



교수자의 Serious Game 설계 지원을 위한 BBPL 플랫폼 기능 비교 분석

A Comparative Analysis of BBPL Platform Features for Supporting Instructors in Designing Serious Game

오서진[†] · 이주영^{††} · 임예린^{††} · 신윤희^{†††}

Seojin Oh[†] · Jooyoung Lee^{††} · Yealin Im^{††} · Yoonhee Shin^{†††}

요약

본 연구는 GBL에서 교수자의 교수매체로 활용 가능한 Serious Game 설계를 지원하기 위해 Scratch, Entry, Code.org, App Inventor, Alice 등 블록 기반 프로그래밍 언어(BBPL) 플랫폼 5종을 비교 분석하였다. 교수자의 경험, 접근성, 교수 지원 측면에서 플랫폼별 기능의 가능성과 한계를 탐색하고, 분석 준거를 바탕으로 주요 기능을 비교하였다. 분석 결과, Scratch와 Code.org는 교수 지원이 우수하며, Entry는 커뮤니티 활용성과 자료 접근성이 뛰어났다. 반면 App Inventor와 Alice는 콘텐츠 제작에는 적합하나 교수자 지원과 협업 기능이 제한적이었다. 본 연구는 교수자의 플랫폼 선택과 활용 전략 수립에 기여하고, 향후 연수 프로그램 개발을 위한 기초 자료로 활용될 수 있다는 점에서 의의가 있다.

주제어 블록기반 프로그래밍 언어 플랫폼, 컴퓨팅 사고, 게임 기반 학습, Serious Game, 교수 설계

ABSTRACT

This study compared and analyzed five block-based programming language (BBPL) platforms, including Scratch, Entry, Code.org, App Inventor, and Alice, to support the design of Serious Games that can be used as a teaching medium by instructors in GBL. The study explored the possibilities and limitations of platform-specific functions in terms of instructor experience, accessibility, and teaching support, and compared key functions based on analysis criteria. The analysis showed that Scratch and Code.org offer strong teaching support, while Entry excels in community features and resource accessibility. App Inventor and Alice are suitable for content creation, but had limited instructional support and collaboration features. This study is significant in that it can inform instructors' platform selection and utilization strategies, and serves as a basis for the development of future professional development programs.

Keywords Block-Based Programming Language Platform, Computational Thinking, Game-Based Learning, Serious Game, Instructional Design

†정회원 한양대학교 대학원 교육공학과 석사과정
 ††정회원 한양대학교 대학원 교육공학과 박사과정
 †††정회원 한양대학교 사범대학 교육공학과 조교수 (교신저자)
 논문투고 2025년 03월 20일
 심사완료 2025년 05월 02일
 게재확정 2025년 05월 07일
 발행일자 2025년 05월 15일

1. 서론

컴퓨팅 사고력(CT; Computational Thinking)은 컴퓨터 과학의 원리를 바탕으로 문제를 분석하고 자동화 및 추상화를 통해 해결하는 사고 방식으로, 컴퓨터와 협력하는 능력을 강조하며 점점 중요성이 증가하고 있다[1, 2]. 2025년부터 시행될 『2022 개정 교육과정』에서는 소프트웨어와 인공지능 교육을 확대하고, 이를 과학, 수학, 인문 사회 등 다양한 분야와 융합하여 제공할 것을 목표로 하고 있다[3]. 이러한 변화는 창의적 문제 해결과 융합적 사고를 함양하여 미래 교육의 방향성을 반영하고 있으며, 컴퓨팅 사고력이 이러한 목표를 실현하는 핵심 역량임을 시사한다. 그러나 컴퓨팅 사고력 교육은 학습자들이 추상적인 개념을 이해하고 문제를 단계적으로 해결하는 알고리즘적 사고를 습득하는 데 어려움을 겪는다는 점에서 도전 과제가 있다[4, 5]. 이를 해결하기 위해 블록 기반 프로그래밍 언어(BBPL)가 도입되었으나, 주로 결과물 중심 학습에 치우쳐 학생들의 학습 동기와 지속성을 충분히 지원하지 못한다는 한계가 지적되고 있다[6]. 이러한 문제를 해결할 대안으로 게임 기반 학습(GBL; Game-based Learning)이 주목받고 있다. GBL은 학습 동기를 높이고 컴퓨팅 사고력을 촉진하는 데 효과적이며, 문제 해결 중심의 대화형 학습 환경을 제공한다[7].

특히 GBL의 한 형태인 Serious Game은 오락보다는 교육과 훈련을 목적으로 설계된 게임으로, 현실 문제 해결을 위한 실질적 연습과 상호작용을 통해 몰입형 학습 경험을 제공하며, 추상적 개념에 관한 구체적 이해와 컴퓨팅 사고력 향상에 중요한 도구로서 기여할 수 있다[8]. 그러나 교사들이 이를 설계하고 활용하는 데 기술적, 시간적 한계가 있어, 교육 현장에서 GBL을 효과적으로 활용하지 못한다[9]. 이를 극복하기 위해 교수자가 쉽게 활용할 수 있는 직관적 설계 도구와 충분한 교수 자료를 제공하는 BBPL 플랫폼 환경이 요구된다.

본 연구는 교수자가 Serious Game을 설계할 때 BBPL 플랫폼이 제공해야 할 주요 기능을 탐색하는 데 목적이 있다. 이를 위해 다음과 같은 연구 목표를 설정하였다. 첫째, BBPL 플랫폼과 관련된 선행 연구를 바탕으로 교수자가 Serious Game을 설계 및 운영하는 데 필요한 기능을 도출하고 체계적으로 분류한다. 둘째, 국내 교육에서 주로 활용되는 BBPL 플랫폼 5개(Scratch, Entry, Code.org, App Inventor, Alice)를 선정하여, 도출된 기능 준거를 기준으로 각 플랫폼이 해당 기능을 어떻게 지원하는지 비교 분석한다. 이를 통해 BBPL 플랫폼 개발 및 활용에 필요한 개선 방향을 제안하고, 교수자가 자신의 교육 목표에 적합한 플랫폼을 선택하는 데 참고할 수 있는 기준을 제공하고자 한다.

2. 이론적 배경

2.1 컴퓨팅 사고와 교육적 활용

컴퓨팅 사고는 창의성, 알고리즘적 사고, 비판적 사고,

문제 해결, 의사소통 및 협력 등을 포함하는 핵심 역량으로, 학습과 일상생활에서 문제를 체계적으로 해결하는 데 필수적이다[10]. 이는 알고리즘적 사고와 사회적 협력, 창의적 사고를 통합한 능력으로, 프로그래밍 교육은 이러한 컴퓨팅 사고를 바탕으로 문제 해결과 논리적 사고를 효과적으로 함양하는 방법으로 입증되었다[11, 12].

특히 프로그래밍 과제를 통해 학습자는 시퀀스, 반복, 조건부, 이벤트 처리와 같은 개념을 명시적으로 표현하며, 문제 분해, 디버깅, 패턴 인식과 같은 사고 과정을 연습하게 된다[13]. 그러나 어린 학습자들에게 프로그래밍 과제를 제공하기에는 복잡한 내용과 교수(teaching)의 어려움이 존재한다[14]. 이에 따라 컴퓨팅 사고를 보다 쉽게 가르칠 수 있는 도구와 교수법 탐색이 교육 현장의 중요한 과제로 부상하고 있다[15, 16].

또한 컴퓨팅 사고는 프로그래밍 교육을 넘어 초중등 교육의 다양한 과목에도 적용되고 있다. 예를 들어, Rodriguez-Martinez 등은 수학적 개념을 가르칠 때 컴퓨팅 사고를 적용하여 학습 효과를 증진한 사례를 제시하였다[17]. 이는 컴퓨팅 사고가 다양한 학문적 영역에서 문제 해결과 창의적 사고를 통해 학습 목표를 달성하는 데 중요한 역할을 한다는 점을 보여준다.

2.2 컴퓨팅 사고 교육과 BBPL 플랫폼

컴퓨팅 사고를 촉진하기 위해 연구자들은 시각화 도구, 교육용 로봇, 비주얼 프로그래밍 언어 등 다양한 디지털 도구를 개발하여 교육 현장에 도입해 왔다[18]. Garneli 등은 K-12 컴퓨팅 사고 교육을 다룬 연구를 검토하며, 다양한 유형의 프로그래밍 도구가 학생들의 참여와 컴퓨팅 사고 촉진에 효과적임을 확인하였다[19]. 특히, 블록 기반 프로그래밍 언어(BBPL) 플랫폼은 초보자들에게 긍정적인 학습 경험을 제공하며 K-12 교육 현장에서 널리 활용되고 있다[20, 21].

BBPL 플랫폼은 Fig. 1과 같이 프로그래밍 명령어를 퍼즐 조각처럼 시각적으로 표현해 초보자가 복잡한 프로그래밍 개념을 쉽게 이해하도록 돕는다[22]. 이 접근 방식은 컴퓨터 과학 분야(예: 메타버스, 인공지능 등)로의 진입을 지원하며, 로봇 코딩[23], 데이터 과학[24], 모바일 앱 개발[25] 등 다양한 분야에서 활용된다. 또한, BBPL 플랫폼은 교사에게도 간단한 콘텐츠 제작과 수업 활용을 지원하는 도구로 평가받는다[17].

허균은 국내 코딩 교육 프로그램 연구를 분석하며, Scratch, Entry, App Inventor, Code.org, Alice가 대표적인 BBPL 플랫폼으로 채택되고 있음을 확인하였다[26]. BBPL 플랫폼은 직관적인 인터페이스와 단계별 학습 지원을 통해 초보자에게는 기초 학습을, 전문가에게는 심화 학습과 창의적 프로젝트 개발을 지원한다. 이처럼 BBPL 플랫폼은 컴퓨팅 사고 교육의 접근성을 높이며, 학습자와 교사의 창의적 역량을 지원하는 핵심 도구로 자리 잡고 있다.



Figure 1. Block coding in Scratch

2.3 게임 기반 학습(GBL)과 Serious Game의 교육적 효과

초보자나 저연령 학습자를 위한 컴퓨팅 사고 교육에는 다양한 접근 방식이 있으며, 그중 하나는 게임 기반 학습이다. 게임 기반 학습(GBL)은 학습 동기를 높이고 문제 해결 과정에 몰입을 유도하며, 학습 성과를 향상시키는 데 효과적이다[27]. 또한, GBL을 교실에 효과적으로 통합하기 위해 교사가 적절한 지식과 기술을 갖추고 이를 지원하는 도구와 지침이 필요하다는 연구가 제시되었다[28, 29].

Serious Game은 일반적인 GBL보다 학습 목표가 명확하고 설계가 체계적이다. 또한 학습자의 능동적 지식 구성과 고차적 사고를 촉진하는 데 초점을 둔다[30, 31] Fig. 2와 같은 Serious Game은 프로그래밍 교육에서 학습자의 흥미와 동기를 유발하고, 컴퓨팅 사고 및 프로그래밍 기술 학습을 지원한다. 또한 보상, 즉각적인 피드백, 점진적 도전 등 게임 요소를 통해 학습 동기를 더욱 강화한다[32]. Malliarakis 등은 Serious Game이 컴퓨팅 사고와 프로그래밍 학습에 효과적이라고 강조하며, 다양한 교육적 성과 사례가 보고되고 있다고 언급했다[33]. 특히 교사가 설계 과정에 참여할 경우, 학생 요구를 반영한 효과적인 학습 목표 구현이 가능하다[34]. 그러나 일부 교사들은 게임 설계 역량 부족으로 어려움을 겪고 있으며[35], 이를 극복하기 위한 설계적 및 교수적 도구가 필요하다[36].

Vahldick 등은 Serious Game 설계와 활용을 위한 가이드라인을 제시하며, BBPL 플랫폼을 활용한 연구를 통해 다섯 가지 핵심 요소를 도출했다[37]: 대상 연령층 고려[38], 심미성 조작 가능성[38], 소스 코드 오픈 가능성[39], 커뮤니티 기반 동료 지원[40], 교수 지원 자료 제공[41]. 이러한 요소들은 교사가 BBPL 플랫폼을 활용해 Serious Game을 설계할 때 실질적인 지침이 될 수 있다.



Figure 2. Serious game for programming (Run Marco)

3. 연구 방법 및 절차

3.1 BBPL 플랫폼 기능 분석을 위한 준거

본 연구는 국내 교육에서 활용되는 BBPL 플랫폼의 Serious Game 설계 지원 기능을 분석하고, 이를 통해 교수자가 BBPL 플랫폼을 활용하여 적극적으로 Serious Game을 설계할 수 있도록 지원하는 데 목적이 있다. 이를 위해 플랫폼 기능을 다룬 선행 연구를 기반으로 분석 준거를 마련하였다. 이상민 & 김승현의 메타버스 플랫폼 교육 활용 연구[42], 김수환 등의 초등 인공지능 교육 플랫폼 분석 연구[43], 이재호 & 이승훈의 인공지능 교육 플랫폼 전문가 인식 조사[44], Kraveva 등의 BBPL 교육적 활용 연구[45]를 비교하여 공통적으로 나타난 기능을 도출하였다. 이를 바탕으로 사용자 경험 지원, 플랫폼 접근성 및 사용성, 교수 지원의 세 가지 대분류와 해당 세부 기능을 정리한 결과는 Table 1과 같다.

Table 1. Standard for analysis

Category	Subcategory	Function
User Experience Support	Target User	Age
	Reward System	Reward System
	Remix/ Remake	Project Remake
	Personalization	Template and Character control
		Sound Use
Accessibility and Usability	Tools for Painting and Drawing	
	Community Support	Community
Instructional Support	Development Content	Games
	Educational Material	Video Tutorials
		Text Tutorials
	Learner Management	Teacher Account
		Feedback System
	Class Management	Curriculum
		Open Classes

Serious Game 설계 지원 기능의 세부 요소는 다음과 같다. 첫째, 사용자 경험 지원 측면에서는 보상 체계, 프로젝트 리믹스 및 리메이크, 개인화 기능이 포함된다. 둘째, 플랫폼 접근성 및 사용성 측면에서는 커뮤니티 기능, 게임 제작 및 공유 기능이 요구된다. 셋째, 교수 지원 측면에서는 교육 자료 제공, 교사용 계정, 피드백 시스템, 학급 개설 및 관리 기능이 필요하다.

3.2 분석대상 선정

본 연구는 BBPL 환경을 지원하는 5개의 플랫폼을 분석 대상으로 선정하였다. 이를 위해 연구진은 교육 현장에서 활용되는 BBPL 플랫폼과 관련된 선행 연구를 탐색하였다. 연구 목적에 따라 Serious Game 개발이 가능하며 BBPL 환

경을 지원하는 플랫폼을 중심으로 최종 대상을 선정하였다.

현재 교육 현장에서는 Scratch, Code.org, Kodable, Sphero, App Inventor 등 다양한 플랫폼이 활용되고 있으나, 블록 기반 외의 프로그래밍 언어를 지원하거나 교수자가 플랫폼에서 Serious Game을 직접 개발할 수 없는 경우는 제외하였다. 국내 BBPL 플랫폼 관련 선행 연구 중 허균[26]은 국내 교육 현장의 최근 흐름을 반영한 최신 연구로, 36편의 코딩 교육 프로그램 연구를 분석하며 Scratch, SAP Fiori, Python, Entry 등을 소개하였다. 이 중 텍스트 기반 언어를 지원하는 SAP Fiori와 Python을 제외하고, 초·중등 교육에서 널리 활용되는 Scratch, Entry, Code.org, App Inventor, Alice를 분석 대상으로 선정하였다.

3.3 연구 절차

BBPL 플랫폼 내 기능 분석은 최종 선택한 Scratch, Entry, Code.org, App Inventor, Alice를 연구자 3인이 1개월 동안 독립적으로 플랫폼을 실행하여 플랫폼 비교 분석을 위해 마련한 준거를 기준으로 각 플랫폼의 기능 분석 결과표를 작성했다. 분석 과정에서 플랫폼 내 모호한 기능은 연구자 간 논의를 통해 해당 기능을 재확인 후 합의 과정을 거쳤다. 최종적으로 도출된 기능이 맞는지 확인하기 위해 개별적으로 각각의 플랫폼을 활용하며 올바르게 분석이 되었는지 재확인 과정을 거쳤다.

4. 연구 결과

4.1 BBPL 플랫폼별 분석

4.1.1 Scratch 분석 결과

Scratch는 MIT 미디어 랩에서 개발된 블록 기반 프로그래밍 언어(BBPL) 플랫폼으로, 주로 8~16세 대상의 프로그래밍 교육에 활용된다. 이 플랫폼은 공식 웹사이트를 통해 쉽게 접속할 수 있으며, 사용자는 프로젝트를 직접 제작하거나 타인의 프로젝트를 리메이크하여 공유할 수 있다. Scratch의 보상 체계는 리메이크 시 원본 제작자에게 크레딧을 제공하는 구조로, 금전적 가치는 없지만 높은 크레딧 보유자는 자신의 프로젝트가 널리 활용되고 있음을 인식할 수 있다. 이러한 비물질적 보상 요소는 학습자의 동기를 유발하는 데 효과적이다[46]. Scratch는 Sprite라는 캐릭터 중심의 제작 환경을 제공하며, 사용자는 Sprite의 위치, 방향, 크기 조정 및 효과음 삽입이 가능하다. 라이브러리 기능을 통해 음악 삽입이 가능하며, 그리기 도구를 활용하여 다양한 창작 프로젝트를 제작할 수 있다. 변수, 조건문 등의 코딩 블록을 사용해 점수 및 레벨 시스템을 포함한 게임 제작도 가능하다.

컴퓨터교육 측면에서 Scratch는 프로그래밍의 기초 개념을 시각적으로 익힐 수 있도록 지원하고, 창의적 표현과

문제해결력, 컴퓨팅 사고력 함양에 효과적인 장점을 지닌다. 반면, 실시간 협업 기능 부재 및 텍스트 기반 언어와의 연계 한계는 심화 학습으로의 확장에 제약이 될 수 있다.

교수 지원 기능으로는 공식 웹사이트와 YouTube 채널에서 제공되는 튜토리얼 영상과 자료가 있어 활용이 용이하며, 교사용 계정 생성 기능을 통해 학습자 관리 및 개별 학습 상황 확인, 피드백 제공이 가능하다. 또한, Scratch는 분석 대상 플랫폼 중 가장 많은 이용자와 사례를 보유하고 있어, 다양한 수준별 커리큘럼 자료를 쉽게 검색하고 활용할 수 있는 강점을 지닌다[47]. Scratch에서 제공하는 교육 자료는 Fig. 3과 같다.



Figure 3. Educational resources offered by Scratch

4.1.2 Entry 분석 결과

Entry는 한국의 커넥트재단에서 개발된 블록 기반 프로그래밍 언어(BBPL) 플랫폼으로, 초등학생부터 성인까지 쉽게 이용할 수 있도록 설계되었다. 웹사이트를 통해 무료로 접근 가능하며, 사용자는 프로젝트를 직접 제작하거나 타인의 프로젝트를 리믹스 및 공유할 수 있다. Entry는 보상 체계를 제공하지 않지만, 리믹스 기능을 통해 학습자 간 협업과 창의적 아이디어 공유를 장려한다. 그리고 다양한 캐릭터와 오브젝트 라이브러리를 제공하며, 사용자는 이들의 위치, 방향, 크기를 자유롭게 조절할 수 있다. 또한 배경 및 소리 추가, 드로잉 도구를 활용한 창작이 가능하여 프로젝트의 완성도를 높인다[48].

컴퓨터교육 측면에서 Entry는 실시간 커뮤니티 기능과 직관적인 한글 기반 인터페이스를 바탕으로 초등 및 중등 수준의 학생들에게 접근성과 사용성을 높이는 강점을 지닌다. 특히 실시간 채팅을 통한 협업 기능은 학습자 간 상호작용과 공동 문제 해결 경험을 촉진한다. 반면, 보상 체계의 부재와 제한적인 고급 기능은 고학년 혹은 중급 이상의 학습자에게는 다소 단조롭게 느껴질 수 있으며, 텍스트 기반 프로그래밍으로의 확장에는 제약이 따른다.

교수 지원 기능으로는 공식 웹사이트의 튜토리얼 영상과 텍스트 자료 제공, 교사용 계정을 활용한 학습자 관리 및 피드백 제공이 포함된다. 특히 학습 상황을 실시간으로 관찰할 수 있어 효과적인 수업 운영을 지원한다. 한국어 기반의 플랫폼으로, 국내 교육 환경에 최적화된 풍부한 커리큘럼 자료와 수준별 학습 예제를 제공하여 교육 활용도가 높다[49]. Entry의 실시간 온라인 커뮤니티 서비스는 Fig. 4와 같다.



Figure 4. Entry's live online community service

4.1.3 Code.org 분석 결과

Code.org는 K-12 학생을 대상으로 한 블록 기반 프로그래밍 언어(BBPL) 플랫폼으로, 무료로 접근 가능하며 사용자는 프로젝트를 제작하거나 리믹스 기능을 통해 타인의 프로젝트를 수정 및 공유할 수 있다. 이 플랫폼은 학습 동기를 강화하기 위해 Hour of Code 활동 완료 시 인증서를 발급하는 보상 체계를 제공하여 성취감을 부여한다. Code.org는 아바타 꾸미기, Sound Lab을 활용한 음악 삽입, 그림 그리기 도구 등 다양한 창작 도구를 통해 학습자의 창의적 표현을 지원한다.

컴퓨터교육 측면에서 Code.org는 단계적이고 체계적인 커리큘럼을 기반으로 컴퓨팅 사고력과 문제 해결 능력을 점진적으로 신장시키는 데 효과적인 플랫폼이다. 특히 STEM 융합형 콘텐츠를 활용한 교육은 과학·기술·수학 등의 타 교과와의 통합 교육을 가능하게 하며, 초등부터 고등까지 폭넓은 학습 수준을 아우를 수 있다는 점에서 교육적 확장성이 높다. 반면, 실시간 협업 기능이나 커뮤니티 기반의 상호작용 기능은 상대적으로 제한적이어서, 협력학습 중심의 수업에는 다소 부적합할 수 있다. 또한 창의적 프로젝트 활동보다는 개별 학습 중심으로 구성되어 있어, 자율성과 탐구 중심 수업을 중시하는 환경에서는 제약이 될 수 있다.

교수 지원 기능은 Code.org의 주요 강점으로, 비디오 및 단계별 텍스트 튜토리얼을 통해 학습자가 체계적으로 학습할 수 있도록 돕는다. STEM 연계 과목 커리큘럼과 유명 애니메이션 기반 교육 자료를 제공하여 다양한 학습 콘텐츠 제작을 지원한다[50]. 교수자는 교사용 계정을 생성해 학급을 운영하고 학생의 학습 진도를 관리하며, 댓글을 통해 피드백을 제공할 수 있다. 또한, 수준별 커리큘럼으로 학습 지도와 학급 운영을 효과적으로 지원한다. Code.org에서 제공하는 다양한 커리큘럼은 Fig. 5와 같다.

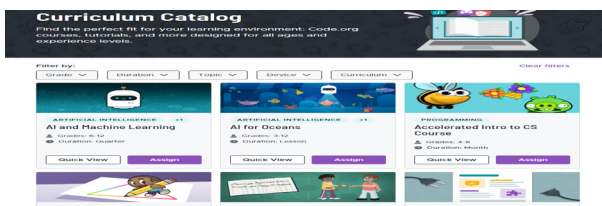


Figure 5. The variety of curriculum on Code.org

4.1.4 App Inventor 분석 결과

App Inventor는 모든 연령층을 대상으로 프로그래밍 교육을 지원하는 블록 기반 프로그래밍 언어(BBPL) 플랫폼이다. 공식 웹사이트를 통해 모바일 콘텐츠 제작에 특화된 도구를 활용하여 애플리케이션을 쉽게 개발할 수 있지만[51], 리믹스 기능은 지원되지 않아 프로젝트 파일을 다운로드 후 수정해야 하며, 보상 체계도 마련되어 있지 않아 학습 동기 부여 측면에서 한계가 있다. 또한 이미지 업로드, 음악 및 미디어 파일 삽입, 소리 조절 기능을 제공하며, 캔버스를 활용해 직접 그림을 그릴 수 있어 다양한 멀티미디어 요소를 활용한 애플리케이션 제작이 가능하다. 특히, 스마트폰과 태블릿 환경에서 실행 가능한 모바일 콘텐츠 제작은 App Inventor의 주요 강점이다.

컴퓨터교육 측면에서 App Inventor는 실생활 문제 해결 중심의 모바일 애플리케이션 개발 활동을 통해 학습자의 논리적 사고력과 응용 능력을 키울 수 있는 장점이 있다. 특히 학습 결과물이 실제 스마트폰에서 작동한다는 점은 학습자의 성취감과 몰입도를 높이는 데 효과적이다. 반면, 블록 코딩 환경이지만 인터페이스가 비교적 복잡하고, 리믹스나 협업 기능이 부족하여 초급 학습자에게는 진입 장벽이 될 수 있다. 또한, 컴퓨터과학 개념 학습보다는 앱 개발 기능에 초점이 맞춰져 있어, 알고리즘적 사고나 컴퓨팅 사고력 중심의 교육 목표와는 방향성이 다소 어긋날 수 있다.

교수 지원 기능은 제한적이며, 비디오 및 PDF 형식의 튜토리얼 자료는 제공되나, 교사용 계정이 없어 학습자 관리와 피드백 제공이 어렵다. 다만, 교사용 커리큘럼을 통해 교수자는 이를 수업 계획에 활용할 수 있다. App Inventor의 앱 제작 화면은 Fig. 6과 같다.

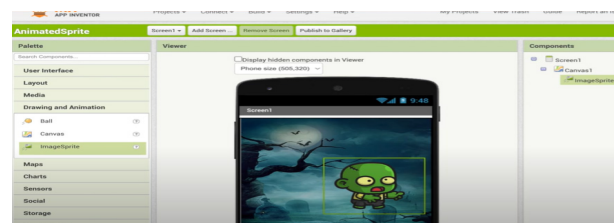


Figure 6. App Inventor for app development

4.1.5 Alice 분석 결과

Alice는 초등학교 고학년부터 중학생 이상을 대상으로 프로그래밍 교육을 지원하는 블록 기반 프로그래밍 언어(BBPL) 플랫폼이다. 프로그램 설치 방식으로 사용되며, 3D 콘텐츠 제작에 특화되어 시각적이고 몰입감 있는 프로젝트 개발이 가능하다[52]. 그러나 리믹스 기능은 공식적으로 지원되지 않아 프로젝트 파일을 다운로드 후 수정해야 하는 제한점이 있으며, 보상 체계도 마련되어 있지 않아 학습 동기 부여 요소가 부족하다. 그리고 카메라 위치 조정, 객체 크기 및 속성 변경, 음악 라이브러리 등을 통해

3D 환경에서 시뮬레이션과 애니메이션 제작을 지원하지 않, 그리기 기능은 제공하지 않는다. 공식 포럼 사이트 링크는 제공되나 접속이 불가해 사용자 간 커뮤니티 활용이 제한적이다.

컴퓨터교육 측면에서 Alice는 스토리텔링 기반의 3D 시뮬레이션을 활용함으로써 학습자의 시각적 사고력과 창의적 문제 해결 능력을 촉진하는 데 효과적이다. 특히 객체 중심 프로그래밍과 애니메이션 구성 등은 고차원적 컴퓨팅 개념의 이해를 지원하며, STEM 융합 교육에 유용하게 적용될 수 있다. 반면, 플랫폼의 인터페이스가 비교적 복잡하고 진입 장벽이 높아 프로그래밍 초급자에게는 학습 동기를 저하시킬 수 있으며, 협력 학습이나 피드백 제공을 위한 시스템도 부족하여 수업 내 즉각적인 상호작용 설계에는 제약이 있다. 또한 설치형 소프트웨어 특성상 접근성이 제한될 수 있다.

교수 지원 기능은 제한적이며, 공식 웹사이트에서 비디오 및 PDF 튜토리얼 자료를 제공하지만, 교사용 계정이 없어 학습자 관리와 피드백 제공이 어렵다. 다만, 교사용 커리큘럼이 제공되어 교수자는 이를 수업에 활용할 수 있다. Alice의 3D 콘텐츠 제작 화면은 Fig. 7과 같다.

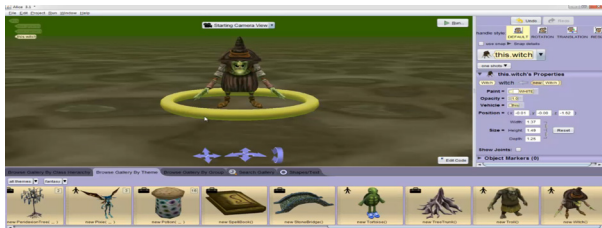


Figure 7. Alice for 3D contents development

4.2 BBPL 플랫폼 내 Serious Game 설계 지원 기능

교수자의 Serious Game 설계 지원 관점에서의 BBPL 플랫폼을 분석한 전체 결과는 Table 2와 같다.

Table 2. Analysis result on BBPL platform

Category	Function	BBPL Platform				
		S	E	C	AI	AL
User Experience Support	Age	8+	8+	K-12	8+	13+
	Reward System	O	X	O	X	X
	Project Remake	O	O	O	△	△
	Template and Character control	O	O	O	O	O
	Sound Use	O	O	O	O	O
	Tools for Painting and Drawing	O	O	O	O	X
Accessibility and Usability	Community	O	O	O	O	X
	Games	O	O	O	O	O

Instructional Support	Video Tutorials	O	O	O	O	O
	Text Tutorials	O	O	O	O	O
	Teacher Account	O	O	O	X	X
	Feedback System	O	O	O	X	X
	Curriculum	O	O	O	O	O
	Open Classes	O	O	O	X	X

Note: S = Scratch, E = Entry, C = Code.org, AI = App Inventor, AL = Alice

4.2.1 Scratch 내 Serious Game 설계 지원 기능

Scratch는 Serious Game 설계를 지원하는 다양한 기능을 포함하며, 특히 보상 체계와 교수 지원에서 강점을 보인다. 교수자는 학습자의 성취도에 따라 보상을 제공하고, 리메이크 가능한 콘텐츠를 활용할 수 있다. 활발한 커뮤니티는 질의응답과 협력 학습을 유도하며, 수준별 학습자와 다양한 커리큘럼에 적합한 자료도 제공한다[53]. 다만, 보상으로 제공되는 크레딧은 금전적 가치를 지니지 않아 일부 학습자의 지속적인 동기 유발에는 한계가 있으며, 실시간 협업 기능도 부족해 다인용 게임 설계 시 추가 작업이 필요하다.

결론적으로 Scratch는 직관적인 설계 도구와 풍부한 리소스를 바탕으로 초보 학습자와 교수자 모두에게 적합한 플랫폼이다. 창의적이고 몰입감 있는 게임 설계가 가능하지만, 일부 기능은 교수자의 전략적 보완이 요구된다. Scratch 내 Serious Game 설계 지원 기능 분석 결과표는 Table 3과 같다.

Table 3. Analysis of serious game design support functions in Scratch

Category	Function	Scratch
User Experience Support	Age	8+
	Reward System	Provide "Credit" to the original author when others remake the project
	Project Remake	Can Be Shared and Remixed on The Website
	Template and Character control	Control Sprite (character) Position and Resize, etc
	Sound Use	Allow users to embed their own music
	Tools for Painting and Drawing	Lines, Shapes, Fills, Erasers, and More
Accessibility and Usability	Community	Has the most users and the most active community
	Games	2D Game
Instructional Support	Video Tutorials	Materials from The Official Website or YouTube
	Text Tutorials	From The Website Archive
	Teacher Account	Available
	Feedback System	Comments and "Likes"
	Curriculum	ScratchED Offers Curriculum by Level and Type
	Open Classes	Available

4.2.2 Entry 내 Serious Game 설계 지원 기능

Entry는 실시간 커뮤니티와 한국 교육 환경에 최적화된 사용자 친화적 기능에서 강점을 보인다. 교수자는 한국어 기반 인터페이스와 자료를 활용해 학습자 수준에 맞는 콘텐츠를 설계하고, 실시간 커뮤니티를 통해 상호작용을 강화할 수 있다. 실시간 질의응답이 가능한 커뮤니티 환경은 협력 학습을 유도하며, 다양한 캐릭터·배경·소리 라이브러리는 창의적이고 몰입감 있는 게임 설계를 지원한다. 반면, Scratch에 비해 심미성 조작 기능이 제한되고, 그래픽 요소의 세부 조정과 맞춤화에 제약이 있다. 또한 보상 체계 부재로 학습자의 지속적 동기 유발이 어렵고, 일부 고급 학습자에게는 기능이 단순하게 느껴질 수 있다. 복잡한 프로젝트 설계에는 추가 도구가 필요할 수 있다.

결론적으로 Entry는 초보 학습자와 교수자 모두에게 적합한 플랫폼으로, 실시간 커뮤니티와 풍부한 자료를 통해 학습 동기와 협력 학습을 촉진한다. 다만, 심화 학습이나 고급 설계를 위해선 기능 보완이 요구된다. Entry 내 Serious Game 설계 지원 기능 분석 결과표는 Table 4와 같다.

Table 4. Analysis of serious game design support functions in Entry

Category	Function	Entry
User Experience Support	Age	8+
	Reward System	Unavailable
	Project Remake	Can Be Shared and Remixed on The Website
	Template and Character control	Allow Users to Create Their Own Avatars
	Sound Use	Allow Users to Embed Their Own Music
	Tools for Painting and Drawing	Lines, Shapes, Fills, Erasers, and More
Accessibility and Usability	Community	Provide Community Services with Live Chat and Online Collaboration
	Games	2D Game
Instructional Support	Video Tutorials	Materials from YouTube
	Text Tutorials	Materials From The Website Archive
	Teacher Account	Available
	Feedback System	Comments and "Likes"
	Curriculum	Provide Textbook Labs
	Open Classes	Available

4.2.3 Code.org 내 Serious Game 설계 지원 기능

Code.org는 대부분의 기능을 포함하나, 특히 체계적인 커리큘럼과 보상 체계에서 강점을 보인다. K-12 학습자를 대상으로 설계된 이 플랫폼은 단계적인 학습 과정을 통해 프로그래밍 개념을 익힐 수 있도록 지원한다. 또한, STEM 연계와 애니메이션 기반 프로젝트를 제공하여 체계적인 목표 달성을 돕고, 인증서 제공을 통해 학습자의 동기를 유

발한다. 그러나 Code.org는 커뮤니티 기능이 비교적 약해 협력 학습보다는 개별 학습에 중점을 두고 있다. 따라서 협력적 프로젝트 설계나 실시간 상호작용이 필요한 수업에서는 추가적인 도구나 전략이 필요하다. Code.org 내 Serious Game 설계 지원 기능 분석 결과표는 Table 5와 같다.

Table 5. Analysis of serious game design support functions in Code.org

Category	Function	Code.org
User Experience Support	Age	K-12
	Reward System	Create and Award Certificates to Users Who Achieve Goals
	Project Remake	Can Be Shared and Remixed on The Website
	Template and Character control	Avatar customization is available
	Sound Use	Allow Users to Embed Their Own Music
	Tools for Painting and Drawing	Lines, Shapes, Fills, Erasers, and More
Accessibility and Usability	Community	Asynchronous Community Support
	Games	2D Game
Instructional Support	Video Tutorials	Materials From The Website Archive
	Text Tutorials	Step-by-step tutorials for each project
	Teacher Account	Available
	Feedback System	Comments and "Likes"
	Curriculum	Diverse Curriculum Based on STEM Education, Animation, Film, and More
	Open Classes	Available

결론적으로, Code.org는 체계적이고 구조화된 학습 환경을 제공하는 플랫폼으로, Serious Game 설계와 운영에 적합하다. 단계적 학습 과정과 다양한 커리큘럼 제공은 교수자와 초보 학습자 모두에게 유익하지만, 창의적 설계나 협력학습을 강조하는 수업에는 보완이 요구된다.

4.2.4 App Inventor 내 Serious Game 설계 지원 기능

App Inventor는 모바일 애플리케이션 개발에 특화된 블록 기반 프로그래밍 언어(BBPL) 플랫폼으로, 모바일 중심의 Serious Game 설계를 위한 독창적인 기능을 제공한다. 이 플랫폼은 프로그래밍 학습을 실생활 앱 제작과 연계해 학습자가 실질적 결과물을 경험할 수 있도록 하며, 사용자는 블록 코딩으로 UI와 동작을 설계하고 점수, 레벨, 사용자 입력 반응 등 다양한 게임 요소를 구현할 수 있다 [54]. 그러나 앱 개발에 초점이 맞춰져 있어 몰입형 게임이나 복잡한 그래픽 기반 게임 설계에는 한계가 있으며, 심미성 조작과 캐릭터 창작 기능이 Scratch나 Alice에 비해 부족하다. 또한, 리메이크와 협력학습 기능도 제한적이다.

결론적으로 App Inventor는 실생활 앱 제작에 적합한 플랫폼으로, 학습자의 동기와 성취감을 높이는 데 효과적이나, 몰입형 게임 설계나 협력 수업에는 추가 전략과 도구가 필요하다. App Inventor 내 Serious Game 설계 지원 기능 분석 결과표는 Table 6과 같다.

Table 6. Analysis of serious game design support functions in App Inventor

Category	Function	App Inventor
User Experience Support	Age	8+
	Reward System	Unavailable
	Project Remake	Available but Limited
	Template and Character control	Upload An Image of the User's Choice
	Sound Use	Allow Users to Embed Their Own Music
	Tools for Painting and Drawing	Can Draw With Canvas Components
Accessibility and Usability	Community	Asynchronous Community Support
	Games	Can create Mobile App-Based Games
Instructional Support	Video Tutorials	Materials From The Website Archive
	Text Tutorials	Materials From The Website Archive
	Teacher Account	Unavailable
	Feedback System	Unavailable
	Curriculum	Provide curriculum for teachers
	Open Classes	Unavailable

4.2.5 Alice 내 Serious Game 설계 지원 기능

Alice는 3D 환경 기반 블록 프로그래밍 언어(BBPL) 플랫폼으로, 몰입감 있는 Serious Game 설계에 강점을 지닌다. 스토리텔링과 시각적 표현 중심의 설계를 통해 학습자가 창의적이고 상호작용적인 게임을 개발할 수 있으며, 3D 캐릭터와 환경을 활용해 움직임, 크기, 회전 등을 제어하고 배경음악과 효과음을 삽입할 수 있다[55]. 초보 학습자는 블록 기반 인터페이스로 프로그래밍 논리를 익히고, 3D 설계를 통해 창의적 문제 해결 능력을 기를 수 있다.

다만, 복잡한 인터페이스는 초보자에게 부담이 될 수 있고, 3D 콘텐츠 설계는 2D 플랫폼보다 시간이 더 소요된다. 또한, 리메이크 및 협력 기능이 부족해 협력 설계를 촉진하기 어렵고, 커리큘럼 자료도 제한적이며 설치가 필요해 데스크톱 환경에만 적합하다.

결론적으로 Alice는 3D 환경과 스토리텔링 기반 설계를 통해 창의적이고 몰입감 있는 Serious Game 개발에 적합하지만, 복잡한 인터페이스와 협력 기능 부족은 보완이 필요하다. Alice 내 Serious Game 설계 지원 기능 분석 결과표는 Table 7과 같다.

Table 7. Analysis of serious game design support functions in Alice

Category	Function	Alice
User Experience Support	Age	13+
	Reward System	Unavailable
	Project Remake	Available but Limited
	Template and Character control	Change Camera View, Size or Properties of 3D Objects
	Sound Use	Allow Users to Embed Their Own Music
	Tools for Painting and Drawing	Unavailable
Accessibility and Usability	Community	Unavailable
	Games	3D Games
Instructional Support	Video Tutorials	Materials From The Website Archive
	Text Tutorials	Materials From The Website Archive
	Teacher Account	Unavailable
	Feedback System	Unavailable
	Curriculum	Provide curriculum for teachers
	Open Classes	Unavailable

5. 논의 및 결론

본 연구는 BBPL 플랫폼이 제공하는 사용자 경험 지원, 접근성 및 사용성, 교수 지원의 세 가지 측면에서 교수자의 Serious Game 설계 지원 가능성을 탐구하였다. 이를 기반으로 다음과 같은 시사점을 도출하였다. 첫째, 사용자 경험 지원 측면에서 대부분의 플랫폼은 창의적 활동을 촉진하는 다양한 기능을 제공하지만, 보상 체계와 리메이크 기능의 지원 범위에는 차이가 있다. Scratch와 Code.org는 보상 체계를 통해 학습 동기를 강화하며, 특히 Scratch는 리메이크 시 창작자에게 크레딧을 제공해 학습자의 창의적 참여를 유도한다. 반면, Entry, App Inventor, Alice는 보상 체계가 없어 지속적인 동기 유발에는 한계가 있다. 이에 교수자는 각 플랫폼의 강점을 활용한 동기 유발 전략을 수업 설계에 포함해야 한다. 예를 들어, Scratch에서는 리메이크 기능과 크레딧 제공을 활용해 학생들이 작품을 탐구하고 확장하도록 유도하고, Code.org는 단계별 목표마다 인증서를 보상으로 제공해 성취감을 높일 수 있다. 보상 체계가 없는 플랫폼에서는 Entry의 실시간 협업 기능을 활용한 팀 프로젝트와 교수자 주도의 보상, App Inventor의 실생활 문제 해결 앱 개발 후 발표 및 보상 제공, Alice의 3D 콘텐츠 발표와 동료 평가를 통한 보상 방식을 고려할 수 있다. 둘째, 접근성 및 사용성 측면에서는 커뮤니티 서비스와 게임 개발 기능이 플랫폼 간에 차별적으로 제공된다. Entry는 실시간 온라인 커뮤니티로 학습자 간 상호작용과 피드백을 촉진할 수 있는 환경을 지원하며, Alice는 3D 콘텐츠와 게임 제작을 통해 몰입형 학습 환경 설계에 적합하지만, 커뮤니티 접근성이 제한적이다. 따라서 협력 학습에는 Entry를, 심화 학습이나 고급 프로그래밍에는 Alice를 학습 목표와 연계해 활용해야

한다. 셋째, 교수 지원 측면에서 Scratch와 Code.org는 풍부한 교수 자료와 교사용 계정을 제공해 수업 설계와 학습자 관리에 유리하다. Code.org는 STEM 연계 커리큘럼과 애니메이션 기반 학습 자료를 제공하며, Scratch는 ScratchED 사이트에서 수준별 가이드를 제공해 교수자와 학습자가 쉽게 활용할 수 있다. 반면, App Inventor와 Alice는 기본적인 교수 자료만 제공하며, 학습자 관리 기능이 부족해 교수자가 외부 도구나 자료로 보완할 필요가 있다. 결론적으로, 교수자가 Serious Game 설계 및 운영을 위해 BBPL 플랫폼을 효과적으로 활용하기 위해서는 각 플랫폼의 강점과 한계를 명확히 이해하고 이를 학습 목표와 수업 환경에 맞게 연계하는 전략이 중요하다. 이를 위해 교수자가 학습 목표에 맞는 플랫폼을 선택할 수 있도록 다양한 플랫폼에 대한 연구 프로그램을 운영하거나 플랫폼 간 강점을 비교한 교육 자료를 제공하는 것도 필요하다.

본 연구의 주요 기여는 다음과 같다. 첫째, 교수자가 Serious Game을 설계하고 수업에 적용할 때 요구되는 기능적 요소를 도출하고 이를 BBPL 플랫폼 관점에서 체계적으로 분석함으로써, 플랫폼 선택과 활용 전략 수립에 실질적인 기준을 제공하였다. 둘째, 국내 교육 현장에서 널리 사용되는 Scratch, Entry, Code.org, App Inventor, Alice 등 5개의 플랫폼을 비교함으로써, 국내 교육 맥락에 적합한 실천적 시사점을 도출하였다. 셋째, 본 연구에서 제시한 기능 분석 준거는 향후 교육용 플랫폼 설계, 기능 개선, 정책 개발 등의 기초 자료로 활용 가능하며, 특히 교수자 중심의 교수 설계 지원 도구 개발에 이론적 토대를 제공할 수 있다.

본 연구의 제한점은 다음과 같다. 첫째, 본 연구는 Scratch, Entry, Code.org, App Inventor, Alice 등 국내 초·중등 교육 환경에서 널리 활용되는 BBPL 플랫폼 5개를 대상으로 Serious Game 설계 지원 기능을 분석하였다. 그러나 고등학생 이상의 학습자나 텍스트 기반 프로그래밍 환경이 가능한 플랫폼은 포함되지 않았다. 향후 연구에서는 다양한 학습 수준과 요구를 반영하여 BBPL 플랫폼의 활용 가능성을 폭넓게 탐구할 필요가 있다. 둘째, 본 연구는 교수자의 Serious Game 설계 지원 기능에 초점을 맞추어 BBPL 플랫폼을 분석하였다. 이는 기존 학습자 중심 연구에서 다루지 않았던 교수자 지원 영역을 보완했다는 점에서 의의가 있다. 다만, Serious Game 설계라는 특정 기능에 국한되었기 때문에, 향후 연구에서는 수업 목표 설정, 학습 평가 설계 등 교수설계의 다른 단계와 BBPL 플랫폼의 연계성을 심층적으로 탐구할 필요가 있다. 이를 통해 교수설계 전반에서 BBPL 플랫폼의 활용 가능성을 보다 포괄적으로 논의할 수 있을 것이다.

참고문헌

- [1] Ministry of Education. (2022). *Information Curriculum* (Notification No. 2022-33). Ministry of Education. <https://www.moe.go.kr/>
- [2] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49, 33-35. <https://doi.org/10.1145/1118178.1118215>
- [3] Ministry of Education. (2022). *Comprehensive Plan for Fostering Digital Talent*. Ministry of Education. <https://www.moe.go.kr/>
- [4] Al-Sakkaf, A., Omar, M., & Ahmad, M. (2019). A systematic literature review of student engagement in software visualization: A theoretical perspective. *Computer Science Education*, 29(2-3), 283-309. <https://doi.org/10.1080/08993408.2018.1564611>
- [5] Mladenović, M., Žanko, Ž., & Agličević, M. (2021). The impact of using program visualization techniques on learning basic programming concepts at the K-12 level. *Computer Applications in Engineering Education*, 29(1), 145-159. <https://doi.org/10.1002/cae.22315>
- [6] Lai, X., & Wong, G. K. W. (2022). Collaborative versus individual problem solving in computational thinking through programming: A meta-analysis. *British Journal of Educational Technology*, 53(1), 150-170. <https://doi.org/10.1111/bjet.13157>
- [7] Sun, L., Guo, Z., & Hu, L. (2021). Educational games promote the development of students' computational thinking: A meta-analytic review. *Interactive Learning Environments*, 31(6), 3476-3490. <https://doi.org/10.1080/10494820.2021.1931891>
- [8] Karagiorgas, D. N., & Niemann, S. (2017). Gamification and game-based learning. *Journal of Educational Technology Systems*, 45(4), 499-519. <https://doi.org/10.1177/0047239516665105>
- [9] Dimitriadou, A., Djafarova, N., Turetken, O., Verkuyll, M., & Ferworn, A. (2020). Challenges in serious game design and development: Educators' experiences. *Simulation & Gaming*, 52(2), 132-152. <https://doi.org/10.1177/1046878120944197>
- [10] Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., et al. (2014). Computational thinking in K-9 education. *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference (ITiCSE-WGR 2014)*, Uppsala, Sweden, 1-29. <https://doi.org/10.1145/2713609.2713610>
- [11] Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking* [Ph.D. dissertation], Massachusetts Institute of Technology.
- [12] Jiang, B., Wang, X., Zhang, Y., & Liu, J. (2021). Understanding the relationship between computational thinking and computational participation: A case study from Scratch online community. *Educational Technology Research and Development*, 69(5), 2399-2421. <https://doi.org/10.1007/s11423-021-10021-8>

- [13] Basogain, X., Olabe, M. A., Olabe, J. C., & Rico, M. J. (2018). Computational thinking in pre-university blended learning classrooms. *Computers in Human Behavior*, 80, 412-419. <https://doi.org/10.1016/j.chb.2017.04.058>
- [14] Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- [15] Qualls, J. A., & Sherrell, L. B. (2010). Why computational thinking should be integrated into the curriculum. *Journal of Computing Sciences in Colleges*, 25(5), 66-71.
- [16] Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia - Social and Behavioral Sciences*, 47, 1991-1999. <https://doi.org/10.1016/j.sbspro.2012.06.938>
- [17] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67. <https://doi.org/10.1145/1592761.1592779>
- [18] Fang, X., Ng, D. T. K., Tam, W. T., & Yuen, M. W. (2023). Integrating computational thinking into primary mathematics: A case study of fraction lessons with Scratch programming activities. *Asian Journal for Mathematics Education*, 2(2), 220-239. <https://doi.org/10.1177/27527263231181963>
- [19] Garneli, V., Giannakos, M. N., & Chorianopoulos, K. (2015). Computing education in K-12 schools: A review of the literature. *Proceedings of the 2015 IEEE Global Engineering Education Conference (EDUCON)*, Tallinn, Estonia, 543-551. <https://doi.org/10.1109/EDUCON.2015.7096027>
- [20] Jiang, B., Zhao, W., Zhang, N., & Qiu, F. (2019). Programming trajectories analytics in block-based programming language learning. *Interactive Learning Environments*, 30(1), 113-126. <https://doi.org/10.1080/10494820.2019.1643741>
- [21] Marwan, S., Akram, B., Barnes, T., & Price, T. W. (2022). Adaptive immediate feedback for block-based programming: Design and evaluation. *IEEE Transactions on Learning Technologies*, 15(3), 406-420. <https://doi.org/10.1109/TLT.2022.3180984>
- [22] Lin, Y., & Weintrop, D. (2021). The landscape of block-based programming: Characteristics of block-based environments and how they support the transition to text-based programming. *Journal of Computer Languages*, 67, 101075. <https://doi.org/10.1016/j.jcola.2021.101075>
- [23] Devine, J., Finney, J., de Halleux, P., Moskal, M., Ball, T., and Hodges, S. "MakeCode and CODAL: Intuitive and efficient embedded systems programming for education," *Journal of Systems Architecture*, vol. 98, pp. 468-483, May 2019. <https://doi.org/10.1016/j.sysarc.2019.05.005>
- [24] Bart, A. C., Tibau, J., Kafura, D., Shaffer, C. A., & Tilevich, E. (2017). Design and evaluation of a block-based environment with a data science context. *IEEE Transactions on Emerging Topics in Computing*, 8(1), 182-192. <https://doi.org/10.1109/TETC.2017.2729585>
- [25] Wolber, D., Abelson, H., & Friedman, M. (2015). Democratizing computing with App Inventor. *GetMobile: Mobile Computing and Communications*, 18(4), 53-58. <https://doi.org/10.1145/2721914.2721935>
- [26] Heo, G. (2023). A meta-analysis of the effects of coding education programs. *Journal of Fisheries and Marine Education Research*, 35(1), 111-119. <https://doi.org/10.13000/JFMSE.2023.2.35.1.111>
- [27] Abdul Jabbar, A. I., & Felicia, P. (2015). Gameplay engagement and learning in game-based learning: A systematic review. *Review of Educational Research*, 85(4), 740-779. <https://doi.org/10.3102/0034654315577210>
- [28] Shah, M., & Foster, A. (2015). Developing and assessing teachers' knowledge of game-based learning. *Journal of Technology and Teacher Education*, 23(2), 241-267. <https://www.learntechlib.org/primary/p/147391/>
- [29] Silseth, K. (2012). The multivoicedness of gameplay: Exploring the unfolding of a student's learning trajectory in a gaming context at school. *International Journal of Computer-Supported Collaborative Learning*, 7, 63-84. <https://doi.org/10.1007/s11412-011-9132-x>
- [30] Breuer, J., & Bente, G. (2010). Why so serious? On the relation of serious games and learning. *Journal for Computer Game Culture*, 4(1), 7-24. <https://doi.org/10.7557/23.6111>
- [31] Krath, J., Schürmann, L., & von Korflesch, H. F. (2021). Revealing the theoretical basis of gamification: A systematic review and analysis of theory in research on gamification, serious games, and game-based learning. *Computers in Human Behavior*, 125, 106963. <https://doi.org/10.1016/j.chb.2019.106963>
- [32] Rojas-López, A., Rincón-Flores, E. G., Mena, J., García-Peñalvo, F. J., & Ramírez-Montoya, M. S. (2019). Engagement in the course of programming in higher education through the use of gamification. *Universal Access in the Information Society*, 18(3), 583-597. <https://doi.org/10.1007/s10209-019-00680-z>
- [33] Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2017). CMX: The effects of an educational MMORPG on learning and teaching computer programming. *IEEE Transactions on Learning Technologies*, 10(2), 219-235. <https://doi.org/10.1109/TLT.2016.2556666>
- [34] Gómez, R. L., & Suárez, A. M. (2021). Gaming to succeed in college: Protocol for a scoping review of quantitative studies on the design and use of serious games for enhancing teaching and learning in higher education. *International Journal of Educational Research Open*, 2, 100021. <https://doi.org/10.1016/j.ijedro.2020.100021>
- [35] Bunt, L. R., & Greeff, J. J. (2020). Design and development of a narrative-based tabletop game for promoting soft skills acquisition in technical students in the South African context. *Journal of Games, Game Art, and Gamification*, 5(2), 41-47. <https://doi.org/10.21512/jggag>

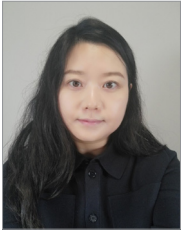
v5i2.7475

- [36] Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105, 106185. <https://doi.org/10.1016/j.chb.2019.106185>
- [37] Vahldick, A., Farah, P. R., Marcelino, M. J., & Mendes, A. J. (2020). A blocks-based serious game to support introductory computer programming in undergraduate education. *Computers in Human Behavior Reports*, 2, 100037. <https://doi.org/10.1016/j.chbr.2020.100037>
- [38] Shahid, M., Wajid, A., Haq, K. U., Saleem, I., & Shujja, A. H. (2019). A review of gamification for learning programming fundamentals. *Proceedings of the 2019 International Conference on Innovative Computing (ICIC)*, Lahore, Pakistan, 1-8. <https://doi.org/10.1109/ICIC.2019.8936050>
- [39] Vahldick, A., Mendes, A. J., & Marcelino, M. J. (2014). A review of games designed to improve introductory computer programming competencies. *Proceedings of the 2014 IEEE Frontiers in Education Conference (FIE)*, Madrid, Spain, 1-7. <https://doi.org/10.1109/FIE.2014.7044123>
- [40] Oksanen, K. (2013). Subjective experience and sociability in a collaborative serious game. *Simulation & Gaming*, 44(6), 767-793. <https://doi.org/10.1177/1046878113513079>
- [41] Li, Z., O'Brien, L., Flint, S., & Sankaranarayana, R. (2014). Object-oriented Sokoban solver: A serious game project for OOAD and AI education. *Proceedings of the 2014 IEEE Frontiers in Education Conference (FIE)*, Madrid, Spain, 1-4. <https://doi.org/10.1109/FIE.2014.7044236>
- [42] Lee, S., & Kim, S. (2023). Analysis of characteristics for the educational use of metaverse platforms. *Journal of Digital Contents Society*, 24(9), 2205-2214. <https://doi.org/10.9728/dcs.2023.24.9.2205>
- [43] Kim, S., Kim, G., Kim, S., Kim, Y., Kim, J., Kim, J., et al. (2021). Analysis of functions and services for the development of AI education platforms. *Journal of the Korea Association of Computer Education*, 24(2), 25-37. <https://doi.org/10.32431/kace.2021.24.2.003>
- [44] Lee, J., & Lee, S. (2020). A study on experts' perceptions of elementary AI education platforms. *Journal of Information Education Society*, 24(5), 483-494. <http://dx.doi.org/10.14352/jkaie.2020.24.5.483>
- [45] Kraveva, R., Kravev, V., & Kostadinova, D. (2019). A methodology for the analysis of block-based programming languages appropriate for children. *Journal of Computational Science & Engineering*, 13(1), 1-10. <http://dx.doi.org/10.5626/JCSE.2019.13.1.1>
- [46] Hellín, C. J., Calles-Esteban, F., Valledor, A., Gómez, J., Otón-Tortosa, S., & Tayebi, A. (2023). Enhancing student motivation and engagement through a gamified learning environment. *Sustainability*, 15(19), 14119. <https://doi.org/10.3390/su151914119>
- [47] Roque, R., Rusk, N., & Resnick, M. (2016). Supporting diverse and creative collaboration in the Scratch online community. In *Mass Collaboration and Education*, 241-256. https://doi.org/10.1007/978-3-319-13536-6_12
- [48] Dietz, G., Tamer, N., Ly, C., Le, J. K., & Landay, J. A. (2023). Visual StoryCoder: A multimodal programming environment for children's creation of stories. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, Hamburg, Germany, 1-16. <https://doi.org/10.1145/3544548.3580981>
- [49] Han, G. (2017). The impact of Entry programming education on elementary pre-service teachers' engagement. *Journal of Information Education Society*, 21(4), 403-413. <http://dx.doi.org/10.14352/jkaie.2017.21.4.403>
- [50] Lim, H. (2017). Teaching strategies utilizing 'code.org' in programming education emphasizing computational thinking. *Journal of Multimedia Society*, 20(2), 382-395. <https://doi.org/10.9717/kmms.2017.20.2.382>
- [51] Oh, K. S. (2019). App Inventor programming based on software engineering. In *Proceedings of the International Conference on Future Information & Communication Engineering*, Tokyo, Japan, June 2019.
- [52] Atir Abdullah. (2018). *Survey on block-based software education tools* [Master's thesis], Kookmin University, Seoul, South Korea.
- [53] Moreno-León, J., Robles, G., & Román-González, M. (2017). Towards data-driven learning paths to develop computational thinking with Scratch. *IEEE Transactions on Emerging Topics in Computing*, 8(1), 193-205. <https://doi.org/10.1109/TETC.2017.2734818>
- [54] Xie, B., & Abelson, H. (2016). Skill progression in MIT App Inventor. *Proceedings of the 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Cambridge, MA, USA, 213-217. <https://doi.org/10.1109/VLHCC.2016.7739687>
- [55] Lee, S. (2021). *Building an Alice library for programming education using storytelling* [Master's thesis], Ulsan University, Graduate School of Industrial Studies, Ulsan, South Korea.



오서진

- 2023년 영남대학교 영어교육과(문학사)
- 2024년 ~ 현재 한양대학교 교육공학과 석사과정
- ✚ 관심분야 : 게임기반학습(Game-Based Learning), 기능성 게임(Serious Game), 교수설계(Instructional Design), 생성형 AI
- ✉ osj9793@naver.com



이주영

- 1999년 동덕여자대학교 데이터정보학과(이학사)
- 2006년 이화여자대학교 교육학과 석사 수료
- 2024년 한양대학교 교육공학과(석사)
- 2025년 ~ 현재 한양대학교 교육공학과 박사과정
- ✚ 관심분야 : 교수설계(Instructional Design), 문제해결(Problem Solving), 컴퓨팅 사고(Computational Thinking), AI 사고(AI Thinking), CSCL
- ✉ siott9999@hanyang.ac.kr



임예린

- 2017년 Franklin & Marshall College 정치학과
- 2022년 Sciences Po Paris 국제개발학과(석사)
- 2024년 ~ 현재 한양대학교 교육공학과 박사과정
- ✚ 관심분야 : 교수설계(Instructional Design), 협력적 문제해결(Collaborative Problem Solving), 맞춤형 학습(Adaptive Learning)
- ✉ yealinim@gmail.com



신윤희

- 2018년 한양대학교 교육공학과(박사)
- 2020년 ~ 2021년 단국대학교 교양교육대학 코딩교과 조교수
- 2021년 ~ 현재 한양대학교 사범대학 교육공학과 조교수
- ✚ 관심분야 : 이러닝(E-learning), 컴퓨터기반 협력 학습(CSCL), 교수설계(Instructional Design), AI 리터러시
- ✉ yoonheeshin@hanyang.ac.kr