

Custom Text Editor Technical Report

[2025-2] 실용프로그래밍실습

C277030 임경화

1. 프로젝트 개요

본 프로젝트는 PyQt 기반 GUI 위에 텍스트 편집기의 핵심 로직을 직접 구현한 과제이다.

QPlainTextEdit에 의존하지 않고, 텍스트 버퍼, 커서, 편집 명령, Undo/Redo, 검색/치환, 렌더링을 직접 설계 및 구현하는 것을 목표로 하였다.

- Github: https://github.com/lkh7764/-2025-2-PracticalProgrammingPractices/tree/main/PPP_Final_TextEditor
- 실행영상(MacOS 보안 정책으로 인해 파일I/O는 영상에 담지 못함): https://drive.google.com/drive/folders/1d2jQmEGisjrsItOq-Q_EB9LKTLN0LV1q?usp=share_link

2. 주요 기능

- 텍스트 입력 및 편집
 - 문자 입력, 줄바꿈(Enter), Backspace
 - 방향키 기반 커서 이동 (방향키, Home, End)
- Undo/Redo
 - Command 패턴 기반 Undo/Redo 스택
- 검색/치환
 - 다음 찾기(Find Next)
 - 현재 매치 치환(Replace Next)
 - 현재 검색 결과 하이라이트 표시
- 파일 I/O
 - UTF-8 텍스트 파일 열기/저장
 - 내부 줄바꿈은 \n으로 통일

3. 시스템 구조 main.py

```
└── EditorWidget (PyQt GUI)
    └── TextBuffer (core logic)
        ├── Cursor
        ├── Command (InsertCommand / DeleteCommand)
        └── UndoStack
```

4. 시스템 구현

- TextBuffer
 - 내부 표현: 문자열 리스트를 사용하여 데이터를 줄 단위로 저장.
 - 커서 표현: (row, col)
- Undo/Redo
 - 각 편집 동작을 Command 객체로 캡슐화.
 - UndoStack이 undo/redo 스택을 관리.
 - Replace 기능은 내부적으로 Delete+Insert 명령 조합으로 처리.
- 검색/하이라이트
 - 단순 문자열 검색 방식(find).
 - 커서 이후 문자열에서 매치되는 1개의 subarray만 하이라이트.
 - 하이라이트는 렌더링 단계(paintEvent)에서 배경 사각형으로 표시
- 플랫폼 호환성
 - macOS/Windows에서 동일한 코드로 실행 가능.
 - Qt의 표준 단축키(StandardKey)를 활용하여 Cmd(macOS)/Ctrl(Windows) 기반 처리.
 - 폰트는 고정폭 폰트를 사용하며, 시스템에 따라 대체 폰트를 적용.

5. 개발 환경 및 실행 방법

- 개발 환경
 - Python 3.11.14 / PyQt6
 - macOS
- 실행방법

```
pip install PyQt6
```

```
python main.py
```