

ARCS 모형을 적용한 컴퓨팅사고력 기반 코딩 프로젝트 개발*

남충모** · 김종우***

제주대학교 대학원** · 제주대학교 초등컴퓨터교육전공***

요약

초등학생을 위한 소프트웨어교육에서 파이썬과 같은 텍스트 기반 프로그래밍언어를 사용해 코딩을 교육하려는 연구가 활발히 이루어지고 있다. 일반적으로 블록 기반의 프로그래밍언어에 비하여 이러한 고급언어는 피지컬 컴퓨팅용 키트 또는 다양한 프로그래밍언어와 결합해 수행하는 학습 활동을 지원하고 있다. 본 연구는 텍스트 기반 언어의 어려움을 극복하기 위해 ARCS 모형을 적용한 컴퓨팅사고력 기반의 코딩 프로젝트를 수행하였다. 실험 결과에서 학생들은 전반적으로 동기유발 측면에서 프로그래밍에 대한 자신감 및 흥미를 나타내고 있으며, 특히 컴퓨팅사고력의 변화에서 반복, 함수, 객체에 대한 이해가 높게 나타났는데, 이러한 경향은 텍스트 기반 언어 사용과 파이썬 모듈의 효과로 여겨진다.

키워드 : 컴퓨팅사고력, 코딩프로젝트, ARCS 모형, 파이썬, 교수학습지도안

Development of computational thinking based Coding_Projects using the ARCS model*

Choong Mo Nam** · Chong Woo Kim***

Graduate School, Jeju National University** · Jeju National University***

ABSTRACT

Elementary students are studying software training to teach coding education using text-based languages such as Python. In general, these higher-level languages support learning activities in combination with a kits for physical computing or various programming languages, in contrast to block-coding programming languages. In this study, we conducted a coding project based on computational thinking using the ARCS model to overcome the difficulties of text-based language. The results of the experiment show that students are generally confident and interested in programming. Especially, the understanding of repetition, function, and object was high in the change of computational thinking power, so this trend is believed to be due to the use of text-based languages and the Python module.

Keywords : Computational Thinking, Coding projects, ARCS Model, Python, Teaching-Learning Process

* 이 논문은 2018학년도 제주대학교 교원성과지원사업에 의하여 연구되었음.

교신저자 : 김종우(제주대학교 교육대학 초등컴퓨터교육전공)

논문투고 : 2019-07-30

논문심사 : 2019-08-14

심사완료 : 2019-08-17

1. 연구의 필요성 및 목적

1.1 연구의 필요성 및 목적

우리 사회는 4차 산업혁명에 대한 대비를 위해 컴퓨팅사고력(CT: Computational Thinking) 기반의 문제해결을 위한 대안으로 소프트웨어교육의 필요성을 강조하고 있으며[8][11], 이를 반영한 2015 개정 교육과정에서 소프트웨어교육의 목표는 컴퓨팅사고력 기반의 문제해결 방안을 제시하고 있다.

소프트웨어교육의 핵심 요소인 프로그래밍학습을 통한 컴퓨팅사고력 기반 교수법은 아직도 프로그래밍언어 학습을 위한 교수법수준에 머물러 있고, 타 교과에서의 컴퓨팅사고력 기반 교수법에 대한 연구는 초기 단계에 있다[3][4]. 컴퓨팅사고력 기반 학습에 대한 필요성은 Wing(2006)은 읽기, 쓰기, 셈하기와 더불어 모든 학생들이 갖추어야 할 기본 기술로 알고리즘을 포함해야 한다고 주장하였으며[15], NRC(2010)에서는 컴퓨팅사고력을 컴퓨터과학의 테두리 안에만 가두지 않고 다양한 분야의 학문과 사회적 문제 해결을 위해 바탕이 되는 사고로 정의하고 있다[12][13]. 반면에, 2019년부터 초등학교에서 소프트웨어교육을 실시하며 컴퓨팅사고력을 강조하고 있으나, 초등학교 수업에 바로 사용할 수 있는 프로그래밍 교육을 위한 구체적인 컴퓨팅사고력 기반 교수학습방법이 부족하다[6][16].

본 연구는 초등학생을 대상으로 하는 소프트웨어교육에 사용할 수 있는 파이썬을 사용한 컴퓨팅사고력 기반 코딩 프로젝트 학습지도 방안을 제시하였다. 초등학생에 대한 파이썬 언어의 사용은 텍스트 기반 프로그래밍언어의 장점을 크게 갖는 반면에 초등학생들이 일반적으로 학습하는 스크래치, 엔트리와 같은 블록 코딩 언어에 비해 어렵게 느끼는 부담감이 있다. 이러한 어려움의 정도를 비교적 수월하게 다가갈 수 있는 방법으로 동기유발 ARCS 모형[1][14] 학습모형을 적용하여 파이썬 거북이 그래픽을 중심으로 사용하였으며, 초등 소프트웨어교육과정에서 추구하는 선택, 반복, 제어에 대한 학습효과에 대한 컴퓨팅사고력의 변화를 측정하였다.

본 연구는 2015 개정 교육과정에 따른 소프트웨어교육의 방안으로, 초등학생들을 대상으로 ARCS 모형에 따른 코딩 프로젝트 학습을 실시하고, 학습자의 컴퓨팅

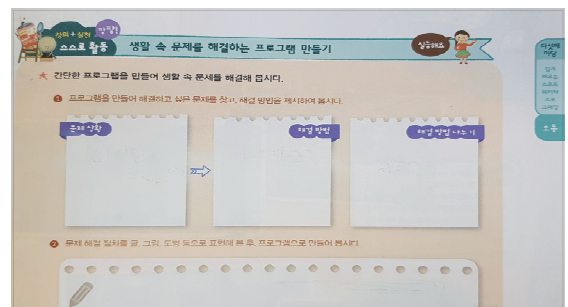
사고력에 대한 변화를 분석하여 문제점을 도출한 후, 컴퓨팅사고력을 함양시킬 수 있는 교수법을 제시하였다.

2. 이론적 배경

2.1 코딩 프로젝트

교육학에서 프로젝트(project)는 학습자가 스스로 자기 활동을 선택, 계획하고 방향을 설정해가는 문제해결 학습을 말한다. 따라서 ‘코딩 프로젝트(Coding project)’는 코딩을 통해 스스로의 문제를 해결하는 일련의 문제해결 과정을 의미한다[10].

실과의 2015 개정 교과서에서 코딩 프로젝트의 한 장면이 담겨있다. 이 장면에서 학생은 자신이 부딪힌 실생활과 관련된 문제를 철차적사고를 통해 순차적으로 문제를 해결해나간다. 그리고 문제 해결 과정에서 프로그램을 작성하게 되는데, 초등학교 교과서에는 엔트리로 프로그램을 작성하고 있다(Fig. 1).



(Fig. 1) Coding Project contents in Practical Arts Textbook

2.2 2015 개정교육과정 속 프로그래밍 언어

2015 개정 교육과정에 따라 초, 중학교에서 소프트웨어교육이 의무화되었다. 초등학교는 실과 교과에서 소프트웨어교육을 하는데, 실과 교과서 100% 모두 엔트리를 교육용 프로그래밍 언어로 채택했다. 중학교 교과서에서는 엔트리와 스크래치를 사용하는 비율이 1:1로 나타났다. 초등학교에서보다 더 나아가 센서보드를 활용한 피지컬 컴퓨팅 단계까지 배우게 된다[10]. 텍스트

기반 프로그래밍 언어는 고등학교 심화선택 ‘정보’ 교과에서 배우게 되는데, C언어, java, 파이썬 등 다양한 언어를 배우게 된다. 현재 추세로는 C 언어에 비해 코드가 간결하고, 빅데이터 처리 능력의 호환성 때문에 파이썬이 더 인기가 있다[9].

파이썬(Python)은 1991년 프로그래머인 귀도 반 로섬(Guido van Rossum)이 발표한 고급 프로그래밍 언어 중 하나로 초보자부터 전문가까지 다양한 사용자층을 보유하고 있다. 텍스트 기반(text) 프로그래밍으로 다양한 플랫폼에서 쓸 수 있고, 라이브러리(모듈)가 풍부하여, 대학을 비롯한 여러 교육 기관, 연구 기관 및 산업계에서 이용이 증가하고 있다. 또 파이썬은 순수한 프로그램 언어로서의 기능 외에도 다른 언어로 쓰인 모듈들을 연결하는 풀언어(glue language)로써 자주 이용된다. 실제 파이썬은 많은 상용 응용 프로그램에서 스크립트 언어로 채용되고 있다. 도움말 문서도 정리가 잘 되어 있으며, 유니코드 문자열을 지원해서 다양한 언어의 문자 처리에도 능하다.

2.3 학습동기 ARCS 모형

Keller의 ARCS 모형은 학습자의 동기를 유발하고 계속 유지시키기 위한 방법으로 주의집중(Attention), 관련성(Relevance), 자신감(Confidence), 만족감(Satisfaction) 4가지로 구성하였으며, 각 범주는 3개씩 하위요소를 가지고 있으며 다음 <Table 1>과 같다[1][14].

<Table 1> Components of the ARCS model

Attention	Confidence
Perceptual arousal	Learning
Provide novelty and surprise	Requirements
Inquiry arousal	Inform students about learning
Stimulate curiosity by posing questions or problems to solve	and performance requirements
and assessment criteria	
Variability	Successful
Incorporate a range of methods and media to meet student's varying needs	Opportunities
	Provide challenging and meaningful opportunities for successful learning
	Personal
	Responsibility
	Link learning success to students' personal effort and ability

Relevance	Satisfaction
Goal Orientation	Intrinsic Reinforcement
Present objectives and purpose of instruction and specific methods for successful achievement	Encourage and support learning experience
Motive Matching	Extrinsic Rewards
Match objectives to student needs and motives	Provide positive reinforcement and motivational feedback
Familiarity	Equity
Present content in ways that are understandable and related to the learners' experiences and values	Maintain consistent standards and consequences for success

2.4 컴퓨팅사고력

컴퓨팅사고력에 대한 학습 방안을 제시하려는 노력은 여러 연구자들을 통해 제시되고 있다. 특히 이들 중 미국 컴퓨터과학교사회(CSTA)와 국제교육공학협회(ISTE)에서 명시한(Operational Definition of Computational Thinking for K - 12 Education) 컴퓨팅사고력은 Wing(2006)이 제안한 추상화와 자동화를 보다 세분하여 자료 수집, 자료 해석, 자료 표현, 문제 분해, 추상화, 알고리즘 및 절차, 자동화, 시뮬레이션, 평행화로 구성된 총 9가지로 세분화시켜 컴퓨팅사고력 교육 방안을 제안하였다[5]. 또한 Google for Education(2015)에서는 이러한 분류에 패턴 인식과 일반화를 포함시켜[7] 제안하였으며, 본 연구에서는 CSTA&ISTE(2011)의 제안을 중심으로 코딩 프로젝트에 적용하였다<Table 2>.

<Table 2> Components of Computational Thinking

	Wing (2008)	CSTA & ISTE (2011)	Google for Education (2015)
Abstraction	Data Collection		
	Data Analysis		Data Analysis
	Pattern Recognition		
	Data Representation		
	Problem Decomposition		Decomposition
Automation	Abstraction		
	Pattern Generalization		
	Algorithm and Procedures		Algorithm Design
	Automation		
	Parallelization		
Simulation			

3. 코딩 프로젝트 개요

3.1 연구대상 및 기간

본 연구는 초등학교 6학년 학생을 대상으로 블록 기반 언어(엔트리)과 텍스트 기반 언어(파이썬)을 단계별로 구성하여 산출물을 제작하는 교육을 실시하였으며, 수업시간은 교과시간 중에 실과, 창의적 체험활동, 국어 시간을 활용하여 진행되었다.

연구기간은 총 13주에 걸쳐 실시하였으며, 1주에서 11주에 걸쳐서 엔트리와 파이썬을 단계별로 실시하였고, 산출물 수업은 12주와 13주에 걸쳐 제작하였다

3.2 차시별 계획 수립

엔트리와 파이썬을 첫 만남을 갖는 학생 수준에서 시작하였으며, 엔트리는 초급 단계에서 소개와 실습을 다루고 주로 파이썬으로 구성하였다. 산출물 수업은 파이썬을 사용한 코딩 프로젝트를 위하여 12주와 13주에 걸쳐 수행하였다<Table 3>.

<Table 3> The teaching schedule

Contents
Week 1. What is software? What is Python? · What software do I know? · Share stories of software applications around us · What is Python?
Week 2. Exercise Python & Entry · What is block coding? · Sign up for entries and follow entries · Complete simple programming
Week 3. Installing Python · Installing Python · Print practice · IDLE Practice · IDLE Preferences
Week 4. Working with Entry Objects · Understanding the object · Add and edit objects · Rotate on wall · Understanding and utilizing random numbers
Week 5. Understanding Variables · Understand the concept of variables and create variables · Store values in 'answer' and compute · Entry Python Mode Check

· Understanding '=='

Week 6. Creating a List

- Adding variables and lists
- Conditional block utilization
- Enter values into the list
- Change and confirm entry Python mode

Week 7. Solving textbook SW with entries

- Change the attributes of an object
- Change the appearance of objects and set their values
- Conversation animation

Week 8. Drawing with Python Tortoise

- Understanding the import module
- Go to lt, rt, ft
- Draw triangles and rectangles
- Seeing pictures and inferring how to draw

Week 9. Create a simple game

- Knowing the meaning of input
- Create variables and put data in them
- Take advantage of input and print
- Simple game programming

Week 10. Drawing Figure

- Draw regular pentagons
- Draw a circle repeatedly
- How to draw an n-square
- Free drawing and search

Week 11. Learn to repeat

- Know Python's loop structure
- Output using loops
- Create a program using loops

Week 12. Preparing to Program for Real-Time Problem Solving

- Collect data to solve problems [DC]
- Analyze collected data and find commonalities [DA]
- Solution Pictures, Writing [DR]
- Dividing the problem into smaller parts (dividing the coding steps) [PD]

Week 13. Program Presentation and Project Closure

- Think about the key elements of a program [AB]
- Programming by choosing a language [AL][AU]
- Demonstration and error correction [AU][SI]
- Role division and self-coding [PA]
- Program presentation [DR][AB]

※ [DC]: Data Collection, [DA]: Data Analysis, [DR]: Data Representation, [PD]: Problem Decomposition, [AB]: Abstraction, [AL]: Algorithms, [AU]: Automation, [SI]: Simulation, [PA]: Parallelization

3.3 컴퓨팅사고력 기반 교수·학습 지도안

산출물 제작을 위한 컴퓨팅사고력 기반 산출물 제작 교수학습 지도안은(12주~13주) 실과교과서 '03 문제해결

과 프로그래밍' 단원을 대상으로 CSTA&ISTE(2011)의 9 가지 구성요소<Table 2>를 Keller's ARCS Model<Table 1>의 주의집중(Attention), 관련성(Relevance), 자신감 (Confidence), 만족감(Satisfaction)에 적합한 요소별로 <Table 4>와 같이 재구성하여 사용하였다.

<Table 4> Teaching-Learning Process of Practical Arts education based on Computational Thinking

Concepts
<ul style="list-style-type: none"> · Date and Time: 12 to 13 weeks · Students: 15th graders (M: 5, F: 10) · Contents: <Fifth Unit> Easy-to-Learn Software and Programming <ul style="list-style-type: none"> - 03. Troubleshooting and Programming · Model: Motivational Design Model · Topic: Preparing to Program for Real-World Problem Solving · Learning objectives: Collect resources to solve real-world problems and break them down into smaller ways. · Focus: Be prepared to solve problems based on computing thinking skills, and make sure that the program can be helpful in real life. · Intention of Investigator: This learning topic aims to find out what you feel to be a problem in real life, to collect data for problem solving, and to graphically describe the solution. Furthermore, consider the key elements (methods) for problem solving, and break down the steps to solve problems before using the program as a problem solving method. Aim.
Contents
<p>Attention: Identify motivation and learning problems (time: 5 min)</p> <ul style="list-style-type: none"> · Geometry quiz story for brother (all activities) [DC] <ul style="list-style-type: none"> - Let's take a look at the shape quiz that may interest you in math. · Learning problem check[DA] <ul style="list-style-type: none"> - Prepare to create a simple program that solves the problems in your life. <p>Relevance: Preparing for Problem Solving (time: 10 min)</p> <ul style="list-style-type: none"> · Identify the problem I need to solve [DA] <ul style="list-style-type: none"> - Let's check the problem situation and organize it into text or picture. · Collect similar problems and summarize examples that solved them [AB] <ul style="list-style-type: none"> - Look for similar problem situations and see how you organized them. <p>※ Note: Collect data through the Internet and identify commonalities in the found data.</p> <p>Confidence1: Design for Problem Solving (time: 10 min)</p> <ul style="list-style-type: none"> · I envision ways to solve the problem [AR]

- Let me summarize how I will solve it with text, pictures, shapes, etc.
 - Defining learning questions [AB]
- ※ Note: Use the activity sheet.
- Confidence2: Thinking about a problem (time: 10 min)
- How to Solve Problems [PD]
 - Let me break down my solution.
 - Let's decide on the type of structure to solve.
 - Sequential structure, selection structure, repeat structure selection.[AL, AU]
 - Let's create and combine shapes by dividing them into groups [PA]
 - Let's find commonalities and differences of shapes
 - Create coherence in the way peers express themselves.
- ※ Note: Break down the problem using procedural thinking.
- Satisfaction: Organize and present assignments (time: 5 min)
- Preparing for writing program [AL, AU, SI]
 - Let's see if there is anything I can do to simplify the problem.
 - Assignment Check [DC]
 - Next time, decide which programming language you want to complete the program.
- ※ Note: Choose either entry or Python to write.

4. 현장 적용 및 결과분석

4.1 연구대상 및 연구절차

본 연구의 연구대상은 초등학교 6학년 학생을 대상 (남 5명, 여 10명)으로 블록 프로그래밍(엔트리)과 텍스트 프로그래밍(파이썬)을 단계별로 구성하여 산출물을 제작 하는 교육을 실시하였으며, 실과, 창의적 체험활동, 국어 시간에 수행하였다.

연구절차는 교육이 3월부터 6월까지 13주에 걸쳐 수행되었으며, 컴퓨팅사고력 기반 학습은 12주와 13주에 ARCS 모형 수업설계에 따라 파이썬 언어 학습을 실습 하고, 최종 산출물을 과제로 해결하는 방식으로 진행하였다<Table 5>.

13주에 걸친 코딩 프로젝트의 효과를 측정하기 위해 검사지를[1] 사용해 사전검사는 3월에, 사후검사는 6월에 실시하여 실험집단의 교육 전·후 변화에 대해 통계 분석 하였다.

<Table 5> The experimental design

	Pre-	Experiment	Post-
Experimental group	O_1	X	O_2
O_1 : Pre-test (2019.3)			
O_2 : Post-test (2019.6)			
X : Python(& Entry) education			

4.2 검사도구 와 결과분석

초등학생의 파이썬을 사용한 코딩 프로젝트 학습에서 교육에 따른 컴퓨팅사고력의 변화를 측정하기 위해 김병수[2]가 개발한 컴퓨팅사고력 검사지를 활용해 사전, 사후 검사를 실시하였다. 이 검사지는 총 15문항으로 구성되었으며, 문항별 검사주제는 순차구조, 알고리즘, 조건분기, 반복, 동시성, 변수, 함수, 객체, 재귀의 주제를 다루고 있다.

실험집단을 대상으로 코딩 프로젝트를 진행한 후 컴퓨팅사고력 검사지로 컴퓨팅 사고력을 검사한 결과는 다음과 같다<Table 6>.

<Table 6> Results of Computational Thinking Test

No.	Questionnaire	Correct		change
		Pre-test	Post-test	
1	Sequency	0	0	0
2	Sequency, Algorithm	3	4	+1
4	Conditional	9	12	+3
5	Repeat	8	14	+6
6	Sequency, repeat	7	5	-2
7	Sequency, Conditional, Repeat, Concurrency	12	8	-4
8	Sequency, Conditional, Concurrency	11	10	-1
9	Sequency, Variable, Function	8	14	+6
10	Sequency, Variable	7	7	0
11	Variable, Random	5	4	-1
12	Random, Algorithm	2	3	-1
13	Object	3	7	-4

코딩 프로젝트를 진행한 후 컴퓨팅사고력 검사를 한 결과, 증가한 항목을 살펴보면, 조건분기(+3), 반복(+6), 순차구조/변수/함수(+6), 객체(+4)으로 나타났으며, 반면

에 오히려 감소한 항목으로는 함수(-2), 순차구조/조건분기/반복/동시성(-2), 순차구조/반복(-2) 으로 나타났다. 이러한 특징은 파이썬 학습에서 이루어지는 반복, 함수, 객체에 대해 텍스트 기반 프로그래밍을 하면서 반복문을 사용하였고, 모듈의 사용이 빈번했기 때문으로 여겨진다.

실험 결과에서 학생들은 전반적으로 프로그래밍에 대한 자신감 및 흥미를 나타나고 있으며, 특히 컴퓨팅사고력의 변화에서 반복, 함수, 객체에 대한 이해가 높게 나타났다는데, 이러한 경향은 텍스트 기반 언어 사용과 파이썬 모듈의 효과로 여겨진다.

4.3 파이썬 코딩 프로젝트의 관찰

이번 코딩 프로젝트를 통해 학생들은 블록코딩 학습에 비해 전반적으로 프로그래밍 학습의 주요 요소인 반복, 함수, 객체에 대한 이해가 높아졌다. 이는 직접 프로그래밍을 하면서 반복문을 사용하였고, 파이썬의 모듈을 활용한 결과라고 생각된다. 또한 파이썬의 특성인 모든 데이터가 객체로 존재하도록 설계되었기 때문에 객체에 대한 이해가 높아진 것으로 여겨지며, 텍스트 언어의 특성인 프로그램의 변형에 접근이 쉽게 이루어졌다. 이러한 특성은 소프트웨어교육에서 추구하는 ‘따라하기 => 바꾸어보기 => 적용하기’ 교육의 추구 방향과 일치하는 경향을 보였다.

반면에 학생들은 파이썬 언어에 대한 높은 자신감과 흥미를 보이면서도 몇 가지 어려움을 호소하였다.

첫째 어려움은 영어의 사용이다. 언어의 익숙하지 못함에 대한 어려움과 엔트리 언어 직관적으로 이해하기 쉬운 블록 형태에 반하여 파이썬은 문법에 따른 규칙(예, 반복문인 경우 들여쓰기를 해야하는 것) 때문에 초등학생들이 새로운 규칙에 익숙해지는 것이 쉽지 않았다.

둘째, 오류 수정에 대한 어려움이 있다. 파이썬의 경우 오류가 생기면 오류 메시지 또한 영어로 나타난다. 초등학생은 이 메시지를 잘 이해하지 못하였다.

반면에 파이썬을 익히면서 호응이 좋았던 영역은 특히 비주얼 접근 방법으로 사용한 거북이 모듈을 사용한 그림그리기, 반복문을 통한 픽셀 아트 등은 파이썬을 처음 배우는 학생들에게 학습동기를 높이는데 효과적으로 관찰 되었으며, 학생들 대부분이 좋아하는 부분이었다.

또한, 그래픽 학습에서 간단한 도형을 그려보게 한 후 다각형을 그리거나, 크기를 스스로 변경하도록 유도하는 활동 등에는 적극적으로 참여하였으며, 또한 반복문을 활용하는 활동에도 흥미를 보였다. 반복문을 활용한 문장 출력, 반복문을 활용한 그림 그리기 등은 학습자들이 스스로 값을 바꿔보면서 자유 탐색하는 모습을 보여주었다.

학생들이 큰 관심을 보이는 간단한 게임을 만드는 프로그램 작성에서 파이썬의 내장된 모듈을 활용하는 import time을 통해 시간과 관련된 기능(time)을 요청한 후 input을 사용하여 time.time()으로 시간 값을 입력한 후 절대값을 계산하여 ‘마음 속으로 20초를 세어 맞는 게임’을 만들어 보았을 때 학생들은 모듈에 대한 이해와 절대값에 대한 이해, input에 대한 이해를 할 수가 있었다.

5. 결론 및 제언

2015 개정 교육과정을 통해 소프트웨어교육이 초등학교 현장에 본격적으로 도입이 되었다. 초등학교에서는 6학년 실과 교과에서만 다루고 있지만, 타교과와 융합하여 전개할 수 있도록 안내하고 있다.

본 연구에서 진행한 코딩 프로젝트는 초등 6학년에 재학 중인 소수인원(15명)을 대상으로 개별관찰을 중심으로 한 연구의 제한점을 갖고 있으나, 소프트웨어교육의 적은 시수를 고려하여, 국어, 실과, 창체 등을 융합하여 진행하였으며, 초등학생에게 고급 프로그래밍언어인 난이도가 높은 파이썬을 사용해 코딩 프로젝트를 수행하였다는 것에 의의가 있다. 파이썬의 사용으로 학생들은 블럭 기반 언어뿐만 아니라 텍스트 기반 프로그래밍언어를 다룰 수 있는 기회를 접할 수 있었고, 몇몇 학생들은 마치 진짜 프로그래머가 된 것 같은 기분이었다고 소감을 말하였다. 그러나 많은 학생들이 파이썬을 어려워하는 모습을 보이고 있으며, 산출물 제작 시간에(12~13주) 적용한 동기유발 학습모형(ARCS)은 텍스트 기반 언어의 어려움을 극복시키는데 도움을 주는 것으로 관찰되었으나, 일반 초등학생에게 텍스트 기반 프로그래밍언어 교육을 위해서는 무엇보다 학습에 몰입할 수 있는 추가적인 시간 배정이 필요한 것으로 여겨진다.

텍스트 기반 프로그래밍언어의 교육이 필수적임을 고

려한다면 컴퓨팅사고력 기반의 다양한 접근으로 이에 대한 일반 교과와의 연결에 대한 인식의 확장과 학습동기를 높일 수 있는 그래픽 그리기 프로그램, 게임 제작 프로그램을 중심으로 하는 프로그래밍으로 교육과정을 재구성한다면 텍스트 기반 언어에서의 강점이 초등학생에게 도입이 충분히 가능하다. 또한 각 교과와의 확장성을 고려하면 컴퓨팅사고력 기반의 교과에서의 문제를 프로그래밍하는 교육과정을 구성하고 연구가 활성화 될 필요가 있다.

참고문헌

- [1] ARCS_Model(2018), Retrieved from https://en.wikipedia.org/wiki/File:ARCS_Model
- [2] ByeongSul Kim(2014), *Programming education program based on PPS to improve computational thinking ability*, Doctor Thesis, Jeju National University.
- [3] ChoongMo Nam, ChongWoo Kim(2011). Analysis of teaching and learning activities in elementary mathematics based on Computational Thinking, *The Journal of Science of Education*, 2, 1-20.
- [4] Creative Computing(2019), Retrieved from <http://computing.or.kr/>
- [5] CSTA & ISTE(2011), *Computational Thinking teacher resources second edition*.
- [6] Daewook Kim (2019), Concept and strategy of unplugged coding for young children based on computing thinking, *The International Promotion Agency of Culture Technology*, 5(1), 297-303.
- [7] Google for education(2015), Retrieved from https://edu.google.com/?modal_active=none
- [8] Jinicoding(2019), Retrieved from <http://www.jinicoding.net>
- [9] Jungah Kim, Mingyu Kim, Hyejin Yu, Yongmin Kim, Jonghoon Kim(2019), Effect of data visualization education with using Python on computational thinking of six grade in elementary school, *Journal of The Korean Association of Information Education* 23(3),197-206.

- [10] Juyeon Park(2019), Evaluation of Computational Thinking through Code Analysis of Elementary School Students' Scratch Projects, *Journal of The Korean Association of Information Education*, 23(3), 207-217.
- [11] MSIT, NIPA(2019), Software Orinted Society, Retrieved from <https://www.software.kr/um/um03/um0305/um030503/um030503View.do?postId=34546&>
- [12] NRC(National Research Council)(2010), *Committee for the Workshops on Computational Thinking; Report of a Workshop on The Scope and Nature of Computational Thinking*.
- [13] NRC(2011), *REPORT OF A WORKSHOP OF Pedagogical Aspects of COMPUTATIONAL THINKING*.
- [14] SunOk Oh(2009), *The Relationship between Learning Motivation Variables and Academic Achievement in the ARCS Model*, Master Thesis, Busan National University of Education.
- [15] Wing J. M.(2006), Computational Thinking. *Communications of ACM*, 49(3), 33-35.
- [16] Young Hur(2019), Development of Unplugged Coding Education Program for the Elementary School, *Bulletin of Korean Society of Basic Design & Art*, 20(1), 586-598.

저자소개

남 총 모



2005. 제주교육대학교(학사)
2013. 제주대학교 교육대학원 초등
컴퓨터교육전공 교육학석사
2005.~현재 초등학교 교사
관심분야 : computational thinking,
VR교육, 프로그래밍 교육
e-mail : moyanika@naver.com

김 종 우



1997 동국대학교 대학원졸업 전산
통계(이학박사)
1989~ 제주대학교 교육대학 초등
컴퓨터교육전공 교수
관심분야: 컴퓨터교육, computational
thinking
e-mail : woo@jejunu.ac.kr