

예비교사를 위한 게임 프로그래밍 교육모델 4E 개발

성영훈

진주교육대학교 컴퓨터교육과

요약

프로그래밍 교육은 일반적으로 문제분석 과정, 알고리즘과 프로그래밍을 통한 자동화, 일반화 과정을 포함하고 있어 학습자의 컴퓨팅사고 향상에 좋은 소프트웨어 교육방법이다. 그러나 초보자의 경우 명령어 사용법에 대한 이해, 알고리즘 작성과 프로그래밍 구현단계에서 어려움을 겪는 것으로 나타났다. 본 연구에서는 예비교사들의 프로그래밍 교육을 위해 게임 프로그래밍 교육모델과 교육과정을 개발하였다. 4E 모델은 공감단계, 탐색단계, 강화단계 및 평가단계로 구성되었다. 또한 각 단계별로 게임 핵심 요소와 핵심 명령어 블록들을 학습할 수 있도록 구성하였다. 예비교사가 프로그래밍 명령어 활용에 대한 이해를 돕기 위해 예제 학습, 자기 게임 생성 및 팀 기반 프로젝트 형태로 구성된 3단계 교수학습 방법을 제시하였다. 15주간 교육과정으로 적용하고 검증한 결과 설계한 모델, 예비교사들의 블록 프로그래밍 역량 인식 등에서 유의미한 결과를 보였으며 제출한 결과물에 대한 컴퓨팅사고 수준도 높은 결과를 보였다.

키워드 : SW교육, 프로그래밍 교육, 게임 기반 학습, 교수 방법

Development of Game Programming Education Model 4E for Pre-Service Teachers

Younghoon Sung

Chinju National University of Education

ABSTRACT

Programming education generally includes problem analysis process, automation through algorithms and programming, and generalization process. It is a good software education method for students in improving computing thinking. However, it was found that beginners had difficulties in understanding instruction usage, writing algorithms, and implementing programming. In this study, we developed a game programming education model and curriculum for programming education of pre-service teachers. The 4E model consisted of empathy, exploration, engagement and evaluation. In addition, it is configured to learn game core elements and core command blocks by each stage. To help the pre-service teachers understand the use of various programming blocks, a three-step teaching and learning method was presented, consisting of example learning, self-game creation, and team-based projects. As a result of applying and verifying the curriculum for 15 weeks, it showed significant results in the 4E model and pre-service teachers' perception of block programming competence and the level of computational thinking on the submitted game project results was also high.

Keywords : SW Education, Programming Education, Game Programming Education Model, Teaching Method

교신저자: 성영훈(yhsung@cue.ac.kr)

논문투고 : 2019-11-17

논문심사 : 2019-11-26

심사완료 : 2019-11-27

1. 서론

최근 2015 개정교육과정에 소프트웨어교육이 필수적으로 도입됨에 따라 초등교원양성대학에서도 예비교사들을 위해 다양한 소프트웨어 교육과정이 개발되고 적용되고 있다. 이에 초등학생들의 소프트웨어 교육을 담당하는 현장 교사뿐만 아니라 예비교사들의 소프트웨어 교육 역량도 향상 시킬 필요가 있다[10].

소프트웨어교육을 처음 접하는 예비교사들의 컴퓨팅 사고력을 향상시키기 위해서 언플러그드 컴퓨팅, 블록 프로그래밍 교육, 피지컬 컴퓨팅 등 다양한 교육형태를 적용하고 이를 평가할 수 있는 방법 등이 연구되고 있다. 특히 프로그래밍 교육은 일반적으로 문제해결을 위한 문제분석 과정, 알고리즘과 프로그래밍, 테스트 및 구현단계의 과정을 거치게 되는데 이 때 컴퓨팅 사고는 문제해결 전반에 걸쳐 주요한 사고 요소로 작용하게 된다. 그러나 전통적인 방식의 프로그래밍 교육의 경우 텍스트 형태의 명령어 이해, 알고리즘 작성과 프로그래밍 구현 영역 등에서 학습의 수준이 올라갈수록 학습자의 흥미와 지속 등에서 어려움을 겪고 있다[11][13][30].

이러한 면에서 블록 프로그래밍 언어는 실행 가능한 명령문들이 블록 형태로 되어 있고 그래픽 기반의 UI를 통해 사용자가 쉽게 결합하여 프로그래밍을 할 수 있는 언어로 프로그래밍 경험이 부족한 초보자들에게 많이 활용되고 있다.

초보자가 쉽게 블록 프로그래밍 교육과정을 접할 수 있도록 기초블록 사용방법, 변수와 함수, 프로젝트로 구성된 전통적 형태의 교육모델[29], 프로젝트 형태 [17][23], 디자인 사고 적용[27] 블랜디드 러닝 기법 활용[24], 게이미피케이션 전략[6] 등 다양한 형태의 연구 방법을 적용한 것으로 나타났다. 이를 통해 예비교사들은 프로그래밍 경험이 많을수록 결과물의 질적 수준이 높아진 것으로 나타났다. 그러나 프로그래밍 내부 절차와 관련하여 복잡한 조건과 논리연산 등에 대한 내용을 다루는 것에는 부족한 면을 보였다[23].

프로그래밍 교육을 통한 컴퓨팅사고 수준을 측정하는데 있어서 문제해결 과정 속 추상화 단계, 정리된 알고리즘을 기반으로 프로그래밍으로 구현하는 과정에서 자기 평가 방법과 Dr.Scratch 등을 활용한 다양한 컴퓨팅 사고 측정과 피드백은 학습자에 긍정적인 영향을 주는

것으로 나타났다[17][23].

반면, 프로그래밍 교육을 처음 접하는 예비교사 및 비전공자들의 경우 문법에 대한 이해, 문제해결에 필요한 명령어 사용, 작성한 알고리즘을 프로그래밍으로 전이하는 단계에서 논리적인 오류들이 많이 나타나며 어려움을 겪는 것으로 나타났다[11][25]. 결국 초보자들이 프로그래밍 교육에서 겪는 어려움은 학습의 전반적인 질적 수준을 떨어뜨릴 뿐만 아니라 학습 동기의 지속적 유발과 흥미 유지와 참여 등에 부정적인 영향을 미칠 수 있다.

이러한 면에서 게임 프로그래밍 교육은 학습자의 흥미 지속과 적극적인 참여를 유도할 수 있는 방법으로 논리력, 창의성 등 고등사고능력을 향상시키는 데 효과적인 것으로 알려져 있다[12][19]. 또한 게임 규칙 설계, 게임 알고리즘 제작, 게임 프로그래밍 구현 요소들은 학습자 중심의 학습지속과 동기유발이 가능하여 효과적인 프로그래밍 교육방법으로 제시되고 있다[30].

따라서 본 연구에서는 초등 예비교사들을 위해 체계적인 게임 프로그래밍 교육이 가능한 교육 모델을 연구하고자 한다.

2. 관련연구

2.1 컴퓨팅사고와 프로그래밍 교육

컴퓨팅사고(Computational Thinking)는 우리가 살아가는 삶 속에서 발생하는 다양한 문제들을 컴퓨터과학의 원리를 적용하여 해결해 나가는데 필요한 사고를 의미한다[26]. 이러한 컴퓨팅사고는 기술의 발전에 따라 빅데이터, 융합기술과 초연결 등의 형태로 진화하는 소프트웨어 중심의 4차 산업혁명에서 더욱 필요한 역량으로 인식되고 있다.

컴퓨팅사고가 가지고 있는 요소들은 일반적으로 문제분해, 문제분석, 패턴찾기, 알고리즘 작성, 프로그래밍을 통한 자동화 및 결과물 또는 제작물에 대한 다양한 적용이 가능한 일반화 요소들을 포함하고 있다[28][31].

소프트웨어교육 방법 중에서 프로그래밍 교육은 제작하고자 하는 프로그램 결과물을 만드는 과정에서 자연스럽게 컴퓨팅사고 요소들을 학습할 수 있는 장점이 있다. 그러나 초보자의 경우 프로그램 언어적 특성과 학습

자의 수준에 따라 학습자의 흥미와 학습 지속도에 있어 영향을 받는 경우가 있다.

특히 블록 프로그래밍과 관련하여 초보 학습자의 경우 프로그래밍에 필요한 블록 제어에 대한 어려움, 문제 해결 절차 구현을 위한 다양한 블록 구조 만들기, 제어와 관련된 방법 및 논리적 오류에 대한 개선 등에 어려움을 느끼고 있는 것으로 나타났다[3]. 결국 이러한 요인들은 기초적인 내용을 학습할 때보다 다양한 블록 명령어들이 복합적으로 연계되어 구현하는 수준으로 심화될수록 알고리즘 설계와 프로그램 구현에 어려움을 느끼는 것으로 나타났다[25]. 따라서 학습자의 흥미를 기반으로 학습지속과 프로그래밍 개념에 대한 체계적인 이해가 가능할 수 있는 교육방법 적용이 필요하다.

2.2 게임 구성 요소

게임 프로그래밍 교육은 학습자의 참여유도와 흥미 유발과 관련하여 디지털 매체와 놀이에 익숙한 학습자들에게 적합한 방법으로 주목받고 있다[19].

게임에 대한 용어적 정의를 살펴보면 게임은 정해진 규칙에 의해서 플레이어가 인공적으로 만들어지는 다양한 충돌을 경험하는 시스템으로 결과들을 양적으로 측정할 수 있는 것이라고 정의하고 있다[21].

일반적으로 게임을 구성하는 요소를 형식적 요소와 몰입 환경적 요소로 구분하여 살펴보면 다음과 같다 [16][7].

우선 형식적인 요소를 게임 시작, 진행, 결과적 측면에서 살펴보면 게임 시작 전에는 게임 속에서 달성해야 하는 과제인 목표, 게임 속 참여자 또는 접속자인 플레이어, 게임의 진행 과정을 나타내는 절차, 게임에서 다양한 상호작용 관계를 규정하는 규칙 요소가 필요하다. 게임 진행 중에서는 게임 속 통용되는 자원, 게임의 규칙과 과제 달성 과정을 방해할 수 있는 충돌상황, 게임과 게임이 아닌 것 사이의 경계 요소가 작용한다. 마지막으로 게임 결과에서는 예측 가능한 결과이나 불확실성을 보장하는 결과 요소가 필요하다.

다음으로 몰입 환경 요소에 대해 살펴보면 일반적인 게임들은 캐릭터, 사운드, 배경으로 구성되는 스토리가 다양한 도전과제 등과 융합되어 있다. 특히 게임에 참여하는 플레이어는 정해진 규칙내에서 자원과 다양한 오브

젝트들과의 충돌 상황, 게임 속에서 정의된 경계와의 상호작용을 통해 도전과제를 완수함으로써 더욱 몰입할 수 있는 장치를 제공한다.

이를 기반으로 <Table 1>에서와 같이 게임 개발과정에 따른 구성요소들과 컴퓨팅 사고 요소를 연계하여 살펴보면 다음과 같다.

<Table 1> Game Construct Elements with Computational Thinking[7][16]

Game development process	Game structure elements	Computational thinking elements
Generate ideas	Game goal	Problem understanding
Formalize ideas	Player, resources, Rules, operation	Decomposition Algorithms
Test ideas	Data processing, conflict, boundaries and prototype test	Automation
Evaluate ideas	Evaluate a game project	Generalization

게임 프로그래밍의 경우 학습자의 흥미 있는 주제에 대한 목표 설정, 게임 속 플레이어, 자원 설계, 규칙과 절차 설계 과정에서 필요한 문제분해, 알고리즘 제작 과정을 학습하게 된다. 또한 게임의 충돌 요소와 관련하여 플레이어의 자원, 경계 설정과 상호작용 처리 등을 통하여 프로그래밍의 다양한 구조적 접근과 이벤트 처리과정을 경험할 수 있다.

특히 게임을 제작하는 것뿐만 아니라 게임을 하는 과정에서 학습자의 주의집중을 강화할 수 있는 다양한 몰입요소를 통해 학습자의 인지전략과 플레이어와의 상호작용을 통한 자기 효능감에도 긍정적인 영향을 주는 것으로 나타났다[30].

2.3 선행연구 분석

게임 프로그래밍을 학습하는 것은 교육적 측면에서 컴퓨팅 사고 이해, 학습자 참여와 협업을 강화할 수 있는 좋은 접근방법 중의 하나이며 이와 관련된 국내외 선행연구들을 살펴보면 다음과 같다[2][8].

첫째, 국내외 선행연구들을 살펴보면 정영식 외(2014)의 연구에서는 게임 프로그래밍 학습단계를 안내, 개발,

게임, 수정하기 단계로 구분하여 게임에 대한 활용실태와 태도에 대해 긍정적인 효과가 있는 것으로 나타났다[30].

박판우(2016)의 경우 초등학생을 대상으로 스크래치와 피코보드를 활용한 융합 SW교육방안을 개발하여 학습자의 융합적 지식, 논리적인 프로그래밍 학습방법을 제시하였다. 특히 게임 프로그래밍에서 단계별, 점진적인 문제 확대 방식을 활용하여 학습자의 수준에 맞는 학습이 이루어질 수 있도록 하였다[20].

김성경외(2018)의 연구에서는 중학생을 대상으로 앱인벤터를 활용한 게임 프로그래밍 교육을 적용하였는데 학습자의 인지적 발달수준을 고려하여 토론식 하브루타 학습방법을 적용한 것이 특징이다. 학습과정에서도 기초, 알고리즘, 변수 오퍼레이터, 함수단계로 점진적으로 수준을 높였다[22].

둘째, 국외 선행연구들을 살펴보면 다음과 같다.

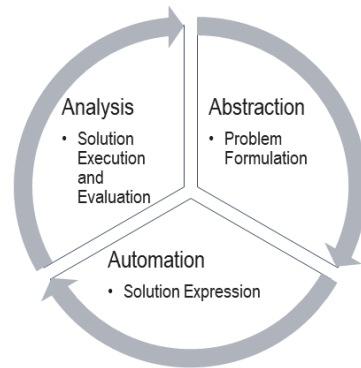
Lau, W. W.(2018)의 경우 게임 개발에서 디지털스토리텔링을 활용하여 게임 개발 학습경험을 제공하였는데 게임 창작을 통한 프로그래밍 학습과 게임 수정하는 것을 통한 프로그래밍 학습방법을 적용하였다[15].

de Paula, B.H.외(2018)의 연구에서 14세 소년 2명을 대상으로 문학작품을 기반으로 MissionMaker 프로그램을 활용한 게임제작 교육과정을 운영을 통해 컴퓨팅사고와 STEAM 영역의 연계하였다[5].

Artym,C.외(2019)의 경우 예비교사의 컴퓨팅사고 향상을 위해 Papert의 구성주의 학습프레임워크를 기반으로 예비교사를 위해 스크래치 프로그램을 활용하여 비디오 게임을 디자인하고 컴퓨팅사고를 평가하는 연구를 진행하였다[1].

Lamprou,A.(2018)은 예비교사를 위한 게임 프로그래밍 교육을 개발하여 컴퓨팅사고 향상에 긍정적인 효과를 얻었다. 3D게임제작 도구인 AgentCubes와 스크래치 등을 활용하였으며 교육방식은 스캐폴딩 기법과 프로젝트 우선 접근 방식을 활용하였다. 또한 (Fig. 1)에서와 같이 추상화(Abstraction), 일반화(Automation), 분석(Analysis) 단계로 구성된 AAA 모델을 제시하여 체계적인 프로그래밍 개념 이해를 지원할 수 있도록 제시하였다[14].

반면, 제시하고 있는 AAA 모델의 경우 3D기반의 게임 프로그래밍 교육이고 게임 구현을 위한 3D 공간내 오브젝트 배치와 상호작용 등에 대한 이해가 필요함으로 초보자를 위한 게임 프로그래밍 교육 방법으로 적용



(Fig. 1) The AAA Model[14]

하기에는 다소 부족한 면이 있다.

이와 같이 국내의 선행연구를 통해 얻은 시사점을 프로그래밍 개념화, 학습자 교육전략, 프로그래밍 도구 면에서 살펴보면 다음과 같다[5][14].

첫째, 게임 프로그래밍 교육에서 체계적인 개념화의 필요성이다. 프로그래밍의 핵심요소 중의 하나는 데이터를 처리하고 설계한 알고리즘을 프로그래밍으로 구현하는 것이다. 특히 컴퓨팅사고 관련 문제해결에 필요한 창의성, 문제분해 방식과 절차에 대한 추상화를 기반으로 자동화 및 일반화와 관련된 개념을 이해할 수 있도록 체계화된 프로그래밍 교육과정 모델이 필요하다[4].

둘째, 학습자의 동기부여 방법과 학습전략이 필요하다. 선행연구들은 학습자의 프로그래밍에 대한 개념이해 향상을 위해 스캐폴딩, 플립러닝, 토론식 수업진행방법, 프로젝트 우선 접근 방법 등 학습 주제에 따라 학습자의 흥미와 관심을 지속적으로 유도할 수 있는 교수학습 방법을 다양하게 적용하였으며 공통적으로 점진적이며 수준이 향상되는 프로그래밍 개념과 관련하여 의사소통을 강화하는 방법을 적용하였다.

셋째, 학습자 수준에 맞는 프로그래밍 도구 선택이 필요하다. 일반적으로 게임 제작방법 면에서 게임을 만들기 위해서는 다양한 학문적인 지식과 이해, 경험을 필요로 한다. 특히 제작도구로 활용되는 프로그래밍 언어의 경우 Java, Unity3D 등이 많이 사용되고 있는데 초보자의 경우 프로그램에 대한 문법과 복잡한 구조 명령어, 다양한 자원의 관리와 이벤트 설계와 구현 등 처음 프로그래밍을 접하는 학습자가 쉽게 접근하기에는 어려움이 많이 있다[9]. 선행연구에서도 3D 기반 Kodu와

AgentCubes, 2D 기반 블록 프로그래밍 언어인 스크래치, 앱인벤터 등이 활용되었다. 특히 블록 프로그래밍 언어인 스크래치의 경우 게임의 몰입요소들과 연계한 다양한 멀티미디어, 스프라이트 처리, 이벤트 구현 등 프로그래밍의 핵심요소인 데이터 처리에 있어 쉽고 유연하게 적용할 수 있는 장점을 가지고 있다.

3. 게임 프로그래밍 교육모델 설계

3.1 설계 전략

초등 예비교사들의 게임 프로그래밍 교육을 위한 교육과정 모델 설계를 위해 기존 선행연구에서 얻은 시사점을 기반으로 다음과 같은 전략을 적용하였다.

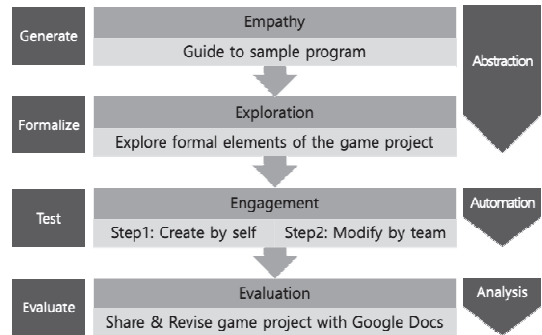
첫째, 게임 프로그래밍 교육과정 모델은 게임 프로그래밍에 대한 체계적인 개념을 학습할 수 있도록 게임 개발 과정 융합, 창의적 아이디어 생성, 게임 요소와 디자인을 융합한 추상화, 단계적 프로그래밍 훈련, 프로젝트 기반 피드백 강화 전략을 마련하였다.

둘째, 학습자의 동기부여 및 개념 강화를 위한 전략으로 게임 제작과정 속에 학습자의 다양한 가치와 사고방식이 전달될 수 있는 예제학습, 자기 게임 생성, 팀 기반 프로젝트 개발단계로 된 3중 구조의 교수학습방식을 제시하였다.

셋째, 학습자 수준에 맞는 프로그래밍 언어로 스크래치 언어를 사용하였다. 스크래치 언어는 블록기반 프로그래밍 교육언어로 스프라이트 중심의 프로그래밍 구현, 다양한 멀티미디어자료에 대한 쉬운 접근, 온라인 기반 공유 환경 등을 제공하여 처음 게임을 제작하려는 학습자들에게 유용한 언어이다.

3.2 교수학습모형 설계

예비교사를 위한 게임 프로그래밍 교육을 위해 4E(Empathy-Exploration-Engagement-Evaluation) 모델을 AAA 모델을 기반으로 (Fig. 2)에서와 같이 설계하였으며 단계별 세부내용을 살펴보면 다음과 같다[14].



(Fig. 2) Game Programming Education Model 4E

첫 번째 단계는 공감(Empathy)하는 단계이다. 모델 설계 전략에서 제시한 것과 같이 문제해결을 위한 간단한 예제 게임 프로그램을 제작해 봄으로써 해당 차시에서 핵심적으로 배워야할 개념 요소를 학습하게 된다. 이를 통해 학습자는 게임 목표를 설정하기 위해 자신만의 창의적인 아이디어를 추가할 수 있고 절차적 접근 면에서 다른 학습자에게 자신의 사고와 의미를 전달할 수 있는 준비를 갖추게 된다[5].

두 번째 단계는 탐색(Exploration)단계이다. 학습자는 공감단계에서 예제 프로그램을 통해 학습한 프로그래밍 명령어 블록들을 통해 얻은 지식을 기반으로 자신만의 게임 아이디어를 추가한다. 이 때 게임의 규칙, 플레이어 및 오브젝트 생성과 배치 등 게임 요소들을 가지고 디자인하고 다양한 조건과 절차들을 설계한다.

세 번째 단계는 강화(Engagement)단계이다. 학습자 동기부여 강화 전략에 따라 개발한 3중 구조의 프로그래밍 학습방법에 따라 예제 프로그램을 통한 학습 지식을 기반으로 자신만의 게임 프로그램 생성 및 테스트하고 이를 기반으로 팀 기반의 프로젝트 개발로 구성한다[14].

3중 구조의 프로그래밍 학습방법을 세부적으로 살펴보면 1차는 예제학습을 통한 경험과 탐색 단계에서 설계한 게임 요소들을 기반으로 2차로 자신만의 간단한 게임 프로그램을 만들고 실행해 봄으로써 명령어 블록에 대한 개념과 구조에 대해 심화된 학습이 이루어진다. 3차는 팀 기반의 프로젝트에서 자신의 게임 제작 경험을 바탕으로 교수자가 제시했던 간단한 예제 게임을 발전시킨 게임을 제작한다. 이 때 자원에 대한 데이터 처리, 플레이어와 오브젝트에 대한 충돌 상황처리, 다양한

이벤트 처리와 오류 등을 수정하면서 개발한다[3].

네 번째 단계는 평가(Evaluation)단계로 같은 수강반 학습자들을 대상으로 온라인 구글 공유문서에 팀별로 제작한 프로젝트 결과물을 링크하여 학습자들간의 다양한 피드백과 의사소통을 통해 팀별 프로젝트에 대한 개선사항을 반영하여 일반화한다.

3.3 교육과정 설계

<Table 2>와 같이 4E 모델을 적용한 예비교사의 게임 프로그래밍 교육과정을 개발하였으며 세부적인 내용은 다음과 같다.

첫째, 교육과정 진행방법은 일반적인 게임개발 과정의 핵심 개념들을 학습할 수 있도록 일반적인 게임개발 단계에 따라 총 15주차로 구성된 아이디어 생성, 게임요소 형식화, 게임 프로그램 개발과 테스트, 프로젝트 평가와 개선 단계로 구성하였다. 또한 각 단계에서 핵심적으로 다루는 명령어블록들을 게임과 연계하여 학습할 수 있도록 하였다.

또한 설계한 4E 교수학습모형 모델에서 제시하고 있는 4단계는 각 주차에서 진행되는 학습모형일 뿐만 아니라 전체 4단계로 진행되는 게임개발 단계에도 동시에 적용되는 형태로 실시하였다.

<Table 2> Game programming curriculums based on 4E model

Week	Learning Contents	Learning Activities
Step 1: Generate ideas, [Blocks: looks, sound]		
1~2	Concept of Computing Formal elements of Game Scratch Programming Basic	Writing about computing
3~4	Sprite Moving (Moving, look blocks) Simple Events(sound blocks) Make a team and participate!	Making a simple algorithm within 3 steps
Step 2: Formalize game elements, [variables, loop blocks]		
5	Game: Aircraft I Sprite actions control with move, sound and pen blocks, Default game elements	Drawing a Time-warp with polygon, Player and object conflicts processing
6	Game: Aircraft 2 Sprite duplicate, Operating variables with control and sensing blocks	Implementing the time-warp and a missile launching procedures

Step 3: Develop and Test, [Blocks: broadcasting, control]		
7	Game: Maze 1 Design of game rule and level Maze graphic design, Conflict events processing between players and objects	Design of game level, Data processing of score according to game rule and conflict events
8	Game: Maze 2 Players action(jump, run, etc.) handling with key input, Understanding of computing	Jump action implementation(automation)
9	Learning about game storytelling elements(character, events, image, background music playing, etc.), Talking and reflecting on the game story by each team created	Events procedures handling about the project with a mind map tool
10	Game: Music 1 Scratch and micro:bit, Learning about player control techniques	Simple input value handling through various sensors
11	Game: Music 2 Learning about list blocks, Resources control with various game conflict events and boundaries between objects	Implement of various conflict events and boundaries operation in the game
Step 4: Evaluate project, [Blocks: string, my blocks]		
12	Game: quiz Using a list block, Player expression and control with external blocks or resources: ex. using translate blocks	Data linkage process handling with external blocks or resources
13~15	Team project creation	Generalization of game programming

둘째, 전략에서 제시한 것과 같이 각 영역별 게임 프로그래밍과 관련된 알고리즘의 요소들이 점진적으로 심화될 수 있도록 구성하였다. 컴퓨팅의 원리, 기초 블록에 대한 이해, 데이터 처리, 다양한 블록을 활용한 이벤트 구현 등을 통해 게임의 형식적 요소들을 체계적이고 단계적으로 학습할 수 있도록 하였다.

셋째, 개발한 알고리즘을 기반으로 프로그래밍으로 구현하는 과정 속의 어려움을 해소하기 위해 3중 구조로 된 학습절차를 마련하였다[3]. 예제 프로그램을 통한 게임 추상화(Abstraction), 개별 게임 만들기과 팀별 게임 프로젝트를 통한 자동화(Automation), 스크래치 소스 공유와 리믹스 기능을 활용하여 온라인 문서공유 기반 환경에서의 프로젝트 개선을 위한 분석화(Analysis)로 제시하였다[14]. 이를 통해 학습자는 알고리즘을 설계하고 프로그래밍으로 구현하는 과정에서 필요한 절차에 대한 의미와 사고를 전달하는 과정 속에 겪는 어려움을 해소할 수 있도록 하였다.

4. 연구 분석

‘초등컴퓨팅 III’ 교양필수과정을 통해 3학년 예비교사를 대상으로 설계한 4E 모델과 게임 프로그래밍 교육과정을 2019년 1학기 15주 동안 실시하였다. 설문조사 방법은 5점 리커트 척도로 구성된 각 영역별 설문문항을 구성하여 온라인으로 실시하였고 분석은 SPSS 도구를 활용하였다.

<Table 3>에서와 같이 크게 블록 프로그래밍 역량, 4E모델 단계 속의 컴퓨팅사고 요소에 대해 사전, 사후 인식을 조사였고 게임 프로그래밍에 필요한 명령어들에 대한 개념 이해 부분은 사후 결과를 측정하였다. 또한 학습자들의 컴퓨팅사고 수준을 분석하기 위해 팀별로 제출받은 교육게임 결과물들을 Dr.Scratch 도구를 활용하여 평가하였다[23].

<Table 3> Questionnaires of the survey

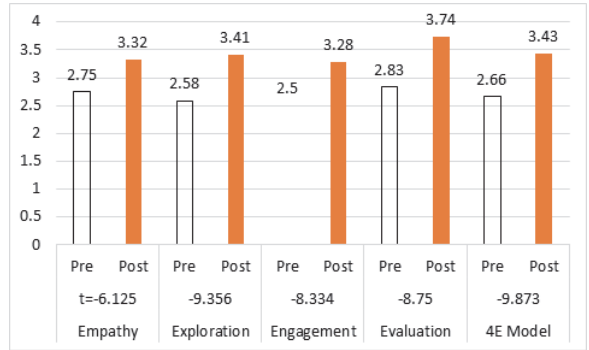
Div.	Details
Block	
Programming Block	programming competency
Competency	
4E Model	Empathy: Thinking about elements to solve the problem
	Exploration: Structured and deployed by computing process
	Engagement: Implement the game with elements by self and create a team project based on self-game
	Evaluation: Sharing and revising team projects on cloud
	Skills Using Programming Blocks
Generate: looks, sound blocks	
Formalize: variables, loop blocks	
Develop&Test: broadcasting, control blocks	
Evaluate: string, my blocks	

설문 응답자 현황을 살펴보면 설문 사전, 사후 검사 결과자료의 결측값을 제외하고 남자 40명, 여자 77명 총 117명을 대상으로 하였다. 설문 응답에 대한 신뢰도인 크론바흐 알파값은 사전 .805, 사후 .865로 높게 나타났다.

4.1 4E 모델 효과 분석

4E 모델에 대한 사전, 사후 대응표본 t-검증을 실시한

결과는 (Fig. 3)과 같다. 4E 모델에 대한 사전 사후 종합 검증 결과 사전 평균은 2.66, 사후 평균은 3.43이며 p 값은 .000으로 통계적으로 유의미한 결과를 얻었다($p<.001$).



(Fig. 3) Paired t-test results of 4E Model

세부적으로 4E 모델 단계별 사전, 사후 대응표본 t-검증 결과를 살펴보면 모든 단계에서 유의미한 효과 ($p<.001$)를 얻은 것으로 나타났다.

또한 4E 모델의 평가단계 및 탐색단계가 가장 큰 변화를 나타낸 것으로 분석되었다. 탐색단계는 게임 프로그램 구현을 위해 예제 샘플을 보고 필요한 게임의 요소들을 파악하여 자신만의 알고리즘을 설계하는 단계로 추상화 과정이 효과적으로 작용한 것으로 분석된다. 평가단계는 강화단계에서 완성된 팀별 프로젝트를 개선하는 과정을 의미하며 다양한 학습자들간의 피드백을 반영하여 자동화하는 단계가 유의미한 효과를 나타낸 것으로 보인다.

4.2 블록 프로그래밍 역량

예비교사들의 게임 프로그래밍 역량에 영향을 미치는 요인들을 분석하기 위해 학습자들의 블록 프로그래밍 역량 향상에 대한 인식 변화와 사용하는 블록 명령어들의 활용정도에 대해 분석하였다.

먼저 <Table 4>에서와 같이 학습자들의 블록 프로그래밍 역량에 대한 사전, 사후 t-검증결과는 사전 평균 2.59, 사후 평균 3.54로 $p<.001$ 수준에서 통계적으로 유의미한 효과가 있는 것으로 나타났다.

<Table 4> Analysis Result of Block Programming Competency

Div.	M	N	SD	t	p
Pre	2.59	117	.910	-8.106	.000***
Post	3.54	117	.726		

***p<.001

이와 관련하여 게임 프로그래밍을 위한 총 15주의 교육과정을 적용하면서 설계한 교육과정에 따라 블록 명령어에 대한 이해도가 학습자의 블록 프로그래밍 역량에 미치는 영향을 알아보기 위해 <Table 5>에서와 같이 영역별 핵심 블록 명령어에 대한 활용도를 독립변수로 하고 블록 프로그래밍 역량을 종속변수로 하는 다중회귀분석을 실시하였으며 그 결과를 살펴보면 다음과 같다.

<Table 5> Analysis of Factors Affecting Block Programming Competency

Variables	B	Std. Error	β	t	p
(constant)	1.43	0.349	-	4.095	.000
Generate: looks, sound blocks	0.144	0.056	0.229	2.581	.011*
Formalize: variables, loop blocks	0.228	0.089	0.247	2.572	.011*
Develop & Test: broadcasting, control blocks	0.244	0.094	0.255	2.585	.011*
Evaluate: string, my blocks	-0.068	0.07	-0.087	-0.968	.335

R=.542, R²=.294, F=11.655, p=.011(* p<.05), Durbin-watson=2.088

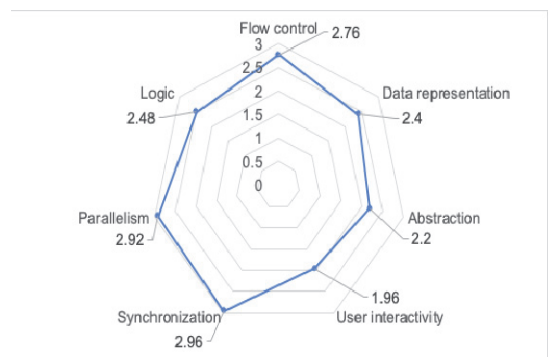
설계한 게임 프로그래밍 교육과정에서 아이디어 생성(Generate), 게임에 필요한 요소 구상(Formalize), 게임 제작(Develop&Test)에서 핵심적으로 활용하는 블록들은 모두 통계적으로 유의미한 영향을 주는 것으로 나타났다. 특히 표준화된 베타계수의 값을 살펴보면 형태 및 소리블록 0.229, 변수 및 반복블록 0.247, 방송 및 조건블록 0.255로 나타났다. 이는 점진적으로 명령어 개념이 해가 체계화 될수록 학습자의 블록 프로그래밍 역량에 더 많은 효과를 주는 것으로 분석된다.

다만 평가(Evaluate)에서 활용되었던 퀴즈게임 속 문자열 복합처리, 추가블록에 대한 활용도는 영향을 주는 요인으로 나타나지 않았다. 이것은 해당 단계에 대한 학습기간이 약 1~2주로 짧고 및 초보 학습자의 수준에서 사용자 블록 활용은 다소 어려움이 따르는 것으로 판단된다.

4.3 컴퓨팅사고 수준 평가

컴퓨팅사고 평가 방법은 시험을 통한 역량평가, 사전 사후 베브라스 CT 검사, 스크래치 소스 분석틀 활용 통한 CT 평가 등 다양한 방법을 적용할 수 있다[23]. 특히 Dr.Scratch는 스크래치 소스에 포함되어 있는 컴퓨팅사고 요소를 평가하는 도구로 프로젝트 결과물을 업로드하거나 간단히 공유된 URL을 입력함으로써 다른 CT 평가방법과 달리 결과를 즉시에 얻을 수 있어 프로그래밍 교육 관련 평가와 피드백 연구에 다양하게 활용되고 있다[13][23]. 또한 Dr.Scratch는 절차 제어, 데이터표현, 추상화, 사용자 상호작용, 동기화, 병렬화, 로직 7개로 구분되고 항목별에 대한 등급은 1~3점으로 매기고, 각 항목별 점수 합산으로 0~7점은 기본, 8~14점은 개발자, 15~21점은 마스터로 제시한다[18][23][32].

제출한 팀 프로젝트 결과물 총 25개에 대한 평가 결과 전체 평균은 17.68로 나타났다. 또한 (Fig. 4)에서와 같이 전체적인 항목별 평균을 살펴보면 동기화 2.96점, 병렬화 2.92점 순으로 나타났고 가장 낮은 항목은 사용자 상호작용 1.96점으로 분석되었다.



(Fig. 4) Students' Computational Thinking Dimension by Team Project

팀별 제작 프로젝트에 활용된 블록들이 게임처리를 위한 방송하기 블록, 2개 스크립트 이상 병렬처리 등과 같이 게임의 충돌, 경계설정 등의 요인들을 잘 이해하고 적용하였음을 나타내고 있다. 또한 앞 절에서 블록 프로그래밍의 역량에 미치는 큰 영향 요인이었던 방송블록 활용 등의 결과와 유사하게 나타났다. 반면 사용자 상호

작용 항목에서는 다소 낮은 평균을 보였는데 제작한 게임에서 키 입력과 같은 단순한 입력처리에 제한되어 발생하는 요인으로 분석되며 박선주(2018)의 연구와 유사한 결과를 보였다[23]. 따라서 사용자 상호작용 부분을 강화하기 위해서는 센서보드 등과 같은 입력소스를 다양화할 수 있는 교육도구를 추가하여 적용할 필요가 있다.

5. 결론 및 제언

본 연구에서는 초등 예비교사들을 위한 체계적인 프로그래밍 교육을 위해 게임 프로그래밍 교육모델인 4E 모델과 교육과정을 설계하고 15주간 적용하였으며 이에 대한 주요한 결과를 요약하면 다음과 같다.

첫째, 학습자의 체계적인 프로그래밍 교육을 위해 게임 기반의 교육모델을 제시하였다. 교육모델은 공감(Empathy), 탐색(Exploration), 강화(Engagement), 평가(Evaluation) 단계로 구성하였고 학습자의 프로그래밍 개념이해를 위해 예제 프로그램 작성, 자신만의 게임 프로그램 만들기, 팀별 프로젝트로 완성하는 3중 구조의 학습방법을 적용하여 모든 단계에서 통계적으로 유의미한 결과를 얻었다.

둘째, 교육모델을 기반으로 설계한 교육과정은 아이디어 생성(Generate), 게임에 필요한 요소 구상(Formalize), 게임 제작(Test), 평가(Evaluate) 단계로 총 15주 분량으로 제시하였다. 또한 게임 프로그래밍 교육과정에서 각 단계별로 핵심적으로 활용하는 블록 프로그래밍 명령어들을 학습할 수 있도록 하였다. 이를 통해 학습자의 블록 프로그래밍 역량에 미치는 효과를 분석한 결과 평가단계를 제외한 나머지 모든 단계에서 유의미한 효과가 있는 것으로 나타났다. 다만, 평가단계에 활용된 블록의 경우 초보자 수준에서 다소 이해하기 어려운 사용자 블록 활용에 대한 내용으로 깊이 있게 학습할 기간이 더 필요한 것으로 분석된다.

셋째, 학습자들의 컴퓨팅사고 수준에 대해 제출한 25개의 팀별 프로젝트 결과물을 Dr.Scratch 도구를 활용하여 분석한 결과 기본수준은 없었으며 개발자 단계 2개, 마스터단계 23개로 아주 높은 수준의 역량을 보였다. 또한 세부적으로도 동기화나 병렬화 영역에서 높은 점수를 나타내어 학습자들이 게임의 충돌, 경계설정과 처리와 같은 요소들을 잘 이해하고 프로그래밍으로 구

현하였음을 보여주었다. 반면 사용자 상호작용면은 다소 낮게 나타났으나 이는 게임의 입력처리 부분이 대부분 키보드 입력으로 제한적이었으며 센서보드와 같은 피지컬 컴퓨팅 도구를 활용하여 입력방법을 확대하여 보완할 필요가 있었다.

본 연구는 학습자의 체계적인 프로그래밍 교육과 관련하여 게임 제작을 통한 명령어 이해활동과 역량에 대한 부분을 다루었다. 그러나 학습자의 프로그래밍에 대한 어려움 해소와 보다 체계적인 프로그래밍 교육 전략을 수립하기 위해서 프로그래밍 교수학습활동 속에서 발생하는 학습활동, 학습자 상호작용과 역량 관계 등에 대한 후속 연구가 필요하다.

참고문헌

[1] Artym, C., & Carbonaro, M. (2019, June). Examining Pre-Service Teachers' Computational Thinking: Are there Differences Between Gender in Digital Game Construction?. *In EdMedia+ Innovate Learning* (pp. 552-557). Association for the Advancement of Computing in Education (AACE).

[2] Bernadette Spieler, Petri, A., Schindler, C., Slany, W., Betran, M., Boulton, H., ... & Smith, J. (2018). *Pocket Code: a mobile app for game jams to facilitate classroom learning through game creation*. arXiv preprint arXiv:1805.04461.

[3] Chul-Hyun Lee (2019). Elementary School Teachers' Difficulties in Learning Programming EPL. *Journal of Korean Practical Arts Education*, 32(2), 49-63.

[4] Cuny, J. (2015). Transforming K-12 computing education: AP® computer science principles. *ACM Inroads*, 6(4), 58-59.

[5] de Paula, B. H., Burn, A., Noss, R., & Valente, J. A. (2018). Playing Beowulf: Bridging computational thinking, arts and literature through game-making. *International journal of child-computer interaction*, 16, 39-46.

[6] do Carmo Nogueira, T., de Souza Campos, E., &

- Ferreira, D. J. (2018, July). Cognition Developing of Computer Higher Education Students Through Gamification in the Algorithm Teaching-Learning Process. *In 26th Workshop sobre Educação em Computação (WEI 2018)*, 26(1). SBC.
- [7] Fullerton, T. (2018). *Game design workshop: a playercentric approach to creating innovative games*. AK Peters/CRC Press.
- [8] Gee, E. R., & Tran, K. M. (2016). Video game making and modding. *In Handbook of research on the societal impact of digital media*(pp. 238-267). IGI Global.
- [9] Hong-Sub Lee, Hyung-Won Jeong, & Young-Kyo Kim (2016). Design and Implementation of Game for Learning Game Production Principles: Centering on Scratch Language. *Journal of Digital Convergence*, 14(5), 403-410.
- [10] Jeongmin Lee, & Somang Kim. (2019). Qualitative research of perception and experience of elementary pre-service teachers about SW education. *Journal of The Korean Association of Information Education*, 23(1), 39 - 53.
- [11] Jeongrang Kim (2016). Effects of Pair Programming on Creative Problem-Solving Ability and Efficiency. *Journal of The Korean Association of Information Education*, 20(1), 21 - 28.
- [12] Jin-Suk Park, Hyun-Jin Jung, & Jong-Seung Park (2008). Utilizing Computer Games To Improve Academic Learning Ability. *Journal of The Korean Society for Computer Game*, no.14, 71-80.
- [13] Juyeon Park (2019). Evaluation of Computational Thinking through Code Analysis of Elementary School Students' Scratch Projects. *Journal of The Korean Association of Information Education*, 23(3), 207 - 217.
- [14] Lamprou, A., & Repenning, A. (2018). Teaching how to teach computational thinking. *In Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 69-74.
- [15] Lau, W. W. (2018). Learning With Games and Digital Stories in Visual Programming. *In Encyclopedia of Information Science and Technology, Fourth Edition* (pp. 3309-3316). IGI Global.
- [16] Lee, U.-J., & BYUN, D. H. (2019). Gamification Convergence Contents Study: Formal Elements of the Game Implemented in. *The Journal of the Korea Contents Association*, 19(8), 483-489.
- [17] Misong Kim & Hyungshin Choi (2018). Fostering Primary Pre-service Teachers' Computational Thinking through Self-Assessment. *Journal of The Korean Association of Information Education*, 22(1), 61 - 70.
- [18] Moreno-León, J., & Robles, G. (2015, November). Dr. Scratch: a Web Tool to Automatically Evaluate Scratch Projects. *In WiPSCE*, 132-133.
- [19] Myunghee Kang, Juyeon Park, Seonghye Yoon, Minjeng Kang, & JeeEun Jang (2016). The Mediating Effect of Learning Flow on Affective Outcomes in Software Education Using Games. *Journal of The Korean Association of Information Education*, 20(5), 475-486.
- [20] PhanWoo Park (2016). A Method on the Convergent Software Education with GameProgramming. *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, 6(2), 27-34.
- [21] Salen, K., Tekinbaş, K. S., & Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. MIT press.
- [22] Sungkyung Kim, Sangchul Kim (2018). Middle-School Programming Classes Utilizing App Inventor Game Creation and the Analysis of their Educational Outcomes. *Journal of Korea Game Society*, 18(3), 49-60.
- [23] Sunju Park (2018). Analysis of Computational Thinking Level Through the Scratch Project Analyzation. *Journal of The Korean Association of Information Education*, 22(6), 661 - 669.
- [24] Ui-Sung Song, & Joon-Min Gil (2017). Development and Application of Software

Education Program Based on Blended Learning for Improving Computational Thinking of Pre-Service Elementary Teachers. *KIPS Transactions on Software and Data Engineering*, 6(7), 353-360.

- [25] Wae-shik Moon (2018). Analysis of error data generated by prospective teachers in programming learning. *Journal of The Korean Association of Information Education*, 22(2), 205 - 212.
- [26] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- [27] 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.
- [28] Young-Hoon Sung (2016). Development and Application of CT-SPI Model for Improving Computational Thinking for Elementary School Students. *Journal of The Korea Society of Computer and Information*, 21(1), 169-180.
- [29] Youngshin Han(2018). Analysis of Effectiveness of Programming Learning for Non-science Major Preliminary Teachers' Development of Computational Thinking. *Journal of The Korean Association of Information Education*, 22(1), 41 - 52.
- [30] Youngsik Jeong, & Jeongsu Yu(2014). The Effects of Elementary Students' Perception of the Game Programming Using KODU. *Journal of Korea Association of Information Education*, 18(3), 453 - 460.
- [31] Young-Sik Jeong, Soo-Bum Shin, & Young-Hoon Sung (2016). Analysis of Appropriateness in Information Curriculum for Algorithm and Programming Education. *Journal of The Korean Association of Information Education*, 20(6), 575-584.
- [32] Dr.Scratch(2019). Dr.Scratch. Retrieved from <http://www.drscratch.org/>

저자소개



성 영 훈

2010. 경상대학교 컴퓨터과학(공학박사)
 2011~2015. 한국교육학술정보원 연구원
 2015~현재 진주교육대학교 컴퓨터교육과 조교수
 관심분야 : SW교육, 컴퓨팅융합교육, 국가행정정보시스템
 E-mail : yhsung@cue.ac.kr