

데이터 에러 검출과 수정에 대한 초등교육자료 개발

고형철* · 김종우**

제주 장전초등학교* · 제주대학교 초등컴퓨터교육전공**

요 약

2015 개정 교육과정에 따른 초등학교 소프트웨어교육의 내용에서 큰 변화는 컴퓨터과학 언플러그드교육의 제시이다. 이 교육과정에서 담겨 있는 데이터에러 검출과 수정에 대한 교육자료와 교수법은 부족하여 우리나라 현장학교에 도입하기에 어려움이 있다. 본 연구에서는 이 주제와 관련된 선행 연구를 바탕으로 초등 고학년에 적합한 해밍코드를 활용한 교육자료를 개발하였다.

교육자료의 개발과정은 단계별 학습을 위해 도입부에 카드마술을 소개하고, 해밍코드의 원리를 바탕으로 '에러 검출 및 수정' 교육자료를 활동중심교육으로 구성하였다. 본 연구에서 제시한 교육자료의 현장적용에 대한 평가에서 학습자들은 컴퓨터과학에 대한 이해도를 향상시키는데 긍정적인 영향을 준 것으로 나타났다.

키워드 : 초등정보교육, 컴퓨터과학 언플러그드, 데이터에러 검출과 수정, 활동중심학습, 교육자료

Development of Elementary learning materials for Data error detection and correction

Hyeongcheol Ko*, Chongwoo Kim**

Jeju Jangjeon Elementary School*

Major in Elementary Computer Education, Jeju National University**

ABSTRACT

CS Unplugged education at the base of computer science is emphasized as an instrument for teaching the basic principles of elementary SW education, but these materials for elementary education are very lacking. So We'll present the data error detection and correction materials for elementary school classes. Based on previous studies related to this topic, we developed learning materials for elementary higher grade students using Hamming code. We introduces the card magic in the introduction part. 'error detection and correction' learning materials based on the principle of Hamming code, were composed as activity-based education. The results of the questionnaire survey showed that it had a positive effect on improving learners' understanding of computer science.

Keywords: elementary computer education, CS Unplugged, data error detection and correction, activity-based learning, learning materials

교신저자 : 김종우(제주대학교 초등컴퓨터교육전공)

논문투고 : 2018-02-08

논문심사 : 2018-02-09

심사완료 : 2018-02-21

1. 서론

1.1 연구의 필요성 및 목적

현대사회의 수많은 사람들은 컴퓨터를 이용하여 필요한 자료를 주고받는 것을 일상처럼 여기며 살아간다[6]. 컴퓨터를 통하여 전달되는 자료는 데이터로 처리되고 물리적인 장치에 저장하여 이동하거나 또는 전자 우편과 같은 네트워크 기반의 통신을 통해 주고받는다[6]. 대부분의 사람들은 이 과정에서 주고받은 데이터에는 큰 문제가 생기지 않을 것이라는 가정하에 주고받는다. 하지만 사람들의 기대와는 달리 데이터의 저장 또는 전송 과정에서 데이터가 손실 또는 에러가 생기는 현상이 일어나고 이로 인해 원하는 데이터를 주고받지 못하는 불편함을 겪는 경우가 종종 발생한다[2][9].

컴퓨터과학자들은 데이터의 저장 또는 전송 과정에서 원본 데이터의 에러 여부를 확인하는 과정에 대한 데이터 에러 검출과 수정 하는 알고리즘을 개발하고 있으며, 이에 대한 학습은 컴퓨팅 사고력을 향상시키는데 매우 효과적이다[10][11][14].

본 연구는 에러 검출과 수정 방법의 교육자료 개발을 위해 패리티 비트를 활용한 해밍코드 기법을 사용하여 학습도구를 활동중심교육자료로 만들고 현장교육에서 적용하였다. 이 연구물의 평가를 위해 교육자료가 데이터 에러 검출과 수정에 대한 학생들의 이해도를 얼마나 높일 수 있었는지를 설문조사와 관찰을 통해 검증하였다.

2. 데이터 에러 검출과 수정

2.1. 컴퓨터과학 언플러그드 학습

컴퓨터과학 언플러그드(computer science unplugged: CS Unplugged) 학습은 초등컴퓨터과학을 교육하기 위해 개발된 널리 알려진 자료이다 [2][4][6]. 이 교육자료는 Tim Bell, Ian H. Written, Mike Fellows[14]가 개발한 자료이며(<Table 1>), 특히 놀이 중심 학습은 초등학생에게 매우 효과적이다[9][13]. 컴퓨터과학이 실생활에서 사용되고 있는 사례를 중심으로 체험 활동중심 학

습으로 구성하여 학습자가 학습몰입을 높이고 있으며 [8], 게임과 놀이를 통해 컴퓨터과학의 기본 개념인 디지털(이진수), 알고리즘, 프로그래밍 등의 원리를 배울 수 있도록 제시하고 있다[15].

<Table 1> The contents of CS Unplugged[14]

Data:therawmaterial – Representinginformation
Count the Dots – Binary Numbers
Colour by Numbers – Image Representation
You Can Say That Again! – Text Compression
CardFlipMagic – ErrorDetection&Correction
Twenty Guesses – Information Theory
PuttingComputersto Work – Algorithms
Battleships – Searching Algorithms
Lightest and Heaviest – Sorting Algorithms
Beat the Clock – Sorting Networks
The Muddy City – Minimal Spanning Trees
The Orange Game – Routing and Deadlock in Networks
Tablets of Stone – Network Communication Protocols
TellingComputersWhatToDo – Representing Procedures
Treasure Hunt – Finite-State Automata
Marching Orders – Programming Languages
Really hard problems – intractability
Thepoorcartographer – Graphcoloring
Tourist town – Dominating sets
Ice roads – Steiner trees
Sharingsecretsandfightingcrime – Cryptography
Sharing secrets – Information hiding protocols
The Peruvian coin flip – Cryptographic protocols
Kid Krypto – Public-key encryption
Thehumanfaceofcomputing – Interacting with computers
The chocolate factory – Human interface design
Conversations with computers – The Turing test

본 연구에서 자료개발에 중점을 둔 영역은 데이터 에러와 검출 영역이다.

“ Data: the raw material – Representing information (데이터: 원시-정보표현)

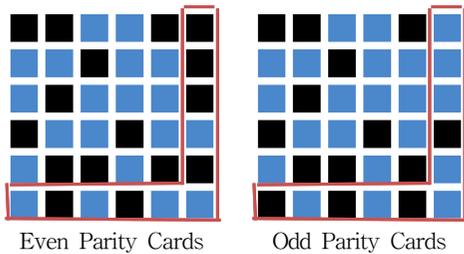
Card Flip Magic – Error Detection & Correction (카드 뒤집기 마술-에러검출과 수정)

<Table 1>의 자료 중에 카드마술을 에러검출의 도입부로 활용해서 해밍코드에 기초한 초등학교 고학년(5~6)을 대상으로 1시간에 걸친 학습 자료를 개발하였다.

2.2. 패리티 비트와 해밍코드

에러검출을 위한 카드 뒤집기 마술에서 5×5 카드배열에 패리티 카드를 추가할 때 카드의 색상 배열을 짝수 또는 홀수로 할 것인지 결정하여 카드를 놓기 때문에 6×6 카드배열은 다음과 같이 일정한 규칙을 갖게 된다.

<Table 2> Even/Odd Parity Cards



패리티 비트란 정보전달 과정에서 에러를 검사하기 위해 추가된 비트로 홀수 패리티(odd parity), 짝수 패리티(even parity) 2가지 형태로 만들어진다.

해밍코드는 기존의 데이터에 특정 규칙을 가진 여러 개의 짝수 패리티 비트를 삽입하여 새로운 데이터를 만들며 수신측에서는 이 짝수 패리티 비트들을 검사해서 컴퓨터 스스로 데이터의 오류를 검출하고 수정한다. 일반적으로 에러검출 코드가 에러를 검출하여 오류를 알려주는데 그치는 반면 해밍코드는 에러를 검출하여 스스로 수정할 수 있도록 개선되었다[12].

2.3. 데이터에러 제어 관련연구

원자료 데이터를 최소의 오류로 원하는 목적지까지 전달하는 방법에 관한 데이터 에러 제어 기법 연구는 1940년대 말 샤논(Shannon, Claude Elwood) 연구에서 출발점을 찾을 수 있다[16]. 데이터 에러 제어의 수행은 원자료에 여분의 정보(비트)를 포함시켜 부호화된 신호를 전송하는 방식으로 이뤄진다.

부호화된 신호를 전송할 때에 발생하는 에러유형은 4가지 범주로 나눌 수 있다.

첫째, 부호에 의해서 정정되어 마치 오류가 발생하지 않은 것처럼 정보를 목적지까지 전송하는 것

둘째, 검출은 되지만 수정되지 않는 에러 셋째, 전혀 검출되지 않는 에러 넷째, 검출은 되나 수정을 할 경우 잘못된 결과를 일으키는 에러가 있다.

데이터 오류를 검출하기 위한 원리는 다음과 같다. 데이터가 전송될 때 Error Detection Code를 구성하는 부가적인 비트를 첨가시킨다. 송수신 데이터의 비트를 일정한 규칙을 갖는 연산을 통해 전송 전, 후 값을 비교하여 일치여부를 확인하고 불일치의 경우 데이터 오류로 판정한다.

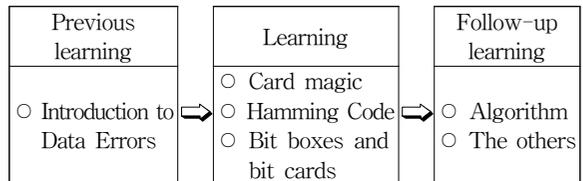
데이터 에러 검출에 일반적으로 사용되는 기법은 패리티 비트(Parity Bit), 세로 중복 검사(Longitudinal Redundancy Check), 순환 중복 검사(Circle Redundancy Check), 해밍 코드(Hamming Code) 기법이 있다[6].

3. 교수 · 학습의 실제

컴퓨터의 데이터는 원하는 정보를 찾고 네트워크를 통해서 정보를 보내는 작업에서 데이터의 안정성을 위해 에러검출(error detection)과 에러수정(error correction) 과정을 수행한다.

에러검출은 데이터를 저장하거나 주고받을 때, 데이터가 변경되거나 손상되었는지를 확인하는 과정이며, 에러수정은 데이터가 변경 또는 손상되었을 때 원본 데이터로 복원하는 과정으로 학습의 진행은 선수학습과 본 학습 그리고 후속학습으로 진행된다<Table 3>.

<Table 3> The flow of contents



3.1 선수학습

생활 속에서 데이터 에러가 발생하는 사례를 통해 발생하는데, 대표적인 세 가지 요소는 다음과 같다.

첫째, 정전기, 전기불꽃, 불안정한 전기 공급 등으로

인한 전기적 간섭.

둘째, 하드웨어에 외부충격으로 물리적 손상.

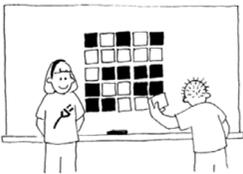
셋째, 악성코드 및 바이러스로 인한 데이터 손실 또는 해킹으로 인한 데이터 변형

3.2 본 학습

[활동 1] 카드마술과 패리티 비트

Tim Bell의 2인[14]에서 제시된 카드 뒤집기 마술을 사용해 다음과 같은 순서로 진행한다.

첫째, 색이 다른 양면 카드를 36장 준비한다.



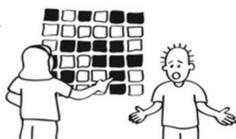
둘째, 마술을 보여줄 상대에게 카드 25장을 무작위로 5*5 형태로 배치하도록 이야기한다.



셋째, 마술을 시연하는 마술사는 난이도를 더욱 높인다고 말하며 직접 25장의 카드가 놓인 가로와 세로줄에 각각 카드 1줄씩 추가 배치한다. 이때 마술사가 추가하는 카드를 ‘패리티 카드’라 하며 마술사는 카드의 가로 세로 색상 배열이 짝수 또는 홀수가 되도록 놓는다.



넷째, 마술사가 눈을 가리고 있는 동안 상대방은 자신이 원하는 카드 1장을 선택하여 뒤집는다.



다섯째, 마술사는 상대방이 찾는 방법을 쉽게 알아내지 않게 주의하며 짝수 배열일 경우 가로 세로 홀수 줄을 찾고, 홀수 배열일 경우 가로

세로 짝수 줄을 찾아 뒤집어진 해당 카드를 찾아낸다.

이 활동을 통해 숨어있는 비트(패리티비트)가 에러를 검출하고 수정하는데 사용될 수 있음을 학습자가 알게 한다.

[활동 2] 데이터 에러 검출과 수정

마술카드는 2차원상에서 에러를 점검하지만, 해밍코드를 활용한 데이터 검출과 수정은 단계별로 패리티 검사를 통해 데이터 에러를 검출하고 수정하기 위해 다음 알고리즘에 따라 수행한다.

[에러 검출 알고리즘]

- ① 짝수(또는 홀수) 패리티를 사용한다고 가정하고,
- ② 1, 3, 5, 7 번째 비트의 합이 ①에서 설정된 패리티에 맞으면 0을 틀리면 1을 패리티1 검사결과 값에 적는다.
- ③ 2, 3, 6, 7 번째 비트의 합이 ①에서 설정된 패리티에 맞으면 0을 틀리면 1을 패리티1 검사결과 값에 적는다.
- ④ 4, 5, 6, 7 번째 비트의 합이 ①에서 설정된 패리티에 맞으면 0을 틀리면 1을 패리티1 검사결과 값에 적는다.
- ⑤ 검사결과 값을 앞서 진행해 온 ④,③,② 순으로 이진수를 완성한다. 이 수의 위치에 있는 비트가 오류로 검출된다(Fig. 1).

P1	1	0	1	1	Error 1	↑	011 ₍₂₎ = 3
P2	0	0	0	1	Error 1		
P3	0	1	0	1	Correct 0		

ERROR	1	0	0	0	1	0	1
SOURCE	1	0	1	0	1	0	1

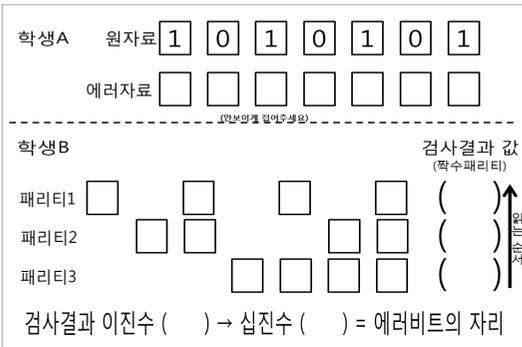
(Fig. 1) The Error detection position

【교구의 마련】

◎ 해밍코드 활동지

두 명이 한 조(A, B)를 이뤄 학생 각자 활동지를 (Fig. 2)와 같이 만든다. 원자료는 7개의 비트로 이뤄지고, 에러자료는 그 중 하나의 비트를 선택하여 0또는 1로 바꾼다.

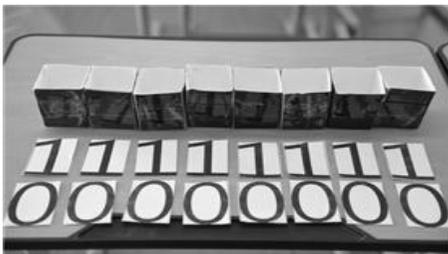
원자료와 에러자료를 만들었으면 점선을 따라 뒤로 접어 상대방이 볼 수 없도록 한다.



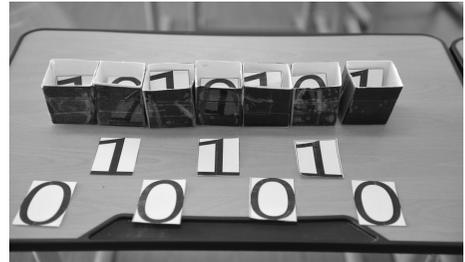
(Fig. 2) The Activity Sheets

◎ 비트상자와 종이비트

우유팩을 활용하여 (Fig. 3)과 같은 형태의 비트상자를 만든다. 상자 4개의 면에 각각 빨강, 파랑, 초록, 검정 색종이를 붙여 색상을 구분할 수 있도록 한다.



(Fig. 3) The bit box



(Fig. 4) The step 1 in activity

2단계) 첫 번째 패리티비트를 확인하기 위하여 1, 3, 5, 7번째의 상자를 파란색으로 맞추고 그 안의 비트를 꺼내어 확인한다.



(Fig. 5) The step 2 in activity

3단계) 두 번째 패리티비트를 확인하기 위하여 2, 3, 6, 7번째의 상자를 빨간색으로 맞추고 그 안의 비트를 꺼내어 확인한다.

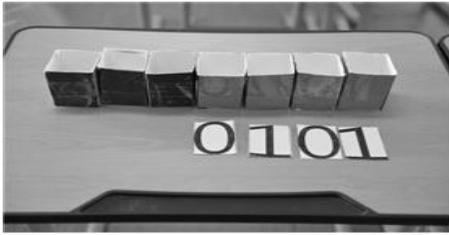


(Fig. 6) The step 3 in activity

【활동 단계】

1단계) 비트상자에 종이비트를 활동지(Fig. 2)에 맞춰 넣는다.

4단계) 세 번째 패리티비트를 확인하기 위하여 4, 5, 6, 7번째의 상자를 초록색으로 맞추고 그 안의 비트를 꺼내어 확인한다.



(Fig. 7) The step 4 in activity

5단계) 에러 검출 및 수정 확인을 위해 원자료 및 에러제시자(학생A)와 에러검출자(학생B)는 에러비트를 확인하고 검출과정을 설명한다.



(Fig. 8) The step 5 in activity

3.3 후속 학습

학습의 정리를 위하여 다음과 같은 요점을 정리 한다. 첫째, 카드마술에서 패리티 비트의 역할은 무엇인지를 찾아본다.

둘째, 에러 검출 알고리즘의 다른 형태를 찾아보고 간략화된 기호로 표시해본다.

셋째, 학습과정에서 한 개 이상의 에러가 발생할 경우의 수정방법을 생각해 본다.

4. 교육자료 실험의 분석

4.1. 적용대상 및 검사도구

본 연구는 교육부의 소프트웨어운영지침(2015)에 따른 컴퓨터과학 교육의 현장 적용자료 개발을 위한 언플

러그드 중심 교수법과 교육자료의 개발 과정에서 데이터 오류의 검출과 수정을 위한 교육자료를 개발하였다. 실험 대상은 파일럿 실험 중인 점을 고려하여 CS Unplugged 교육을 수행 중인 제주도의 특정 초등학교에서 실행되었으며, 초등 고학년인 5~6학년 20명을 대상으로 실시되었다. 또한 초등학생을 대상으로 하는 소프트웨어교육 행사(언플러그드 데이, 2017.9)에서 초등 4학년 이상의 불특정 다수의 학생(200여명)에게도 수행되었으며, 컴퓨터교육관련 전문가와 교사들에 의해 학습의 진행은 관찰되었다. 따라서, 교육자료의 제작과 학습은 8월부터 12월까지 지속적인 피드백을 통해 진행되었다(Table 4).

<Table 4> Summary of Survey

Category	Contents
Objects	elementary students Grade 5~6
No. of sample	class: 20 event: about 200
sampling technique	Quota Sampling
sampling period	2017.8~12

분석대상으로 학교 수업에서 시행한 학생들과 소프트웨어교육 전문가 및 정보 담당 교사의 의견을 바탕으로 교육내용의 적절성, 교재의 난이도 및 수업에 활용가능여부를 조사하였으며, 주요 설문 문항분석 결과는 <Table 5>와 같다. 검사도구로는 자체 개발한 설문지와 수업관찰을 시행하였다.

<Table 5> The Results of Survey

Question-naire	Pre-Experiment	Post-Experiment N(%)*	Change rate
I became interested in unplugged educational activities.	15(75%)	18(90%)	+15%
Error Detection and Correction The contents of the activity are well understood.	11(55%)	16(80%)	+25%
The concept of parity bits can be described.	15(75%)	16(80%)	+5%
I could see how the parity bit works on a real computer.	4(20%)	12(60%)	+40%

(* N:Affirmative Size, %: Affirmative Rate)

4.2. 결과분석

본 연구에서 사전, 사후 응답결과 분석은 모든 항목에서 매우 뚜렷한 향상을 보였다. 그 중에 특히 ‘패리티비트가 실제 컴퓨터에서 어떻게 작동하는지 알 수 있었다’는 항목에서 확연한 차이(+40%)를 보였으며, 반면에 ‘패리티비트의 개념에 대한 이해’ 수준에서 비교적 향상된 수준의 변화(+5%)는 개발된 교육자료가 학습자의 학습내용에 대한 이해도를 변화시키는데 영향을 준 것으로 나타났다.

이러한 원인은 전문가와 정보담당교사의 자문을 통한 분석에서 활동중심실험교육이 학습자의 이해를 높이는 데 기여하고 있으며 개념의 이해를 위한 시간적 여유가 부족한 것으로 분석되었다.

5. 결론

본 연구는 소프트웨어운영지침(교육부,2015)에 따른 언플러그드교육 활동을 위한 교수법과 교육자료로 개발하였다.

컴퓨터과학 교육용 중에 에러 검출과 수정에 대한 교육자료는 대표적으로 팀벨의 2인[14]에서 제시된 ‘패리티비트를 활용한 카드마술’이 있다. 그러나, 이 교육자료는 저학년용(7세용)으로서 우리나라 교육과정(5~6학년)에 따른 초등 고학년을 위한 교육자료의 개발을 위해 ‘해밍코드를 활용한 에러 검출과 수정’ 교육자료의 개발을 목표로 하였다.

본 연구에서 개발된 교육자료의 특징은 활동중심학습으로 ‘마술카드’에서 제시된 에러 검출과 수정 활동들을 기반으로 하여 ‘해밍코드’를 사용해 실제 컴퓨터에서 패리티비트가 어떤 과정을 통해 에러를 찾고 수정하는지 실제 눈으로 확인 할 수 있도록 제시하였으며, 교육자료의 현장적용 실험에서 에러 검출과 수정에 대한 이해도가 매우 크게 높아지는 것으로 나타났다<Table 5>.

컴퓨터과학 원리를 교육하기 위한 언플러그드교육은 비형식적 교육방법으로 본 연구에서 개발된 교육자료는 다양한 방법으로 변형이 가능하며, 이를 응용한 사례연구가 폭넓게 이루어져 현장교육에 적합한 컴퓨터과학교육이 개발 될 수 있기를 기대한다.

참고문헌

[1] ByoungRae Han (2013). The Research of Unplugged Computing Method for Computational Thinking in Elementary informatics Education. *Journal of The Korean Association of Information Education*, 3(2), 159-167.

[2] Denning P. J. (2009). Beyond Computational thinking. *Communications of the ACM*, 52(6), 28-30.

[3] Gagne R. M. (1963). The learning requirements for enquiry. *Journal of Research in Science Teaching*, 1(2), 144-153.

[4] HyeongCheol Ko, ChongWoo Kim (2016), Development of Finite State Automata Learning Materials for Elementary School Students. *Journal of The Korean Association of Information Education*, 20(4), 401-408.

[5] Hyunbe Kim, Kapsu Kim (2014). A Study on the Achievement Goals, Teaching-Learning Methods, and Evaluation Methods in Computer System Education. *Journal of The Korean Association of Information Education*, 18(1), 195-202.

[6] Ata Elahi (2005). *Data, Network, & Internet Communications Technology*. Thomson Learning.

[7] Berry, M. (2013). *Computing in the national curriculum. A guide for primary teachers*. Bedford: Computing at School.

[8] CSTA (2011). *Computational Thinking in K-12 Education Teacher Resources*, 2nd.

[9] HyeongCheol Ko (2009). *The Development of the Information Discretion Activity in the elementary school teaching materials for the creative problem-solving ability*. Jeju National University, MS Thesis.

[10] NRC (2010). *Report of a Workshop on The Scope and Nature of Computational Thinking*. The National Academies Press, National Research Council. National Research Center.

[11] ____ (2011). *Report of a Workshop of Pedagogical Aspects of Computational Thinking*. The National

Academies Press. National Research Center.

- [12] *Network Hamming Code*. Retrieved January 15, 2018, from <http://eastroot1590.tistory.com/entry/%EC%98%A4%EB%A5%98-%EA%B2%80%EC%B6%9C-%EC%BD%94%EB%93%9C-%ED%95%B4%EB%B0%8D%EC%BD%94%EB%93%9C-Hamming-Code>
- [13] Code.org. Retrieved September 15, 2015, from <https://code.org/>
- [14] Bell T. C., Witten I. H. & Fellows M. (2015). Computer Science without a computer. Retrieved September 15, 2017, from <http://csunplugged.org/>
- [15] *National curriculum in England: framework for key stages 1 to 4.* Retrieved October 10, 2014, from <https://www.gov.uk/government/publications/national-curriculum-in-england-framework-for-key-stages-1-to-4/the-national-curriculum-in-england-framework-for-key-stages-1-to-4>.
- [16] Claude Shannon. *Wikipedia*. Retrieved January 15, 2018, from https://ko.wikipedia.org/wiki/%ED%81%B4%EB%A1%9C%EB%93%9C_%EC%84%80%EB%84%8C

저자소개

고형철



2004. 제주교육대학교(학사)
 2009. 제주대학교 교육학석사
 2017. 제주대학교 대학원 컴퓨터교육전공 박사과정
 2004~현재 초등학교 교사
 관심분야 : 컴퓨터교육, 프로그래밍 Computational Thinking, CS Unplugged
 e-mail: sseven007@naver.com

김종우



1997 동국대학교 대학원졸업
 전산통계(이학박사)
 1989~ 현재: 제주대학교교육대학
 초등컴퓨터교육전공 교수
 관심분야: 컴퓨터교육,
 computational thinking
 e-mail: woo@jejunu.ac.kr