

# 디자인 중심 모형이 SW 교육의 학습동기 및 학업성취도에 미치는 효과

양권우

공주교육대학교 컴퓨터교육과

## 요약

소프트웨어의 중요성이 확산됨에 따라 소프트웨어 교육 방법에 대한 연구가 활발히 진행되고 있다. 프로그래밍은 컴퓨터 과학에서 중요한 과목이지만, 프로그래밍 교육을 담당해야 할 예비 초등교사 대부분은 프로그래밍 원리에 대한 이해가 부족하다. 프로그래밍 학습에 도움을 줄 수 있는 새로운 교수법에 대한 연구가 필요하다. 그래서 본 연구에서는 디자인 중심 모형을 활용하여 소프트웨어 교육을 할 수 있는 교수·학습법을 제안하였다. 또한 본 논문에서 제안한 교수·학습법을 적용한 실험집단과 전통적인 강의법을 사용한 통제집단간의 프로그래밍 교육에 대한 학습동기와 학업성취도를 통계적으로 분석하였다. 프로그래밍 교육의 학습동기, 학업성취도를 비교한 결과 디자인 중심 모형을 활용한 소프트웨어 교육이 학습동기와 학업성취도 측면에서 통계적으로 유의미한 차이가 있음을 알 수 있었다.

키워드 : SW 교육, 코딩, 학습동기, 학업성취도, 디자인 중심 모형

## The Effect of Design Oriented Model on Learning Motivation and Academic Achievement in SW Education

GwonWoo Yang

Department of Computer Education, GongJu National University of Education

## Abstract

As the importance of software has been spread, the studies on ways of teaching software education have been actively progressed. A programming is important subject in computer science. However, most of undergraduates as prospective elementary teachers who will be in charge of teaching programming education have a lack of the understanding of programming principles. New pedagogical methods to help students to learn programming are needed. Therefore, in this paper, we investigated how programming education using design oriented model. And this study analyzed statistically the learning motivation and academic achievements of software education between a treatment group used the way suggested by this study and a control group used a traditional teaching way. As a result, the way suggested by this study was more meaningful in terms of the learning motivation and study achievements of software education than a traditional one.

Keywords : Software Education, Coding, Learning Motivation, Academic Achievement, Design Oriented Model

## 1. 서론

컴퓨터 사용자들은 프로그래밍 언어를 사용하여 직·간접적으로 컴퓨터와 소통할 뿐만 아니라 일상생활 속에 존재하고 있는 문제들과 지금까지 풀기 어려웠던 문제들을 해결하기도 한다. 또한 앞으로 다가올 미래에 발생할 일들을 예측하고, 해결하기도 한다. 프로그래밍 언어를 이용하여 이러한 문제들을 해결할 수 있는 프로그래밍 능력은 컴퓨터 과학뿐만 아니라 ICT(Information and Communication Technology) 산업 분야에서도 중요하게 생각하는 분야이다[1][2][3][12].

또한 우리는 사물인터넷(IoT), 빅 데이터와 인공지능 기술의 발달로 말미암아 급진적으로 변화하는 4차 산업혁명의 시대에 살고 있다. 이러한 기술들은 3D 프린터, 웹3.0(Web3.0) 등을 통해 디지털 제조업이란 새로운 업종을 만들어내었으며, 이러한 산업 분야의 변화는 급변하는 사회에 대응할 수 있는 문제해결력을 갖춘 창의·융합형 인재 양성을 필요로 하고 있다[5][6][8].

이에 2015년 교육부는 소프트웨어 중심사회 실현 정책을 지원하기 위해 소프트웨어 중심사회를 위한 인재양성 추진 계획을 발표하였다[17][18][19]. 문·이과 통합 교육으로 창의·융합 인재 양성을 목표로 하는 2015 교육과정에 소프트웨어 교육을 포함하여, 초등학교에서는 실과 교과에서 17 시간, 중학교에서는 정보 교과를 필수화하여 문제 해결과 알고리즘 및 프로그램 개발 등의 내용으로 34 시간, 고등학교에서는 정보 과목을 일반 선택으로 하여 다양한 분야와 융합한 알고리즘 설계 및 프로그램 개발 중심의 교육을 실시하기로 하였다[19].

해외에서도 소프트웨어 교육에 대한 관심이 많다. 영국에서는 정보통신 기술 교과를 컴퓨팅으로 하여 5세부터 프로그래밍 교육을 필수화하고 있으며, 에스토니아에서는 초등학교 1학년부터 프로그래밍 교육을 실시하고 있다. 또한 미국, 중국, 일본, 인도 등 국외 여러 나라에서는 국가 정책을 통해 프로그래밍 교육을 하고 있다 [8][9].

소프트웨어교육 연구 선도학교에서는 프로그래밍 교육에 프로젝트 학습을 주요 학습 방법으로 도입하여 적용하고 있으나 학생들이 프로젝트 학습의 문제 탐색 단계에서 우리 주변의 문제에 대해 공감하지 못하고 문제 발견의 필요성에서 어려움을 겪고 있다. 이러한 문제를

해결하기 위해 교사가 문제 또는 미션을 주고 해결하는 과정을 적용하는 사례가 있으나 문제 해결과 관련된 공감 없이 없는 상태에서는 동기를 부여받거나 지속적으로 학습에 참여하기가 어렵다[8].

그래서 본 논문에서는 해결하고자 하는 문제를 문제 자체가 아닌 프로토타입 형태로 제공한다. 학생들은 제공받은 프로토타입을 실행해 봄으로써 해결할 문제를 이해하고 공감한 다음 학생들의 생각을 제공된 프로토타입에 첨가하면서 프로젝트를 완성해가는 교수·학습법을 제안한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 디자인 사고에 관련된 이론적 배경, 디자인 사고 과정을 활용한 기존 연구들에 관한 내용을 기술한다. 제 3장에서는 연구대상, 연구방법 그리고 연구 설계를 기술한다. 제 4장에서는 본 논문에서 제안한 교수법이 학습동기와 학습성취도에 미친 영향을 통계적으로 분석한 결과를 기술하고 마지막으로 5장에서는 결론을 기술한다.

## 2. 이론적 배경

### 2.1 디자인 사고

디자인 사고는 디자이너가 디자인 과정에서 활용하는 창의적인 전략을 의미하며, 디자인 사고의 실행 과정은 다양한 모형이 있으나, 스탠포드 대학교 d-school에서 실행하였던 5단계 과정이 대표적이며 (Fig. 1)과 같다 [15][16].

공감(Empathize) 단계는 실제로 체험자를 관찰하고 체험하는 과정에서 사용자의 진정한 요구가 무엇인지를 파악하고 그것을 문제 해결과 연결시킬 필요가 있다. 공감의 기본 전제 조건은 체험자에 대한 배려와 선입견 및 편견의 불식에 있다. 내가 정한 가설을 검증하는 것이 아니라 체험자의 요구가 무엇인지를 확실하고 냉철하게 분석하는 자세가 필요하다[15][16].

문제정의(Define) 단계는 문제를 정의하는 단계로, 문제를 정의하는 방법은 수없이 많지만 일반적으로 브레인스토밍, 마인드매핑, 인터뷰, 포커스 그룹, 비주얼 리서치, 브랜드 매트릭스, 브랜드 북, 현장 조사와 같은 방법이 주로 사용된다[15][16].



(Fig. 1) Design thinking

아이디어도출(Ideate) 단계는 정확한 분석과 직관적 판단에 의해 정리된 문제를 해결할 방법을 강구하는 과정으로, 다양한 직종과 많은 경험을 쌓은 사람들의 지혜와 논의가 요구된다. 팀원들이 최대한 많은 아이디어를 도출한 뒤 의견을 교환한다. 양질 전환의 범칙에 입각해 초기에 엉뚱하더라도 많은 양의 생각을 모으고 차차 질 높은 아이디어로 다듬어간다[15][16].

프로토타입과 테스트(Prototype & test) 단계는 프로토타입을 만든 다음 테스트와 개선 과정을 반복하는 마지막 과정으로 통해 완벽한 상태의 제품을 출시하는 것이 디자인 사고의 최종 목표이다[15][16].

## 2.2 디자인 중심 모형

SW 교육을 위한 교수 학습 모형 개발 연구에서 스탠 퍼드 대학교의 d-school에서 제시한 디자인 사고 과정을 따르는 디자인 중심 모형을 제안하였다. 디자인 중심 모형의 단계별 학습 방법은 (Fig. 2)와 같다[13].

요구분석 단계는 인간 중심의 관찰과 사용자 요구분석에 주안점을 둔다. 인간중심 요구분석은 인간에게 이로우며 주고 발전적인 방향으로 분석하는 것을 목표로 한다.

설계 단계는 단순한 계획서가 아니라 사고의 확장과 창의적인 아이디어 산출에 초점을 맞춘다. 문제의 분해와 패턴 등을 표, 다이어그램, 이미지, 마인드 맵, 그래프 등으로 시각화할 수 있다.

구현 단계는 제작, 재구성, 개발을 포함한다. 프로그래밍과 피지컬 컴퓨팅을 통해 구현함으로써 학습자의 컴퓨팅 사고력(Computational thinking)를 종합적으로

신장시키도록 구성한다.

공유 단계는 단순히 산출된 작품의 소개를 넘어 제작 의도와 설계 및 구현 과정에 대한 전 과정을 공유한다. 공유를 통해 요구분석과 설계에 대한 근본적인 평가를 하게 된다.



(Fig. 2) Design oriented model

## 2.3 관련 연구

서영호 및 김종훈(2017)은 디자인 사고를 적용한 소프트웨어 교육이 초등학교 예비교사의 창의성에 미치는 효과를 분석한 연구로, 연구 결과 디자인 사고를 적용한 소프트웨어 교육이 초등학교 예비교사들의 창의성 향상에 효과적임을 보였다[15].

전수진(2017)은 소프트웨어 교육에서 디자인 중심 모형을 학습자들에게 적용한 다음 컴퓨팅 사고력에 미치는 효과를 분석하였다. 실험집단 학생들이 컴퓨팅을 이용한 실생활 문제 해결에 자신감과 컴퓨팅 사고력 향상에 대해 긍정적인 인식 변화를 보였다[13].

차현진 및 김민하(2020)은 학생들이 수행하는 창의적 문제 해결 프로젝트에서 저수준 프로토타이핑 과제 수행과 고수준 프로토타이핑 과제 수행의 2개 그룹으로 나누어 학업적 자기효능감과 문제해결력에 미치는 영향을 분석하였다. 그 결과 저수준의 프로토타이핑을 수행한 그룹에서 더 높은 역량 변화를 가져온 것으로 나타났다[7].

홍전순 및 장환영(2020)은 디자인 사고를 초등학교 3학년 정규교육과정에 도입하여, 학급문제해결이라는 주제로 디자인 사고 기반 학습프로그램을 개발하고 초등학생 대상으로 적용하는 실험연구로 공감, 문제정의, 아이디어 발상, 프로토타이핑 & 테스트로 구성된 디자인 사고 과정을 제안하였다[10].

김정량(2018)은 디자인 사고 모형을 분석하여 초등학교 프로그래밍 교육의 요소와 디자인 사고의 공감 및 발산·수렴 사고 과정을 융합한 초등 프로그래밍 교육을 위한 디자인 사고 모형을 탐색·수집단계, 요구분석 단계, 알고리즘 단계, 프로그래밍 단계, 발전 단계로 개발하였다[8].

신윤희 및 정효정(2019)은 대학교 컴퓨터 비전공자 학생 대상으로 디자인 썬킹을 접목한 코딩교육을 실시하였으며, 성찰일지 분석을 통하여 학습자들이 무엇을 배웠고, 어떠한 어려움을 경험하였는지를 분석하였다 [14].

서해인 및 김효정(2020)은 디자인 썬킹을 적용한 미술수업이 청소년들의 창의적 문제해결력에 긍정적인 효과를 미쳤으며 자신감, 협동심, 다양한 관점의 표출 등 정서적 영역을 함양하는 것에 있어서도 교육적 가치가 있음을 보였다[4].

기존의 연구들은 디자인 사고를 활용하여 창의성 향상, 컴퓨팅 사고력 향상, 문제해결력 향상 또는 교수학습 모형 개발에 초점을 두었지만, 본 연구는 예비 초등교사를 대상으로 소프트웨어 교육에 대한 학습동기 및 학업성취도에 초점을 두고자 한다.

### 3. 연구 방법

#### 3.1 연구 대상 및 측정 도구

본 연구는 교육대학교 2학년 2개 심화과정 학생 57명을 대상으로 하였으며, 프로그래밍과 알고리즘 수업에서 소프트웨어 교육 내용을 적용하였다. 2개의 심화과정 학생들 중 A교육학과 29명의 학생들을 실험집단으로 배정하여 본 연구에서 제안한 교수·학습법으로 수업하였으며, B교육학과 28명의 학생들을 통제집단으로 배정하여 강의 후 실습하는 전통적인 교수·학습법으로 수업을 진행하였다. 본 연구에 참여한 학생들은 1학년 때 컴퓨터의 이해 및 실습수업에서 컴퓨터 이론에 대한 기초 지식과 스크래치의 블록들을 개괄적으로 학습하였지만, 코딩 경험은 거의 없는 상태이며 동일한 교수가 수업을 진행하였다.

강의를 들은 다음 실습하는 전통적인 소프트웨어 교육과 본 논문에서 제안한 교수·학습법을 이용한 소프트

웨어 교육이 학습동기에 어떤 영향을 미치는지 알아보기 위하여 양권우(2020)의 학습동기 검사지와 조오근(2004)의 학습동기 검사지를 참고하여 본 연구 목적에 맞게 수정·보완하여 학습동기 측정 도구를 만들었으며 [3][11], 각 문항 당 리커트(Likert) 5점 척도로 20 문항을 제시하여 100점 만점으로 평가하였다. 검사 도구 제작 경험이 많은 2명의 교수를 통해 검증받은 문항에 대한 신뢰도 검사 결과는 Cronbach  $\alpha$ 계수의 값이 사전검사 .85, 사후검사 .88로 비교적 높은 수준을 나타내고 있다. 따라서 본 연구에서 제작한 학습동기 측정 도구는 측정 오차가 낮으며 문항 내적 일관성이 높은 것으로 확인되었다.

학업성취도에 대한 사전 측정값은 실험 대상 학생들이 1학년 때 실시한 컴퓨터 과학에 관련된 기초 지식과 스크래치의 구문에 대해 기술하는 서술식 기말고사 성적(60점 만점)을 사용하였다.

본 논문에서 제안한 교수·학습법을 이용한 소프트웨어 수업이 학업성취도에 미친 영향을 측정하기 위해 사후 학업성취도 평가문항을 만들었으며, 논리 오류를 찾고 수정하는 5문항, 프로그램의 빈 부분을 완성하는 5문항, 실행 결과를 기술하는 5문항, 총 15문항이며 문항당 2점 총 30점으로 평가하였다.

#### 3.2 연구 설계

교육대학교 2학년 학생들 중 A교육학과 학생들을 통제집단, B교육학과 학생들을 실험집단으로 배정한 다음 수업을 진행하기에 앞서 학습동기, 학업성취도에 대한 사전검사를 실시하였다. 본 연구에서 제작한 측정 도구를 사용하여 학습동기에 대한 사전검사를 실시하였고, 학업성취도에 대한 사전 측정값은 실험 대상 학생들이 1학년 때 실시한 컴퓨터 과학에 관련된 기초 지식과 스크래치의 구문에 대해 기술하는 서술식 기말고사 성적을 사용하였다.

학습동기와 학업성취도에 대한 사전 검사를 진행한 다음, 본 연구에서 제안한 교수법으로 12주 12차시 동안 소프트웨어 교육 수업을 진행하였다. 제안한 교수법이 소프트웨어 교육의 학습동기와 학업성취도에 어떠한 영향을 미치는지 알아보기 위하여 학습동기와 학업성취도에 대한 사후 검사를 진행하였다.

### 3.3 연구 방법

교육대학교에서 컴퓨터 비전공자 학생들을 대상으로 프로그래밍 수업을 하면서 겪는 어려움들 중 하나는 학생들이 소프트웨어를 개발할 때 문제 탐색 단계에서 우리 주변에 존재하는 문제에 대해 공감하지 못하고 문제 발견에서 많은 어려움을 겪고 있다는 것이다.

그래서 본 연구에서는 KERIS에서 개발한 소프트웨어 교육 교수·학습 모형인 디자인 중심 모형을 변형한 소프트웨어 교육 교수·학습법을 제안한다. 본 연구에서 제안한 교수·학습법은 문제를 학생들에게 제공할 때, 해결하고자 하는 문제 자체를 제공하는 것이 아니라 스크래치를 이용하여 작성한 프로토타입 형태로 제공한다. 제공된 프로토타입은 코드가 있는 스프라이트(무대)와 코드가 없는 스프라이트(무대)로 구성된다. 이후 본 논문에서 이용한 디자인 중심 모형의 각 단계에서 학생들이 수행하게 될 활동들은 다음과 같다.

요구분석 단계 : 학생들은 제공된 프로토타입을 여러 번 실행해봄으로써 해결해야 할 문제가 무엇인지를 인식하고, 공감한다. 프로토타입의 코드를 분석할 때 스크래치의 주석 기능을 사용하여 주석을 작성하면서 이해하고, 이해하기 어려운 부분은 교수자의 도움을 받는다.

설계 단계 : 요구분석 단계에서 이해한 프로토타입과 코드가 없는 스프라이트(무대)들을 관찰하면서 다양하고 자기 자신만의 창의적인 아이디어를 산출하고, 산출한 아이디어를 해결할 수 있는 방법을 설계하고 계획한다.

구현 단계 : 설계 단계에서 자기 자신이 독창적으로 설계하고 계획한 문제 해결 방법을 스크래치를 사용하여 코딩한다. 코딩하기 어려운 부분은 교수자의 도움을 받는다.

공유 단계 : 자신이 완성한 프로젝트를 친구들과 공유하면서 차이점, 공통점, 아쉬운 점등에 대해 이야기하면서 더 나은 문제 해결 방법에 대해 배운다.

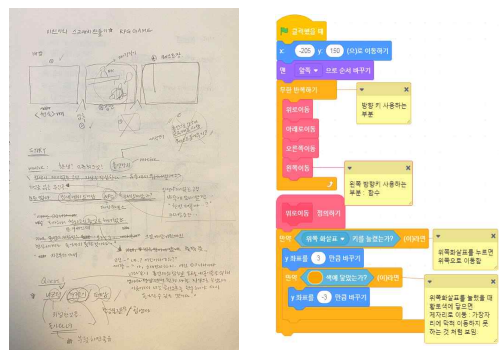
(Fig. 3)은 미로를 탈출하라란 주제로 수업을 진행한 수업 예를 나타내고 있다. 미로를 탈출하라 프로젝트는 10개의 스프라이트와 무대로 구성되어 있으며, 세 개의 스프라이트에만 코드를 제공하였고, 나머지 스프라이트와 무대에는 코드를 제공하지 않았다. 미로를 탈출하라 프로젝트 내용은 다음과 같다. 톱니바퀴 모양의 회전문

은 같은 위치에서 계속 회전하고 있고, 꼬리달린 도마뱀은 좌우로 계속해서 왔다 갔다 하고 있으며, 주인공은 무대의 좌측 위쪽 모서리 부근에 위치하고 있다. 사용자가 키보드의 방향키를 사용하여 주인공을 상·하·좌·우로 이동하다가 톱니바퀴 모양의 회전문이나 꼬리달린 도마뱀에 닿으면 주인공이 시작 위치에서 도전을 계속할 수 있는 프로토타입이다.



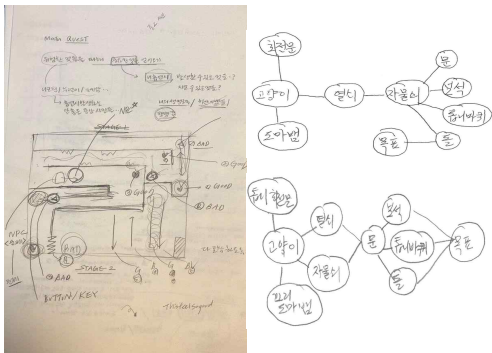
(Fig. 3) Class example(Maze)

요구분석 단계에서 학생들은 직접 프로토타입을 실행해 보면서 주인공이 여러 난관을 극복하면서 목표지점에 도착하는 문제임을 이해하고 공감한다. 또한 학생들은 코드가 제공되지 않은 스프라이트를 코딩하기 위해 프로토타입에서 제공된 코드들을 분석, 이해하게 되는데, 코드를 보다 더 잘 분석하게하기 위하여 주석 기능을 사용하여 주석을 달도록 하였다(Fig. 4).



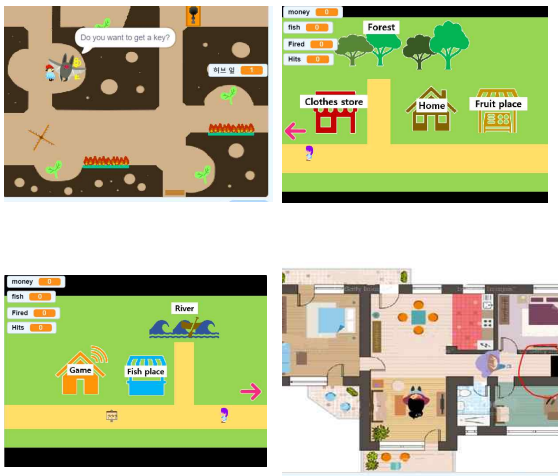
(Fig. 4) Requirement analysis

설계 단계에서는 요구분석 단계에서 이해하고 공감한 문제와 코드가 제공되지 않은 스프라이트들을 관찰하면서 자기 자신만의 새롭고, 독창적인 아이디어를 산출해 낸다. (Fig. 5)는 설계 단계에서 학생들이 자기만의 창의적인 아이디어(스토리)를 생각해낸 다음, 생각한 아이디어를 설계하고 계획한 예이다.



(Fig. 5) Design

구현단계에서는 설계 단계에서 설계한 내용을 스크래치를 사용하여 구현하게 된다. (Fig. 6)은 미로를 탈출하라는 주제로 진행한 수업에서 학생들이 분석하고 설계한 내용을 구현한 예들이다.



(Fig. 6) Implementation

#### 4. 연구 결과

##### 4.1 학습동기 결과

연구가설1 : 실험집단과 통제집단 간에는 학습동기 측면에서 통계적으로 유의한 차이가 있을 것이다.

실험집단과 통제집단의 사전 학습동기 측정값에 대한 독립표본 t-검증 결과는 <Table 1>과 같다.

<Table 1> Learning motivation(Pretest)

Independent samples test					
Group	N	Mean	Std.	t	Sig. (2-tailed)
Control	28	62.0	10.0	.811	.421
Treatment	29	59.8	10.1		

$t = .811(p < .05)$

집단통계량에서 통제집단의 평균 62.0 표준편차 10.0, 실험집단의 평균 59.8 표준편차 10.1로 통제집단의 평균이 실험집단의 평균보다 좀 더 높았지만, 독립표본검정 결과 두 집단은 유의수준 .05에서 p 값이 .421로 통계적으로 유의한 차이를 보이지 않았다. 즉, 실험집단과 통제집단은 통계적으로 동질집단임을 알 수 있다.

12주 12차시 동안 수업을 진행한 실험집단과 통제집단을 대상으로 학습동기에 대한 사후 검사 결과를 측정하였다. 사후 학습동기 측정값에 대한 독립표본 t-검증 결과는 <Table 2>와 같다.

<Table 2> Learning motivation(Posttest)

Independent samples test					
Group	N	Mean	Std.	t	Sig. (2-tailed)
Control	28	64.2	8.0	-3.433	.001
Treatment	29	70.6	5.9		

$t = -3.433(p < .05)$

집단통계량에서 통제집단의 평균 64.2 표준편차 8.0, 실험집단의 평균 70.6 표준편차 5.9로 실험집단의 평균이 높게 나왔으며 독립표본 t-검증 결과 두 집단은 유의수준 .05에서 p 값이 .001로 통계적으로 유의한 차

이가 있는 것으로 나타났다. 이 결과는 본 논문에서 제안한 교수·학습법을 사용한 수업이 전통적인 강의 방식의 소프트웨어 수업보다 더 높은 학습동기를 유발함을 보여 준다.

4.2 학업성취도 결과

연구가설2: 실험집단과 통제집단 간에는 학업성취도에서 통계적으로 유의한 차이가 있을 것이다.

실험을 진행하기 전에 실시한 통제집단과 실험집단에 속하는 학생들의 사전 학업성취도 측정 점수에 대한 독립표본 t-검증 결과는 <Table 3>과 같다.

<Table 3> Academic achievement(Pretest)

Independent samples test					
Group	N	Mean	Std.	t	Sig. (2-tailed)
Control	28	43.9	8.8	1.074	.287
Treatment	29	41.5	8.1		

t=1.074(p<.05)

집단통계량에서 통제집단의 평균 43.9 표준편차 8.8, 실험집단의 평균 41.5 표준편차 8.1로 통제집단의 평균이 높았다. 그러나 독립표본검정 결과 두 집단은 유의수준 .05에서 p 값이 .287로 유의한 차이를 보이지 않았다. 즉, 통계적으로 실험집단과 통제집단은 동질집단임을 알 수 있다.

12주 동안 수업을 진행한 통제집단과 실험집단을 대상으로 학업성취도에 대한 사후 결과를 측정하였다. 사후 학업성취도 측정값에 대한 독립표본 t-검증 결과는 <Table 4>과 같다.

<Table 4> Academic achievement(Posttest)

Independent samples test					
Group	N	Mean	Std.	t	Sig. (2-tailed)
Control	28	21.6	3.2	-4.924	.000
Treatment	29	25.3	2.3		

t=-4.924(p<.05)

집단통계량에서 통제집단의 평균 21.6 표준편차 3.2,

실험집단의 평균 25.3 표준편차 2.3으로 실험집단의 평균이 높게 나왔으며 독립표본 검증 결과 두 집단은 유의수준 .05에서 p 값이 .000으로 두 집단 간의 유의한 차이가 있는 것으로 나타났다. 이 결과는 본 연구에서 제안한 교수·학습법을 이용한 소프트웨어 교육이 전통적인 강의 방식의 소프트웨어 교육보다 더 높은 학업성취도를 얻을 수 있음을 보여준다.

5. 결론

본 논문의 연구 내용은 디자인 중심 모형을 변형한 소프트웨어 교육 교수·학습법을 활용하여 프로그래밍 교육을 실시하는 것이며, 연구 목적은 컴퓨터 비전공자들인 예비 초등교사로 구성된 두 집단에 프로그래밍과 알고리즘 수업을 진행한 다음 학습동기와 학업성취도에 미치는 영향을 알아보는 것이다. 본 논문에서 제안한 교수·학습법이 예비 초등 교사들의 소프트웨어 교육에 대한 학습동기와 학업 성취도에 미친 영향은 다음과 같다.

사전 학습동기 검사를 통해 통계적으로 동질집단으로 판단된 실험집단과 통제집단간의 사후 학습동기 측정값을 대상으로 독립표본 t-검증을 실시한 결과, 두 집단은 유의수준 .05에서 p 값이 .001로 통계적으로 유의한 차이가 있는 것으로 나타났다. 이는 프로토타입을 실행하면서 해결할 문제에 대해 생긴 호기심, 관심, 자기 자신만의 창의적 방법으로 문제를 해결한 자신감 등이 소프트웨어 교육을 위한 학습동기에 긍정적인 영향을 미쳤을 것이라 여겨진다.

사전 학업성취도 검사를 통해 얻은 측정값을 통계 처리한 결과 동질집단으로 판단된 실험집단과 통제집단간의 사후 학업성취도 측정값을 대상으로 독립표본 t-검증을 실시한 결과, 실험집단과 통제집단은 유의수준 .05에서 p 값이 .000로 통계적으로 유의한 차이가 있는 것으로 나타났다. 이는 프로토타입을 이해하고, 코드를 분석하고, 이해하고, 응용해 보는 것이 학업성취도에 긍정적인 영향을 미쳤을 것이라 여겨진다.

본 연구는 코딩 경험이 없는 컴퓨터 비전공자 대상으로 수업 시간에 진행한 실험인 관계로 전체적인 시각으로 문제를 해결하기 보다는 단계적으로 문제를 해결하는 경우가 많았다. 추후 전체적인 시각으로 문제를 바라볼 수 있는 교수·학습법에 대한 연구가 필요하다. 또한

1차시 수업인 관계로 시간적인 제한점 때문에 설계 단계와 공유 단계에서 충분한 시간을 제공하지 못한 점을 해결한다면 더 나은 결과가 나왔을 것이라 사료된다.

### 참고문헌

- [1] EunJung Kwon, EunKyoung Lee, YoungJun Lee(2009). The Effect of Algorithm Learning by Playing on Learning Motivation and Achievement. *The Journal of Korean association of computer education, 12*(6), 33-39.
- [2] GwonWoo Yang(2018). Unplugged Role-Play on Learning Motivation and Academic Achievement Focusing How Computers Work. *Journal of Knowledge Information Technology and Systems, 13*(2), 221-229.
- [3] GwonWoo Yang(2020). The Effect of Software Education Using Pair Programming on Learning Motivation and Academic Achievement. *Journal of Knowledge Information Technology and Systems, 15*(1), 57-65.
- [4] HaeIn Suh, HyoJung Kim(2020). The Effects of Design-Thinking Art Classes on Teenagers' Creative Problem-Solving Ability: Focusing on the first and second graders of the high school. *Journal of Art Education, 60*(1), 107-144
- [5] HyeKyung Cho, KangBak Park, JeongHye Han, DugKi Min, KukWon Ko(2008). Education + Robots: the Vision and the Action Plans. *Communications of the Korean Institute of Information Scientists and Engineer, 26*(4), 56-57.
- [6] HyungJin Ahn, DaiSung Ma(2013). Development of Primary School Scratch Curriculum for Improving the Ability to Solve Problems, *Journal of The Korean Association of information Education, 17*(3), 317-327.
- [7] HyunJin Cha, Minha Kim(2020). Effectiveness Analysis of Prototyping-Fidelity Levels on Creative Problem-Solving Projects in Software Education : Focused on perceived academic self-efficacy and problem-solving skills. *Journal of Learner-Centered Curriculum and Instruction, 20*(3), 689-711.
- [8] Jeong Rang Kim(2018). Development of Design Thinking Model for Elementary Programming Education. *KAIE Research Journal, 9*(1), 49-54.
- [9] JungSook Sung, HyeonCheol Kim(2015). Analysis on the International Comparison of Computer Education in Schools. *The Journal of Korean association of computer education, 18*(1), 45-54.
- [10] JeongSoon Hong, Hwan Young Jang(2020). Action Research on Development and Application of Learning Program based on Design Thinking for Elementary School Students Focusing on the Design Thinking Process. *Korean Education Inquiry, 38*(1), 1-31.
- [11] OGun Jo(2004). *Development of physics inquiry learning materials and application in the play-based inquiry context*. ph.d.s' thesis, Busan national university of education.
- [12] SooHwan Kim, WonGyu Lee, HyeonCheol Kim (2009). Applications of Educational Programming Languages in K-12 Information Curriculum. *The Journal of Korean association of computer education, 12*(2), 23-31.
- [13] SooJin Jun(2017). The Effect of Design-Oriented Model(NDIS) based on Computational Thinking in SW Education. *Journal of Computer Education, 20*(2), 13-21.
- [14] YoonHee Shin, Hyojung Jung, JongSuk Song(2019). Analysis of Learning Experience in Design Thinking-Based Coding Education for SW Non-major College Students. *Journal of Digital Contents Society, 20*(4), 759-768.
- [15] YoungHo Seo, Jonghoon Kim(2017). The effect of SW education applying Design Thinking on creativity of elementary school pre-service teachers. *Journal of The Korean Association of Information Education, 21*(3), 351-360.
- [16] Roger, M(2009). *Design Thinking*. Seoul,



WoongJin Wings.

- [17] Ministry of Education(2015). Human Resource Development Plan for the SW-oriented society. <http://www.msip.go.kr/>
- [18] Ministry of Education(2016). Direction and strategy of medium and long-term education policy in response to intelligence information society. <http://www.msip.go.kr/>
- [19] National Curriculum Information Center(2019), Revised National Curriculum, <http://ncic.re.kr>
- [20] Scratch(2020). Scratch. <https://scratch.mit.edu>.

### 저자소개

#### 양 권 우



2000.2 고려대학교 일반대학원 컴퓨터학과 컴퓨터학 전공(이학박사)

2000년 9월 ~ 현재 공주교육대학교 컴퓨터교육과 교수

관심분야 : 로봇 교육, 소프트웨어 교육, 교육용 프로그래밍 언어, 알고리즘, 피지컬 컴퓨팅, 언플러그드 프로그래밍 교육, 언플러그드

e-mail:kwyang153@hanmail.net