Journal of The Korean Association
of Information Education Vol. 19, No. 2,
June 2015, pp. 225-232

http://dx.doi.org/10.14352/jkaie.2015.19.2.225


# 스크래치 프로그래밍이 예비교사에게 미치는 영향 : 컴퓨팅 사고 및 블룸의 텍사노미 활용 평가

최형신 · 김기범
춘천교육대학교 컴퓨터교육과

**요 약**

본 연구에서는 스크래치 프로그래밍 수업이 초등예비교사에게 미치는 영향을 컴퓨팅 사고력과 블룸의 텍사노미를 활용한 인지적 평가를 통해 확인하고자 하였다. 이를 위해 한 학기 동안 스크래치 프로그래밍 수업을 진행하고 예비교사의 객관적인 프로그래밍 능력을 평가하기 위해 인지적 평가 문항을 블룸의 텍사노미를 기반하여 측정하였다. 또한 초등예비교사들의 컴퓨팅 사고(Computational Thinking) 개념, 수행, 관점에 초점을 둔 설문을 실시하는 한편 팀프로젝트를 가지고 산출물 기반 인터뷰(Artifact-based Interview)를 실시하고 팀프로그래밍 경험을 정성적으로 분석하였다. 본 연구는 프로그래밍 수업을 통한 예비교사의 컴퓨팅 사고를 양적 및 질적으로 보다 포괄적으로 평가하였다는 점에서 의미가 있다. 또한 프로그래밍 능력을 블룸의 텍사노미 체계에 의해 객관적으로 평가해 봄으로써 향후 예비교사 프로그래밍 수업 설계에 대한 평가 측면의 시사점을 제시하고 있다.

키워드 : 컴퓨팅 사고, 스크래치 프로그래밍, 구성주의(Constructionism), 블룸의 텍사노미, 산출물 기반 인터뷰

# The Effects of Scratch Programming on Preservice Teachers: Assessment Utilizing Computational Thinking and Bloom's Taxonomy

Hyungshin Choi · Kibum Kim
Dept. of Computer Education, ChunCheon National University of Education

ABSTRACT

The goal of this study is to assess the effects of Scratch programming classes on preservice teachers by using computational thinking and cognitive evaluations based on Bloom's taxonomy. To pursue this research goal we measured preservice teachers' programming skills using cognitive evaluation items based on Bloom's taxonomy after preservice teachers took one-semester Scratch programming course. In addition, a survey focused on computational thinking (CT) concepts, CT practices, and CT perspectives was conducted. We also conducted artifact-based interviews to unpack preservice teachers' experiences of working on team projects and analyzed their experiences qualitatively. The results of this study are meaningful because we assessed preservice teachers' experiences comprehensively with both quantitative and qualitative methods. In addition, this study provides us with

교신저자 : 최형신(춘천교육대학교 컴퓨터교육과)
논문투고 : 2015-05-27
논문심사 : 2015-05-29
심사완료 : 2015-06-26

implications for evaluation perspectives in designing programming courses for preservice teachers by adopting Bloom's taxonomy scheme.

Keywords : Computational Thinking, Scratch Programming, Constructionism, Bloom's Taxonomy, Artifact-based Interview

## 1. 서론

디지털 사회에서 소프트웨어 관련 역량의 중요성이 미국, 영국, 이스라엘 등의 국가에서 부각되었고 최근 우리 정부도 공교육 소프트웨어 의무 교육 정책을 발표하였다[12]. 소프트웨어 중심 사회를 대비하기 위해 초등학생에게도 21세기 언어인 코딩 능력을 길러주어야 한다는 데 공감대가 형성되고 있다[6]. 가까운 미래에 초등교육 현장에서 이러한 변화를 이끌어 갈 예비교사의 코딩 교육이 더욱 중요해지는 시점이라고 할 수 있다.

본 연구에서는 예비교사를 대상으로 코딩 수업을 한 학기 동안 진행하고 학생들의 컴퓨팅 사고력(CT: Computational Thinking)[10][11]과 프로그래밍 능력을 확인해 보았다. CT 평가 프레임워크를 기반으로 CT 개념, CT 수행, CT 관점 측면을 포괄하는 설문을 구성하고 실시하였다. 또한 객관적인 형태의 평가인 지필 시험지를 블룸의 텍사노미 체제를 활용하여 유목화한 뒤 학생들의 역량을 평가하고자 하였다. 더 나아가 학생들의 팀프로젝트 프로그래밍 과정 경험을 정성적으로 분석해 보고자 하였다. 이를 위해 방법론적으로 산출물 기반 인터뷰(Artifact-based Interview)를 수행하여 인터뷰 내용을 분석하고 이를 통해 향후 예비교사 소프트웨어 교육의 방향 및 평가 방법에 대한 시사점을 도출하고자 하였다.

## 2. 관련 연구

### 2.1 블룸의 텍사노미(Bloom's Taxonomy)

생물학에서 종, 속, 과, 목, 강, 문계의 분류학을 이용하여 의사소통의 정확성을 기하고, 다양한 동식물계의 조직과 상호관계를 이해하는 수단으로 사용하고 있는 사실은 잘 알려져 있다. 마찬가지로, 교육학에서는 1956년에 블룸(Bloom, B.S.)이 주축이 되어 학교의 교육 목표를 분류하여 교육과정이나 평가문제를 다루는 모든 교사, 행정가, 연구가들이 교육문제를 논의할 때 보다 분명한 개념을 가지고 논의할 수 있도록 도움이 되게 하기 위하여 교육분류학을 제안하고 있다[2]. 블룸의 분류체계는 이원분류학으로서 지적 차원과 인지과정 차원으로 이루어져 있다[1]. 지적 차원은 지식의 일반 유형을 사실적 지식, 개념적 지식, 절차적 지식, 메타인지 지식 네 가지 유형으로 나누었다. 사실적 지식(Factual Knowledge)은 별개의 분리된 내용 요소인 "정보단위"에 대한 지식이다. 개념적 지식(Conceptual Knowledge)은 다소 복잡하고 조직화된 지식 형식을 말한다. 절차적 지식(Procedural Knowledge)은 어떤 것을 수행하는 방법에 대한 지식이다. 그리고 메타인지 지식(Meta-cognitive Knowledge)은 지식의 인지에 대한 인식 및 지식과 인지 전반에 대한 지식을 말한다.

한편, 인지과정 차원에는 기억하기, 이해하기, 적용하기, 분석하기, 평가하기, 그리고 창안하기의 여섯 개 유형이 있다. 기억하기(Remembering)는 장기기억에서 관련된 정보를 인출하는 것을 뜻한다. 이해하기(Understanding)는 말이나 글, 그래픽 등을 통해 전달된 교육 메시지로부터 의미를 구성하는 것을 지칭한다. 적용하기(Applying)는 특정 상황에서 절차를 실행하거나 활용하는 것을 말한다. 분석하기(Analyzing)는 자료를 구성요소로 나누고 구성요소 상호 간의 관계와 구성요소와 전체 구조 혹은 의도와 어떻게 관련되어 있는가를 결정하는 것을 의미한다. 평가하기(Evaluating)는 준거와 기준에 근거하여 판단하는 것을 뜻한다. 마지막으로, 창안하기(Creating)는 부분 요소들을 결합해서 새롭고 일관성이 있는 전체로 결합하거나 독창적인 산물을 만들어 내는 것을 뜻한다.

불룸의 분류체계를 이용하면, 다양한 관점에서 교육

목표 및 학습의 가능성을 고려할 수 있으며, 그 목표와 학습을 평가하는 방법 사이의 일관성을 보다 분명하게 드러나게 할 수 있다는 장점이 있다.

### 2.2 선행 연구

스크래치 수업을 통한 컴퓨팅 사고(CT) 향상을 평가한 선행 연구로 다음과 같은 국내외 사례를 발견할 수 있었다.

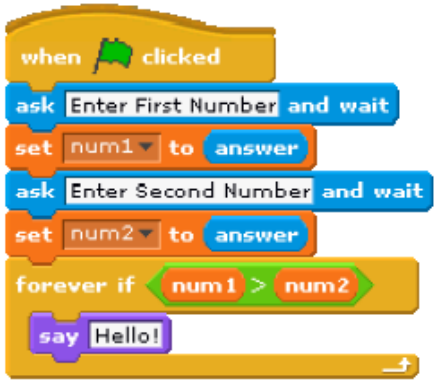
김수환과 한선관은 초등학교 4학년 20명을 대상으로 5일간 스크래치 수업을 실시 후, CT의 컴퓨터적 관점 영역의 인식이 제대로 형성되었는지 알아보기 위해 문항지를 기반으로 하는 자기 설문 평가를 실시하였다[7]. 표현하기('나는 스크래치를 통해 새로운 창작물을 만들 수 있었나'), 연결하기('스크래치 작품을 만들 때 친구들과 협력하고 교류 하였는가'), 질문하기('스크래치로 현실의 문제를 풀기 위해 다양한 생각과 의문을 가졌는가')의 세 항목에 대한 학습자들의 인식을 5개 문항으로 설문 구성 평가하였다.

송정범 외는 초등학교 6학년 재량활동 시간에 스크래치 프로그래밍 학습을 실시하고, 학습자의 내재적 동기 유발과 문제 해결력 향상에 효과가 있는지를 작업 선호도 검사도구(The Work Preference Inventory: WPI. Amabile, 1994)와 OECD의 학업 성취도 국제비교 조사(PISA, 2003)의 12개 문제 해결 유형 문항을 추출, 선발하여 실시하였다[9].

이은경은 스크래치를 활용한 창의적 프로그래밍 학습을 설계하여 중학교 1학년 학생들 34명의 창의성 잠재력 및 CT 능력 발달에 미치는 영향을 창의성 학습 평가 연구소(http://www.clec.co.kr)에서 개발한 검사 도구를 사용하여 평가하였다[8]. 총 51문항으로 가능한 점수 분포는 51점에서 최고 255점까지이다.

최형신은 교육대학교 학생들의 스크래치 수업 학기 프로젝트 산출물에서 다양한 측면의 CT 능력의 세부 요소를 보이는 증거들을 기초, 발달, 능숙의 3단계의 수준으로 평가할 수 있는 루브릭을 고안 제안하였다[4]. 평가된 6가지 CT 세부 요소들은 절차 및 알고리즘, 병행화 및 동기화, 자료표현, 추상화, 문제 분해, 그리고 시뮬레이션이다.

그 외에도, 이스라엘 정부는 중학교 1학년 교육 과정에 CT를 강화시키기 위해 새로이 도입된 커리큘럼인 '스크래치를 이용한 컴퓨터 과학 개론' 과목을 평가하기 위해 2012년에 5,194명의 학생들을 대상으로 전국적인 스크래치 시험을 실시했다[13]. 시험 문제는 4가지 주제인, 순차적인 실행(8%), 조건(30%), 반복(12%), 그리고 루프(50%)에 대해서, 블룸의 텍사노미를 적용하여, 기억하기와 이해하기(Remembering & Understanding)(50%), 적용하기와 분석하기(Applying & Analyzing)(20%), 그리고 평가하기와 창안하기(Evaluating & Creating)(30%)를 테스트한 총 9 문제로 구성되어 있다. (Fig. 1a, 1b)는 무한 반복문에 대해서 블룸의 텍사노미에 기반을 두어서 테스트한 문제 예를 보여준다(답은 진하게 표시되었음).




(Fig. 1a) 스크립트

(Fig. 1a)의 스크립트에 대해서 아래 질문에 답하세요.

A. 다음 보기 중 "Hello"라고 말하기를 실행시키는 입력쌍은?

- ☐ num1 = 5, num2 = 8
- ☐ num1 = 3, num2 = 9
- ☐ num1 = 8, num2 = 8
- ☐ **num1 = 7, num2 = 6**

B. (Fig. 1b)의 두 스크립트는 동일한가요?

- ☐ 아니오. 스크립트 A는 "hello" 말하고, 스크립트 B는 아무런 말도 안한다.
- ☐ **아니오. 스크립트 B는 "hello" 말하고, 스크립트 A는 아무런 말도 안 한다.**

□ 예. 스크립트 A, B 모두 “hello” 말한다.
□ 예. 스크립트 A, B 모두 아무런 말도 안 한다.


(Fig. 1b) 스크립트 A(왼쪽), B(오른쪽)

A번 문제는 스크립트를 제대로 따라 갈 수 있는가를 물어보는 문제이므로, 기억하기와 이해하기(R&U) 수준을 테스트하는 문제였다. B번 문제는 학생들에게 두 스크립트를 비교하는 것을 묻는 문제로서, 적용하기와 분석하기(A&A) 수준을 테스트하는 문제였다.

## 3. 연구 방법 및 절차

### 3.1. 평가 및 모니터링 도구

#### (1) 컴퓨팅 사고 능력 평가 설문지

학생들의 컴퓨팅 사고 능력을 측정하기 위해 14개의 문항을 개발하였다. 개발된 문항은 컴퓨팅 사고 프레임워크[3]의 세 가지 차원에 기반을 두어 컴퓨팅 사고 개념 7문항, 수행 4문항, 관점 3문항을 개발하였다. 설문 문항은 4점 척도(전혀 그렇지 않다, 약간 그렇지 않다, 약간 그렇다, 매우 그렇다)로 구성되었다. 이 프레임워크는 스크래치 창시자가 학생들의 온라인 스크래치 프로젝트 결과물을 기반으로 컴퓨팅 사고능력을 평가 하는데 사용된 도구이다. 이 프레임워크를 바탕으로 아래와 같은 문항을 개발하였다(<Table 1> 참조).

<Table 1> CT Competencies Survey

| Dimension | Construct | Items |
|---|---|---|
| Computational Concepts | Sequence | I can program for sprites to move in the order I created. |
| | Loops | I can program for sprites to do the same movement repeatedly. |
| | Parallelism | I can program for sprites to move at the same time. |
| | Events | I can program for a sprite to start moving when it is clicked. |
| | Conditional | I can program for a sprite to start moving only when the conditions are met. |
| | Operators | I can program for sprites to move by making simple formulas. |
| | Variables /Lists | I can save required values for later use in my programs. |
| Computational Practices | Incremental/ Iterative | I can try making programs as I wanted and improve it gradually. |
| | Testing/ Debugging | I can detect if sprites move in a wrong way and can fix the problem. |
| | Reusing/ Remixing | I can reuse my peers’ program to try something new and better. |
| | Abstracting/ Modularizing | I can program by dividing a big and difficult problem into smaller and easy problems. |
| Computational Perspectives | Expressing | I can use Scratch to create what I want to express. |
| | Connecting | I can handle more difficult and complex programming if I collaborate with my peers. |
| | Questioning | I can ask questions and find answers about why Scratch programs work this way. |

#### (2) 스크래치 지필 시험

스크래치 지필 시험은 블룸의 디지털 텍사노미의 6수준에서 2수준씩 묶어 ‘기억하기와 이해하기’(Remembering & Understanding)(43%), ‘적용하기와 분석하기’(Applying & Analyzing)(24%), 그리고 ‘평가하기와 창안하기’(Evaluating & Creating)(33%)를 테스트한 총 14문항으로 구성되었다. 예를 들어 ‘기억하기와 이해하기’ 문항으로는 ‘방송하고 기다리기’ 코드를 제시하고 정확히 무엇을 기다리는 것인지를 답하도록 하였다. ‘적용하기와 분석하기’ 문항의 예로는 반복적 구조를 가진 코드 블록을 제시하고 반복의 종료 후에 최종적으로 어떤 결과가 나타나는지를 분석하여 제시하도록 하였다. 끝으로 ‘평가하기와 창안하기’ 영역에서는 초기 화면과 최종 화면을 제시하고 주어진 조건을 만족하면서 최종 화면의 결과를 나타내기 위해 주어진 코드블록으로 프로그램을 작성하도록 하였다. 이때 필요한 코드블록은 제

시해 주어 선택하도록 함으로써 코드블록을 기억하지 못해서 작성하지 못하는 경우를 배제하였다.

#### (3) 산출물 기반 인터뷰

인터뷰 질문은 학생들이 지금까지 받아온 컴퓨터 관련 교육 및 프로그래밍 능력에 대한 백그라운드 정보, 개선점 및 다음 학기 프로그래밍 수업에 바라는 점에 대한 피드백으로 구성되었다. 산출물 기반 인터뷰 질문은 팀프로젝트 내용을 살펴보면서 CT에 대한 3가지 프레임워크(개념, 수행, 관점)에 관련된 문항으로 선행 연구에서 사용된 CT 프레임워크 기반 인터뷰 프로토콜을 활용하였다[5].

### 3.2. 연구 대상 및 연구 절차

본 연구는 2014년 2학기에 예비교사 90명을 대상으로 진행되었다. 스크래치 시험은 90명이 참여하였고, 컴퓨팅 사고 설문에는 82명이 참여하였다. 인터뷰는 스크래치 수업을 한 학기 동안 수강한 예비교사 중 자발적인 지원을 받아 그중 3명의 여학생들을 대상으로 실시되었다. 연구에 참여한 2명의 교수가 인터뷰를 진행하였으며 2014년 12월 15일과 16일 이틀 동안 학교에서 실시되었다. 인터뷰 시간은 총 4시간 정도 소요되었다. 전체 인터뷰 내용은 오디오 녹음되었으며, 주요 응답은 연구자에 의해 종이와 펜을 사용하여 인터뷰 시 기록되었다.

## 4. 연구 결과

### 4.1 예비교사 컴퓨팅 사고 평가 결과

한 학기 동안 스크래치 프로그래밍 수업을 진행한 뒤 학생들에게 자가 평가 설문을 실시한 결과는 <Table 2>와 같이 나타났다.

컴퓨팅 사고 프레임워크의 영역별로 보면, 컴퓨팅 사고 개념 영역 M=3.42, 수행 영역 M=3.10, 관점 영역 M=3.13으로 나타났으며, 전체 역량은 M=3.21(최고점은 4.0점)로 나타났다. 세부적으로 살펴보면 개념에서 이벤트(M=3.69)에서 가장 높게 나타났고, 반복 프로세싱 개념(M=3.04)이 그다음 순으로 나타났으며, 변수(M=3.06)와 연산(M=3.26)이 상대적으로 어려운 개념으로 나타났다. 수행 영역에서는 단계적 수행(M=3.39)이 가장 높게 나타난 반면 추상 및 모듈화(M=2.96)가 가장 낮게 나타났다. 관점 영역에서는 연결하기(M=3.27)가 가장 높게 나타나고 질문하기(M=3.02)가 상대적으로 낮게 나타났다. 이러한 결과는 컴퓨팅 사고 개념이 컴퓨팅 사고 수행과 관점영역에 비해 보다 단편적이고 구체적이기 때문이라고 할 수 있다. 또한 수행 영역에서 추상화 및 모듈화의 수행이 단계적 수행, 테스팅, 재사용보다 고차원적이기 때문으로 해석할 수 있다.

### 4.2 스크래치 지필 시험 결과

시험 문항들을 블룸의 텍사노미의 6수준을 3개로 묶어 기억하기와 이해하기, 적용하기와 분석하기, 평가하기와 창안하기로 유목화하여 시험 결과를 분석하였다.

<Table 2> Students CT Competencies(n=82)

| CT | Construct | Mean | CT | Construct | Mean |
|---|---|---|---|---|---|
| Computational Concepts | Sequence | 3.35 | Computational Practices | Incremental/Iterative | 3.39 |
| | Loops | 3.63 | | Testing/Debugging | 3.01 |
| | Parallelism | 3.56 | | Reusing/Remixing | 3.02 |
| | Events | 3.69 | | Abstracting/Modularizing | 2.96 |
| | Conditional | 3.37 | | **Sub Total** | **3.10** |
| | Operators | 3.26 | Computational Perspectives | Expressing | 3.10 |
| | Variables/Lists | 3.06 | | Connecting | 3.27 |
| | **Sub Total** | **3.42** | | Questioning | 3.02 |
| | | | | **Sub Total** | **3.13** |
| | | | | **Total** | **3.21** |

전체 시험의 평균은 53.33점(100점 만점)이었으며, 기억하기와 이해하기는 52.22점, 적용하기와 분석하기는 71점, 평가하기와 창안하기는 41점으로 나타났다. 상대적으로 하위단계인 기억하기와 이해하기가 적용하기와 분석하기보다 낮은 점수를 보인 것은 문항 중에 참 또는 거짓으로 답하는 문항에 정확하게 답하지 못한 것으로 보인다. 텍사노미에서 상위 단계인 평가하기와 창안하기가 가장 낮은 점수를 보였는데 이는 코드 블록이 병행되어 동시에 처리될 수 있는 점을 활용하지 못하고 하나의 루프에서 처리하려고 해서 어려움을 겪는 것으로 파악되었다. 개념적으로는 병렬 수행의 의미를 이해하고 있지만 많은 예비교사가 구체적인 맥락에서 병렬 수행을 효과적으로 활용하는 데 취약함을 보여주는 결과였다.

### 4.3 산출물 기반 인터뷰

한 학기 동안 스크래치 프로그래밍 수업을 들은 학생들을 상대로, 보다 구체적인 코딩 경험을 파악하기 위해, 팀 프로젝트 산출물에 기반을 둔 심층 면담을 실시했다. 학생들의 실제 프로젝트 경험(Authentic Project Experience)을 바탕으로 진행되는 산출물 기반 인터뷰(Artifact-based Interview)는 관련된 경험과 정성적 요소들을 보다 구체적으로 이끌어낼 수 있다는 장점을 가지고 있다[3].

인터뷰한 세 명의 학생들 중 한 명을 제외한 나머지 학생들은 수업을 수강하기 전까지 스크래치 및 다른 프로그래밍 언어에 대한 경험이 전혀 없었다. 경험이 있다고 응답한 한 명도 초등학교 때 HTML를 사용하여 홈페이지 만드는 것과 고등학교 때 간단하게 스크래치에 대한 소개를 받았을 뿐이었다. 학생들은 컴퓨터 수업에 대해서 기계를 다루어야 한다는 점에서 두려움과 어렵다는 선입관이 있었다. 초・중・고등학교나 학원에서 컴퓨터를 배운 또 다른 학생도 있었지만, 정보 검색하는 방법 그리고 워드프로세서로 문서 작업하는 수준으로, 실제 프로그래밍하는 법은 배울 기회가 없었다고 하였다.

학생들은 스크래치 수업을 통해서 프로그래밍을 재미있게 할 수 있겠다는 생각을 가지게 되었다고 대답하였다. 특히 직접 만들어 보고, 플레이해보는 부분에서 재미를 느꼈다고 이야기했다. 인터뷰 학생들의 프로젝트 작품들은 5학년 과학 과목 중 작은 동물 생태계에 대한 소단원과 음악 과목 중 여러 가지 악기로 소리내기 소단원을 스크래치로 구현한 프로젝트였다. 스크래치를 쉽게 적용할 수 있는 실기활동 분야라는 생각이 가장 먼저 떠올라서 그런 주제를 선택하게 되었다고 대답하였다. 프로젝트는 한 학기 동안의 팀 프로젝트로서 2명의 학생이 한 팀을 이루어 진행되었다. 학생들의 협업 형태는 초기에는 같이 모여서 주제 설정과 게임 아이디어 생성을 진행 하였으며, 코딩 작업의 경우는 처음에는 컴퓨터 한대에서 같이 하다가, 나중에는 교육문제를 푸는 게임 부분을 두 파트로 나누어서 분업을 하고 최종적으로 합치는 방식을 취한 경우도 있었고, 처음부터 끝까지 한 컴퓨터에서 같이 프로그래밍 한 경우도 있었다.

학생들은 수업시간에 배운 내용을 확장 변형해서 프로젝트를 개발하게 된 점이 도움이 많이 되었다고 하였다. 처음에는 기대했던 대로 프로그램이 돌아가지 않아 어려움도 있었지만, 시간을 투자해 시행착오를 겪으면서 마침내 제대로 돌아가게 만들었을 때 보람을 느꼈다고 하였다. 한 예로 배경 전환시 소리가 겹쳐서 나오는 어려움이 있었는데, '배경'이라는 변수를 추가함으로서 방송을 받았을 때, 변수 값을 확인을 해서 소리를 내주는 방법을 써 문제를 해결할 수 있었을 때 보람을 느꼈다고 하였다. 또 다른 예는 겹쳐서 방송하게 되어, 의도하지 않게 실행이 동시에 되는 경우가 있어서, 스크립트 멈추기라는 블록을 사용하게 되었다고 하였다. 문제가 발생했을 때 팀원들끼리 같이 의논 하면서 해결해 나갈 수 있어 혼자 했을 때 보다는 많은 도움이 되었다고 대답했다.

스크래치 프로그래밍 개념 중 '방송하기' 개념과 '변수' 개념은 수업과 시험에서는 잘 이해가 되지 않았다가 프로젝트를 함으로써 확실히 알게 되었다고도 언급했다. 또 다른 예는 '방송하기' 와 '방송하고 기다리기' 블록의 차이를 직접 프로젝트를 함으로써 이해가 쉽게 되었다고 하였다.

지필 시험에서 프로그래밍 문제를 풀 경우하고는 달리, 프로젝트 과제는 한 번에 끝나는 게 아니라, 시간을 가지고 테스트하고 디버깅하는 과정을 반복해서 원하는 바를 완성하게 되니까, 문제가 생겼을 때는 코드를 보

면서 사고를 많이 하게 되었다고 했다. 디버깅 과정이 쉽지는 않았지만, 마침내 문제를 해결해 낼 때는 기분이 아주 좋았다고 이야기했다.

이외에도 수업시간에 배운 내용을 리믹스, 재사용 하였으며, 스프라이트별로 나누어서 모듈화 하는 방법을 익혔다고 했다. 수업시간에 완전히 이해하지 않아도 참고해서 그냥 따라 하는 것과 프로젝트를 통해 내가 막상 직접 처음부터 만드는 것은 큰 차이가 있었다고 느끼고 있었다. 따라서 과제를 좀 더 일찍 학생들이 시작하도록 하면 좋았을 것 같다는 의견도 제시하였다. 하지만 그런 경우, 기능들을 다 알지 못하고 제한된 범위 내에서 프로젝트를 진행해야 하는 제약이 생길 수도 있다고 반문하자, 기능을 빨리 가르쳐주고 과제를 하게 함으로써 숙달 시키는 방향이 좋을 것 같다는 의견도 제시하였다.

전반적으로, 스크래치 과제를 하는 과정에서 CT 개념(순차적 실행, 병렬 실행, 조건부 실행 등)이 자연스럽게 습득이 되었다는 것을 인터뷰를 통해 발견할 수 있었다. 한 학생은 스크래치를 통해 순서도 같은 개념에 익숙해져서, 문제 해결 시 절차를 생각해 보게 되었고, 틀렸을 때 문제해결 하는 방법에 대해서 생각을 하게 되었다고 했다.

또 다른 언급으로는 초등학생들이 게임이나 캐릭터 만든 것에 흥미를 가지니까 스크래치를 활용해서 표현능력을 키울 수 있다고 생각한다는 의견도 있었다. 특히 스크래치를 배우는 것 자체가 그렇게 어렵지 않아서 활용도가 높겠다는 생각이 든다고 하였다. 예비교사로서 선생님이 반 전체를 운영할 때 쓸 수 있는 프로젝트에 특히 관심이 있고, 많이 활용될 것 같다는 의견도 제시하였다.

## 5. 결론 및 제언

본 연구를 통해 향후 예비교사 소프트웨어 교육 및 평가 방법에 대해 도출한 시사점을 다음의 몇 가지로 정리할 수 있다. 첫째, 블룸의 텍사노미 체제로 프로그래밍 역량을 구분해서 분석한 결과 상위 수준인 창안하기와 평가하기 영역에서 예비교사들의 능력이 취약함이 드러났다. 향후 코딩 수업 설계 시 목표 설정과 평가에서 보다 일관성을 분명하게 할 필요를 보여준다고 할 수 있다. 둘째, 자기 보고식 CT 설문에서 비교적 높은 능력을 보고하고 있지만 실제 코딩 맥락에서 각 개념을 유창하게 구현하는 능력과는 차이를 보이고 있다. 이는 자기 보고식 CT 역량 평가의 한계를 드러낸 것이라고 할 수 있다. 향후 CT 역량 평가와 프로그래밍 역량 평가를 통합한 총체적 평가 방법이 마련될 필요가 있음을 시사한다고 할 수 있다. 끝으로 산출물 기반 인터뷰에 의해 드러났듯이 팀프로젝트의 경험은 학생들에게 CT 수행 능력 배양에 중요한 영향을 미치는 것으로 나타났다. 특히 지필 시험에서는 발휘할 수 없었던 능력도 일정 기간 동안 테스트하고 디버깅하는 과정에서 문제해결력을 향상시키며 리믹싱과 재사용하는 방법과 팀프로젝트를 진행하며 모듈화 능력을 습득할 수 있는 것으로 나타났다. 향후 산출물을 통해 CT 역량을 평가할 수 있는 지시자(indicators)의 개발도 필요함을 시사한다. 본 연구를 토대로 향후 블룸의 인지적 수준에 기반한 일관적인 프로그래밍 역량 평가 방법과 일반적 문제해결에 연계되는 컴퓨팅 사고 평가 시에 자기보고식 평가의 한계를 보완할 수 있는 평가 방법에 대한 보다 심층적 연구가 지속적으로 이루어져야 할 것이다.

## 참고문헌

[1] Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., Raths, J., & Wittrock, M. C. (2005). A Taxonomy for Learning, Teaching, and Assessment: A Revision of Blooms' s Taxonomy of Education Objectives. Pearson Education Korea Ltd, and Academy Press Publishing Co.

[2] Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). Taxonomy of educational objectives: the classification of educational goals. Handbook I: Cognitive domain, New York: David McKay Company.

[3] Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the annual

American Educational Research Association (AERA) meeting, Vancouver, BC, Canada.

[4] Choi, H. (2014). Developing Lessons and Rubrics to Promote Computational Thinking. *Journal of the Korean Association of Information Education, 18*(1), 57-64.

[5] Choi, H. (2014). Computational Thinking Framework-based Analysis of Afterschool Scratch Team Project Experiences. *Journal of the Korean Association of Information Education, 18*(4), 549-558.

[6] Ettimes (2015.5.17). Cornerstone for cultivating creative talents, software education.

[7] Kim, S., & Han, S. (2012). Design-Based Learning for Computational Thinking. *Journal of the Korean Association of Information Education, 16*(3), 319-326.

[8] Lee, E. (2013). Creative Programming Learning with Scratch for Enhancing Computational Thinking. *Journal of Korean Association of Computer Education, 16*(1), 1-9.

[9] Song, J., Cho, S., & Lee, T. (2008). The Effect of Learning Scratch Programming on Students' Motivation and Problem Solving Ability. *Journal of the Korean Association of Information Education, 12*(3), 323-332.

[10] Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 19*(3), 33-35.

[11] Wing, J. M. (2011). Computational thinking－What and Why?. CMU Research Notebook. Retrieved from http://link.cs.cmu.edu/article.php?a=600

[12] Yonhap News (2014.7.22). Mandatory SW education starting from the first-year junior high school students next year.

[13] Zur-Bargury, I., Parv, B., & Lanzberg, D. (2013). A National Exam as a Tool for Improving a New Curriculum. Proceedings of ITiCSE Conference, 267-272.

## 저자소개

**최 형 신**

1988 이화여자대학교(전자계산학 학사)

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

2007 이화여자대학교(교육공학 박사)

2009～현재 춘천교육대학교 컴퓨터교육과 교수

관심분야: 컴퓨팅 사고, 뉴미디어 기반 학습

e-mail: hschoi@cnue.ac.kr

**김 기 범**

1998 고려대학교(컴퓨터학 학사)

2000 (미)University of Illinois at Urbana-Champaign(컴퓨터학 석사)

2007 (미)Virginia Tech(컴퓨터학 박사)

2014～현재 춘천교육대학교 특성화사업단 연구교수

관심분야: 인간과 컴퓨터 상호작용, 컴퓨터교육

e-mail: kibumkim@cnue.ac.kr