



유니티를 활용한 문제해결과정 중심의 프로그래밍 교육 프로그램이 일반계 고등학생의 컴퓨팅사고력과 학습 몰입에 미치는 영향: 문제해결 보조도구로서 생성형 AI의 활용 가능성 탐색

The Effects of a Problem Solving Process-Oriented Programming Education Program Using Unity on General High School Students' Computational Thinking and Learning Flow: An Exploratory Study on the Potential Use of Generative AI as a Problem Solving Support Tool

유진솔[†] · 김귀훈^{††}
Jinsol Ryu[†] · Kwihoon Kim^{††}

요약

본 연구에서는 고등학교 프로그래밍 수업에 적용하기 위한 유니티 활용 문제해결과정 중심의 교육 프로그램을 개발하고 타당성 및 효과를 검증하였다. 교육 프로그램은 컴퓨팅사고력 구성 요소를 기반으로 한 문제해결과정 단계를 중심으로 설계되었으며, 전문가 검토를 통해 타당성을 확보한 뒤 20차시로 구성하여 일반계 고등학교 2학년에게 적용하였다. 사전·사후 검사 결과, 대응표본 t-검정과 효과 크기(Cohen's d) 분석을 통해 컴퓨팅사고력과 학습 몰입 모두에서 통계적으로 유의한 향상이 확인되었다. 또한 문제 정의 및 프로그래밍 과정(7~12차시)에서는 생성형 AI를 활용하여 학습자의 아이디어 구체화와 디버깅을 위한 효과적인 지원 방안을 모색하였다. 본 연구는 고등학교 프로그래밍 교육에서 학습자의 자발적 참여와 몰입을 증진할 수 있는 교수·학습 전략의 확장 가능성을 제시했다는 점에 의의가 있다.

주제어 유니티, 문제해결과정, 프로그래밍 교육, 컴퓨팅사고력, 학습 몰입, 생성형 AI

ABSTRACT

This study developed and validated a Unity-based, problem solving process-oriented programming education program for second-year general high school students, structured around computational thinking components and implemented over 20 sessions. Expert review ensured design validity, and paired-sample t-tests and effect size (Cohen's d) analyses confirmed significant improvements in computational thinking and learning flow. Additionally, generative AI was integrated into the problem definition and programming stages (sessions 7~12) to support idea refinement and efficient debugging, enhancing student engagement and the overall effectiveness of programming education.

Keywords Unity, Problem Solving Process, Programming Education, Computational Thinking, Learning Flow, Generative AI

†정회원 한국교원대학교 대학원 인공지능융합 교육전공 박사과정
††중신회원 한국교원대학교 인공지능융합교육전공 부교수 (교신저자)
논문투고 2025년 11월 07일
심사완료 2026년 01월 08일
게재확정 2026년 01월 22일
발행일자 2026년 03월 15일

1. 서론

디지털 대전환 시대를 살아가는 학생들에게 컴퓨팅사고력(Computational Thinking)은 미래 사회의 핵심 역량으로 자리매김하고 있다. 컴퓨팅사고력은 4차 산업혁명 시대의 필수 역량으로 주목받고 있으며, 인공지능, 데이터 과학, 자동화 등 다양한 기술 변화 속에서 문제 해결과 창의적 사고를 이끌어내는 기반이 된다[1]. 이러한 맥락에서 컴퓨팅사고력은 단순히 컴퓨터를 다루는 기술을 넘어, 복잡한 문제를 분석하고 알고리즘적으로 사고하며 해결책을 창출하는 인지적 과정이라 할 수 있다.

2022 개정 정보과 교육과정에서도 컴퓨팅사고력을 핵심 역량으로 제시하며, "컴퓨팅을 활용한 문제 해결을 위해 문제를 발견, 이해하여 다양한 해법을 설계하고 프로그램을 통해 구현해 가는 자동화의 중요성을 이해하고 실천하는 태도를 기르는 것"이라 명시하고 있다[2]. 이 또한 컴퓨팅사고력이 단순한 지식의 습득을 넘어 추상화, 자동화 등의 핵심 요소를 기반으로 한 실천적 문제해결 역량임을 강조하는 것이다.

이러한 교육과정의 방향에 따라 정보교과에서도 프로그래밍 교육이 점차 확대되고 있으며, 교육 방법 역시 문법 위주의 단편적 교수에서 벗어나 컴퓨팅사고력의 핵심 요소를 효과적으로 신장시키기 위한 다양한 교육적 접근이 요구된다. 이에 따라 학습자가 스스로 문제를 정의하고 해결해 나가는 문제해결과정 중심의 수업이 주목받고 있으며, 문제해결과정 전반에 컴퓨팅사고력의 주요 요소를 통합적으로 반영하는 교육적 실천 방안이 지속적으로 논의되고 있다[3-5].

한편, 컴퓨팅사고력을 효과적으로 향상시키기 위해서는 학습자의 동기와 자발적 참여를 유도하는 몰입감 있는 학습 환경이 필요하다. 이에 본 연구에서는 게임 개발 엔진인 유니티를 프로그래밍 교육 도구로 활용하는 방안을 제안하고자 한다. 유니티는 게임·시뮬레이션 개발을 지원하는 통합 개발 환경으로[6], 이를 활용한 프로그래밍 수업은 문법에 대한 부담을 완화하고 게임 내 요소 설계 및 개발 활동을 통해 학습자의 흥미와 몰입을 유도할 수 있다[7]. 또한 프로그래밍 과정에서 생성형 AI를 보조 도구로 활용할 경우, 문제 해결 과정에서 학습자의 즉각적인 피드백 경험을 지원함으로써 과제 수행에 대한 성취감과 자기효능감을 높이고 문제 해결 활동에 지속적으로 참여하며 학습자의 몰입을 유지·증진하는 데 기여할 수 있다[8].

따라서 본 연구는 유니티를 활용한 문제해결과정 중심 프로그래밍 교육 프로그램을 ADDIE 교수학습모형을 기반으로 설계하고, 이를 일반계 고등학생에게 적용하여 컴퓨팅사고력과 학습몰입에 미치는 영향을 검증하고자 한다. 이를 통해 일반계 고등학교에서 실천 가능한 효과적인 문제해결과정 중심 프로그래밍 교육 방법을 제시하고, 학습자의 컴퓨팅사고력과 학습몰입을 동시에 증진할 수 있는 교수학습 전략을 설계하는 것을 목적으로 한다. 이러한 연구 목적을 달성하기 위하여 연구 문제를 다음과 같이 설정하였다.

첫째, 일반계 고등학교의 프로그래밍 수업에 적용하기 위한 유니티 활용 문제해결과정 중심의 교육 프로그램은 어떻게 구성되어야 하는가?

둘째, 개발된 교육 프로그램은 일반계 고등학생의 컴퓨팅사고력과 학습 몰입을 증진하는 데 타당한가?

2. 이론적 배경

2.1 컴퓨팅사고력

컴퓨팅 사고력은 Wing에 의해 처음 제안된 개념으로, 컴퓨터 과학의 기본 개념과 원리를 활용하여 문제를 효과적으로 해결하는 사고 과정을 의미한다[9]. Wing은 컴퓨팅 사고력이 단순한 프로그래밍 능력을 지칭하는 것이 아니라, 추상화와 자동화를 중심으로 문제를 분석하고 해결하는 사고 방식이라고 정의하였다[9].

ISTE와 CSTA는 컴퓨팅 사고력을 컴퓨터를 활용하여 문제를 해결하기 위한 기법으로 정의하고, 이를 구체화하기 위한 9가지 조작적 정의 요소를 제시하였다[10]. 이러한 정의는 컴퓨팅 사고력을 단순한 기술 습득이 아닌 문제해결과정 중심의 사고 기법으로 접근하고 있다는 점에서 특징적이다.

교육부와 Keris는 SW교육 운영지침을 통해 컴퓨팅 사고력의 요소를 자료 수집, 자료 분석, 구조화, 추상화, 자동화, 일반화의 여섯 가지로 구분하여 제시하였다[11]. 각 기관과 연구자들이 제시한 컴퓨팅 사고력의 세부 구성 요소를 Table 1에 제시하였다.

Table 1. Comparison of CT Components

Wing(2006)	ISTE&CSTA(2011)	Ministry of Education·KERIS (2015)	
Abstraction	Data Collection	Data Collection	
	Data Analysis	Data Analysis	
	Data Representation	Structuring	
	Problem Decomposition	Abstraction	Decomposition
	Abstraction		Modeling
Algorithms and Procedures	Algorithm		
Automation	Automation	Automation	Coding
	Simulation		Simulation
	Parallelization	Generalization	
	-	Generalization	

본 연구에서는 학습자들이 주어진 문제 상황을 이해하고 이를 해결해나가는 절차에 중점을 두어, 각 세부 요소별 과정지향성을 지니는 ISTE&CSTA의 컴퓨팅사고력 구성요소를 기준으로 활용하고자 한다.

2.2 문제해결과정의 단계

정보교육에서 능동적인 문제해결은 학습자의 컴퓨팅사고력 향상을 위해 강조되어 온 방법 중 하나이다. 이를 위해 컴퓨팅 사고력의 핵심 요소를 문제해결과정과 연결지어 일련의 단계로 구분한 대한 연구가 다음과 같이 이루어지고 있다.

컴퓨팅 사고력의 핵심 요소들이 일반적인 문제 해결 과정에 어떻게 적용되는지 분석한 연구에서는 문제 해결 과정을 '문제인식', '문제분석', '해결책 고안', '문제 해결의 실천', '문제 해결의 평가'로 나누고, 각 단계에 컴퓨팅 사고력의 핵심 요소를 대응시켜 제시하였다[12]. 2015 개정 교육과정을 중심으로 개발된 컴퓨팅 사고력 기반의 창의적 문제해결(CT-CPS) 수업 모형에서는 창의적 문제해결 단계를 '문제인식 및 분석', '아이디어 구상', '설계', '구현 및 평가'의 네 단계로 구성하여, 컴퓨팅 사고력이 창의적 사고 과정과 유기적으로 결합될 수 있음을 제시하였다[4]. 더불어 2022 개정 중등 정보과 교육과정의 '알고리즘과 프로그래밍' 영역 단계별 학습 요소 분석 연구에서는 '문제해결을 위한 프로그램 개발' 요소를 중심으로, 학습 단계를 '문제해결', '객체 활용', '문제해결 프로그램', '성능', '프로그램 공유'의 단계로 나누어 제시함으로써 프로그래밍 학습 과정에 내재된 문제해결적 사고를 구체화하였다[5].

선행연구의 시사점을 종합하면 컴퓨팅 사고력은 단순한 프로그래밍 기술 습득을 넘어 문제를 인식하고 분석하며, 해결책을 구상·구현·평가하는 전 과정에서 사고의 구조를 형성하는 핵심 역량임을 알 수 있다. 따라서 본 연구에서는 이러한 논의의 흐름을 반영하여 문제해결과정 중심의 프로그래밍 교육을 '컴퓨팅 사고력의 핵심 요소가 문제 해결의 전 과정에 통합적으로 적용되는 교육 접근'으로 정의하고, ISTE&CSTA의 컴퓨팅사고력 구성요소를 기반으로 한 모형[12]을 기준으로 교육 프로그램을 개발하고자 한다.

2.3 학습 몰입과 유니티 활용 교육

학습몰입(Learning Flow)은 Csikszentmihalyi가 제안한 몰입 개념을 학습 상황에 적용한 것으로, 학습자가 학습 과제에 완전히 몰두하여 집중하는 최적의 심리 상태를 의미한다[13, 14]. 즉, 학습몰입은 학습자가 외적 보상보다 학습 그 자체에서 즐거움과 성취감을 느끼며, 학습 과정에 깊이 빠져드는 심리적 경험을 의미한다. 이러한 학습 몰입의 증진을 위해 본 연구에서는 유니티 엔진을 프로그래밍 교육 플랫폼으로 활용하고자 한다.

유니티는 2005년에 개발된 게임 개발 및 시뮬레이션 환경을 제공하는 엔진이자, 통합 제작 도구이다[6]. 초보자도 쉽게 활용 가능한 직관적인 인터페이스를 제공하며, C# 프로그래밍 언어를 기반으로 하여 객체지향 프로그래밍 개념을 시각적 시뮬레이션과 연계해 학습할 수 있는 환경을 제공한다[5]. 또한 최근 유니티 AI의 개발로 디버깅 지원, 코드 자동완성, 에셋 생성 등 생성형 AI를 프로그래밍의 보조 도구로 활용할 수 있는 가능성을 확장하고 있다[15].

유니티를 교육에 활용한 기존 연구는 주로 게임 프로그래밍을 전공하는 대학생을 대상으로 하거나 상업용 게임 제작

목적에 집중되는 경향이 있었다[16, 17]. 시뮬레이션을 목적으로 중·고등 과학 교육[7, 18]에 적용한 연구도 있었으나, 정보교과에 적용가능한 연구는 상대적으로 부족하다. 본 연구는 유니티를 활용한 게임 프로그래밍 과정을 통해 문제해결의 절차를 경험하는 데 중점을 둔다는 점에서 기존 연구와 구별된다.

3. 연구 방법

3.1 연구 대상

본 연구의 대상은 충북 소재 일반계 남자 고등학교 2학년 학생 30명이다. 연구 대상자는 프로그래밍 교과를 수강하는 학생들로, 교육 프로그램 적용 전 수업을 통해 Python의 표준 입출력부터 변수, 연산자 및 제어 구조까지의 개념을 학습하였다. 이들에게 본 연구에 대한 취지와 절차를 충분히 설명한 후, 모두에게 자발적 참여 의사 및 연구 동의를 얻어 연구를 진행하였다.

3.2 연구 설계

본 연구는 문제해결과정 중심의 프로그래밍 교육 프로그램이 컴퓨팅사고력과 학습 몰입에 미치는 영향을 분석하기 위해 수행된 사전-사후 설계 연구이다. 동일 학교 내에서 실험집단과 동일한 내용 요소를 학습하는 교과 및 학급이 부재하여 비교 집단을 설정하는 데 어려움이 따라 연구 설계를 단일집단을 대상으로 한 형태로 구성하였다.

문제해결과정 중심의 프로그래밍 교육 프로그램이 컴퓨팅사고력과 학습 몰입에 미치는 효과를 파악하고자 수업 전 사전 검사를, 수업 이후 사후 검사를 실시하였다. 연구 설계는 Fig. 1과 같다.

G	O ₁	X ₁	O ₂
G: Experimental group (N=30)			
O ₁ : Pre-test (CT Scale, Learning Flow Test)			
X ₁ : Problem Solving Process-Oriented Programming Education Program Using Unity			
O ₂ : Post-test (CT Scale, Learning Flow Test)			

Figure 1. Research Design

3.3 검사 도구

3.3.1 컴퓨팅 사고력 검사 도구

컴퓨팅 사고력을 측정하기 위하여 본 연구에서는 최숙영(2016)의 문제해결과정에서의 컴퓨팅사고력 정의와 학습활동 내용을 기반으로 김미영(2018)이 개발한 문제해결기반 컴퓨팅사고력 측정도구(PSB-CTQ)를 활용하였다[19]. 이 검사 도구는 본 연구에서 활용하고자 한 ISTE&CSTA의 컴퓨팅사고력에 대한 9가지 조작적 정의요소를 바탕으로 개발

되었으며, 국내 중고등학생 326명에게 적용하고 탐색적 요인분석을 실시하였으며, 이를 통해 ‘문제인식, 프로그래밍, 해결방안탐색’ 세 가지 하위요인의 총 31문항으로 구성되었다. 응답 방식은 Likert 5점 척도를 적용하였으며, 1점은 ‘전혀 그렇지 않다’, 5점은 ‘매우 그렇다’로 구성하였다. 검사 도구의 내용은 Table 2와 같다.

Table 2. Contents of the PSB-CTQ Scale

Factor	CT Components	N	Cronbach's α
Problem Recognition	Data Collection	4	.89 (pre) .92 (post)
	Data Analysis	3	
	Data Representation	4	
Solution Exploration	Problem Decomposition	3	
	Abstraction	3	
	Algorithms and Procedures	1	
Programming	Algorithms and Procedures	4	
	Automation	3	
	Simulation	2	
	Parallelization	4	

3.3.2 학습 몰입 검사 도구

학습 몰입을 측정하기 위하여 Csikszentmihalyi의 9가지 몰입 요소에 근거하여 석임복, 강이철(2007)이 개발한 학습 몰입도 측정 도구를 사용하였다[20]. 이 도구는 ‘도전과 능력의 조화, 명확한 목표, 구체적인 피드백, 통제감, 과제에 대한 집중, 행동과 의식의 통합, 자의식의 상실, 시간감각의 왜곡, 자기목적적 경험’ 9개 하위요인의 총 35문항으로 구성되어 있으며, 응답 방식으로는 3.3.1과 동일한 5점 Likert 척도를 활용한다. 검사 도구의 내용은 Table 3과 같다.

Table 3. Contents of the Learning Flow Scale

Factor	N	Cronbach's α
balance of challenge and skill	4	.91 (pre) .89 (post)
merging of action and awareness	5	
clear goals	2	
specific feedback	5	
concentration on the task	3	
sense of control	2	
loss of self-consciousness	5	
distortion of time perception	3	
autotelic experience	6	

3.4 자료 분석

본 연구에서는 실험집단의 컴퓨팅 사고력과 학습 몰입 수준 변화를 분석하기 위해 대응표본 t-검정을 적용하였으며, 향상 효과의 통계적 유의성을 확인하기 위해 단측 검정

을 실시하였다. 사후 검사에서의 처치 효과 크기 분석에는 Cohen's d를 사용하였고, 0.2, 0.5, 0.8을 기준으로 작은, 중간, 큰 효과크기로 해석된다. 수집된 자료는 SPSS 22 프로그램을 활용해 통계 분석을 실시하였다.

4. 프로그램 개발

본 연구는 ADDIE 교수설계모형을 기반으로 20차시의 문제해결과정 중심 프로그래밍 교육 프로그램을 개발하였다. 이는 분석-설계-개발-실행-평가의 다섯 단계를 따라 설계 및 적용되었다.

4.1 분석

‘분석’ 단계에서는 교육 프로그램의 차시별 학습 내용 및 학습활동 요소 선정에 위해 교육과정 및 선행연구 분석을 진행하였다. 2022 개정 정보과 교육과정에서는 학교급간 위계를 고려해 내용 요소를 재구조화하였으며, 특히 고등학교 ‘알고리즘과 프로그래밍’ 영역에서는 중첩 제어 구조 및 함수 관련 내용 요소가 중학교로 이전되고, ‘클래스와 인스턴스’ 내용 요소가 추가되며 프로그래밍 관련 영역을 심화·확장하였다[21]. 관련 내용은 Table 4에 정리하였다.

프로그래밍에서 객체지향 기반의 사고는 실생활의 복잡한 문제를 객체로 추상화하고, 이를 기반으로 문제의 구조를 쉽게 표현할 수 있어 효율적으로 문제를 해결할 수 있다 [2]. 이를 바탕으로, 본 연구에서 개발한 교육 프로그램의 프로그래밍 학습 내용으로 ‘클래스와 인스턴스’ 내용 요소를 선정하였다.

Table 4. Restructured Curriculum - Focusing on Classes and Instances

Category	Curriculum Area	
	2015 Revised Curriculum "Problem-Solving and Programming"	2022 Revised Curriculum "Algorithm and Programming"
Content Element (Knowledge & Understanding)	<ul style="list-style-type: none"> • Program development environment • Variables and data types • Operator • Standard and File I/O • Nested Control Structure • Array • Function • Application of programming 	<ul style="list-style-type: none"> • Problem analysis and modeling • Sorting, Searching • Data Types • Standard and File I/O • Utilization of Multidimensional Data • Application of Control Structure • Class and Instances
Related Achievement Standard and Solutions	[12Inf03-08] Use classes and instances to create objects and apply them to programming. [Solution] By utilizing real-life cases of implementation, students should be able to understand and use the basic concepts needed for solving problems. By creating and using classes and objects, students can apply them to problem solving situations.	

그러나 객체지향의 추상적 설계(클래스)와 실체화(인스

턴스)의 관계, 절차적 사고와의 차이, 생소한 용어 등으로 인해 초보 학습자들이 직관적으로 이해하는 데 어려움이 따를 수 있다[22]. 이에 따라 객체지향 개념을 보다 친숙하고 구체적인 형태로 학습할 수 있는 방안으로 게임 요소 기반의 수업을 제안할 수 있다[23]. 본 연구에서는 이러한 관점을 바탕으로 유니티를 활용해 게임의 요소를 직접 설계·구현하는 과정을 통해 객체지향 개념을 직관적으로 이해하고 학습에 몰입할 수 있도록 학습활동을 구성하고자 하였다.

4.2 설계

‘설계’ 단계에서는 교육 프로그램의 구체적인 내용과 교육 도구를 선정하였다. 본 연구의 프로그램 교육 목표는 ‘클래스와 객체를 활용하여 문제 해결을 위한 프로그램을 구현할 수 있다’로 설정하였으며, 이를 바탕으로 학습 내용과 차시별 활동을 설계하였다. 특히 단순한 문법 습득이나 기능 구현에 그치지 않고 학습자가 문제를 인식하고 분석하여 해결 방안을 구상·구현·평가하는 문제해결과정 중심의 사고를 구조적으로 경험하도록 하는 데 초점을 두었다.

이를 위해 문제해결과정의 각 단계에서 강조되는 컴퓨팅 사고력 요소를 고려하여 교수·학습 전략을 설계하였다. 문제 인식 및 분석 단계에서는 자료 수집과 문제 분해를 통해 문제 상황을 구조화하도록 하였으며, 해결책 고안 단계에서는 추상화와 알고리즘 설계를 통해 문제 해결의 논리적 흐름을 명확히 표현하도록 유도하였다. 이후 문제 해결의 실천 및 평가 단계에서는 실제 프로그래밍 산출물을 구현하고, 디버깅과 개선 과정 경험할 수 있도록 하였다.

교육 도구로는 유니티와 ChatGPT를 선정하였다. 유니티는 게임 오브젝트와 컴포넌트 기반 구조를 통해 복잡한 문제를 객체 단위로 분해하고 객체 간 상호작용을 설계하는 경험을 제공한다. 이러한 특성은 객체지향 개념과 알고리즘 설계 과정을 시각적으로 확인 가능하게 하여 학습자가 문제해결과정의 구조를 보다 직관적으로 이해하도록 지원한다.

한편 ChatGPT는 문제해결을 대신하는 도구가 아닌, 학습자의 사고 과정을 보조하는 학습 지원 도구로 제한적으로 활용하였다. 문제 정의 단계에서는 학습자가 설정한 문제의 범위와 난이도, 구현 가능성을 점검하고 아이디어를 구체화하는데 활용하였으며, 프로그래밍 구현 단계에서는 코드 디버깅 과정에서 오류의 원인을 탐색하고 수정 방향을 검토하는데 활용하였다. 이때 ChatGPT가 완성된 정답이나 코드를 직접 제공하지 않도록 활용 범위를 명확히 설정함으로써 학습자의 주도적인 사고와 문제 해결 과정을 유지하도록 설계하였다.

4.3 개발

‘개발’ 단계에서는 교육 프로그램의 차시별 주요 학습 활동에 맞추어 교수학습 과정안과 수업 자료의 초안을 설계하고, 전문가 검토를 거쳐 최종 프로그램을 개발하였다.

4.3.1 교육 프로그램 초안

본 연구에서 개발한 교육 프로그램의 초안은 총 20차시로 구성되었으며, 최숙영(2016)이 제시한 문제 인식-문제 분석-해결책 고안-문제 해결의 실천-문제 해결의 평가의 다섯 단계 문제해결과정을 기반으로 설계되었다[12]. 각 단계는 컴퓨팅사고력의 핵심 요소들이 유기적으로 연계되도록 구성되어 있다.

교육 프로그램은 크게 3단계로 구분된다. 첫째, 1-2차시 유니티 기초 실습 단계에서는 학습자가 유니티 에디터의 주요 구성요소, Game Object, Component 등의 개념을 파악하고, 기본 조작을 익히며 2D 게임을 체험하여 학습 동기를 유발한다.

둘째, 3-8차시 유니티 게임 제작 실습 단계에서는 시범실습법으로 2D 플랫폼 게임 제작 과정을 실습한다. UI를 배치하고, 클래스와 객체 개념을 학습한 뒤 C# 스크립트 템플릿의 빈칸에 코드를 작성하며 캐릭터 이동, 충돌 감지, 점수 계산 등의 기능을 구현하면서 객체 활용 프로그래밍 방법을 파악한다. 아래 Fig. 2는 유니티 활용 객체 프로그래밍 실습 장면이다.

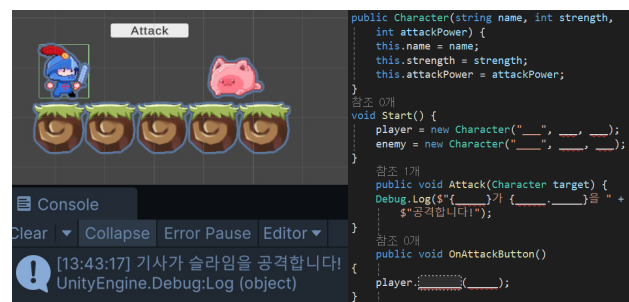


Figure 2. Scene of Object-Oriented Programming Practice (Asset source: Hong, S. (2025). Introduction to Unity Game Development by a Game Developer, AKIT.)

왼쪽 화면은 학습자가 유니티의 UI 환경에서 직접 배치한 캐릭터와 공격 버튼을 활용하여 실습을 진행하는 모습이며, 오른쪽 화면은 Character 클래스의 속성과 메서드를 기반으로 작성된 C# 코드 일부를 보여준다. 학습자는 Start() 메서드에서 new Character() 구문을 사용해 객체를 생성하고, OnAttackButton() 이벤트를 통해 객체 간 상호작용을 실행하도록 구성하였다. 버튼 클릭 시 콘솔에 “기사가 슬라임을 공격합니다!”라는 메시지가 출력되며, 학습자는 코드의 빈 칸을 직접 작성함으로써 Attack() 메서드의 동작이 시각적으로 확인되는 과정을 경험할 수 있다.

셋째, 9-20차시에서는 문제해결과정을 도입하여 수업을 전개한다. 9-10차시 ‘문제 인식’ 단계에서는 학습자가 「제로웨이스트 챌린지」 또는 「우리고장 홍보」 중 하나의 주제를 선택하여 문제해결 중심의 게임의 전체적인 설계를 구상하도록 하였다. 또한 에셋, 데이터 등 게임 개발을 위해 필요한 자료를 수집하도록 하였다.

11차시 ‘문제 분석’ 단계에서는 수집된 자료를 기반으로 핵심적인 기능만을 선별하고 불필요한 요소를 제거함으로써 문제를 단순화하도록 유도하였다. 또한 구현 가능성과

관련 자료의 확보 여부를 고려하여 실현 가능한 문제로 범위를 조정하도록 하였다.

이 과정에서 생성형 AI를 활용하여 초기 아이디어를 구체화하거나, 제시된 주제의 난이도와 구현 가능성을 검토함으로써 보다 현실적인 문제 정의가 이루어지도록 지원하였다. 이를 통해 학생들이 초기에 지나치게 거창하거나 추상적인 문제를 설정하는 경향을 완화하고, 실제 개발 가능한 수준에서 문제를 명확히 규정하는 능력을 함양할 수 있도록 하였다.

12차시 ‘해결책 고안’ 단계에서는 순서도와 의사 코드를 활용하여 알고리즘을 구체적으로 설계함으로써 논리적, 절차적 사고를 중심으로 문제해결방안을 구안해 낼 수 있도록 하였다.

13-18차시 ‘문제해결의 실천’ 단계에서는 유니티를 활용하여 UI를 구성하고, 3-8차시에서 실습했던 C# 스크립트를 활용하여 설계한 알고리즘을 실제로 구현함으로써, 추상적 설계가 구체적 산출물로 전환되는 과정을 경험하도록 하였다.

마지막 19-20차시 ‘문제해결의 평가’ 단계에서는 완성된 게임을 테스트하고 오류를 디버깅하는 과정을 수행한다. 수행 과정에서 필요 시 생성형 AI의 도움을 받아 효율적인 오류 탐지와 수정 과정을 경험하며, 구현 결과를 고도화·일반화함으로써 학습의 완성도를 높일 수 있도록 하였다.

4.3.2 전문가 검토

교육 프로그램 초안 개발 이후, 현직 정보·컴퓨터 교과 교사 및 컴퓨터 교육 분야 전문가 13명에게 교수학습 과정안 및 수업 자료를 제시하고 온라인 설문지를 이용하여 전문가 검토를 받았다.

전문가 검토 결과는 Table 5에 제시하였으며, 각 문항의 내용 타당도(CVR)는 Lawshe가 제안한 산출식을 참고해 계산하였다[24]. 전문가 13명에 따른 CVR 최솟값은 .54이며, 검토 결과 모든 문항의 CVR 값이 최솟값 기준 이상으로 내용타당도가 확보되었다고 할 수 있다.

Table 5. Export survey results

Item	Detail	CVR
Class necessity	Necessity of Problem-solving process-oriented programming class	1
	Appropriateness of 20-session class length	0.69
Educational Content and Appropriateness of Teaching Materials	Appropriateness of 1st-2nd session content	1
	Appropriateness of 3rd-8th session content	1
	Appropriateness of the order of sessions 1-8	1
	Appropriateness of the topic of the 9th-10th session project	0.85
	Appropriateness of the objectives of the 11th-20th session	1
	Appropriateness of the content of the 11th-20th session	0.69
	Appropriateness of the class composition of 11th-20th session	1
	Appropriateness of the order of the 11th-20th session	1
	Appropriateness of the connectivity between the educational contents of the 11th-20th session	1
	Appropriateness of the difficulty level of the 11th-20th session	1
Class Application	Appropriateness of the application in the actual educational field	1

Table 6. Detailed Design of the Class Program

Problem Solving Process (Choi, S. Y.)	CT Components (ISTE&CSTA)	Session	Main Activities	Instructional Focus
-	-	1st-2nd	<ul style="list-style-type: none"> Understanding the Unity interface Creating and manipulating objects Experiencing simple 2D games 	<ul style="list-style-type: none"> Minimized theoretical explanations and fragmented grammar drills. Provided fill-in-the-blank scripts that can be completed once the concept of objects is clearly understood.
		3rd-6th	<ul style="list-style-type: none"> Practicing production of a 2D platformer game <ul style="list-style-type: none"> Arranging and configuring UI Understanding ‘Class and Object’ concepts and basic C# syntax in Unity Writing scripts to implement game functions (filling blank scripts) 	
Problem Recognition	Data Collection	7th-9th	<ul style="list-style-type: none"> Selecting the topic and conceptualizing the game to produce Collecting essential materials for game development (assets, data, etc.) 	<ul style="list-style-type: none"> Expanded the scope to include environmental and local themes. Encouraged students to choose topics based on their own experiences.
	Data Analysis		<ul style="list-style-type: none"> Finding meaningful information based on the collected data 	
Problem Analysis	Data Representation	10th	<ul style="list-style-type: none"> Visualizing the analysis results 	<ul style="list-style-type: none"> Guided students to define realistic, implementable problems. Allowed students to use ‘generative AI’ to refine ideas and assess topic feasibility
	Problem Decomposition		<ul style="list-style-type: none"> Analyzing required functions and breaking them into smaller problems 	
	Abstractio	Abstractio	<ul style="list-style-type: none"> Identifying and removing unnecessary objects, data, and functions 	
Solution Design	Algorithms and Procedures	12th	<ul style="list-style-type: none"> Writing code or scripts specifying the order and method of each process in the game 	<ul style="list-style-type: none"> Encouraged clear and concrete algorithm design
Solution Implementation	Automation	13th-15th	<ul style="list-style-type: none"> Arranging and configuring Unity UI 	<ul style="list-style-type: none"> Enabled students to enhance and generalize their implementations by efficiently detecting and correcting errors with ‘generative AI’ support when needed.
		16th-18th	<ul style="list-style-type: none"> Writing C# scripts to implement algorithms 	
Solution Evaluation	Simulation	19th	<ul style="list-style-type: none"> Testing whether the game meets the intended purpose Debugging to solve errors 	
	Parallelization	20th	<ul style="list-style-type: none"> Identifying tasks executed in parallel in the game Improving and applying the completed program 	

4.3.3 최종 교육 프로그램

전문가 검토에서 CVR 값이 비교적 낮게 나타난 항목들에 대해 개방형 응답에서 제시된 의견을 반영하여 교육 프로그램을 수정·보완하고, 이를 토대로 최종안을 완성하였다. 전문가 검토 결과에 따른 초안 프로그램의 구체적인 수정 사항은 다음과 같다.

‘수업 분량의 적절성’ 항목에서 유니티 기본 실습의 차시가 비교적 방대하며 컴퓨팅사고력의 향상을 위해 문제 인식 및 알고리즘 설계 단계의 활동에 필요한 시간이 부족하다는 의견이 다수 제기되었다. 이를 반영하여 플랫폼퍼 2D 게임 제작의 차시를 줄이고, 자료 수집 및 추상화 단계의 차시를 각각 1차시씩 증가하였다.

또한, 기존 9-10차시의 ‘교육 주제의 적합성’ 및 ‘차시별 내용의 적절성’ 부분에서 문제 정의를 위한 교육 주제가 제한되어 제시되므로 학습자의 사고력 확장 범위 또한 제한될 수 있다는 의견이 있었다. 이를 수렴하여 주제의 범위를 확장하고, 자신의 경험에 따라 주제를 설정할 수 있도록 보강하였다. 최종 프로그램의 차시별 세부 설계 내용은 Table 6에 제시하였다.

4.4 실행 및 평가

‘실행’ 단계에서는 최종 교육 프로그램을 학교 현장에 적용하였다. 학생들은 6차시의 유니티 활용 관련 이론 및 실습 수업 이후, 이어 14차시의 문제해결과정 중심의 프로그래밍 교육 프로그램에 참여하였다. 수업 장면 및 교수-학습 자료 예시는 Fig. 3, Fig. 4과 같다. 마지막으로 ‘평가’ 단계에서는 교육 프로그램의 적용 전후에 대한 효과성 분석을 하였으며, 관련 내용은 ‘5. 연구 결과’에 정리하였다.



Figure 3. Scene of the class



Figure 4. Examples of Teaching and Learning Materials

분석 결과는 Table 7에 제시하였으며, 학생들의 컴퓨팅 사고력 및 학습 몰입도 모두 .05의 유의 수준에서 통계적으

Table 7. Results of paired-sample t-test (N=30)

Area	Factors	Test	Descriptive statistics		t	p	Cohen's d
			M	SD			
Computational Thinking	Problem recognition	pre post	36.03 40.30	4.582 6.979	-3.816	.001**	0.723
	Solution exploration	pre post	25.37 28.07	2.671 3.571	-7.724	.000***	0.856
	Programming	pre post	44.90 48.87	6.880 6.490	-3.680	.001**	0.593
Learning Flow	Balance of challenge and skill	pre post	13.20 14.67	2.657 2.758	-3.958	.000***	0.543
	Merging of action and awareness	pre post	17.33 18.23	2.496 2.334	-2.955	.006**	0.372
	Clear goals	pre post	6.20 7.03	1.375 1.217	-3.470	.002**	0.641
	Specific feedback	pre post	16.80 18.33	3.210 2.845	-3.857	.001**	0.504
	Concentration on the task	pre post	9.87 11.43	2.360 2.239	-5.395	.000***	0.678
	Sense of control	pre post	5.87 6.80	1.432 1.349	-3.247	.003**	0.670
	Loss of self-consciousness	pre post	15.53 16.77	1.548 2.459	-2.779	.009**	0.600
	Distortion of time perception	pre post	10.07 12.53	2.766 2.596	-6.298	.000***	0.917
	Autotelic experience	pre post	19.63 21.07	3.189 3.393	-2.733	.011*	0.437

로 유의한 차이를 확인하였다. Cohen's d 분석에서도 대다수 영역이 중간 이상 수준의 효과 크기를 보였다. 특히, 컴퓨팅사고력 효과 크기에서는 '해결방안 탐색' 요소에서의 문제분해, 추상화, 알고리즘과 절차 영역이, 학습 몰입 효과 크기에서는 '과제에 대한 집중' 및 '시간 감각의 왜곡' 요소에 대한 향상이 두드러졌음을 파악할 수 있었다. 이에 대한 해석을 '6. 결론'에서 제시하고자 한다.

5. 결론

본 연구에서는 컴퓨팅사고력의 구성 요소와 연계한 문제 해결과정을 중심으로, 몰입감 있는 학습 환경 조성을 위해 유니티를 교육 도구로 활용하는 일반계 고등학생 대상 프로그래밍 교육 프로그램을 개발하고 이를 적용하여 그 효과를 확인하고자 하였다. 연구 결과, 개발한 교육 프로그램이 일반계 고등학생의 컴퓨팅사고력 및 학습 몰입 모두에 긍정적인 영향을 미칠 수 있음을 확인하였다.

특히, 컴퓨팅사고력의 하위 요소 중에서도 문제분해·추상화·알고리즘 설계를 포함하는 '해결방안 탐색' 관련 영역에서 상대적으로 높은 효과 크기가 확인되었다. 이는 단순한 코드 작성 중심의 실습을 넘어 학습자 스스로 주제의 난이도와 구현 가능성을 고려하여 실현 가능한 수준에서 문제를 정의하고 해결 과정을 구체화하는 데 중점을 둔 교육 프로그램(7~12차시)을 통해 최적의 해결 흐름을 구상하는 능동적인 사고의 과정을 경험함으로써 해당 요소에 긍정적인 영향을 미친 것으로 해석된다.

또한, 유니티를 활용하여 게임 제작이라는 목표 아래 기능 구현을 위한 문제를 정의하고 이를 직접 설계·구현해 보는 경험이 학습자의 학습 몰입, 특히 과제에 대한 집중과 시간 감각의 왜곡 측면을 자극한 것으로 보인다. 이는 유니티 환경이 시각적 피드백과 실시간 상호작용을 제공함으로써, 학습자들이 과제에 몰입하고 결과물에 대한 성취감을 즉각적으로 경험할 수 있었기 때문으로 해석된다.

한편, 본 연구에서는 문제 정의 및 프로그래밍 과정에 생성형 AI를 활용할 수 있도록 하였으며, 이는 학습자가 아이디어를 구체화하고 주제의 난이도와 구현 가능성을 검토하는 과정에서 학습 보조 도구로 활용되었다. 또한 프로그램 구현 단계에서는 오류 탐지와 수정 과정을 효율적으로 지원함으로써 문제 해결의 절차적 완성도를 높이는 데 기여한 것으로 판단된다. 이러한 경험은 학습자의 사고 과정을 보조하고 디버깅 부담을 완화하여, 추상화와 자동화 중심의 컴퓨팅사고력을 강화하고 프로그래밍 학습의 효과성과 몰입을 향상시키는 데 일정 부분 기여하였을 것이라 추정된다. 또한 이러한 활용이 생성형 AI가 문제 해결을 대체하는 도구가 아닌, 문제해결과정 중심의 프로그래밍 학습을 보조하는 지원 도구로서 교육적 가능성을 지닌다는 점을 시사한다.

이와 같이 본 연구에서 제시한 교육 프로그램으로 고등학

교 프로그래밍 교육에 있어 학습자의 자발적인 참여와 몰입을 촉진하는 문제해결과정 중심 교수·학습 방법의 다양화에 긍정적인 영향을 미칠 수 있을 것이라 기대된다.

6. 연구의 한계 및 제언

본 연구는 다음과 같은 한계를 지닌다. 첫째, 본 연구는 프로그래밍의 기본 개념을 사전에 학습한 특정 지역 일반계 고등학생 30명만을 대상으로 수행되었으므로, 연구 결과를 일반화하는 데 어려움이 있을 수 있다. 둘째, 단일집단 사전-사후 설계를 적용함에 따라 비교집단을 통한 통제가 이루어지지 못하였으며, 이에 따라 성숙 효과나 외생 변인이 연구 결과에 영향을 미쳤을 가능성을 완전히 배제하기 어렵다는 내적 타당성의 한계가 존재한다. 셋째, 자기보고식 설문으로 효과를 측정하였으므로 실제 학생들의 수행 능력이나 산출물의 질적 변화를 객관적으로 파악하는 데 한계가 존재할 수 있다.

이러한 한계를 보완하기 위해 다음과 같은 후속 연구를 제안한다. 통제집단을 포함한 실험 설계를 적용하고 다양한 지역, 성별, 학년의 대규모 표본을 대상으로 확장 적용한 연구가 필요하다. 또한 자기보고식 설문 외에 컴퓨팅사고력을 객관적으로 평가 가능한 지필형 검사지를 활용하거나, 산출물 기반 평가, 질적 분석을 병행하여 이를 보강하는 연구가 필요하다.

더 나아가 본 연구에서 탐색적으로 활용한 생성형 AI의 교육적 가능성을 확장하기 위해, 향후 생성형 AI를 문제 정의에서 평가에 이르는 문제해결 전 과정의 학습 보조 도구로 활용할 수 있는 교수·학습 전략 및 교육 모델 개발에 대한 논의 및 연구가 요구된다.

참고문헌

- [1] Shute, V., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- [2] Ministry of Education. (2022). *2022 Revised Curriculum Overview*. Ministry of Education Notice No. 2022-33 [Appendix 4]. Sejong: Ministry of Education. <https://www.moe.go.kr/>
- [3] Lee, C. (2016). Development of a computational thinking based problem solving model(CT-PS Model) for software education. *Journal of Korean Practical Arts Education*, 22(3), 97-117. <https://www.kci.go.kr/kciportal/ci/sereArticleSearch/ciSereArtiView.kci?sereArticleSearchBean.artiId=ART002136776>
- [4] Jeon, Y. (2017). The development and application of a CT-CPS(computational thinking-based creative problem solving) instructional model for the software education of new curriculum [Doctoral Dissertation], Korea National

- University of Education.
- [5] Kim, J. (2024). Extracting the hierarchical learning elements for the “Algorithms and Programming” Area of the 2022 Revised Secondary Informatics Curriculum. *Journal of Internet Computing and Services*, 25(2), 123–132. <https://doi.org/10.7472/JKSII.2024.25.2.123>
- [6] Unity Technologies. (2025). Unity educational solution. Unity Technologies. <https://unity.com/kr/solutions/education>
- [7] Lee, H., Kim, Y., & Park, C. (2019). Development of quest-based mobile STEAM content for middle school science experiments. *Journal of the Korea Contents Association*, 19(2), 88–98. <https://doi.org/10.5392/JKCA.2019.19.02.088>
- [8] Yang, T., Hsu, Y., & Wu, Y. (2025). The effectiveness of ChatGPT in assisting high school students in programming learning: evidence from a quasi-experimental research. *Interactive Learning Environments*, 33(6), 3726–3743. <https://doi.org/10.1080/10494820.2025.2450659>
- [9] Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- [10] ISTE & CSTA. (2011). *Operational definition of computational thinking for k-12 education*.
- [11] Korea Education and Research Information Service (KERIS). (2015). *Development of software operation guidelines policy research (CR 2015-3)*. Ministry of Education & KERIS. <https://www.keris.or.kr/main/ad/pblcte/selectPblcteRRInfo.do?mi=1138&pblcteSeq=11563>
- [12] Choi, S. (2016). A study on teaching-learning for enhancing computational thinking skill in terms of problem solving. *The Journal of Korean Association of Computer Education*, 19(1), 53–62. <http://doi.org/10.32431/kace.2016.19.1.006>
- [13] Csikszentmihalyi, M. & Csikszentmihalyi, I. (1988). *Optimal experience: Psychological of flow in consciousness*. Cambridge University Press.
- [14] Min, S., & Na, S. (2007). The relationship between learning flow and classroom climate of vocational high school students. *Journal of Agricultural Education and Human Resource Development*, 39(3), 45–68.
- [15] Unity Technologies. (2025). *Unity AI features*. Unity. <https://unity.com/kr/features/ai>
- [16] Yoon, S. (2015). A study on Unity 3D engine in the educational perspective. *Proceedings of the Korean Society of Computer Information Conference*, 23(2), 246–249.
- [17] Kim, C., & Park, S. (2012). The raise the efficiency of game graphics design education using game engine: In focus of Unity3D and Torque. *Korean Society of Cartoon Animation Studies*, 29, 151–172. <https://doi.org/10.7230/koscas.2012.29.151>
- [18] Lee, K. (2019). A Study of spiderwebs via Unity physics engine. *School Science Journal*, 13(3), 261–270. <https://doi.org/10.15737/SSJ.13.3.201908.261>
- [19] Kim, M. (2018). Korean secondary students computational thinking based on problem solving. *The Journal of Learner-Centered Curriculum and Instruction*, 18(16), 807–830. <https://doi.org/10.22251/jlcci.2018.18.16.807>
- [20] Suk, I., & Kang, E. (2007). Development and validation of the learning flow scale. *Journal of Educational Technology*, 23(1), 121–154. <https://doi.org/10.17232/KSET.23.1.121>
- [21] Ministry of Education. (2022). *Practical (Technology / HomeEconomics) / Information Curriculum*. Ministry of Education Notice No. 2022-33 [Appendix 10]. Sejong: Ministry of Education. <https://www.moe.go.kr/>
- [22] Choi, S. (2016). A study on teaching the object oriented programming language. *The Journal of the Korea Institute of Electronic Communication Sciences*, 11(8), 751–758. <https://doi.org/10.13067/JKIECS.2016.11.8.751>
- [23] Kim, Y., Jang, Y., Yoon, I., Kim, J., & Lee, W. (2014). A case study on course game based elements for learning oibject-oriented concepts. *The Journal of Korean Association of Computer Education*, 17(5), 1–13. <https://doi.org/10.32431/kace.2014.17.5.001>
- [24] Lawshe, C. (1975). *A quantitative approach to content validity*. *Personnel Psychology*, 28(4), 563–575. <https://doi.org/10.1111/j.1744-6570.1975.tb01393.x>



유진솔

- 2019년 한국교원대학교 컴퓨터교육과(교육학사)
- 2025년 한국교원대학교 인공지능융합교육전공(교육학석사)
- 2025년~한국교원대학교 인공지능융합교육전공 박사과정

✚ 관심분야 : 컴퓨터교육, 인공지능융합교육, 정보 교육 과정, 유니티

✉ wlstf0090@knue.ac.kr



김귀훈

- 1998년 KAIST 전기및전자공학과(공학사)
- 2000년 KAIST 전자전산학과(공학석사)
- 2019년 KAIST 전기및전자공학부(Ph.D.)
- 2000년~2005년 LG 데이콤 주임연구원
- 2005년~2020년 ETRI 실장, 책임연구원
- 2020년~현재 한국교원대학교 인공지능융합교육 전공교수
- 2006년~현재 ITU-T SG11 Rapporteur, Editor
- 2022년~현재 한국시융합교육연구소 소장
- 2024년~현재 교육정보원 원장

✚ 관심분야 : 인공지능융합교육, AI/디지털교과서, 지능형 에지컴퓨팅, 강화학습

✉ kimkh@knue.ac.kr