Journal of Korea Game Society 2018 Jun; 18(3): 49-60
http://dx.doi.org/10.7583/JKGS.2018.18.3.49

게임 프로그래밍

앱 인벤터 게임 작성을 활용한 중학생의 프로그래밍 수업 및 학습 성과 분석

김성경, 김상철
한국외국어대학교 교육대학원 컴퓨터교육전공
ksg2231@naver.com, kimsa@hufs.ac.kr

Middle-School Programming Classes Utilizing App Inventor Game Creation and the Analysis of their Educational Outcomes

Sungkyung Kim, Sangchul Kim
Comp. Edu. Major, Education Graduate School, Hankuk Univ. of Foreign Studies

요 약

많은 나라들에서 최근 초중고생에게 컴퓨터 프로그래밍 교육을 의무적으로 실시하고 있다. 일반적으로 컴퓨터 프로그래밍이 초중고생이 배우기 어렵기에, 게임 프로그래밍을 프로그래밍 수업에 적용해 학생들의 수업 동기와 만족도가 높아졌다는 기존 연구들이 다수 발표되었다. 우리는 중학생의 컴퓨터 프로그래밍 수업에 앱 인벤터 게임 작성을 활용하고 그 학습 효과를 분석하는 연구를 진행했다. 본 논문에서는 우리가 수업 시간에 활용했던 앱 인벤터 게임들의 명세 및 우리의 수업 모델을 기술한다. 또한 게임 프로그래밍을 활용한 프로그래밍 수업이 컴퓨팅 사고력, 창의적 사고력 및 프로그래밍에 대한 인식에 미치는 변화를 분석했다. 분석 결과, 중학생들의 이들 능력 및 긍정적 인식이 크게 개선됨을 알 수 있었다. 또한 짝 학습인 하브루타(Havruta) 학습을 프로그래밍 수업에 적용해 이것이 일반 방식에 비해 학습 성과를 높임을 알 수 있었다.

Keywords : 프로그래밍, 앱 인벤터, 게임, 학습 성과, 중학생

ABSTRACT

In many countries, recently computer programming eduction has become mandatory for K-12 students. Generally since programming is difficult for K-12 students to learn, a lot of previous works have been published in which the study motivation and class satisfaction increased when game programming was applied to programming classes. We investigated a research in which we used game programming in the programming classes for middle-school students, and analyzed educational effects. In this paper, we described the specification of the games and our class model that were applied in the research. Also, we analyzed changes made to computation thinking abilities, creative thinking abilities, and perceived support for programming after the programming classes. As a result, it was found that those abilities and the perceived support were increased. Additionally, we applied Havruta learning, a kind of paired learning, in the programming class, and have known that it increased the educational outcome with respect to the traditional method.

Keywords : programming, app inventor, game, educational outcome, middle-school student

Received: Apr, 16, 2018 Revised: Jun. 15. 2018
Accepted: Jun. 20. 2018
Corresponding Author: Sangchul Kim(Hankuk Univ. of Foreign Studies)
E-mail: kimsa@hufs.ac.kr
ISSN: 1598-4540 / eISSN: 2287-8211

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.otg/licenses/by-nc/3.0), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서 론

미래 사회에서는 새로운 지식과 기능의 습득뿐만 이 아니라 문제 발견과 논리적 해결 능력이 중요하다. 소프트웨어 프로그래밍은 학생들의 컴퓨팅 사고력, 논리력 및 창의적 문제해결 능력을 키울 수 증대시키는 효과가 있음은 잘 알려져 있다[1,2]. 영국, 핀란드 등의 많은 국가에서는 이미 초등학교 과정부터 프로그래밍 교육이 의무화되어 있다.

우리나라에서도 2015년 개정 교육과정에서 소프트웨어 교육을 강조하고 있다[3]. 소프트웨어 교육 과정은 2018년부터 중학교를 시작으로 의무화되고, 2019년엔 초등학교 고학년까지 확대될 계획이다. 소프트웨어 교육의 핵심은 프로그래밍 교육으로서, 학생들은 이 교육을 통해 프로그램 언어를 이용해 소프트웨어 설계 및 개발하는 경험을 쌓고 새로운 정보 기술과 도구를 활용해 문제를 해결하는 능력을 키우게 된다.

최근 초중고 학생들을 위해 프로그래밍 수업 을 보다 쉽고 흥미롭게 받을 수 있도록 하는 방안에 대한 관심이 높다. 그 중 한 시도로서 컴퓨터 프로그래밍 수업에 게임 프로그래밍을 도입하는 것이다[4,5,6,7,8]. 기존 연구에 따르면, 게임 프로그램 작성을 활용한 프로그래밍 수업에서 학생들의 학습동기와 수업 만족도가 높아졌다. 하지만 우리가 조사한 바, 중학생의 프로그래밍 수업에서 게임 프로그래밍을 활용할 때 학생들의 문제해결 능력 및 프로그래밍에 대한 인식 변화 등의 종합적 측면에서 학습 성과를 분석한 연구는 거의 발표되지 않았다.

우리는 프로그래밍 경험이 거의 없는 중학생을 대상으로 앱 인벤터 코딩 수업에 게임 프로그램을 활용한 수업안을 설계하고 진행하는 연구를 수행했다. 앱 인벤터는 레고 블록 형태로 표현된 각종 프로그래밍 요소들을 적절히 조립함으로써 스마트용 앱을 쉽게 생성할 수 있는 개발 도구이다. 앱 인벤터는 일반 프로그램 언어보다 높은 추상화 수준의 사용법을 제공해 학습자의 코딩 작업 부담을 크게 줄여 주는 장점이 있기 때문에, 초보자의 프로그래밍 교육에 자주 활용되고 있다[9].

우리의 연구에서는 전체 6주에 걸쳐 일반 코딩 교육과 함께 두더지 잡기 게임과 갤러그 슈터 게임의 프로그래밍을 진행했다. 학생들이 게임 플레이와 규칙을 이해하기 위한 노력 대신에 프로그래밍 자체에 더 시간을 쓸 수 있도록, 조작법과 로직이 상대적으로 간단하고 잘 알려진 게임을 선택했다.

본 논문에서는 우리가 수행했던 앱 인벤터용 게임 프로그래밍을 활용한 프로그래밍 연구에서 설계했던 수업 모형과 학습 성과를 기술한다. 학습 성과는 컴퓨팅 사고력(Computational Thinking Ability), 창의적 사고력(Creative Thinking Ability) 및 컴퓨터 프로그래밍에 대한 긍정적 인식(Perceived Support for Computer Programming) 변화 측면에서 분석했다. 이를 위해 수업 전후에 이들 측면에서의 검사를 각각 실시했다. 이들 검사의 결과를 분석한 결괴, 학생들의 컴퓨팅 사고력과 창의적 사고력이 증가했고 컴퓨터 프로그래밍에 대한 긍정적 인식이 크게 개선됨을 알 수 있었다. 또한 수업 중에 학생들은 게임 개발을 직접 한다는 사실에 큰 흥미와 높은 수업 몰입도를 보였다.

최근 하브루타(Havruta) 방식의 교육에 대한 관심이 높다. 하브루타란 짝을 나타내는 용어로서, 하루루타 방식 수업에서는 학생들이 짝을 이루어 수업 내용을 토론하는 행위가 이루어진다[10]. 하브루타 방식이 여러 측면에서 학습 성과를 높인다는 연구들이 다수 존재한다. 우리의 연구에서는 학생들을 무작위로 두 그룹으로 나누어 한 그룹은 하브루타 방식을, 다른 그룹은 일반 방식을 적용했다. 수업 결과를 분석해 본 결과, 중학생 프로그래밍 수업에서도 하브루타 방식이 일반 방식에 비해 의미 있는 수준의 더 좋은 학습 성과를 보였다. 하브루타식 토론을 중학생의 프로그래밍 수업에 적용한 기존 연구는 거의 발표되지 않았다.

2. 관련 연구

우리의 조사의 의하면 게임 작성을 활용해 대학

에서 컴퓨터 프로그래밍, 소프트웨어 공학 등의 컴퓨터 공학 주제들을 가르치려는 시도는 많이 발표되었다. 반면에 초중고생을 대상으로 게임을 활용한 프로그래밍 교육에 관한 연구는 많지 않다.

초중고생들이 프로그래밍 능력을 습득하는데 가장 어려운 점은 추상화 개념을 이해하는 것이라고 한다[12]. 여기서 추상화 개념은 문제의 해결책 설계, 문제를 작은 문제들로 분해, 에라 상황을 가정한 테스팅 등을 말한다.

최근 초중고 학생들을 대상으로 게임을 활용해 프로그래밍을 교육한 사례 연구들이 발표되고 있다[4,5,6,7,8]. [4,5,6,8]에서는 게임 작성을 프로그래밍 실습으로 도입해 본 결과, 학생들의 프로그래밍에 대한 흥미와 학습 동기가 크게 증가함을 확인했다. [7]에서는 자료구조 및 알고리즘 설계를 배우는데 보드 게임을 학습도구 활용해 학습 성과 향상에 효과가 있음을 관찰했다.

[13]에서는 중학생부터 대학생까지 학생의 사전 배경지식 및 그 수준과 무관하게 게임 설계 및 프로그래밍에 큰 흥미를 보였고, 그 결과 컴퓨터 공학의 전반적 개념들에 큰 관심을 가지게 되었다는 결과를 발표했다. [14]에서는 게임 프로그래밍 수업이 여중생들의 컴퓨터 기술과 프로그래밍에 대한 긍정적 인식을 향상시킨 사례를 보고하고 있다.

앞에서 언급한 대로 기존 관련 연구들은 대부분 게임 프로그래밍이 학생들의 수업 만족도, 학습 동기, 프로그래밍에 대한 인식 변화에 대해 효과가 있음을 확인하는 사례 연구들이다. 반면에 게임 프로그래밍 수업이 이들 효과와 함께 계산적 사고력 등의 학업 성취도에 미치는 효과를 종합적으로 분석한 연구는 많지 않다[15,16,18].

[15]에서는 중학생들이 진행한 게임 프로그래밍 프로젝트의 결과물을 분석해서 이들의 평가 모델을 제안했는데, 이 모델에서는 프로그래밍 요소(programming constructs) 측면, 게임 미캐닉스(game mechanics) 측면 및 게임 코드 패턴(code pattern) 측면에서 수준별 완성도를 정의했다. [16]에서는 학생들의 수업 동기를 여러 측면에서 측정하는 일반 모델로 ARCS(Attention, Relevance, Confidence, Satisfaction)을 제안했고 [8]은 이를 게임 제작 수업에 적용해 본 연구 결과를 발표했다. 하지만 프로그래밍 수업의 특성을 고려한 맞춤형 수업 동기 모델은 아직 발표되지 않았다.

하브루타(Havruta) 방식의 학습에서는 두 사람이 짝을 이루어 서로 질문하고 토론하면서 이해 못하는 부분을 같이 점검하고 서로의 이해를 돕게 된다[10]. 일반 방식과 비교해서 하브루타 방식 수업이 가진 특성 및 장점을 여러 측면에서 비교 분석하는 연구들이 최근 발표되고 있다. 여기서 일반 방식이라 학생 혼자서 강사의 도움을 받으며 수업 내용을 이해하고 부여된 실습 등의 과제를 진행하는 것을 말한다. [18]에서는 영화관련 수업을 수강생을 대상으로 하브루타식 토론이 개인 및 집단 창의성 증가에 일반 방식에 비해 더 크게 기여했음을 주장했다. 하브루타 수업은 수업 교재와 대화하는 느낌을 증가시켜 학생들의 수업 이해에 긍정적인 효과를 미친다는 주장도 있다[10]. [20]에서는 대학생의 자료구조 및 프로그래밍 수업에서 초보 수준의 학생 대부분은 하브루타 방식을 일반 방식에 비해 선호함을 확인했다.

우리의 조사에 의하면, 하브루타를 적용한 중고생을 대상의 프로그래밍 교육에서 적용하고 그 성과를 분석한 기존 연구는 거의 발견하기 어렵다.

3. 게임 프로그래밍을 활용한 중학생 프로그래밍 수업의 설계

3.1 수업 범위 및 절차

교육부 2015년 개정 교육과정에서는 초중고의 정보 과학과 정보 윤리 및 정보기기 영역을 축소하고 컴퓨터를 이용한 문제해결 방법과 절차 영역을 확대하였다. 확대된 영역의 핵심은 추상화, 알고리즘 및 프로그래밍 능력을 신장하는데 있다. 중학교 교과과정에서 프로그래밍 요소 학습의 핵심

개념들로 입출력, 변수와 연산, 제어구조(반복, 선택, 순차), 응용을 명시하고 있다[3]. 따라서 프로그래밍 교육에서 사용하는 게임 프로그래밍 교육에서도 이들 개념들을 포함해야 할 것이다.

우리는 2015년 개정 정보교과 교육과정을 기반으로, 중학생 프로그래밍 수업에서 게임 프로그래밍을 활용할 시, 다루어야 할 앱 인벤터 개념들을 파악했다. [Table 1]은 학습 요소별로 해당 앱 인벤터 개념들을 정리한 것이다.

[Table 1] Learning Elements and App Inventor Concepts for Middle School Students

Learning Elements	App Inventor Concepts
algorithm	• main game loop using the timer event handler of the clock block • event-response programming model
abstraction	• hierarchical structure of blocks
variable	• variable block • property variables of game entities (i.e, balls, sprites)
operator	• operator block
control structure	• if block, if-else block, while block • close-application block
function	• function call block • setter block, getter block

알고리즘은 문제해결을 위한 명령어 또는 규칙들의 연속(sequence)로 정의할 수 있다. 게임 프로그램의 가장 높은 단계인 주 게임 루프(main game loop)는 앱 인벤터의 클락(clock) 블록이 지원하는 타이머 이벤트 핸들러로 구현 가능하다. 즉, 타임아웃 이벤트를 반복적으로 발생시키고 매 이벤트 발생 시에 화면 업데이트 등의 게임 로직을 지속적으로 수행하는 것이다. 프로그램 초보자는 프로그램의 개념적 실행 모델을 이해하는데 큰 어려움을 겪는다고 한다. 앱 인벤터 코드는 개체들과 연결된 이벤트 핸들러의 집합으로 볼 수 있다. 일반적으로 게임 프로그램이 이벤트-반응 모델을 기반으로 동작되는데, 앱 인벤터 코드의 작성 및 실행을 이해하게 되면 게임 프로그램의 실행 모델을 쉽게 이해할 수 있을 것이다.

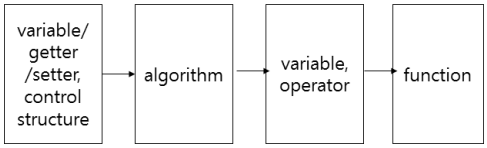
프로그래밍에서 추상화는 코드 블록의 상세한 것을 숨기고 보다 높은 수준에서 다루는 것을 말한다. 우리는 앱 인벤터에서는 추상화 개념을 함수 선언 및 호출, 코딩 블록들의 계층구조를 통해 학습이 가능하다고 본다. 중학교 정보교과에서 함수는 포함되지 않지만, 효과음 출력과 개체 위치의 변경 등과 같이 게임 프로그램에 필수적인 기능을 위한 적절한 내장 함수의 호출은 필요하다. 앞에서 언급했듯이 앱 인벤터에서는 각종 프로그래밍 요소(변수, 연산자 등)와 개체들이 블록 형태로 사용자에게 제공된다. 사용자는 블록들을 선택해 더 큰 블록을 조립하게 되는데, 이와 같은 블록의 계층구조를 만들고 이해하면서 추상화의 개념을 익힐 수 있을 것이다.

주요 프로그래밍 요소인 변수와 연산자의 학습은 해당되는 앱 인벤터 블록들로 다룸으로써 가능하다. 앱 인벤터에서는 복잡한 제어 블록들을 제공하지만, 우리는 중첩된 if문, case 문등과 같은 복잡한 제어구조는 배제했다. 그 이유는 사실 단순 조건문과 반복문만으로도 중학생 수준의 게임 프로그램에서 필요한 대부분의 제어 구조를 표현할 수 있고, 경험상 복잡한 제어구조는 대학생도 이해하기 어려워하기 때문이다. 함수에 대해서는, 기본적으로 함수 선언은 고려하지 않고, 함수 호출 블록과 변수의 값을 읽어 오는 게터(getter) 및 값을 저장하는 세터(setter) 블록을 위주로 학습했다.

2015년 개정 정보교과 교육과정에 따르면 중학교에서는 약 40시간을 정보화 교육에 할당하게 된다. 이런 시간 제약과 중학생의 컴퓨터 프로그래밍 이해력을 감안해, 앱 인벤터 게임 개발을 활용하는 중학교 프로그래밍 수업에서는 앱 인벤터의 기능들 중 [Table 1]에서 언급한 프로그래밍을 위한 기본적 요소 블록들과 게임 작성에 필요한 기본적인

개체 블록들을(예를 들면, 스프라잇, 타이머, 사운드 등)을 수업 대상으로 한정했다.

프로그래밍 요소별 학습 순서는 교사의 판단에 따라 정할 수 있다. 우리의 연구에서는 먼저 단순 메시지를 출력하는 앱과 같은 간단한 앱 제작 실습을 통해 앱 인벤터의 사용법을 학습한 후, [Table 1]에서 명시한 프로그래밍 요소들을 그림 1과 같은 순서로 학생들이 학습하도록 했다. 본 그림의 학습의 순서는 우리의 경험에 따른 것으로서, 기본적으로 학생들이 앱 인벤터로 프로그램을 제작할 때 프로그램 복잡도의 증가에 따라 이해가 필요한 프로그래밍 학습 요소 순을 반영한 것이다.



[Fig. 1] A Sequence of the Learning Elements to be Taught

3.2 우리의 프로그래밍 수업용 게임의 명세 및 특징

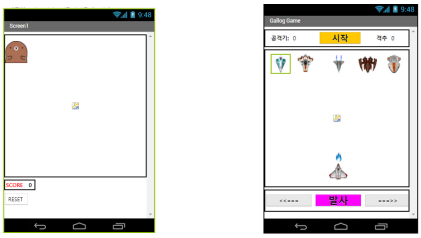
기존연구 [21]에서는 교육용 게임이 가져야 하는 주된 특징으로서, 게임 규칙이 단순해 게임 동작원리 파악이 쉬워야하고, 전체 수행시간이 짧아야하고, 숨은 정보(예를 들면, 카드 게임에서 손에 숨겨진 카드)를 피해야하는 것을 들었다. 우리는 중학교 정보교과에서의 프로그래밍 수업용으로 2개의 게임을 선택했다. 하나는 두더지 잡기(Whack-A-Mole)이고 다른 하나는 미사일을 발사하는 갤러그 슈터(Galaga Shooter)이다. 이들은 잘 알려진 단순 캐주얼 게임으로서, 앞에서 언급한 교육용 게임의 특징을 가지고 있다.

[Table 2]는 이들 게임의 목표(goal), 게임플레이(game play) 및 수행할 학습 내용 (things to be accomplished)들 기술한다. 그림 2는 이들 게임의 스크린 샷을 보여주고 있다. 두더지 게임 프로그램 작성을 통해, 프로그램에서의 알고리즘의 역할 및 설계 방법, 터치 이벤트 처리를 통한 이벤트-반응 방식의 프로그램 실행 모델, 스프라잇 개체를 화면에 재현하는 방법을 익힌다. 두 번째 갤러그 슈터 게임을 통해, 앞선 게임에서 습득한 지식을 강화하고 애니메이션 및 함수에 대한 개념을 학습하게 된다. P4나 P9에서 명시된 대로, 학습 성과 기준에 수업 초반에 학생들에게 주어진 게임 명세에 포함되지 않은 기능을 추가하는 것을 학습 내용에 포함해, 학생들의 창의력과 학습 동기를 높일 수 있도록 했다.

[Table 2] The Specification of the Two Games for Our Research and Things to Be Accomplished in Classes

Title	Game Specifications
Whack-a-Mole	Goal: To whack a mole and add to the player's score. Game play: A player tries to click on a mole continuously moving around until a timeout expires. Things to be accomplished: P1) To devise an algorithm to support the given game play, and a code structure to implement the algorithm P2) To understand the role of touch event, and implement event handlers in the context of the game program P3) To distinguish between the concepts of image sprites and image files, and use them properly. P4) To add a new game play scenario (for example, penalties due to misclicks)
Galaga Shooter	Goal: To shoot down space enemies falling down and add to the player's score. Game play: A player moves a missile gun left and right by clicking buttons, and shoot missiles to destroy them

	until no lives remain. Things to be accomplished: P5) the same as P1 P6) the same as P2 P7) To understand the velocity and direction of an animated game object, and change them in the context of a game program. P8) To understand how functions work, and declare a simple function by completing partially construction code. P9) the same as P4



(a) Whack-a-Mole (b) Galaga Shooter

[Fig. 2] Screen Shots for the two Games

앞에서 언급한 두 개의 게임이 프로그래밍 수업 교재로서 가진 특징을 정리하면 [Table 3]과 같다. 게임의 주요 구성 요소들 중에 도전(challenge)과 판타지(fantasy)가 있는데, 우리의 게임은 전자 요소는 약하고 후자 요소가 강한 게임으로 분류될 수 있다. 이것은 [11]의 결과를 참고한 것으로서, 해당 연구에 따르면 도전 요소가 강한 교육용 게임과 판타지 요소가 강한 교육용 게임을 비교할 때 전자가 후자에 비해 교육 효과가 뛰어나다고 한다. 또한 우리의 수업에서 작성한 게임은 전략성보다는 액션성이 높은 게임이다. 대체적으로 전략 게임이 액션게임보다 게임 규칙이 복잡해, 프로그램 초보자들이 전략 게임의 규칙과 절차를 이해하는데 시간을 지나치게 소모해 프로그래밍 자체의 시간이 줄어들 게 될 것을 막기 위함이다.

기존 연구[15]에 따르면, 프로그래밍 학습 교재로 제작되는 게임은 프로그래밍 요소, 게임 미캐닉스(mechanics), 프로그래밍 패턴(pattern) 측면에서 적절히 설계되어야 한다. 우리 게임들을 이들 측면애서 분석해 보면 다음과 같다. 프로그래밍 요소 측면에서는, [Table 1]과 같이 교육부 정보교과 개정안에서 요구하는 것들을 충실히 반영할 수 있도록 게임을 설계했다. 게임 미캐닉스 측면에서는, 프로그래밍 교육용 게임에 사용하는데 적합한 요소를 각 게임당 2개씩을 가지고 있다. 시간 제한적 도전(timed challenge)란 타임아웃 전에 도전 과제를 마무리하기를, 때리기(hitting)이란 물체를 터치하거나 타격하기를, 슈팅(shooting)은 무엇을 발사하기를, 회피(avoidance)는 충돌을 방지하기 위해 노력하기를 말한다. 기존 연구에서는 프로그래밍 교육용 게임을 위한 10개의 프로그래밍 패턴을 제시하고 있다. 우리의 게임들은 이들 중 약 5개를 각각 활용해 구현하게 된다. 일부 조작(manipulating subpart)이란 객체의 일부 (예를 들면, 공 객체의 위치, 속도 정보 등)를 조작하는 것을 말한다.

[Table 3] Main Features of the Two Games writ Educational Tools

	Analysis Factors		Description
WM	Style	Challenge vs Fantasy	challenge
		Genre	action
	Game Mechanics		timed challenge, hitting
	Game Programming Patterns		sound, movement, manipulating subpart, counter, timer
GS	Style	Challenge vs Fantasy	challenge
		Genre	action
	Game Mechanics		shooting, avoidance
	Game Programming		sound, movement,

	Patterns	manipulating subpart, counter, collision

*WM: Whack-a-Mole, GS: Galaga Shooter

3.3 게임 프로그래밍 수업 방식

우리는 전체 프로그래밍 수업을 전체 6주로 구성하면서 크게 두개 부분으로 나누었다. 전반부는 4주간에 걸쳐 두더지 잡기 게임을, 후반부는 2주간에 걸쳐 갤러그 슈터 게임을 위주로 진행했다. 각 부분에서는 다음 5 단계로 수업이 진행되었다.

Step 1에서는 제작할 게임을 포함한 학습 내용 전반에 대해 그림 1의 내용을 중심으로 학생들에 설명했다. Step 2에서는 학생들은 주어진 앱 인벤터 실습 문제를 해결함으로써 프로그래밍 능력을 키웠다. 수업 시간의 제약으로 인해, 표 1의 프로그래밍 요소를 위한 기능과 게임 작성에 필요한 최소한의 앱 인벤터 블록들을 학습했다. 이들 블록에는 사용자 인터페이스 개체, 그리기 개체, 애니메이션 및 미디어 개체, 센서 개체 등이 포함되었다.

Step 3과 4에서는 학생들은 게임의 기능들을 어떻게 앱 인벤터로 구현할 지를 먼저 고안한 후 주어진 게임 기획에 따라 앱 인벤터 게임을 작성했다. 시간의 한계와 학생들의 능력을 감안해, 전체 게임 코드의 약 절반은 이미 작성된 상태로 학생들에게 주어졌다. 학생들은 미완성 부분을 코딩하도록 하였다. 수업 시간 내 마치지 못한 부분은 숙제로 처리했다. Step 5는 학생의 최종 코드에 대해 평가를 피드백 해주었다.

Step 1은 첫 주에, Step 3 및 4는 마지막 주에 진행되었다. Step 2는 두더지 잡기 게임의 경우에는 첫째 주에서 세 번째 주에 걸려 진행되었고, 갤러그 슈터 게임의 경우에는 첫 번째 주에 진행되었다.

우리는 학생들을 무작위로 두 그룹으로 나누어, 한 그룹에서는 한 그룹에서는 2명씩 한 조를 이루어 짝과 하브루타식 토론을 하면서 수업 내용 이해와 실습을 진행하도록 했고, 나머지 그룹은 일반 방식으로 수업을 진행했다. 학생들간의 하브루타식 토론의 원활한 진행을 위해, 토론할 내용을 미리 정리한 학습지를 학생들에게 나누어주었다. 본 학습지의 자세한 내용은 [17]에 나와 있다.

4. 실 험

4.1 실험 방법

우리는 앞 장에서 기술한 수업 모델을 경기도 Y 지역의 중학생 38명을 대상으로 적용했는데, 이들을 소그룹 6개로 나누어 각 그룹에 대해 별도의 수업을 실시했다. 우리가 조사해 본 바, 현재 교육 현장에서는 프로그래밍 교육이 소그룹으로 진행되고 있다. 또한 소그룹 3개에는 하브루타 학습을 적용하고, 다른 소그룹 3개에는 일반 수업 방식을 적용했다.

먼저 수업 전에 사전 검사지(Pretest)를 이용해 학생들의 컴퓨팅 사고력, 창의적 문제해결력 및 컴퓨터 프로그래밍에 대한 인식을 검사했다. 수업을 진행 후에는 사후 검사지(Posttest)를 이용해 이들 능력과 프로그래밍에 대한 긍정적 인식의 변화를 분석했다.

컴퓨팅 사고력 측정을 위해 한국교육학술정보원이 개발하여 보급해 준 컴퓨팅 사고력 검사지[23]을 활용하였다. 검사지의 검사 항목들은 자료에 대한 이해(CT-1로 표기), 문제에 대한 이해(CT-2로 표기) 및 문제해결을 위한 프로그래밍(CT-3으로 표기)의 3가지 영역으로 구성된다. 사전 검사지의 문항수는 총 15개로서, 영역별로 각각 4개, 5개 및 5개로 구성된다. 사전 검사지는 총 21개 문항으로 구성되고, 영역별로 각각 6개, 9개 및 6개씩 나뉜다. 중학생들이 전문적 프로그래밍 교육을 처음 받는 입장을 감안해, 사전 검사지는 사후 검사지에 비해 알고리즘 관련 문항의 비율을 줄이고 비교적 일상과 연관된 문항을 중심으로 구성했다. 그러나 사전-사후 검사지간의 영역별 문항수 비율은 거의 동일하다.

창의적 문제해결력 검사는 한국교원개발원의 연구를 기반으로 수정 개발한 검사지[19]로 실시했다. 해당 검사는 많은 프로그래밍 연구에서 활용되고

있으며 그 신뢰도가 검증되었다고 볼 수 있다. 본 검사지는 4가지 영역으로 이루어져 있다. 4 가지 영역은 특정 영역 기능•기술의 이해 및 숙달 여부(CRT-1로 표기), 확산적 사고(CRT-2로 표기), 비판적• 논리적 사고(CRT-3으로 표기), 동기적 요소(CRT-4로 표기)로 구성된다. 이들 영역들의 문항수는 각각 5개, 5개, 5개 및 4개이다. 각 문항의 답은 5점 리커트(Likert) 척도인 1=매우 그렇지 않다, 2=그렇지 않다, 3=보통이다, 4=그렇다, 5=매우 그렇다로 평가된다.

우리는 프로그래밍에 대한 긍정적 인식 파악을 위한 검사지를 5문항으로 설계했다. 이들은 프로그래밍에 대한 이해 정도(PS-1로 표기), 프로그래밍 능력의 중요성(PS-2로 표기), 프로그래밍 직업에 대한 선호도(PS-3로 표기)라는 3개 영역으로 나누어진다. 영역별 문항수는 각각 1개, 2개 및 2개이다. 각 문항의 답은 5점 평정척도로 평가되는데, 3점이 보통 수준이면서 수치가 커질수록 긍정적 인식을 나타낸다.

4.2 실험 결과 및 분석

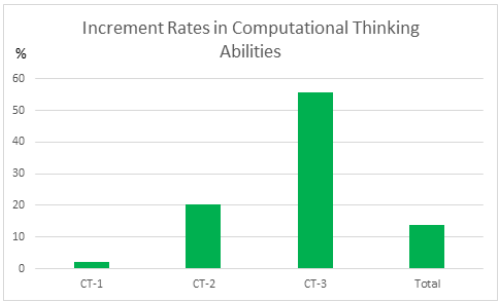
앞에서 언급했듯이 프로그래밍 수업은 여러 회차에 걸쳐 진행되었다. 우리가 관측한 바에 따르면, 학생들은 일반적 프로그래밍 요소를 학습하는 회차에 비해 게임 코드를 수정 및 기능 추가를 진행하는 회차에서 수업 내용에 보다 흥미를 느끼고 집중하는 모습을 보였다

학생들의 컴퓨팅 사고력의 변화는 [Table 4]와 [Fig. 3]에 정리되어 있다. 본 장의 표에서 'Pre'는 사전 검사의 점수를, 'Post'는 사후 검사의 점수를 나타낸다. 사전 검사지와 사후 검사지의 문항 수가 서로 다르기에, 비교 분석을 위해 각 영역의 총점을 100점 만점으로 환산해 사용했다. 표 4에서 보다시피, 모든 영역에서 증가를 보였고 총 증가치는 47점 정도이다. 자료에 대한 이해 영역(즉, CT-1)의 증가율을 크지 않지만, 나머지 영역들에서는 20% 이상의 증가율을 보였다. 전체 수업기간이 6주라는 점을 감안하면, 자료에 대한 이해 능력은 타 능력에 비해 상대적으로 짧은 기간 동안에 향상되지 않음을 알 수 있다. 특히 문제해결을 위한 프로그래밍 영역(즉, CT-3)의 증가율은 56% 정도로, 앱 인벤터 프로그램 작성 수업이 학생들의 코딩 능력을 크게 향상 시켜줌을 알 수 있다.

우리는 각 영역별로 사전 검사 점수를 독립변수로, 사후 검사 점수를 종속변수로 정해 양측 t-검정을 실시했다. 유의 수준 5%를 기준으로 각 영역의 유의 확률 p를 살펴보면, 첫 번째 영역에서의 사고력의 변화는 현저하지 않다. 그러나 나머지 영역들에서는 사후 컴퓨팅 사고력의 증가가 사전에 비해 확률적으로 유의미하게 크다는 것을 알 수 있다.

[Table 4] Changes in Computational Thinking Abilities

Category	Test Time	Mean	Std. Dev.	p
CT-1	Pre	81.57	17.12	0.56
	Post	79.82	12.34	
CT-2	Pre	57	16.27	0.0048
	Post	64.91	14.95	
CT-3	Pre	64.73	32.36	0.00542
	Post	78.28	17.36	
Total	Pre	203.33	47.26	0.00223
	Post	222.02	30.40	



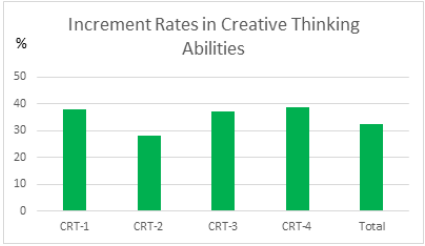
[Fig. 3] Increment Rates in Computational Thinking Abilities

[Table 5]와 [Fig. 4]는 학생들의 창의적 사고력의 변화를 보여준다. 모든 영역들에서 사후 검사의

점수가 사전 검사에 비해 25% ~ 38% 만큼 증가됨을 보였다. 우리는 각 영역별로 사전-사후 검사 점수간의 양측 t-검증을 실시했고, 그 결과는 표 5의 p 열에 나타나있다. 유의 확률 p가 0.05보다 훨씬 적음을 감안하면, 이들 증가율이 유의미하게 크다는 것을 나타내고 있다. 컴퓨팅 사고력의 경우, 문제해결을 위한 프로그래밍 영역에서만 30%가 넘는 큰 증가를 보인 반면에 창의적 사고력의 경우에는 모든 영역에서 골고루 30% 정도 또는 그 이상의 증가를 보이고 있다. 따라서 앱 인벤터 프로그래밍 수업이 중학생의 다양한 영역에서의 창의적 사고를 키우는 좋은 수단중 하나임을 보여주고 있다.

[Table 5] Changes in Creative Thinking Abilities

Category	Mean		Std. Dev.	p
CRT-1	Pre	13.47	3.27	1.5×10^{-7}
	Post	17.63	3.85	
CRT-2	Pre	13.73	3.48	0.0002
	Post	16.65	3.45	
CRT-3	Pre	14	3.09	0.0007
	Post	18.71	7.73	
CRT-4	Pre	13.26	2.89	3.9×10^{-9}
	Post	17.84	3.81	
Total	Pre	54.47	8.99	1.6×10^{-8}
	Post	70.84	13.74	

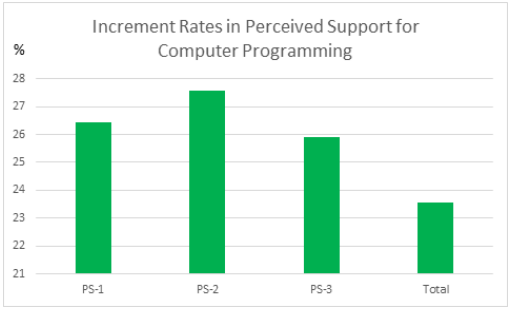


[Fig. 4] Increment Rates in Creative Thinking Abilities

[Table 6]과 [Fig. 5]는 컴퓨터 프로그래밍에 대한 인식 변화를 보여주고 있다. 각 영역에서 사전 검사 대비 사후 검사의 점수가 26% 이상 증가함을 보였다. 각 영역별 사전-사후 검사 점수간의 양측 t-검정 결과인 유의확률 p를 살펴보면, 사전 점수와 사후 점수 간 차이가 확률적으로 유의미하게 크다는 것을 알 수 있다. 프로그래밍 수업은 중학생들이 컴퓨터 프로그래밍 지식 및 중요성을 깨닫는 좋은 수단이 됨을 보여준다. 평균 증가율은 23.5%로 각 영역의 개별 증가율보다 낮은 나타나 있다. 이것은 앞에서 언급한대로 프로그래밍에 대한 이해 정도 영역(즉, PS-1)의 만점은 다른 영역 만점의 절반이기 때문이다.

[Table 6] Changes in the Perceived Support for Computer Programming

	Mean		Std. Dev.	p
PS-1	Pre	2.86	0.93	0.006
	Post	328	0.8	
PS-2	Pre	5.92	1.58	2.4×10^{-5}
	Post	7.13	1.31	
PS-3	Pre	5.71	1.92	3.3×10^{-5}
	Post	6.7	1.45	
Total	Pre	14.5	3.91	6×10^{-7}
	Post	17.1	2.82	



[Fig. 5] Increment Rates in Perceived Support for Computer Programming

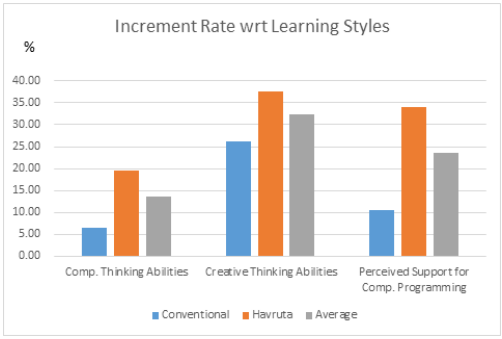
[Fig. 5]는 하브루타 방식과 일반 방식의 수업에 대해 학생들의 사전 및 사후 검사 점수들을 서로

비교한 결과이다. 그림에서 'Conventional'은 학생 혼자 진행하는 일반 방식을 나타낸다. 그림 5에서 보다시피 두 방식 모두에서 컴퓨팅 사고력, 창의적 사고력 및 프로그래밍에 대한 인식 측면에서 사후 검사의 점수가 사전 검사에 비해 증가했음을 알 수 있다. 또한 하브루타 방식에서의 증가율이 일반 방식에 비해 크다는 점을 통해, 중학생의 프로그래밍 수업에서 하브루타 방식이 일반 방식에 비해 학습 성과가 높다는 것을 알 수 있다.

프로그래밍에 대한 긍정적 인식 검사에서 일반 방식과 하브루타 방식간의 증가율 차이는 23% 정도로서, 다른 두 검사보다 증가율 차이가 컸다. 하브루타 학습이 일반 방식의 학습에 비해 학생들의 수업 몰입도가 높아졌고, 그 과정에서 프로그래밍에 대한 이해와 필요성을 보다 쉽게 인식하게 된 결과로 보인다.

우리는 하브루타 및 일반 수업 방식간의 사후 검사 점수 차이가 유의미하게 현저한 지 여부를 위해 2 가지 테스트를 실시했다. 하나는 하브루타 수업 방식의 점수를 종속변수로, 일반 수업 방식의 점수를 독립변수로 해 실시한 t검정이다. 그 결과는 [Table 7]의 "t-test" 열의 유의 확률 p에 나타나 있다. 표에서 보다시피 컴퓨팅 사고력과 프로그래밍에 대한 인식 측면에서는 점수 차이가 크지만, 창의적 사고력 축면에서는 그 차이가 확률적으로 유의미한 수준은 아니었다.

또 다른 테스트는 사전 검사 점수를 공변량(Covariate)으로, 사후 검사 점수를 종속 변수로 정해 실시한 공변량분석(ANCOVA)이다. 이 검사의 이유는 두 방식의 수업에 참여한 학생 그룹간의 동질성 여부를 감안해 두 수업 방식간의 사후 점수의 차이가 유의미한 지를 판단해 볼 필요가 있기 때문이다. [Table 7]의 "ANCOVA" 열에서 보다시피, 세 분야 모두에서 p 값이 유의수준 0.05에 비해 적기에 하브루타 방식과 일반 방식간의 사후 점수 차이는 유의미하게 크다고 할 수 있다. 이들 두 테스트에 앞서, K-S(Kolmogorov - Smirnov) 테스트를 실시해 두 수업 방식의 사후 검사 점수 및 사후 검사 점수가 각각 정규성(normality)을 가짐을 확인했다.



[Fig. 6] Increment Rates w.r.t. Learning Styles

[Table 7] Results of t-test and ANCOVA

	t-test	ANCOVA	
	p	F	p
Comp. Thinking Abilities	0.05	5.44	0.026
Creative Thinking Abilities	0.22	5.6	0.023
Perceived Support for Programming	0.01	18.52	0.003

6. 결 론

최근 영국, 핀란드 등 많은 나라들에서 초중고생에게 컴퓨터 프로그래밍 교육을 의무적으로 실시하고 있다. 우리나라도 2015년 개정 교육과정을 통해 초중고교 정보교과목에서 프로그래밍의 비중을 늘리고 프로그래밍 교육을 의무화했다. 하지만 프로그래밍 수업이 초중고생들에게는 어렵기에, 최근 국내외적으로 이를 효과적으로 가르치는 방법에 대한 관심이 높다.

우리는 중학생을 대상으로 앱 인벤터 기반의 컴퓨터 프로그래밍 수업에서 일반 수업 내용에 게임 프로그래밍을 혼합하는 연구를 수행했다. 본 논문

에서는 우리의 연구에서 학생들이 수업에서 작성했던 게임들의 특징과 게임 프로그래밍을 어떻게 수업에 활용했는지를 기술했다. 프로그래밍 수업 후, 학생들의 컴퓨팅 사고력, 창의적 사고력 및 프로그래밍에 대한 인식에 미치는 변화를 검사했다. 검사 결과에서, 중학생들의 이들 문제 해결 능력 및 프로그래밍에 대한 긍정적 인식이 수업전과 비교해서 유의미한 수준으로 크게 개선됨을 보였다.

하브루타 수업은 학생들이 짝을 이루어 토론하면서 상대방과 협력적으로 학습 내용을 이해해 나가고 또한 상대방으로부터 자극받아 학습 의욕을 높이는 학습 방식이다. 하브루타식 토론을 우리의 수업에 적용한 결과, 게임 프로그래미 작성을 활용한 중학생 프로그래밍 수업에도 일반 방식에 비해 학습 효과가 높다는 것을 알 수 있었다.

ACKNOWLEDGMENTS

This work was supported by Hankuk University of Foreign Studies Research Fund of 2017.

REFERENCES

[1] Emrah Orhun, “Learning problem solving through computer programming”, Advanced Educational Technologies for Mathematics and Science, 1993, pp. 339-362.

[2] Deller James Ferreira, "Fostering the creative development of computer science students in programming and interaction design", Proc. of Computer Science, Vol. 18, 2013, pp. 1446-1455.

[3] General curriculum and subject education courses of elementary and secondary school (in Korean), Ministry of Education Notice, No. 2015-74, 2015.

[4] Savin-Baden, M. Facilitating problem-based learning: McGraw-Hill Education, 2003.

[5] Savin-Baden, M., & Major, C. H. , Foundations of problem-based learning: McGraw-Hill Education, 2004.

[6] E. R. Hayes and I. A. Games. "Making computer games and design thinking: A review of current software and strategies", Games and Culture, 3, 3-4, 2008. pp. 309-332.

[7] L. Bezakova, J. E. Heliotis, S. P. Strout, “Board game strategies in introductory computer science”, Proc. of the 44th ACM technical symposium on Computer science education, 2013, pp. 17-22.

[8] H. Tsukamoto, et. al, "Comparative Analysis of 2D Games and Artwork as the Motivation to Learn Programming", 39th ASEE/IEEE Frontiers in Education Conference, W2J-1 - W2J-6.

[9] F. Turbak, et. al., "Events-first programming in APP inventor", Journal of Computing Sciences in Colleges archive, Volume 29 Issue 6, 2014, pp. 81-89.

[10] O. Kent, A Theory of Havruta Learning, https://www.brandeis.edu/mandel/pdfs/TheoryofHavrutaLearning.pdf

[11] K. Amr, Learning through Games: Essential Features of an Educational Game, Partial fulfillment of Ph.D Dissertation, Graduate School of Syracuse University, 2010.

[12] M. Esteves, et. al., "Contextualization of programming learning: A virtual environment study", Proc. of 38th Annual Frontiers in Education Conference, 2008, F2A-17 - F2A-22.

[13] A. R. Basawapatna, K. H. Koh, and A. Repenning. “Using scalable game design to teach computer science from middle school to graduate school”, Proc. of the Annual Conf. on Innovation and Technology in Computer Science Education. 2010, pp. 224 - 228.

[14] J. Denner. 2007. “The Girls Creating Games program: An innovative approach to integrating technology into middle school”, Meridian: A Middle School Computer Technologies Journal, Vol. 1, No. 10, 2007.

[15] Linda Werner, et. al., “Children Programming Games: A Strategy for Measuring Computational Learning“, Journal ACM Transactions on Computing Education, Vol.

14, No. 4, 2015, pp. 24:1 - 22.
[16] J. M. Keller, K. Suzuki, "Use of the ARCS motivation model in courseware design", In Jonnasen, D. H. (Ed.), Instructional designs for microcomputer courseware, Lawrence Erlbaum Associates, 1987, pp.401-434.
[17] S. Kim, Design and application of a programming class using the Havruta teaching method in middle school computer subject, Master Thesis, HUFS Education Graduate School, 2018.
[18] Y. Hur, "The Effect of Havruta Teaching Methods on University Students' Individual and Group Creativity", Korean Journal of General Education. Vol. 10, No. 3, 2016, pp.73-106.
[19] S. Chou, et. al, Research on Simple and Creative Problem-solving Ability Development, final report, KEDI, 2001.
[20] Y. Kwon, "Effects of Group Learning in Programming Class", Proc. of WORLDCOMP, 2013.
[21] P. Drake, K. Sung, "Teaching introductory programming with popular board games", Proc. of ACM technical symposium on Computer science education, 2011. pp. 619-624.
[23] KERIS, http://edpolicy.kedi.re.kr/FileRoot/EpSchoolDB/Files/EpSchoolDB0000122071F.pdf

김 성 경 (Kim, Sungkyung)

2013.12 덕성여자대학교 컴퓨터공학전공 학사
2017.12 한국외국어대 교육대학원 컴퓨터교육전공 석사
2015.9- 컴퓨터 프로그래밍교육 강사

관심분야: 컴퓨터교육, 프로그래밍교육, 기능성게임

김 상 철 (Kim, Sangchul)

1994.5 미시간주립대학교 컴퓨터공학과 박사
1983.3-1994.8 ETRI 연구원
1994.9- 한국외국대학교 컴퓨터공학과 교수

관심분야 : 기능성게임, 게임 AI, 멀티미디어시스템