

# 알고리즘 교육을 위한 소프트웨어 배틀 - 정렬 알고리즘을 중심으로

정인기

춘천교육대학교 컴퓨터교육과

## 요 약

2018학년도부터 초·중·고등학교에서 소프트웨어 교육이 시행될 예정이다. 소프트웨어 교육의 핵심 내용 중의 하나는 알고리즘 교육이라고 할 수 있다. 다른 분야와 마찬가지로 알고리즘 교육도 학생들의 주도적 활동이 성공을 좌우한다고 볼 수 있으며 학생들의 주도적 학습은 학생들의 흥미와 몰입도를 높일 때 이끌어낼 수 있다. 이에 본 논문에서는 알고리즘 교육에 스포츠 경기를 접목하여 소프트웨어 배틀이라는 개념을 제안하였다. 소프트웨어 배틀은 가상의 세계에서 프로그래밍된 개체들이 중간에 사람의 개입 없이 주어진 문제를 해결하는 경기를 하는 것으로 정의하였다. 이는 다양한 주제에 적용될 수 있는데 본 논문에서는 알고리즘 교육의 대표적 주제인 정렬 분야에 적용을 시도하였다. 알고리즘 교육을 소프트웨어 배틀에 적용한 결과는 학생들이 생소한 면을 느끼기는 하였지만 대체적으로 긍정적인 반응을 보였다. 이는 소프트웨어 배틀이 여러 교육 방법 중에 하나로 활용될 수 있음을 나타낸다고 할 수 있다.

키워드 : 소프트웨어 교육, 컴퓨팅 사고, 알고리즘 교육, 소프트웨어 배틀

## Software Battle for Algorithm Education - Focused on Sorting Algorithm

InKee Jeong

Dept. of Computer Education, ChunCheon National University of Education

## ABSTRACT

Software education will be implemented in elementary, middle and high schools from the 2018 school year. One of the core contents of software education is algorithm education. As with other subjects, algorithmic education can also be seen as the success of students' initiative. And students' initiative learning can lead to increasing students' interest and commitment. In this paper, I propose the concept of software battle by applying sports games to algorithm education. The software battle defined that the programmed objects play to solve a given problem without human intervention in the virtual world. This can be applied to various topics. In this paper, I tried to apply it to sorting numbers which is a representative subject of algorithm education. The results of applying the algorithm education to the software battles showed a positive response, although the students felt the unfamiliar side. This indicates that the software battle can be used as one of the various education methods.

Keywords : SW Education, Computational Thinking, Algorithm Education, SW Battle

---

논문투고 : 2018-02-28

논문심사 : 2018-03-08

심사완료 : 2018-04-17

## 1. 서론

4차 산업 혁명이 도래하면서 세계 각국은 초·중·고등 학생들에게도 소프트웨어 교육을 하기 시작했다. 이에 우리나라에서도 중등에서는 2018학년도부터, 초등에서는 2019학년도부터 소프트웨어 교육을 실시할 예정이다. 소프트웨어 교육의 목표는 다양하게 주장될 수 있지만 대체적으로 Computational Thinking으로 대표되는 것이 일반적이다. 또한 학생들에게 문제 해결 방법을 알고리즘으로 표현하도록 하는 것이 학습 내용 중에서 많은 부분을 차지하는 것이 사실이다.

성민숙[7]은 알고리즘 학습 과정 속에서 자신이 학습의 주체로서 주어진 과제를 이해 분석하고, 과제 수행을 위한 방법을 찾으며, 그 방법을 목적인대로 실행하고 수정하는 작업을 거치는 가운데 보다 논리적인 사고를 할 수 있다고 하였으며, 결국 문제 해결력 향상, 수학적 사고력 신장 및 창의력 신장을 할 수 있다고 주장하였다.

일반적으로 알고리즘이란 우리가 어떤 문제를 해결하기 위하여 해결 절차를 알기 쉽도록 나타내는 논리적인 절차과정을 의미한다. 간단히 말하면 어떤 일을 하기 위한 방법을 절차를 말하는 것이다[3].

이에 많은 국가와 학자들은 학생들에게 알고리즘을 효과적으로 교육할 수 있는 방법에 대하여 많은 연구가 이루어지고 있다. 특히, 학생들의 흥미를 끌어내고 몰입할 수 있는 교육 방법에 대하여 다각도로 연구되고 있다. 따라서 본 논문에서는 학생들이 흥미를 가지고 몰입할 수 있는 형태인 소프트웨어 배틀 형식을 제안하고 이를 대표적인 알고리즘 주제인 정렬에 적용하여 보았다.

## 2. 이론적 배경

임서은, 정영식[6]은 학생들의 컴퓨팅 사고력을 향상시키기 위해서는 컴퓨터 과학의 문제를 학생 수준에 맞게 제공하고, 학생이 문제를 해결해나가는 과정에서 알고리즘 기반의 사고를 할 수 있도록 해야 한다고 하였다.

또한 김경훈[5]은 알고리즘 관련 지식의 성격은 명제적 지식을 바탕으로 문제 해결을 위한 방법적 지식을 습득하고 이를 명시적으로 표현하는 지식 체계의 특성을 갖는다고 주장하였다. 즉, 문제 해결을 위하여 시도하는 다양한 방법 등이 알고리즘의 개념이라고 하였다.

김철[1]은 컴퓨터 과학적 사고의 핵심은 문제를 이해하고, 그것을 해결하기 위해 프로그램으로 구현하기 전에 해결 방법과 절차를 구안하는 알고리즘을 설계하는 것이 중요하다고 하였으며, 문교식[2]은 알고리즘 교육의 중요성을 다음과 같이 주장하였다.

- 알고리즘의 이해와 개발의 과정을 통한 논리력, 사고력, 창의력 함양
- 알고리즘은 프로그래밍과 컴퓨터에 대한 기본 개념
- 알고리즘 교육은 컴퓨터 프로그래밍의 핵심 과목
- 알고리즘 교육을 통한 컴퓨터 사용에 관한 문제 해결력 신장

이러한 알고리즘 교육은 지금까지 C 혹은 Java와 같은 텍스트 기반의 프로그래밍 언어를 사용하여 이루어져 왔다. 그러나, 초등을 비롯한 중등 학생들은 아직 텍스트 기반의 프로그래밍 언어에 익숙하지 못하며, 잘못하면 소프트웨어 교육 자체에 흥미를 잃게 될 우려도 있는 것이 사실이다. 따라서, 본 논문에서는 어린 학생들에게 친숙한 스크래치 프로그래밍 도구를 사용하고자 한다.

스크래치는 미국의 MIT 미디어 랩의 Lifelong Kindergarten Group에서 개발한 교육용 프로그래밍 언어이다. 만 8세에서 16세 사이의 학생들에게 프로그래밍을 가르치기 위한 목적으로 개발되었고, 무료로 사용할 수 있는 공개 소프트웨어이다[4].

스크래치 프로그램의 특징은 다음과 같다. 첫째, ‘블록쌓기’ 방식이다. 두 번째, ‘멀티미디어’ 지원이다. 세 번째 ‘공유’이다[8].

현재 어린 학생들이 가장 많이 사용하고 있는 프로그래밍 도구로써 많은 프로젝트가 공개되어 있기 때문에 학생들이 문제 해결에 많은 도움을 받을 수 있는 장점이 있다.

## 3. 소프트웨어 배틀

### 3.1 소프트웨어 배틀의 개념

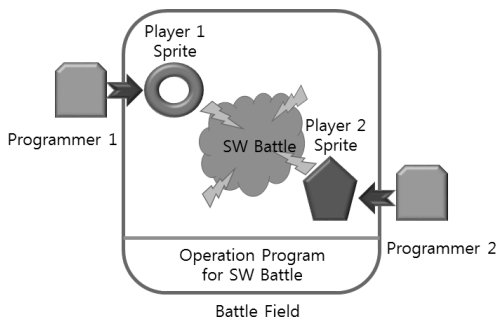
많은 사람들은 스포츠에 열광한다. 직접 참가하는 것뿐만 아니라 보는 것도 즐거워한다. 성별과 나이를 가리

지 않는다고 볼 수 있다. 이는 스포츠가 일정한 규칙에 따라 승부를 가리며, 상대성이 있기 때문이라고도 볼 수 있다. 알고리즘의 경우에도 하나의 정답이 있는 것이 아니며 주어진 상황에 따라 결과가 다르게 나올 수 있는 특성을 가지고 있다. 따라서 이를 접목하고자 한 것이 소프트웨어 배틀이다. 마치 경기장 안에서 프로그래밍된 로봇이 배틀하는 것을 가상의 세계에서 프로그래밍된 소프트웨어 개체로 배틀을 하는 것이라고 볼 수 있다.

본 논문에서는 자신들이 만든 플레이어 스프라이트를 가지고 대결하는 것을 “소프트웨어 배틀”이라고 명명하고자 한다. 참가자는 자신이 프로그래밍한 개체(스크래치인 경우에는 스프라이트)를 가지고 배틀에 참여하게 되며, 일단 배틀이 시작되면 프로그램을 수정할 수 없다. 즉, 미리 작성한 프로그램에 의해 플레이어 스프라이트(개체)가 자율적으로 배틀을 하게 되는 것이다.

### 3.2 소프트웨어 배틀의 구성

소프트웨어 배틀은 크게 배틀 필드와 플레이어 스프라이트(개체)로 구성된다. 소프트웨어 배틀이 벌어지는 곳인 일종의 경기장을 “배틀 필드”라고 하는데, 이러한 개념을 그림으로 표현하면 (Fig. 1)과 같다.



(Fig. 1) Conceptual Diagram for SW Battle

### 3.3 소프트웨어 배틀의 필요 조건

소프트웨어 배틀을 하기 위해서는 프로그래밍 도구가 다음과 같은 특성을 가지고 있어야 한다.

- ① 개체(혹은 스프라이트) 단위로 저장하고 삽입할 수 있어야 한다.
- ② 신호에 따라 개체(혹은 스프라이트)가 동시에 실행될 수 있어야 한다.

스크래치의 경우에는 스프라이트 단위로 다운로드 및 업로드할 수 있으며, 방송하기 블록을 사용하여 동시에 스프라이트들을 실행시킬 수 있으므로 소프트웨어 배틀을 진행할 수 있는 조건을 만족한다고 볼 수 있다.

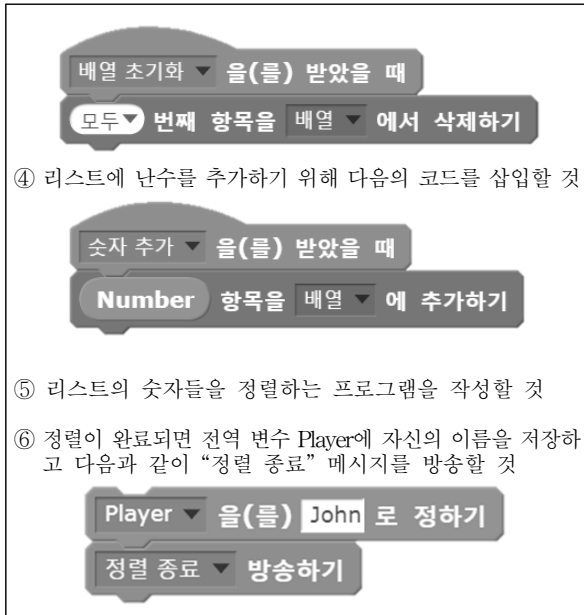
## 4. 소프트웨어 배틀의 예 - 정렬 알고리즘

정렬 알고리즘은 소프트웨어 교육에서 많이 활용되고 있는 주제이다. 다양한 특성을 가진 알고리즘이 개발되어 있어 학생들이 알고리즘을 평가하기에 적당하기 때문에 알고리즘의 평가를 시각적으로 할 수 있는 소프트웨어 배틀의 예로서는 적당하다고 할 수 있다. 정렬 알고리즘을 주제로 한 소프트웨어 배틀의 예는 다음과 같다.

### 4.1 소프트웨어 배틀의 준비

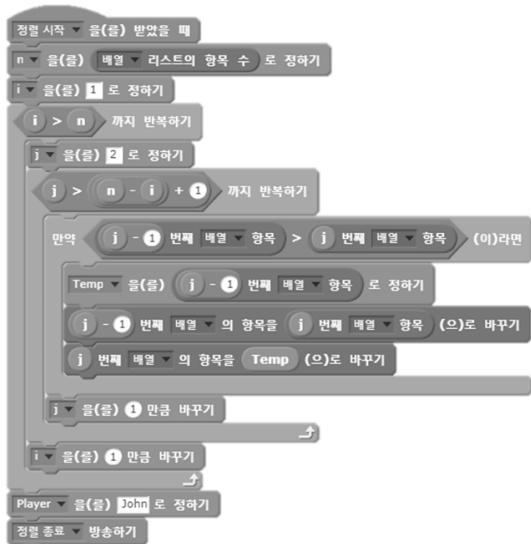
소프트웨어 배틀은 일정한 규칙 하에서 대결하는 것으로 플레이어의 프로그램을 작성할 때 지켜야 하는 규정이 있다. 이를 지켜야 소프트웨어 배틀을 시행할 수 있다. 예를 들어, 정렬 알고리즘을 주제로 한 소프트웨어 배틀을 위한 플레이어 스프라이트 프로그램을 작성할 때 지켜야 하는 규정을 안내한 예는 다음과 같다.

프로그램 작성 시 규정
① 다음의 두 변수만 전역 변수 (“모든 스프라이트에 대해” 선택)로 만들 것 - Number : 난수로 생성된 숫자의 저장 - Player : 자신의 이름을 저장
② 그 외의 모든 변수 및 리스트는 지역 변수 (“이 스프라이트에 대해” 선택)로 만들 것
③ 리스트의 초기화를 위해 다음의 코드를 삽입할 것



#### 4.2 플레이어 스프라이트 프로그램의 개발

플레이어 스프라이트는 배틀에 참가하는 스프라이트 (개체)로 참가자가 배틀을 위해 안내된 규정에 따라 프로그램을 작성한다. 정렬 알고리즘을 주제로 한 소프트웨어 배틀을 위한 플레이어 스프라이트 프로그램을 앞에서 언급한 규정을 참조하여 작성한 예는 (Fig. 2)와 같다.



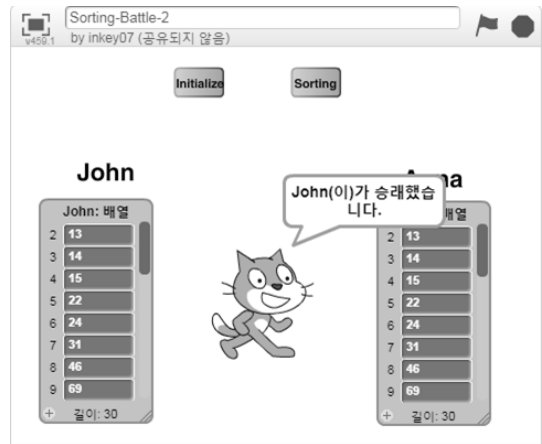
(Fig. 2) Example of Sprite Program for Sorting

#### 4.3 소프트웨어 배틀의 진행

정렬 알고리즘을 주제로 한 소프트웨어 배틀은 다음과 같이 진행할 수 있다.

- ① 교수는 배틀 필드 프로그램을 작성하고, 학습자들에게 소프트웨어 배틀을 위한 주제와 프로그래밍 작성 규정을 안내한다.
- ② 학습자들은 안내된 규정에 따라 플레이어 스프라이트에 대한 프로그램을 작성하고, 스프라이트를 저장하여 교수자에게 제출한다. (이때, 저장하는 것은 스크래치 프로젝트가 아닌 스프라이트이다.)
- ③ 제출한 플레이어 스프라이트 프로그램을 배틀 필드에 업로드한 후 배틀을 시작한다.

정렬 알고리즘을 주제로 소프트웨어 배틀을 진행한 예는 (Fig. 3)과 같다.



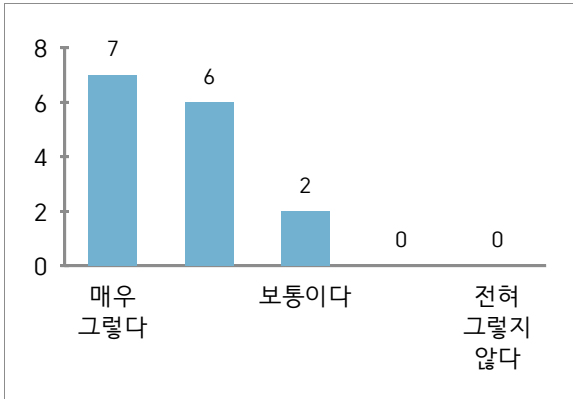
(Fig. 3) Execution of Sorting Battle

#### 5. 소프트웨어 배틀 진행 결과

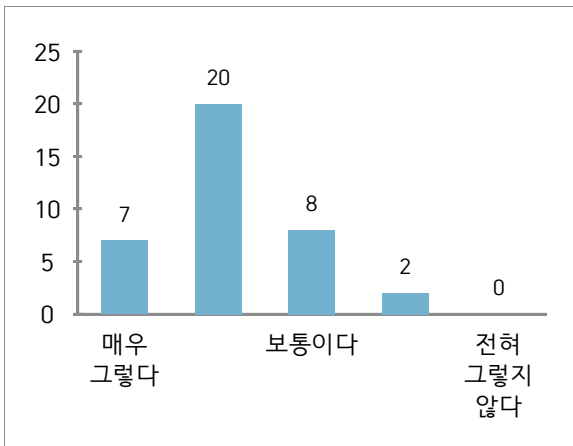
##### 5.1 교육 효과

C 교육대학교의 컴퓨터교육과 학생들을 대상으로 2년 동안 실시하면서 조사한 설문 결과는 다음과 같다.

먼저, 스크래치로 알고리즘을 구현하는 것에 도움이 되었는가의 질문에 대한 남학생 및 여학생들의 응답은 (Fig. 4) 및 (Fig. 5)와 같았다.



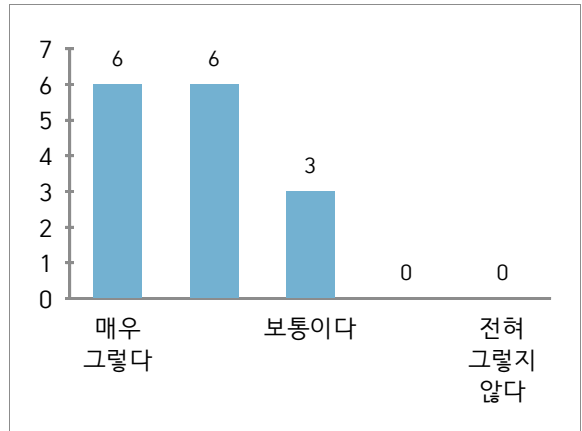
(Fig. 4) Was it helpful to implement the algorithm? (Male)



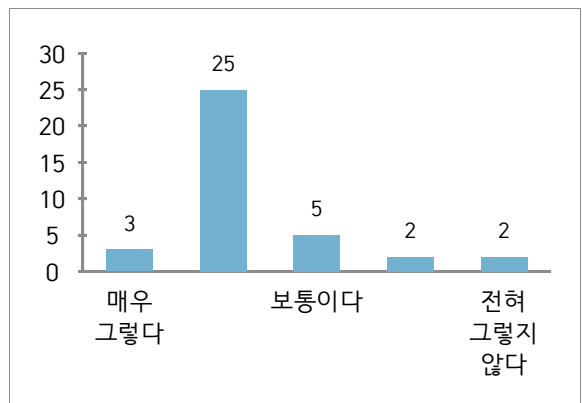
(Fig. 5) Was it helpful to implement the algorithm? (Female)

(Fig. 4)와 (Fig. 5)에서 보는 바와 같이 남학생은 약 86.7%, 여학생은 약 73%의 학습자들이 긍정적으로 응답하였다. 이는 스크래치로 알고리즘을 구현하는 것이 적어도 초등학교 수준에서는 충분히 가능하다는 것으로 해석할 수 있다.

소프트웨어 배틀로 하는 수업이 알고리즘 공부에 도움이 되었는가에 대한 남학생 및 여학생들의 응답은 (Fig. 6) 및 (Fig. 7)과 같았다.



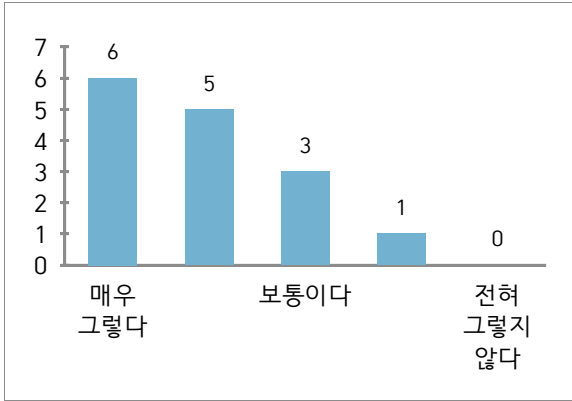
(Fig. 6) Was it helpful to study the algorithm? (Male)



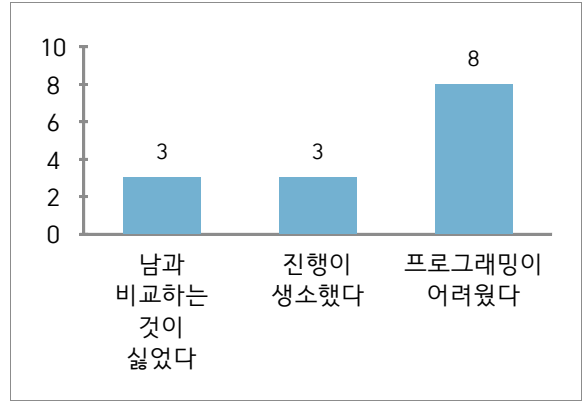
(Fig. 7) Was it helpful to study the algorithm? (Female)

(Fig. 6)과 (Fig. 7)에서 보는 바와 같이 남학생은 약 80%, 여학생은 약 75.7%의 학습자들이 긍정적으로 응답하였다. 이는 소프트웨어 배틀을 소프트웨어 교육에 도입하는 것에 대한 긍정적인 신호로 해석할 수 있다.

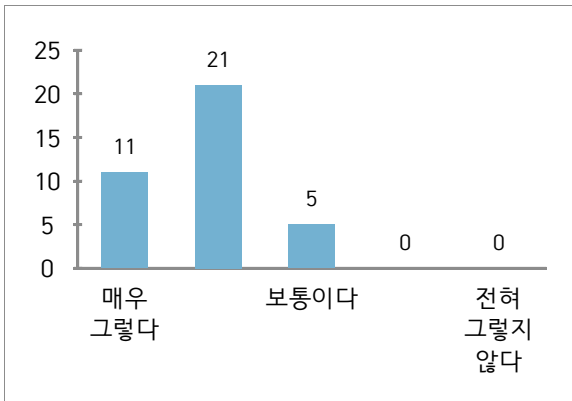
소프트웨어 배틀이 각 알고리즘을 비교하는데 되었는데 가에 대한 남학생 및 여학생들의 응답은 (Fig. 8) 및 (Fig. 9)와 같았다.



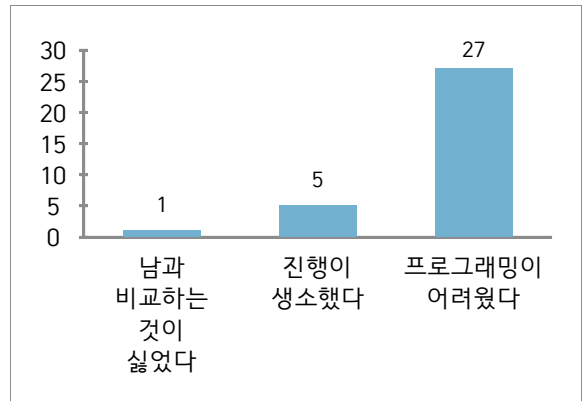
(Fig. 8) Was it helpful to compare algorithms? (Male)



(Fig. 10) What was difficult? (Male)



(Fig. 9) Was it helpful to compare algorithms? (Female)



(Fig. 11) What was difficult? (Female)  
(No Response - 4)

(Fig. 8)과 (Fig. 9)에서 보는 바와 같이 남학생은 약 73.3%, 여학생은 약 86.5%의 학습자들이 긍정적으로 응답하였다. 이는 소프트웨어 배틀을 통한 알고리즘 비교가 효과가 있었던 것으로 해석할 수 있다.

소프트웨어 배틀을 하는데 있어서 어려웠던 점을 묻는 질문에 대한 남학생 및 여학생들의 응답은 (Fig. 10) 및 (Fig. 11)과 같았다.

(Fig. 10)과 (Fig. 11)에서 보는 바와 같이 남학생은 약 53.3%, 여학생은 약 73%의 학습자들이 긍정적으로 응답하였다. 이는 교육대학교 학생들에게 있어서 효율적인 알고리즘을 직접 설계하고 구현하는 것이 쉽지 않은 작업으로 해석할 수 있다.

## 5.2 문제점 및 추후 과제

스크래치를 활용한 소프트웨어 배틀을 진행하면서 나타난 문제점은 다음과 같다.

- ① 무승부가 없다. 동일한 알고리즘을 사용하여도 미

세하게라도 승부가 갈린다. 동일한 알고리즘을 사용하지 않는 경우에는 문제가 없으나 정확도 면에서는 신뢰성이 떨어질 수 있다.

- ② 학생들의 설문 결과에서 가장 어려웠던 점이 프로그래밍이 어려웠다는 점이다. 생소한 면과 함께 소프트웨어 배틀을 진행하기 위해서는 스크래치와 주어진 주제에 대한 알고리즘 교육이 선행되어야 하기 때문에 필요 시간이 많이 소요될 수 있다.
- ③ 배틀 필드의 정교한 프로그램이 필요하다. 주제에 따라 다양한 배틀 규정 즉, 배틀 필드 프로그램이 필요한데 전문가의 도움이 필요하다.

위와 같이 나타나는 문제점에도 불구하고 학생들은 긍정적인 의견을 많이 보였다. 따라서 소프트웨어 배틀을 성공시키기 위하여 다음과 같은 분야의 연구를 계속 진행할 계획이다.

- ① 소프트웨어 배틀의 전형적인 플랫폼을 개발할 계획이다. 현재는 정렬에 맞춰줘 있기 때문에 배틀 필드 프로그램을 작성하기 위한 가이드라인이 없다. 이를 개발할 계획이다.
- ② 소프트웨어 배틀을 체계적으로 분류하고자 한다. 소프트웨어 배틀은 다양한 주제에 적용될 수 있다. 따라서 체계적으로 분류하여 이에 적합한 소프트웨어 배틀 형태를 제공하고자 한다.

## 6. 결론 및 제언

알고리즘 교육은 소프트웨어 교육의 핵심 내용이지만 학생들에게 어렵기만한 분야로 인식되었던 것이 사실이다. 특히, 목표가 정해지지 않은 상황에서의 학습은 학습 동기가 완벽하게 부여되지 않음으로써 지루한 수업이 이루어져 왔다. 또한 문제에 대해서 여러 해결 알고리즘이 주어질 수 있으며 이는 상황에 따라 효과성이 달라질 수 있음을 경험하기가 어려웠다.

그런데, 가상의 세계에서 자신이 프로그래밍한 소프트웨어 배틀로 중간에 사람의 개입 없이 배틀을 한다는 개념은 학생들에게 생소하지만 신선한 형태로 다가온

것으로 평가되었다. 또한 상황에 따라 알고리즘의 효과가 달라질 수 있음을 직접 경험할 수 있게 된 것은 시사하는 바가 크다고 할 수 있다.

소프트웨어 배틀은 여러 주제에 대하여 활용될 수 있는 방법이다. 앞으로 다양한 주제에 대하여 소프트웨어 배틀 개념을 적용할 수 있는 방법을 모색하여 학생들이 흥미를 가지고 소프트웨어 교육에 몰입할 수 있는 방안들이 개발된다면 소프트웨어 교육의 목표를 달성하는데 도움이 될 것으로 기대한다.

## 참고문헌

- [1] Chul Kim (2015). A Study on Algorithm Teaching and Learning Methods and Assessment for Elementary School Students. *The Korean Association of Information Education*, 19(4), 489-498.
- [2] Gyo Sik Moon (2007). On the Direction of the Computer Algorithm Education Based on Conceptual Algorithms. *Journal of The Korean Association of Information Education*, 11(1), 29-38.
- [3] KEDI (2005). *Teaching and Learning Materials for Information Science Gifted Education* (Understanding Algorithm).
- [4] Kim, Hyo Sung (2016). *A Development and Application of Programming Education Program focused on Algorithm Design to enhance Creative Problem Solving Ability*. Dissertation of Master Grade. Seoul National University of Education.
- [5] Kyung-Hoon Kim (2006). A Study on determining hierarchy about the domain specific knowledge of the algorithm in middle schools. *The Journal of Korean Association of Computer Education*, 9(5), 41-51.
- [6] Seo Eun Lim, Youngsik Jeong (2017). Development of Teaching and Learning Methods Based on Algorithms for Improving Computational Thinking. *Journal of The Korean Association of Information Education*, 21(6), 629-638.
- [7] Sung, Min-suk (2007). *A Study on the Improvement*

*of Computer Education Curriculum for Algorithm unit in Element, Middle, High School.* Dissertation of Master Grade. Chun-Ang University.

- [8] YoungJun Lee, EnuKyoung Lee (2010). *Scratch Programming*. KyoHakSa Ltd.

## 저자소개

### 정 인 기



1988 고려대학교 전산학과 (이학사)

1990 고려대학교 대학원 수학과 전산학전공 (이학석사)

1996 고려대학교 대학원 전산학과 전산학전공 (이학박사)

1997~현재 춘천교육대학교 컴퓨터교육과 교수

관심분야: 컴퓨터과학교육, 프로그래밍교육

e-mail: inkey@cnue.ac.kr