

프로그래밍 학습을 위한 코드 변경 내역 재생 및 시각화 시스템 개발

Development of a Code Writing Playback and Visualization System for Programming Study

프로그래밍 초보자들은 코딩 학습 과정에서 반복적인 실수와 문제 해결의 어려움을 겪는 경우가 많으며, 이러한 상황은 학습 효율을 저하시킬 수 있다. 특히, 초보자들은 코드 작성 및 수정 과정에서 실수를 파악하고 이를 바로잡는 데 어려움을 느끼기 쉽다. 기존의 SVN, Git 등 버전 관리 시스템은 코드 변경 사항을 기록하지만 초보자 학습 지원에는 한계가 있으며, 수정 과정을 한눈에 파악하기 어렵다. 본 연구에서는 이러한 한계를 극복하기 위해 코드 입력 및 수정 과정을 시각적으로 재생하고, 자주 발생하는 오류와 디버깅 지점을 시각화하여 초보자가 학습의 집중 포인트를 쉽게 파악할 수 있는 VSCode 확장 프로그램을 개발했다. 이를 통해 학습자는 문제 해결 과정의 구조를 이해하고 유사한 실수를 예방하며, 문제 해결 능력을 효과적으로 향상시킬 수 있을 것으로 기대된다.

1. 서론

다양한 산업 분야에서 소프트웨어 개발 역량이 핵심 경쟁력으로 자리 잡으면서, 프로그래밍 교육의 중요성이 점점 더 커지고 있다[1]. 이에 따라 학생뿐만 아니라 직장인, 비전공자까지 코딩을 배우려는 학습 수요가 지속적으로 증가하고 있다. 프로그래밍은 단순히 문법적인 지식을 익히는 것을 넘어, 실질적인 문제 해결 능력을 키우는 것이 중요하다.

하지만 프로그래밍 초보자가 처음 문법을 배우고 나서, 코딩 테스트와 같이 문제 해결 능력 향상을 위한 코드 작성 시 문제와 실수를 명확히 인식하지 못해 비슷한 오류를 반복하거나 비효율적인 학습 패턴에 빠지는 경우가 많다. 따라서 자신이 작성한 코드의 변경 이력을 다시 복습하고 문제 해결 과정을 추적하는 것이 중요하다. 코딩 과정에서 어떤 부분에서 막혔고, 어떻게 문제를 해결했는지를 명확히 이해하면 유사한 실수를 방지할 수 있다.

SVN, Git 등의 버전 관리 시스템은 코드 변경 사항을 기록하는 데 널리 사용되고 있다. 하지만 상당 수의 코드 변경 내용은 버전 관리에 포함되지 않기 때문에 [3] 프로그래밍 초보자가 코딩 과정에서 어떤 어려움을 겪었는지 파악하는 데는 한계가 있다. 또한 기존의 버전관리 시스템은 변경이력이 시각적으로 표현되어 있지 않기 때문에 코딩 중 발생한 에러 등이 어떤 과정을 통해 수정되었는지를 한눈에 알아보기 어렵다[6].

이러한 한계를 개선하기 위해 본 연구에서는 자신이 작성한 코드의 전체 입력 및 수정 과정을 재생하고, 자주 발생하는 오류와 디버깅 위치를 시각적으로 표시해주는 시스템을 제안한다. 이 시스템은 학습 과정에서 집중해야 할 부분을 쉽게 파악할 수 있도록 돕고, 반복되는 실수를 줄이며, 문제 해결 능력을 효과적으로 향상시키는 데 기여할 수 있다.

2. 관련 연구

2.1 코드 변경 기록

사용자가 어떤 과정을 통해 코드를 변경했는지 기록하기 위해, SVN, Git 등의 버전 관리 시스템(Version Control System)이 널리 쓰이고 있다. 버전 관리 시스템은 코드의 변경 이력을 추적하기 좋다는 장점이 있지만 커밋 전의 코드 변경 이력은 저장되지 않는다. 저장 단위가 사용자가 임의로 결정한 커밋에 의존하기 때문에 소스 코드의 세부적인 변경 사항을 정확하게 추적하기 어렵다는 문제점이 있다[2, 3]. 일부 버전 관리 도구의 경우, 코드 변경 사항이 중앙 저장소에 통합되어 기록되는 구조적 한계를 가지고 있다[4]. 코드를 학습하는 과정에서는 단순한 코드 변경 이력 보다 더 세밀한 코드 입력 과정과 수정 내역을 추적할 수 있는 도구가 필요하다. 어디에서 개념적 오류가 발생했는지, 반복해서 어려움을 겪는 부분이 어디인지 커밋보다 더 세밀한 단위로 추적할 필요가 있다.

이러한 정보는 학습자가 코드 작성 과정에서의 실수와 오류 패턴을 정확히 이해하고, 이를 기반으로 효율적으로 복습할 수 있게 도와줄 수 있다.

2.2 코드 변경 시각화

소스 코드의 변경 이력 시각화는 소프트웨어 개발 과정에서 코드의 진화와 품질 향상을 이해하는 데 핵심적인 역할을 한다. 기존의 연구는 파일 수정에 걸린 시간을 시각화하거나 타임라인 시각화와 코드 히스토리 차이 뷰를 통해 코드 변화를 쉽게 탐색할 수 있도록 지원했다[5, 7]. 하지만 별도의 프로그램이 필요하며 디버깅 관련 기능이 부족하고 자동 재생이 불가능하다는 한계가 있었다. 또한, 레벤슈타인 편집 거리를 활용한 정밀한 코드 변경 추적 방법이 제시되었으나, 이는 전체적인 시각화보다는 알고리즘 구현에 집중되어 있었다[4].

본 연구에서는 코드 작성 중 발생하는 세밀한 수정 내역을 기록하고, 타임라인 형식으로 시각화하여 학습자의 코드 작성 이력을 효과적으로 파악할 수 있는 코드 기록 재생 및 시각화 서비스를 제안한다. 이러한 시스템을 통해 학습자가 자신의 코드 변경 과정을 시각적으로 추적하고 피드백을 받을 수 있어, 학습 동기와 자기 효능감을 향상시키는 효과를 기대할 수 있다[6].

3 시스템 구조

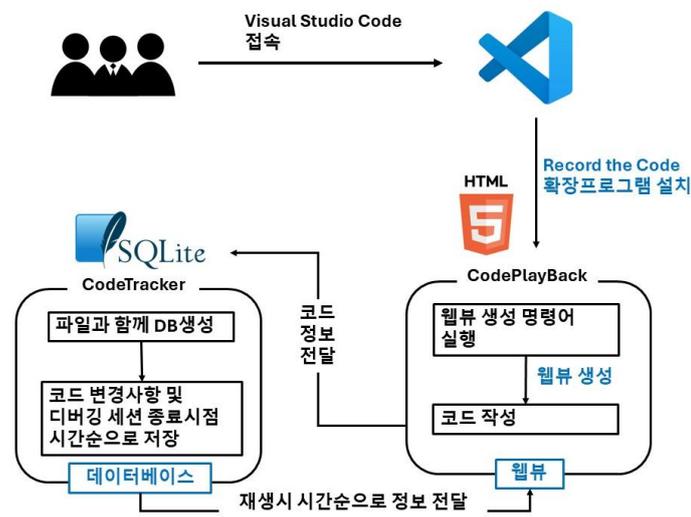


그림 1. 시스템 구조도

본 연구에서 제안하는 시스템의 구조는 [그림 1]과 같다. VSCode(MS Visual Studio Code)에서 작성하는 세밀한 코드 작성 내역을 기록하고 이를 시각적으로 복습할 수 있는 시스템을 개발하기 위해 두 가지 주요 모듈을 구현하였다.

3.1 코드 변경 이력 저장 (CodeTracker)

코드 작성 내역을 체계적으로 기록하기 위해 SQLite 데이터베이스를 활용하였다. VSCode에서 입력되는 모든 코드의 변경 이력(글자 삽입/삭제, 타임스탬프, 글자 좌표(행,열), 코드 실행 및 에러 발생 여부 등)을 데이터베이스에 저장했다.

3.2 코드 수정 내역 재생 (CodePlayBack)

HTML 기반의 웹뷰를 사용해서, 기록된 코드 변경 이력을 시각적으로 확인할 수 있는 CodePlayBack 모듈을 구현했다. 사용자는 코드 변경 이력을 마치 동영상 플레이어로 영상을 보듯이 재생하며 확인할 수 있다. 뿐만 아니라 재생바의 위치를 조정하면서 원하는 위치의 코드 변경 이력을 자유롭게 확인할 수 있다. 사용자가 작성한 코드에 맞는 구문 강조(Syntax Highlighting) 기능도 함께 제공해서 코드의 가독성을 높였으며, 코드의 실행 지점 및 해당 지점에서 발생한 에러 메시지까지 함께 표시할 수 있도록 구현했다.

4. VSCode 확장 프로그램

본 시스템은 TypeScript를 활용해서 VSCode 확장 프로그램으로 개발되었다. Azure DevOps를 통해 배포 프로세스를 자동화하였으며, VSCode Marketplace에 배포했다. 이를 통해, 본 확장프로그램을 설치한 사용자는 자유롭게 본인의 코드 변경 이력을 기록하고 시각적으로 변경 이력을 추적할 수 있다[8].

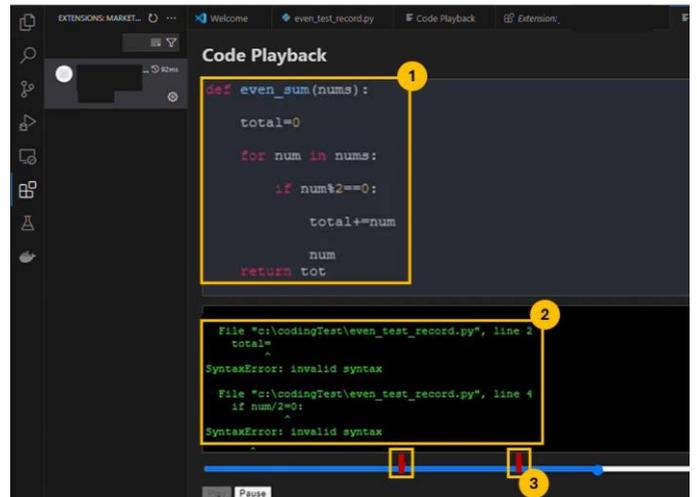


그림 2. 시스템 사용 화면

본 확장 프로그램을 사용하면 사용자의 모든 코드 변경 이력 및 에러 발생 지점이 기록되고, 이를 재생바 형식으로 재생할 수 있다. 이를 통해 특정 코드가 반복해서 지워지고 다시 작성되는 부분을 쉽게 파악할 수 있다. 사용자가 Play버튼을 눌러서 코드 변경 이력을 재생하면 [그림 2 ③]과 같이 재생바가 진행되면서 [그림 2 ①]의 코드 변경

내역이 동영상처럼 재생된다. 코드를 작성하는 과정에서 발생한 모든 오류도 재생바 상에 빨간 점으로 기록되어 있다, 코드가 재생되다가 실행 및 에러가 발생했던 시점에서는 디버그 창에 해당 에러가 표시된다[그림 2 ②]. 이를 통해, 사용자가 자신의 코드 작성 습관에서 어떤 부분에서 실수를 반복하는지 인지할 수 있으며, 이를 개선함으로써 코딩 실력을 향상시킬 수 있다.

4. 결론

본 연구에서는 프로그래밍 초보자가 코드 작성 및 수정 과정을 시각적으로 추적하고, 자주 발생하는 오류와 디버깅 위치를 자동으로 표시하는 VSCode 확장 프로그램을 개발했다. 이를 통해 학습자는 코드 변경 과정을 쉽게 복습하고, 반복적인 오류 패턴을 파악하여 문제 해결 능력을 향상시킬 수 있다. 본 시스템은 학습자의 집중 포인트를 명확히 제시하여 자기주도적 학습을 돕고, 프로그래밍 학습 과정의 효율성을 높이는 데 기여할 수 있을 것으로 기대된다.

참고문헌

[1]하수민. 코딩 교육 의무화 바람 ... “형태 변해도 코딩 열풍 이어질 것”, 고대신문, 2024.05.12

[2]Toth et al.“Using Version Control History to Follow the Changes of Source Code Elements”,2013 17th European Conference on Software Maintenance and Reengineering, 2012

[3]Negara et al. “Is It Dangerous to Use Version Control Histories to Study Source Code Evolution?”, ECOOP 2012 – Object-Oriented Programming, 79–103, 2012

[4]Canfora et al. “Identifying Changed Source Code Lines from Version Repositories”, Fourth International Workshop on Mining Software Repositories, 2007

[5]Yoon et al.“Visualization of fine-grained code change history”, 2013 IEEE Symposium on Visual Languages and Human Centric Computing, 2013

[6]전인성 and 송기상. On the Effect of a Pilot Coding Education Support System for Complex Problem Solving Tasks. The International Journal of Advanced Smart Convergence, 7(4), 128–137.

[7]Voinea et al. “CVSscan: visualization of code evolution”, SoftVis '05: Proceedings of the 2005 ACM symposium on Software visualization ,47 – 56, 2005

[8]YouTube. (2024). [시스템 시연 영상 - 프로그래밍 학습을 위한 코드 변경 내역 재생 및 시각화 시스템]. YouTube. <https://youtu.be/MZy0zsL-SJI>