

중학교 정보교과에서 짝 프로그래밍이 4Cs 향상에 미치는 영향

박윤미* · 이효종**

전북대학교 교육대학원 컴퓨터교육학과* · 공과대학 컴퓨터공학부, 영상정보신기술연구소**

요약

협동학습의 한 모형인 짝 프로그래밍을 중학교 정보 교과 수업에 적용하였다. 짝 프로그래밍을 통하여 알고리즘 및 프로그래밍 교육을 2주간에 걸쳐서 교육시키고, 전미교육협회에서 제시한 4Cs에 준거하여 실험 분석하였다. 실험결과 통제집단은 사전-사후1검사에서 비판적사고가 통계적으로 유의하게 증가하였고, 실험집단은 비판적사고, 의사소통, 협업, 창의력 모든 측정에서 통계적으로 유의하게 향상되었다. 사전-사후2검사에서 통제집단은 4Cs 평균점수 차이가 전체적으로 0.2점대로 근소하게 나타나 별다른 변화가 없었으며, 실험집단은 비판적사고, 의사소통, 협업, 창의력, 모든 측정에서 통계적으로 유의하게 향상하였다. 사후1-사후2검사에서는 두 집단 모두 4Cs에 유의미한 차이가 없었다. 모든 사후검사에서 실험집단은 통제집단보다 4Cs 모든 측정에서 높은 점수를 보였다. 이에 따라 짝 프로그래밍의 적용은 4Cs에 긍정적인 영향을 미치는 것으로 나타났다.

키워드 : 짝 프로그래밍, 4Cs, 협동학습, 컴퓨터교육, 알고리즘, 프로그래밍

The Influence of Pair Programming on 4Cs Improvement in Middle School Informatics

Yoon-mi Park* · Hyo Jong Lee**

Jeonbuk National University, Dept. of Computer Education*

Dept. of Computer Science and Engineering, CAIT**

Abstract

A model of cooperative learning, Pair Programming was applied to regular class for middle school students. Algorithm and programming skill were taught through the pair programming for two week and the achievement was analyzed based on 4Cs(Critical thinking, Communication, Collaboration, Creativity) proposed by the NEA(National Education Association). The control group was statistically significant on Critical Thinking between pre-test and post-test1, while the experimental group was statistically significant on all measurements of 4Cs. Between pre-test and post-test2, the control group showed overall little difference in range of 0.2 points on the 4Cs average scores. while the experimental group was statistically significant on all measurements of 4Cs. Between post-test1 and post-test2, there was no significant difference on 4Cs average scores in both groups. The experimental group showed higher score results on all 4Cs than the control group. This study verifies that the benefits of Pair Programming on 4Cs.

Keywords : Pair Programming, 4Cs, Cooperative Learning , Computer Education, Algorithm, Programming

교신저자 : 이효종(전북대학교 컴퓨터공학부, 영상정보신기술연구소)

논문투고 : 2020-06-19

논문심사 : 2020-06-25

심사완료 : 2020-07-08

1. 서론

오늘날 일상에 모바일, 사물 인터넷, 빅 데이터, 인공지능 등 제4차 산업혁명 기술들이 스며들면서 사람들은 미래사회를 대비할 수 있는 교육에 지대한 관심을 가지고 있다.

전미교육협회(National Education Association: NEA)[12]는 2015년에 21세기형 인재를 기르기 위해 필요한 역량으로 '4Cs'를 제시하였으며, 이는 비판적 사고(Critical thinking), 의사소통(Communication), 협업(Collaboration) 및 창의력(Creativity)을 의미한다.

미래 핵심역량으로 요구되는 4Cs의 중요성에 발맞추어 2015 정보과 교육과정[10]의 '컴퓨팅 사고력', '정보문화소양' 및 '협력적 문제해결력'에서 4Cs를 비중 있게 다루고 있다. 정보 교과 역량을 기르기 위해 학습자의 사회 정서적 발달을 고려하고, 교과에 대한 사전지식 유무, 흥미 및 학습수준을 이해하여 학습자에게 적절한 학습 환경을 제공할 필요성이 있다.

중학교 정보과 15개정 교육 과정에는 학습자가 엔트리나 스크래치라는 플랫폼에서 코드 일부가 적힌 블록을 기반으로 프로그래밍을 할 수 있도록 학습 환경을 지원하고 있다. 여러 연구자들이 엔트리와 스크래치 등을 활용한 소프트웨어(SW)교육 효과에 대한 연구를 진행한 결과, 프로그래밍 교육은 비판적사고력, 논리적사고력, 컴퓨팅 사고력 등 고차원적 사고력 배양에 효과가 있다고 밝혀졌다[6].

하지만 SW교육 연구는 고차원적인 사고력 발달에만 한정하지 않는다. 정보 교과 역량은 컴퓨팅 사고력뿐만 아니라 의사소통과 협업, 창의력도 고려한다. 오늘날 현대사회 조직은 대규모적이고, 복잡하여 전문화와 분업화가 되어있다. 이에 따라 조직 구조는 목표 달성 과정에서 조직 구성원들 간에 협력과 의사결정이 필수적이라는 점을 볼 때 SW교육에서 이러한 사회의 실제적 환경을 반영할 필요성이 있다.

그러나 엔트리나 스크래치 등을 사용하여 개별실습이 이루어지는 SW교육환경에서는 협동학습을 어렵게 한다. 엔트리나 스크래치는 동료들이 각자의 컴퓨터로 협력하여 공동의 목표를 달성할 수 있도록 하는 기능을 제공하지 않기 때문이다. 다시 말하면, 엔트리와 스크래치는 동료들이 컴퓨터를 각각 한 대씩 보유한 상황에서

팀 과제를 완성하기 위해 하나의 작업을 여러 개의 작은 부분들로 나누고, 각자 부분 작업을 수행한 후 부분작업들을 모아 하나의 작업으로 다시 합칠 수 있는 직소(Jigsaw) 같은 협동학습 모형을 기술적으로 지원하지 않는다. 또한 국내에서 짝 프로그래밍 수업에 대한 연구 사례는 많지 않으며, 짝 프로그래밍 수업 방법에 대한 연구가 잘 이루어지지 않은 실정이다.

본 연구는 중학생들을 대상으로 정보교과 수업에서 실제로 이루어질 수 있는 협동학습 모형인 짝 프로그래밍을 적용해보고, NEA에서 제시한 4Cs에 비추어 학습 효과를 분석하였다. 본 연구에서는 학생들이 자율적으로 진행되는 짝 프로그래밍 방식을 기본으로 하되, 학생들이 해답을 찾을 수 없을 때에는 교사의 도움을 받도록 하였다. 학습 후 4Cs결과에 어떤 차이가 발생하는지를 살펴보고, 결과에 영향을 주는 요인들을 분석하여 효과적인 짝 프로그래밍 방법을 제안하였다.

2. 이론적 배경

2.1. 짝 프로그래밍

짝 프로그래밍이란 한 대의 컴퓨팅 기기에서 문제를 해결하기 위해 2명의 프로그래머가 협력하여 하나의 문제 해결 프로그램을 완성하는 것이다. 즉, 2명의 프로그래머가 동일 프로그램의 설계, 알고리즘 개발, 코딩, 실험 및 디버깅 등을 분업의 형태로 개발한다.

짝 프로그래밍의 분업 역할은 드라이버(Driver)와 네비게이터(Navigator)로 나타난다. 이 둘은 각자에게 주어진 역할을 수행하는데, 드라이버 역할은 컴퓨터 키보드나 마우스 등 기기를 사용하여 프로그램을 작성한다. 네비게이터 역할은 알고리즘과 프로그래밍을 하는 과정에서 전략과 전술을 세우며, 드라이버의 타이핑 실수나 문법적 오류를 보완하여 협력한다. 드라이버는 파트너인 네비게이터의 전략에 대해 질문하고 설명을 들을 수 있으며, 네비게이터는 프로그래밍 작업의 방향을 보다 객관적으로 접근할 수 있다[2][7]. 두 역할은 주어진 시간이 만료되거나 소작업을 완료했을 때 역할 교환이 이루어질 수 있다.

정은우 외[4] 연구에서 짝 프로그래밍의 효과는 짝협업(Pair negotiation)와 짝 용기(Pair Courage), 짝 디버깅(Pair debugging) 능력이 나타날 수 있다고 하였다.

2.2. 4Cs

짝 협의와 짝 용기, 짝 디버깅 과정을 거치는 짝 프로그래밍은 4Cs 각 요소들에 영향을 미칠 것이다. 한 팀을 구성하는 2명의 학습자들은 1대의 컴퓨터를 사용하여 프로그래밍 할 때 필연적으로 의사소통이 이루어질 것이고, 서로의 생각을 교환하는 과정에서 현재 상황에 대해 비판적 사고와 창의력이 일어날 것이기 때문이다.

짝 프로그래밍 역할에 충실히 하다보면 협업을 발휘하게 될 것이다. 그러므로 4Cs를 구성하는 비판적사고와 의사소통, 협업, 창의력에 대해 정의를 명확히 할 필요가 있다. 다음은 4Cs 구성요소의 정의이다[12].

(1)비판적 사고 : 학습 경험에 대해 비판적으로 생각할 줄 아는 역량을 일컫는다. 상황에 적절하게 귀납법, 연역법 등 다양한 추리방법을 사용할 수 있다.

(2)의사소통 : 상황에 적절하게 구두, 서면, 비언어적 스킬 등을 사용하여 생각과 아이디어를 효과적으로 표현할 수 있는 역량이다. 이러한 역량은 지식, 가치, 태도, 의도 등 의미를 잘 파악하고, 다양한 목적을 가지고 정보를 알리고, 동기부여하고, 설득할 수 있다.

(3)협업 : 다양한 팀들과 함께 효과적으로 일하고, 존중할 수 있는 역량을 말한다. 팀 멤버들의 개인적인 헌신을 소중히 생각하며, 공동 작업에 대한 책임 의식을 공유한다.

(4)창의력 : 새롭고 가치 있는 다양한 아이디어를 창조해내는 역량이다. 창의력은 새로운 아이디어를 정교하게 만들고, 효과적으로 개발, 구현 및 전달할 수 있다.

3. 관련 연구

짝 프로그래밍에 대한 연구는 초기 SW개발과 더불어 활발하게 이루어져 왔다. Jill Denner 외[1] 연구에서 게임 프로그래밍을 수행한 320명의 중학생들로부터 데이터를 수집하였다. 솔로 프로그래밍과 비교했을 때 짝 프로그래밍은 특히 컴퓨터 경험이 적은 학생들에게 컴퓨팅 사고력과 프로그래밍 지식에 이점이 있었다. 또한 파트너보다 더 많은 경험을 가진 학생들은 컴퓨터에 대한 자신감과 긍정적인 태도를 더욱 성취하였다.

Thomas 외[14] 연구에서는 프로그래밍 경험이 있는 학생들에게 프로그래밍 흥미와 수행능력의 자기 인식에

관한 설문지 작성을 요청하였다. 설문지를 토대로 프로그래밍 인식에 대해 유사한 태도 또는 정반대의 태도를 가진 학생들끼리 그룹을 배치시킨 후 짝 프로그래밍을 수행했다. 연구 결과, 대부분 학생들은 짝 프로그래밍이 과제 수행에 도움이 된다고 생각했다. 비슷한 레벨의 자신감을 가진 학생들끼리 짝지어졌을 때는 최상의 작업을 보였다고 하였다. 하지만 프로그래밍에 대한 자신감이 넘치는 학생들은 다른 학생들보다 협력을 선호하지 않아 짝 프로그래밍을 좋아하지 않았다고 보고하였다.

Lewis[9] 연구에서는 초등학교 6학년생을 대상으로 짝 프로그래밍 방식과 간헐적 협력 방식의 차이점을 비교했다. 짝 프로그래밍 방식에서는 5분마다 한 대의 컴퓨터에서 키보드와 마우스를 파트너가 번갈아 교환했다. 간헐적 협력방식은 학생들이 자신의 컴퓨터로 각각 작업하는 대신 강사에게 도움을 요청하기 전에 파트너에게 반드시 문제를 논의하고, 5분마다 프로그래밍 진행과정을 논의하기 위해 프로그래밍을 멈추어야 했다. 연구 결과, 짝 프로그래밍 방식은 지원과 협력 측면에서 우수하였고, 간헐적 협력방식은 개인적 노력으로 실력을 향상시킬 수 있는 기회를 제공하는 점에서 우수하였다.

Linda Werner 외[15] 연구에서 189명의 중학생들을 대상으로 짝 프로그래밍을 10주간 실시하여 프로그래밍 경험, 우정 점수, 프로그래밍 지식 및 자신감을 측정하였다. 20분마다 역할 교환이 이루어졌으며, 연구 결과에서 자신감이 비교적 적은 학생은 파트너가 친구인 경우에만 자신감이 많은 파트너에게 영향을 미치는 것을 발견했다. 또한 자신감이 많은 파트너는 Alice 프로그래밍 초기 지식을 상대적으로 많이 가진 친구와 짝을 이룰 때 프로그래밍 지식이 가장 많이 증가함을 보였다.

짝 프로그래밍은 드라이버와 네비게이터가 자신의 역할에 단순히 충실하기만 해서 교육효과가 발휘된다는 보장은 없다. Zhong 외[16] 연구에서 짝 프로그래밍 효과에 영향을 주는 요인을 파악하였고, 과제의 복잡성, 파트너의 스킬과 경험, 파트너의 학습스타일, 파트너의 성격과 기질, 사회적 요인으로 분류했다.

국내 사례인 정충교[3] 연구에서는 짝 프로그래밍의 학습 태도를 측정하였다. 짝 배정방법으로는 대학생들이 언제든지 희망하는 짝과 바꿀 수 있도록 했다. 하지만 학생들은 한 번 짝이 정해진 이후 짝을 자주 바꾸려 하지 않았고, 일정 시간마다 역할 교대를 위해 자리 바

꾸는 것을 학생들이 귀찮게 여겨서 교사는 자리를 바꾸어 앉는 걸 독려해야만 했다. 그리고 설문을 토대로 한 학생들의 짝 프로그래밍 생각은 학습에 도움이 되고, 내년에도 적용하면 좋겠다는 등 약 70%가 긍정적인 평가를 하였다. 그러나 역할 교대 효과에 대한 평가는 약 27%만 긍정적인 평가를 하였다.

김정량[7] 연구에서는 프로그래밍에 대한 경험이 있는 초등학생들을 대상으로 엔트리 도구를 이용해서 짝 프로그래밍인 실험집단과 통제집단에 순차/반복/선택 학습을 각각 실시하였다. 연구 결과, 짝 프로그래밍이 창의적 문제해결력의 하위요소인 비판적·논리적 사고력에 효과가 있음을 확인할 수 있었고, 완성된 프로그램의 효율성에 대해 유의미한 차이가 있는 것으로 나타났다.

정향연 외[5] 연구는 기초 학습의 자질이 부족한 특성화 고등학교 학생들을 대상으로 웹 프로그래밍 교과 수업에 짝 프로그래밍을 적용하여 학습 동기를 유발하고자 하였다. 연구 결과는 실험집단이 통제집단보다 주의집중과 자신감 요소에서 유의미한 차이를 보여 학습 동기 유발에 효과가 있었다. 지필평가 점수에서는 실험집단과 통제집단 간에 유의미한 차이는 없었으나, 실습평가 6회 중 3회에서 통계적으로 유의미한 차이가 있었고, 평균점수는 6회 모두 실험집단이 통제집단보다 높게 나타났다.

김용욱 외[8] 연구에서 협력적 문제해결력과 코딩이해도를 높이기 위해 협업도구를 개발하여 짝 프로그래밍에 활용하였다. 초등학생 5·6학년들을 대상으로 동일한 코딩 능력 수준을 갖춘 학생들끼리 짝을 구성하고, 역할분담을 했다. 그리고 짝 프로그래밍에 적용할 수 있는 훈련기간을 둔 후 짝 프로그래밍을 실시했다. 연구 결과, 협업도구 유무로 나뉜 실험집단과 통제집단은 협력적 문제해결력과 코딩이해도에서 모두 유의하게 향상했다. 협력적 문제해결력과 코딩이해도에 대한 점수 상승폭은 통제집단보다 실험집단이 더 큰 것으로 나타났다.

앞의 연구 사례들을 종합하면 짝 프로그래밍은 교육적 효과가 있음을 알 수 있다. 학생들은 짝 프로그래밍을 경험하면서 자신감과 긍정적 태도를 얻을 수 있으며, 지원과 협력, 사고력 측면에서 학습 성취에 긍정적 영향을 받는다. 그러나 국내에서 수행된 짝 프로그래밍 연구에서 [2][3][4][5][7][8] 중학생을 대상으로 진행된 연구는 찾아볼 수 없었다.

또한 2015개정 교육과정의 현장 안착을 위해 한국교육학술정보원(Korea Education & Research Information Service)[11]은 정보교과의 교수학습 자료를 제공하고 있으며, 교수학습 자료에는 짝 프로그래밍의 정의와 진행시 유의사항 등이 소개되어있다. 그러나 실제 수업 현장에서 적용할 수 있는 교수학습방법으로서의 구체적인 내용은 설명되어 있지 않다. 짝 프로그래밍 모형을 학교 수업에 실제로 적용한 국내 연구 사례는 해외 연구에 비해 적은 편이다. 정은우 외[4]는 짝 프로그래밍과 관련된 선행연구가 부족하여 교육방식 등을 확립하기엔 아직 이른 상황이라고 지적했으나, 최근 들어 다양한 선행연구가 수행되었다. 그러나 체계적인 짝 프로그래밍 교육방식의 확립이 미진한 것은 사실이다. 한건우 외[2]에 따르면 짝 프로그래밍 본질은 학습을 위해 제작된 것이 아니라 소프트웨어 개발의 효율성을 높이기 위해 만들어진 것이기 때문에 짝 프로그래밍의 교육적 효과를 높이기 위해 수정이 불가피하다고 하였다.

국내에서는 중학생에게 효과적인 학습을 위한 짝 프로그래밍 방법에 대해 보편화되지 않은 실정이다. 이에 따라 각 연령의 눈높이에 맞추어 효과적인 짝 프로그래밍을 수행하도록 하는 여러 제반 조건과 요인들에 대한 분석의 한계를 가지고 있다.

먼저 학습자의 연령과 선수학습정도 등 요인들이 고려되지 않고, 드라이버와 네비게이터 역할이 분담되고 있다. 정보수업에서 역할 분담에 따라 짝 프로그래밍이 잘 이루어지려면 학생들은 동일한 수업시간 동안 얻는 학습 내용과 지식이 균등하도록 유의할 필요가 있다. 그리고 설계, 코딩, 디버깅 등 세분화시켜 드라이버와 네비게이터 역할을 정하여 프로그래밍 하는 일은 프로그래밍에 이미 경험이 있고, 익숙한 사람들에게 가능한 것이다. 정보교과의 수업환경은 프로그래밍의 기초부터 배워나가야 할 중학생들이 대상이며, 정보수업에서의 실제적 적용을 위해 2015교육과정과정에 맞춘 짝 프로그래밍 수업구성이 불가피하다.

본 논문에서는 짝 프로그래밍을 위한 수업지도안을 구성하고, 학생들의 학습 수준을 고려하여 역할을 조정할 후 연구가 실시될 것이다. 연구 결과에 대한 평가 및 분석은 4Cs에 비추어 이루어질 것이다.

4. 연구 방법

4.1. 연구 절차

연구 실험 기간에 수행했던 일들을 순서대로 요약하였다. 차시별 수업방식과 검사 절차 및 분석을 고려하였다. 첫째, 1차시 수업은 엔트리 기초를 배우는 시간으로 학생들은 엔트리 로그인 방법과 오브젝트 개념을 익히며, 간단한 블록들을 이용하여 개별 실습하였다. 2-3차시는 짝 프로그래밍 학습지도안을 수업에 적용했고, 검사를 수행하였다. 둘째, 4Cs 점수 변화를 알아보기 위해 사전-사후 설문지를 사용하였다. 사전검사는 수업시작 직전에 실시하고, 사후검사는 수업종료 직전에 실시하였다. 셋째, 짝 프로그래밍 수업의 효과성을 분석해보았다. 작성된 설문지들을 토대로 통제집단과 실험집단 간의 차이를 비교·분석하였다.

4.2. 연구 설계 및 검사

4Cs 점수 변화를 알아보기 위해 설문지는 5단계 Likert Scales를 사용했다. 설문지의 문항 수는 비판적 사고 3문항, 의사소통 3문항, 협업 3문항, 창의력 1문항으로 총 10문항으로 구성되었다. 문항 내용은 공학설계에 대한 인식 조사 연구[13]에서 쓰인 문항들(문제 정의 영역, 정보수집 및 활용 영역, 탐구수행 영역, 협동 영역, 의사소통 영역) 중에 4Cs에 관한 문항들로 적절히 선정·조합하여 문항을 구성하였다. 교육 전문가 6인을 통해 검증받은 문항에 대한 Cronbach α 계수의 값이 .961로 나타나 신뢰 수준을 검증하였다.

<Table 1> 4Cs Measurement Question Example[13]

4Cs	Survey items
Critical thinking	I can interpret and conclude the data obtained while solving the inquiry problem.
Collaboration	I can reconcile with other member when disagreements happen.
Communication	In the course of resolving the inquired problem, I can exchange opinions with other member to solve the problem.
Creativity	I can find research topics in everyday life or in nature.

실험대상은 통제집단과 실험집단이 있다. 두 집단의 4Cs 점수 차이를 분석하기 위해 두 집단의 분산이 동일함을 가정해야했다. 그에 따라 Levene's test와 Independent T test를 실시하였다. 그리고 사전-사후1-사후2검사로 반복측정이 이루어졌으므로 반복측정 분산 분석(Repeated Measured ANOVA)을 이용하였고, 4Cs 점수를 시점과 집단을 기준으로 비교했다. 대응표본 T 검정(Paired-Samples T test)을 실시하여 집단 내에서 시점별 4Cs 점수 변화를 살펴보았다.

4.3. 연구 가설

시간이 경과함에 따라 전체 집단에서 4Cs 점수 변화가 일어나는지 가설을 설정하였다. 만약 전체집단에서 점수 변화가 있다면 집단별로 4Cs 점수에도 차이가 있는지 알아보았다. 각 집단 내에서 4Cs 점수는 어떤 변화가 있는지에 대한 가설을 설정했다.

<Table 2> Experimental hypothesis

Criteria	Hypothesis
Score by time	Each score of 4Cs will increase over time.
Group score	For each score of 4Cs, the experimental group will be higher than the control group.
In-group score	Each score of the 4Cs component in the experimental group will increase over time.

4.4. 연구 실험

4.4.1. 연구 대상 및 기간

전주에 위치한 △중학교 1학년 학생들을 대상으로 연구를 수행하였으며, 실험은 2주 동안 이루어졌다. 무작위로 실험집단 30명과 통제집단 28명으로 구성하였다. 실험집단은 드라이버와 네비게이터 역할로 구분한 짝 프로그래밍을 적용한 집단이며, 통제집단은 기존의 강의식 실습을 적용하여 프로그래밍 교육을 하는 집단이다.

실험집단, 통제집단 모두 엔트리에 대한 사전지식은 전문하였고, 실험 전 학습능력은 동등하였다. 이에 대해 집단별 사전검사의 분산 동일성을 나타내는 <Table 3>

에서 확인할 수 있다. ‘Levene의 등분산 검정’ 항목에서 4Cs의 유의확률이 모두 .05보다 크기 때문에 귀무가설은 채택되어 실험집단과 통제집단의 분산이 동일한 것으로 나타났다. 등분산이 가정됨에 따라 ‘평균의 동일성에 관한 t-검정’ 항목을 보면 4Cs 유의확률이 유의수준 =.05보다 크기 때문에 연구가설은 기각되고 귀무가설은 채택되어 ‘사전검사에서는 두 집단 간의 4Cs 점수의 차이가 없다’라고 할 수 있다.

<Table 3> Test of homogeneity

Measure	Levene's Test for Equality of Variances		t-test for Equality of Means		
	F	Sig.	t	df	Sig.
Critical thinking	.01	.926	.34	56	.738
Communication	.72	.397	-.12	56	.906
Collaboration	.60	.442	.95	56	.348
Creativity	3.18	.080	.92	56	.359

4.4.2. 수업구성

수업시간은 총 45분으로 20분은 교과서 위주 수업을 하였고, 25분은 엔트리 기반의 실습을 하였다. 다음 실습 규칙은 통제집단과 실험집단에 대해 공통으로 적용되었다. 학생들은 엔트리의 사전지식이 전무함에 따라 교사 주도의 실습이 이루어졌다. 엔트리의 완성작품을 빔 프로젝터 스크린으로 학생들에게 제시한 후 실습을 시작하였다. 교사가 블록을 옮기면 학생들은 스크린을 보고 따라서 블록을 옮겼다. 학생들이 하나의 작은 기능을 만들 때마다 결과를 출력해볼 수 있는 시간을 주었고, 작품이 점점 완성되어가는 과정을 확인하도록 하였다.

실습 규칙에 있어 통제집단과 실험집단에 대한 차이는 다음과 같았다. 통제집단은 한 학생 당 컴퓨터 1대를 사용하였고, 실습 중에 학생들이 질문할 때마다 교사는 일대일로 지도했다. 문제에 봉착할 경우 통제 집단에서는 교사가 도와주지만, 실험 집단에서는 <Table 4>에 나와 있는 규칙과 같이 구성원이 먼저 해결을 시도하고 문제를 해결할 수 없는 경우에만 교사의 도움을 받았다.

짝 프로그래밍을 적용한 실험집단의 경우, 규칙 설명은 수업 당일에 <Table 4> 내용을 보드지에 활용하여

학생들에게 알려주었다. 드라이버와 내비게이터 역할에 대한 사전훈련은 따로 하지 않았으며, 짝 프로그래밍 방식을 연구 수업에 바로 적용하였다.

<Table 4> Rules of pair programming

Rules	
1	Two people pair and share a single computer. Both students sit side by side.
2	Each student takes the controls of mouse-keyboard in turn.
3	If a partner has difficulty during the learning process, other student explains and helps a partner.
4	Each student solves problems with a partner first. If both member can't solve the problem, then the pair ask a teacher.

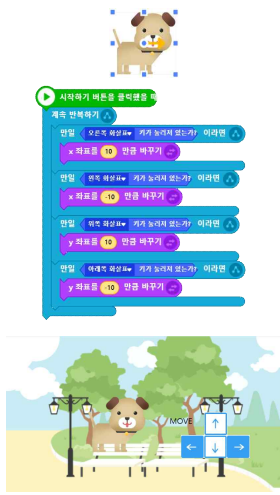
실험집단은 두 명의 학생이 1대의 컴퓨터를 같이 보며 의사소통할 수 있도록 두 명의 학생 중앙 위치에 컴퓨터를 두었다. 그리고 마우스와 키보드 사용 횟수를 1회 간격으로 두고, 매1회마다 두 사람이 드라이버와 내비게이터 역할을 번갈아 교환했다.

드라이버가 마우스, 키보드를 1회 사용하면 드라이버는 내비게이터 역할로 바꾸고, 내비게이터인 사람은 드라이버 역할로 바꾸어서 마우스, 키보드를 1회 사용하였다. 일정 시간(예로써 15분)마다 역할을 교환하는 대신 일정 횟수(1회)마다 역할을 교환하였으므로 학생들은 프로그래밍 과정에서 시계를 자주 들여다보지 않아도 되었다. 또 역할을 교환할 때 의자를 바꾸어 앉는 대신 마우스, 키보드만 교환하는 방식으로 하여 프로그래밍에 집중할 수 있도록 번거로움을 줄였다.

4.4.3. 프로그래밍 실습

가. 2차시 프로그래밍

2차시 수업은 ‘알고리즘’ 단원으로 구성하였다. 학습 주제는 ‘알고리즘의 의미와 표현’이다. 학습 목표는 ‘알고리즘의 의미와 중요성을 이해하고, 간단한 문제의 해결과정을 나열할 수 있다’와 ‘사용할 프로그래밍 언어의 프로그램 개발 환경의 구성 요소를 나열할 수 있다’, ‘자료의 입력과 출력을 위한 명령문을 작성할 수 있다’로 이루어진다[10].



(Fig. 1) Code of doggy walking.



(Fig. 2) Code of feeding penguin.

엔트리로 ‘강아지 산책하기’ 작품을 프로그래밍 하는 시간을 가졌다. 이 작품은 키보드 입력이 있을 때마다 강아지 오브젝트 위치를 이동시킬 수 있는 처리가 이루어지고, 이러한 처리가 화면에 출력된다.

나. 3차시 프로그래밍

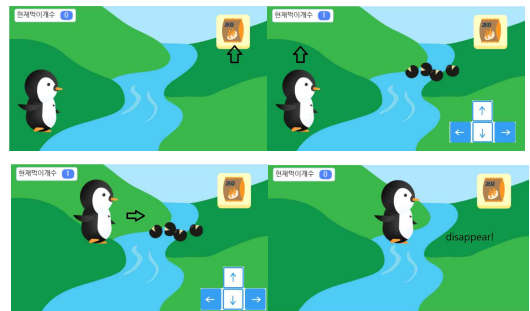
3차시 수업의 학습 단원은 ‘프로그래밍’이었다. 학습 주제는 ‘프로그래밍의 이해’, ‘변수의 정의와 이해’, ‘연산의 정의와 이해’가 된다.

학습 목표는 ‘프로그래밍의 기본 개념을 설명할 수 있다’가 된다. 그리고 ‘다양한 형태의 자료를 입력 받아 처리하고 출력하는 프로그램을 작성할 수 있다’, ‘변수와 연산의 개념을 설명할 수 있다’, ‘변수와 연산을 활용한 프로그램을 작성할 수 있다’로 이루어진다[10].

3차시는 학습목표와 부합되도록 변수와 연산이 쓰이는 블록들을 사용하는 ‘펭귄 먹이주기’ 작품을 프로그래밍 하였다.

작품은 키보드 입력에 의해 펭귄 오브젝트의 위치를 이동시켜 산책할 수 있고, 펭귄이 먹이 오브젝트를 향해 움직여서 먹이에 닿았을 때 화면에 먹이가 없어지는 내용이다. 새로운 먹이를 생성하려면 과자 오브젝트를 클릭하면 되며, 이때 변수와 연산이 사용된다. 변수는 화면에 존재하는 현재 먹이개수를 조절할 수 있는 역할이며, 연산은 펭귄의 이동 이외에도 먹이 생성 위치와 먹이 개수에 활용된다.

(Fig. 2)의 실행 결과는 다음 (Fig.3)에서 확인할 수 있다.



(Fig. 3) The results of execution.

4.5. 결과 및 토론

4.5.1 연구 결과

가. 집단 내 사전-사후1검사 점수 비교

통계집단과 실험집단으로 구분된 사전검사와 사후1검사를 점수에 대해 비교하였다. 사전검사는 실험 전에 집단의 학습능력의 동등함을 검정하기 위한 검사이며, 사후1검사는 알고리즘단원으로 구성된 2차시수업이 이루어진 뒤 실시한 검사이다. 검사 측도는 4Cs(비판적사고와 의사소통, 협업, 창의력)가 해당된다. 4Cs에 준하여

사전검사와 사후1검사를 비교분석한 결과 <Table 5>와 같이 나타났다.

<Table 5> 4Cs score differences by groups in the pre-test and post-test1.

Measure	Test	M	SD	Mean Difference	Sig.
Control Group					
Critical thinking	pre	3.36	.66	.43	.013*
	post1	3.79	.57		
Communication	pre	3.52	.71	.23	.271
	post1	3.75	.75		
Collaboration	pre	3.96	.80	-.08	.698
	post1	3.88	.71		
Creativity	pre	3.32	.86	.39	.062
	post1	3.71	.71		
Experiment Group					
Critical thinking	pre	3.30	.63	.78	.000**
	post1	4.08	.65		
Communication	pre	3.54	.62	.58	.002**
	post1	4.12	.70		
Collaboration	pre	3.78	.70	.44	.024
	post1	4.22	.72		
Creativity	pre	3.13	.68	.90	.000**
	post1	4.03	.77		

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

<Table 5>의 통제집단 결과를 살펴보았을 때 비판적 사고는 t=2.65, 유의확률=.01로 유의수준 α =.05보다 작기 때문에 귀무가설은 기각되어 사전검사-사후검사1 간에 차이가 있는 것으로 나타났다. 비판적사고의 평균점수는 사전 3.36점, 사후1 3.79점으로 사전검사보다 0.43점이 상승했음을 알 수 있다.

실험집단 결과에서는 비판적사고와 의사소통, 협업, 창의력 모든 측정에서 유의확률이 유의수준 α =.05보다 작게 나타나 통계적으로 유의하였다.

실험집단의 비판적사고 평균점수는 사전 3.30점, 사후1 4.08점으로 사전검사보다 0.78점 상승하였다. 의사소통 평균점수는 사전 3.54점, 사후1 4.12점으로 사전검사보다 0.58점 상승하였으며, 협업에서는 사전 3.78점, 사후1 4.22점으로 사전검사보다 0.44점 상승하였다. 창의

력에서는 사전 3.13점, 사후1 4.03점으로 사전검사보다 0.90점 상승하였다.

실험집단의 사전검사-사후1검사 간에 평균점수 차는 창의력(0.90)>비판적사고(0.78)>의사소통(0.58)>협업(0.44) 순으로 나타났다.

나. 집단 내 사전-사후2검사 점수 비교

사후1검사보다 일주일이 경과한 후에 실시한 사후2검사를 사전검사의 점수와 비교하였다. 사후2검사는 프로그래밍단원으로 구성된 3차시수업에서 변수와 연산을 학습 후에 실시한 검사이다. 사전검사와 사후2검사 점수를 비교분석한 결과를 <Table 6>에서 살펴볼 수 있다.

<Table 6> 4Cs score differences by groups in the pre-test and post-test2

Measure	Test	M	SD	Mean Difference	Sig.
Control Group					
Critical thinking	pre	3.36	.66	.23	.275
	post2	3.58	.75		
Communication	pre	3.52	.71	.26	.189
	post2	3.79	.61		
Collaboration	pre	3.96	.80	-.21	.323
	post2	3.75	.70		
Creativity	pre	3.32	.86	.29	.255
	post2	3.61	.74		
Experiment Group					
Critical thinking	pre	3.30	.63	.83	.000**
	post2	4.13	.78		
Communication	pre	3.54	.62	.57	.004**
	post2	4.11	.79		
Collaboration	pre	3.78	.70	.63	.001**
	post2	4.41	.69		
Creativity	pre	3.13	.68	.70	.008**
	post2	3.83	1.10		

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

<Table 6>의 통제집단 결과에서는 비판적사고, 의사소통, 협업, 창의력 모든 측정에서 유의확률이 유의수준

$\alpha=.05$ 보다 크기 때문에 귀무가설을 기각하지 못하였다.

통제집단의 평균점수 차이가 전체적으로 0.2점대로 근소하게 나타나 별다른 변화가 없는 것으로 보인다.

실험집단 분석결과에서는 비판적사고, 의사소통, 협업, 창의력 모든 측정에서 유의확률이 유의수준 $\alpha=.05$ 보다 작게 나타나 통계적으로 유의하였다.

실험집단의 비판적사고 평균점수는 사전 3.30점, 사후2 4.13점으로 사전검사보다 0.83점 상승하였다. 의사소통 평균점수의 경우 사전 3.54점, 사후2 4.11점으로 사전검사보다 0.57점 상승하였다. 협업에서는 사전 3.78점, 사후2 4.41점으로 사전검사보다 0.63점 상승하였다. 창의력의 평균점수는 사전 3.13점, 사후2 3.83점으로 사전검사보다 0.70점 상승하였다.

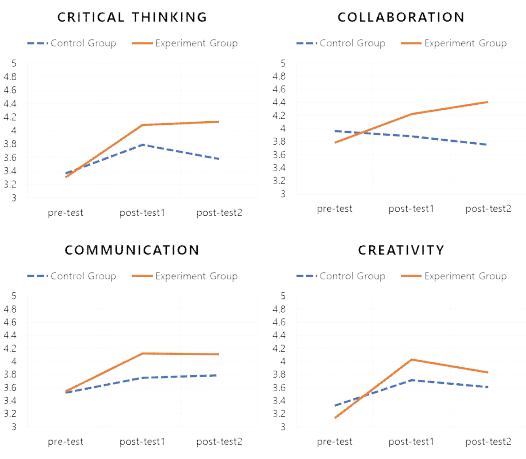
실험집단의 사전검사-사후2검사 간에 평균점수 차는 비판적사고(0.83)>창의력(0.70)>협업(0.63)>의사소통(0.57)순으로 나타났다.

다. 집단 내 사후1-사후2검사 점수 비교

사후1검사와 사후2검사의 비교 결과, 모든 집단에서 4Cs측도의 유의확률이 유의수준 $\alpha=.05$ 를 초과하여 통계적으로 유의하지 않았다.

4.5.2 연구 결과 분석

각 집단에 대한 실습시점의 경과에 따른 4Cs 점수는 (Fig. 4)에서 살펴볼 수 있다.



(Fig. 4) 4Cs score trends between groups

4Cs점수는 3-4점대 구간에 속해있었다. 모든 집단에서 사전검사에 대비해서 사후1·2검사의 4Cs점수가 상승하는 추세를 보였으며, 점수의 상승 폭은 통제집단보다 실험집단이 더 크게 나타났다.

짝 프로그래밍인 실험집단과 통제집단 두 집단에 대해 실시된 사후1검사, 사후2검사를 실험 전과 비교했을 때 4Cs점수는 대체로 향상되었다. 그리고 모든 사후검사에서 실험집단이 통제집단보다 4Cs의 모든 항목에서 더 높은 점수 결과를 보였다.

실험 결과를 통해 엔트리에 대한 사전지식이 없는 중학생에게는 ‘자료의 입력과 출력’, ‘변수’, ‘연산’ 등을 학습하는 알고리즘과 프로그래밍 단원에 대해 기존의 학습방식보다 짝 프로그래밍방식이 4Cs에 더 긍정적인 영향을 줄 수 있음을 판단할 수 있다.

앞서 언급된 선행연구들은 짝 프로그래밍 결과에 대해 프로그래밍 지식과 비판적·논리적 사고력, 주의집중과 자신감 등 유의미한 효과가 있음을 확인했다. 본 연구의 짝 프로그래밍 결과인 4Cs에도 학습효과가 있음을 확인했다.

프로그래밍의 수행이나 자신감 측면에서 동일 수준을 가진 학생들끼리 그룹을 지어 짝 프로그래밍 한 연구 [8][14]에서 협력적 문제해결력과 코딩 이해도, 작업 능률이 올라갔음을 보고했다. 이와 관련하여 본 연구에서 프로그래밍의 사전경험이 없고, 동일한 수행 수준을 가진 학생들이 대상이었다는 점이 같다고 볼 때, 4Cs 학습효과가 향상되고 있는 점이 일치하고 있다.

여러 선행연구들과 본 연구를 비교했을 때 프로그래밍 도구(예로써 엔트리)와 학생들의 프로그래밍 사전 경험 유무, 역할 교환, 교수법에서 차이가 있었다. 선행 연구 [1][3][7][8][14][15]에서는 프로그래밍 경험이 있는 학생들을 대상으로 함으로서 일정한 시간 간격마다 역할 교환을 했으며, 짝 프로그래밍이 자율적으로 이루어질 수 있었다. 본 연구에서는 일정한 시간 대신에 기기 사용 횟수를 기준으로 역할 교환을 했고, 교사 주도의 짝 프로그래밍 방식으로 진행함에 따라 프로그래밍 사전경험이 없는 학생들을 대상으로도 수업을 진행할 수 있었다.

내용을 종합해보았을 때 짝 프로그래밍은 중학생의 프로그래밍 지식이 부족하거나 전무한 상황일 때에도 4Cs는 효과를 보일 수 있다. 또한 프로그래밍 지식이

비슷한 파트너끼리 짝 프로그래밍을 수행할 때 더욱 효과적이다[14]. 이는 청소년기의 정서발달상 짝 프로그래밍이 학생들의 자신감에 도움을 주면서 비판적사고와 의사소통, 협업, 창의력이 요구되는 프로그래밍 상황에 적절히 행동하고, 발휘할 수 있다는 기대와 신념을 반영한 것으로 판단된다.

짝 프로그래밍 효과는 교실 내에서 여러 요인들이 서로 복잡하게 얽힌 상호작용을 통해 일어난다[16]. 실험집단에서 사전검사와 사후1검사 간에 점수 차이와 사전검사와 사후2검사 간에 점수 차이는 모두 비판적사고와 창의력항목이 중·상위권에 위치하였고, 협업과 의사소통 항목이 중·하위권에 위치하였다. 비판적사고와 창의력은 협동학습에 관련된 의사소통과 협업보다 더 향상된 점수 결과를 보였다. 이러한 결과는 짝 프로그래밍 수업이 의사소통과 협업보다는 비판적사고와 창의력 중심으로 영향을 주었을 것이라고 추측된다.

사후1검사와 사후2검사 간의 점수 차이는 통제집단과 실험집단 모두에서 유의한 변화를 보여주지 않았다. 두 검사는 사전검사에 비해 4Cs 점수는 전체적으로 향상되었음을 확인했으나 두 검사 사이의 시간 격차가 작아서 학습 발전 정도가 드러나지 않은 상태로 판단되며, 충분한 학습을 수행한 뒤에 사후2검사를 시행할 경우에는 다른 결과를 볼 수도 있을 것으로 보인다.

결과적으로 4차 산업혁명시대를 대비해야하는 중학생들에게는 단지 프로그래밍 기술만을 배우는 것 이상이 필요하다. 본 연구 결과가 보여주는 바와 같이 알고리즘과 프로그래밍에 대한 초기경험을 짝 프로그래밍을 통해 시작한다면 후호적인 분위기 속에서 비판적 사고력이나 의사소통 능력은 물론, 협업능력과 창의력의 향상을 가져오는 것을 확인하였다.

5. 결론

본 연구는 엔트리에 대한 사전지식이 전무한 중학생들을 대상으로 하여 교사 주도의 짝 프로그래밍을 진행하였다. 알고리즘과 프로그래밍 단원을 기반으로 짝 프로그래밍을 수업에 설계 및 적용했고, 4Cs에 비추어 학습효과를 분석하였다. 학습자들의 눈높이를 고려하면서 비판적 사고와 창의력 같은 인지적 영역 뿐 아니라 정의적 영역인 의사소통, 협업 능력에 효과를 주고, 중학

교 정보수업에 보편화할 수 있는 짝 프로그래밍 수업을 설계하고자 하였다.

연구 분석한 결과 짝 프로그래밍은 4Cs 학습에 효과가 있는 것으로 나타났다. 실험집단은 모든 사후검사에서 통제집단보다 4Cs의 모든 측정에 높은 점수 결과를 보였다. 이에 따라 짝 프로그래밍은 4Cs의 4가지 요소인 비판적사고, 의사소통, 협업, 창의력에 효과가 있음이 확인되었다.

다만 본 연구는 짧은 수업 기간 중에 진행되었다는 한계를 지니고 있다. 교육 효과의 잠재적 성격 측면에서 수업 전후에 짝 프로그래밍 효과를 학생들의 자기보고식 설문 검사를 통하여 연구 결과를 분석한 사실을 유념할 필요가 있다.

참고문헌

- [1] Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students?. *Journal of Research on Technology in Education*, 46(3), 277-296.
- [2] Han, K. W., Lee, E. K., & Lee, Y. (2006). The Effects of Pair Programming on Achievement and Motivated Strategies in Programming Course. *The Journal of Korean association of computer education*, 9(6), 19-28.
- [3] Jeong, C. K. (2018). Pair Programming in Programming Lab: The Effects, Limits, and Guidelines Based on the Student Receptivity. *Journal of Digital Contents Society*, 19(9), 1663-1669.
- [4] Jung, E. W., Yi, S. Y., & Lee, Y. J. (2018). A Study on Pair Programming Applications in Education Implications Reviews. *Proceedings of The KACE*, 55-56.
- [5] Jung, H. Y., & Nam, Y. H. (2018). A Design and Evaluation of Learning Motivation Strategy Based on Pair Programming: Focused on the Specialized High School. *Journal of Knowledge Information Technology and Systems*, 13(6), 687-694.

[6] Kim, D. M., & Lee, T. W. (2018). A Meta-Analysis on the Effects of Educational Programming Language on High-level Thinking. *Journal of the Korea Society of Computer and Information*, 23(6), 81-89.

[7] Kim, J. (2016). Effects of Pair Programming on Creative Problem-Solving Ability and Efficiency. *Journal of The Korean Association of information Education*, 20(1), 21-28.

[8] Kim, Y. O., & Chun, S. J. (2019). SW Education Program using Pair Programming Collaboration Tools. *Journal of The Korean Association of information Education*, 23(4), 375-384.

[9] Lewis, C. M. (2011). Is pair programming more effective than other forms of collaboration for young students?. *Computer Science Education*, 21(2), 105-134.

[10] Ministry of Education. (2015). Practical Course(Technology and Home Economics) / Computer and Information Curriculum. Proclamation of the Ministry of Education 2015-74.

[11] Ministry of Education, Daejeon Metropolitan Office of Education, KERIS. (2016). Computer and Information 2015 Curriculum teaching material. GyeongGi: KBUWEL.

[12] National Education Association. (2015). Preparing 21st Century Students for a Global Society. National Education Association.

[13] Song, S. C., Han, H. J., & Shim, K. C. (2017). A Study on Perceptions of Scientifically Gifted Middle School Students about Engineering Design Process. *Journal of The Korean Association For Science Education*, 37(5), 835-846.

[14] Thomas, L., Ratchliffe, M., & Robertson, A. (2003). Code warriors and code-a-phobes: a study in attitude and pair programming. *ACM SIGCSE Bulletin*, 35(1), 363-367.

[15] Werner, L., Denner, J., Campe, S., Ortiz, E., DeLay, D., Hartl, A. C., & Laursen, B. (2013, March). Pair programming for middle school students: does

friendship influence academic outcomes?. *In Proceeding of the 44th ACM technical symposium on Computer science education*, 421-426.

[16] Zhong, B., Wang, Q., Chen, J., & Li, Y. (2017). Investigating the period of switching roles in pair programming in a primary school. *Journal of Educational Technology & Society*, 20(3), 220-233.

저자소개

박 윤 미



2017 홍익대학교 세종캠퍼스 게임 소프트웨어학과 학사
 2018~현재 전북대학교 교육대학원 컴퓨터교육전공 석사
 관심분야: 소프트웨어 교육 및 창의적 프로그래밍
 e-mail: cahihyoom@gmail.com

이 효 종



1986, 1988, 1991 미국 유타대학교 컴퓨터과학과 학사, 석사, 박사 졸업
 1991~현재 전북대학교 컴퓨터공학부 교수
 관심분야: 영상처리, 패턴인식, 인공지능 및 정보교육
 e-mail: hlee@jbnu.ac.kr