

초등학교에서 코두를 활용한 프로그래밍 교육이 문제해결력에 미치는 영향

홍태경 · 전석주

서울교육대학교 컴퓨터교육과

요 약

본 연구는 초등학생들을 대상으로 코두를 활용한 프로그래밍 수업이 문제해결력에 미치는 영향에 대해 조사한다. 어린 학생들이 컴퓨터 프로그래밍을 좀 더 쉽고 편하게 배울 수 있도록 다양한 도구들이 개발되었는데 그 중에서도 코두는 게임, 애니메이션과 음악 등과 같은 멀티미디어 응용을 개발하도록 만들어진 도구이다. 코두를 이용한 프로그래밍 수업을 통해 학생들이 자신들의 게임과 애니메이션 및 음악을 만들면서 문제해결력이 향상되는 것을 기대할 수 있다. 본 논문은 코두를 활용하여 초등학생들을 위한 프로그래밍 교육 프로그램을 개발하고 이를 서울시 소재 초등학교 5학년 학생 27명을 대상으로 수업을 적용하였다. 2014년 3월부터 5월까지 총 3개월간의 프로그래밍 수업을 통해 대부분의 학생들이 수업에 흥미를 가졌으며 수업 후에 문제해결력이 향상되는 결과를 보였다.

키워드 : 프로그래밍 교육, 코두, 문제 해결력, 계산학적 사고, 초등학교

The Effects of Programming Education with KODU on Problem-Solving Abilities in an Elementary School

Taekyung Hong · Seokju Chun

Dept. of Computer Education, Seoul National University of Education

ABSTRACT

This study investigates the effectiveness of computer programming with KODU on problem-solving abilities of elementary school students. Diverse tools were developed for novice programmers to become more easy and comfort in learning a computer programming. Among them, KODU has focused on facilitating the programming of various multimedia applications to create games, animations and music videos etc. It is expected that KODU will help elementary school students to improve their problem-solving abilities through creating their own games, animation, and music etc. In this paper, we developed the educational program using KODU for teaching the elementary school students. The developed programming course was offered at the 5th grade students of a elemen-

이 논문은 2014년도 서울교육대학교 교내연구비에 의하여 연구되었음.

교신저자 : 전석주(서울교육대학교 컴퓨터교육과)

논문투고 : 2014-09-05

논문심사 : 2014-09-09

심사완료 : 2015-02-17

tary school in Seoul and a total of 27 elementary school students were participated in our programming courses. Most of the students stated that the course was interesting, and their problem-solving abilities were improved after the course from March 2014 through May 2014.

Keywords : Programming Education, KODU, Problem-Solving Ability, Computational Thinking, Elementary School

1. 서론

현재 많은 나라에서 프로그래밍의 조기교육이 강조되고 있어서 어린 학생들에게 프로그래밍의 개념과 기술을 교육하는 것을 필수 교육과정으로 정하고 있다[7]. 이에 우리나라 정부도 소프트웨어 교육을 정규과목으로 교육과정에 도입하려는 움직임을 보이고 있다. 프로그래밍 교육을 도입하는 것은 융합적인 사고를 하는 창의적 인재 양성을 목표로 하는 국가교육 정책에도 부합하는 것이다[19].

프로그래밍 교육은 학습자에게 여러 가지 장점을 제공할 수 있는데 학습자는 프로그래밍 과정에서 문제해결을 위한 체계적인 접근에 의해 규칙을 찾아낼 수 있으며 논리적 흐름을 고려한 프로그래밍과 오류를 수정하는 일련의 과정에서 계산학적 사고(Computational Thinking)력을 갖게 된다[12]. 계산학적 사고는 컴퓨터 과학에서 학생들이 반드시 배워야 하는 기초적인 기술들, 즉 문제형식화, 논리적 데이터 구성, 알고리즘사고를 통한 자동 문제 해결, 자료의 추상화 표현 등과 같은 기초적인 기술들을 모두에 적용된다[15].

초등학생들이 쉽게 이해할 수 있고 흥미를 갖고 배울 수 있는 교육용 프로그래밍 언어들에 관한 연구들은 이미 활발하게 이루어져 왔다. 그러나 초등학교 교육과정에서 정보영역이 차지하는 비중이 크지 않고 교사들의 관심과 소양 부족, 열악한 학교 컴퓨터 시설과 같은 이유로 실제로 프로그래밍 교육을 접하는 경우는 정보영역과 같은 일부 학생들에 그치고 있는 실정이다.

코두(Kodu)는 간단한 명령어와 시각적인 프로그래밍을 통해 창의성과 논리, 통합에 관해 자연스럽게 학습이 이루어지도록 하는 3D 게임 제작 언어이다[2]. 이 언어에 대한 연구는 Microsoft사의 코두 랩[18]에서 제공하고 있는 기술적인 내용들이 주를 이루고 있으며, 국내외에서 코두를 활용한 연구들이 나오고 있으나 구체

적인 교수학습 방법에 대한 연구는 아직 미비한 상태이다[2]. 게임을 제작하는 것은 창의적 사고력, 문제해결 능력, 스토리텔링이 중요한 요소로 작용하는 것으로 학습자에게 창의적 학습을 위한 주제로 적합하기도 하다. 많은 학생들이 흥미를 갖고 있는 게임이라는 주제로 간단한 명령어를 통해 프로그래밍을 할 수 있다는 점에서 프로그래밍 학습을 위한 도구로서 가능성이 있다고 판단된다.

본 연구에서는 코두를 이용하여 초등학교에서 프로그래밍 교육을 하기 위한 학습과정을 개발하고 실제로 적용해 보고자 한다. 코두가 프로그래밍 교육 도구로서 가능성이 있는지 살펴보고, 이를 이용한 초등학생들을 위한 프로그래밍교육 프로그램을 개발하여 적용해 본 후 학습자의 만족도와 문제해결력 향상을 검증하고자 한다.

2. 이론적 배경

2.1 교육용 프로그래밍 언어

그동안 프로그래밍 초급자들이 프로그래밍을 익히는데 있어서 문법적인 어려움으로 인해 발생하는 어려움을 해결하기 위해 비주얼 프로그래밍 방식의 다양한 교육용 프로그래밍 언어들에 개발되었고 이를 통해 프로그래밍 초급자들이 코드를 입력하는 노력들을 줄여주었다[8][15]. 그중에서도 특히, 스크래치[10]와 코두[2][18]는 다양한 멀티미디어 응용을 프로그래밍할 수 있도록 하여 게임, 음악, 뮤직비디오 등을 쉽게 만들 수 있도록 했다. 또한 엘리스[17]는 스토리텔링 기반으로 Sims[1]와 같은 유명 게임 캐릭터를 활용하거나 또는 직접 캐릭터를 설계하는 도구를 제공하여 응용을 개발할 수 있도록 제공하고 있다[9]. 또한 최근에 영국에서는 MIT에

서 만든 클럽하우스를 본떠 초등학교 방과 후 학교 프로그래밍 교실 네트워크인 코드클럽을 운영하고 있다. 코드클럽은 코드를 활용하여 학생들에게 프로그래밍에 대한 흥미와 디지털 창작물을 만들면서 얻는 성취감을 불어넣어주는 교육을 실시하고 있다[14].

교육용 프로그래밍 언어는 교육의 목적으로 이용되므로 문제해결을 위한 가설을 세우고 수행할 수 있는 학습 환경을 제공한다. 즉각적인 피드백이 이루어지며, 학습자는 프로그래밍 과정에서 자신이 학습의 주체로서 주어진 문제를 이해하고 분석하며 문제 해결 수행을 위한 방법을 찾아내고 오류를 고쳐나가는 과정에서 창의적인 문제해결력을 기를 수 있다. 또한 프로그래밍 과정을 탐구하며 학습자의 메타인지 능력 향상을 도모할 수 있다[6].

이러한 교육용 프로그래밍 언어의 학습은 교육학의 구성주의적 교수학습 이론에 부합하는 형태의 학습으로 볼 수 있다. 즉 창의적이고 생산적인 능력을 요하는 현대사회에서 학습자가 능동적으로 지식을 구성하고 창조하며 활용해 나가는 능력을 기를 수 있는 학습 방법으로 볼 수 있다. 어떤 언어를 선택하는지는 향후 학습자의 프로그래밍 인식 수준에 중요한 영향을 미칠 수 있으므로 교수자의 선호도나 대중적인 인지도, 시대적인 패러다임에 따라서 선택하기보다는 분명한 목표와 기준이 있을 때 교육적인 효과가 클 것이다. 교육용 프로그래밍 언어들을 선택하는 기준은 여러 가지가 있겠지만 크게 컴퓨터과학 연계성, 명확하고 강력한 명령어, 편리한 개발환경을 들 수 있다.

2.2 관련연구

최근에 코드를 이용한 프로그래밍 교육에 대한 연구와 컴퓨터에 기반 하여 학생들의 문제 해결력을 높이기 위한 다양한 연구들이 수행되었다[2][3][4][11][16].

정영식 등[16]은 코드를 활용한 게임 프로그래밍 수업이 학생들의 프로그래밍에 대한 이해를 넓혀주고 더 나아가 게임을 하는 것보다 게임을 제작하는 것과 같은 일을 더 좋아하게 되는 계기가 될 것으로 전망했으며 코딩기반의 프로그래밍 교육이 초등학생들의 문제해결력과 창의력 향상에 기여할 수 있는지에 대한 지속적인 연구의 필요성에 대해 언급하였다. Fowler[2] 등은 프로

그래밍의 개념을 학습하기 위한 동기를 높이기 위한 방법으로 게임을 학교 수업에 적용하였다. 이를 위해 제공된 코딩기반 게임개발 환경이 학생들의 컴퓨터 프로그래밍 수업에 대한 흥미를 유발하고 창의성을 높였다는 실험적인 검증을 실시하였다. Areti Panaoura[3]는 어린 학생들이 인터랙티브 컴퓨터 애니메이션 도구를 활용하여 교사의 직접적인 참여 없이 학생들에게 코멘트, 힌트 및 피드백을 주는 방법을 제안하였다. 학생들은 문제 해결과정에서 만나는 다양한 장애물과 어려움을 컴퓨터 만화 애니메이션과 소통하면서 자연스럽게 자기조절 능력과 수학에서의 문제해결력을 향상할 수 있다. Wang[11] 등은 학생들의 잘못 구조화된 문제해결 능력을 개선하기 위해 3차원 기반 가상기업(3D Virtual Company)을 설계하였다. 사전테스트, 3D 가상기업훈련, 사후테스트의 3단계로 절차를 통해 3차원 기반의 학습 환경이 학생들의 잘못 구조화된 문제해결 능력을 개선하는 데 유용하다고 주장하였다. Pola[4] 등은 컴퓨터 기반 힌트가 학생들의 문제 해결력에 얼마나 효과가 있는지를 조사하였다. 특히, 물리 문제를 푸는 과정과 푼 후에 이러한 간접적인 디지털 힌트(도움)가 문제해결력에 상당히 도움을 준다고 입증하였다.

3. 코딩 교육 프로그램 설계

3.1 프로그래밍 교육요소 추출

‘초·중등학교 정보통신기술교육 운영지침’의 초등학교 5, 6학년을 대상으로 하는 3단계 내용체계 중 ‘정보처리의 이해’ 영역을 바탕으로 하여 초등학생을 대상으로 코드를 지도할 때 고려해야 할 프로그래밍 학습요소를 다음과 같이 추출하였다. 추출한 프로그래밍 학습요소는 순차, 조건문, 반복, 및 변수이다. 코딩프로그래밍의 교육요소의 개념과 적용 방법은 다음과 같다.

<순차>

설명: Rule은 아이콘을 배열한 순서에 따라 실행되며, 각 행은 위에서 아래의 순서로 실행된다.

적용: 명령이 만들어지는 순서를 이해하고 그에 따라 아이콘을 배열할 수 있다.

4 한국정보교육학회 논문지 제19권 제1호

<조건문>

설명: 코두의 모든 Rule은 ‘~할 때, ~하라’로 이루어져 있다.

적용: When절과 Do절에 적용할 수 있는 명령들을 알고 상황에 맞게 사용할 수 있다.

<반복>

설명: 끝내는 Rule을 만들지 않을 경우 일련의 명령들을 반복되게 할 수 있다.

적용: 명령을 끝내야 하는 경우와 그렇지 않은 경우를 구분하여 아이콘을 배열한다.

<변수>

설명: 코두에서 변수는 점수(score)로 나타난다.

적용: 점수 구현하기

‘초·중등학교 정보통신기술교육 운영지침’의 3단계 ‘정보처리의 이해-프로그래밍의 이해와 기초’를 중심으로 12차시에 걸쳐 초등학교 5학년에게 지도할 수 있는 단계별 학습 목표는 <Table 1>과 같이 구성한다.

<Table 1> Learning contents in the kodu programming step

Step (hours)	Learning Goal
Understanding (1h)	-KODU Lab. understanding -To understand the menus and interface of KODU
	-When-Do Statements -‘Object’ concepts and Coding
Learning Functions (4h)	-To understand the interaction between objects -To code a simple game program
	-Variable concepts -To create Virtual World
	-To know the basic elements for racing games and then design the game
Game Design (4h)	-To know the basic elements for shooting games and then design the game
	-To understand the basic elements for side scrolling games and then design the game
	-To understand the basic elements for maze games and then design the game

Game Construction (1h)	-To plan and design diverse games with originality concepts
Test (2h)	-To test the peer’s games and then grade their peers

3.2 학습자 분석

코두를 이용한 프로그래밍 교육을 실시하기 전에 학습자 분석을 위하여 서울시 소재의 M초등학교 5학년 27명을 대상으로 설문조사를 실시하였다. 설문조사에서는 학교에서 받고 있는 정보 교육에 대한 만족도, 학교 정보교육에서 배우고 싶은 내용, 소프트웨어 교육 이해도, 가정에서의 정보 교육 환경 등을 조사하였다. 설문지 분석 결과 대부분의 학생들이 집에서 컴퓨터를 활용할 수 있는 환경이며 정보 교육에 흥미가 있음을 알 수 있었다. 프로그래밍 언어를 접해본 경험이 없는 학생들이 대다수이므로 프로그래밍이라는 새로운 학습에 흥미를 갖고 참여할 것으로 예상된다.

3.3 학습내용의 설계

코두 프로그래밍의 단계별 학습 내용을 바탕으로 구체적인 학습지도 과정을 설계하였다. 설문조사를 통해 얻은 학생들의 흥미도와 수준을 고려하여 프로그래밍 개념 요소들을 직접 학습하기보다 코두의 규칙들을 자연스럽게 습득하며 체득할 수 있도록 하는 점에 주안을 두었다. 또한 학생들이 실제로 플레이해 본 경험이 있는 게임들을 상상하며, 게임을 즐기는 입장과 개발하는 입장이 다름을 알고 창의성을 발휘하여 프로그래밍 할 수 있도록 돕고자 하였다.

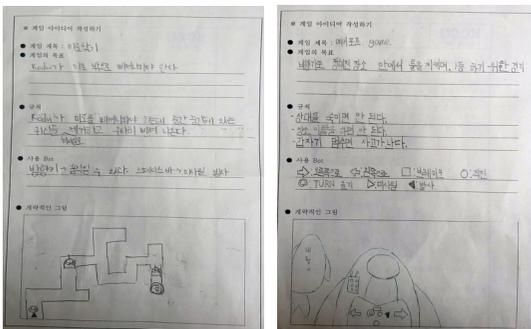
4. 코두 교육프로그램의 적용

4.1 학습설계

코두는 간단한 규칙과 시각적 인터페이스로 쉽게 배울 수 있는 프로그래밍 언어이지만 객체들의 특성이 달라 똑같은 명령을 입력해도 객체마다 실행이 되는 경우

가 있고 실행이 되지 않는 경우가 있어 각각의 특성과 성질을 알아야 게임 제작에 수월하게 임할 수 있다. 이에 학생들의 이해를 돕고 원활한 학습 진행을 위해 학습지에 해설과 함께 예제 화면을 제시하고, 뒷면에는 게임을 직접 제작하기 전에 제작에 필요한 요소들을 적어보고 전체적인 개요를 작성할 수 있도록 구성하였다.

수업시간에 제작한 게임은 학급 구성원들과 공유하여 서로 제작한 게임을 플레이해 보고 평가할 수 있도록 하였으며, 학급 SNS를 통해 코두와 관련된 내용과 새로운 정보를 공유하고 학급 친구들과 소감을 나누며 정보를 교류할 수 있도록 하였다. 학습지는 학습자들이 제작하는 게임의 목표와 사용되는 규칙, 객체들, 화면 구성 등을 생각해보게 하고 실제로 구현하는 데 도움을 주고자 제작하였다.



(Fig. 1) Game Idea Sheets

(Fig. 1)은 학습자들이 코두로 게임을 구현하기 이전에 게임 아이디어들을 작성한 학습지 예시이다. 학습지는 학습자들이 제작하는 게임의 목표와 사용되는 규칙, 객체들, 화면 구성 등을 생각해보게 하고 실제로 구현하는데 도움을 주고자 제작하였다.

코두언어의 기초를 학습하는 5차시까지는 개별학습으로 수업하였다. 실제로 게임을 제작해 보는 6차시부터는 학생들의 기호에 따라 개별 또는 그룹으로 실습할 수 있도록 하였다. 차시별로 제작 실습을 한 후에는 자신이 제작한 게임을 소개하고 사용한 규칙들을 어떻게 만들었는지 설명하는 시간을 통해 아이디어를 공유하고 상호 간의 자연스러운 평가가 이루어지도록 유도하였다. (Fig. 2)는 학습 및 결과물을 서로 발표하는 장면이다.



(Fig. 2) Learning and presentations

마무리 학습으로는 학생들의 창의력과 아이디어를 반영한 게임을 제작하는 활동을 통해 학습한 내용들을 실제 산출물로 만들어 내는 과정으로 구성하였다.

(Fig. 3)은 여러 가지 색깔의 'wisp' object를 자유롭게 움직이도록 명령하여 이용하여 도깨비불이 움직이는 모습을 구현한 것이다. (Fig. 4)는 배경을 바다 속으로 가정하여 미사일을 이용하여 물고기를 잡아 점수를 얻어 일정한 점수를 획득하면 게임에서 승리하는 게임을 구현한 것이다. 코두에서 제공하는 다양한 객체들의 특성과 명령어의 조합을 이용하여 여러 가지 방식의 게임들을 창작해 내는 결과를 얻을 수 있었다.



(Fig. 3) Students' work(1)



(Fig. 4) Students' work(2)

4.2 적용

4.2.1 연구 가설

본 연구에서는 초등학생의 프로그래밍 학습을 위하여 코두를 이용한 교육 프로그램을 개발하고 적용하였다. 코두를 이용한 프로그래밍 교육의 효과를 검증하기 위하여 프로그래밍 학습 전후 문제해결능력의 향상과 프로그래밍 흥미도에 변화가 있는지 확인하고자 하였다. 본 연구의 독립변인은 코두를 이용한 프로그래밍 교육이며 종속변인은 문제해결능력이다. 독립변인이 종속변인에 미치는 영향을 알아보기 위해 초등학교에서 ‘코두를 이용한 프로그래밍 교육은 문제해결력 향상에 유의미한 영향을 미칠 것이다’라는 가설을 설정하였다.

4.2.2 적용 기간 및 대상

적용 기간은 2014년 3월부터 5월까지 3개월이며, 매주 금요일 3교시 실과교과 시간에 앞서 개발한 총 12차시의 ‘코두를 이용한 프로그래밍교육 프로그램’을 실시한 후 결과를 분석하였다.

적용 대상은 서울시 구로구 소재의 M초등학교 5학년 학생 27명이다. 본 연구자가 답임을 맡고 있는 학급의 학생들을 실험집단으로, 사전 문제해결력 검사 결과가 동질 집단인 1개 학급은 통제집단으로 선정하였다. 연구대상의 구성은 <Table 2>와 같다.

<Table 2> Students participated in the course

SEX \ Group	Experiment Group	Control Group
M	11	13
F	16	14
N	27	27

4.2.3 연구설계

코두를 이용한 프로그래밍 교육의 효과를 검증하기 위해 실험집단과 통제집단을 구성하여 비교하였다. 본 연구자가 답임을 맡고 있는 학급의 학생들을 실험집단으로 하여 코두를 이용한 프로그래밍 교육을 12주에 걸

쳐 실시하고, 사전 문제해결력 검사 결과가 동질 집단인 1개 학급은 통제집단으로 선정하여 일반적인 정보교육(실과)을 실시하였다. 두 집단의 학습시간과 학습 환경은 동일하였다.

사전 검사를 통해 두 집단의 문제해결력을 측정하고, 12차시의 수업이 끝난 후 사후검사를 통해 ‘문제 해결 능력’과 ‘프로그래밍에 대한 흥미도’ 향상 정도를 사후 검사를 통해 측정하여 분석하였다. 본 연구에서 실시한 연구의 설계는 <Table 3>과 같다.

<Table 3> Research plan

Group	Pre-Test	Experiment	Post-Test
G1	O1 O5	X1	O3 O6
G2	O2		O4
G1 : Experiment Group		O1 O2: Problem Solving Ability (Pre-test)	
G2 : Control Group		O3 O4: Problem Solving Ability (Post-test)	
X1 : KODU Programming Class		O5: Programming Interest (Pre-test)	
		O6: Programming Interest (Post-test)	

4.2.4 연구도구

문제해결력 검사도구로 Heppner와 Peterson(1982)이 개발한 PSI(Personal Problem Solving Inventory)를 사용하였다. 본 연구에서는 김은희(1994)가 변안한 PSI 검사를 사용하였다. 문제해결력 검사는 세 가지 요인으로 구성되어 있다. 첫째, 문제해결 확신(Problem solving confidence)은 자신의 문제해결 능력에 대해 얼마나 긍정적으로 생각하는지 측정한다. 둘째, 접근 회피 스타일(Approach avoidance style)은 문제를 스스로 해결할 수 있다고 믿는 자신감의 정도를 측정하는 것으로 얼마나 문제에 적극적으로 접근하는지 측정한다. 셋째, 자신의 통제(Personal control)는 문제를 해결해 가는 데 있어서 정서적으로 얼마나 잘 통제하는지 측정한다. 이 검사의 신뢰도는 문제 해결 확신이 .85, 접근 회피 스타일은 .84, 자신의 통제는 .72로 만족할 만하다. 전체 문항의 내적 일치도가 .90으로 만족스러우며, 검사-재검사 신뢰도에서 상관계수 r은 각각 .85, .88, .83, .89로 유의미하다.

프로그래밍에 대한 흥미도 검사는 권재은(2012)의 연구에서 사용한 프로그래밍 언어 흥미도 조사 설문지를 바탕으로 본 연구와 관련된 설문으로 수정, 보완하여 실험집단에게 사전, 사후 검사를 실시하였다. 5점을 최고점으로 하여 총 10개의 문항으로 구성하였다[5].

4.3 분석

본 연구에 사용된 통계 프로그램은 김정성(2008)에 의해 개발된 i-STATistics 프로그램을 사용하였다. 두 집단의 문제해결력 사전, 사후 검사의 평균을 비교, 분석하여 결과를 도출하였다. 문제해결력 검사 결과는 <Table 4>과 <Table 5>에 제시되어진다.

<Table 4> The Result of Pre-Post Test (experiment group)

	Mean	SD	N	t	p
Pre-test	142	27.46	27	3.6809**	0.0011
Post-test	134	29.74	27		

**p<0.01

<Table 5> The Result of Pre-Post Test (control group)

	Mean	SD	N	t	p
Pre-test	135.41	27.06	27	1.3964	0.1744
Post-test	135.74	26.94	27		

코드를 이용한 프로그래밍 학습을 실시한 실험 집단의 사후검사 평균은 사전검사 평균보다 통계적으로 **p < 0.01 수준에서 유의미하게 높았다. 반면 일반적인 정보교육을 실시한 통제집단의 사후검사와 사전검사의 평균은 통계적으로 차이가 없는 것으로 나타났다. 코드를 이용한 프로그래밍 학습을 한 후 문제해결력이 향상되었음을 알 수 있었다.

학생들의 프로그래밍에 대한 이해와 흥미도를 알아보기 위하여 사전, 사후에 설문지를 이용하여 조사를 실시하였다. 웹사이트 code.org에서 제공하는 프로그래밍 기초학습인 ‘Hour of code’를 일부 학습한 후 사전 설문을 실시하였으며, 코두프로그래밍 학습을 진행한 후 사후 설문을 실시하였다.

학생들의 프로그래밍 흥미도에 관한 설문결과 총 10개의 문항 중 대부분의 문항에서 ‘매우 그렇다’와 ‘그렇

다’로 응답한 학생이 사전보다 사후검사에서 보다 많이 늘어났다. 이러한 결과를 통해 프로그래밍에 관한 흥미도가 이전보다 높아졌다고 할 수 있다.

5. 결론

프로그래밍 교육을 도입하려는 움직임들이 당장 프로그래머들을 기르기 위한 것을 목표로 하는 것은 아니다. 창의적인 생각을 할 수 있고, 문제를 자기 주도적으로 해결하며 성취의 기쁨을 느낄 수 있는 하나의 과정으로 프로그래밍을 교육을 도입하자는 것이다. 자신이 구상한 것을 실제로 제작해 보며 활용할 수 있으며, 그것에 대해 성취감을 느끼고 흥미가 생겨 점차 발전시켜 나갈 마음이 생긴다면 그 교육은 의미가 있는 것이다.

본 연구에서는 코드를 이용하여 초등학교에서 프로그래밍을 교육하기 위한 프로그램을 개발하고 적용해 보았다. 프로그래밍을 처음 접하는 학생들도 쉽게 학습할 수 있을 정도로 간단한 규칙을 갖고 있으며, 초등학교생들이 좋아하는 게임이라는 소재로 쉽게 흥미를 유발하는 점은 코드가 가진 큰 장점이었다. 학습 과정에서 자연스럽게 협동학습이 일어나고, 자신의 아이디어로 게임을 제작한다는 성취와 만족감이 높았다. 또한 코두 학습이 문제해결력 향상과 프로그래밍에 대한 흥미도 증가에도 긍정적인 효과가 있음을 알 수 있었다.

그러나 코두의 한계점도 분명히 있었다. 프로그래밍 요소들을 직접 이해하며 학습하기에는 부족한 점이 있고, 심화된 단계의 프로그래밍 언어들을 학습하기 위한 연계성도 부족했다. 다른 교육용 프로그래밍 언어와 비교했을 때 설치과정이 복잡하고, 운영체제에 따른 호환성이 낮아 오류가 잦은 문제는 극복해야 할 점이다. 게임을 사회악으로 여기는 사회 분위기의 변화도 필요할 것이다.

향후 프로그래밍 교육이 더욱 활발해지고 널리 보급된다면, 코두가 처음 프로그래밍을 접하는 도구로서 충분히 사용할 만한 가치가 있다고 여겨진다. 프로그래밍 학습 도구로서 코두가 더욱 널리 활용되기 위해서는 학습자들을 위한 장기적인 교육과정을 개발하여 검증하고, 코두에 대한 심화된 분석을 통해 다양한 활용방안을 모색하는 후속 연구가 필요할 것이다.

참고문헌

- [1] Reppenning, A. and Ioannidou, A. (2008). Broadening participation through scalable game design. In Proceedings of the 39th ACM Technical Symposium on Computer Science Education (SIGCSE 2008), 305-309.
- [2] Allan Fowler, Brian Cusack (2011). Kodu game lab: improving the motivation for learning programming concepts. Proceedings of the 6th International Conference on Foundations of Digital Games (FD 2011), 238-240.
- [3] Areti Panaoura (2012). Improving problem solving ability in mathematics by using a mathematical model: A computerized approach. *Computers in Human Behavior*, 28(6), 2291-2297.
- [4] Henk J. Pola, Egbert G. Harskampc, Cor J. M. Suhrea, and Martin J. Goedharta (2009). How indirect supportive digital help during and after solving physics problems can improve problem-solving abilities. *Computers & Education*, 52(1), 34-50.
- [5] Jae-Eun Keon (2004). Development and Application of a Teaching-Learning Model for Programming Language Courses, Master's Thesis, Andong University.
- [6] Jae-Hyung Kim, Jong-hoon Kim (2012). Design of kodu programming education for improving logical thinking of primary school students. *Proceedings of the KACE*, 16(2), 67-71.
- [7] Katherine Howland, and Judith Good (2015). Learning to communicate computationally with Flip: A bi-modal programming language for game creation. *Computers & Education(CE)*, 80, 224-240.
- [8] Kelleher, C. and Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37(2), 83-137.
- [9] Kelleher, C. and Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM*, 50(7), 58-64.
- [10] Resnick, M., Maloney, J., Hernandez, A. M., Rusk, N., Eastmond, E., Brennan, K., Millner, A. D., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. B. (2009). Scratch: Programming for everyone. *Communications of the ACM*, 52(11), 60-67.
- [11] Shwu-Huey Wang, Mei-Chung Lin, and Chin-Wen Liao (2014). A Virtual Experiential Learning and Students' Ill-Structured Problem-Solving Ability. *Interacting with Computers*, 26(4), 334-347.
- [12] Sungwon Kang et al. (2003). Programming Education for the Gifted of Elementary School Students in Information Science (Approach Using Visual Basic). *KAIE Journal*, 78(3), 363-371.
- [13] Stolee, K. T. (2010). Kodu Language and Grammar Specification. Microsoft research.
- [14] Stolee, K. T., and Fristoe, T. (2011). Expressing computer science concepts through kodu game lab. In Proceedings of the 42th ACM Technical Symposium on Computer Science Education(SIGCSE 2011), 99-104.
- [15] Yasmin B. Kafai, Quinn Burke (2013). The Social Turn in K-12 Programming: Moving from Computational Thinking to Computational Participation, In Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE 2013), 241-246.
- [16] Youngsik Jeong, and Jeongsu Yu (2014). The Effects of Elementary Students' Perception of the Game Programming Using Kodu. *Journal of KAIE*, 18(3).
- [17] <http://www.alice.org>. ALICE.
- [18] <http://www.kodugamelab.com>. Kodu Game Lab.
- [19] <http://www.yhn.com>. Yonhap News, Jan. 15th, 2015.

저자소개



홍 태 경

2014 서울교육대학교 교육대학원
컴퓨터교육학과 졸업

2010~현재 서울미래초등학교 교사

관심분야: 컴퓨터교육, 교육용프
로그래밍 언어

e-mail: arsenal86@naver.com



전 석 주

2002 한국과학기술원 전산학박사

2003 서강대학교 정보통신대학원
강사

2004~현재 서울교육대학교 컴퓨
터교육과 교수

관심분야: 컴퓨터교육, 프로그래
밍교육, 데이터마이닝

e-mail: chunsj@snue.ac.kr

