The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 24, No. 5, pp.105-110, Oct. 31, 2024. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2024.24.5.105
JIIBC 2024-5-15

ChatGPT와 Roslyn을 활용한 게임 퀘스트 생성 연구

A Study on Game Quest Creation Using ChatGPT and Roslyn

임무결*, 김정이**

Moo-Gyeol Lim*, Jung-Yi Kim**

요 약 인공지능은 다양한 분야에서 활용되고 있으며 코딩 패러다임 또한 변화하고 있다. ChatGPT, GitHub의 Copilot등 다양한 코딩지원 AI들의 발전이 가속화 되고 있지만 이러한 코딩 도구들은 정확도, 보안, 라이선스 그리고 코드의 품질에 대한 과제가 남아있다. 따라서 이런 문제를 해결하기 위해 AI가 생성한 코드를 실시간으로 컴파일 함으로써 즉시 피드백을 받을 수 있도록 하여 개발 생산성을 높이고자 한다. 본 연구에서는 ChatGPT와 Roslyn의 결합을 테스트 해보고 실제 게임에 적용시킴으로써 해당 기술의 활용 가능성을 검증하고자 한다.

Abstract Artificial intelligence is being used in various fields, and the coding paradigm is also changing. Although the development of various coding-enabled AIs such as ChatGPT and GitHub's Copilot is accelerating, these coding tools remain challenged with accuracy, security, licensing, and code quality. Therefore, in order to solve this problem, AI-generated code can be compiled in real-time to receive immediate feedback to increase development productivity and improve code quality. This study aims to test the combination of ChatGPT and Roslyn and apply it to real games to verify the practical feasibility of the technology.

Key Words : ChatGPT, Game, Quest, Roslyn, Unity

Ⅰ. 서 론

인공지능(AI)은 다양한 분야에서 활발히 사용되고 있으며, 그 중요성은 증가하고 있다. 인공지능은 금융업, 제조업, 의료계, 서비스업 등 다양한 산업에서 활용되고 있고[1] 소비자들의 질문에 대해 답변하는 것뿐만 아니라 컨텐츠 생성 분야에서도 기여하고 있다. 웹 드라마를 개발하는데 이용[2]하거나 NPC의 대화체를 조선시대 어투로 바꾸는데 이용[3]하는 등 개발자의 수고를 덜어주는 방향으로 발전하고 있다. 또한 코딩 분야에서도 AI로 인해 패러다임이 변화하고 있다. ChatGPT를 시작으로 GitHub의 Copilot, Open AI Codex, Replit의 Ghostwirter 그리고 Untiy의 Muse 같은 AI 코딩 도구들이 등장하면서 코딩 지원 AI의 발전이 가속화되고 있다. MIT 연구 기사에 의하면 AI를 활용하면 평균적으로 작업시간을 절약하고 더 높은 평가를 받았다고 밝혔다[4]. 그리고 GitHub 공식 블로그의 연구에 의하면 GItHub Copilot을 사용하는 개발자들이 그렇지 않은 개발자보다 작업을 효율적이고 빠르게 완료하는데 도움을 준다[5].

하지만 개발 생산성을 크게 향상시키며 작업을 빠르고 효율적으로 완료하는데 도움을 주는 것으로 알려진 AI 도구들에 대한 문제점도 다수 지적되고 있다. 삼성SDS의

*학생회원, 성결대학교 미디어소프트웨어학과
**정회원, 성결대학교 미디어소프트웨어학과
접수일자 2024년 6월 30일, 수정완료 2024년 8월 30일
게재확정일자 2024년 10월 4일

Received: 30 June, 2024 / Revised: 30 August, 2024 / Accepted: 4 October, 2024
*Corresponding Author: ecesss@sungkyul.ac.kr
Dept. of Media Software, Sungkyul University, Korea

인사이트 리포트에서는 AI 기반 코딩 툴에 대한 해결 과제로 정확도, 보안, 라이선스 문제 등이 남아있다고 지적했다[6]. 그리고 GitClear 보고서에 따르면, AI 코딩 보조 도구를 개발자들이 사용함에 따라 기존코드를 리팩토링하고 재사용하는 대신, 복사/붙여넣기로 제작된 코드를 사용하면서 코드의 품질과 유지 보수성을 저하시킨다고 하였다[7].

이러한 도구들은 빠른 코드 생성에 초점을 맞추어 개발자들이 코드의 정확성이나 보안성, 라이선스 준수 여부를 충분히 고려하지 못하게 만든다. 게임을 개발할 때도 같은 문제가 발생하므로 AI 도구를 이용한 게임 개발 과정의 효율을 높이는 데 한계가 있다.

이러한 단점을 개선하기 위해 Untiy 게임 프로젝트에서 생성형 AI인 ChatGPT와 실시간 컴파일러인 Roslyn을 결합하여 ChatGPT로 생성된 코드를 즉각적으로 피드백 함으로써, 코드에서 발생하는 오류나 부적절한 코드를 실시간으로 방지하는 방법을 제안하고자 한다. 이 방법을 통해 개발 생산성을 높이고 코드 품질향상을 도모할 것이다. 그리고 이를 실제 게임에 적용하여 해당 기술의 실전 가능성을 검증하고자 한다.

II. 문헌 고찰

1. Roslyn(.NET Compiler Platform SDK)

Roslyn은 Microsoft가 개발한 .NET Compiler Platform의 코드네임으로 Roslyn은 오픈 소스 컴파일러 및 코드 분석 API로서, C# 및 Visual Basic.NET 코드를 분석하고 컴파일하기 위해 사용되며[8], Visual Studio, VS Code 등에서 사용된다[9]. Roslyn은 코드의 구문과 의미 분석을 수행하고, 그 결과를 API를 통해 외부 프로그램에서 사용할 수 있게 한다. 이를 통해 개발자는 컴파일 타임에 코드를 동적으로 생성하고 수정할 수 있으며, IDE 같은 도구에서 코드 분석 기능을 향상시킬 수 있다.

2. 생성형 AI 활용한 코드생성

ChatGPT의 코딩역량에 대한 많은 연구들이 진행되어왔다. 이 중 Surameery와 Shakor는 ChatGPT의 디버깅 지원, 버그 예측 및 설명 활용하여 프로그래밍 문제를 해결하는 데 도움이 되는 방법을 조사하고 버그를 더 효과적으로 식별하고 수정하기 위해 ChatGPT와 다른 디버깅 도구의 강점을 결합하는 이점을 강조하였다[10]. 최수지 외(2023)는 한국어 코딩 테스트에서의 인간 대 ChatGPT 3.5 & 4.0 성능 비교 및 평가 체계에서 GPT 모델은 기본적이고 기초적인 코딩문제는 해결하지만 복잡하거나 어려운 알고리즘 유형의 문제는 잘 해결하지 못함을 알 수 있었다[11].

이러한 연구를 바탕으로 ChatGPT의 코딩 능력을 활용한 다양한 선행 연구가 진행되었다. 김수정 외(2023)은 ChatGPT기반 코딩 도우미 연구에서 생성형 AI를 활용한 프롬프트가 학습자 수준에 맞는 코드와 연습 문제를 제공하여 교육적 효과를 높이는 데 기여할 수 있음을 확인하였다[12]. 그리고 김슬기(2023)은 생성형 AI를 활용한 프로그래밍 교육용 코드 생성 프롬프트 개발 연구[13]에서 AI의 도움으로 프로그래밍 문제를 더 효율적으로 해결할 수 있었지만, ChatGPT의 응답 속도와 정확성 그리고 반복되는 질문에 대한 최적화가 필요하다는 한계를 확인하였다. 그러나 선행 연구 모두 단일 프로그램 수준에서의 적용을 확인한 연구들이어서 규모가 크고 상호작용이 많은 프로그램에서의 적용 가능성에 대한 검토는 진행되지 않았다. 따라서 규모가 큰 프로젝트에서 ChatGPT와 Rosyln의 결합을 통해 ChatGPT 코드 생산의 한계점을 개선하고 개발 생산성 증가와 코드 품질 향상의 가능성 검증이 요구된다.

III. ChatGPT와 Roslyn 결합 테스트

ChatGPT와 Roslyn의 결합 가능성을 탐구하기 위해 ChatGPT에게 랜덤한 색과 랜덤한 회전 값이 주어진 Cube를 만드는 코드를 요청하고 Roslyn을 통해 실시간으로 컴파일하는 실험을 했다. 아래의 그림1은 유니티에서 ChatGPT와 Roslyn의 상호작용 과정을 도식으로 나타낸 것이다.

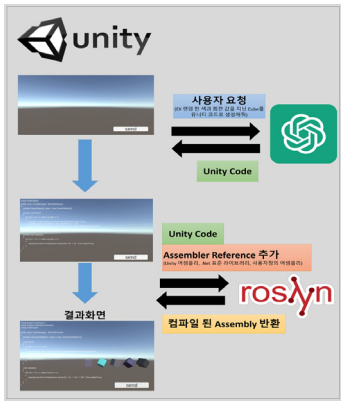

그림 1. 결합 테스트 도식화
Fig. 1. Combination Test Diagram

ChatGPT는 사용자가 요청한 대로 코드를 생성하고 Unity로 전송한다. Unity는 코드를 수신한 코드와 Untiy와 관련된 Assembler Reference을 Roslyn으로 전송한다. Roslyn은 코드를 분석하고 컴파일을 성공하면 컴파일 된 어셈블러를 Untiy로 반환한다. 이 과정을 통해 성공적으로 컴파일 된 코드가 런타임 환경에 실시간으로 반영 되도록 설계하고 구현하였다. 그 결과 사용자가 입력한 요구사항이 ChatGPT를 통해 그림 2 처럼 코드로 생성되고 Roslyn의 컴파일을 거쳐 실시간으로 Untiy에 통합되고 반영됨을 확인할 수 있었다. 이를 그림 3에 나타내었다.

그림 2. ChatGPT의 코드 생성
Fig. 2. Create code for ChatGPT

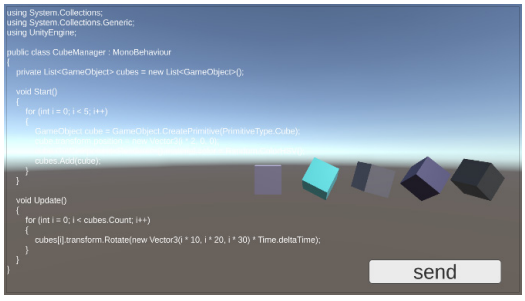

그림 3. 결합 테스트 실행
Fig. 3. Running a Combination Test

위와 같이 ChatGPT와 Roslyn를 결합하여 테스트한 결과를 활용하여 게임 개발 과정에 적용하면 다양하고 복잡한 코드를 개발하는 시간을 단축하고 효율을 높임으로써 개발자의 작업 능률을 높이고 편리함을 도모할 수 있을 것으로 기대된다.

이에 본 연구에서는 플레이어의 다채로운 퀘스트를 중심으로 진행되는 조선시대 배경의 게임을 개발하는 과정에 사용자에 요구에 맞게 코드를 생산하고 ChatGPT가 생삭한 코드의 오류를 Roslyn을 통해 실시간으로 컴파일 한 테스트 결과를 적용하여 게임의 볼륨을 키우고 게임 내 재화를 얻기 위한 반복적인 서브 퀘스트의 개발 시간을 단축시킴으로써 개발 생산성을 높이고 코드 품질 향상을 도모할 것이다. 이를 통해 규모가 크고 상호작용이 많은 게임 프로젝트에 적용하여 해당 기술의 실전 가능성을 검증하고자 한다.

Ⅳ. ChatGPT와 Roslyn을 활용한 게임퀘스트 생성 연구

1. 게임의 개요

조선관문일지는 플레이어가 조선시대 통행을 관리하는 검문소 관리가 되어 농민봉기를 막기 위해 탐관오리와 백성들의 요구를 만족시키며 다채로운 퀘스트를 반복적으로 수행하는 싱글 플레이 시뮬레이션 게임이다.

게임 내에서 백성의 민심을 달래기 위한 비슷한 퀘스트가 반복적으로 생성된다. 이러한 서브 퀘스트를 매번 개발하는 것은 개발자 업무에 시간적 비효율을 초래한다. 이 과정에서 AI의 도움을 받는 것이 효율적일 것이라 기대할 수 있다. 따라서 위의 테스트와 같이 ChatGPT와 Roslyn을 결합하여 정확도, 코드품질 등의 단점 개선한 방법을 적용하여 개발하였다. 아래의 그림 4와 5는 조선관문일지 게임의 첫 화면과 실행 화면이다.

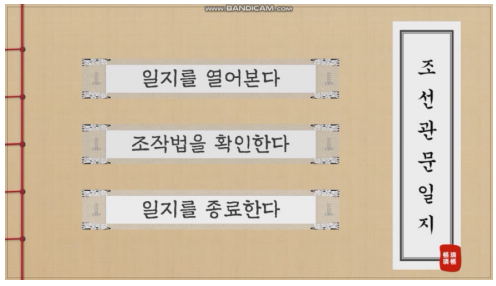

그림 4. 게임의 첫 화면
Fig. 4. Game Start Screen

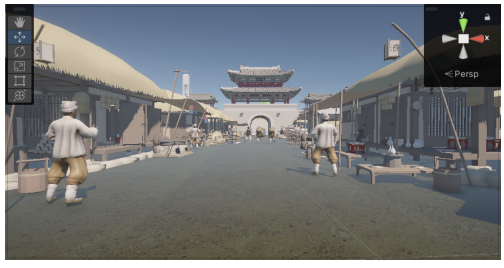

그림 5. 게임 실행 화면
Fig. 5. Game Launch Screen

2. 게임 퀘스트 생성

게임 제작과정에서 사용 중인 모든 소스코드를 ChatGPT에게 알려주고 알맞은 코드를 받는 것은 현실적으로 불가능했다. 이에 대한 대안으로 ChatGPT가 생성한 코드가 예측이 가능한 범주에 들어오도록 유도하기 위해 프롬프트에 템플릿 코드를 미리 전달하도록 하였다. 전달한 프롬프트는 그림 6에 나타내었고 프롬프트 내 템플릿 코드는 그림 7에 나타내었다.

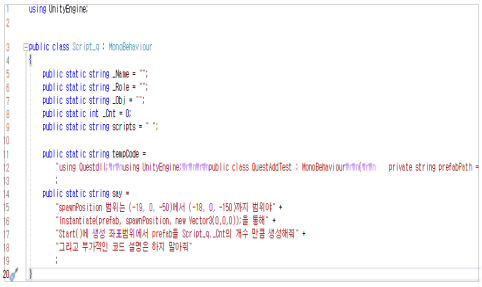

그림 6. 프롬프트
Fig. 6. Prompt

그림 7. 프롬프트 내 템플릿 코드
Fig. 7. Template code within Prompt

ChatGPT가 생성한 코드와 NPC, UI 등이 상호작용이 가능하도록 구현하기 위해 Quest를 생성클래스로 만들고 QuestManager를 싱글톤으로 만들었다. 그러나 생성한 외부 스크립트는 Roslyn 컴파일러가 알지 못한다는 문제가 있었다. 이를 해결하기 위해 그림 8과 같이 연관된 클래스와 프롬프트를 DLL파일로 제작한 뒤 Assemble Reference로 변환하여 컴파일러가 읽을 수 있도록 했다. 그리고 사용자의 요구를 받을 수 있도록 구현하기 위해 프롬프트의 핵심 키워드들을 전역변수로 만들었다. 이는 그림 6에서 확인이 가능하다.

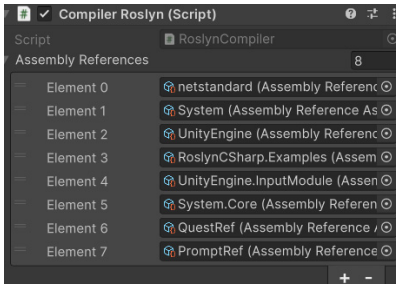

그림 8. Roslyn 컴파일러에 Assembly Reference를 전달
Fig. 8. Assembly Reference passed to Roslyn Compiler

3. 실험 결과

ChatGPT와 Roslyn을 활용한 게임 내 퀘스트 생성 작업을 통해 구조가 비슷하면서 내용이 다양한 퀘스트를 반복적으로 생성할 때 코드의 오류 없이 개발 과정에 적용하는 것이 가능함을 확인할 수 있었다. 그리고 Roslyn을 통한 실시간 컴파일을 통해 ChatGPT가 생성한 코드가 올바른 코드인지 잘못된 코드인지 즉각적인 피드백을 받을 수 있었다.

그림 9에서는 그 예로 InputText에 “퀘스트 이름이 순찰 퀘스트인 비콘 5개를 통과하면 클리어하는 퀘스트를 만들어줘” 라고 입력하면 코드 개발이 실행되는 장면을 나타내었다.

그림 9. 순찰 퀘스트 생성 화면
Fig. 9. Create Patrol Quest Screen

이번에는 게임 개발 과정에서 사용자가 퀘스트에서 획득할 아이템을 추가하고 이를 이용한 퀘스트를 ChatGPT와 Roslym을 통해 생성하는 방법을 시도하였다. 그 예

로 비단 Prefab을 그림 10과 같이 /Resource/Pregabs에 추가한 후 입력 문장을 적절하게 바꾸어 “퀘스트 이름이 수집 퀘스트인 비단 3개를 획득하면 클리어 되는 퀘스트를 만들어줘”를 입력하면 그림 11과 같이 적절한 퀘스트 코딩이 완성되고 실행되는 것을 확인할 수 있다.

그림 10. Prefab 경로
Fig. 10. Prefab Path

그림 11. 수집 퀘스트 생성 화면
Fig. 11. Create Collection Quest Screen

4. AI 코드 도구 사용과 ChatGPT와 Roslyn 방법 비교

기존 방법대비 개선된 부분을 확인하기 위해 AI 도구와 ChatGPT-Roslyn 통합 방식의 차이를 코드 작성 과정과 개발 경험의 측면에서 비교하였다.

표 1. 코드 작성 과정에서 ChatGPT와 Roslyn 활용법의 사용 절차의 차이점
Table 1. The difference between ChatGPT and Roslyn usage procedures in the code writing process

코드 작성	AI코드 생성 도구 사용	ChatGPT와 Roslyn 통합 방법 사용
코드작성	AI가 사용자 명령에 따라 코드 생성	ChatGPT가 사용자 명령에 따라 코드생성
2. 코드 컴파일	사용자 IDE에서 수동으로 컴파일	Roslyn이 실시간으로 자동 컴파일
3. 반복적 수정	오류가 있을 경우 사용자가 직접 수정 및 재 컴파일	실시간으로 피드백을 받고 사용자가 수정
4. 실행 및 디버깅	사용자가 수동으로 디버깅	피드백을 바탕으로 코드 수정 후 Roslyn으로 컴파일

표 2. 개발경험 측면에서 차이점
Table 2. Differences in terms of development experience

	일반 AI코드 생성 도구 사용	ChatGPT와 Roslyn 통합 방법 사용
작성 시간	코드는 빠르게 생성되지만, 오류 수정 및 컴파일에 시간이 소요	코드 생성과 피드백 제공을 통해 오류 탐지 시간 절약
개발 생산성	코드 생산이 빠르지만, 오류 수정이 길어질 수 있음	실시간 오류 탐지로 시간을 절약 그러나 수동으로 수정이 필요함

이 연구를 통해 ChatGPT가 생성한 코드의 정확성을 실시간으로 확인할 수 있었다. 앞으로는 더 나은 개발 생산성을 위해, 피드백을 바탕으로 ChatGPT가 생성한 코드를 자동으로 수정하는 기능을 개선할 필요가 있다. 이러한 수정 기능이 완성되면, 시간 절약과 오류 수정 속도의 차이를 정량적으로 분석하여 더욱 구체적인 결과를 도출할 계획이다.

V. 결 론

본 연구에서는 ChatGPT를 활용한 코드 생성과 Roslyn을 통한 실시간 코드 컴파일을 구현함으로써, 기존 코드 생성 AI 도구들의 즉각적인 피드백이 불가능하다는 단점을 극복하였다.

ChatGPT와 Roslyn을 통합하는 과정에서 ChatGPT와 Roslyn의 통신 지연이 발생하고 프롬프트에 수정이 필요할 때마다 새로운 DLL 파일을 만들어야 한다는 단점이 있었다. 그럼에도 불구하고, 이 연구를 통해 ChatGPT가 생성한 코드가 맞았는지 틀렸는지 실시간으로 확인이 가능함과 더불어 사용자의 입력과 코드 생성을 자동화하였으므로 개발 절차를 줄여 작업 능률 향상에 기여하였음을 확인할 수 있었다. 본 연구의 결과는 게임 개발뿐만 아니라 다양한 소프트웨어 개발 환경에서 응용 가능하여, 다양한 프로젝트에서 개발 생산성 증대에 기여할 수 있을 것이라 기대된다.

향후 연구에서는 프롬프트 엔지니어링 최적화를 통해 모델 입력의 품질을 향상시킴으로써 ChatGPT로부터 코드를 받아내는데 소요되는 시간을 감소시키기 위한 연구가 필요하다. 또한, 예외처리를 통해 오류 코드 검증 및 테스트를 ChatGPT와 상호작용 통해 스스로 수정하도록 만들어 코드의 안정성을 높일 것이다. 그리고 ChatGPT가 출력하는 코드가 사용 가능한지 검증하는 평가 항목을 만들어 생성되는 코드의 신뢰성을 높일 계획이다.

References

[1] Ju-Eun Kim, "An Analysis of the effect of Artificial Intelligence on Human Society", The International Promotion Agency of Culture Technology, Vol. 5, No. 2, pp. 177-182. May 2019.
DOI: https://doi.org/10.17703/JCCT.2019.5.2.177

[2] Hyun-Su Lee, Jung-Yi Kim, "Development of Story Recommendation through Character Web Drama Cliché Analysis", The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 23, No. 4, pp. 17-22, Aug 2023.
DOI: https://doi.org/10.7236/JIIBC.2023.23.4.17

[3] Jin-Seok Lee, In-Chal Choi, Jung-Yi Kim, "A Study on Expression of NPC Colloquial Speech using Chat-GPT API in Games against Joseon Dynasty Settings", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 24, No. 3, pp.157-162, Jun 30, 2024.
DOI: https://doi.org/10.7236/JIIBC.2024.24.3.157

[4] Zach Winn, "Study finds ChatGPT boosts worker productivity for some writing tasks", MIT News Office, Jul 2023.
https://news.mit.edu/2023/study-finds-chatgpt-boosts-worker-productivity-writing-0714

[5] GitHub Blog, "Research: quantifying GitHub Copilot's impact on developer productivity and happiness" Sep 2022.
https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/

[6] Samsung SDS, "Advantages and Disadvantages of Generative AI-based Coding Tools", Apr 2024.
https://www.samsungsds.com/kr/insights/generative-ai-coding-tool.html

[7] Visual Studio Magazine, "New GitHub Copilot Research Finds 'Downward Pressure on Code Quality'", Jan 2024.
https://visualstudiomagazine.com/Articles/2024/01/25/copilot-research.aspx

[8] Microsoft, "Welcome to the .NET Compiler Platform SDK (Roslyn APIs)", Jun 2024.
https://learn.microsoft.com/ko-kr/dotnet/csharp/roslyn-sdk/

[9] Microsoft. "Code analysis using .NET compiler platform (Roslyn) analyzers", Apr 5, 2024.
https://learn.microsoft.com/en-us/visualstudio/code-quality/roslyn-analyzers-overview?view=vs-2022

[10] N. M. S. Surameery, M. Y. Shakor, "Use Chat GPT toSolve Programming Bugs," International Journal ofInformation technology and Computer Engineering, Vol. 3, No. 1, pp. 17-22, January 2023.
DOI: https://doi.org/10.55529/ijitc.31.17.22

[11] Suzy Choi, Hae-Won Byun, "Human Programmers versus ChatGPT 3.5 & 4.0: A Comparison of Coding in Korean", Journal of Digital Contents Society Vol. 24, No. 12, pp. 3167-3178, Dec 2023.
DOI: https://doi.org/10.9728/dcs.2023.24.12.3167

[12] Soo-Jung Kim, Hi-jae Song, Dong-won Noh, Mi-ya Kim, Soo-bin Jeon, Dong-mahn Seo. "AI Coding Helper System Based on ChatGPT", Korean Institute of Information Scientists and Engineers, pp. 1901-1903, Dec 2022.

[13] Seul-ki Kim. "Developing Code Generation Prompts for Programming Education with Generative AI", The Korean Association Of Computer Education, Vol. 26, No. 5, pp. 107-117, Aug 2023.
DOI: https://doi.org/10.32431/kace.2023.26.5.009

[14] Moo-Gyeol Lim, Jin-Seok Lee, Jong-Mun Jeong, Jun-Seok Kyung, In-Chal Choi, Jung-Yi Kim, "Study on Programming Methods Using ChatGPT and Roslyn C# to Enhance Code Productivity in the Unity Engine", The IPACT 2024 Domestic Conference, Jun 2024.

저 자 소 개

임 무 결(학생회원)

- 2019년 3월 ~ 현재 : 성결대학교 미디어소프트웨어학과(재학)
- 관심분야 : 게임 프로그래밍, 인공지능

김 정 이(정회원)

- 2012년 2월 : 이화여자대학교 대학원 디지털미디어학부(박사)
- 현재 : 성결대학교 미디어소프트웨어학과 조교수
- 관심분야 : UX Design