정에 초점을 두는 것으로 프로그래밍뿐만 아니라 다양한 미디어 디자인 활동에도 유용하다.

셋째, CT 관점(perspectives)은 창의적인 미디어 프로젝트 경험에서 자신에 대한 이해를 포함하여 타인과의 관계와 테크놀로지 세상에 대한 이해를 반영한다. CT 관점은 표현하기, 연계하기, 질문하기의 세 가지 요소를 가진다.

## 2.2 CT 기반 소프트웨어 교육 평가 연구

소프트웨어 교육 연구에서 교육의 효과성을 검증하는 방법으로 학습 동기, 사고력, 문제해결력, 자기 주도성 평가 등이 사용되어 왔으며, 최근 들어 CT 프레임워크를 기반으로 하는 평가 방법이 도입된 연구들이 진행되기 시작했다[5]. 국내 연구로 김수환과 한선관은 스크래치 프로그래밍 교육이 Computational Literacy(CL) 능력 향상에 통계적으로 유의함을 검증하였는데, CL에는 정보적 문제해결력, self-프로그래밍 능력, self-프로그래밍 흥미도가 포함된다[8]. 한편 이은경은 스크래치 활용 프로그래밍 학습의 효과로 CT 능력과 창의성을 평가하였다[9]. CT 능력 평가를 위해 프로젝트 결과물을 수집하고 CT 평가 프레임워크 중 하나의 영역인 CT 개념의 6가지 요소인 순차, 반복, 조건, 이벤트, 자료, 연산을 평가하였다. 가장 최근의 연구로 최형신은 예비초등교사를 위한 스크래치 프로그래밍 수업 평가 방안으로 CT 세부 요소를 도출하고 이를 평가하기 위한 루브릭을 제안하였다[2][3].

국의 연구로, Brennan과 Resnick은 스크래치 프로그

래밍 효과성 검증의 방법으로 CT 프레임워크를 제안하고 각 요소들을 평가하기 위해 프로젝트 분석, 산출물 기반 면담, 디자인 시나리오 접근 방법을 제시하였다[1]. Davis 등은 중학교 학생들을 대상으로 Makey Makey 콘트롤러를 활용하는 게임을 스크래치로 개발하는 교육 프로그램을 실시하였다. 평가방법으로 Brennan과 Resnick의 CT 프레임워크를 적용하였는데, 모든 학생들의 프로젝트에서 CT 개념인 순차, 반복, 조건 프로세싱, 이벤트가 사용되었음을 발견하였다[4]. Kafai 등은 고등학생을 대상으로 한 e-textiles 수업에서 학생들의 결과물을 CT 프레임워크를 적용하여 평가하였다. 연구 결과 학생들이 프로그래밍의 개념뿐만 아니라, 재사용 및 리믹싱하는 CT 수행과정과 자신을 컴퓨터 과학자로 보기 시작하는 CT 관점의 변화도 관찰되었다[7].

하지만 CT를 적용한 다수의 연구가 CT의 개념 평가에만 국한되는 경우가 많음이 최근 발표된 리뷰 논문에서 확인되었다. Lye와 Koh는 CT를 활용한 중재연구 27편을 분석하였는데, 그중 23(85%)편의 연구가 CT 개념을 평가결과로 보고했음에 반해, 단지 8편의 연구만이 CT의 수행 및 관점을 평가했음을 발견했다[10]. 선행연구를 검토해 볼 때, CT 프레임워크의 개념, 수행, 관점을 총체적으로 고려하는 연구가 더욱 필요함을 알 수 있다.

## 3. 방과후 소프트웨어 교육 프로그램 설계

### 3.1 교육 프로그램

본 연구의 프로그램은 초등 방과후 수업으로 진행할 소프트웨어 교육 프로그램의 형태로 설계되었다. 총 12 주차로 구성된 본 프로그램은 전반부에는 프로그래밍 개요와 스크래치 프로그래밍을 배우고 후반부에는 2인 1팀으로 게임 등과 같은 디지털 콘텐츠를 프로젝트로 기획하고 구현하여 발표하는 과정으로 구성했다. 교수설계적 측면에서 본 프로그램은 스크래치 프로그래밍의 근본적 이해뿐만 아니라, 구체적 디자인 및 개발 경험을 통해 학생들이 프로그래밍 요소의 내재화와 외적 구현을 문제해결 과정을 통해 모두 경험할 수 있도록 수업을 구성하였다. 구체적인 프로그램 주제와 내용은 <Table 1>에 제시되었다.

<Table 1> Software Education Program

Wk	Topic	Activities
1	What is programming?	• Self Introduction • Understanding Programming
2	Having Fishes in a Computer	• Understanding Scratch • Moving Sprites
3	How are you?	• Programming Animation • Conversation
4	Creating a Drawing Pad	• Understanding Input/Output • Understanding Iterations
5	Play the Rock-paper-scissors Game with a Computer	• Understanding using Operators • Understanding Selective Controls
6	Can I Create Games?	• Compose Various Commands • Understanding Programming Design Methods
7	Design Team Projects	• Plan Creative Projects • Write a Project Plan
8	Collaborate with a Peer	• Work on the Project Collaboratively • Set a role in the Project
9	Project Implementation	• Integrate and Complete the Project
10	Presentation Preparation	• Plan a Presentation of the Project • Rehearse the Presentation
11	Presentation	• Deliver a Presentation
12	Reflection	• Reflect on the Project • Write a Reflection

4. 연구 방법

4.1 연구 방법 및 절차

본 연구에서 진행한 소프트웨어 교육 프로그램은 2013년 2학기 한 초등학교 방과후 프로그램으로 12 주간 매주 월요일 오후에 1시간 30분 동안 진행되었다. 본 교육 프로그램의 주제는 ‘Computational Thinking을 활용한 문제해결’이며, 프로그램 진행은 본 연구의 저자 2명이 전반부와 후반부 수업을 나누어 진행하고 초등예비교사 2명이 보조 교사로 참여하였다. 학생은 5학년 7명과 6학년 10명이 방과후 수업에 참여하였다.

4.2 자료 수집

(1) 산출물 기반 인터뷰

본 연구에서의 CT 프레임워크는 단순히 CT 개념의

이해를 양적으로 확인하는 방법을 탈피하여 스크래치 팀 프로젝트 경험의 질적 자료를 수집 및 분석하였다. 특히 연구 방법론적으로 학생들의 팀 프로젝트 산출물에 기반한 인터뷰를 활용하였다. 산출물 기반 인터뷰 (artifact-based interviews)는 학생들의 실제 프로젝트 경험(authentic project experience)에 기반하여 인터뷰를 진행하는 연구 방법으로 학생들로부터 산출물과 관련된 경험과 감정적 요소들을 보다 구체적으로 이끌어 낼 수 있다는 장점을 가지고 있다[1].

인터뷰에는 연구 참여 교수 3명과 방과후 수업 참가 학생 중 6학년 여학생 2명과 남학생 3명이 자발적으로 참가하였으며 2014년 2월 6일에 학교에서 실시하여 시간은 1시간 30분 정도 소요되었다.

인터뷰의 주요 질문 내용은 (1) 컴퓨터 관련 사전 경험 (2) 본 방과후 수업 참여 동기 (3) CT 개념 (4) CT 수행 (5) CT 관점으로 구성되었다. 산출물 관련 질문은 학생들이 2명으로 구성된 팀으로 제작한 프로젝트를 노트북 컴퓨터 스크린을 보면서 이루어졌다.

(2) 스크래치 팀프로젝트

본 연구에서 산출물 기반 인터뷰에 포함된 스크래치 팀 프로젝트는 3개로 모두 게임 형식으로 제작되었다. 구체적인 팀 프로젝트 내용과 화면은 <Table 2>에 제시되었다.

<Table 2> Students’ Team Projects

Team	Project	Main Screen
Hedgehog	Name: ‘Scary Hedgehog’ The thorns of the hedgehog burst balloons. (2 Girls)	
BLACK-scratch	Name: ‘Sweet Potato Fighters’ The sweet potato character fights against other characters and gains scores. (2 Boys)	
Blue	Name: ‘Cookie Run’ The cookie-shape character runs and gains scores by getting treasure. (2 Boys)	

**(3) 인터뷰 프로토콜**

인터뷰를 진행할 때 사용한 CT 프레임워크를 기반으로 인터뷰 프로토콜을 작성하였다. 활용한 질문은 <Table 3>에 제시되었다.

<Table 3> Interview Protocols

Dimension	Interview Questions
CT Concepts	Do you understand each concept of CT?
	Do you know why each concept is useful?
	Can you show an example of the concept in your project?
CT Practices	Did you experience that you created your projects a little bit, then trying it out, and then developing further?
	Did you experience that you found a bug and fixed it as you develop your project?
	Did you experience that you create something new using other people's projects?
	Did you experience that you are putting together collections of smaller parts as you complete your project?
CT Perspectives	Do you feel that you could create and express yourself with Scratch?
	What is your favorite part about Scratch?
	Do you feel that you could create something better when you work with others?
	Do you feel that you could create something for others?
	Do you feel that you could ask questions to make sense of the world?

**5. 연구 결과**

**5.1 학습자 경험**

**(1) 사전 컴퓨터 관련 교육 경험**

인터뷰에 참가한 학생들은 대부분 초등학교에 입학하기 전에 가정에서 컴퓨터를 접하고 동영상, 게임 등의 콘텐츠를 통해 컴퓨터를 경험하였으며 학교에서는 문서 작성 기회를 가진 것으로 나타났다. 전반적으로 컴퓨터 기초 활용교육은 비행식 학습 경험을 통해 습득한 것으로 나타났으며, 소프트웨어 프로그래밍 경험

을 체계적으로 습득한 학생은 없었다.

**(2) 방과후 수업 참여 동기**

본 방과후 수업에 참가한 동기는 부모님의 권유도 있었으나, 자발적 동기 및 관심으로 참여한 학생들이 많음을 알 수 있었다. 학생들은 스스로 프로그래밍을 배우고 싶고 친구가 프로그래밍으로 게임을 만든 것을 보고 신기하고 재밌어 보여서 참여하거나, 향후 프로그래머가 되고 싶어서 등 자발적 동기 및 또래 친구의 영향을 참여 동기로 언급하였다.

**(3) CT 개념**

CT 개념의 이해 정도를 알아보기 위해 앞서 설명된 CT 프레임워크에 포함된 주요 7가지 개념(순차 프로세싱, 반복 프로세싱, 병행 프로세싱, 이벤트, 조건 프로세싱, 연산, 데이터)에 대해 질문하였다.

먼저 산출물 기반 인터뷰에서 모든 세 그룹의 학생들은 게임 속에 포함된 블록들을 CT개념 요소들과 연결시켜 인식하는 것으로 답변하였다. 예를 들어, 필요한 블록을 순서대로 이어 붙이면서 순차 프로세싱을 이해하였고, 동일한 블록의 모음이 반복적으로 필요한 것을 이해하였다. 또한 게임에 등장하는 여러 캐릭터가 동시에 움직이는 것을 구현하면서 병행 프로세싱을 인식하였고, 프로그램을 시작하거나 캐릭터를 움직이게 클릭하거나 키보드를 누르는 등 이벤트를 발생시키는 방법을 숙지하고 있었다.

또한, 캐릭터들 간의 충돌을 감지하면 점수를 올리거나 기회의 수를 모두 사용하면 게임을 종료하는 등의 로직을 구현하면서 조건 프로세싱, 연산, 데이터에 대한 개념을 습득한 것으로 나타났다.

**(4) CT 수행**

CT 수행은 스크래치 팀 프로젝트를 기획하고 구현하는 총체적 경험에서 나타난 사고의 과정과 학습의 과정에 초점을 두고 질문하였다. 먼저 본 방과후 프로그램에서 가장 재밌었던 부분에 대해서는 다른 사람이 만든 게임만 하다가 스스로 게임을 만들 수 있는 것이 재미

있었고, 수업 후반부에서 자신이 원하는 것을 만들어 보는 것이 좋았다고 응답했다.

‘나 스스로 만들고 생각하는 대로 연출할 수 있으니까. 내가 터뜨리고 싶을 때 터뜨리고. 게임 만들어 보는 것이 재밌었어요. 만들어진 게임을 하다가 내가 직접 만들어 보니까 좋았어요.’  
(남학생1)

방과후 수업 중 가장 도전적으로 느낀 것에 대해서는 자기가 원하는 프로젝트 개발 시 새로운 것을 스스로 해 보려고 할 때 어려웠고, 특정 주제가 주어지지 않고 스스로 아이디어를 내는 것이 어렵게 느껴졌다고 응답했다. 또한 스크래치 홈페이지에 공유된 것을 일부 가져와서 자신의 프로젝트로 도입할 때 부분적으로 이해가 잘 가지 않거나 기대한 대로 작동하지 않아 수정하는 작업에 어려움을 느낀 것으로 나타났다.

팀별로 프로젝트를 기획할 때 특정 주제를 어떻게 선정하였는지에 관한 질문에 대해서는 대부분의 학생들이 수업의 전반부에서 배운 내용을 응용해서 새로운 것을 착안하였다고 하였다.

‘슈팅 게임에서 총으로 물건을 맞으면 그 물건이 없어지는걸 보고 착안했어요. 우리 프로젝트에서 풍선을 생각하고... 그 풍선을 터뜨리려면 바늘이 있어야 할 것 같아서 고슴도치를 아래에 깔아야 했고. 그게 메인 게임이 됐는데... 슈팅 게임을 응용해서 고슴도치 게임을 만들었어요.’  
(여학생2)

또한 이미 나와 있는 인기 있는 게임을 만드는데 도전해 보고 싶어서 유사한 주제를 선정하였다고 대답한 학생도 있었다.

### (5) CT 관점

CT 관점에서는 스크래치 팀 프로젝트 기획 및 구현 경험에서 자신에 대한 이해를 포함하여 타인과의 관계에 초점을 두고 질문하였다. 프로젝트를 팀으로 진행하는 과정 중 협력 경험에 대해서 의견 조율, 역할 분담, 성취감, 팀원의 강점 활용, 완성도 향상 등 긍정적인 의견

이 도출되었다.

‘혼자 있으면 자기가 하고 싶으면 하면 되는데 둘이 하면 의견 조율이 필요해요. 그때 논쟁도 있었고 말이 좀 격해질 때도 있었지만 그래도 같이 의견 조율을 해서 하니깐 조금 더 성취감이 높아진 것 같아요.’(여학생2)

‘제가 원래 뭘 하는데 시간이 오래 걸려요. 사람이 없으면 완전 느릿느릿해서 뭘 잘 못 만들어요. 근데 ○○랑 할 때 역할 분담을 해서 여러 게임을 만들었는데... 각자 할 수 있는 것을 해서 게임을 만든 게 기뻐요.’(여학생1)

‘아이디어는 ○○가 내고 제가 그림을 못 그려서 해냈을 텐데... ○○가 그림을 잘 그려준 덕분에 잘할 수 있었어요.’(남학생1)

향후 개선하거나 도전해 보고 싶은 것에 대한 질문에 대해 게임 속에 스토리를 더 넣고 싶고, 보다 재미있고 캐릭터를 추가하여 보다 복잡하고 완성도 있는 게임의 형태로 발전시키고 싶다는 의견이 나왔다.

‘지도를 더 길게 해야 하고 캐릭터를 선택할 수 있게 하고, 캐릭터마다 기능을 다르게 하고, 경우의 수를 다르게 할 수 있을 것 같아요.’(남학생3)

‘오류 수정 후 따로 고칠 것이 있으면 화려하게, 더 재밌게, 그림을 늘리거나, 분량을 늘리거나 캐릭터를 더 늘린다거나 할 수 있을 것 같아요.’(여학생2)

한편 방과후 수업 후에 집에서도 스크래치 공유 사이트에 자주 방문하고 집에서 부모님과도 내용을 공유하며, 다른 나라 아이들이 만든 것들을 보면서 새로운 것을 배우는 등 학생들의 학습이 교실 경험에만 국한되지 않고, 교실밖 환경 및 온라인 커뮤니티로 연결되고 있음을 확인할 수 있었다.

‘애네들은 이런 것들도 만드는데구나. 이걸 발전시켜서 이렇게도 만드는데구나 라고 생각했어요.’  
(여학생1)

또한 스크래치 웹사이트 커뮤니티에서 다른 학생들이 공유한 콘텐츠에 한글이나 영어로 댓글을 달아 본 적도 있고 스스로도 잘 만들면 공유하고 싶다는 자기표현 및 공유에 적극적 의지를 보인 학생도 있었다.

## 6. 결론 및 제언

본 연구에서는 방과후 소프트웨어 교육으로 진행된 스크래치 수업에서 학생들의 팀 프로젝트 경험을 CT 프레임워크를 기반으로 분석해 보았다. 본 연구에서는 CT 개념 이해 정도를 단순한 수치로 확인하는 것을 탈피하여 소프트웨어를 기획하고 구현해 가는 디자인 및 개발의 총체적 팀프로젝트 과정에서 나타난 학생들의 본질적인 경험의 질을 탐색하였다. 아울러 CT 프레임워크를 기반한 구체적 평가방법론으로 산출물 기반 인터뷰를 수행하여 질적 분석을 시도하였다.

CT 프레임워크는 소프트웨어 제작 팀 프로젝트 경험을 질적으로 도출하기에 유용한 접근을 제공해 준다. 또한 산출물 기반 인터뷰는 학생들의 학습 과정을 성찰하고 사고 과정을 가시화할 수 있다는 점에서 도움이 되었다.

본 연구의 결과에서 도출된 내용을 통해 볼 때, 학생들에게 특정 주제가 주어지지 않고 스스로 아이디어를 창출하는 오픈된 학습구조가 처음에는 낯설고 도전이 되었지만 이러한 경험을 통해 스스로 아이디어를 내고 연출할 수 있다는 것에 크게 고무되었고 프로젝트를 완성했을 때 성취감도 높은 것으로 나타났다. 또한 협동적 프로젝트를 수행하면서 소프트웨어 제작 과정에서 자신의 강점과 약점을 인식하게 되고 프로젝트의 완성에 도움이 되도록 타인의 강점을 인식 및 활용하는 행동을 통해 문제해결의 상호의존성 및 협력적 사고의 중요성을 경험하였음을 알 수 있었다. CT가 컴퓨터과학의 근본적인 개념들을 적용한 문제 해결, 시스템 디자인, 인간 행동의 이해라고 정의할 때, 본 방과후 소프트웨어 교육에서 학생들은 CT의 핵심 역량에 접근하고 있다고 판단된다.

향후 CT 역량 개발을 목표로 하는 소프트웨어 교육에 대해 본 연구 결과의 시사점을 다음의 몇 가지로 정리할 수 있다. 첫째, 소프트웨어 교육은 작은 것이라도

학생 스스로 아이디어를 내고 처음부터 마지막까지 기획, 구현, 완성하는 디자인 및 개발의 총체적 경험을 제공하는 것이 바람직하다는 점이다. 이러한 일련의 과정을 통해 학생들은 자율성을 기르고 성취감과 자신감을 느낄 수 있을 것이다. 둘째, 팀으로 소프트웨어 프로젝트를 완성에 가는 경험을 제공하는 것이 문제해결 및 협력적 사고 능력 함양에 바람직하다. 프로그래밍뿐만 아니라, CT가 활용되는 영역은 복잡한 변수 및 다양성을 함축한 문제 영역일 가능성이 높으며, 이러한 복잡한 문제해결 과정에서 협력적 사고는 필수적이다. 협력 과정은 협상하는 역량의 개발과 역할 분담과 타인의 강점을 프로젝트의 요구사항에 연결시키는 훈련을 동반하기 때문이다. 셋째, 소프트웨어 교육을 통해 일정 기간 기본 기능을 익힌 뒤에는 프로젝트 공유 사이트를 통해 타인의 아이디어를 보고 자신의 프로젝트에 재사용(reusing)하고 리믹싱(remixing)하는 경험을 제공하는 것이 바람직하다. 이러한 경험은 학생들의 다양한 상상을 촉진하며, 학생 자신이 가지고 있는 도구의 활용 잠재력을 확장시키게 되기 때문이다. 더불어 현실적으로 소프트웨어 교육을 실행할 수 있는 교과시수 또는 방과후 프로그램의 기회가 제한적이라는 점을 고려할 때, 학생들의 온라인 커뮤니티상에서의 참여적 활동 및 공유를 장려함으로써, 소프트웨어 교육의 장소를 물리적 시간적으로 확대할 수 있을 것이다.

이상을 종합해 볼 때, 본 연구는 디자인에 기반한 소프트웨어 교육의 연구와 실천이라는 두 가지 측면에서 의의를 가진다. 연구 측면에서는 CT 프레임워크를 소프트웨어 교육 평가에 활용할 수 있는 구체적 방안을 제안하였다는 점이며, 실천적 측면에서는 도출된 학생들의 경험의 질적 분석 자료가 향후 초등 소프트웨어 교육의 내용 및 평가 방법 설계에 시사점을 제공하고 있다는 점이다.

## 참고문헌

- [1] Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Paper presented at annual American Educational Research Association meet-

- ing, Vancouver, BC, Canada.
- [2] Choi, H. S. (2013). Study on approaches of teaching and measuring computational thinking. *Korea Association of Information Education*, 4(1), 283-288.
- [3] Choi, H. S. (2014). Developing lessons and rubrics to promote computational thinking. *Journal of The Korean Association of Information Education*, 18(3), 57-64.
- [4] Davis, R., Kafai, Y., Vasudevan, V., & Lee, E. (2013). The education arcade: crafting, remixing, and playing with controllers for scratch games. In Proceedings of the 12th International Conference on Interaction Design and Children(pp.439-442). ACM.
- [5] Grover, S., Cooper, S., & Pea, R. (2014). Assessing computational learning in K-12. In Proceedings of the 2014 conference on Innovation & technology in computer science education(pp.57-62). ACM.
- [6] Kafai, Y. B., & Burke, Q. (2013). The social turn in K-12 programming: Moving from computational thinking to computational participation. In Proceeding of the 44th ACM technical symposium on Computer science education(pp.603-608). ACM.
- [7] Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education*, 14(1), 1-20.
- [8] Kim, S. H., & Han, S. K. (2012). Design-based learning for computational thinking. *Journal of The Korean Association of Information Education*, 16(3), 319-326.
- [9] Lee, E. K. (2013). Computer education curriculum and instruction: Creative programming learning with Scratch for enhancing computational thinking. *Journal of Korean Association of Computer Education*, 16(1), 1-9.
- [10] Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior*, 41, 51-61.
- [11] National Academies of Sciences(2010). Report of a workshop on the scope and nature of computational thinking. Washington DC: National Academies Press.
- [12] Papert, S. (1993). Mindstorms: Children, computers, and powerful ideas. Cambridge, MA: Perseus Publishing.
- [13] Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67.
- [14] Wing, J. M. (2006). *Computational thinking. Communications of the ACM*, 19(3), 33-35.
- [15] Yonhap News(2014.7.22). Mandatory SW education starting from the first-year junior high school students next year.

### 저자소개



#### 최형신

1988 이화여자대학교(전자계산학 학사)

1993 (미) New Jersey Institute of Technology (컴퓨터정보과학 석사)

1994~(미) 존슨 앤 존슨 HCS 근무

2007 이화여자대학교(교육공학 박사)

2008 이화여자대학교 연구교수

2009~현재 춘천교육대학교 컴퓨터교육과 교수

관심분야: 컴퓨팅 사고(Computational Thinking), 뉴미디어기반 학습(스크래치, 교구로봇, 3D 프린팅 활용)

e-mail: hschoi@cnue.ac.kr



**정 인 기**

1988 고려대학교 전산학과(이  
학사)  
1990 고려대학교 대학원수학과 전  
산학전공(이학석사)  
1996 고려대학교 대학원 전산과학  
과 전산학전공(이학박사)  
1997~현재 춘천교육대학교 컴퓨  
터교육과 교수  
관심분야: 컴퓨터과학교육, 프로  
그래밍 교육  
e-mail: inkey@cnue.ac.kr



**소 효 정**

2000 이화여자대학교(특수교육학  
학사)  
2002 (미)Indiana University (교  
수체제공학과 석사)  
2005 (미)Indiana University(교수  
체제공학과 박사)  
2005~2013 싱가포르 난양공대  
Learning Sciences &  
Technologies학과 교수  
2013~현재 포항공과대학교 창의  
IT융합공학과 교수  
관심분야: 모바일 학습, HCI  
e-mail: hyojeongso@postech.ac.kr

