

스크래치 프로젝트 분석을 통한 컴퓨팅사고력 수준 분석

박선주

광주교육대학교 컴퓨터교육과

요 약

2015 개정교육과정에 의해 2018년부터 SW 교육이 필수화되면서 SW 교육에 대한 연구가 활발하게 이루어지고 있다. 본 논문에서는 예비교사들의 컴퓨팅 사고력 수준을 파악하기 위해 2016년부터 2018년까지 3년 동안 K 교육대학교 1학년 학생 325명을 대상으로 스크래치 수업을 실시한 후 연도별로 산출된 스크래치 프로젝트를 Dr. Scratch 분석도구로 평가하여 연도별과 학점별로 CT요소 점수의 연관성을 분석하였다. 그 결과, 연도와 학점이 높을수록 CT요소, 점수, 등급도 높아짐을 알 수 있었다. 이를 통해, 학생들은 Dr. Scratch와 같은 자동화된 평가 도구를 사용하여 평가해 본 후 스스로 수정 보완해보는 과정과 필요한 자료를 직접 제작하여 활용하는 ICT 능력이 필요하며, 복잡한 조건과 논리연산 활용 등 논리적 사고 요소를 이해하고 잘 활용될 수 있도록 교육될 필요가 있다.

키워드 : 스크래치, Dr. Scratch, 컴퓨팅 사고, SW교육, EPL

Analysis of Computational Thinking Level Through the Scratch Project Analyzation

SunJu Park

Dept. of Computer Science Education, Gwangju National University of Education

ABSTRACT

As SW education has become essential since 2018 due to the revised curriculum in 2015, researches on SW education are actively being carried out. In order to understand the level of pre-service teachers' computational thinking level, we analyzed a correlation of CT element scores with each year and each grade based on the calculated Scratch project by years using the analysis tool Dr. Scratch, which was targeted for 325 students in K National University of Education who passed the scratch course from 2016 to 2018. The result indicated that there is a positive correlation between all the CT related factors and both the year and the grade. Conclusionally, it is crucial to have students undergo revising process by using an automated evaluation tool such as Dr. Scratch and cultivate ability to create and utilize required materials. Furthermore, it is necessary to educate students to utilize logical thinking elements such as complex conditions and logic operations.

Keywords : Scratch, Dr. Scratch, Computational Thingking, SW education, EPL

논문투고 : 2018-12-17

논문심사 : 2018-12-21

심사완료 : 2018-12-21

1. 서론

2015 개정 교육과정에서 SW 교육 중요성이 증대되어 초등학생들도 쉽게 접할 수 있도록 언플러그드 컴퓨팅, 블록형 프로그래밍 언어, 피지컬 컴퓨팅 영역에서 다양한 교육적 방법과 도구들을 활용한 SW 교육이 이루어지고 있으며, SW 교육 평가방법 및 평가도구도 개발되고 있다[5,7].

SW 교육후 산출된 결과물에 대한 분석 및 평가를 통해 SW 교육 피드백이 필요하며, 특히 효율적인 코딩을 위해 프로그래밍 소스를 분석할 필요가 있다. 이와 같이 EPL 교육에서 학습자들은 자신의 프로그램을 테스트하고 디버깅하는 과정을 필수적으로 경험하게 되는데 교사나 학생들이 자신의 스크래치 프로젝트를 파악하고 오류를 발견할 수 있는 자동화된 도구가 필요하다. 스크래치는 명령어 블록이 8개의 종류로 분류되어 있어 각 블록들의 사용횟수로 코드 분석이 가능하며, 분석도구로는 Scrape, Dr. Scratch 등이 있다[13,14,16].

또한, 컴퓨팅 사고력은 여러 요소를 포함하고 있어서 단순히 지필평가와 같은 방법으로 평가하기가 어렵다. MIT 미디어랩에서도 학습자의 컴퓨팅 사고력 개념을 평가하기 위한 방법으로 프로젝트 분석법을 제안하고 있다[9]. 프로젝트 분석법은 컴퓨팅 사고력의 개념을 평가하는데 활용할 수 있으므로[1], 학습자들의 스크래치 프로젝트 분석을 통해 학습자들의 컴퓨팅 사고력 수준을 파악할 수 있다.

그러므로 본 논문에서는 예비교사들의 컴퓨팅 사고력 수준을 파악하기 위해 교양필수과정인 ‘정보사회와 컴퓨터’ 과목에서 연도별로 산출된 스크래치 프로젝트를 Dr. Scratch 분석도구로 평가하여 학생들의 CT요소별 점수와 등급을 확인하여, 학생들에게 부족한 CT요소는 무엇이고 선호하는 블록은 무엇인지 파악하고자 한다. 또한, 담당교수가 부여한 학점과 Dr. Scratch의 평가결과와의 차이 여부 등을 통해 평가자와 평가도구의 신뢰성을 파악하고자 한다. 이를 위해 2016년부터 2018년까지 3년동안 K 교육대학교 1학년 학생 325명을 대상으로 스크래치 수업을 실시한 후 제출한 결과물을 연도별, 학점별로 CT요소 점수에 차이가 있는지 그 연관성을 분석하고자 한다.

2. 스크래치 코드 분석

교육용프로그래밍 언어의 코드를 분석할 수 있는 도구에는 Scrape, Dr. Scratch 등이 있다[13,14,16].

Scrape는 스크래치 코드 분석툴로 각 블록의 횟수와 저장횟수, 스프라이트 수, 블록수, 스택수 등의 분석이 가능하다.

Dr. Scratch도 스크래치 코드 분석툴이며 Scrape의 기능을 발전시켜 각 블록의 횟수를 분류하여 데이터 표현, 논리적 사고, 사용자 상호작용, 플로우 제어, 추상화, 병렬화, 동기화의 7개 CT요소를 사용하는데, 그 정의는 Brennan & Resnick[1]의 연구와 유사하다. 또한 각 CT요소들의 점수를 합산하여 0~7점은 기본단계, 8~14점이면 개발자단계, 15~21점이면 마스터단계로 등급을 제시하고, 코드 복사, 스프라이트 이름변경, 죽은 코드, 스프라이트 속성을 보여준다[13,14]. Dr. Scratch의 7개 CT요소별 점수를 부여하는 블록의 기준은 표 1과 같다.

Martin & Fields[10]이 개발한 도구도 스크래치를 분석하는 툴로 Scrape 와 유사한 기능을 가지고 있으며 스프라이트들 간의 방송 주고받기에 대한 흐름을 한눈에 볼 수 있는 기능을 제시하고 있다.

<Table 1> Dr. Scratch's Score Assignment

CT dimension	Competence Level		
	Basic (1point)	Medium (2points)	Proficient (3points)
Flow control	Sequence of blocks	Repeat, forever	Repeat until
Data representation	Modifiers of object properties	Variables	Lists
Abstraction and problem decomposition	More than one script	Use of custom blocks	Use of 'clones' (instances of sprites)
User interactivity	Green flag	Keyboard, mouse, ask and wait	Webcam, input sound
Synchronization	Wait	Message broadcast, stop all, stop program	Wait until, when backdrop changes, broadcast and wait
Parallelism	Two scripts on green flag	Two scripts on key pressed or sprite clicked	Two scripts on receive message, video/audio input, backdrop change
Logical thinking	If	If else	Logic operations

Dr. Scratch를 활용한 국내의 CT 평가연구로 김수환은 컴퓨터 비전공 대학생 45명을 대상으로 스크래치 프로젝트를 분석한 결과, 학습자 수준이 높을수록 평가요소의 점수가 높게 나타나 Dr. Scratch의 각 항목이 학생들의 CT 수준을 평가하는 요소로 활용될 수 있음을 보고하였다[13,14]. 최형신, 김미송은 52명의 예비교원을 대상으로 3가지 CT 평가방법(베브라스 CT 검사, Dr. Scratch 자동평가, CT 역량평가 지필시험)을 실시한 결과 3가지 평가방법의 통합이 긍정적인 효과를 보이는 것으로 나타났다[3,4,11]. 국외 연구로 Moreno-Leon과 Robles는 8개 학교에 재학중인 10-14세 학생 100명을 대상으로 해서, 각자 만든 스크래치 프로젝트를 Dr. Scratch를 이용해서 평가할 수 있고, 평가 결과물뿐만 아니라 관련된 피드백을 이용해서 스스로 스크래치 프로젝트를 향상시킬 수 있음을 보여주었다[6].

알고리즘을 설계한 후 스크래치 프로젝트를 개발하여 제출하였다.

연도별로 제출받은 개인별 프로젝트 결과물을 Dr. Scratch로 평가하여 학생들의 CT요소별 점수와 등급을 확인하여, 학생들에게 부족한 CT요소는 무엇이고 선호하는 블록은 무엇인지 파악할 수 있어 교수자는 강의 피드백을 받을 수 있다. 또한, 담당교수가 부여한 학점과 Dr. Scratch의 평가결과와의 차이 여부 등을 통해 평가자와 평가도구의 신뢰성을 파악할 수 있다.

그러므로 본 논문에서는 연도별로 산출된 스크래치 프로젝트를 Dr. Scratch로 평가하여 연도별로 CT요소 점수와 등급에 차이가 있는가?, 학점별로 CT요소 점수와 등급에 차이가 있는가? 를 분석하여 연도별, 학점별로 CT요소 점수에 차이가 있는지 그 연관성을 분석하고자 한다. 연구절차는 그림 1과 같다.

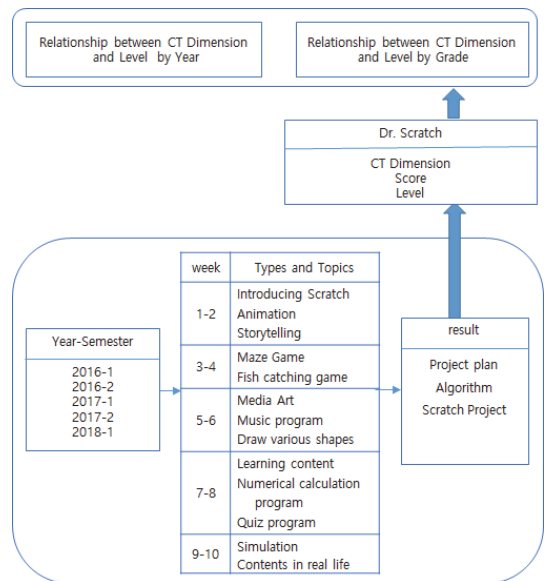
3. 연구방법

3.1 분석대상

예비교사들의 스크래치 프로젝트를 분석한 후 CT 수준을 파악하여 SW 교육 피드백을 받고자, 2016년부터 2018년까지 3년 동안 K 교육대학교 1학년 학생 325명을 대상으로 교양필수 강의에서 제출한 스크래치 프로젝트를 평가하여 분석하였다. 평가도구는 스크래치 코드 분석틀인 Dr. Scratch를 사용하였다.

3.2 연구절차

‘정보사회와 컴퓨터’ 강의는 교양필수과정으로 SW 교육영역인 언플러그드, EPL, 피지컬컴퓨팅을 배우며 EPL은 스크래치를 사용하여 스토리텔링, 게임, 미디어아트, 학습콘텐츠, 시뮬레이션, 실생활중심콘텐츠 등 프로젝트 유형별로 활동주제를 구성하여 10주 동안 실시한 후, 결과물로 프로젝트 기획서, 알고리즘 설계서 및 스크래치 프로젝트 파일을 제출받았다. 학생들은 교육용 스크래치 프로젝트를 개발하기 위하여 프로젝트 주제, 유형, 활용대상, 교육과정 활용방안 등을 구상하고



<Figure 1> Research Procedure

4. 연구결과

4.1 연도별 CT요소 점수와 등급의 연관성 분석

2016년, 2017년, 2018년 연도별로 제출받은 스크래치

프로젝트를 Dr. Scratch로 평가해본 결과, 2018년에는 흐름제어, 동기화, 병렬화, 논리적 사고 요소에서 2016년 2017년에 비해 점수가 점점 높아짐을 알 수 있다.

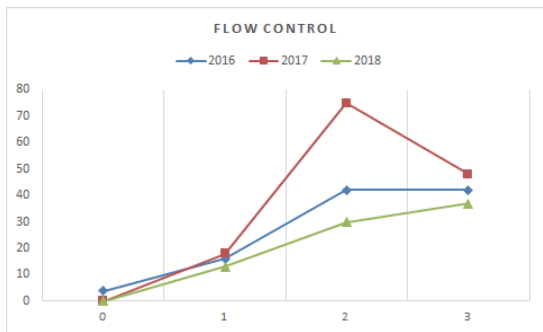
특히 그림 2,4,6,7과 같이 흐름제어, 추상화, 동기화, 병렬화 요소에서 2018년 마스터의 빈도가 기대빈도보다 많이 나온 것을 통해 2018년에 ‘까지 반복하기, 복제, 까지 기다리기, 배경 바꿀때 방송하기 기다리기’, ‘방송주고받기, 조건실행, 배경바꾸기 블록 2개사용’ 등의 블록을 더 많이 사용한 것으로 보인다.

또한, 자료표현 요소는 그림 3과 같이 2016년~2018년 모두 2점의 비율이 많은 것으로 나타났으며, 이는 리스트보다는 변수를 사용한 프로젝트가 60%를 차지하고 있음을 알 수 있다.

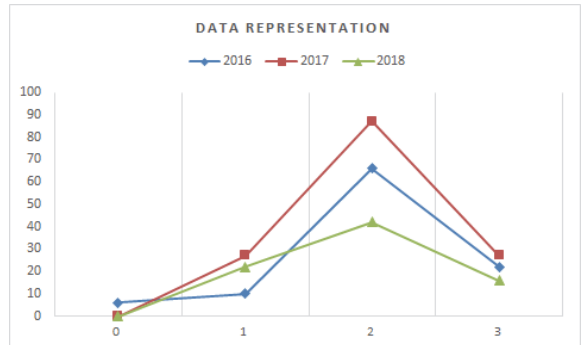
그림 4와 같이 추상화 요소는 1점의 비율이 80%를 차지하고 있고 2점(3.4%), 3점(11.4%)로 사용자 블록과 복제는 많이 사용되지는 않은 것으로 보인다.

그림 5와 같이 사용자 상호작용 요소에서는 2점이 89.8%로 키입력, 마우스 블록, 묻고 기다리기 등은 어려움 없이 많이 활용되나 웹캠이나 사운드 입력 등은 활용도가 많지 않음을 알 수 있다.

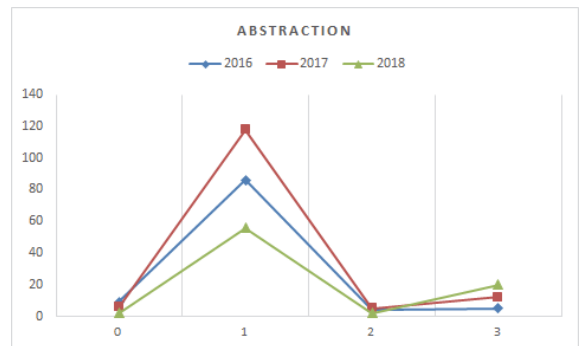
병렬화 요소는 그림 7과 같이 2점이 0.3%, 3점이 65.2%로 키입력 이벤트를 두 개이상 사용하거나 스프라이트 클릭을 2개이상 사용하는 것보다는 방송주고받기, 복제본생성, 배경바꾸기, 조건문 실행하는 것을 더 선호하고 많이 활용한다는 것을 알 수 있다.



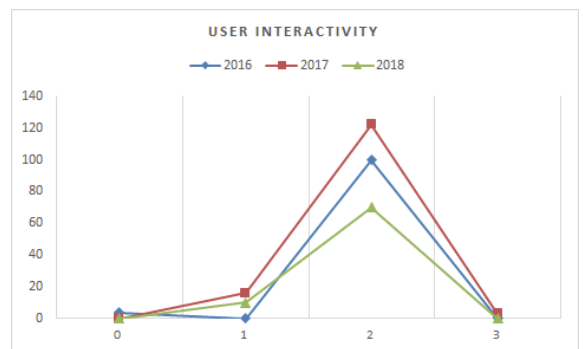
<Figure 2> Result of Flow Control (2016-2018)



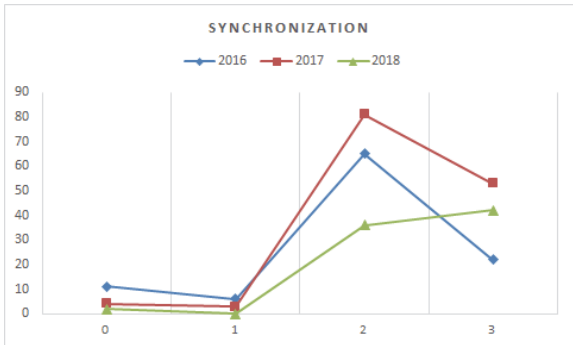
<Figure 3> Result of Data Representation (2016-2018)



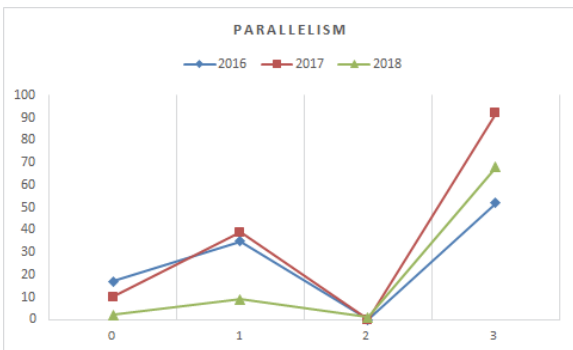
<Figure 4> Result of Abstraction (2016-2018)



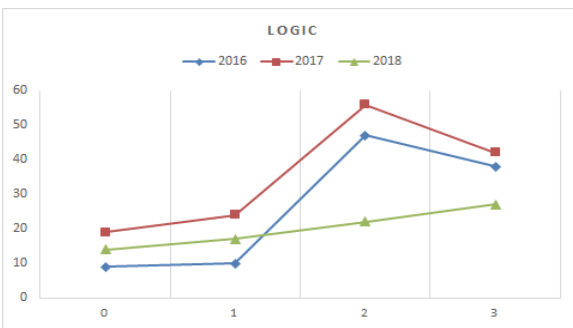
<Figure 5> Result of User Interactivity (2016-2018)



<Figure 6> Result of Synchronization (2016-2018)



<Figure 7> Result of Parallelism (2016-2018)



<Figure 8> Result of Logic (2016-2018)

4.1.1 연도별 CT요소 점수 분석

연도에 따라 CT요소 점수가 달라지는가를 알아보기 위해 연도와 CT요소별로 교차분석과 피셔의 정확도 검정을 이용하여 분석을 실시한 결과, 표 2와 같이 논리적 사고요소를 제외한 6개 CT요소에서 유의확률이 유의수

준 0.05 이내이므로 연도가 CT요소 점수에 영향을 미친다는 것을 알 수 있다.

<Table 2> Result of Chi-square Test of CT Dimension in Each Year

CT Dimension	Pearson chi square value	df	Fisher p-value
Flow control	14.605	6	0.041
Data representation	21.999	6	0.002
Abstraction	23.323	6	0.001
User interactivity.	25.418	6	0.000
Synchronization	29.255	6	0.000
Parallelism	31.531	6	0.000
Logic	11.610	6	0.062

그림2~그림8과 같이 연도가 증가하면서 CT요소 점수가 증가하는지 알아보기 위해 서열상관분석을 실시한 결과, 표 3과 같이 추상화, 동기화, 병렬화 항목의 상관 계수가 유의함을 알 수 있었다. 즉, 연도가 증가함에 따라 추상화, 동기화, 병렬화 블록인 ‘복제, 까지 기다리기, 배경바뀔 때 방송하기 기다리기, 방송주고받기 조건실행 배경바꾸기 블록 2개사용’ 등의 블록을 더 많이 사용한 것으로 보인다.

<Table 3> Result of Spearman’s rank correlation analysis of year and CT Dimension

	Flow control	Data representation	Abstraction	User interactivity	Synchronization	Parallelism	Logic
correlation coefficient	.047	-.067	.215	-.102	.281	.284	-.112
p	.403	.230	.000	.065	.000	.000	.044
N	325	325	325	325	325	325	325

4.1.2 연도별 Dr. Scratch 점수평균 비교

Dr. Scratch의 점수는 7개 CT요소 점수의 합으로 연도별 Dr. Scratch 점수 평균은 2016년(13.00), 2017년(13.67), 2018년(14.54)로 연도가 높아질수록 평균이 상승한 것으로 보인다. 이는 연도가 증가할수록 CT요소

점수가 증가하기 때문이라 할 수 있다.

이러한 연도별로 평균점수의 증가가 유의미한지를 알아보기 위해 연도별 점수의 분산이 동질성 가정을 만족하므로($p=0.212$) 일원배치 분산분석을 실시한 결과 연도가 점수에 영향을 미치는 것을 알 수 있다($F=5.689$, $p=0.004$). 따라서 어떤 연도에서 점수평균이 유의미한 차이가 있는지 알아보기 위하여 LSD, Bonferroni, Duncan 방법을 이용하여 사후검정을 실시한 결과, LSD 방법에서는 2018년의 평균이 2016년, 2017년과 유의미하게 차이가 있었으며, Bonferroni 방법에서는 2018년과 2016년의 평균이 유의하게 차이가 있었다. 표 4와 같이 Duncan 방법에서도 2018년의 평균이 2016년, 2017년의 평균과 차이가 있음을 알 수 있었다.

<Table 4> Result of Homogeneous Subset by Duncan

year	N	Subset for $p=0.05$	
		1	2
2016	104	13.00	
2017	141	13.67	
2018	80		14.54
p		.116	1.000

따라서 Dr. Scratch 점수평균은 2016년, 2017년보다 2018연도의 점수평균(14.54)이 가장 높으며, 다음으로 2017연도의 평균이 13.67, 2016연도의 평균이 13.00 으로 나타나지만 이 둘의 차이는 유의미하지 않음을 알 수 있다.

4.1.3 연도별 Dr. Scratch 등급 비교

Dr. Scratch 점수가 0~7점은 기본단계, 8~14점이면 개발자단계, 15~21점이면 마스터단계로 구분된다.

예비교사들이 제출한 325개 스크래치 프로젝트를 분석한 결과, 개발자 단계(54.77%), 마스터 단계(42.15%)가 96.92%를 차지하고 있었다. 2016년과 2017년에는 개발자 단계의 비율이 높았으나 2018년에는 마스터 단계의 비율이 53.75%로 분석대상의 50% 이상이 마스터 단계의 프로젝트를 개발한 것으로 나타났다. 즉, 2016년과 2017년에 비해 2018년에는 마스터의 비율이 증가함을 알 수 있다. 이는 연도가 증가하면서 CT요소 점수와 점

수평균이 높아졌고, 2018년에 점수평균이 가장 높게 나타났다기 때문에 판단된다.

연도별로 등급과 CT요소들에 차이가 있는가를 알아보기 위해 연도와 등급과 CT요소별로 교차분석과 피셔의 정확도 검정을 이용하여 분석을 실시한 결과, 표 5와 같이 모든 CT요소에서 유의확률이 0.000으로 유의수준 0.05 내에서 귀무가설을 채택하므로 연도와 등급, CT요소는 서로 영향을 미친다고 할 수 있다. 즉, 연도와 등급이 높아질수록 흐름제어, 데이터표현, 추상화, 사용자 상호작용, 동기화, 병렬화, 논리적 사고 요소의 점수도 증가함을 알 수 있다. 2018년 마스터단계의 CT요소 점수들이 가장 높다는 것을 알 수 있으며, 학생들은 연도가 증가할수록 등급이 높아갈수록 모든 블록들을 활용하여 프로그래밍함을 알 수 있다.

<Table 5> Result of chi-square test of levels and years in each CT Dimension

CT Dimension	Pearson chi square value	df	Fisher p-value
Flow control	234.249	6	0.000
Data representation	240.081	6	0.000
Abstraction	155.954	6	0.000
User interactivity.	152.291	6	0.000
Synchronization	154.920	6	0.000
Parallelism	123.182	6	0.000
Logic	166.691	6	0.000

4.2 학점별 CT요소 점수와 등급의 연관성 분석

학점은 담당교수가 스크래치 프로젝트 결과물을 보고 평가기준에 의해 부여한 것으로 자동화된 Dr. Scratch 평가도구의 결과와 어느 정도 일치성을 갖는지 분석하고자 한다.

즉, 학점이 높으면 Dr. Scratch 등급과 CT요소 점수도 높은가를 알아보기 위해 학점을 3그룹으로 분류하였다. A+, A, A-학점을 A그룹으로, B+, B, B-학점을 B그룹으로, C+, C, C-학점을 C그룹으로, D+, F학점은 각각 1명의 학생으로 평균비교를 하는데 무의미하여 결측으로 제외하였다.

4.2.1 학점별 CT요소 점수 비교

학점이 높으면 CT요소 점수가 높은지 알아보기 위해 서열상관분석을 실시한 결과, 표 6과 같이 흐름제어, 추상화, 동기화, 병렬화 항목의 상관계수가 유의함을 알 수 있었으며 모두 양의 상관관계로, 학점이 높을수록 흐름제어, 추상화, 동기화, 병렬화 점수도 높아지는 추세라고 할 수 있다.

<Table 6> Result of Spearman's rank correlation analysis of grade and CT Dimension

	Flow control	Data representation	Abstraction	User interactivity	Synchronization	Parallelism	Logic
correlation coefficient	.206	.093	.124	.084	.139	.287	.059
p	.000	.093	.026	.129	.012	.000	.285
N	325	325	325	325	325	325	325

4.2.2 학점별 Dr. Scratch 점수평균 비교

학점별로 Dr. Scratch 점수 평균은 A(14.67), B(13.72), C(12.77)과 같으며, 학점이 높을수록 평균이 상승한 것으로 보인다.

높은 학점일수록 평균점수가 높아지는 경향이 유의미한지를 알아보기 위해 일원배치 분산분석을 실시한 결과(F=8.719, p=0.000) 학점별 점수의 차이는 의미가 있음을 알 수 있었다(학점별 점수의 분산이 동질성 가정을 만족함(p=0.248)). 따라서 어떤 학점군에서 점수평균이 차이가 있는지 알아보기 위하여 Duncan을 이용한 사후검정을 실시한 결과, 표 7과 같이 A, B, C 그룹 모두 유의하게 차이가 있음을 알 수 있었다. A그룹 평균이 14.67로 가장 높고, 다음으로 B그룹 평균이 13.72로 두 번째로 높고, 마지막으로 C그룹 평균 점수가 12.77로 가장 낮았다. 즉, 담당교수의 평가와 Dr. Scratch 평가결과가 일치함을 알 수 있었다.

<Table 7> Result of homogeneous subset by Duncan

Grade	N	Subset for p=0.05		
		1	2	3
C	62	12.77		
B	172		13.72	
A	88			14.67
p		1.000	1.000	1.000

4.2.3 학점과 Dr. Scratch 등급 비교

학점별 등급비율을 살펴보면 마스터 단계에는 A학점을 받은 학생들이, 개발자 단계에는 B학점 학생들이, 기본단계에는 C학점 받은 학생들의 비율이 높은 것으로 보인다.

또한, 표 8과 같이 기본단계에서는 C학점의 빈도가 기대빈도보다 높게 나타나고 있고, 개발자 단계에서는 C학점과 B학점의 빈도가 기대빈도보다 높게 나타났으며, 마스터단계에서는 A학점의 빈도가 기대빈도보다 높게 나타난 것을 볼 수 있다. 따라서 등급이 높을수록 학점이 높다고 할 수 있다.

<Table 8> Cross-tabulation analysis of grade and level

Level	Grade	A	B	C	total
		Basic	freq.	0	4
Development	Expected freq.	1.9	3.7	1.3	7.0
	freq.	37	105	36	178
Master	Expected Freq.	48.6	95.1	34.3	178.0
	freq.	51	63	23	137
Total	Expected Freq.	37.4	73.2	26.4	137.0
	freq.	88	172	62	144
	Expected Freq.	88.0	172.0	62.0	144.0

등급과 학점이 서로 연관성이 있는가를 알아보기 위해 학점과 등급의 피셔의 정확도 검정 결과, 표 9와 같이 Pearson 카이제곱값 14.626, Fisher의 정확검정 유의확률 0.004로 등급과 학점이 연관성이 있다고 할 수 있다.

<Table 9> Result of chi-square test of grade and level

Pearson chi square value	df	Fisher's exact test p-value
14.626	4	.004

분석결과를 종합해보면, 예비교사들의 스크래치 프로젝트는 하나이상의 스프라이트를 사용하여 마우스와 키 입력 이벤트와 묻고 기다리기 상호작용을 선호하며, 변수 등의 자료표현 요소와 방송주교받기, 복제본 생성 등의 병렬처리도 사용하나 복잡한 조건과 논리연산은 어려워 하고 사용빈도도 적은 것으로 나타났다. 즉, 블록들의 순차, 반복보다는 적절한 조건문 사용을 더 어려워하는 것으로 보인다. 연도가 증가하면서 추상화, 동기화, 병렬화 요소가 증가하며, CT점수 평균과 등급도 증가하였다. 2016년 2017년에 비해 2018년의 CT점수 평균이 가장 높았으며 마스터 단계의 비율도 증가함을 알 수 있었다. 즉, 학생들은 연도가 증가할수록 등급이 높아갈수록 모든 블록들을 활용하여 프로그래밍함을 알 수 있다. 또한, 학점이 높을수록 흐름제어, 추상화, 동기화, 병렬화 요소도 높아지고, CT점수 평균과 등급도 증가하였다. B, C학점에 비해 A학점 그룹의 CT점수 평균이 가장 높았으며 마스터 단계의 비율도 높음을 알 수 있었다

5. 결론

2015 개정교육과정에 의해 2018년부터 SW 교육이 필수화되면서 SW 교육에 대한 연구가 활발하게 이루어지고 있다.

본 논문에서는 예비교사들의 EPL 수업후 산출된 결과물을 Dr. Scratch로 평가하여 연도별, 학점별로 CT요소와의 연관성을 분석하였다. 그 결과, 연도가 증가할수록 CT요소, 점수, 등급도 증가하고, 학점이 높을수록 CT요소, 점수, 등급도 높아짐을 알 수 있었다. 특히 2018년도 마스터 단계의 학생들이 CT요소와 점수가 가장 높았다.

그 원인 중 하나로 연도가 증가할수록 담당교수의 강의평가 결과가 증가하였는데, 표 10과 같이 연도별 강의평가 카이제곱 검정결과, 강의평가 문항별(강의계획서, 강의내용, 강의방법, 과제물, 평가)로 모두 유의함을 알 수 있다. 즉, 연도가 증가할수록 교수자에 대한 강의만족도가 높아졌으며, 이에 따라 학생들의 과제 결과물도 좋아졌음을 알 수 있다.

<Table 10> Result of chi-square test of lecture evaluation by years

Lecture evaluation question	Pearson chi square value	df	p
Lecture plan	25.180	8	.001
lecture content	31.440	8	.000
Lecture method	16.988	8	.030
Assignment	16.057	8	.042
evaluation	24.012	8	.002
total	112.487	8	.000

분석결과를 바탕으로 첫째, 학생들이 자신들이 만든 스크래치 프로젝트를 Dr. Scratch와 같은 자동화된 평가도구를 사용하여 평가해본후 피드백을 받고 스스로 수정 보완해보는 과정도 필요할 것으로 본다. 둘째, 저장소에 있는 스프라이트 활용뿐만 아니라 오디오, 비디오 등 필요한 자료를 직접 제작하여 활용하는 수준까지 확대되도록 ICT 교육도 함께 이루어질 필요가 있다. 셋째, 복잡한 조건과 논리연산 활용 등 논리적 사고 요소를 이해하고 잘 활용될 수 있도록 교육되어질 필요가 있다.

참고문헌

[1] Brennan, K., & Resnick, M.(2012). New frameworks for studying and assessing the development of computational thinking. *American Educational Research Association meeting*, Vancouver, BC, Canada.

[2] HeeWon Yun, SungUk Choi(2017), Scratch Lesson Design for Effective Programming Concept Learning, *Journal of The Society of Convergence Knowledge* 5(2).

[3] Hyungshin Choi(2016), Developing Pre-service Teachers' Computational Thinking:Analysis of the Five Core CT Competencies, *Journal of The Korean Association of Information Education* 20(6), 553-562.

[4] Hyungshin Choi, Mi Song Kim(2017), A Complementary Approach of Three Methods for

- Computational Thinking Assessment, *Journal of The Korean Association of Information Education* 21(6), 639-646.
- [5] JeongBeom Song(2017), Effects of Learning through Scratch-Based Game Programming on Students' Interest in and Perceived Value of Mathematics Curriculum, *Journal of The Korean Association of Information Education* 21(2), 199-208.
- [6] Jesus Moreno-Leon, Gregorio Robles(2015). Dr. Scratch: a Web Tool to Automatically Evaluate Scratch Projects. *WiPSCE '15 November* 09-11.
- [7] Jiyae Noh, Jeongmin Lee(2018), Computational Thinking Assessment in SW Education Using Robot: Focused on Test, Bebras Challenge and Self-Report Questionnaire, *Journal of Educational Technology* 34(3), 849-876.
- [8] JungKyo Jung, Wonsuk Lee(2018), A Study of Current State and Direction of Evaluation of Software Education in Elementary School according to the 2015 Revised Curriculum, *Journal of Qualitative Research* 19(1), 22-30.
- [9] Juyeon Park, Jonghye Kim, Soohwan Kim, Sughee Kim, Hyunsook Lee(2017), Development of evaluation factors for SW education in elementary and secondary schools, *The Journal of Korean association of computer education* 20(6), 47-59.
- [10] Martin, T., & Fields, D. (2014). Macro data for micro learning: Developing FUN! for Automated Assessment of Computational Thinking in Scratch. *Cyber learning Summit*, Madison, WI.
- [11] Mi Song Kim, Hyungshin Choi(2018), Fostering Primary Pre-service Teachers' Computational Thinking through Self-Assessment, *Journal of The Korean Association of Information Education* 22(1), 61-70.
- [12] Minja Kim, Hyeoncheol Kim(2018), Effectiveness analysis based on computational thinking of a computing course for non-computer majors, *The Journal of Korean association of computer education* 21(1), 11-21.
- [13] Soohwan Kim(2015a), Analysis of Scratch code for Student Assessment about Computational Thinking Capability, *The Journal of Korean association of computer education* 18(5), 25-34.
- [14] Soohwan Kim(2015b), Development of Scratch Code Analysis System for Assessment about Concepts of Computational Thinking, *The Journal of Korean association of computer education* 18(6), 13-22.
- [15] Soojin Jun, Seonkwan Han(2016), Descriptive Assessment Tool for Computational Thinking Competencies, *Journal of The Korean Association of Information Education* 20(3), 255-262.
- [16] Dr.Scratch. <http://www.drscratch.org>

저자소개

박 선 주



1995 전남대학교 전산통계학과
(이학박사)

2003 George Mason University 객
원교수

1996~현재 광주교육대학교 컴퓨
터교육과 교수

관심분야 : 컴퓨터교육, SW교육,
앱개발, 빅데이터

E-mail : sjpark@gnue.ac.kr