

창작로봇을 이용한 비전공자의 소프트웨어 융합 교육에 관한 연구

구진희
목원대학교 정보통신융합공학부

A Study on the Software Convergence Education for Non-Majors Computer Science using Creative Robot

Jin-Hee Ku

Department of Information Communication Engineering, Mokwon University

요약 4차 산업혁명의 도래로 우리 삶의 방식과 산업 구조가 IT 기반 자동화를 넘어 지능화 단계로 진화하면서 다양한 분야에서 소프트웨어 역량을 갖춘 인재를 요구하고 있다. 이러한 요구를 반영하여 최근 초중등뿐만 아니라 대학에서도 비전공자에 대한 소프트웨어 기초 교육을 강화하고 있다. 본 연구에서는 관련 연구를 통해 비전공자의 소프트웨어 교육 목표를 컴퓨팅 사고 기반의 일반적인 문제해결능력과 자신의 학문 영역에서 소프트웨어 융합 역량을 향상시키고자 함에 두고, 이러한 목표를 달성하고자 창작로봇을 이용한 비전공자의 소프트웨어 융합 교육에 대한 방안을 제안하였다. 다양한 전공의 학생들로 구성되는 핵심교양 교과목 수강생 91명을 대상으로 창작로봇 과정과 컴퓨팅 사고 및 융합 요소와의 관련성에 기초하여 수업을 설계하고 13주 동안 연구를 수행하였다. 창작로봇을 통한 소프트웨어 융합 교육의 효과를 알아보기 위해, 수업 전 사전 진단 조사를 토대로 수업 후 학생들의 학습 성과, 창작로봇 활동 과정에 대한 만족도, 타 학문에 대한 인식의 변화를 관찰하였다. 연구의 결과, 학생들은 학습내용의 이해도, 다 학제적 협력학습에 대한 전반적인 만족도, 타 학문에 대한 이해도, 창작로봇 활동을 통한 문제해결능력에 대한 자기 평가 등 12개 문항에서 수업 전과 비교하여 자신의 능력이 향상되었다고 응답하였다.

Abstract In the age of the Fourth Industrial Revolution, lifestyle and industrial structures are faced with evolution from IT-based automation to the intelligent stage, demanding talents with software capabilities in various fields. Reflecting these demands, the government has enhanced basic software education for non-majors in elementary and secondary schools as well as universities. In this study, the software convergence education of Non-Majors is proposed to improve the general problem solving ability based on computational thinking and the software convergence ability in the field of their own by developing robot activity. The subjects of this study were 91 students, who were composed of various majors. The class was designed with computing thinking, convergence elements, and creative robot activity. The study was conducted for 13 weeks. To examine the effects of software convergence education through the creative robot activity, this study observed changes in the students' learning outcomes, satisfaction with creative robot activities, and perceptions of other disciplines after class based on pre-diagnosis surveys. The survey asked 12 questions including an understanding of the learning contents, overall satisfaction with multidisciplinary collaborative learning, understanding of other disciplines, and self-evaluation of problem solving ability through creative robot activities, which were compared with that before the class. They answered that their ability was improved.

Keywords : Computational Thinking, Creative Robot, Non-Majors Computer Science, Problem Solving Ability, Software Convergence Education, The Fourth Industrial Revolution

*Corresponding Author : Jin-Hee Ku(Mokwon Univ.)

Tel: +82-42-829-7647 email: jhku@mokwon.ac.kr

Received January 2, 2017

Revised February 2, 2017

Accepted February 3, 2017

Published February 28, 2017

1. 서론

전 세계가 4차 산업혁명에 대한 본격적인 준비에 들어가면서 산업구조 및 사회는 물론이고 우리 삶의 방식에 엄청난 변화를 예고하고 있다. 세계경제포럼에 따르면 인공지능과 첨단기술로 대변되는 4차 산업혁명의 영향으로 2020년까지 선진국에서만 약 710만개의 일자리가 사라질 것으로 예측하였다. 또 현재 초등학교에 입학한 어린이의 약 65%는 기존에 존재하지 않았던 새로운 직업에 종사하게 될 것으로 전망하였다[1].

인공지능, 사물인터넷, 지능형 로봇, 빅데이터 등 혁신 기술을 이용한 지능형 공장 및 초연결 사회로의 진화를 의미하는 4차 산업혁명 시대에는 글로벌 경쟁력으로 소프트웨어가 강조된다[2]. 이는 모든 혁신 기술들이 ICT를 바탕으로 한 소프트웨어를 통해 구현되기 때문이며, 따라서 다양한 분야에서 소프트웨어 융합 역량이 매우 중요하게 된다.

이러한 전 세계적 흐름 속에 최근 정부에서는 초중등뿐만 아니라 대학까지 연계되는 소프트웨어 교육을 강조하며 2018년부터 적용하는 중등 정보 교육과정에 이를 반영하였고, 대학에서도 소프트웨어 중심 대학을 선정하여 전공, 비전공 학생들에게 차별화된 소프트웨어 교육을 추진하고 있다[3, 4].

초중등 정보 교육의 방향은 컴퓨터과학의 원리와 컴퓨팅 사고력 신장에 두고 있으며[5], 같은 맥락으로 대학의 소프트웨어 중심대학의 교육목표 또한 보편적 교육으로서의 소프트웨어 교육을 강조하며 컴퓨터과학에 대한 기본 개념과 원리, 컴퓨팅 사고에 기반을 두고 있다. 컴퓨터과학(Computer Science)은 전산(computational) 및 그 응용기술에 대한 과학적이고 실제적인 접근을 연구하는 학문[6]으로 이의 중심에는 컴퓨팅 사고가 있다. 컴퓨팅 사고는 컴퓨터 과학자뿐만이 아니라 누구나 갖춰야 하는 기본적인 역량으로 읽기, 쓰기, 셈하기와 같이 인간의 가장 기본적인 역량 항목으로 강조되고 있으며[7], 오늘날에는 모든 분야에 걸쳐 일반적인 문제 해결에 필요한 능력으로 인식되고 있다.

특히, 컴퓨팅 기술이 예술, 인문학, 사회 과학 등 거의 모든 학문 분야에 걸쳐 융합되는 현재 상황에서 소프트웨어 교육은 컴퓨팅 사고를 기반으로 문제해결 능력을 키울 수 있도록 하고 각 학문 영역과 융합할 수 있도록 교육과정이 운영될 필요성이 있다[8].

세계적으로 급성장하는 기업 가치 1조원 이상으로 평가받은 비상장 스타트업(일명 유니콘 기업) 기업 창업자와 CEO 140명을 분석해 본 결과 전체의 30%가 공학과 인문·사회과학을 병행해서 학습한 융합형 인재인 것으로 나타났다[9].

궁극적으로 코딩 교육이나 프로그래밍 자체가 목적이 아니라 소프트웨어 교육은 다양한 분야에서 역량을 발휘할 수 있도록 하는 수단이 되어야 할 것이다. 따라서 본 연구에서는 창작로봇을 이용한 소프트웨어 융합 역량을 키울 수 있는 교육과정을 개발 및 적용하고 그 결과를 분석하고자 한다.

2. 본론

2.1 비전공자의 소프트웨어 교육

소프트웨어는 일반적으로 컴퓨터 및 관련 장치를 작동시키는 데 사용되는 다양한 종류의 프로그램을 말한다[10]. National Academy of Sciences는 모든 대학생들이 정보 기술에 능통해야 함을 보장하는 프로그램을 배워야 한다고 주장했다[11]. 이 보고서에 따르면 급변하는 정보 기술에 학생들이 그들의 직장 및 일상생활에서 생산적으로 대처할 수 있도록 하기 위해 필요한 지식으로 다음 세 가지 유형을 제시하였다.

현대 기술: 다양한 컴퓨터 응용 프로그램을 사용할 수 있는 능력

기초 개념: 컴퓨터 과학의 기초를 형성하는 컴퓨팅의 기본 원리 및 개념

지적 능력: 특정 상황에서 정보 기술을 적용하고 이 기술을 사용하여 새로운 문제를 해결하는 능력

Urban-Lurain과 Weinsank는 컴퓨터과학 비전공자에게 있어서 프로그래밍의 역할을 보면, 학습 이론 문헌에 기초한 컴퓨팅에 대한 개념적 이해가 프로그램을 배우지 않고도 습득 될 수 있다고 주장하였다[12]. Carlisle, Wilson, Humphries와 Hadfield는 또한 비전공자 프로그램 학습은 구문을 학습하기 전에 플로우차트의 알고리즘 시각화를 통해 문제해결을 먼저 학습해야 함을 설명한다[13]. 이러한 연구들을 보면, 정보 기술을 이해하는데 있어서 프로그래밍의 역할을 두고 컴퓨터 과학자들 사이에서 많은 논쟁의 주제였음을 알 수 있다.

Soloway는 프로그램 학습이 범용 문제 해결 및 사고

기술을 가르치는 것부터 학생들이 컴퓨터가 작동하는 방식을 이해하고 이해할 수 있도록 돕는 것까지 다양한 이점이 있다고 주장하였다[14]. 같은 맥락에서 Walker의 연구를 보면, 컴퓨터과학 비전공 과정은 학생들이 컴퓨팅 사고에 있어 자신의 기술을 향상시키는 데 도움이 되도록 노력한 다음, 솔루션을 정확하게 작성할 수 있어야 하며, 솔루션의 정확성을 분석하고 대안 솔루션을 비교할 수 있어야 한다고 말한다[15].

이상 살펴본 바와 같이 비전공자의 소프트웨어 교육은 단지 프로그램 학습이 목적이 아니라 프로그래밍 기술을 통해 다양한 상황에서 문제를 해결할 수 있도록 한 것임을 알 수 있다.

2.2 비전공자를 위한 컴퓨터과학 교육과정

소프트웨어 중심 대학은 산업현장의 요구를 반영하고 대학 소프트웨어 교육을 혁신하기 위해 2015년 8개 대학, 2016년 6개 대학이 선정되어 운영되고 있다[16]. 또한, 최근 소프트웨어 융합 인력 수요가 크게 증가함에 따라 소프트웨어 중심 대학을 포함한 많은 대학에서 비전

공자에 대한 소프트웨어 기초 교육을 강화하고 있다.

Table 1은 소프트웨어 중심 대학의 비전공자 소프트웨어 기초 교육을 위한 교육과정을 나타낸 것이다[4].

Table 1의 소프트웨어 기초 교육을 위한 필수 이수 교과목을 보면, 거의 모든 학교의 교육과정에서 컴퓨팅 사고와 프로그래밍을 통한 문제해결 관련 과목이 포함되어 있음을 알 수 있다[4, 16].

이와 같이 소프트웨어 기초 교육을 대학의 신입생 전 체로 확대하고 소프트웨어 융합 전공을 강화하는 배경으로는 인문·경영 등 타 전공과 소프트웨어의 새로운 융합 교육을 통한 지능정보사회를 주도할 다양한 분야의 소프트웨어 역량을 겸비한 융합인재 양성에 있다고 볼 수 있다[16, 17].

해외의 대학은 소프트웨어 교육보다는 그것이 속하는 학문분야에서의 비전공자를 위한 코스를 운영하고 있다. 다음 Table 2는 워싱턴 대학의 비전공자를 위한 입문 교육과정을 나타낸 것이다[18].

Table 3은 보스턴 대학의 비전공자를 위한 컴퓨터과학 코스로 다음과 같이 운영하고 있다[19].

Table 1. Software center university's non-majors software basic education

Univ.	SW Fundamentals Course	Description
Gachon	Computational Thinking and Coding	1 course or more required, SW basic and SW convergence education
Kyungpook	SW Problem Solving Basics, SW courses by college	SW basic education (3-6 credits required)
Korea	Computational Thinking	Scratch, Python, HomePage, Physical Computing, etc.
Sogang	Computational Thinking(3), Courses by college(3)	All students up to 6 credits SW education with basic literacy
Sungkyungkwan	Computational Thinking and SW Coding(2), Problem Solving and Algorithm(2)	All students SW basic course liberal arts required
Sejong	SW Basic Coding, Introduction to Programming	Programming education for all of the non-major (C, Python, etc.)
Ajou	C Programming and Matlab Python Excel, R, etc.	Required 3 credits for each college + 3 credits for each department
Chungnam	Computational Thinking, Programming Fundamentals	Strengthening SW courses for all students
Kookmin	Computer Programming I·II	Excel, Scratch, Python, etc.
Dongguk	Computational Thinking (before enrollment), SW and Future Society	Overall trend of IT SW convergence technology
Busan	Computational Thinking, Computer Programming	6 hours 3 credits required, SW understanding & problem solving, SW production
SeoulWomen's	Software and creative thinking	SW education for each major specialty (6 courses)
Kaist	Basics of Programming	Required for all students, strengthening software convergence capabilities of non-majors
Hanyang	Creative Computing, Creative Programming	Computational thinking, Problem solving by programming, Computer Science Fundamentals, Advanced Software Skills

Table 2. Introductory non-major courses

Courses	Course Overview
Computer Science Principles	Introduces fundamental concepts of computer science and computational thinking. Includes logical reasoning, problem solving, data representation, abstraction
Computer Programming I	Basic programming-in-the-small abilities and concepts
Computer Programming II	Concepts of data abstraction and encapsulation including stacks, queues, linked lists, binary trees, recursion, instruction to complexity and use of predefined collection classes.
Web Programming	Covers languages, tools, and techniques for developing interactive and dynamic web pages
Data Programming	Assignments solve real data manipulation tasks from science, engineering, business, and the humanities.

Table 3. Boston univ. non-major courses

Courses	Course Overview
Introduction to Computing	learning a deeper appreciation of the capabilities and limitations of computing(the Principles of Computation, Programming, Algorithm
Introduction to Internet Technologies and Web Programming	learn how modern Web technologies like search exploit fundamental aspects of networks, basics of Web programming
Introduction to Databases and Data Mining	learn to relational databases and the SQL, the writing of programs to analyze data, the data visualization, and data-mining techniques
Computational Systems	learning the common thread of computing that runs through evolution, molecular biology, personal computers, behavior, and psychology
Introduction to Applications Programming	what is computer software, and how is it developed? develop problem-solving skills and ways of thinking
The Art and Science of Quantitative Reasoning	mathematics, statistics, and computer science for predicting the weather, or controlling disease outbreaks

Table 2와 Table 3을 살펴보면, 비전공자의 입문 교육 과정으로 컴퓨터과학(컴퓨팅)의 기본 원리 및 컴퓨팅 사고와 프로그래밍, 다양한 학문 분야에서의 응용 등이 공통적으로 구성되어 있음을 알 수 있다. 따라서 비전공자 소프트웨어 교육의 방향은 프로그래밍과 함께 자신의 분야에서 문제해결을 위한 기본적인 사고 방법을 학습할 수 있도록 해야 한다.

2.3 창작로봇을 이용한 소프트웨어 융합

창작로봇은 자신의 상상과 설계에 따라 로봇키트에서 구할 수 있는 주어진 부품, 자신의 필요에 따라 스스로 설계해서 만들어야 하는 부품, 또는 주변에 쉽게 활용할 수 있는 재활용품이나 소품 등으로 만드는 로봇을 말한다[20].

이러한 일련의 창작로봇 과정은 컴퓨팅 사고 뿐만 아니라 프로그래밍 그리고 융합적 사고가 동시에 요구된다. Cuny, Snyder와 Wing 등도 “컴퓨팅 사고는 일반적으로 코딩 및 컴퓨터 프로그래밍과 관련이 있지만, 문제 해결, 시스템 설계 및 인간 행동 이해와 관련된 것 이상이다”라고 하였다[21].

Rusk, Resnick, Berg와 Pezalla-Granlund 등은 로봇 기술을 학생들에게 소개하기 위해 다양한 관심사와 학습 스타일을 가진 학생들을 참여시키기 위한 방법으로 다음 4가지 전략을 제안하였다. 1) 주제에 중점 두기(focusing on themes), 2) 예술과 공학을 결합하기(combining art and engineering), 3) 이야기를 격려하기(encouraging storytelling) 4) 전시회(시연회) 조직하기(organizing exhibitions). 이는 로봇에 여러 경로를 제공하는 것이 중요하다는 점을 강조하는 것이다. 창작 로봇은 입력을 기반으로 동작을 수행하는 모든 유형의 프로그램 가능한 기계를 설계하고, 프로그램 가능한 구조를 사용하여 자신만의 컴퓨터 제어 로봇을 구축하며, 자신들이 만든 창작 로봇을 시연해 보는 과정을 통해 다양한 배경의 학생들이 자연-공학-예술 분야를 체험할 수 있음을 보여주고 있다[22, 23].

Catlin와 Woollard는 그들의 연구에서 학생들이 아이 디어와 주변 세계를 탐구하기 위해 프로그램하고 사용할 수 있는 창작 로봇을 통해 컴퓨팅 사고력 및 문제해결 능력 향상에 어떻게 역할을 하는지 보여주고 있다[24]. 이러한 연구를 토대로, 본 연구에서는 창작로봇을 이용

하여 컴퓨팅 사고와 융합적 사고를 경험할 수 있도록 교육과정을 설계하고자 한다. 또한 학생들의 창작로봇 제작 활동 과정들이 컴퓨팅 사고의 각 개념과 어떤 관계가 있는지 그리고 어떤 융합 요소와 연관이 있는지를 제시하고자 한다.

3. 연구방법

Fig. 1은 본 연구의 연구방법 및 절차를 나타낸 것이다.

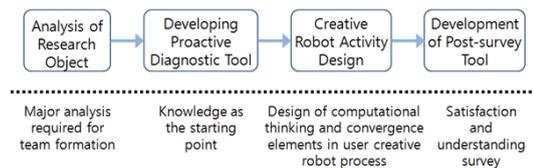


Fig. 1. Research method and procedures

3.1 연구 대상 분석

연구 대상 분석은 본 연구에서 중요한 단계이다. 창작로봇 과정을 통한 소프트웨어 융합 교육을 위해서는 다양한 학문 영역의 학생들로 그룹을 편성할 필요가 있다. 연구 기간은 2016년 9월 8일부터 12월 7일까지 13주 동안 진행되었다. 대학 핵심교양 교과목을 수강하는 학생들이 그 대상이며, 연구대상의 구체적 특징은 Table 4와 같다.

Table 4. Distribution of research subjects

	Spec.	Number of People	Percentage	Sum
gender	male	68	74.7	91
	female	23	25.3	
major/college	Humanities & Social Sciences	36	39.6	91
	Natural science and engineering	41	45.0	
	Arts and Physical Education	14	15.4	

3.2 사전 진단 도구 개발

수강생의 컴퓨터과학 기초 개념에 대한 지식이 어느 정도인지 파악하기 위해 수업 전 사전 설문조사를 실시하였다. 사전 설문조사의 가장 큰 목적은 교과목에 대한 학생들의 선수지식 진단이다. 또한 수업 후 학습 성과 진단의 목적을 위해 실시하였다.

3.3 창작로봇 과정 설계

창작로봇 과정은 다 학제적 협력학습 관점에서 설계되었다. 다양한 전공 또는 단과대학의 학생들로 그룹을 구성하여 서로 다른 관점과 능력을 가진 학생들 간의 상승효과를 유도하였다.

창작로봇 과정은 주제 선정(Idea / Storytelling), 기능설계(Engineering), 디자인(Art), 문제해결 및 프로그래밍(Mathematics / Technology), 발표 및 시연(Design/Art) 등 일련의 과정에서 자신의 전공과 장점을 살려 협력함으로써 결과물을 얻을 수 있도록 설계되었다. 창작로봇교육의 주요 목표는 통섭(Consilience)적 학습으로 STEM(Science, Technology, Engineering, Mathematics)교육을 위한 동기유발과 창의성 및 문제해결력의 증진이라고 할 수 있다[20]. 본 연구에서는 다 학제적 협력학습 기반의 아이디어 생성, 문제해결 방안 설계, 활동 구현, 시연 등 창작로봇 과정을 통해 소프트웨어 융합 교육을 하는데 그 목적이 있다.

4. 연구결과

4.1 창작로봇을 이용한 소프트웨어 융합 교육 과정

Table 5는 13주 동안의 수업 주제를 요약한 것이다. 창작로봇 과정은 다 학제적 협력 그룹 구성 및 스토리 추상화, 로봇을 이용한 하드웨어/통신/소프트웨어 기본 원리 학습, 창작로봇 구현, 발표 및 시연, 결과분석(체험 보고서)등으로 설계하였다.

Table 5. Lesson topic overview

Week	Lesson Topic Overview	Activity
1-5	IT core topics, learning basic concepts of computer science	Cognitive activity
6	Robot overview, discuss various topics related to robots	
7	Robot input and output devices	Experiential activity
9	Sensors, Motors, LED, LCD	
10	Robot Programming Practice	
11	Creative Robot Design, Story Generation, Idea Sketch for solving real-life problems of present and future	
12	Functional design and implementation testing Completion of story, design and function of creative robot	
13	Team presentation and demonstration	

Table 6은 창작로봇 과정에서 학생들의 활동과 컴퓨팅 사고 및 융합 요소와의 관계에 대해 나타낸 것이다. Table 6의 관계는 학생들이 창작로봇 과정을 수행하는데 있어서 가이드 및 평가 기준이 된다.

Table 6. Relationship of CT Concepts, Student Activity and Curriculum Subjects for the Creative Robot Activity.

CT Concepts	Student Activity	Related Subjects
Abstraction	What are the key features of our team's creative robot?	Science/Art
Algorithm	Defining what the creative robot will do	Mathematics
Decomposition	Detailed design drawing for creative robot	Design Technology
Generalization	How do creative robot behave in their environment	Science
Simulation	Simulate a model to create realistic effects.	Science/Art
Evaluation	Does my creative robot meet my design criteria?	Design Technology

4.2 수업 후 설문조사 결과

설문조사는 학습내용의 이해도, 다 학제적 협력학습에 대한 만족도, 타 학문에 대한 이해도, 창작로봇 과정을 통한 문제해결능력에 대해서 7점 척도로 자기 평가를 실시하였다. 설문조사 결과는 Table 7과 같다.

Table 7에서와 같이 학생들은 학습내용의 이해도, 다 학제적 협력학습에 대한 전반적인 만족도, 타 학문에 대한 이해도, 창작로봇 활동을 통한 문제해결능력에 대한 자기 평가 등 12개 문항에서 수업 전과 비교하여 자신의 능력이 향상되었다고 응답하였다.

Table 7. Survey of satisfaction and understanding through creative robot process

Question	Average
1) I carefully examined the lecture plan of this lecture and applied for enrollment.	6.01
2) Cooperative activities with team members of different majors were beneficial.	6.20
3) Our team has good communication and role sharing.	6.14
4) Interdisciplinary cooperative learning has increased understanding of the other major.	5.93
5) This course has helped me to understand the basic concepts of computer science.	6.10
6) I have improved my knowledge of IT compared to before the class.	6.00
7) The creative robot process helped to understand the concept of sensor / communication / hardware.	5.99

8) I think this course has improved my computational thinking ability.	5.85
9) The creative robot process helped to understand the software concept.	6.00
10) Through this lecture, I helped to connect the concept of software with my major.	6.02
11) I think that the creative robot process helps improve real life problem solving ability.	5.83
12) I would like to recommend this course to other students.	6.43

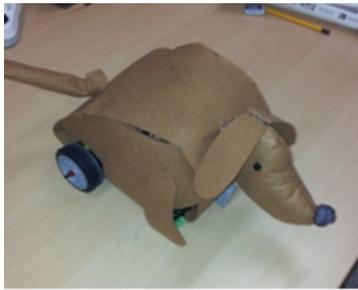
4.3 학습 성과에 대한 전문가 평가 및 분석

본 연구에서는 창작로봇을 활용한 소프트웨어 융합 교육의 학습 성과에 대한 평가의 객관성과 타당성을 위해서 인문학, 수학, 물리, 컴퓨터 공학, 컴퓨터과학교육 분야 5명의 전문가를 대상으로 학습 성과를 평가하였다. 전문가의 전공 및 평가 영역은 Table 8과 같다.

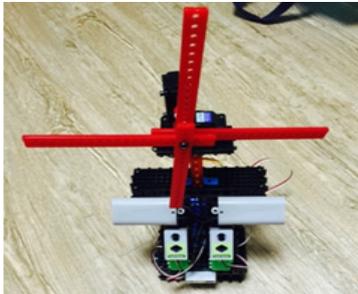
Table 8. Expert Evaluation Area

Evaluator Major	Main Evaluation Area
Korean Literature	Identify Team Ideas (Stories)
Mathematics	Implementing functions of robots
Physics	Details of robot motion (dynamics)
Computer Engineering	Robot programming logic
Computer Science Education	Presentations and Creativity

전문가 평가 결과를 반영하여 최종 학습 성과를 요약하면 다음과 같다. Fig. 2는 학생들의 창작로봇 구현 사례로 스토리 추상화와 디자인을 나타낸 것이다. Fig. 2의 (a)는 반려견 문제를 개선할 수 있는 인공지능 강아지 로봇으로 외롭다고 혹은 예쁘다고 강아지들을 분양 받고 막상 키워보니 힘들어서 강아지를 버리는 사람들이 늘어나고 있어 강아지를 키우기 전에 인공지능 강아지를 키워보고 결정할 수 있도록 한 것이다. 가장 어려운 배변과 끼니를 안 챙겨도 되고 독거노인, 어린아이도 키울 수 있는 인공지능 강아지 로봇을 구현한 것이다. Fig. 2의 (b)는 기존의 선풍기의 단점을 보완하여 음성과 리모컨으로 이동이 가능한 움직이는 선풍기로 고정되어 있는 기존 선풍기의 단점을 보완하여 리모컨과 음성인식으로 이동이 가능한 선풍기를 제작한 것이다. 무거운 선풍기를 움직이는 불편함을 덜고자 만든 이 로봇 선풍기는 편리함과 실용성을 갖춘 로봇을 구현한 것이다.



(a)



(b)

Fig. 2. Student learning outcomes, examples of creative robot: (a) Artificial Intelligence Puppies INGUK-e, (b) A Moving Electric Fan

5. 결론

본 논문은 소프트웨어 교육을 통해 궁극적으로 목표하는 교육적 효과가 무엇인지에 대한 연구에서부터 시작하였다. 관련 연구를 통해 오늘날 대학생들이 전공에 상관없이 누구나 4차 산업혁명의 생존도구로서 소프트웨어를 배워야 함을 전제하고 소프트웨어 교육이 목표가 아니라 수단으로서 소프트웨어를 통해 문제해결을 하고 각자 자신의 학문 영역에서 소프트웨어 역량을 융합할 수 있도록 하는 창작로봇 과정을 설계하고 적용하였다. 연구의 결과를 토대로 본 연구의 결론을 다음과 같이 요약하고자 한다.

첫째, 비전공자의 소프트웨어 교육은 전공심화의 교육목표와는 차별화 하여야 하며, 오늘날 보편적 지식으로 인식되고 있는 컴퓨팅 사고 기반의 문제해결능력 향상에 중점을 둘 필요가 있다.

둘째, 비전공자의 소프트웨어 교육은 자신의 학문분야의 스펙트럼을 넓히는데 그 목적을 두고, 각 학문영역과 융합할 수 있도록 교육과정이 운영될 필요성이 있다.

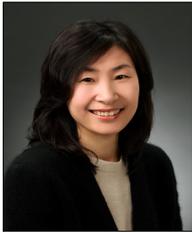
References

- [1] World Economic Forum, "The Future of Jobs-Employment, Skills and Workforce Strategy for the Fourth Industrial Revolution", January, 2016.
- [2] J. R. Yun, "4th Industrial Revolution and Soft Power", *TTA Journal*, Vol. 167, pp. 4-7, 2016.
- [3] Korea Institute for Curriculum and Evaluation, "Introduction to the revised curriculum in 2015", Research Report CRC 2015-28, 2015.
- [4] Software Oriented Society, Available From: <https://www.software.kr>(accessed Sep, 2016)
- [5] Ministry of Education, "Practical Arts (Technology & Home Economics) / Informatics Curriculum", Ministry of Education Notice 2015-74, 2015.
- [6] Wikipedia, <https://ko.wikipedia.org>(accessed Sep, 2016)
- [7] Wing JM, "Computational Thinking", *Communication of the ACM*, Vol. 49, No. 3, March, 2006. DOI: <https://doi.org/10.1145/1118178.1118215>
- [8] J. H. Ku, S. B. Shin, M. Y. Kim, "Instructional Design using App Inventor to Improve Computational Thinking of Non-majors Computing", *Proceedings of the Korea Academia-Industrial cooperation Society*, May, 2016.
- [9] Maeil Business News Korea, "Career Analysis of 140 Global Venture CEOs", Available From: <http://news.mk.co.kr/newsRead.php?year=2016&no=882284>(accessed Nov, 2016)
- [10] TechTarget, "What is software?", Available From: <http://searchsoa.techtarget.com/definition/software> (accessed Oct, 2016)
- [11] Committee on Information Technology Literacy, *Being fluent with information technology*, National Academy Press, Washington, DC, 1999.
- [12] Urban-Lurain M, Weinshank DJ., "Is There A Role for Programming in Non-Major Computer Science Courses?", *Frontiers in Education Conference*, FIE 2000 30th Annual, 2000. DOI: <https://doi.org/10.1109/fie.2000.897590>
- [13] Carlisle MC, Wilson TA, Humphries JW, Hadfield SM. "RAPTOR : Introducing Programming to Non-Majors with Flowcharts", *Journal of Computing Sciences in Colleges*, Vol. 19, Issue 4, pp. 52-60, April, 2004.
- [14] Soloway E., "Should we teach students to program?", *Communications of the ACM*, Vol. 36, No. 10, pp. 21-24, 1993. DOI: <https://doi.org/10.1145/163430.164061>
- [15] Walker HM., "Computational thinking in a non-majors CS course requires a programming component", *ACM New York, NY, USA*, Vol. 6, Issue 1, pp. 58-61, March, 2015. DOI: <https://doi.org/10.1145/2727126>
- [16] Ministry of Science, ICT and Future Planning, "Software oriented university promotion plan", 2015.
- [17] Ministry of Science, ICT and Future Planning, "Minister of Science, ICT and Future Planning-Software oriented university President Meeting Press release", October, 2016.

- [18] Computer Science & Engineering University of Washington, "Non-Major Course Options", http://www.cs.washington.edu/prospective_students/under_grad/admissions/nonmajor(accessed Nov, 2016)
- [19] Boston University Department of Computer Science, "CS for Non-Majors", Available From: <https://www.bu.edu/cs/courses/divisional-study-courses>(accessed Nov, 2016)
- [20] J. O. Kim, J. H. Han, "Action Plans of Green r-Learning Services based on UCR(User Created Robots)", *Korea IT Service*, Vol. 10, No. 3, pp. 21-30, 2011.
- [21] Carnegie Mellon University, "Center for Computational Thinking", Available From: <https://www.cs.cmu.edu/~CompThink>(accessed Oct, 2016)
- [22] Rusk N, Resnick M, Berg R, Pezalla-Granlund M., "New Pathways into Robotics: Strategies for Broadening Participation", *Journal of Science Education and Technology*, February, 2008.
DOI: <https://doi.org/10.1007/s10956-007-9082-2>
- [23] Robotics Design Studio, Available From: <http://cs.wellesley.edu/~rds>(accessed Oct, 2016)
- [24] Catlin D, Woollard J. "Educational Robots and Computational Thinking", *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education*, pp. 144-151, July, 2014.

구 진 희(Jin-Hui Ku)

[중신회원]



- 2001년 2월 : 충남대학교 컴퓨터과
학교육학과(교육학석사)
- 2010년 2월 : 충남대학교 공업(컴
퓨터)교육학과(교육학박사)
- 2010년 9월 ~ 현재 : 목원대학교
정보통신융합공학부 교수

<관심분야>

컴퓨터과학 교육, 빅데이터, 로봇 프로그래밍, 퍼지컬 컴퓨팅