

The Effects of Programming Education using App inventor on Problem-solving Ability and Self-efficacy, Perception

Seong-Won Kim*, Youngjun Lee**

Abstract

The ability to use information technology has become increasingly important as technological advances continue to sweep through the computing world, and education for improving computational thinking has become globally instituted. In South Korea, informatics subjects have been modified in the 2015 curriculum and are now compulsory in primary and secondary education. However, despite substantial financial investment and numerous studies promoting informatics education, there continues to be a serious lack of pre-service teachers capable of teaching computational thinking. This study investigated pre-service teacher programming education using App Inventor, their perceptions of App Inventor, and how use of the program affected teacher problem-solving abilities and self-efficacy. In the pre-test, the control group and experimental group showed no statistically significant difference; however, the post-test revealed that the two groups showed statistically significant differences in problem-solving skills and self-efficacy. The participants initially showed interest in using App Inventor; however, after practice-teaching and project-based learning, the participants demonstrated a growing negativity toward the program when they made errors and the functional limits of App Inventor became apparent. Although most participants stated that they would not use App Inventor in their classes, the positive statistically significant differences in problem-solving skills and self-efficacy indicate that this study could be utilized as a basis for building a teaching-learning program using App Inventor and creating an educational plan for teaching computational thinking.

▶ Keyword : Pre-service teacher, Problem-solving ability, Self-efficacy, Perception, App inventor

1. Introduction

2016년 다보스 포럼에서는 정보기술의 발달을 통하여 사회의 모습이 기존에 경험하지 못하는 새로운 형태로 변할 것이라고 예측하였다. 이와 같은 변화를 제4차 산업혁명이라고 말하였고, 이러한 변화를 주도할 원동력으로 인공지능, 로봇, 머신러닝 등을 제시하였다[1]. 이와 같이 사회의 모습의 변화를 주도하는 기술에 핵심에는 컴퓨팅 기술이 있다. 컴퓨팅 기술의 발전에 따라 사회 전반에 소프트웨어와 컴퓨팅 기기가 활용되고

있으며, 컴퓨팅 기술은 기존의 학문 영역과 융합되어 새로운 학문의 출현과 발달을 촉진하고 있다[2].

컴퓨팅 기술의 중요성이 강조됨에 따라서 전 세계적으로 컴퓨팅 기술을 활용할 수 있는 역량을 가르치기 위한 교육이 활발히 진행되고 있다. 미국에서는 컴퓨터과학 교육을 위한 프레임워크와 기준안을 개발하였으며[3,4], 영국은 교과목의 이름을 변경하고 모든 유·초·중등학생에게 필수로 컴퓨팅 교육을 실

• First Author: Seong-Won Kim, Corresponding Author: Youngjun Lee

*Seong-Won Kim (sos284809@gmail.com), Dept. of Computer Education, Korea National University of Education

**Youngjun Lee (yjlee@knu.ac.kr), Dept. of Computer Education, Korea National University of Education

• Received: 2016. 11. 14, Revised: 2016. 12. 01, Accepted: 2016. 12. 13.

• This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2013R1A2A2A03068459) and the Korea government (MSIP) (No. 2016R1A2B4010522).

시하고 있다[5]. 핀란드는 2016년부터 코딩 교육을 도입하고 있으며[6], 일본은 2020년부터 프로그래밍 교육을 교육과정에 필수화하겠다고 발표하였다[7]. 이외에도 다양한 나라에서 컴퓨팅 기술을 활용하여 문제를 해결할 수 있는 역량인 컴퓨팅 사고력을 기르기 위한 교육을 교육과정에 도입하고 있다[8, 9]. 한국도 이와 같은 흐름에 발맞추어 2015 개정 교육과정에서 정보 교과를 초·중학교 교육과정에 필수 교과로 지정되었으며[10], 정보 문화 소양, 컴퓨팅 사고력, 협력적 문제해결력을 핵심 역량으로 지정하고, 교육과정 내용의 많은 개정이 이루어졌다[11]. 또한, 소프트웨어 교육 생태계를 활성화시키기 위하여 정부와 산업계의 주도로 연구학교 운영, 교사 연수, 학생 참여 프로그램 운영, 다양한 교재 및 교수-학습 자료가 개발되고 있으며, 소프트웨어 교육과 관련된 연구를 진행하고 있다[12,13,14].

하지만, 소프트웨어 교육과 관련된 연구는 오랜 기간 동안 이루어지지 않아서 양과 질이 다른 교과에 비해 턱없이 부족한 상황이다. 연구의 대상을 살펴보면 교사와 학생을 대상으로 하는 연구가 대부분이고, 예비 교사를 대상으로 이루어진 연구는 거의 없었다. 실시된 연구들도 인지적 영역의 변화를 측정하거나 프로그래밍에 대한 추상적인 인식을 조사한 연구밖에 존재하지 않았다[15,16,17]. 2015 개정 교육과정이 도입되면 초·중학교에서는 교육용 프로그래밍 언어를 활용한 교육이 실시되어야 하는 상황인데, 예비 교사를 대상으로 한 교육용 프로그래밍 언어에 대한 인식이나 효과를 조사한 연구가 전무한 상황이다. 이와 같이 예비 교사를 대상으로 한 연구가 부족한 것은 소프트웨어 교육의 활성화에 걸림돌로 작용할 수 있다.

따라서 본 연구에서는 예비 교사를 대상으로 프로그래밍 교육을 실시하여 그 효과를 분석하고, 프로그래밍 언어에 대한 예비 교사의 인식 변화를 조사하였다. 이러한 연구를 위하여 앱 인벤터라는 블록형 프로그래밍 언어를 선정하였고, 예비 교사에게 앱 인벤터를 활용한 프로그래밍 교육을 실시하였다. 교육의 효과를 확인하기 위하여 사전, 사후에 문제 해결 능력과 자아 효능감의 변화와 앱 인벤터에 대한 예비 교사의 인식을 검사 도구를 활용하여 조사하였다. 이와 같은 연구는 앱 인벤터에 대한 실증적인 교육 및 연구 방향을 제시하고, 예비 교사를 위한 교수-학습 프로그램이나 교육 방안 설계의 기초 연구로 활용될 것이라고 생각된다.

II. Literature review

앱 인벤터를 활용한 프로그래밍 교육 연구는 최근 활발하게 연구가 이루어지고 있지만, 국내에서는 많은 연구가 부족하다[18,19,20,21,22,23,24]. 안상진과 이영준(2014)은 앱 인벤터를 초·중등 프로그래밍 교육에 활용할 수 있도록 교사와 학생들에게 앱 인벤터를 활용한 교육을 실시하였고, 교사와 학생들의

앱 인벤터 교육에 대한 의견을 조사하여 교육 방안을 제시하였다[18]. 한건우(2014)는 특성화 고등학교에서 앱 인벤터를 활용한 프로젝트 수업을 진행하고 수업에 대한 만족도를 조사하였다[19]. 유인환(2014)은 효과적인 로봇 프로그래밍 교육을 위하여 앱 인벤터를 연동한 로봇 프로그래밍 교육을 예비 교사에게 실시하였고, 수업, 로봇, 앱 인벤터, 로봇+ 앱 인벤터에 대한 의견을 조사하였다. 조사한 결과 앱 인벤터에 대한 의견은 긍정적으로 나타났다[20]. 임화경(2013)은 초등학생을 대상으로 앱 인벤터를 활용한 안드로이드 앱 제작 교육 프로그램을 개발하였다. 그리고 학생들의 의견을 조사한 결과, 앱 인벤터를 통한 교육에서 초등학생들은 흥미를 느끼고, 적극적으로 문제를 해결하는 태도를 보이는 것을 확인하였다[21]. 설문규와 손창익(2013)은 워터폴 모델을 적용하여 STEAM과 문제중심학습, 앱 인벤터를 활용하여 프로그래밍 교재를 개발하였다. 개발한 교재를 활용하여 초등학생에게 교육을 실시한 결과, 초등학생들의 학습 만족도가 높았으며, 긍정적인 생각과 흥미를 가지게 된 것을 확인하였다[22]. 안상진과 이영준(2016)은 예비 교사를 대상으로 앱 인벤터를 활용한 교육 프로그램을 실시하고, 프로그래밍 교수 내용 지식 변화를 조사하였다. 그 결과 문제기반 학습을 적용하였을 때, 교수내용지식, 교수법 지식이 향상되었고, 프로젝트 기반 학습을 적용하였을 때, 내용, 교수법, 교육과정 지식이 향상된 것을 확인하였다[23]. 김영주와 정현숙(2016)은 초·중등 프로그래밍 도구로써 앱 인벤터의 활용 방안을 모색하고, 고등학생을 대상으로 앱 인벤터 교육을 실시한 결과, 프로그래밍 기본 요소를 이해하는데 효과적이었으며, 학습 만족도가 높은 것을 확인하였다[24].

이와 같이 앱 인벤터에 대한 선행 연구를 살펴보면, 예비 교사를 대상으로 실시된 연구도 2편에 불과하다[20,23]. 또한, 앱 인벤터에 대한 흥미나 만족을 조사하고[20], Pedagogical Content knowledge(PCK)의 변화를 조사한 연구 밖에 없었다[23]. 프로그래밍 연구와 관련된 전체 연구를 살펴보면 예비 교사를 대상으로 실시된 연구가 부족하며, 자아 효능감이나 문제 해결 능력을 측정하는 연구는 거의 없었다.

III. Method

1. Research design

앱 인벤터를 활용한 프로그래밍 교육의 효과를 알아보기 위하여 예비 교사에게 자아 효능감과 문제 해결 능력 변화를 측정하는 검사 도구를 교육 처치 전, 후에 실시하였다. 또한, 앱 인벤터에 대한 예비 교사의 인식 변화를 조사하기 위하여 앱 인벤터의 컴포넌트를 학습한 후(S1), 교육 실습에 참여한 후(S2), 어플리케이션 제작 프로젝트에 참여한 후(S3)에 앱 인벤터에 대한 인식을 조사하는 검사를 실시하였다.

2. Test tool

앱 인벤터를 활용한 프로그래밍 교육이 예비 교사의 자아 효능감과 문제 해결 능력에 미치는 영향을 알아보기 위하여 선행 연구에서 활용한 검사 도구를 활용하였다.

예비 교사의 자아 효능감을 측정하기 위한 검사 도구는 김아영과 차정은(1996)의 연구에서 개발한 자아 효능감 척도를 활용하였다[25]. 김아영과 차정은(1996)은 Bandura(1977, 1986)을 비롯한 다양한 연구에서 개발한 자아효능감 척도를 분석하여, 자신감(E1), 자기 조절 효능감(E2), 과제난이도선호(E3) 3개의 영역으로 구성된 자아 효능감 척도를 개발하였다[26,27,28]. 자아효능감 척도는 총 24문항으로 구성되어 있으며, 5점 리커트 척도로 응답하도록 개발되었다. 검사 도구의 문항의 내적 합치도는 세 개의 척도가 모두 .80이 넘는 것으로 나타났다[26].

예비 교사의 문제 해결 능력을 조사하기 위한 검사 도구는 이석재 외(2003)의 연구에서 개발한 문제 해결 능력 검사지를 활용하였다[29]. 본 검사지는 유현석 등(2002)에서 정의한 문제 해결 능력을 측정하기 위해 개발한 검사 도구이다[29]. 검사지는 초등학생, 중/고등학생, 대학생/성인용으로 개발되었으며, 문제 명료화(P1), 원인 분석(P2), 대안 개발(P3), 계획/실행(P4), 수행평가(P5)라는 하위 영역으로 구성되어 있다. 문항은 총 45개이며, 검사에 참여한 예비 교사가 5점 리커트 척도로 응답하도록 개발되었다. 검사지는 문항의 상관분석, 변별력 분석, 신뢰도, 타당도 분석을 실시하여 완성도를 갖춘 검사 도구이다[29].

앱 인벤터를 활용한 프로그래밍 교육에 대한 예비 교사의 인식을 조사하고, 동시에 수업이 정의적 측면에서 학생들에게 미친 영향을 알아보기 위하여 검사 도구를 개발하여 사용하였다. 앱 인벤터에 대한 예비 교사의 인식 변화를 살펴보기 위하여 앱 인벤터와 다른 프로그래밍 언어와의 비교, 교육 내용에 대한 비교, 선호하는 강의 방식, 사용 여부를 묻도록 문항을 개발하였다. 문항은 모두 주관식으로 구성하였으며, 수업이 끝난 후 30분 동안 실시하였다. 연구자는 학생들의 검사 결과를 살펴보고, 답안 작성에 대한 이유를 묻는 인터뷰를 예비 교사에게 2차례 실시하였다(S1, S2). 인터뷰는 모두 녹음하였으며, 텍스트로 전사하여, 인식 변화 분석에 활용하였다.

3. Participant

본 연구에서 참여한 연구 대상은 충북 소재의 K 대학에 다니고 있는 예비 교사 68명이다. 앱 인벤터를 활용한 프로그래밍 교육의 효과를 알아보기 위하여 23명의 예비 교사(실험군)에게 수업을 적용하였다. 효과를 검증하기 위하여 예비 교사 45명을 통제군으로 설정하였다. 본 연구에서는 예비 교사를 모

Table 1. Gender of Participant

Group*	Male	Female	Total
Con.	19 (42%)	26 (58%)	45 (100%)
Exp.	8 (35%)	15 (65%)	23 (100%)
Total	27 (40%)	41 (60%)	68 (100%)

*Con.: Control group, Exp.: Experimental group

집하기 위하여 K 대학에 프로그래밍과 ICT 교육을 주제로 강의를 개설하였고, 예비 교사는 본인의 관심과 맥락적 요인을 고려하여 강의를 신청하였다. 이러한 과정에서 프로그래밍 수업이 하나가 폐강되어, 실험군과 통제군의 집단 크기의 차이가 발생하였다.

Table 2. Grade of Participant

Group	Fresh man	Sophomore	Junior	Senior	Total
Con.	18 (40%)	15 (33%)	9 (20%)	3 (7%)	45 (100%)
Exp.	0 (0%)	0 (0%)	16 (70%)	7 (30%)	23 (100%)
Total	18 (26%)	15 (22%)	25 (37%)	10 (15%)	68 (100%)

연구에 참여한 대상들의 특성을 살펴보면, 전체 예비 교사의 성별은 전체 68명 중 남성이 27명(40%), 여성이 41명(60%)으로, 남자보다 여자가 더 많이 연구에 참여한 것을 확인할 수 있었다. 이러한 성별 분포는 실험 집단과 통제집단에서도 비슷한 분포로 나타났다, See Table1.

예비 교사의 학년을 살펴보면, 3학년이 예비 교사가 25명(37%)으로 가장 많았고, 4학년인 예비 교사가 10명(15%)으로 가장 적었다. 이러한 결과는 예비 교사들이 교사 임용 시험에 대비하여 4학년에 수업을 많이 듣지 않기 때문에 나타난 현상으로 생각된다. 1, 2학년은 각각 26%와 22%로 전체 연구 대상의 1/4(25%)에 가까운 수치를 보였다. 통제군의 학년 분포를 살펴보면, 1학년(40%)과 2학년(33%)이 압도적으로 많은 것을 확인할 수 있었다. 3학년은 20%정도 수치를 보였지만, 4학년은 3명(7%)로 가장 적었다. 실험 집단에서는 1, 2학년이 한 명도 없었고, 3, 4학년이 대다수를 차지하고 있었다. 이와 같이 연구 대상의 학년이 편중되어 있는 것은 연구 대상 모집이 수강 신청을 통해 자발적으로 이루어졌기 때문이라고 생각된다, See Table 2.

연구 대상의 전공도 집단별로 차이가 존재하였다. 통제군을 살펴보면, 지구과학(18%), 프랑스어(16%), 컴퓨터(11%), 수학(10%), 초등(10%) 등 다양한 전공이 예비 교사로 구성되었다. 모든 전공의 교사가 모집된 것은 아니지만, 다양한 전공의 예비 교사로 통제 집단에 참여하였다. 실험 집단은 통계 집단과 다르게 예비 교사의 전공이 모두 컴퓨터(100%)였다. 강의의 주제가 예비 교사가 느끼기에 ICT 교육에 비해 프로그래밍 교육이 어렵고 전문적인 주제로 받아들여져서, 이와 같은 결과가 나타난 것이라고 생각된다, See Table 3.

경험한 프로그래밍 언어에 따라 다른 결과가 나타날 수 있기

Table 3. Major of Participant

Group	Major				Total
Con.	Earth science	8 (18%)	Fine arts	2 (4%)	45 (100%)
	French	7 (16%)	Chinese	2 (4%)	
	Computer	5 (11%)	Geography	1 (2%)	
	Mathematic	4 (10%)	Home economics	1 (2%)	
	Primary	4 (10%)	English	1 (2%)	
	Korean	3 (7%)	Environment	1 (2%)	
	Pedagogy	2 (4%)	History	1 (2%)	
	Biology	2 (4%)	Physics	1 (2%)	
Exp.	Computer			23 (100%)	23 (100%)

때문에 앱 인벤터 프로그래밍 교육을 받은 예비 교사의 프로그래밍 언어를 조사하였다. 본 연구에서 실험 집단은 모두 컴퓨터 교육이였으며, 학년이 3, 4학년에 몰려있기 때문에 모든 학생이 C(100%)를 사용해 본 것으로 나타났다. 이외에도 스크래치(87%)와 엔트리(39%)와 같이 블록 기반 프로그래밍 언어를 경험한 예비 교사도 많았으며, Java(17%), Visual basic(13%), C++(9%), HTML(9%), PHP(9%)와 같이 다양한 종류의 텍스트 기반 언어를 경험한 예비 교사도 존재하였다. 다음과 같은 프로그래밍 언어 경험을 가지므로 본 연구에서 앱 인벤터에 대한 인식은 프로그래밍에 대하여 경험이 많은 예비 교사가 가지는 인식이다.

4. Treatment

통제 집단에게는 ICT 교육 내용을 중심으로 한 학기동안 강의를 진행하였다. 예비 교사에게 필요한 ICT 역량을 기르는 데 초점을 맞추어, 엑셀, 파워포인트, 워드를 활용 방법을 배우고 배운 내용을 기반으로 결과물을 제작하는 활동으로 강의를 구성하였다[18].

실험 집단의 수업 내용을 살펴보면, 앱 인벤터의 프로그래밍 환경과 컴포넌트를 익히고, 직접 어플리케이션을 설계 및 제작하는 활동으로 구성하였다. 실험 집단이 이미 C나 스크래치, 엔트리와 같은 프로그래밍 언어를 배운 상태였으므로 순차, 반복, 선택과 같은 프로그래밍 기본 개념에 설명은 간략하게 진행하였고, 앱 인벤터라는 프로그래밍 언어에 익숙해질 수 있도록 수업을 실시하였다. 따라서 컴포넌트를 배우고, 컴포넌트를 활용한 어플리케이션을 직접 제작하는 실습으로 구성하였다. 또한, 배운 컴포넌트를 활용하여 특정 기능을 수행할 수 있는 어플리케이션을 만드는 과제를 매주 부여하여, 예비 교사의 문제 해결 능력과 자아 효능감을 향상시키고자 하였다.

앱 인벤터의 컴포넌트의 학습을 실시한 후, 교육 실습을 통하여 앱 인벤터에 대한 모의 수업 자료 개발과 본인이 원하는 어플리케이션을 개발하는 프로젝트를 수행하였다. 이러한 과정을 통해서 실생활에 필요한 어플리케이션을 직접 제작하고, 앱

인벤터를 교육적으로 활용하기 위한 방안을 모색하는 과제를 실시하였다, See Table 4.

5. Statistically analysis

앱 인벤터를 활용한 프로그래밍 강의 전, 후로 통제 집단과 실험 집단의 문제 해결 능력과 자아 효능감의 차이가 있는지 알아보기 위하여 두 집단의 사전, 사후 검사 결과를 각각 독립 표본 t-검정으로 분석하였다. 또한, 앱 인벤터를 활용한 프로그래밍과 ICT 교육이 예비 교사의 문제 해결 능력과 자아 효능감에 미치는 영향을 알아보기 위하여 각 집단의 사전, 사후 검사를 대응 표본 t-검정으로 변화를 살펴보았다. 이러한 분석을 통하여 앱 인벤터를 활용한 프로그래밍 수업의 효과를 검증하고자 하였다. 통계 처리와 분석은 IBM사의 SPSS v.21을 활용하였다. 앱 인벤터에 대한 인식을 조사한 문항은 검사 결과를 범주화하였다. 그 빈도를 분석하였다. 범주화 과정은 4명의 컴퓨터 교육 전문가가 예비 교사의 응답에 해당하는 단어를 여러 개 추출하였고, 추출 결과를 교차 검증을 통해 범주화하였다. 범주화한 결과를 컴퓨터 교육 전문가 3명에게 검증 작업을 실시하여 연구 결과의 타당도를 확보하였다.

Table 4. Content of treatment

Lesson	Content	
	Control group	Experimental group
1		Orientation
2	- Excel(1)	- Environment of App inventor - Camera component - Acceleration sensor - Operator
3	- Excel(2)	- Canvas component - Player component - Voice recognition component - Text-to-speech conversion component
4	- Excel(3)	- direction sensor - Timer - Web viewer - Activity starter
5	- Design of class with Excel	- Position sensor - Database - Phone book - SMS
6	- Powerpoint(1)	- Design of Application
7		Mid-term
8	- Powerpoint(2)	- Teaching practice & project (1)
9	- Powerpoint(3)	- Teaching practice & project (2)
10	- Design of class with Powerpoint	- Teaching practice & project (3)
11	- Word(1)	- Teaching practice & project (4)
12	- Word(2)	- Simulated Instruction with App inventor
13	- Word(3)	- Project
14	- Design of class with Word	- Seminar(Result of application)
15		Final exam

IV. Result & Discussion

1. The Effects of Programming education using a app inventor

1.1 Independent samples t-test result of pre-test

예비 교사의 사전 검사를 독립 표본 t-검정으로 분석한 결과는 다음과 같다. 문제 해결 능력의 경우에는 통제 집단($M=157.98$, $SD=12.59$)과 실험 집단($M=164.48$, $SD=15.70$)이 통계적으로 유의미한 차이를 확인할 수 없었다, $t=-1.85$, $p=.07$. 세부 영역을 살펴봐도 통계적으로 유의미한 차이가 나타난 영역은 하나도 없었다, See Table 5. 이와 같은 결과를 통하여 교육 프로그램을 적용하기 전의 예비 교사의 문제 해결 능력은 집단 간의 차이가 없었다.

자아 효능감을 살펴보면, 문제 해결능력과 같은 결과가 나타났다. 실험 집단($M=77.91$, $SD=18.50$)과 통제 집단($M=77.67$, $SD=5.47$)의 자아 효능감 전체 결과는 통계적으로 유의미한 차이가 나타나지 않았다, $t=-.08$, $p=.93$. 세부 영역인 자신감($t=-.52$, $p=.60$), 자기 조절($t=-1.00$, $p=.32$), 과제 난이도($t=1.19$, $p=.24$)에서도 모두 유의미한 차이가 없었다, see Table 6 이와 같은 결과를 통하여 교육 프로그램 적용 전에 실험 집단과 통제 집단의 자아효능감이 같다는 것을 확인할 수 있었다. 이와 같은 결과를 통하여 실험 집단과 통제 집단의 문제 해결 능력과 자아 효능감이 같은 동질 집단이라고 판단하였다.

1.2 Independent samples t-test result of post-test

사후 검사 결과를 살펴보면, 문제 해결 능력은 실험 집단과 통제 집단이 통계적으로 유의미한 차이를 보였다, $t=-3.16$, $p<.01$. 또한, ICT 교육을 받은 예비 교사($M=159.13$, $SD=11.93$)에 비해 앱 인벤터를 활용한 프로그래밍 교육을 받은 예비 교사($M=170.17$, $SD=16.48$)의 문제 해결 능력 값이 더 큰 것으로 나타났다. 세부 영역을 살펴보면, 문제 명료화($t=-2.30$, $p=.03$), 원인 분석($t=-2.11$, $p=.04$), 계획/실행($t=-2.86$, $p=.01$), 수행평가($t=-3.19$, $p<.01$)에서 모두 실험 집단과 통제 집단이 통계적으로 유의미한 차이를 확인할 수 있었다. 반면에 대안개발 영역에서는 통계적으로 유의미한 차이가 나타나지 않았다, $t=-1.40$, $p=.17$, See Table 7.

자아 효능감을 측정된 검사 결과를 살펴보면, 문제 해결 능력과 마찬가지로 실험 집단과 통제 집단은 통계적으로 유의미한 차이가 나타났다, $t=-2.53$, $p=.01$. 또한, ICT 교육을 받은 예비 교사($M=77.51$, $SD=5.92$)에 비해 프로그래밍 교육을

Table 5. Independent samples t-test results of pre-test to measure for problem-solving ability

Area*	Group	N	M	SD	t	P
P1	con	45	18.40	2.35	-1.48	.15
	exp	23	19.17	1.23		
P2	con	45	34.71	2.97	-0.14	.89
	exp	23	34.83	3.38		
P3	con	45	34.98	3.70	-1.45	.15
	exp	23	36.39	4.03		
P4	con	45	33.24	5.01	-1.39	.17
	exp	23	34.91	3.96		
P5	con	45	36.64	3.40	-1.20	.24
	exp	23	37.87	4.98		
Total	con	45	157.98	12.59	-1.85	.07
	exp	23	164.48	15.70		

P1: Problem clarifying, P2: Analysis of cause, P3: Development of alternative, P4: Planning/implementation, P5: Performance assessment

받은 예비 교사($M=81.91$, $SD=8.23$)의 자아 효능감이 값이 더 큰 것으로 나타났다. 세부 영역을 살펴보면, 자기 조절 영역은 실험 집단($M=41.71$, $SD=4.81$)이 통제 집단($M=46.00$, $SD=6.75$)에 비해 결과 값이 컸으며, 이 차이는 통계적으로 유의미한 것으로 나타났다, $t=-3.03$, $p<.01$. 하지만 자신감($t=-.08$, $p=.93$), 과제난이도($t=.00$, $p=1.00$)에서는 통계적으로 유의미한 차이가 나타나지 않았다. 사후 검사 결과 값을 살펴봐도 자신감과 과제 난이도 영역에서는 실험 집단과 통제 집단의 값 차이가 거의 없었다. 따라서 전체 검사 결과에서 생긴 자아 효능감의 차이가 자기 조절 영역의 값 차이 때문에 발생하였다는 것을 확인할 수 있었다, See Table 8. 사전 검사 결과에서는 예비 교사의 문제 해결 능력과 자아 효능감의 차이는 존재하지 않았다. 하지만, 사후 검사에서는 통계적으로 유의미한 차이가 나타났다. 이러한 변화가 생긴 이유를 살펴보기 위하여 집단별로 대응 표본 t-검정을 실시하였다.

1.3 Paired samples t-test result of control group

통제 집단의 문제 해결 능력을 살펴보면, 사전 검사($M=157.98$, $SD=12.59$)에 비해 사후 검사($M=159.13$, $SD=11.93$) 결과가 커진 것을 확인할 수 있었다. 하지만 사전-사후 검사를 대응 표본 t-검정으로 분석한 결과, 통계적으로 유의미 Table 6. Independent samples t-test results of pre-test to measure for self-efficacy

Area*	Group	N	M	SD	t	P
E1	con.	45	20.13	5.47	0.52	0.60
	exp.	23	19.35	6.63		
E2	con.	45	41.40	5.14	-1.00	0.32
	exp.	23	43.26	10.28		
E3	con.	45	16.13	1.63	1.19	0.24
	exp.	23	15.30	4.09		
Total	con.	45	77.67	5.47	-0.08	0.93
	exp.	23	77.91	18.50		

*E1: Confidence, E2: Self-regulated efficacy, E3: Preference of task difficulty

Table 7. Independent samples t-test results of post-test to measure for problem-solving ability

Area*	Group	N	M	SD	t	P
P1	con	45	18.31	2.38	-2.30	.03
	exp	23	19.78	2.71		
P2	con	45	35.20	2.95	-2.11	.04
	exp	23	37.13	4.58		
P3	con	45	35.24	3.89	-1.40	.17
	exp	23	36.65	4.01		
P4	con	45	33.76	4.29	-2.86	.01
	exp	23	37.26	5.63		
P5	con	45	36.62	3.38	-3.19	.00
	exp	23	39.35	3.24		
Total	con	45	159.13	11.93	-3.16	.00
	exp	23	170.17	16.48		

한 차이는 나타나지 않았다, $t = -.41, p = .68$. 하위 영역을 살펴보면, 문제 명료화($t = .19, p = .85$), 원인 분석($t = -.71, p = .48$), 대안 개발($t = -.30, p = .77$), 계획/실행($t = -.53, p = .60$), 수행 평가($t = .03, p = .98$) 영역 모두에서 통계적으로 유의미한 차이를 확인할 수 없었다, See Table 9. 이와 같은 결과를 통하여 ICT 교육은 예비 교사의 문제 해결 능력에 영향을 주지 않는다는 것을 확인할 수 있었다.

ICT 교육을 받은 예비 교사의 자아 효능감을 살펴보면, 사전 검사($M = 77.67, SD = 5.47$)에 비해 사후 검사($M = 77.51, SD = 5.92$)는 감소한 것을 확인할 수 있었다. 또한, 이러한 차이는 통계적으로 유의미하지 않았다, $t = .13, p = .90$. 하위 영역인 자신감($t = .36, p = .72$), 자기 조절($t = -.29, p = .77$), 과제 난이도($t = .13, p = .90$)에서도 같은 결과가 나타났다, See Table 10.

ICT 교육을 받은 예비 교사의 문제 해결 능력과 자아 효능감을 대응 표본 t-검정으로 분석한 결과, 통계적으로 유의미한 차이가 모든 영역에서 확인할 수 없었다. 따라서 ICT 교육은 예비 교사의 문제 해결 능력과 자아 효능감에 영향을 주지 않는다는 것을 확인할 수 있었다.

1.4 Paired samples t-test result of experimental group

앱 인벤터를 활용한 프로그래밍 교육을 받은 예비 교사의 사전, 사후 검사 결과를 대응 표본 t-검정으로 분석한 결과는 다음과 같다. 문제 해결 능력에서는 사전 검사($M = 164.48, SD =$

Table 8. Independent samples t-test results of post-test to measure for self-efficacy

Area*	Group	N	M	SD	t	P
E1	con.	45	19.71	5.26	-0.08	0.93
	exp.	23	19.83	5.66		
E2	con.	45	41.71	4.81	-3.03	0.00
	exp.	23	46.00	6.75		
E3	con.	45	16.09	1.64	0.00	1.00
	exp.	23	16.09	2.71		
Total	con.	45	77.51	5.92	-2.53	0.01
	exp.	23	81.91	8.23		

Table 9. Paired samples t-test results of control group to measure for problem-solving ability

Area*	Group	N	M	SD	t	P
P1	pre	45	18.40	2.35	0.19	0.85
	post		18.31	2.38		
P2	pre	45	34.71	2.97	-0.71	0.48
	post		35.20	2.95		
P3	pre	45	34.98	3.70	-0.30	0.77
	post		35.24	3.89		
P4	pre	45	33.24	5.01	-0.53	0.60
	post		33.76	4.29		
P5	pre	45	36.64	3.40	0.03	0.98
	post		36.62	3.38		
Total	pre	45	157.98	12.59	-0.41	0.68
	post		159.13	11.93		

15.70)에 비해 사후 검사 결과($M = 170.17, SD = 16.48$)가 증가한 것을 확인할 수 있었다. 하지만 증가한 값은 통계적으로 유의미한 차이가 나타나지 않았다, $t = -1.19, p = .25$. 세부 영역을 살펴보면, 문제 명료화($t = -.96, p = .35$), 원인 분석($t = -2.12, p = .05$), 대안 개발($t = -.21, p = .84$), 계획/실행($t = -1.48, p = .15$), 수행 평가($t = -1.24, p = .23$)에서도 통계적으로 유의미한 차이가 나타나지 않았다, See Table 11.

자아 효능감도 문제 해결 능력과 같은 결과가 나타났다. 예비 교사의 자아 효능감은 사전 검사($M = 77.91, SD = 18.50$)와 사후 검사($M = 81.91, SD = 8.23$)는 통계적으로 유의미한 차이를 확인할 수 없었다, $t = -1.03, p = .32$. 세부 영역에서도 자신감($t = -.28, p = .78$), 자기 조절($t = -1.01, p = .33$), 과제 난이도($t = -.77, p = .45$) 모두 통계적으로 유의미한 차이가 나타나지 않았다, See Table 12.

이와 같은 결과를 통하여 앱 인벤터를 활용한 프로그래밍 교육은 예비 교사의 문제 해결 능력과 자아 효능감의 변화에는 영향을 주지 않는다는 것을 확인할 수 있었다. 따라서 사후 검사에서 예비 교사간의 문제 해결 능력과 자아 효능감의 차이가 발생한 이유가 앱 인벤터를 활용한 프로그래밍 교육 때문에 발생한 차이가 아니었다. 이러한 차이가 생긴 이유는 집단 간의 차이 값이 컸기 때문이라고 생각된다. 문제 해결 능력은 통제 집단과 실험 집단이 모두 사전 검사에 비해 사후 검사에서 값이 증가하였지만, 실험 집단의 증가폭이 더 컸다. 따라서 사전 검사에 비해 사후 검사에서 집단 간의 차이가 커짐에 따라 이

Table 10. Paired samples t-test results of control group to measure for self-efficacy

Area*	Group	N	M	SD	t	P
E1	pre	45	20.13	5.47	0.36	0.72
	post		19.71	5.26		
E2	pre	45	41.40	5.14	-0.29	0.77
	post		41.71	4.81		
E3	pre	45	16.13	1.63	0.13	0.90
	post		16.09	1.64		
Total	pre	45	77.67	5.47	0.13	0.90
	post		77.51	5.92		

러한 차이가 발생하였다고 생각된다. 통계적으로 유의미하지는 않았지만, 집단 간의 차이가 커져서 이와 같은 결과가 나온 것이라고 판단하였다. 자아 효능감은 사전, 사후 검사 결과를 비교한 결과, 통제 집단은 값이 오히려 감소하고, 실험 집단은 값이 증가한 것으로 나타났다. 따라서 사전 검사에 비해 사후 집단에서 두 집단 간의 차이가 증가함에 따라서 두 집단의 자아 효능감이 통계적으로 유의미한 차이를 보인 것으로 생각된다.

이러한 연구는 선행 연구와 다른 결과였다. 양창모(2014a)의 연구에서는 교육용 프로그래밍 언어를 사용한 프로그래밍 교육의 효과를 메타 분석으로 살펴본 결과, 자아 효능감에 높은 효과를 보이고, 문제 해결 능력에는 적은 효과를 보인다고 말하였다. 본 연구의 결과는 앱 인벤터가 교육용 프로그래밍 언어이지만 선행 연구와 반대로 자아 효능감과 문제 해결 능력에 영향을 주지 않고 있음을 보여주었다[30]. 또한, 양창모(2014b)의 연구에서는 프로그래밍 교육이 문제 해결 능력과 자아 효능감에 영향을 주며, 나이가 증가함에 따라 그 효과가 커진다고 말하였다[31]. 본 연구는 예비 교사를 대상으로 이루어졌음에도 선행 연구와 반대되는 결과가 나타났다. 진영학과 김영식(2011)의 연구에서도 교육용 프로그래밍 언어는 문제 해결 능력과 자아 효능감에 영향을 주고 있으며, 나이가 많아질수록 효과가 커진다고 말하였으나 본 연구에서는 이러한 결과와 반대되는 결과가 나타났다[32]. 이와 같이 앱 인벤터는 교육용 프로그래밍 언어이지만, 기존의 교육용 프로그래밍 언어 연구와 프로그래밍 연구와는 다른 결과가 나타났다.

2. The perception on pre-service teacher toward app inventor

앱 인벤터를 활용한 수업이 기존 프로그래밍 수업보다 좋았던 점을 묻는 문항에서는 ‘실습’, ‘흥미’, ‘어플리케이션 개발’, ‘블록 기반 프로그래밍 언어’, ‘기타’, ‘없다’라고 응답한 의견이 있었다. 먼저 ‘실습’은 컴포넌트 학습을 하였을 때 9%지만 최종 프로젝트를 한 후에 19%로 증가한 것을 나타났다. ‘흥미’있어서 좋았다는 의견은 컴포넌트 학습 이후에 19%로 높은 비중을 차지하였지만, 교육 실습과 프로젝트 학습을 한 이후로 13%로

감소하였다. 그 다음으로 ‘어플리케이션 개발’은 컴포넌트 학습에서 22%, 교육 실습 이후에 44%, 프로젝트 수행 이후에 23%를 기록하였다. 또한, ‘블록 기반 프로그래밍 언어’여서 좋았다는 예비 교사는 컴포넌트 학습을 할 때는 높은 비중을 차지하고 있지만(22%), 점점 비중이 작아지는 것을 확인할 수 있었다(11%, 19%). 마지막으로 ‘기타’는 다른 프로그래밍 수업과 다르게 디자인을 할 수 있고, 프로젝트형 수업을 진행할 수 있어서 좋았다는 응답이 있었다. 또한, 프로젝트 수업을 진행한 후 13%의 예비 교사가 성취감을 얻을 수 있어서 좋았다고 응답하였다. 이와 같은 결과를 통하여, 예비 교사는 기존의 프로그래밍 수업보다 앱 인벤터를 활용한 프로그래밍 수업은 처음에는 블록 기반 프로그래밍 언어이며, 어플리케이션을 만들 수 있으며, 흥미를 유발한다고 응답하였다. 하지만, 교육 실습 이후에는 블록 기반 프로그래밍과 흥미에 대한 의견은 급격히 감소하였다. 반면에 어플리케이션을 개발할 수 있어서 좋았다는 의견을 절반에 가까운 예비 교사가 응답하였다. 마지막으로 최종 어플리케이션을 만드는 과제에서는 실습을 통해 어플리케이션을 만들 수 있다는 점이 다른 프로그래밍 언어보다 좋았던 점이라고 응답한 것을 확인할 수 있었다. 따라서 앱 인벤터를 활용한 프로그래밍 수업은 초기에 블록형 프로그래밍 언어라는 점에서 예비 교사의 흥미를 유발하지만, 이러한 흥미는 개발하는 과정에서 감소한다는 것을 확인할 수 있었다. 하지만, 실습을 통해 어플리케이션을 개발한다는 점은 지속적으로 유지된다는 것을 확인할 수 있었다.

다른 프로그래밍 언어와 비교한 문항을 살펴보면, ‘블록 기반 프로그래밍 언어’, ‘흥미’, ‘어플리케이션 개발’, ‘직관적’, ‘기타’, ‘없다’라고 답한 의견이 있었다. 본 연구의 연구 대상은 이미 C나 다양한 텍스트 기반 언어와 스크래치와 같은 블록형 언어를 배우고 왔지만, 앱 인벤터가 블록형 프로그래밍 언어이기 때문에 다른 언어보다 좋았다는 의견이 모든 시점에서 가장 큰 비중을 차지하고 있었다. 다른 언어보다 재미있다고 응답한 의견은 컴포넌트 학습 이후에는 3%에 불과하였지만, 교육 실습과 프로젝트 수행 이후에 13%로 증가하였다. 어플리케이션을 제작할 수 있어서 좋았다는 의견은 처음에는 26%로 높은 비중을 차지하였지만, 교육 실습 이후에 19%로 감소한 것을 확인할 수 있었다. 하지만 프로젝트 수업 이후에 32%로 크게 향상되었다. 직관적이어서 좋았다는 의견도 컴포넌트 학습 이후에 상당한

Table 11. Paired samples t-test results of experimental group to measure for problem-solving ability

Area*	Group	N	M	SD	t	P
P1	pre	23	19.17	1.23	-0.96	0.35
	post		19.78	2.71		
P2	pre	23	34.83	3.38	-2.12	0.05
	post		37.13	4.58		
P3	pre	23	36.39	4.03	-0.21	0.84
	post		36.65	4.01		
P4	pre	23	34.91	3.96	-1.48	0.15
	post		37.26	5.63		
P5	pre	23	37.87	4.98	-1.24	0.23
	post		39.35	3.24		
Total	pre	23	164.48	15.70	-1.19	0.25
	post		170.17	16.48		

Table 12. Paired samples t-test results of experimental group to measure for self-efficacy

Area*	Group	N	M	SD	t	P
E1	pre	23	19.35	6.63	-0.28	0.78
	post		19.83	5.66		
E2	pre	23	43.26	10.28	-1.01	0.33
	post		46.00	6.75		
E3	pre	23	15.30	4.09	-0.77	0.45
	post		16.09	2.71		
Total	pre	23	77.91	18.50	-1.03	0.32
	post		81.91	8.23		

비중을 차지하고 있었지만, 교육 실습 후 감소하고, 최종 프로젝트 이후에 다시 증가한 형태를 보였다. 마지막으로 기타 의견에서는 디자인을 할 수 있어서 좋았다는 의견이 소수였지만, 모든 시점에 나타났고, 실생활과 관련 있어서 좋았다, 한글이어서 좋았다, 문법 오류에 대한 걱정 없이 프로그래밍 할 수 있어서 좋았다는 의견, 성취감을 느낄 수 있어서 좋았다는 의견이 있었다, See Table 13.

이와 같은 결과를 종합해보면, 앱 인벤터를 활용한 프로그래밍 수업에 대해서 예비 교사들이 가장 긍정적으로 생각하는 부분은 어플리케이션을 직접 만들어 볼 수 있다는 점이었다. 다른 프로그래밍과 다르게 실제 결과물을 확인할 수 있다는 점 때문에 어플리케이션 제작할 수 있다는 점을 좋게 평가하고 있었다. 또한, 블록형 프로그래밍 언어라는 점도 다른 프로그래밍 언어보다 좋은 점이라고 응답하였고, 다른 수업과 비교하였을 때, 쉽고 재미를 느끼게 해주는 요소라고 응답하였다. 예비 교사들이 느끼는 앱 인벤터에 대한 긍정적인 부분은 컴포넌트 학습, 교육 실습, 프로젝트 학습을 진행되는 동안 큰 변화는 나타나지 않았다. 이를 통하여 앱 인벤터가 가진 자체 특성(블록형 프로그래밍 언어, 어플리케이션 개발)을 다른 언어보다 긍정적으로 받아들이고 있다고 생각된다.

다른 프로그래밍 수업과 비교하였을 때 부정적인 점에 대해서 응답한 결과를 살펴보면, ‘기능의 한계’, ‘빌드’, ‘오류’, ‘기타’, ‘없다’라고 응답한 의견이 있었다. 의견 중에서 S1, S2, S3에서 모두 ‘없다’라고 응답한 의견이 가장 많았다. 있다고 응답한 답도 일부 예비 교사가 ‘기능의 한계’, ‘빌드’, ‘오류’ 때문에 강의를 듣는데 어려움을 느꼈다고 응답하였다. 또한, ‘기타’에서는 앱 인벤터를 구현하기 위해서는 반드시 안드로이드 기기가 필요하다는 점과 관련 자료가 부족하다는 의견이 있었다. 이와 같은 의견들은 컴포넌트 학습과 교육 실습을 하면서 급격히 줄었다. 따라서 예비 교사의 의견은 배우는 과정에서 발생하는

어려움이었으며, 실질적으로 앱 인벤터를 활용하여 결과물을 제작할 때 느끼는 어려움은 아니었다. 반면에 컴포넌트 학습 이후로 실질적인 어플리케이션 개발이 들어감에 따라 느끼는 좋지 않았던 점에 대해서는 어플리케이션 제작에 블록이 많이 필요하고 용량의 제한 때문에 힘이 들었다고 응답하였다.

앱 인벤터는 프로그래밍 언어가 다른 프로그래밍 언어에 비해 좋지 않은 점으로 기능적 한계라고 응답한 예비 교사가 가장 많았다. 학생들은 구현하고 싶은 어플리케이션 디자인과 기능이 있지만, 앱 인벤터가 교육용 프로그래밍 언어로 개발되다보니 한계로 구현하는데 한계를 느끼고 있었다. 이러한 의견은 프로젝트 이후에 절반에 가까운 학생이 기능 한계가 다른 프로그래밍 언어보다 많다고 응답하였다. 그 다음으로 어플리케이션 결과물을 확인하기 위해서 빌드 작업을 꼭 시행하여야 한다는 점이 프로그래밍 언어로서 좋지 않았던 점이라고 응답하였다. 이러한 의견은 컴포넌트 학습(19%)과 교육실습(25%)에서 높은 수치를 기록하였지만, 프로젝트 활동(12%) 이후에는 감소한 것을 확인할 수 있었다. 이외에도 어플리케이션을 제작할 때 블록이 많이 필요하다는 점도 같은 변화를 보였다. 반면에 앱 인벤터가 가진 오류가 많아 다른 프로그래밍 언어보다 불편하다는 의견과 기타 의견은 컴포넌트 학습에 비해 프로젝트 활동 이후에 증가한 것을 확인할 수 있었다. 기타 의견은 디자인의 한계가 존재한다는 점과 변수 설정이 다른 프로그래밍 언어에 비해 어렵다는 점, 프로그래밍 언어 자체 구동이 느리고 인터페이스가 불편하다는 의견이 있었다. 다른 프로그래밍 수업과 비교한 내용과는 다르게 프로그래밍 언어에서는 없다는 의견이 수업을 진행할수록 낮은 수치를 보였다, See Table 14.

이와 같은 결과를 종합해보면, 예비 교사들은 앱 인벤터를 활용한 프로그래밍 수업에 대해서 다른 프로그래밍 수업보다 좋지 않은 점이 없다고 응답한 학생이 많았다. 또한, 좋지 않은 점도 앱 인벤터 자체의 문제가 많았다. 따라서 프로그래밍 수업에서 앱 인벤터를 사용하는 것에 대하여 부정적인 의견은 거의 없으며, 프로그래밍 수업에서 다른 언어보다 앱 인벤터를 사용하는 것에 대해 긍정적이라는 것을 확인할 수 있었다. 반면에 프로그래밍 언어 자체를 비교하는 부분에서는 다양한 의견이 나타났다. 앱 인벤터가 다른 프로그래밍 언어보다 좋지 않았던 점을 살펴보면, 수업이 진행됨에 따라 빌드나 기능의 한계에 대한 불만은 줄어든 것으로 나타났다. 하지만 본인의 어플리케이션을 구현하는 과정에서 기존의 프로그래밍 언어에서 경험할 수 없었던 용량 문제나 오류, 블록의 복잡함에 대한 의견은 증가한 것을 확인할 수 있었다. 이와 같이 앱 인벤터가 자체 문제들은 학생들이 프로젝트 활동이나 문제 해결 활동을 할 때, 문제 해결을 가로막는 장애물로 작용할 수 있고, 실험 집단의 문제 해결력과 자아 효능감의 향상에 저해하는 요소였다고 생각된다. 이러한 문제는 앱 인벤터가 보완됨에 따라 충분히 해결될 수 있지만, 해결되기 전까지는 앱 인벤터를 활용한 교육 프로그램을 설계할 때 이러한 요소를 고려해야 한다고 생각된다. 시간

Table 13. The positive perception on pre-service teacher of App inventor through a comparison

Area	Domain	S1	S2	S3
Programming course	Practice	3 (9%)	2 (7%)	6 (19%)
	Interest	6 (19%)	2 (7%)	4 (13%)
	Application	7 (22%)	12 (44%)	7 (23%)
	Block based language	7 (22%)	3 (11%)	6 (19%)
	etc.	7 (22%)	6 (22%)	5 (16%)
	None	2 (6%)	2 (7%)	3 (10%)
Total		32 (100%)	27 (100%)	31 (100%)
Programming Language	Block based language	13 (37%)	13 (42%)	10 (32%)
	Interest	1 (3%)	4 (13%)	4 (13%)
	Application	9 (26%)	6 (19%)	10 (32%)
	intuitive interface	4 (11%)	0 (0%)	3 (10%)
	etc.	4 (11%)	6 (19%)	3 (10%)
	None	4 (11%)	2 (6%)	1 (3%)
Total		35 (100%)	31 (100%)	31 (100%)

이 지남에 따라 용량이나 오류, 블록의 복잡함과 같은 문제가 해결된다면 예비 교사가 가진 프로그래밍 언어에 대한 부정적인 요소는 많이 없어질 것이라고 생각된다.

다음으로 본인이 교사가 된다면 학생들에게 앱 인벤터를 활용한 프로그래밍 수업을 어떠한 방법으로 진행할 것인지에 대한 문항 결과를 살펴보았다. 예비 교사들은 기존의 스크래치나 엔트리와는 다른 블록 형식을 가지며, 각각의 컴포넌트에 속성을 지정해주어야 한다는 점 때문에 다른 프로그래밍 언어보다 배우는 데 어려움을 느낀다고 응답하였다. 따라서 컴포넌트를 익히기 위하여 설명을 해주는 강의식과 블록을 따라서 조립해보는 실습형 수업이 필요하다고 응답한 학생이 많았다(61%). 본 연구의 실험 집단은 C나 스크래치를 배우고 온 예비 교사이지만, 앱 인벤터를 배우는 데 어려움을 느끼는 경우가 많았다. 이와 같은 결과는 앱 인벤터가 어플리케이션을 제작하고 블록을 활용한다는 점은 다른 프로그래밍 언어에 비해 긍정적인 요소지만[18], 처음 프로그래밍 언어를 배우는 학생에게는 다른 블록형 프로그래밍 언어보다 어려움을 느낄만한 요소가 많다는 점을 시사한다. 기존의 연구에서는 초, 중학생이 앱 인벤터를 활용한 프로그래밍 수업에 대하여 긍정적이라고 응답하였다[18,23,24] 이와 같은 연구에서는 학생들이 앱 인벤터의 장점인 어플리케이션을 직접 만들어보는 것이 아니라 단순히 rote skill을 배양하기 위한 수업이 이루어졌기 때문이라고 생각된다. 이에 따라 기존의 연구에서 제안한 교육 방안보다 더 세세한 고려가 필요하다고 생각된다[18]. 학생들이 느끼는 앱 인벤터의 단점과 어려움을 조사하고, 앱 인벤터를 정보 교육에 활용하기 위한 교육 방안에 대해서 연구가 필요하다. 교육 실습을 수행한 예비 교사의 응답을 살펴보면, 기존과 다르게 강의식과 프로젝트 방식을 혼합한 방법을 가장 선호하는 것으로 나타났다. 컴포넌트를 학습하고, 교육 실습을 통하여 예비 교사들은 앱 인벤터를 효과적으로 가르치기 위해서는 컴포넌트를 배우고

학생들이 자유롭게 설계해보는 프로젝트 수업이 필요하다고 응답하였다. 자신의 어플리케이션을 개발하는 프로젝트가 끝난 뒤에 예비 교사가 원하는 수업 방식은 큰 변화가 나타나지 않았다. 또한, 앱 인벤터를 활용한 교수-학습 프로그램 개발을 위하여 초·중등 교육에 다른 활용 방안이 연구되어야 한다고 생각된다. 강의식과 프로젝트 수업에 대한 선호가 일부 줄고, 강의식과 실습수업과 기타 형식에 대한 선호가 증가하였다. 기타 의견에서는 일부 예비 교사는 플립러닝을 도입하고, 수업 시간에는 오로지 학생들이 만들고 싶은 어플리케이션을 설계 및 구현하는 프로젝트 수업이 필요하다는 의견이 있었다, See Table 15.

마지막으로 교사가 된다면 앱 인벤터를 활용할 것인지에 대한 인식을 조사한 결과이다. 컴포넌트 학습을 한 이후에 활용하겠다는 예비 교사(74%)가 활용하지 않겠다는 예비 교사(26%)보다 훨씬 많은 것으로 나타났다. 활용하겠다는 의견을 살펴보면, 65%의 예비 교사가 학생들이 흥미를 느낄 수 있고 개발의 경험을 제공할 수 있다는 점 때문에 앱 인벤터를 수업에 활용하고 싶다고 말하였다. 9%의 학생은 수업에 활용은 하겠지만, 앱 인벤터의 특성 때문에 일부 상황에서만 활용하겠다고 응답하였다. 교육 실습을 경험한 이후에 예비 교사의 수업 활용 여부를 살펴보면, 이전과 똑같지만, 수업에 활용하겠다는 예비 교사들 사이에서 흥미와 개발, 성취감을 목적으로 앱 인벤터를 활용하겠다는 예비 교사는 65%에서 57%로 감소하였으며, 학생의 수준을 고려하고 방과 후나 동아리, 제어문을 가르칠 때와 같이 선택적인 상황에서 가르치겠다는 예비 교사가 9%에서 17%로 증가하였다. 이와 같은 결과를 통하여 교육 실습이 실제 학교 현장의 환경과 학생들의 수준을 예비 교사 이해할 수 있는 기회를 제공하고, 앱 인벤터의 수업 활용에 대한 인식 변화를 유도하였다고 생각된다. 마지막으로 어플리케이션 제작 프로젝트 수행한 이후에 수업 활용 여부를 살펴보면, 활용하겠다는 의견이 절반 이하로 떨어졌고(43%), 활용하겠다는 예비 교사의 비율이 상승하였다(57%). 예비 교사들은 스마트 기기가 꼭 필요하다는 점과 블록 기반 프로그래밍 언어가 가진 한계점, 기존의 블록 프로그래밍 언어보다 어렵다는 점, 실제 학교 현장에서 가르치는 교육과정과의 관련성이 적다는 점 때문에 활용하지 않겠다는 예비 교사가 급증하였다. 예비 교사는 앱 인벤터

Table 14. The negative perception on pre-service teacher of App inventor through a comparison

Area	Domain	S1	S2	S3
Program min g course	Functional limits	4 (15%)	0 (0%)	0 (0%)
	Build of application	3 (12%)	0 (0%)	2 (7%)
	Error	0 (0%)	3 (13%)	3 (11%)
	etc.	6 (23%)	7 (30%)	11 (39%)
	None	13 (50%)	13 (57%)	12 (43%)
Total		26 (100%)	23 (100%)	28 (100%)
Program min g Language	Functional limits	9 (28%)	8 (25%)	20 (47%)
	Build of application	6 (19%)	8 (25%)	5 (12%)
	Block based language	4 (13%)	7 (22%)	4 (9%)
	Error	4 (13%)	1 (3%)	6 (14%)
	etc.	4 (13%)	2 (6%)	7 (16%)
	None	5 (16%)	6 (19%)	1 (2%)
Total		32 (100%)	32 (100%)	43 (100%)

Table 15. The perception on pre-service teacher of instructional methods for programming education using App inventor

		S1	S2	S3
Teaching meth od	Teacher-cent ered learning	6 (26%)	3 (13%)	3 (13%)
	Teacher-cent ered learning & Practice	14 (61%)	3 (13%)	4 (17%)
	Teacher-cent ered learning & Project	0 (0%)	13 (57%)	10 (43%)
	etc.	3 (13%)	4 (17%)	6 (26%)
	total	23 (100%)	23 (100%)	23 (100%)

를 활용한 프로그래밍 수업에 대해서 어플리케이션을 개발할 수 있다는 점과 블록 기반 프로그래밍 언어라는 점 때문에 수업에 활용하겠다는 의견이 많았지만, 학교 현장에서 정보 교육을 경험하고, 앱 인벤터를 활용한 프로젝트 학습을 한 이후에 앱 인벤터를 프로그래밍 교육에 활용하지 않겠다는 의견이 증가하였다, See Table 16.

이와 같은 결과는 앱 인벤터를 활용한 프로젝트 학습은 앱 인벤터에 대한 한계점과 부정적인 인식을 증가시키며, 학교 현장에서 앱 인벤터의 활용을 감소시킬 수 있다는 것을 의미한다.

앱 인벤터에 대한 인식 조사 연구를 종합하면, 예비 교사들이 컴포넌트 학습을 실시할 때는 앱 인벤터가 블록 기반 프로그래밍 언어이기 때문에 쉽게 접근하고 프로그래밍 할 수 있다는 점과 어플리케이션이라는 결과물을 직접 만들 수 있다는 점 때문에 앱 인벤터를 긍정적으로 생각했다는 것을 확인할 수 있었다. 하지만, 교육 실습과 프로젝트 활동을 거치면서 앱 인벤터가 가진 기능적 한계, 빌드의 필요성, 블록의 단점, 오류 때문에 앱 인벤터에 대한 부정적인 인식이 증가하였다고 생각된다.

이러한 부정적인 인식은 앱 인벤터가 가진 긍정적인 인식을 감소시키고 앱 인벤터에 대하여 부정적으로 판단하는 요소로 작용했다고 생각된다. 이에 따라 예비 교사들은 앱 인벤터를 활용한 문제 해결에 어려움을 느끼게 되었고, 예비 교사의 문제 해결 능력 향상에 부정적인 영향을 주었다고 생각할 수 있다. 따라서 다른 프로그래밍 언어와 다르게 앱 인벤터가 대응 표본 t-검정에서 통계적으로 유의미한 차이가 나타나지 않은 이유로 앱 인벤터가 가진 특성이 크게 작용했을 것이라고 생각된다.

자아 효능감에서도 문제 해결 과정에서 발생하는 오류와 문제점들이 자신의 능력에 대한 신념과 관련된 자신감과 목표에 대한 선호와 난이도를 설정과 관련된 과제 난이도 영역에 영향을 주었을 것이라고 생각된다[25,26].

이와 같이 앱 인벤터를 교육하는 과정에서도 앱 인벤터에 대한 인식이 변화하고, 긍정적인 인식을 부정적으로 바뀌는 것을 확인할 수 있었다[33,34]. 이러한 인식은 향후에 앱 인벤터에 대한 태도에 영향을 줄 수 있으며, 앱 인벤터를 활용한 교육에도 영향을 미칠 수 있다. 이에 따라 학생들을 대상으로 앱 인벤터 교육을 실시할 때 다양한 요소를 고려해야 할 뿐만 아니라 예비 교사를 대상으로 교육을 실시할 때도 프로그래밍 언어에 대한 요소의 고려가 필요하다[18].

Table 16. The result of whether or not to use the application of app inventor in class

		S1	S2	S3
Appl icati on	Yes	17 (74%)	17 (74%)	10 (43%)
	No	6 (26%)	6 (26%)	13 (57%)
total		23 (100%)	23 (100%)	23 (100%)

V. Conclusions

본 연구에서는 앱 인벤터를 활용한 프로그래밍 교육을 예비 교사에게 실시하여, 예비 교사의 자아 효능감과 문제 해결 능력의 변화를 살펴보았다. 또한, 설문과 인터뷰를 통하여 예비 교사가 가진 앱 인벤터에 대한 장, 단점과 수업 방식 및 활용 여부를 조사하였다. 이와 같은 연구를 통해 얻은 결론은 다음과 같다.

ICT 교육을 받은 예비 교사의 문제 해결 능력과 자아 효능감과 관련이 없었다. ICT 교육을 받은 예비 교사들은 사전-사후로 문제 해결 능력과 자아 효능감이 통계적으로 유의미한 차이가 나타나지 않았다. 앱 인벤터를 활용한 프로그래밍 교육을 받아도 예비 교사의 문제 해결 능력과 자아 효능감은 통계적으로 유의미한 변화는 관찰되지 않았다. 사전 검사에 비해 사후 검사의 값은 증가하였지만, 그 차이는 통계적으로 유의미하지 않았다. 따라서 사후 검사에 나타난 차이는 두 집단 간의 차이가 커짐에 따라서 나타난 현상이라는 것을 확인할 수 있었다.

앱 인벤터에 대한 예비 교사의 인식을 조사한 결과, 예비 교사들은 앱 인벤터가 블록형 프로그래밍 언어여서 학습자에게 흥미를 유발하고, 어플리케이션을 제작할 수 있다는 점이 장점이라고 응답하였다. 반대로 어플리케이션을 만들기 위해 많은 블록이 필요하고, 앱 인벤터가 가진 기능적 한계, 결과물을 확인하기 위해 빌드를 꼭 해야 한다는 점, 예기치 못한 오류가 발생한다는 점이 단점이라고 응답하였다.

앱 인벤터를 활용한 프로그래밍 수업 형식에서 예비 교사는 강의가 진행됨에 따라 프로젝트 학습에 대한 선호가 증가하였다. 앱 인벤터는 다른 블록형 프로그래밍 언어와 다르게 컴포넌트 설정 부분이 더 힘들기 때문에 강의식 수업에 대한 선호가 높았지만, 교육 실습과 프로젝트 학습 이후에 강의식과 실습보다는 프로젝트형 수업에 대한 선호가 증가하였다. 또한, 프로젝트형 수업을 위하여 플립러닝과 같은 교수 방식을 응답한 예비 교사도 있었다.

마지막으로 앱 인벤터를 수업에 활용하겠다는 문항에서는 강의가 진행됨에 따라 앱 인벤터를 활용한다고 응답한 예비 교사의 수는 감소하였다. 앱 인벤터가 가진 장점 때문에 초기에는 수업에 앱 인벤터를 활용하겠다는 예비 교사가 많았지만, 교육 실습과 프로젝트 수행 과정에서 느끼는 앱 인벤터의 한계와 문제점 때문에 앱 인벤터를 수업에 활용하겠다는 예비 교사는 감소하였다.

2015 개정 교육과정의 도입에 따라서 다양한 프로그래밍 언어와 피지컬 컴퓨팅 기기를 활용한 연구가 진행될 것이다. 따라서 스크래치, 엔트리와 같은 프로그래밍 언어를 대상으로 하는 연구도 필요하며, 프로그래밍 언어 간의 비교 연구도 필요하다고 생각된다. 이러한 연구는 향후에 프로그래밍 교육에서 환경에 따라 적절한 프로그래밍 언어 선택을 촉진할 수 있으며, 더욱더 효과적인 프로그래밍 교육을 위한 교수-학습 방안을 수립

하는데 기초 연구로 활용될 수 있다. 본 연구에서는 앱 인벤터를 활용한 프로그래밍 교육을 예비 교사에게 실시하였다. 예비 교사가 향후에 학교 현장에서 정보 교육을 실시할 주체이지만, 현재 정보 교육을 실시하고 있는 교사도 있다. 따라서 교사를 대상으로 하는 연구를 진행하여 교사의 인식이나 효과도 분석하고, 교육을 받는 주체인 학생을 대상으로 하는 연구도 필요하다. 이러한 학생, 예비 교사, 교사를 대상으로 하는 연구와 다양한 프로그래밍 언어를 대상으로 한 연구가 종합되었을 때, 교육 과정을 효과적으로 전달하기 위한 교수-학습 방안의 수립이 이루어질 수 있다. 따라서 본 연구는 프로그래밍 교육에서 기초 연구로써 향후에 다양한 연구로 분화되기 위한 시작점으로 활용될 수 있다.

REFERENCES

- [1] Hyundai Research Institute, "The main contents and implications of the 2016 World Economic Forum (Volume 16-2)," Retrieved from https://www.designdb.com/Share/fileDownload_1.asp?dCate=ContentsInfo&dID=PKID&pID=22765. 2016.
- [2] R. Kemp, "Fourth industrial revolution," *The Lawyer*, Vol. 31, No. 21, pp. 12, 2016.
- [3] Computer Science Teachers Association, "CSTA K-12 Computer Science Standards," Retrieve from http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf. 2011
- [4] K12CS, "A Framework for K-12 Computer Science Education," Retrieved from <http://k12cs.org>. 2016.
- [5] Department for Education in U. K., "The national curriculum in England: Framework document," Retrieved from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/210969/NC_framework_document_-_FINAL.pdf. 2013.
- [6] skshin & ykbae, "Review of Software Education based on the Coding in Finland," *Journal of The Korea Association of Information Education*, Vol. 19, No. 1, pp. 127-138, 2015.
- [7] hylee, "Japan, Compulsory of Computer Programming Education in Elementary school Curriculum," Retrieved from <http://www.yonhapnews.co.kr/bulletin/2016/04/21/020000000AKR20160421080600009.HTML?input=1195> m. 2016.
- [8] swKim & yjLee, "Development of a Software Education Curriculum for Secondary Schools," *Journal of the Korea society of computer and information*, Vol. 21, No. 8, pp. 127-141, 2016.
- [9] ysjeong. "Needs Analysis of Software Education Curriculum at National Universities of Education for the 2015 Revised National Curriculum," *Journal of The Korea Association of Information Education*, Vol. 20, No. 1, pp. 83-92, 2016.
- [10] jwchoi, sjan, & yjlee, "Computing education in Korea—current issues and endeavors," *ACM Transactions on Computing Education (TOCE)*, Vol. 15, No. 2, pp. 1-21, 2015.
- [11] Ministry of education, "Middle School Curriculum, Proclamation of the Ministry of education: #2015-74 [Seperate Volume 3]," 2015.
- [12] sjAn & yjlee, "Educational Objectives in Computing Education," *Journal of the Korea society of computer and information*, Vol. 21, No. 1, pp. 181-189, 2016.
- [13] jwchoi & yjlee, "Educational Application of Puzzles for Algorithm Learning of Informatics Gifted Elementary School Students," *Journal of the Korea society of computer and information*, Vol. 20, No. 5, pp. 151-159, 2015.
- [14] skJeon & yjLee, "Art based STEAM Education Program using EPL," *Journal of the Korea society of computer and information*, Vol. 19, No. 4, pp. 149-158, 2014.
- [15] gwYang, "The Effect of Scratch on Learning Motivation and Academic Achievement for Programming Education," *Journal of The Korean Association of Information Education*, Vol. 14, No. 4, pp. 547-553, 2010.
- [16] chLee, "Change of Perceptions and Images of Pre-Service Elementary School Teachers on Programming according to the Experiences of Programming Learning," *Journal of Korean practical arts education*, Vol. 29, No. 3, pp. 1-19, 2016.
- [17] kjMun, jyMun, smKim, & swKim, "Application of Programming Curriculum for Pre-service Science Teacher and Examination of their Perceptions about Programming," *Journal of Learner-Centered Curriculum and Instruction*, Vol. 16, No 10, pp. 825-842, 2016.
- [18] sjAn, & yjLee, "Elementary and Secondary Programming Education Plan Using App Inventor," *The Journal of Korean association of computer*

- education, Vol. 17, No. 5, pp. 79-88, 2014.
- [19] kwHan, "A case of project education using arduino and app inventor in Specialized commercial High School," Journal of The Korean Association of Information Education, Vol. 22 No. 2, pp. 7-14, 2014.
- [20] ihYoo, "Design a Programming Education Plan for SW Education Using Robot and Mobile Application Development Tool," Journal of The Korean Association of Information Education, Vol. 18, No. 4, pp 615-624, 2014.
- [21] hkKim, "Android App. Implementation Teaching using App. Inventor for Elementary school students," Journal of Korea Multimedia Society, Vol. 16, No. 12, pp. 1495-1507, 2013.
- [22] mgSeol & ciSon, "A Study on Development of Teaching Materials for App Inventor Programming Using the Waterfall Model," Journal of The Korean Association of Information Education, Vol. 17, No. 4, pp 409-419, 2013.
- [23] sjAn, & yjLee, "Designing Programming Curriculum for Developing Programming Pedagogical Content Knowledge of Pre-service Informatics Teachers," The Journal of Korean association of computer education, Vol. 19, No. 2, pp.1-10, 2016.
- [24] yjKim & hsChung, "A Study on the App Inventor 2 for Utilization of Programming Basics Educational Tool in Primary and Secondary Education," Journal of KIIT, Vol. 14, No. 3, pp. 163-173, 2016.
- [25] ayKim & jeCha, "Self-efficacy and measurement," The Proceeding of Korean Society for industrial and Organizational Psychology, 51-64, 1996
- [26] ayKim, "A study on the academic failure-tolerance and its correlates," Journal of Educational Psychology, Vol. 11, No. 2, pp. 1-19, 1997.
- [27] A. Bandura, "Self-efficacy: toward a unifying theory of behavioral change," Psychological review, Vol. 84, No. 2, 1977
- [28] A. Bandura, "Social foundations of thought and action: A social cognitive theory" Prentice-Hall, Inc.
- [29] sjLee, ykChang, hnLee, & kyPark, "A Study on the Development of Life-Skills: Communication, Problem Solving, and Self-Directed Learning," Korea Educational Development Institute (RR 2003-15-3).
- [30] cmYang, "Meta-Analysis on the Effects of Programming Education using Educational Programming Languages," Journal of The Korean Association of Information Education, Vol. 18, No. 2, pp 317-324, 2014a
- [31] cmYang, "Comparison of the Effects of Robotics Education to Programming Education Using Meta-Analysis," Journal of The Korean Association of Information Education, Vol. 18, No. 3, pp 413-422, 2014b.
- [32] yhJin & ysKim, "A Meta-Analysis on the Effects of Educational Programming Language," The Journal of Korean association of computer education, Vol 14, No. 3, pp. 25-36, 2011.
- [33] B. Larrivee, & L. Cook, "Mainstreaming: A study of the variables affecting teacher attitude," The Journal of Special Education, Vol. 13, No. 3, pp. 315-324, 1979.
- [34] T. R. Koballa, "Attitude and related concepts in science education," Science education, Vol. 72, No. 2, pp. 115-126, 1988.

Authors



Seong-Won Kim received the B.S. degree in Computer Education from Korea National University of Education, Korea in 2013. He received the M.S. degree in Biology Education from Seoul National University in 2015. Mr. Kim is currently a doctoral course student in the Department of Computer Education, Korea National University of Education. He is interested in informatics education, robot programming education, STEAM education, and TPACK.



Youngjun Lee received the B.S. degree in Computer Science from Korea University, Korea, in 1988. He received the Ph.D. degree in Computer Science from the University of Minnesota, Minneapolis, in 1994. He is currently a Professor in the Department of Computer Education, Korea National University of Education. His research interests include intelligent system, learning science, informatics education, technology & engineering education.