

# 웹 기반 파이썬 개발 환경과 메타버스 플랫폼\*

## Web-Based Python Development Environment and a Metaverse Platform

송은우<sup>+</sup> · 박용한<sup>+</sup> · 하선우<sup>+</sup> · 조민수<sup>++</sup> · 정기철<sup>+++</sup>

Eunwoo Song<sup>+</sup> · Yonghan Park<sup>+</sup> · Sunwoo Ha<sup>+</sup> · Minsoo Cho<sup>++</sup> · Keechul Jung<sup>+++</sup>

### 요 약

소프트웨어 교육의 중요성이 더욱 부각되면서, 학습자의 흥미를 고려한 플랫폼의 필요성이 증가하고 있다. 본 연구는 웹 기반 파이썬 개발 환경 Book Kit(북 킷)과 웹 기반 메타버스 플랫폼인 Playground(플레이그라운드)의 융합을 통해 학습자의 교육 환경을 개선하고자 한다. Book Kit은 개발 과정의 프로젝트 진행을 위한 파일 시스템, 그래픽 처리, 콘솔 등의 기능을 지원하고, 컴퓨터 그래픽 처리를 위한 프로세싱 블록을 기본적으로 제공한다. Book Kit의 쉬운 개발 환경과 시각적인 결과 제공은 학습자의 흥미를 높이기 위해 적절하다. Playground는 다른 사용자와 소통하고 자유롭게 이동하며 서로 공유한 프로젝트를 실행하거나 자세히 볼 수 있는 메타버스 환경을 제공한다. 또한, 본 연구에서는 프로젝트 공유를 위해 북 파일 포맷을 제안하여 학습자의 프로젝트를 관리한다. 본 연구에서 제안하는 플랫폼의 기능과 프로젝트 공유 솔루션을 통해 소프트웨어 교육에 대한 학습자들의 흥미를 증진시키기를 기대한다.

**주제어:** 블록코딩, 텍스트코딩, 프로세싱, 파이오다이드(Pyodide), 웹 어셈블리, 메타버스

### ABSTRACT

As the importance of software education continues to grow, the need for platforms considering learners' interests is on the rise. We propose an integration of two web-based environments(Book Kit) and Playground, to promote learning environment for students. Book Kit provides functionality for project development, including file systems, graphics processing, and a console, along with basic processing blocks for computer graphics. Its user-friendly development environment and visual feedback are well-suited to increasing learners' interest. Playground offers a metaverse environment where users can interact with each other, freely navigate, execute shared projects, and explore them in detail. Additionally, we introduce the concept of ".Book" file format for project sharing and management within this research. We anticipate that the features of the proposed platform and the project-sharing solution will foster greater interest among learners in the field of software education.

**Keywords:** Block-based programming, text-based programming, processing, pyodide, web-Assembly, sharing

### 1. 서론

컴퓨팅적 사고력은 현대 사회에서 중요한 핵심 역량으로 인식되고 있으며, 소프트웨어 교육의 중요성

과 관심이 증가하고 있다. 우리나라는 이러한 변화에 대응하기 위해, 2015년 개정 교육과정을 기준으로 소프트웨어 교육 시수를 점차 확대하고 있다[1].

학습자들의 흥미 또한 소프트웨어 교육 및 학습용

<sup>+</sup>정 회 원: 숭실대학교 글로벌미디어학부 학사과정 재학

<sup>++</sup>정 회 원: 한양대학교 ERICA 전자공학부 학사과정 재학

<sup>+++</sup>정 회 원: 숭실대학교 글로벌미디어학부 교수(교신저자)

논문 투고: 2023년 11월 01일, 심사완료: 2023년 11월 24일, 게재확정: 2024년 01월 08일

\* 이 논문은 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 지원으로 수행되었음. (2018-0-00209)

플랫폼을 구성하기 위해 고려되어야 할 요소이다. 프로그래밍 학습의 결과물로 시청각 요소는 학습자의 흥미 유발의 긍정적인 요소로 적용될 수 있다[2]. 또한, 소셜 미디어 기반 플랫폼을 사용하여 학습자 서로의 과제를 공유하는 것이 코딩 학습의 효능감 및 흥미에 긍정적인 영향을 미친다[3]. 반면, 텍스트 코딩을 위한 개발 환경 설정, 텍스트 코딩의 어려운 문법과 추상적인 속성, 개발 도구의 낮은 사용성은 학습자에게 부담이 되고, 흥미를 저하할 수 있다[4].

현재 프로그래밍 교육을 위한 도구로 주로 block-based programming(블록 코딩)을 중점적으로 사용하고 있다[5]. 블록 코딩은 코딩 교육의 초기에 컴퓨팅적 사고를 증진하기에 효과적이며, text-based programming(텍스트 코딩)과 비교하면 학생들의 학습 효과와 흥미를 높이기 위해 적절하다[6]. 그러나, 실제 애플리케이션 개발을 위한 프로그래밍 언어들은 텍스트 코딩이 주를 이루고 있어서, 학습자들이 텍스트 코딩 환경에 적응할 필요성이 있다.

본 연구는 다음 목표를 제시하고, 이를 위한 코딩 교육 환경을 제안한다. 1) 학습자들이 블록 코딩과 텍스트 코딩을 모두 활용할 수 있는 기능을 제공해야 하며, 쉬운 문법으로 구성된 프로그래밍 언어를 채택한다. 2) 학습자는 개발 환경의 설치없이 쉽게 사용할 수 있어야 하고, 다른 코드 에디터와 유사한 인터페이스 구성을 통해 비슷한 사용성을 제공해야 한다. 3) 자신이 작성한 결과물을 타인과 쉽게 공유하고, 실행할 수 있어야 하며, 다른 학습자들의 결과물을 자유롭게 돌아다니며 구경할 수 있어야 한다.

## 2. 연구 배경

### 2.1 Python

파이썬은 배우기 쉽고 기능이 강력한 프로그래밍 언어이다. 파이썬은 작성하기 쉬운 구문, 동적 타이핑, 인터프리터 특징을 가지고 있고, 다양한 모듈들이 활성화되어 있으며, 이런 특징들로 인해 많은 소프트웨어 교육에 사용되고 있다[7]. 따라서, 본 연구에서는 파이썬을 학습자의 언어로 선정하였다.

그러나, 이런 환경을 그대로 웹에 구현하기 위해서는 어려움이 있다. 파이썬으로 작성된 코드를 클라이언트-사이드에서 실행하기 위한 라이브러리로는 Transcrypt, Brython, Skulpt가 주로 사용된다. 앞의 라

이브러리 모두 파이썬으로 작성된 코드를 자바스크립트로 변환하여 실행한다. 각각의 라이브러리에서 지원하는 파이썬 패키지들은 쉽게 사용할 수 있으나, 추가적인 패키지를 설치하여 사용하는 것에는 어려움이 있다. 따라서, 파이썬의 온전한 기능을 구현하기 위해서는 다른 방식이 필요하다.

### 2.2 Pyodide

파이오다이드(Pyodide)는 Mozilla에서 제공하는 오픈소스 라이브러리로, 파이썬 언어의 핵심 구현체인 CPython을 Emscripten(C, C++ 등의 코드를 웹 어셈블리로 변환하는 툴 체인)을 이용해 웹 어셈블리(브라우저에서 동작하는 저수준 언어)로 변환한 라이브러리이다[8]. 따라서, 파이썬으로 작성된 코드를 실행하기 위한 툴 체인이 내장되어 있어, 파이썬 코드가 웹 어셈블리로 동작한다는 특징이 있다. 이를 이용하면 파이썬 언어로 작성된 모든 패키지를 동적으로 설치하여 실행할 수 있으며, C언어로 작성된 일부 파이썬 패키지 또한 실행할 수 있다. 또한, 가상의 파일 시스템을 통해 파일 입출력 기능을 지원한다.

본 연구에서는 파이오다이드를 이용하여 학습자에게 파이썬 개발 환경을 제공한다. 학습자는 별다른 설치 과정 없이 온전히 파이썬 개발 환경을 사용할 수 있다. 따라서, 프로그램을 설치하는 과정이 생략되므로 여타 파이썬 개발 환경에 비해 나아진 접근성을 제공하게 되고, 이는 곧 개발 환경 구성에 어려움을 느끼는 학습자에게 긍정적인 요인으로 작용할 것이다. 또한, 학습자는 다양한 파이썬 패키지를 사용할 수 있다. 이는 곧 초급부터 고급 단계의 학습자 모두가 사용할 수 있는 개발 환경을 제공할 것으로 기대된다.

### 2.3 Blockly

Blockly는 Google에서 제공하는 오픈소스 라이브러리로, 웹에서 인터랙티브하게 블록 코딩을 할 수 있는 편집기를 쉽게 추가할 수 있다[9]. 개발자는 편집기에서 사용할 블록을 정의하고, 블록에 해당하는 코드 구문을 작성하여 맞춤 블록을 생성할 수 있다. 블록 코딩으로 작성된 로직은 텍스트 코딩으로 변환할 수 있는 기능을 제공한다. 하지만, 텍스트에서 블록으로 변환하는 기능이 없으며, 작성된 코드의 실행은 기본적으로 자바스크립트로 동작한다는 한계가 존재한다.

본 연구에서는 블록과 파이썬 코드를 일대일로 대응시켜 실시간 변환 기능을 제공한다. 이는 학습자가 블록, 텍스트 코딩을 모두 사용할 수 있으며, 더욱 다양한 사용성을 제공한다. 또한 실시간 변환을 통해 블록과 텍스트 코딩의 관계를 바로 확인할 수 있으므로, 텍스트 코딩 환경에 효과적으로 적용할 수 있을 것이라 기대한다.

### 2.4 Processing

프로세싱은 자바에 기반한 프로그래밍 언어로, 시각 예술, 소프트웨어, 과학 기술 분야 등의 시각적 표현을 위해 무료로 제공되는 개발 환경이다[10]. 프로세싱을 이용하면 변수, 제어 구조, 함수, 픽셀 연산, 질차 및 객체 지행 개념, 신호 처리, 2D/3D 그래픽, 벡터 및 래스터 그래픽, 변환 등의 소프트웨어 기본 원리를 학습할 수 있다[11]. 또한, 다양한 예제 및 추가 라이브러리 등을 제공하고 있어 초보자에게 적합하다. 프로세싱은 현재 p5.js, processing.py, processing for android 등의 다양한 개발 환경을 사용자에게 제공하고 있다.

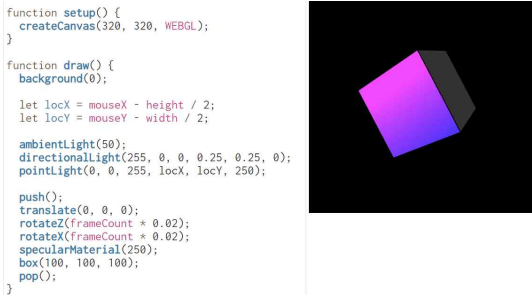


Figure 1. An example of code written in the p5.js environment.

Figure 1은 p5.js 개발 환경에서 작성된 예제 코드와 실행 결과물을 나타낸다. 이 그림에서 확인할 수 있듯이, 프로세싱은 간단한 코드만으로 컴퓨터 그래픽을 처리할 수 있으며, 그 결과물을 바로 확인할 수 있다. 따라서, 컴퓨터 그래픽을 처리하기 위해 적합한 개발 환경이다.

본 연구에서는 프로세싱을 주요 기능으로 제공하여 사용자가 쉽게 컴퓨터 그래픽을 다룰 수 있도록 유도하고, 결과물을 바로 확인하기 위한 레이아웃을 구성한다. 이는 프로그래밍의 추상적인 내용을 일부 시각

적으로 표현하기에 효과적인 방안이며, 학습자의 흥미를 증진하기 위한 요소로 적합하다. 또한, 그래픽을 직접 다루어 보며, 벡터 공간에 대한 이해를 높일 기회를 제공할 수 있을 것이라 기대한다.

### 2.5 Metaverse

메타버스(Metaverse)는 크게 3가지 특징을 가지고 있다. 첫째, 사용자에게 몰입감을 선사한다. 주로 현실이나 게임과 비슷한 그래픽, 작동 방식으로 실현되는 경우가 많다. 두 번째, 실시간 상호작용이 가능하다. 이 특징은 다른 사람과 실시간으로 소통하는 과정에서 특히 뚜렷하게 드러난다. 마지막으로, 사용자를 대신하는 아바타가 존재한다[12]. 현실을 일부 대신한다는 이 특징 또한 대부분의 메타버스에서 잘 나타난다.

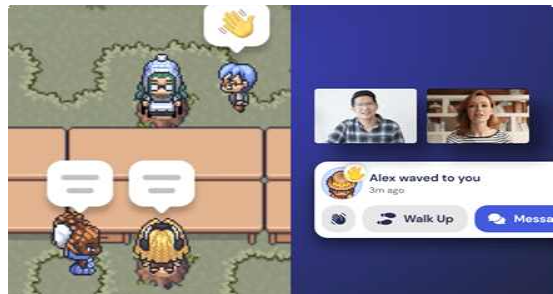


Figure 2. Gather, an example of 2D metaverse for work and study[13].

기존 온라인 학습 플랫폼에서의 문제점인 제한된 상호 작용의 문제점을 메타버스 기능을 통해 개선할 수 있다[14]. Figure 2는 현재 회사 또는 교육 기관에서 종종 사용되는 Gather의 이용 화면이다. Gather는 누구나 쉽게 자신의 아바타를 생성할 수 있으며, 가상의 공간 속에서 음성, 영상, 문자를 통해 다른 사용자와 소통할 수 있는 특징이 있다. 또한, 다양한 형태로 외부 데이터를 공유할 수 있어 공동 문서 작업, 투표 등의 작업을 수행할 수 있다[15]. 이처럼 가상의 공간에서 현실과 비슷한, 하지만 가상 공간의 특징을 활용해 현실에서는 실현되기 어려운 경험, 즉 더 편리한 경험을 사용자에게 제공하는 것이 메타버스의 궁극적 목표라고 볼 수 있다.

또한, 앞서 설명한 것과 같이 메타버스가 가지는 차별점은 모든 행동이 실시간으로 이루어지며 다른 사

용자들에게도 즉각적인 피드백이 들어간다는 것이다. 그리고 이는 곧 사용자들이 서로의 행동에 대해 보다 민감하게 영향을 받는다는 것을 의미한다. 이러한 특성은 사용자가 고무적으로 활동을 계속하게 하는 원동력이 되기도 하며, 나아가 적은 시간 내에 많은 정보가 오갈 수 있게 만들기도 한다.

본 연구에서는 메타버스의 특징을 활용하여 사용자 간의 소통을 유도하고, 실시간 상호작용을 통해 작품의 공유와 다른 이용자와의 소통을 더 자유롭고 효과적으로 경험할 기회를 제공한다.

### 3. 플랫폼 설계 및 구현

#### 3.1 플랫폼 구성 및 기능

본 장에서는 제안하는 플랫폼의 구성 및 기능을 서술한다.

첫 번째, 개발 환경은 사용하기 쉽고, 설정 없이 바로 사용하기 위하여 웹 기반 파이썬 개발 환경(이하, Book Kit)을 제안한다. 또한, Manager, Editor, Player 로 페이지를 나누고, 기능을 분산하여, 사용자가 요구하는 기능을 쉽게 찾아 실행할 수 있도록 구성한다. Manager에서는 사용자의 북 파일을 공유하거나 관리하고, Editor에서는 파이썬 개발 환경을 제공하여 북 파일의 편집을 수행하고, Player에서는 사용자가 공유한 북 파일을 실행할 수 있도록 구성한다.

두 번째, 학습자가 블록과 텍스트 모두를 활용하여 실시간으로 코드 편집이 가능해야 하고, 상용 코드 에디터와 유사한 레이아웃을 구성한다. Blockly 라이브러리에 파이썬 코드와 블록의 관계를 실시간으로 동기화하는 기능을 추가한다. 그리고, Blockly에 기본적으로 p5 모듈의 블록을 제공하여 컴퓨터 그래픽을 다루어 볼 수 있도록 유도한다. 그리고, 블록 코딩과 텍스트 코딩의 전환 버튼을 배치하여 쉽게 기능 간 전환할 수 있도록 배치한다.

세 번째, 프로젝트를 쉽게 공유하고 구경할 수 있으며, 다른 사용자와 자유롭게 소통하기 위한 메타버스 플랫폼(이하, Playground)을 제안한다. Playground에서는 보편적인 2D 픽셀 게임의 그래픽과 작동 방식을 사용한다. 또한 실시간 멀티플레이어 환경을 통해 다른 사용자와 쉽게 소통하기 위한 기능을 제공한다. 이와 같이 플랫폼 속에서 다양한 상호작용을 수행하고, 프로젝트를 구경할 수 있는 기능을 제공함으로써, 해

당 플랫폼을 통해 소프트웨어 교육의 긍정적인 경험과 즐거운 감정을 끌어낼 수 있도록 구성한다.

#### 3.2 프로젝트 관리 및 공유 아키텍처 설계

본 장에서는 학습자의 프로젝트를 관리하고 공유하기 위한 아키텍처를 기술한다. Figure 3은 본 연구에서 제안하는 플랫폼 간의 연결 관계를 간략하게 보여준다.

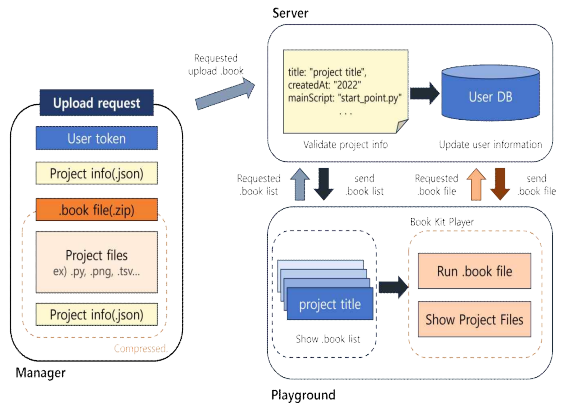


Figure 3. Project Interconnectivity Across Platforms.

첫 번째, 사용자의 프로젝트를 관리하고 공유하기 위해 북(book) 파일이 제공된다. 이 안에는 프로젝트 정보를 관리하기 위한 파일이 내장된다. 이 파일을 통해 프로젝트의 제목, 생성 일자, 작성자, 시작점 등의 중요한 세부 정보를 저장하고 수정할 수 있다. 북 파일은 zip 방식으로 압축되며, 사용자의 로컬 시스템에 저장된다.

두 번째, 북 파일을 사용하기 위해 파일의 접근 권한을 가져와 읽기 및 수정 작업을 수행한다. 다른 로컬 파일들을 북 파일에 추가할 수 있으며, 북 파일에 들어있는 파이썬 파일(.py)을 수정할 수 있다. 파일을 수정하고 관리하는 작업은 모두 클라이언트에서만 수행된다.

세 번째, 사용자는 북 파일의 공유를 위해 로컬에 저장된 북 파일을 서버에 올린다. 서버에 업로드된 북 파일을 통해 다른 사용자의 프로젝트를 확인하고 실행해 볼 수 있다. 이때 서버는 북 파일의 접근 권한을 링크 형식으로 제공한다. 해당 링크를 통해 북 파일을 자동으로 내려받아 실행함으로써 결과물을 확인할 수 있다.

마지막으로, 사용자는 본 연구에서 제안하는

Playground에서 다른 사용자의 프로젝트를 구경할 수 있다. 해당 플랫폼에 배치된 특정 객체와 상호작용을 하면 서버에 요청을 보내 작품의 접근 권한을 가진 링크들을 리스트로 받아오게 된다. 이 리스트에 포함된 작품들은 선택지의 형태로 사용자에게 제공되고, 특정 작품을 고르면 새로 열리는 창에서 구경할 수 있다.

### 3.3 Playground 설계

본 장에서는 메타버스 플랫폼인 Playground의 설계와 구현 구조를 기술한다.

사용자가 처음 접속하게 되면 통합 로그인 서비스를 통해 로그인 절차를 진행하게 된다. 그 후 방향키를 통해 움직일 수 있고, F 키를 통해 다양한 사물 및 non-player character(이하, NPC)와 상호작용할 수 있다. 공유와 몰입을 위해 로비, 도서관이 맵 내에 존재한다.

로비에서는 제공하는 Playground에 익숙해지는 과정을 밟게 된다. 이동과 상호작용을 익힌 후에 하게 되는 행동은 NPC와의 대화이다. NPC와 특정 거리에 들어가게 되면 NPC의 상단에 말풍선이 뜨게 되는데, 이때 상호작용 키를 누르면 대화창이 열리게 된다. NPC와의 대화 시스템은 선택지를 지원하는데, 이는 Yarn-Spinner(게임 내 대화 처리 플러그인) 형식에 기반한다. 그리고 로비에 있는 건물의 문에 접근하면 다른 공간으로 이동할 수 있다.

도서관에서는 다른 이용자의 작품을 확인하는 기능을 제공한다. 이용자는 도서관 내의 NPC와 대화를 통해 다른 이용자의 작품(북 파일)을 구경할 수 있다. 접근할 수 있는 작품의 링크를 선택지로 제공하고, 대화의 선택지에 따라 서버에게 요청을 보내 작품 접근 권한이 있는 링크를 받아와 Book Kit Player 페이지에 연결한다.

위 메타버스 플랫폼이 장점을 가지는 부분은 편리한 파일 공유 및 실행에 있다. 앞서 언급한 Gather의 경우, 서로 작성한 코드를 확인하고 공유하기 위해서는 한 사용자의 실시간 화면 공유를 통해 다른 사용자가 확인하거나, 코드 파일 그 자체를 공유해 각 사용자가 별도의 환경을 구축하여 확인해야 할 필요가 있다. 전자의 경우 사용자에 따라 화면 공유가 잘 이루어지지 않거나 화면 공유가 이루어진 시간에 접속하지 않으면 볼 수 없다. 후자의 경우 각 사용자의 개인 개발 환경이 다른 경우, 실행이 제대로 되지 않는 문

제점이 충분히 발생할 수 있다.

그러한 문제점을 해결하는 것이 본 메타버스 플랫폼이다. 간단한 대화창을 통해 별도의 환경 구축 없이 다른 사용자의 코드를 확인할 수 있을뿐더러, 그래픽을 포함한 실행 결과 또한 확인할 수 있다. 웹으로 통합된 개발 환경을 사용하니 각자 개발 환경이 달라 제대로 실행되지 않는 문제도 발생할 가능성이 없다고 간주해도 될 정도로 적어진다. 이처럼 본 메타버스 플랫폼은 기존 코드 공유에 있어 생기는 문제점을 해결하고, 접근성을 높이는 데 긍정적인 영향을 준다.

## 4. 결과

### 4.1 Book Kit 구현 결과

Figure 4는 Editor의 구현 결과를 제공한다. Editor의 화면 구성은 총 다섯 부분으로 나뉜다. 먼저, 최상단 부분은 코드 에디터를 다루기 위한 메뉴바로 구성되어 있다. 메뉴바를 통해 매니저 화면으로 이동할 수 있고, 새로운 파일을 추가하거나, 수정된 북 파일을 저장할 수 있다.

Figure 5는 메뉴바 하단에 있는 파일 탭 목록과 조작 패널이다. 파일 탭은 수정할 코드 파일을 선택하면 새로 생성되며, 상단의 파일 탭을 통해 다른 코드 파일로 이동할 수 있다. 열려있는 파일 탭이 너무 많은 경우, 생성되는 스크롤을 통해 이동할 수 있으며, 파일 탭 옆의 'X' 버튼을 통해 열려있는 파일 탭을 닫을 수 있다. 또한, 우측의 조작 패널을 통해 현재 수정 중인 코드 파일을 실행하거나, 중단할 수 있으며, 코드 편집 영역을 블록 모드 혹은 텍스트 모드로 변경할 수 있다.

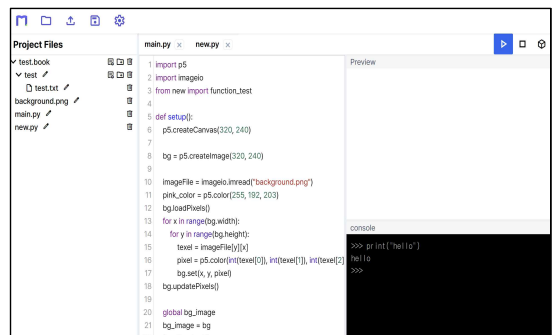


Figure 4. Implementation results of the Editor.



Figure 5. Implementation Results of the File Tab List.

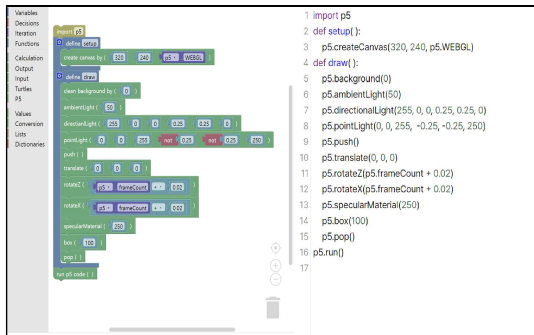


Figure 6. Left: Block Mode, Right: Text Mode.

Figure 6에서는 블록 및 텍스트 모드에 따라 제공되는 편집기의 화면을 보여준다. 먼저 블록 모드에서는 Blockly 편집기 안의 인터랙티브 블록을 사용하여 로직을 개발한다. 또한, p5 모듈에 대한 기본 블록이 제공되어 컴퓨터 그래픽을 쉽게 다룰 수 있다. 다음으로, 텍스트 모드에서는 파이썬 코드를 텍스트로 작성하여 로직을 구현한다. 이전에 블록 모드에서 작성한 코드는 자동으로 텍스트 형식으로 변환되어 표시된다. 또한, 파이썬 예약어와 내장 함수에 대한 문법 추천 기능을 추가하여 텍스트로 코드를 쉽게 작성할 수 있다.

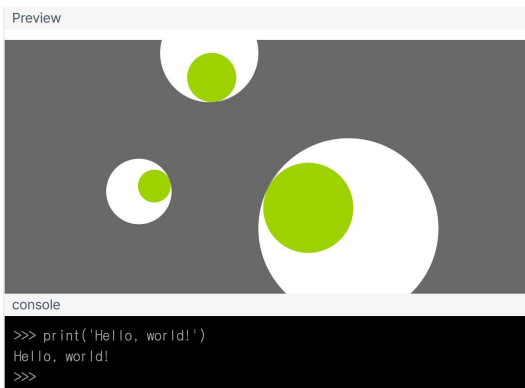


Figure 7. Top: graphics output, Bottom: Python Interpreter console.

Figure 7에서는 사용자가 작성한 로직의 실행 결과를 확인할 수 있다. p5, matplotlib, turtle 등의 모듈에서 처리된 컴퓨터 그래픽 결과는 상단의 preview에서 확인할 수 있다. 반면, 일반적인 텍스트 결과는 아래

console에서 확인할 수 있다. 또한, console에서 파이썬 인터프리터를 사용할 수 있다. 따라서, 사용자는 동적으로 파이썬 코드를 작성하고, 그 결과를 확인할 수 있다.



Figure 8. Implementation results of the File System.

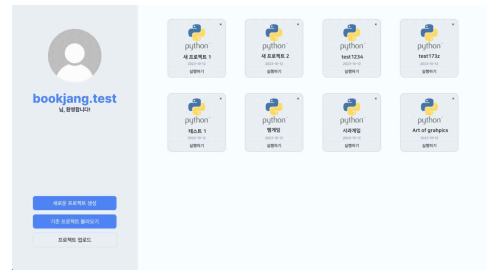


Figure 9. Implementation results of the Manager.

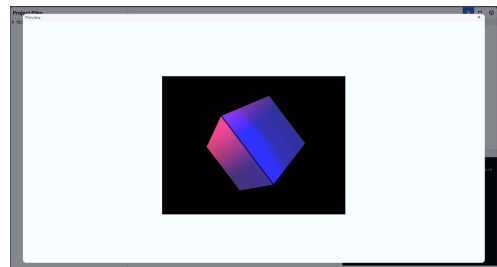


Figure 10. Implementation results of the Player.

Figure 8은 파일 시스템의 구현 결과로써, 파일 시스템은 플랫폼의 좌측에 있다. 사용자는 제공하는 파일 시스템 기능을 통해 파일 및 폴더를 새로 생성하거나, 수정 및 삭제할 수 있다. 파일 시스템에서 파이썬 파일을 클릭하면, 파일 탭이 생성되고, 편집 중이던 코드가 선택된 파일의 내용으로 변경된다.

Figure 9는 Manager의 구현 결과 화면이다. 해당 페이지를 통해 새로운 북 파일 생성, 기존의 북 파일 열기, 자신이 공유한 프로젝트의 실행, 공유 설정 변경을 할 수 있다. 이때, 공유 중인 북 파일을 실행하게 되면 플레이어 페이지로 이동한다.

Figure 10은 Player의 구현 결과를 제공한다. Player에서는 본인 및 다른 사용자들이 공유한 북 파일을 실행하고, 그 실행 결과를 확인할 수 있는 페이지이다. 또한, 우측 상단의 'X' 버튼을 눌러서 북 파일을 구성하고 있는 파일들을 살펴볼 수 있다. 다른 사람들이 작성한 코드를 살펴봄에 다양한 로직을 접할 기회를 사용자에게 제공한다.

#### 4.2 Playground 구현 결과



Figure 11. Playground Lobby Screen.

Figure 11은 Playground의 로비 화면이다. 이곳에서 플레이어는 상하좌우 방향키를 눌러 이동할 수 있으며, 'F' 키를 눌러 감정표현과 NPC 상호작용이 가능하다. 우측 상단에 존재하는 톱니바퀴 버튼, 혹은 키보드의 'ESC' 키를 누르면 Playground의 설정 창이 화면에 나타난다.

Figure 12는 NPC와 상호작용하는 화면이다. 사용자가 NPC와 가까워지면, NPC 상단에 말풍선이 표시된다. 이때, 'F' 키를 통해 해당 NPC와 상호작용을 할 수 있다. 상호작용 시, 이동 및 추가적인 상호작용이 제한되며, 하단에 별도의 검은색 창이 등장한다. 이 창의 좌측에는 NPC의 이름과 초상화를 확인할 수 있고, 중앙에서는 대화 내용이 출력된다.



Figure 12. Playground NPC Interaction Activation Screen.

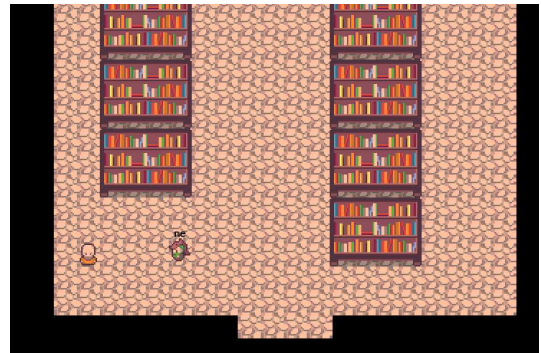


Figure 13. Playground Library Screen.

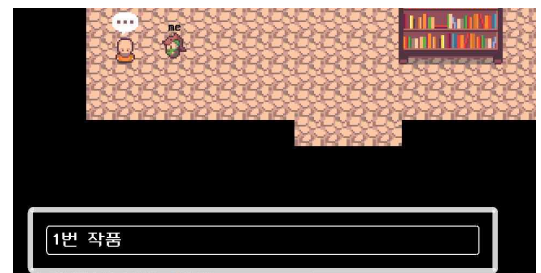


Figure 14. Playground NPC Conversation Choices.

Figure 13은 도서관 내부의 구성을 제공한다. 도서관 내부를 제외한 공간은 검은 화면으로 되어 있으며, 사서 NPC가 있다. 이 사서 NPC에서 다른 사용자의 작품을 열람하는 등의 기능을 수행할 수 있다.

Figure 14는 사서 NPC와 대화를 통해 다른 사용자의 작품 리스트를 출력한 결과이다. 현재 선택되어있는 선택지는 흰색 사각형으로 감싸져 있으며, 기본적으로 첫 번째 항목이 선택되어있다. 상하 방향키를 통해 선택지를 변경할 수 있고, 'F' 키를 누르면 선택지를 확정할 수 있다. 이 항목들은 서버의 DB로부터 다른 사용자의 작품(북 파일) 접근 권한을 가진 링크를 리스트로 받아오고, 이를 선택지와 연결하여 보여준다. 특정 항목을 선택한 경우, 선택지의 링크를 통해 Book Kit Player 페이지에 연결된다. 이는 브라우저의 새 탭이 생성되어 연결된다.

## 5. 결론

본 연구는 학습자들의 소프트웨어 교육에 대한 흥미를 높이기 위해 Book Kit과 Playground를 통합한 웹 기반 파이썬 개발 환경 및 메타버스 플랫폼을 제안한

다. 이를 통해 블록 코딩과 텍스트 코딩 모두 활용할 수 있으며, 파이오다이드를 통해 별다른 설치 과정 없이 파이썬 개발 환경을 사용할 수 있다. 또한, 자신이 작성한 복 파일을 쉽게 공유하고, Playground를 자유롭게 돌아다니며 다른 사용자의 프로젝트를 구경할 수 있다. 이러한 기능을 통해 소프트웨어 교육에 대한 흥미와 참여도가 개선될 것을 기대한다.

### 참고문헌

[1] Ministry of Education. (2022, December 22). *2022 Revised Elementary and Secondary School Curriculum Announcement*. <https://www.moe.go.kr/boardCnts/viewRenew.do?boardID=294&boardSeq=93459&lev=0&searchType=null&statusYN=W&page=1&s=moe&m=020402&opType=N>

[2] Kim S. (2022). A Case Study on Utilizing Open-Source Software SDL in C Programming Language Learning. *Journal of Practical Engineering Education*, 14(1), 1-10. DOI : 10.14702/JPEE.2022.001

[3] Lai, C., Wen, Y., Gao, T., & Lin, C. H. (2020). Mechanisms of the learning impact of teacher-organized online schoolwork sharing among primary school students. *Journal of Educational Computing Research*, 58(5), 978-1002. DOI : 10.1177/0735633119896874

[4] An, S., & Lee, Y. (2013, February). A Study of Programming Interest. *Proceeding of The Korean Association of Computer Education*. (pp. 103-106). Suncheon-si, Republic of Korea.

[5] Lee, A. (2018). Domestic research trends analysis of software education. *The Journal of Educational Information and Media*, 24(2), 277-302. DOI : 10.15833/KAFEIAM.24.2.277

[6] Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, 18(1), 1-25. DOI : 10.1145/3089799

[7] *Python 3.11.5 documentation - The Python Tutorial*. (n.d.). <https://docs.python.org/3/tutorial/index.html>

[8] Pyodide. (n.d.). <https://pyodide.org/>

[9] Blockly. (n.d.). <https://developers.google.com/blockly>

[10] Reas, C., & Fry, B. (2022). *Overview / Processing.org*

<https://processing.org/overview>

[11] Reas, C., & Fry, B. (2006). Processing: programming for the media arts. *Ai & Society*, 20, 526-538. DOI : 10.1007/s00146-006-0050-9

[12] McKinsey&Company. (2022). *What is the metaverse?*. <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-the-metaverse>

[13] Gather. (n.d.). <https://www.gather.town/>

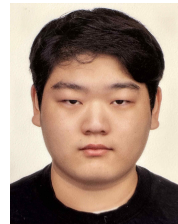
[14] Mystakidis, S. (2022). Metaverse. *Encyclopedia*, 2(1), 486-497. DOI : 10.3390/encyclopedia2010031

[15] Lee HJ, & Gu HH. (2022). Empirical Research on the Metaverse User Experience of Digital Natives. *Sustainability*, 14(22), 14747. DOI: 10.3390/su142214747



송은우

2022년 ~ 현재 송실대학교 글로벌미디어학부 학사과정 재학  
관심분야: 컴퓨터 교육, 컴퓨터 그래픽스, 웹 개발  
E-mail: song200348@gmail.com



박용한

2022년 ~ 현재 송실대학교 글로벌미디어학부 학사과정 재학  
관심분야: 게임 프로그래밍, 게임 기획, 웹 개발, 영상 편집  
E-mail: hodudragon0405@naver.com



하선우

2022년 ~ 현재 송실대학교 글로벌미디어학부 학사과정 재학  
관심분야: 컴퓨터 교육, 프론트엔드 프로그래밍  
E-mail: tjsdn811@gmail.com





조 민 수

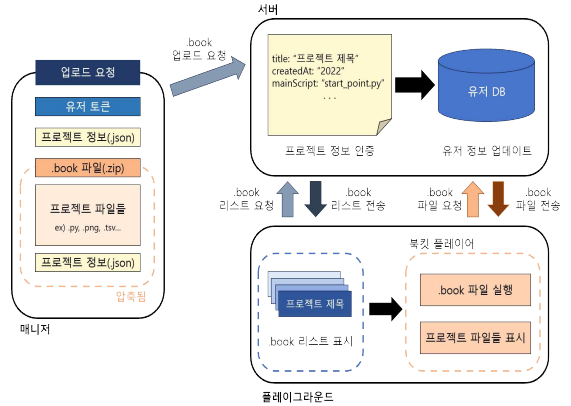
2022년 ~ 현재 한양대학교 ERICA 전자공학부 학사과정 재학  
 관심분야: 컴퓨터 교육, 백엔드 프로그래밍, DB 관리  
 E-mail: me@whiteb.dev



정 기 철

1994년 경북대학교 컴퓨터공학과 (공학사)  
 1996년 경북대학교 컴퓨터공학과 (공학석사)  
 2000년 경북대학교 컴퓨터공학과 (공학박사)

2000년 ~ 현재 숭실대학교 IT대학 글로벌미디어학부 교수  
 관심분야: HCI, EduTech, AI  
 E-mail: jungkeechul@gmail.com



[그림 3] 플랫폼 간 프로젝트 상호 연결도.

부록

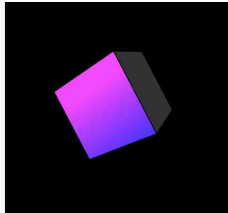
```
function setup() {
  createCanvas(320, 320, WEBGL);
}

function draw() {
  background(0);

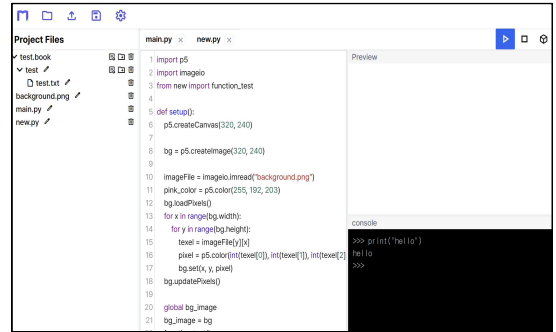
  let locX = mouseX - height / 2;
  let locY = mouseY - width / 2;

  ambientLight(50);
  directionalLight(255, 0, 0, 0.25, 0.25, 0);
  pointLight(0, 0, 255, locX, locY, 250);

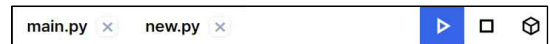
  push();
  translate(0, 0, 0);
  rotateZ(frameCount * 0.02);
  rotateX(frameCount * 0.02);
  specularMaterial(250);
  box(100, 100, 100);
  pop();
}
```



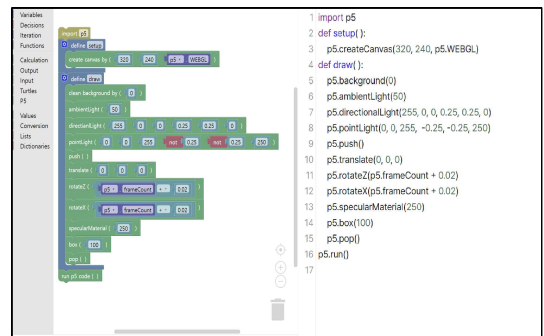
[그림 1] p5.js 환경에서 쓰인 코드의 예시.



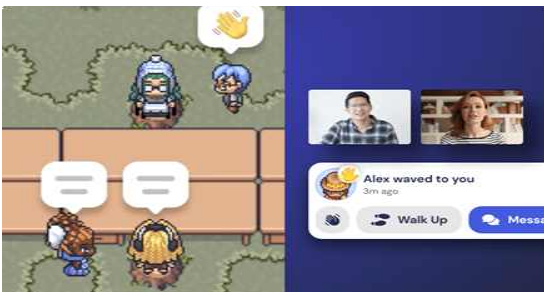
[그림 4] 에디터 구현 결과.



[그림 5] 파일 탭 리스트 구현 결과.



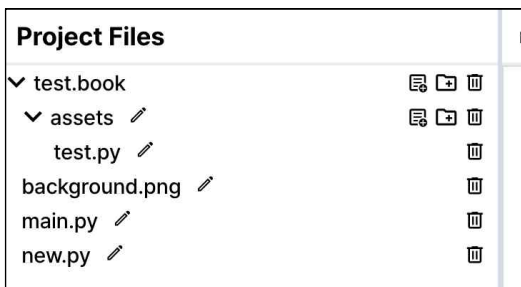
[그림 6] 좌측: 블록 모드, 우측: 텍스트 모드.



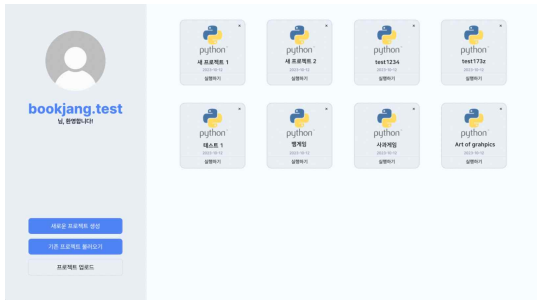
[그림 2] 업무와 학습을 위한 2D 메타버스 플랫폼의 예시, Gather[13].



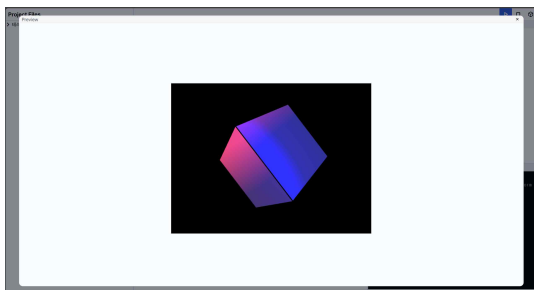
[그림 7] 상단: 그래픽 실행 결과, 하단: 파이썬 콘솔.



[그림 8] 파일 시스템 구현 결과.



[그림 9] Manager 구현 결과.



[그림 10] Player 구현 결과.



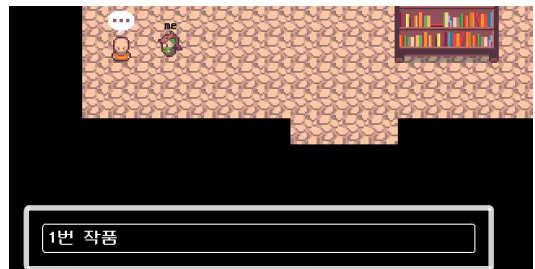
[그림 12] 플레이그라운드 로비 화면.



[그림 13] 플레이그라운드 NPC 상호작용 화면.



[그림 14] 플레이그라운드 도서관 화면.



[그림 15] 플레이그라운드 NPC 대화창 선택지.