

Kodu를 이용한 프로그래밍 학습이 초등학생의 논리적 사고력과 학습 흥미에 미치는 영향

김태훈[†] · 김종훈^{††}

요 약

다양한 디지털 기술들은 현대인의 삶에 많은 변화를 가져왔다. 미래를 이끌어갈 청소년들이 디지털 사회에서의 경쟁력을 갖기 위해서는 프로그래밍 교육이 필요하다. 최근 프로그래밍 교육에 대한 관심이 높아지긴 했지만 초등학교에서 프로그래밍 교육이 실시되기에는 많은 어려움이 따른다. 본 연구에서는 초등학생들이 쉽고 재미있게 프로그래밍을 학습할 수 있는 Kodu를 이용한 교육 프로그램을 계획하여 실험하였다. GALT를 통한 논리적 사고력 검사의 사전·사후 비교 결과 실험집단의 상관논리, 논리 합계 부분에서 유의미한 효과를 얻었다. 또한 프로그래밍에 대한 학습 흥미 측면에서도 초등학생들에게 긍정적인 인식을 심어준 것으로 나타났다.

주제어 : Kodu, 프로그래밍 교육, 논리적 사고력, 학습 흥미

The Effect of Kodu Programming Learning on Logical Thinking and Learning Interest of Elementary Students

Tae-Hun Kim[†] · Jong-Hoon Kim^{††}

ABSTRACT

Digital technology brought about many changes which has changed the life of modern people. Adolescents need programming learning to increase a competitiveness what will lead the future. Recently though an interest about programming learning has raised, there are many trouble to take programming learning. In this paper, we planed and implemented a program that elementary students can learn programming using Kodu easily and excitingly. According to comparison of pre-to-post tests about logical thinking using GALT, correlational reasoning and total of experimental group were growing significantly. Also, elementary students had positive cognition in learning interest about programming.

Keywords : Kodu, Programming Learning, Logical Thinking, Learning Interest

† 정 회 원: 제주대학교 컴퓨터교육전공 박사과정
†† 중신회원: 제주대학교 초등컴퓨터교육전공 교수(교신저자)
논문접수: 2012년 11월 13일, 심사완료: 2013년 03월 29일, 게재확정: 2013년 04월 03일

1. 연구의 필요성

21세기 교육이 지향해야 할 것은 채팅, 검색을 하는 능력이 아니라 새로운 미디어를 가지고 디자인하고 만들어내며 발명해내는 능력이다. 이를 위해서 청소년들은 프로그래밍을 학습해야 할 필요가 있다. 프로그래밍 능력은 학습자에게 중요한 이점을 제공한다. 먼저, 컴퓨터로 만들 수 있는 범위를 확장해준다. 또한 학습능력을 확대시켜준다. 특히, 프로그래밍은 계산적 사고(Computational Thinking)를 길러주고, 중요한 문제해결 및 전략의 설계 능력을 학습하도록 도와주어 프로그래밍 이외의 영역까지 확장시켜준다. 그리고 프로그래밍은 문제해결과정의 외부 연출을 만드는 것을 포함하기 때문에, 프로그래밍은 자신의 생각을 돌아볼 수 있는 기회를 주고, 생각 그 자체를 생각하도록 도와준다[1].

초등학생부터 프로그래밍 교육이 이루어진다면 이러한 장점을 더욱 효과적으로 얻을 수 있을 것이다. 최근 초등학생 대상의 프로그래밍 교육에 대한 다양한 연구가 이루어지고 있는데 교육용 프로그래밍 언어를 활용한 교육방법이 대표적이다. 교육현장에서 많이 사용되는 교육용 프로그래밍 언어로는 Scratch, Doolittle, Logo 등이 있다. 하지만 이러한 언어들은 주로 중학생을 대상으로 하거나 초등학생의 경우에는 정보영재 학생들을 대상으로 교육이 이루어지고 있는 실정이다. 프로그래밍의 경험이 없고 계산적 사고가 낮은 초등학생들이 프로그래밍을 익히기 위해서는 기존의 교육용 프로그래밍 언어보다 좀 더 쉽고 재미있게 접할 수 있는 언어가 필요하다. 이에 본 연구에서는 초등학생들이 쉽고 재미있게 프로그래밍을 학습할 수 있는 프로그래밍 도구로 Kodu를 선정하여 교육 프로그램을 계획하고 투입하였다. 또한 교육 프로그램 전후의 논리적 사고력의 변화와 학습 흥미에 대한 설문결과를 분석하여 Kodu를 활용한 프로그래밍 학습의 효과와 프로그래밍 교육의 도구로서 Kodu의 가능성을 점검해 보았다.

2. 이론적 배경

2.1 프로그래밍 교육과 논리적 사고력

정보 교과는 정보 과학 기술의 기본 개념과 원리를 이해하고, 실생활의 다양한 문제를 관찰하고 해결하는 능력을 향상시키는데 두고 있다. 이러한 문제해결능력의 향상을 위해 프로그래밍 교육의 필요성이 강조되고 있다[2].

프로그래밍 교육은 다양한 측면에서 학습자에게 이득을 준다. 프로그래밍 학습은 학생들이 컴퓨터를 활용하여 다양한 것을 창조해 낼 수 있도록 그 범위를 확장시켜준다. 또한 최근 정보 교육에서 강조하고 있는 계산적 사고를 기반으로 문제해결 능력, 창의력, 논리적 사고력 등을 길러준다[1].

프로그래밍은 논리적 사고력과 밀접한 연관을 갖는다. 프로그래밍 과정에서 문제 해결의 절차를 마련하거나 문제해결방법 및 과정의 옳고 그름을 판단하기 위해서 논리적 사고력을 필요로 한다[3]. 윤일규(2010)는 정보교과에서의 문제해결과정과 논리적 사고력과의 상관관계를 살피고 서열화 논리, 명제 논리, 조합 논리, 비례 논리, 상관 논리, 변인 통제 논리를 정보 교과의 문제해결과정에서 요구되는 논리적 사고력의 구성요소로 선정하기도 하였다[4].

프로그래밍 학습을 통해 학습자의 논리적 사고력을 신장시킬 수 있다는 연구가 많이 진행되었고 그 연관성이 입증되었다. 많은 연구에서 프로그래밍 교육에 대한 효과성을 검증하기 위해 논리적 사고력의 변화 정도를 평가도구로서 사용하고 있으며 본 연구에서도 Kodu를 이용한 프로그래밍 교육의 효과성을 검증하기 위하여 논리적 사고력의 변화를 살펴보았다.

2.2 Kodu(Kodu Game Lab)

Kodu는 프로그래밍이나 디자인에 대한 지식이 없는 어린 학생들도 게임을 만들 수 있도록 Microsoft Fuse Labs에서 개발한 프로그래밍 언어이다. 프로그래밍 모델을 단순화시킴으로써 프로그래밍을 처음 시작하는 학생들이 느낄 수 있는 프로그래밍 학습에 대한 장벽을 낮추고자 하였다. 모델이 단순하여 쉽게 배울 수 있지만 언어에서 지원하는 프로그래밍 표현을 활용하여 학생

들이 원하는 다양한 상황의 게임을 만들어 낼 수 있다. 실시간 3D 게임 환경에 통합되어 있고 콘솔 게임 수준의 높은 그래픽 수준을 보여주며 직관적인 사용자 인터페이스를 갖고 있어 어린 학생들의 흥미를 끌기에 적합하다.

Kodu는 기존에 사용되었던 교육용 프로그래밍 언어와 비교하였을 때 특별한 강점을 갖는다. 본 연구에서는 이러한 강점을 살린 프로그래밍 학습을 진행하기 위해 프로그래밍 언어로 Kodu를 선택하게 되었는데 Kodu가 가지는 강점은 다음과 같다.

먼저 초등학교 학생, 특히 저학년 학생들까지 쉽게 프로그래밍을 접할 수 있다. 뛰어난 그래픽 수준과 타일 형식의 인터페이스는 학생들의 흥미를 이끌어 내기 쉽고 복잡한 명령어를 입력하는 대신에 타일을 배치하여 조건과 실행문을 조합시켜 로봇에게 명령을 내리는 과정에서 프로그래밍에 대한 두려움을 최소화 할 수 있다.

둘째, 실제 물리 환경이 투영된 가상의 3D 물리엔진 구현되어 있기 때문에 실제로 일어날 수 있는 문제 속으로 학생들을 이끌어 들일 수 있다. 현실성 있는 문제의 해결을 위해 어린 학생들이 논리적인 사고를 이용하여 문제를 작은 단계로 나누고 직접 설계한 과정에 따라 반복할 수 있도록 도와준다. 이는 계산적 사고의 핵심 요소인 추상화, 자동화화도 밀접한 관련이 있다[5].

셋째, 스토리에 맞는 지형을 창의적으로 디자인하고 로봇과 오브젝트를 사용하여 스토리텔링할 수 있다. Kodu에는 학생들의 스토리텔링을 위한 특정한 센서와 동작기가 존재하여 스토리텔링에 적합하다. 이는 학생들로 하여금 스토리를 만들어 내는 창의적인 활동에 대한 욕구를 불러일으킨다.

2.2.1 Kodu를 이용한 프로그래밍

Kodu는 Logo, Alice, Greenfoot, Scratch 등의 영향을 받아 제작된 언어로서 고급 언어이고 비주어 언어인 동시에 인터프리터 언어이다.

프로그래머는 각 캐릭터를 프로그래밍하여 World라는 가상공간에서 다른 지능 에이전트(로봇, 나무 등)와 어떻게 상호작용할 것인지를 설명한다. 캐릭터의 프로그래밍은 12개의 페이지에서 이루어지는 데 복수의 페이지에 서로 다른 상태의 캐릭터

를 정의하여 다양한 동작을 하도록 프로그래밍 할 수 있다. 각 페이지는 다른 프로그래밍 언어의 문장(statement)에 해당하는 규칙(Rule)들의 집합으로 구성되어 있고 각각의 규칙은 상태(Condition)와 행동(Action)의 결합 형태로 이루어진다. 상태와 행동은 각각 When절, Do절과 결합하여 조건문과 실행문의 역할을 한다[6][7].

이러한 Kodu의 문법은 행위 기반 로봇의 원리가 반영되었다. 이벤트 기반 방식으로 로봇의 센서가 주변의 변화를 감지하면 로봇이 갖고 있는 동작기가 반응한다[6].

Kodu의 프로그래밍 인터페이스는 타일 형태의 아이콘을 배열하여 로봇에게 명령을 내리도록 구성되어 있다. 초등학교 학생이 사용하기에 충분할 정도로 단순하면서도 직관적이면서도 Kodu가 지원하는 페이지, Creatable 속성 등을 이용하여 변수와 영역, 불 논리(Boolean Logic), 객체 지향, 제어 흐름에 대한 프로그래밍과 개념 학습이 가능하다[7].

2.2.2 Kodu를 이용한 선행 연구

국내에서 Kodu에 대한 연구가 활발하지는 않지만 최근 프로그래밍 교육 도구로서 가능성을 찾고자 하는 연구가 진행 중이다.

김병수(2012)는 초등 예비교사를 대상으로 계산적 사고를 위한 알고리즘 학습에 대한 연구에서 프로그래밍 언어와 계산적 사고를 접하기 위한 도구로서 Kodu를 통한 알고리즘 학습을 진행하였다[8].

김태훈(2012)은 Kodu를 활용하여 STEAM 기반의 초등학교 학생 프로그래밍을 학습하는 교육 프로그램을 개발하는 연구를 진행하였다[9].

반면 해외에서는 국내에 비해 활발한 연구가 이루어지고 있다. 프로그래밍을 처음 접하는 학생, STEM 분야에 흥미가 떨어지는 여학생들을 대상으로 게임 프로그래밍을 통해 프로그래밍과 컴퓨터 과학 개념을 학습하는 연구가 진행되었으며 STEM 융합교육을 위한 게임 제작의 도구로서 활용되고 있었다[10].

3. 예비실험

3.1 예비실험 방법

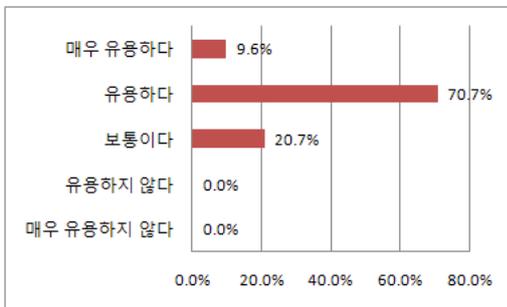
Kodu를 이용한 프로그래밍 교육의 장단점 및 개발한 교재의 효과를 예측해보기 위하여 개발한 학습 프로그램을 투입하는 예비실험을 진행하였다. 예비실험의 대상은 J대학교 2학년에 재학중인 초등 예비교사 58명으로 전체 수강자의 수강 이전부터 45%(26명)은 한 학기 동안 Logo를 활용한 프로그래밍 학습 경험이 있었다.

예비실험에는 Kodu의 기본 인터페이스와 문법을 익히면서 Kodu를 통한 프로그래밍을 접하였으며 이를 바탕으로 제시한 문제에 대한 프로그래밍을 학습자가 스스로 해결하도록 하였다.

실험 후에는 Kodu를 이용한 프로그래밍 교육에서 예상되는 효과를 묻기 위해 자체 제작한 설문지를 통해 예비교사의 의견을 들었다.

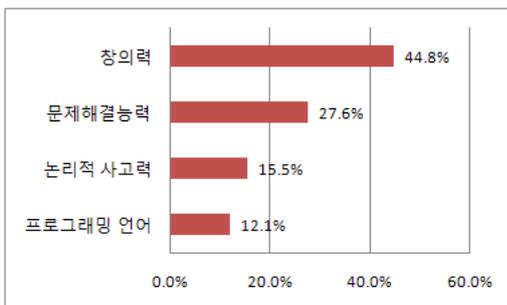
3.2 예비실험 결과

먼저 Kodu를 이용한 프로그래밍 학습의 교육적 가치를 평가해달라는 질문에 그림에서처럼 80% 이상의 응답자가 교육적으로 유용하다는 의견을 나타냈다.



<그림 1> Kodu의 교육적인 유용성 정도

Kodu를 이용한 프로그래밍 학습을 통해 신장될 것이라고 예상되는 능력을 묻는 질문에는 창의력과 문제해결능력을 예상하는 답변이 많았다.



<그림 2> Kodu를 통해 신장될 것이라고 예상되는 능력

Kodu를 이용하여 초등학생에게 프로그래밍 교육

을 할 때 예상되는 장점으로서는 아이들이 프로그래밍에 흥미를 느낄 것이라는 답변이 가장 많았고 주의점으로는 프로그래밍 학습 중 게임이나 그래픽 환경에 관심을 가질 위험이 있다는 것을 들었다.

예비실험을 통해 초등 예비교사들은 Kodu를 이용한 프로그래밍 학습의 교육적 가능성을 높게 보고 있음을 알 수 있었다. 다만 Kodu를 이용한 프로그래밍 교육이 효과를 위해서는 학생들이 흥미가 게임이나 그래픽 환경이 아닌 프로그래밍을 통한 문제해결에 집중될 수 있도록 학습이 진행될 필요가 있다는 결론을 얻었다.

<표 1> Kodu 프로그래밍 지도 시 예상되는 장점

답변내용	답변수
학생들이 프로그래밍에 흥미를 느끼게 될 것이다.	35 (60.3%)
직접 게임을 만들고 수정하는 과정에서 문제해결능력을 기를 수 있다.	17 (29.3%)
단순한 조작으로 쉽게 프로그래밍을 할 수 있다.	4 (6.9%)
단순한 문법을 활용해 문제해결에 집중을 할 수 있다.	1 (1.7%)

<표 2> Kodu 프로그래밍 지도 시 예상되는 주의점

답변내용	답변수
프로그래밍 학습 보다 게임이나 그래픽 환경에 관심을 가질 수 있다.	30 (51.7%)
게임 집중으로 인하여 학습에 대한 피드백이 어려워질 수 있다.	23 (39.7%)
표현이 단순하여 문제해결력 향상에 대한 효과가 적을 것이다.	2 (3.4%)
단순한 문법 때문에 프로그래밍 언어에 대한 효과가 적을 것이다.	1 (1.7%)

4. 연구 방법

4.1 연구 대상

본 연구는 제주특별자치도에 위치한 H초등학교 6학년 중 2개 학급 총 46명(각 23명)을 대상으로 진행하였다. 실제 프로그래밍 경험이 있는 학생은 거의 없었지만 대부분의 학생들이 프로그래밍이라는 용어는 알고 있었다. 2개의 학급 중 한 학급을 실험집단으로 선정하여 Kodu를 이용한 프로그래밍 교육을 실시하였으며 다른 한 학급은 비교집단으로 선정하여 초등학교 현장에서 많이 이루어지고 있는 컴퓨터소양교육 위주의 ICT 활용 수업을 진행하였다.

4.2 검사 도구

본 연구에서는 학생들의 논리적 사고력을 측정하기 위한 도구로 GALT(Group Assessment of Logical Thinking) 검사를 실시하였다. GALT 검사지는 많은 학습자의 인지발달 수준을 측정할 수 있는 지필 평가 형식의 검사도구로서 프로그래밍 학습의 효과를 살펴보기 위해 사용되고 있으며 다양한 연구에서 그 신뢰성이 입증되었다.

GALT 검사지는 총 6개의 하위 유형으로 구분되며 각각 보존논리, 비례논리, 변인통제 논리, 확률논리, 상관논리, 조합논리이다. 노정원(1998)의 연구에서 사용된 검사지를 원본의 의미를 손상시키지 않는 범위 내에서 학생들에게 친숙한 이름과 보충 설명을 추가하여 사용하였다[11]. 총 문항수는 21개이며 하위 논리유형에 따른 문항수와 문항번호는 <표 3>과 같다.

<표 3> 하위 논리유형과 문항수

논리유형	문항수	문항번호
보존논리	4	1, 2, 3, 4
비례논리	6	5, 6, 7, 8, 9, 10
변인통제논리	4	11, 12, 13, 14
확률논리	2	15, 16
상관논리	2	17, 18
조합논리	3	19, 20, 21

Kodu를 이용한 프로그래밍 학습 후 초등학생의 프로그래밍 학습에 대한 흥미가 어떠한지를 분석하기 위해 실험집단을 대상으로 연구자가 제작한 설문지를 투입하였다. 설문지에서는 Kodu를 이용한 프로그래밍 학습에 대한 흥미와 지속 의지, 다른 프로그래밍 언어에 대한 학습욕구, 프로그래밍에 대한 생각의 변화 등을 포함시켰다. 투입한 설문지의 신뢰도 계수(α)는 .646으로 나타났다.

4.3 연구 절차

Kodu를 활용한 프로그래밍 학습이 초등학생들의 논리적 사고에 미치는 영향을 알아보기 위해 학습 전 실험집단과 통제집단에 논리적 사고력 검사를 실시하여 두 집단이 동질집단임을 보이고, 실험 처치 후 논리적 사고력 검사를 실시하여 그

결과를 비교하였다. 또한 실험집단 학생들에게는 Kodu를 이용한 프로그래밍 학습이 프로그래밍 학습에 대한 흥미에 어떠한 영향을 미쳤는지 알아보기 위한 설문조사를 추가로 실시하였다. 연구 설계의 모형을 <표 4>와 같이 도식화하였다.

<표 4> 연구의 설계

실험집단	O ₁	X ₁	O ₂ , O ₃
비교집단	O ₁	X ₂	O ₂
O ₁ : 사전검사(논리적 사고력 검사)			
O ₂ : 사후검사(논리적 사고력 검사)			
O ₃ : 설문조사(프로그래밍에 학습에 대한 흥미)			
X ₁ : Kodu를 이용한 프로그래밍 학습			
X ₂ : 컴퓨터소양교육 위주의 ICT 활용 학습			

4.4 교육 프로그램 개발 및 투입

Kodu를 이용한 프로그래밍 학습을 위한 주제 및 활동을 선정하기 위하여 Kodu와 기존에 사용된 교육용 프로그래밍 언어를 이용한 프로그래밍 학습에 대하여 연구한 문헌을 검토하였다. Kodu를 이용한 프로그래밍의 학습 주제를 추출하였고 이를 투입하기 위해 총 10차시의 투입계획을 수립하였으며 이를 <표 5>에 제시하였다.

<표 5> 투입 내용

차시	학습주제	학습활동
1	Kodu와 인사하기	프로그래밍이란? 로봇에게 명령 내리기 프로그래밍 입력 방법 익히기
2	World 만들기	지형 메뉴 익히기 제두도 모양의 World 만들기
3	기본 명령어 익히기	로봇을 움직이는 방법 알아보기 다양한 센서와 동작기 다루기
4	Score를 사용하기	센서를 이용한 조건문 만들기 Score를 이용한 점수기능 익히기 변수로서 Score 사용하기
5	Timer 사용하기	Timer 이용하여 동작 제어하기 Random 이용한 프로그래밍하기
6	Creatable 사용하기	Creatable 기능 이해하기 Creatable 속성의 객체 로봇 만들기
7	Page 사용하기	Page를 사용하여 로봇 제어하기 다양한 동작을 수행하는 로봇 프로그래밍하기
8	로봇으로 대화하기	Hear 센서와 Say 동작기를 이용한 대화하기
9	디지털 스토리텔링하기	토끼와 거북이 이야기 프로그래밍하기
10	프로젝트 발표	학생이 직접 만든 프로그램으로 발표하기

수립한 계획에 따라 개발한 프로그램을 2012년 5월 1주부터 6월 4주까지 9주에 걸쳐 총 10차시의 재량활동 시간을 이용하여 투입하였다.

프로그래밍 경험이 없는 초등학생들에게 프로그래밍을 지도하기 위해서 Kodu 언어 자체에 대한 학습이 필요하지만 단순한 프로그래밍 언어의 학습만으로는 학생들의 논리적 사고력과 학습 흥미를 증진시키기 어렵다. 또한 예비실험에서 시사 받은 것처럼 학생들의 Kodu로 제작한 게임이나 Kodu의 그래픽 환경 등 단순한 흥미가 아니라 프로그래밍을 통한 문제해결에 집중할 수 있도록 하는 장치가 필요하다. 이러한 문제를 해결하기 위해 각 차시의 수업을 프로그래밍 과제를 제시한 도입 부분과 이를 해결하기 위한 핵심요소를 지도하는 부분, 이를 사용해 과제를 해결하는 부분, 마지막으로 학습한 내용을 활용해 개인별 창의적인 프로그래밍을 진행하는 단계로 구분하였다. 문제해결을 단계별로 제시함으로써 학생들이 문제해결에 집중할 수 있도록 하였다.

예를 들어 학생들에게 프로그래밍이 무엇인가를 설명하기 위해 로봇에게 ‘청소’라는 명령을 어떻게 설명할 지 고민해보도록 하였다. <그림 3>처럼 로봇은 물건을 감지하는 센서와 충돌을 감지하는 센서가 있고 이동하기, 물건을 집고 내려놓기 등의 동작을 수행할 수 있다.



<그림 3> 로봇에게 청소 명령내리기

학생들은 로봇이 갖고 있는 센서와 동작기를 이용하여 청소라는 명령을 설명한다. 이미 갖고 있는 표현과 명령을 이용하여 새로운 명령을 만들어내는 활동을 통해 프로그래밍을 도입하였으며 <그림 4>처럼 학생들의 의사코드를 직접 실제 Kodu에서 프로그래밍 하였다.



When got apple and see tree, move toward.
사과를 잡은 상태에서 나무를 보면 그 쪽으로 이동하라.
When see apple, move toward).
사과를 보면 그 쪽으로 이동하라.
When bump apple, grab it.
사과와 부딪히면 그것을 집어라.
When bump tree, drop.
나무와 부딪히면 물체를 떨어뜨려라.

<그림 4> 간단한 프로그래밍 예제

직접 프로그램의 배경이 되는 지형을 만들고 다양한 센서와 동작기를 사용해 로봇에게 원하는 명령을 내리는 과정을 통해 Kodu를 이용한 프로그래밍을 학습하였다. 정리활동에서는 각 차시에서 학습한 프로그래밍 개념이나 센서, 동작기를 활용하여 자신만의 프로그램을 만들었다. 프로그래밍 학습의 마지막 차시에는 각 차시에 만들었던 프로그램을 활용하여 최종 프로젝트로 디지털 스토리텔링을 위한 프로그램을 만들었다. 이야기나 평소 좋아하는 게임 등에서 생각해낸 주제를 직접 프로그래밍을 통해 구현하도록 하였다. Kodu는 각 로봇들의 서로의 역할을 수행하고 상호작용을 통해 프로그램이 전개된다. 예를 들어 토끼와 거북이에서 등장하는 토끼, 거북이, 나무, 응원단, 구름들에게 각자가 수행할 역할을 프로그래밍하는 과정에서 학생들은 자연스럽게 객체 프로그래밍의 개념을 접하게 된다.

5. 연구 결과

Kodu를 이용한 프로그래밍 교육이 초등학생의 논리적 사고력에 미치는 영향을 검증하기 위해

실험집단과 비교집단에 사전, 사후로 논리적 사고력 검사를 실시하였다.

먼저, 두 집단 사이에는 통계적으로 유의미한 차이가 있는지 알아보기 위해 집단 간 비교를 통해 실험 결과를 분석하였다. 그리고 각 집단별로 실험 처치 전과 후의 논리적 사고력에 어떤 변화가 있는지 분석하였다. 마지막으로 Kodu를 통한 프로그래밍 학습이 초등학생의 프로그래밍 학습에 대한 흥미 및 태도에 대한 미치는 영향을 묻는 설문조사를 통계 분석하였다.

5.1 논리적 사고력 변화

5.1.1 학습 집단 간 비교

실험 처치 전 논리적 사고력 평균 분석을 통해 실험집단과 비교집단이 동질 집단인지 알아보기 위해 논리적 사고력의 하위 요소별 평균에 대한 독립표본 t검정을 실시하였다.

<표 6>에서 제시한 사전검사의 결과에 따르면 논리적 사고력 각 하위 요소 및 합계에 있어서 통계적으로 유의미한 차이가 없는 것으로 나타났고 따라서 실험집단과 비교집단이 논리적 사고력에 대하여 동질집단임을 확인하였다.

<표 6> 논리적 사고력 사전검사 결과

하위요소	집단	N	M	SD	t	유의도
보존	실험	23	2.61	1.118	1.181	.186
	비교	23	2.26	.864		
비례	실험	23	1.48	1.675	.292	.137
	비교	23	1.35	1.335		
변인통제	실험	23	1.26	1.251	.115	.730
	비교	23	1.22	1.313		
확률	실험	23	.52	.790	.187	.896
	비교	23	.48	.790		
상관	실험	23	.13	.344	-.402	.423
	비교	23	.17	.388		
조합	실험	23	1.00	.905	1.249	.306
	비교	23	.65	.982		
합계	실험	23	7.00	4.068	.772	.774
	비교	23	6.13	3.559		

실험 처치 후 실험집단과 비교집단의 논리적 사고력의 각 하위 요소들의 변화가 통계적으로 유의미한지 알아보기 위해 독립표본 t검정을 실시하였으며 그 결과는 <표 7>과 같다.

<표 7> 논리적 사고력 사후검사 결과

하위요소	집단	N	M	SD	t	유의도
보존	실험	23	2.91	.733	2.755	.303
	비교	23	2.30	.765		
비례	실험	23	1.43	1.727	.179	.627
	비교	23	1.35	1.555		
변인통제	실험	23	1.26	1.322	.118	.635
	비교	23	1.22	1.166		
확률	실험	23	.48	.790	-.525	.264
	비교	23	.61	.891		
상관	실험	23	.35	.573	1.206	.018*
	비교	23	.17	.388		
조합	실험	23	1.78	.850	2.618	.897
	비교	23	1.09	.949		
합계	실험	23	8.22	4.210	1.246	.387
	비교	23	6.74	3.828		

*p<.05

전체 논리합계의 평균은 실험집단이 8.22점이고, 비교집단은 6.74점으로 실험집단이 높게 나타났으며 상승폭 역시 더 높은 것으로 나타났지만 통계적으로 유의미한 차이로 나타나지는 않았다.

논리적 사고력 하위 요소 중 상관논리 영역의 평균점수는 실험집단이 0.35점, 비교집단이 0.17점으로 실험집단이 높은 것으로 나타났고 상승폭 역시 더 높았으며 이는 통계적으로 유의미한 차이를 보였다.

보존, 비례, 변인통제, 조합논리에서 실험집단의 평균점수가 높았지만 통계적으로 유의미한 차이로 나타나지는 않았다. 확률논리에서는 오히려 비교집단의 평균점수가 높게 나타났으나 이 역시 통계적으로 유의미한 차이로 나타나지 않았다.

학습 집단 간 논리적 사고력 검사 비교 결과 Kodu를 이용한 프로그래밍 교육이 일반적인 ICT 소양교육에 비하여 논리적 사고력 하위 영역 중 상관논리 영역 향상에 긍정적인 영향을 주는 것으로 확인되었다. 상관논리는 변인들간에 관계가 있음을 인식하는 능력과 관련된 것으로 객체간의 상호작용을 이용하여 게임이나 디지털 스토리텔링을 구현하는 Kodu를 활용한 프로그래밍 활동이 학생들의 상관논리 향상에 영향을 준 것으로 판단된다.

5.1.2 집단별 논리적 사고력 변화

실험집단과 비교집단에 실시한 학습 내용이 각

집단의 논리적 사고력에 끼친 영향 정도를 비교하기 위해 사전, 사후 검사 결과에 대한 대응표본 t검정을 실시하여 <표 8>에 제시하였다.

<표 8> 논리적 사고력 사전·사후검사 차이 검증

하위 요소	집단	시기	M	SD	t	유의도
보존	실험	사전	2.61	1.118	-1.667	.110
		사후	2.91	.733		
	비교	사전	2.26	.864	-.214	.833
		사후	2.30	.765		
비례	실험	사전	1.48	1.675	.182	.857
		사후	1.43	1.727		
	비교	사전	1.35	1.335	.000	1.000
		사후	1.35	1.555		
변인 통제	실험	사전	1.17	1.267	1.385	.704
		사후	1.26	1.322		
	비교	사전	1.22	1.313	.000	1.000
		사후	1.22	1.166		
확률	실험	사전	.61	.839	.720	.479
		사후	.48	.790		
	비교	사전	.48	.790	-1.000	.328
		사후	.61	.891		
상관	실험	사전	.17	.388	-1.283	.213
		사후	.35	.573		
	비교	사전	.17	.388	.000	1.000
		사후	.17	.388		
조합	실험	사전	1.00	.905	-5.591	.000**
		사후	1.78	.850		
	비교	사전	.65	.982	-2.865	.009**
		사후	1.09	.949		
합계	실험	사전	7.04	4.095	-2.287	.032*
		사후	8.22	4.210		
	비교	사전	6.13	3.559	-.1464	.157
		사후	6.74	3.828		

*p<.05, **p<.01

실험집단의 조합논리의 평균은 사전검사에 비해 사후검사의 점수가 0.78점이 상승한 1.78점으로 나타났으며, 이는 통계적으로 매우 유의미한 것으로 나타났다. 비교집단 역시 사전검사에 비해 사후 검사의 평균이 0.44점 올랐으며 통계적으로 매우 유의미하게 나타났다. 이는 초등학교 6학년 학생들의 자연적인 성장에 따른 결과라고 판단할 수도 있지만 프로그래밍 교육과 논리적 사고력에 대한 기존 연구의 결과를 분석한 결과 초등학교 6학년 학생들의 논리적 사고력 검사에서 모든 학습 집단의 조합논리 점수가 상승되지 않는 것으로 미루어 각 집단에 실험 처치된 Kodu를 이용한 프로그래밍 학습이나 ICT 소양 교육이 조합논리 영역의 평균 상승에 영향을 미쳤을 것이라고 판단할 수 있었다. 점수의 상승정도와 유의도 측면

을 고려했을 때 Kodu를 이용한 프로그래밍 교육이 일반적인 ICT 소양 교육에 비해 조합논리에 더욱 긍정적인 영향을 준다고 유추할 수 있다.

실험집단의 경우 논리 합계의 평균이 사전검사에 비해 1.18점 오른 8.22점으로 나타났으며 이는 통계적으로 유의미한 것으로 나타났다. 비교집단의 경우에도 논리 합계의 평균이 다소 상승하기는 했으나 통계적으로 유의미하지 않았다.

보존, 비례, 변인통제, 확률, 상관에서 평균의 차이가 있었으나 통계적으로는 유의미하지 않은 것으로 판명되었다.

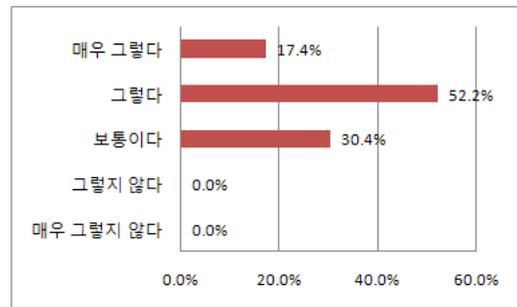
각 학습 집단 별 논리적 사고력 검사 비교 결과 Kodu를 이용한 프로그래밍 교육이 논리 합계에서 논리적 사고력을 유의미하게 향상시킴을 알 수 있었다.

5.2 프로그래밍에 대한 학습 흥미 설문조사

Kodu를 이용한 프로그래밍 학습이 초등학생의 프로그래밍 학습에 대한 흥미에 미치는 영향을 알아보기 위해 Kodu를 학습한 실험집단을 대상으로 설문조사를 실시하였다.

실험집단 23명(남:11명, 여 12명)중 프로그래밍 경험이 있는 어린이들이 2명이고 나머지 21명은 전혀 프로그래밍 경험이 없는 학생들이었다.

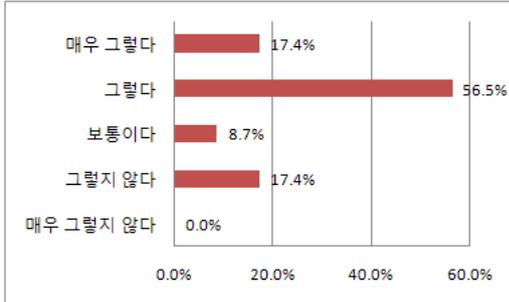
Kodu를 통한 프로그래밍 학습이 재미있었는지를 묻는 질문에 대한 응답을 <그림 5>에 제시하였다. 70%의 학생의 긍정적인 답변을 했는데 게임과 같은 그래픽 환경에서 친숙한 방법으로 로봇에게 명령을 내리는 경험이 학생들의 프로그래밍에 대한 흥미에 좋은 영향을 미친 것으로 보인다.



<그림 5> 프로그래밍이 재미있었나요?

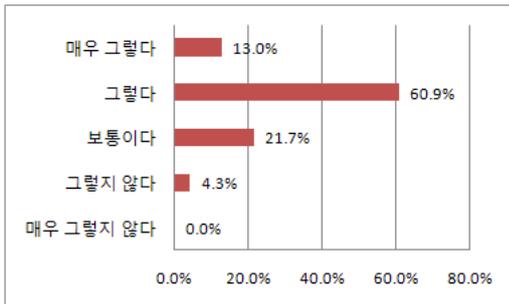
앞으로 Kodu를 이용한 프로그래밍을 계속 하고 싶은가에 대한 질문에 <그림 6>처럼 74%의 학생

이 긍정적인 답변을 했는데 그 이유는 재미있어서 (43%), 프로그래밍을 할 수 있어서(33%), 공부에 도움이 돼서(15%), 쉬워서(10%) 등으로 나타났다. 반면 17%의 학생은 Kodu를 이용한 프로그래밍을 계속 하지 않겠다는 부정적인 답변을 했는데 그 이유는 프로그래밍이 싫어서, 어려워서, 공부와 상관없이 없어서, 영어가 힘들어서 등으로 나타났다.



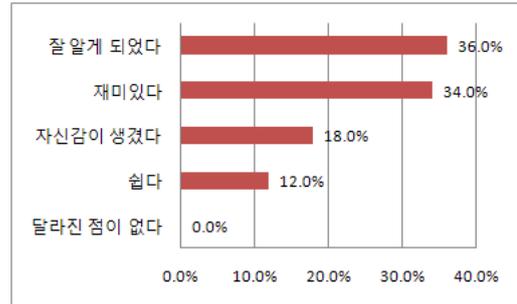
<그림 6> Kodu를 이용한 프로그래밍을 계속 하고 싶은가요?

Kodu를 이용한 프로그래밍 학습이 다른 프로그래밍 언어를 이용한 프로그래밍에 대한 관심에 미치는 영향을 알아보기 위해 다른 프로그래밍 언어를 배우고 싶은지에 대한 질문을 하였고 그 결과를 <그림 7>에 제시하였다. 74%의 학생이 다른 언어도 배우고 싶다고 긍정적으로 응답하였고 4%의 학생이 부정적인 대답을 하였다. Kodu를 이용한 프로그래밍 학습이 다른 프로그래밍 언어 학습에 대한 긍정적인 관심에 도움을 줄 수 있을 것으로 보인다.



<그림 7> 다른 프로그래밍 언어를 배우고 싶은가요?

마지막으로 Kodu를 통한 프로그래밍 학습이 프로그래밍에 대한 학생들의 인식에 어떤 영향을 끼쳤는지 알아보기 위한 설문에 대한 결과를 <그림 8>에 제시하였다.



<그림 8> Kodu를 학습한 후 프로그래밍에 대하여 달라진 생각

프로그래밍이 어떠한 것인지 알게 되었다는 응답이 가장 많았으며 프로그래밍이 재미있다, 프로그래밍에 대한 자신감이 생겼다, 프로그래밍은 쉽다라는 응답이 뒤를 이었다.

설문조사 결과 초등학생들은 Kodu를 이용한 프로그래밍 학습에 재미를 느끼고 그로 인해 프로그래밍에 대한 긍정적인 사고를 갖게 되었다. 또한 프로그래밍 학습 자체에 대한 관심이 높아진 것으로 보아 추후에 이루어질 좀 더 추상화된 프로그래밍 학습에도 긍정적인 태도를 이끌어낼 수 있다고 분석하였다.

6. 결론

본 연구에서는 초등학생들을 대상으로 쉽고 재미있게 프로그래밍을 학습할 수 있는 언어인 Kodu를 사용하여 프로그래밍의 학습을 계획하고 수업을 진행하였다. 그 결과 논리적 사고력의 하위 영역인 상관논리와 논리적 사고력의 전체 합계에서 유의미한 향상이 나타났다. 또한 설문결과를 통한 프로그래밍 학습에 대한 학습 흥미 측면에서도 긍정적인 결과가 나타났다.

추후에는 실험대상의 수를 좀 더 늘려 논리적 사고력 뿐 만 아니라 창의력과 문제해결력 등 프로그래밍과 관련된 다양한 사고력과 관련된 연구를 진행할 계획이다.

아직 많은 연구가 진행되지 않았지만 이번 연구를 계기로 국내에서도 초등학생의 프로그래밍 학습 도구로서 Kodu를 활용하여 초등학생들에게 프로그래밍은 쉽고 재미있는 것임을 알려주고 프로그래밍을 통해 계산적 사고를 신장시킬 수 있는 기회가 많이 제공되었으면 한다.

참 고 문 헌

[1] Resnick, M., Maloney, J., Monroy- Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67.

[2] 교육과학기술부 (2011). **중학교 선택 교과 교육과정** [Electronic version].

[3] 유중현 · 김종혜 (2008). 문제 해결과정에서의 정보과학적 사고 능력에 대한 개념적 고찰. **정보창의교육논문지**, 2(2), 15-24.

[4] 윤일규 · 김종혜 · 이원규 (2010). 정보 교과의 문제해결과정에서 논리적 사고력 구성요소에 대한 조작적 정의. **한국컴퓨터교육학회 논문지**, 13(2), 1-14.

[5] Kodu Game Lab. (n.d.). *Microsoft Research FUSE Labs*. Retrieved October 4, 2012, from fuse.microsoft.com/page/kodu.

[6] MacLaurin, M. (2011). The design of kodu : A tiny visual programming language for children on the xbox 360. *POPL '11 Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. 241-246.

[7] Stolee, K. T., & Fristoe, T. (2011). Expressing computer science concepts through kodu game lab. *SIGCSE '11 Proceedings of the 42nd ACM technical symposium on Computer science education*, 99-104.

[8] 김병수 · 김종훈 (2012). 초등 예비교사의 컴퓨터과학에 대한 인식 변화를 위한 계산적 사고 기반 알고리즘 학습의 설계 및 적용. **수산해양교육연구**, 24(4), 528-542.

[9] 김태훈 · 김종훈 (2012). 초등학생의 프로그래밍 교육을 위한 STEAM 기반의 교과융합 프로그램 개발. **한국컴퓨터교육학회 학술발표 논문집**, 16(2), 73-78.

[10] National STEM Video Game Challenge. (n.d.). *National STEM Video Game Challenge*. Retrieved October 9, 2012, from <http://stemchallenge.org/>.

[11] 노정원 (1998). **과학 교육 연구에 사용된 GALT 원본과 축소본에 대한 조사 연구**. 석사학위 논문, 이화여자대학교.



김 태 훈

2003 제주교육대학교
컴퓨터교육과(교육학학사)
2011~현재 제주대학교
컴퓨터교육전공 박사과정
관심분야: STEAM 교육, 프로그래밍 교육, EPL
E-Mail: gtranu@naver.com



김 종 훈

1998 홍익대학교
전자계산학과(이학박사)
1998~1999 ETRI Post-Doc.
1999~현재 제주대학교 초등컴퓨터교육전공 교수
관심분야: 컴퓨터교육
E-Mail: jkim0858@jejunu.ac.kr