Human ICT Software Engineering

**Final Report**

Simple Merge

20132682 김현빈

20132569 조성주

20134695 김명규

20134220 우정현

20131961 김민조

20162379 김성연

**목차**

1. **Project Description**
2. Summary
3. Development Environment

* Development OS
* Development Tool

1. **Analysis**
2. **Design & Implementation**
3. MVC Pattern
4. Model Design
5. View Design

* Main Frame
* Text Frame

1. Controller Design

* Diff Algorithm

1. Class Diagram
2. **Program UI and Features**
3. **Test**
4. Functional Unit Test Cases
5. GUI Test Case
6. **Execution Screen & Explanation**
7. **Project Management**
8. Program Description
9. Summary

이 프로젝트의 목표는 텍스트 파일 2개를 불러와서 비교하고 서로 다른 부분들을 줄 단위로 구분 해주며, 줄 단위로 복사 할 수 있으며, 불러온 텍스트 파일들을 창에서 편집할 수 있으며, 저장까지 할 수 있는 ‘Simple Merge’ 프로그램을 개발 하는 것이다.

1. Development Environment

- Development OS

프로젝트를 개발 할 때 사용 했던 운영체제는 window7, window10을 중점적으로 사용하였다

- Development Tool

JAVA 구현하였고 이에 사용된 버전은 1.8.0\_144 버전이다. GUI는 JAVA swing/awt를 사용하였고, Junit으로 Unit 테스트를 했다.

팀 프로젝트의 효율을 높이기 Git을 사용하였고 Git HUB를 이용하여 팀원들과 프로젝트 공유를 효율적으로 하였다.

본 문서는 Microsoft Word를 이용하여 작성하였다. SRS와 기타 문서들은 md를 이용하여 gitub에서 작성하였다.

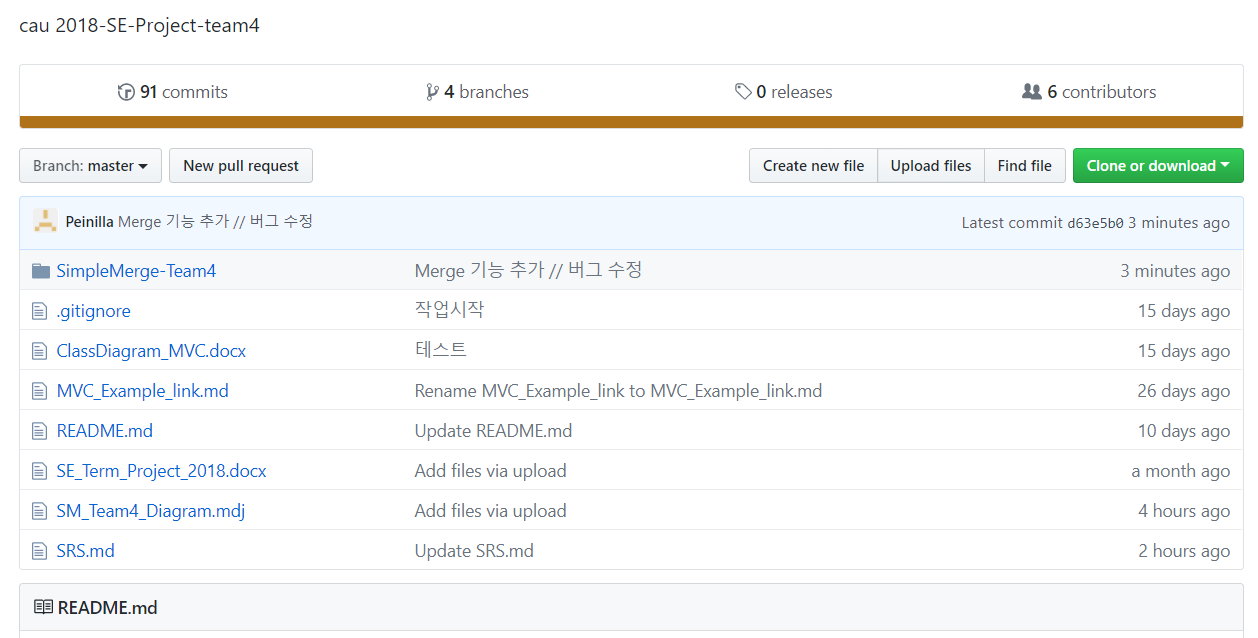


그림 1. Github

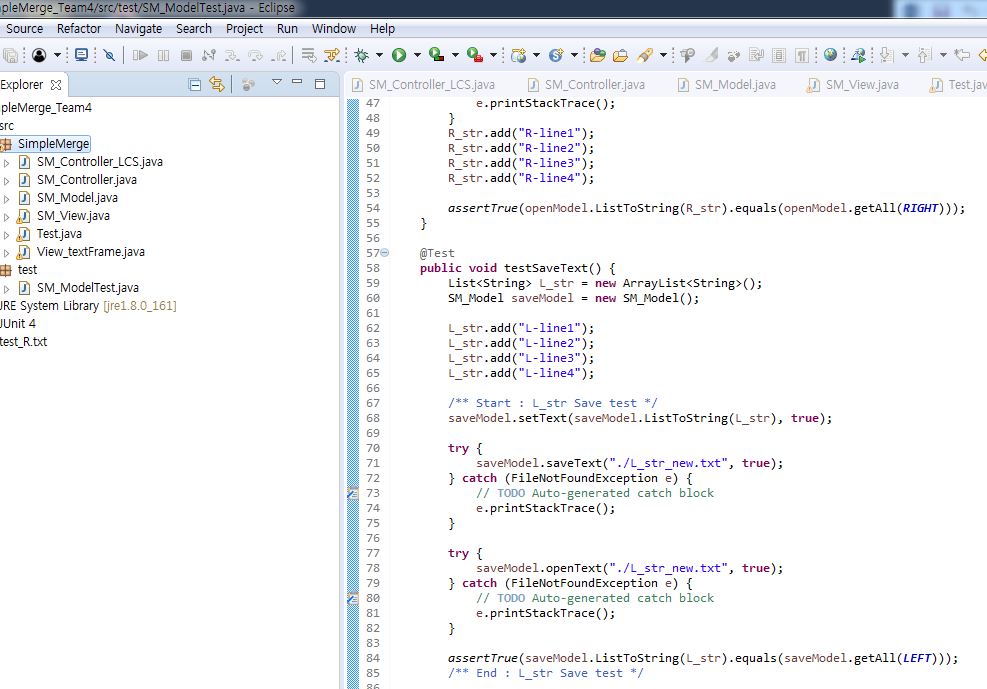


그림 2. 자바(이클립스)

1. Analysis

우리 팀은 목표로 하는 프로그램의 기능과 요구사항이 과제로서 명확하기에, 프로그램의 변화가 필요 없을 것으로 생각하여 폭포수모델(Waterfall Model)을 사용했다. 폭포수모델의 방식대로 요구사항 분석, 소프트웨어 설계, 소프트웨어 구현, 소프트웨어 시험, 소프트웨어 유지보수의 단계로 진행하였다.

먼저 요구사항 분석 단계에서는 우리가 프로그래밍해야 하는 프로그램의 주요기능들을 분석했다. 분석한 요구사항들을 알아보기 쉽게 문서화해서 SRS를 작성했다.

소프트웨어 설계 단계에서는 MVC 아키텍처 패턴에 맞춰 프로그램을 설계했다.

소프트웨어 구현 단계에서는 설계 한걸 기초로 하여 코딩을 하였다.

소프트웨어 시험 단계에서는 다양한 방법으로 프로그램의 테스트를 해보면서 이상한 버그가 없는지 확인을 하였다.

소프트웨어 유지보수 단계는 우리 프로그램의 미숙한 부분들을 고쳐나가는 것으로 하겠다.

**#구현하여야 하는 기능들**

**i. load**

텍스트 파일을 불러오는 기능이다.

Eidt 상태에서는 버튼이 비활성화 되어야한다.

**ii. edit**

불러온 텍스트를 편집하거나 빈 창에서 텍스트를 작성 할 수 있는 기능이다.

**iii. diff**

두 텍스트 파일들을 줄 단위로 서로 다른 부분들을 색칠해서 표시해주는 기능이다.

같은 줄은 색을 안 칠하고, 서로 다른 줄은 분홍색, 추가된 빈 줄들은 회색으로 칠할것이다.

**iv. copy**

‘diff’를 사용하여 나타낸 서로 다른 부분이 있는 줄을 위에서부터 차례대로 왼쪽텍스트를 오른쪽에 붙여넣기 하던가, 오른쪽텍스트를 왼쪽에 붙여넣기 하는 기능이다.

Diff가 활성화 된 상태가 아니면 비활성화 되어야 한다.

Edit 상태에서도 비활성화 되어야 한다.

**v. save**

편집된 텍스트 파일을 저장하는 기능이다.

Edit 상태에서 비활성화 되어야 한다.

1. Design & Implementation

1) MVC Pattern

MVC 패턴은 응용 프로그램을 세 가지의 구성요소인 모델, 뷰, 컨트롤러로 나누는 디자인 패턴이다.

모델은 List<String> 을 이용해서 텍스트를 저장 하며, 컨트롤러에 의해 뷰에게 텍스트를 String으로 넘겨준다. 또한 컨트롤러를 통해 FilePath를 받아와서 텍스트를 파일로 불러오기/저장을 수행한다.

뷰는 swing을 이용하여 UI를 담당한다. 컨트롤러를 통해 변경되는 사항을 TextPane과 Button에 적용한다.

컨트롤러는 뷰의 액션을 받아와서 각 액션에 맞는 기능을 수행한다. 기능에 따라 모델과 뷰를 제어한다.

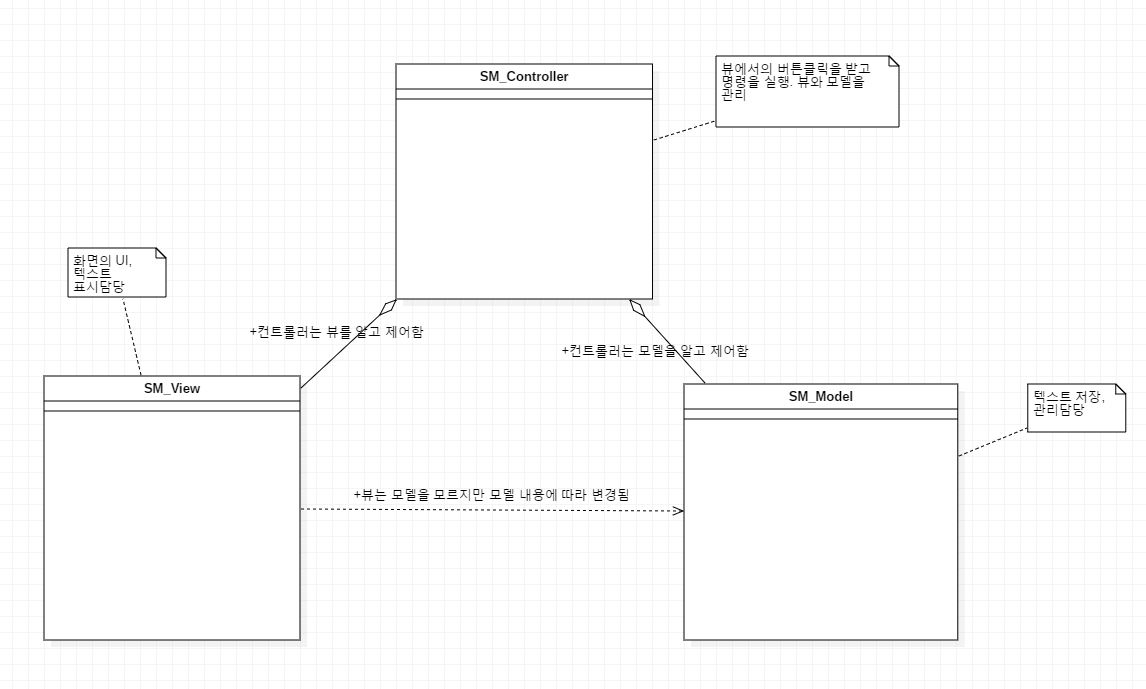


그림 3. MVC 패턴

2) Model Design

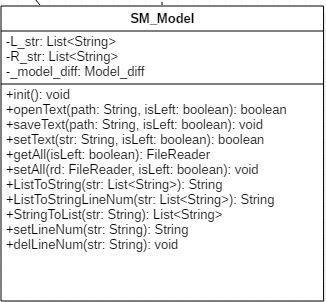


그림 4. Model design

@ SM\_Model

MVC 패턴 중 모델에 속하는 클래스로, 텍스트를 저장할 List<String> L\_str 과 R\_str를 변수로 가진다.

**생성자**를 통해 L\_str 과 R\_str을 ArrayList<String>으로 초기화한다.

**openText / saveText** 는 FilePath에 해당하는 파일을 불러오기/저장한다. 인수로 받는 FilePath가 JFileChooer를 이용한 path이기 때문에 별다른 예외처리를 하지 않는다. 좌 우 방향은 인자로 받는 boolean으로 결정한다.

처음 실행창이 켜졌을 때 완전 빈 텍스트 창이 뜨는걸 막기 위해서 스트링을 추가한 상태로 간다. 실행창에서 텍스트파일을 읽어오는 기능과 저장하는 기능을 구현하면서 리스트마다 줄 번호를 주는 기능까지 있는 클래스이다.

줄 번호를 주는 이유는 diff 상태에서 merge를 구현할 때 LCS 방법을 이용하기 편하기 위해 구현하였다.

3) View Design

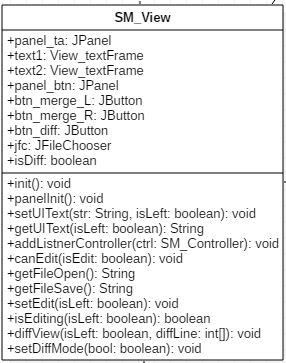


그림 5. SM\_View(Main Frame)

@ SM\_View

MVB패턴 중 뷰에 속하는 클래스로 JFrame을 상속받으며, UI구성하는 컴포너트들을 변수로 가지며 그 중에는 view\_textFrame을 좌 우로 2개가진다.

**생성자를** Frame설정을 변경하며, 컴포넌트들을 초기화한다. 보기 좋게 만들기 위해 컴포넌트가 들어있는 패널을 BorderLayout를 이용해 Frame에 넣는다.

**addListnerController**는 버튼들에게 ActionListener인 SM\_Controller를 추가한다.

**getFileOpen / getFileSave** 는 JFileChooser를 이용하여 불러오기/저장 탐색기를 연다. JFileChooser는 txt 확장자만 열 수 있다. 파일을 선택하지 않고 탐색기 끄면 아무 일도 일어나지 않는다. 파일을 선택하면 파일의 경로를 반환한다.

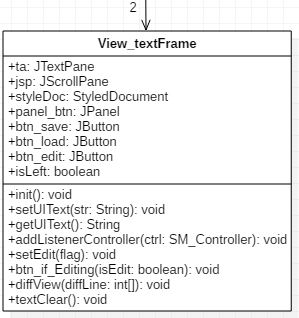


그림 6. View\_textFrame

@ View\_textFrame

MVC패턴 중 뷰에 속하는 클래스로 JPanel을 상속받으며, SM\_View에 의해 사용된다. Save/Load/Edit을 담당하는 버튼과 textPane을 변수로 가진다.

**생성자에** 인자로 받는 boolean형을 이용하여 좌우를 결정한다. 좌우 방향에 따라 버튼의 이름을 변경한다.

**diffView**는 인자로 받는 int[] diffLine을 이용한다. ( 0 = Diff Line / 1 = Same Line / 2 = Blank ) 배열의 인덱스에 해당하는 줄에 배열값에 따라 textPane에 색을 추가한다. ( 0 = 노란색 / 1 = 흰색 / 2 = 회색)

Blank는 라인넘버에 포함되지 않으며, 비교 가독성을 위해 추가한 기능이다.

4) Controller Design

-Main Controller

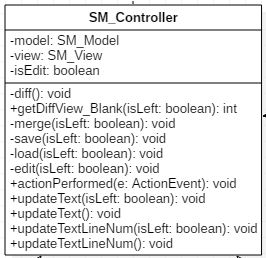


그림 7. SM\_Controller (Main Controller)

@ SM\_Controller

MVC 패턴 중 컨트롤러를 담당하는 클래스로 ActionListener 상속받는다. SM\_Controller은 뷰의 버튼들과 연동되어있어, 뷰의 액션을 통해 기능을 수행한다.

getDiffView는 SM\_Controller\_LCS를 이용해 만들어진 int 배열을 좌우에 따라 반환한다.

Merge는 다른 라인을 선택방향으로 복사한다. 한번에 한 줄씩, 위에서부터 차례대로 동작한다.

SM\_Model과 SM\_Controller\_LCS에서 구현된 것들을 ActionEvent로 받아 들여서

-Diff Algorithm

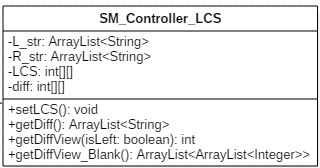


그림 8. SM\_Controller\_LCS (Diff Algorithm)

@ SM\_Controller\_LCS

MVC 패턴 중 컨트롤러를 담당하는 클래스로 Diff 기능 구현을 위해 LCS 알고리즘을 담당한다. 좌우 텍스트를 저장하는 ArrayList<String> L\_str, R\_str 2개 와 LCS 테이블을 담당하는 LCS : int[][], 테이블의 방향성을 저장하는 diff : int[][] 를 변수로 가진다.

생성자는 인자로 받은 ArrayList 들을 각각 L\_str과 R\_str에 복사한다.

setLCS 는 L\_str 과 R\_str을 한 줄씩 비교하여 만들어진 LCS 테이블을 LCS : int[][] 와 diff : int[][] 에 저장한다.

getDiff 를 통해 LCS : int[][] 와 diff : int[][]를 분석하여 최장 공통 부분을 ArrayList<String>으로 만든다.

getDiffView / getDiffView\_Blank 는 getDiff()를 통해 만들어진 최장공통부분 ArrayList<String>을 L\_str과 R\_str에 각각 비교하여 ( 0 = Diff Line / 1 = Same Line / 2 = Blank )을 가지는 int 배열을 만든다. int배열의 인덱스는 각 라인의 상태이다.

5) Class Diagram

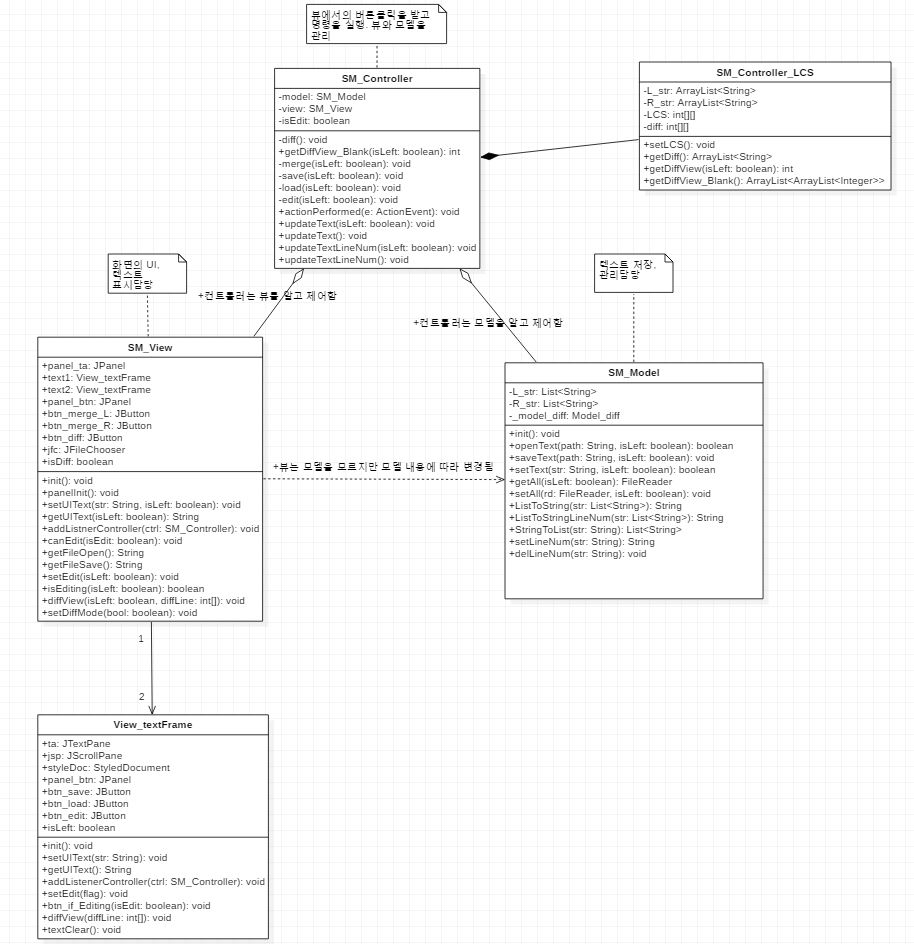
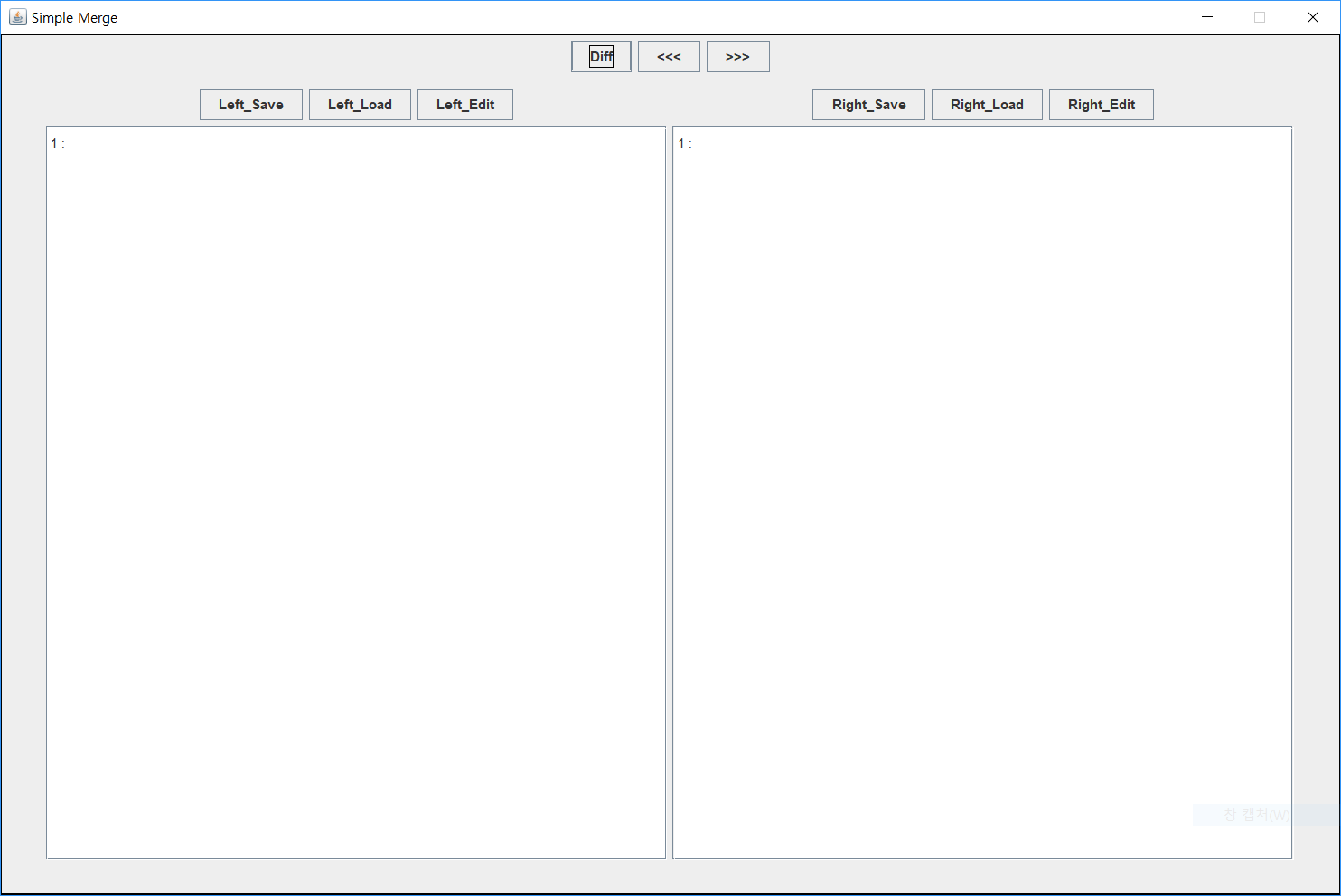


그림 9. Class Diagram

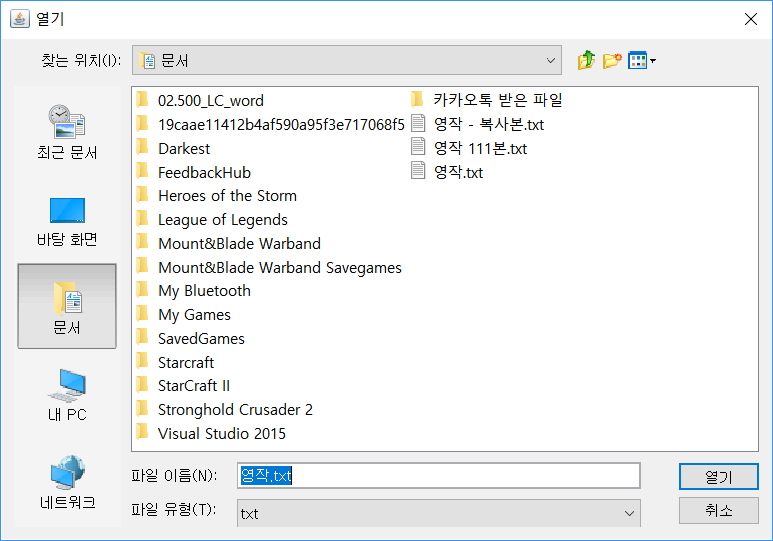
1. Program UI and Features



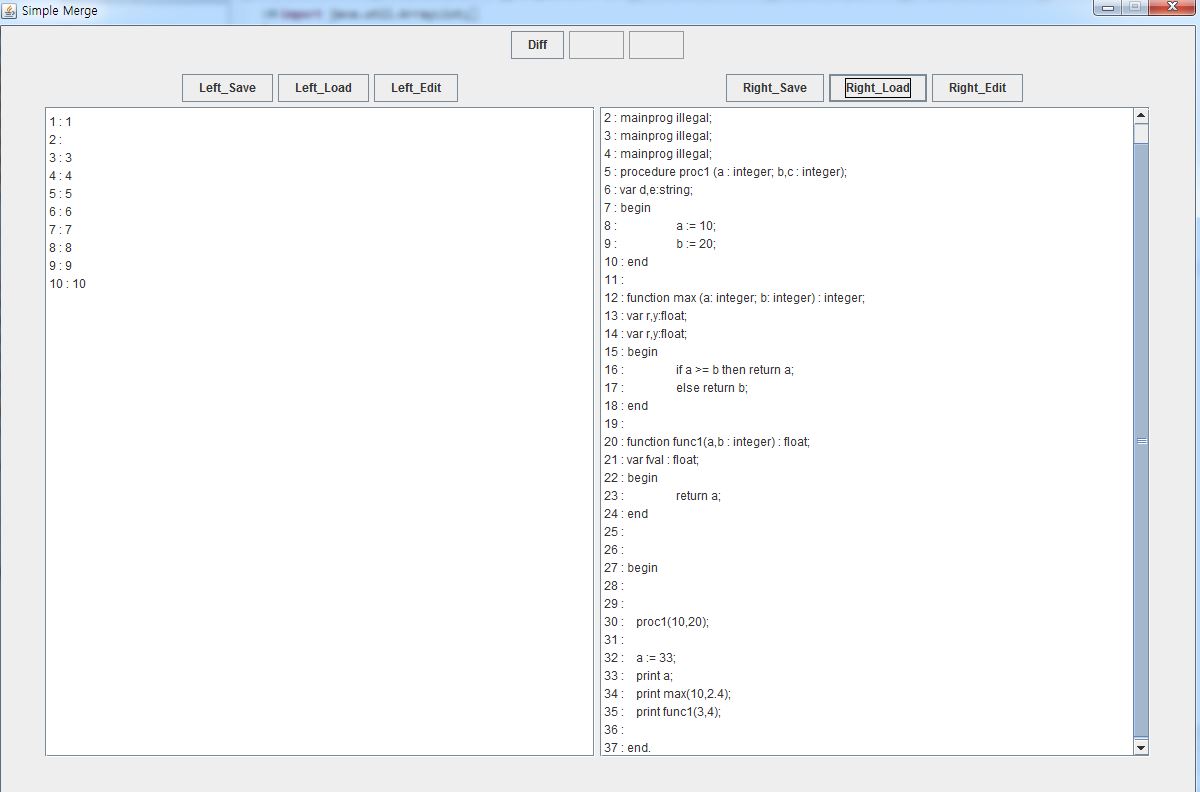
첫 실행 시 위와 같은 화면이 나타난다.

각 좌우 패널의 load버튼을 눌러 파일을 불러오거나 edit을 눌러 패널마다 수정을 할 수 있다.

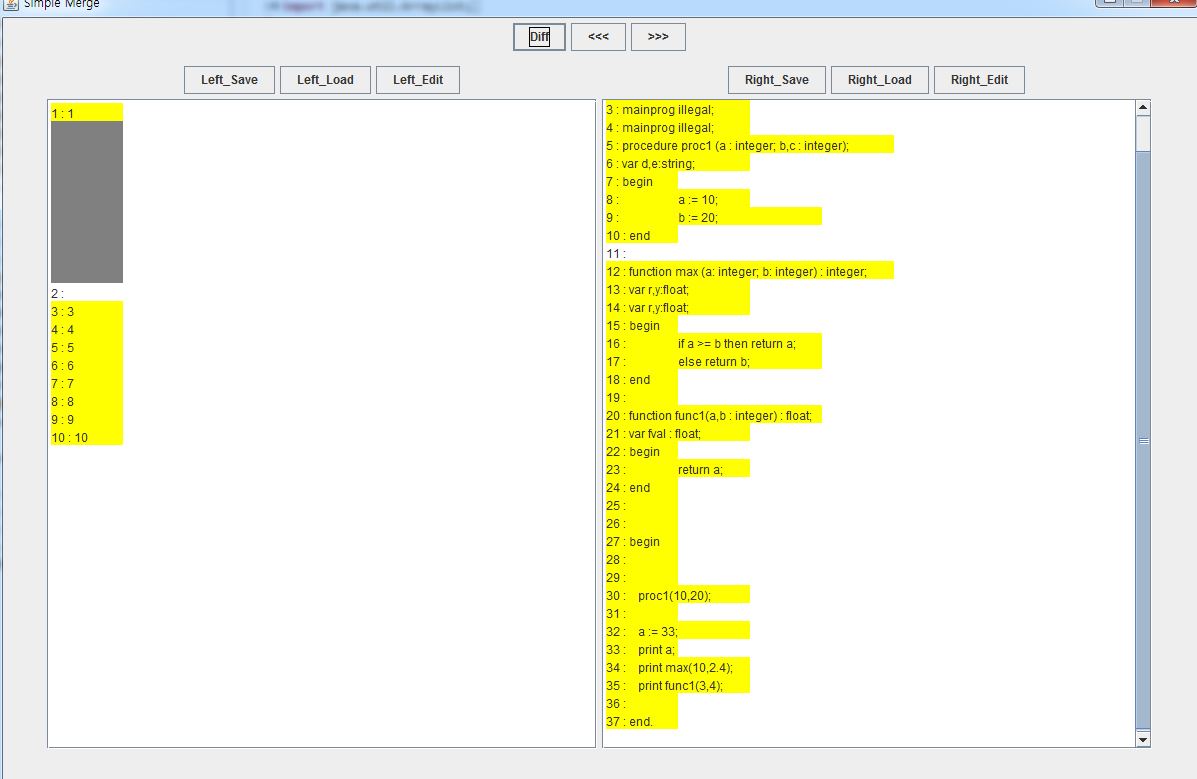
diff버튼을 실행하지 않으면 우측의 <<, >> merge버튼이 비활성화 된다.

Save 혹은 load 버튼을 클릭할 시 위와 같은 창이 나타난다.

이를 통해 파일을 불러오거나 저장할 수 있으며, 불러온 파일이 없는 채로 edit을 하여도 새 파일을 만들도록 하였다.

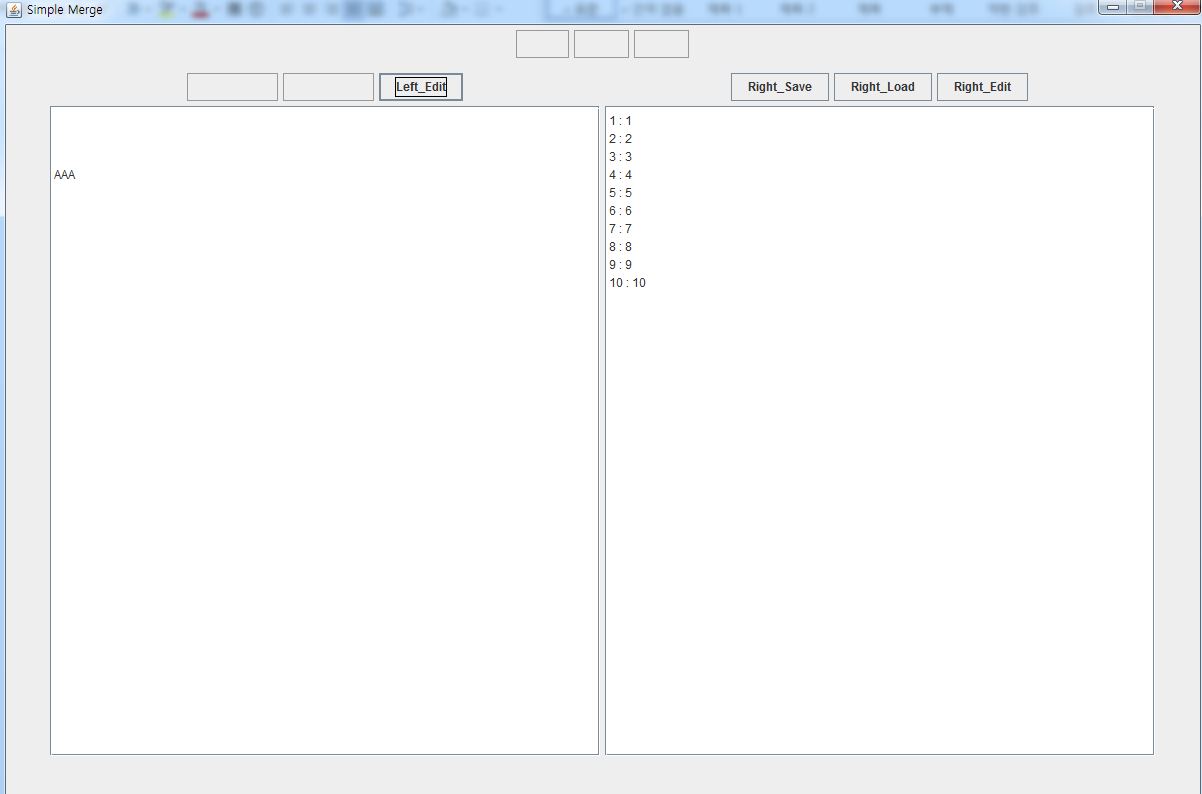
텍스트파일을 불러왔을 때의 화면

편집 및 비교의 용이를 위해 각 줄마다 줄 번호를 좌측에 표시하도록 하였다.



diff버튼을 실행했을 때의 화면

서로 내용이 다른 부분의 경우 노란색으로 표시되며, 반대쪽과 비교해서 없는 부분이 있을 경우 검은색으로 표시하도록 했다.

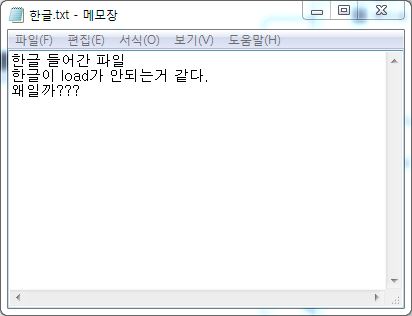


좌측 edit버튼을 눌렀을 때의 화면

Edit이 활성화되면 줄 번호가 사라지며 save, load, diff, merge버튼이 모두 비활성화 된다.

Edit버튼을 한번 더 누르면 edit 상태에서 빠져나온다.

1. Test
2. Functional Unit Test Cases
3. Bug Report



한글이 들어간 파일은 ‘load’가 안 된다.

한글이 들어간 텍스트 파일을 load를 시도해 본다.

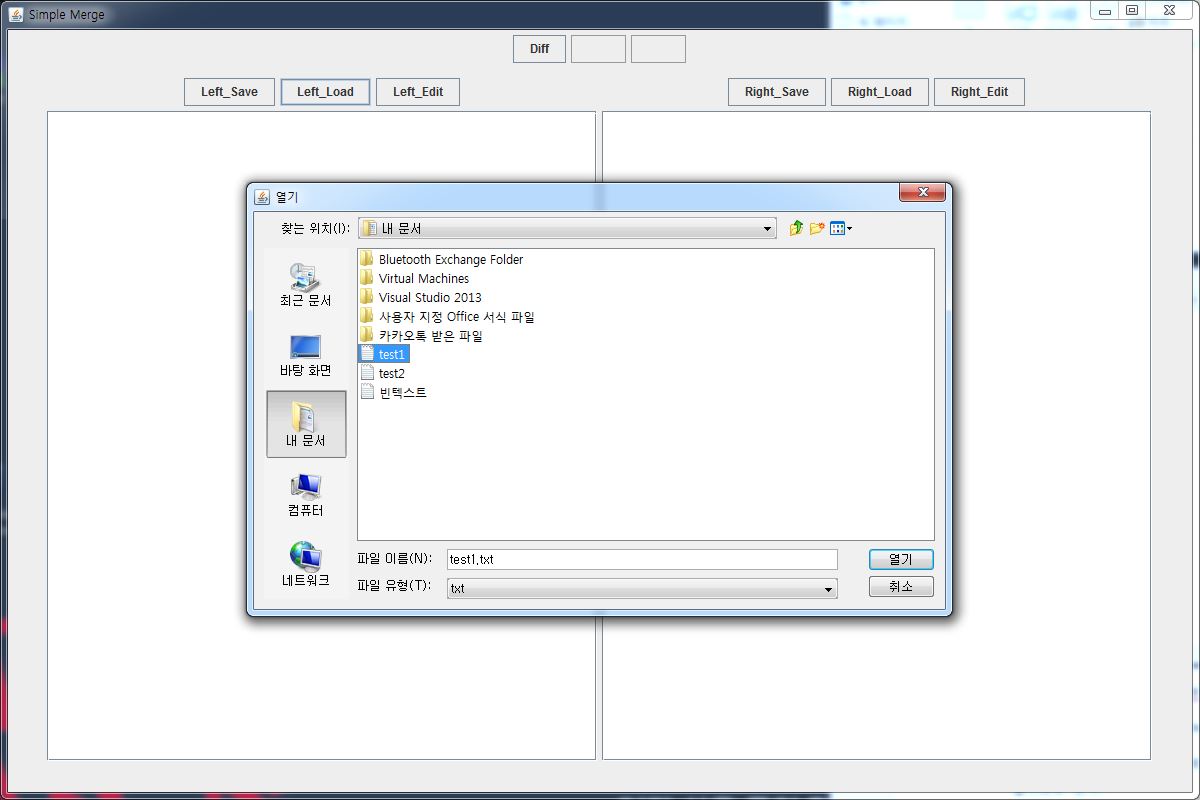
Load를 했는데 파일이 안 불려져 온다.

1. Execution Screen & Explanation

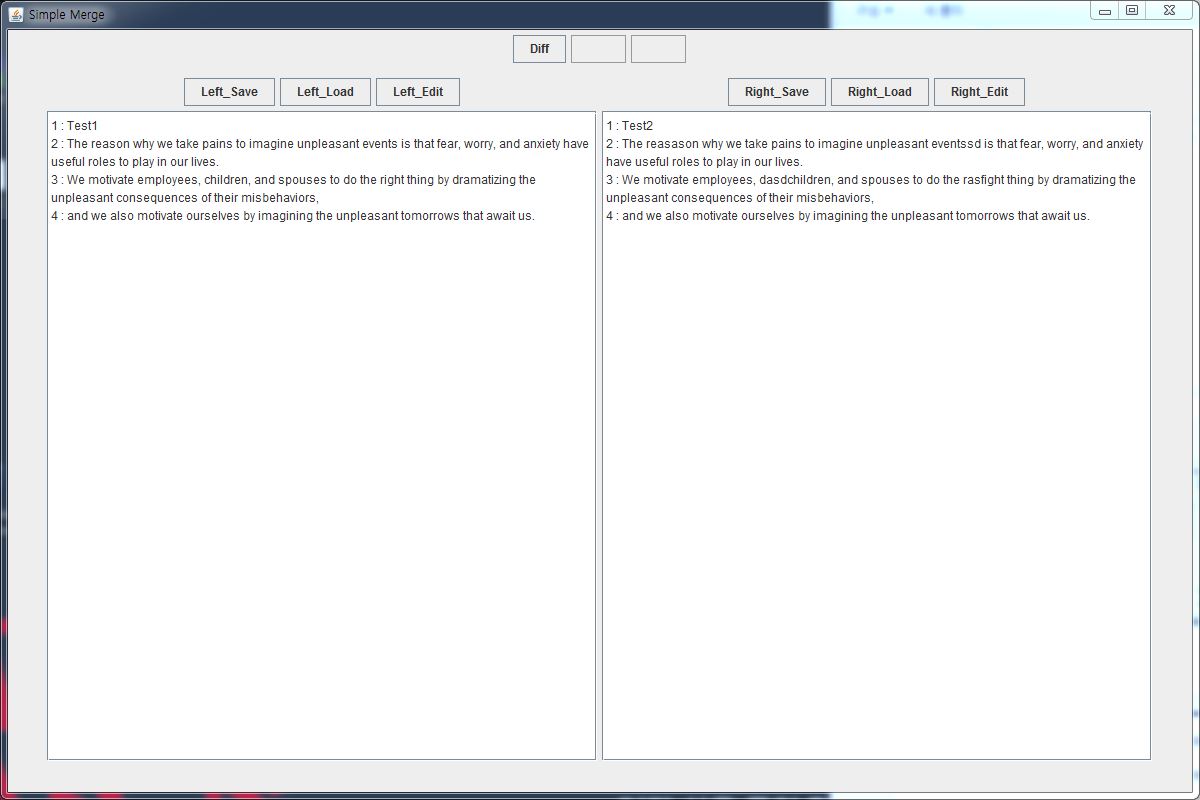
실행을 하게 되면 나오는 기본창

처음에 Diff가 안된 상태라 Merge키가 비활성화 되어있다.

이 상태에서 편집(Edit)을 누르고 패널에 직접 글을 써도 되고, Load를 이용해 텍스트 파일을 불러와도 된다.

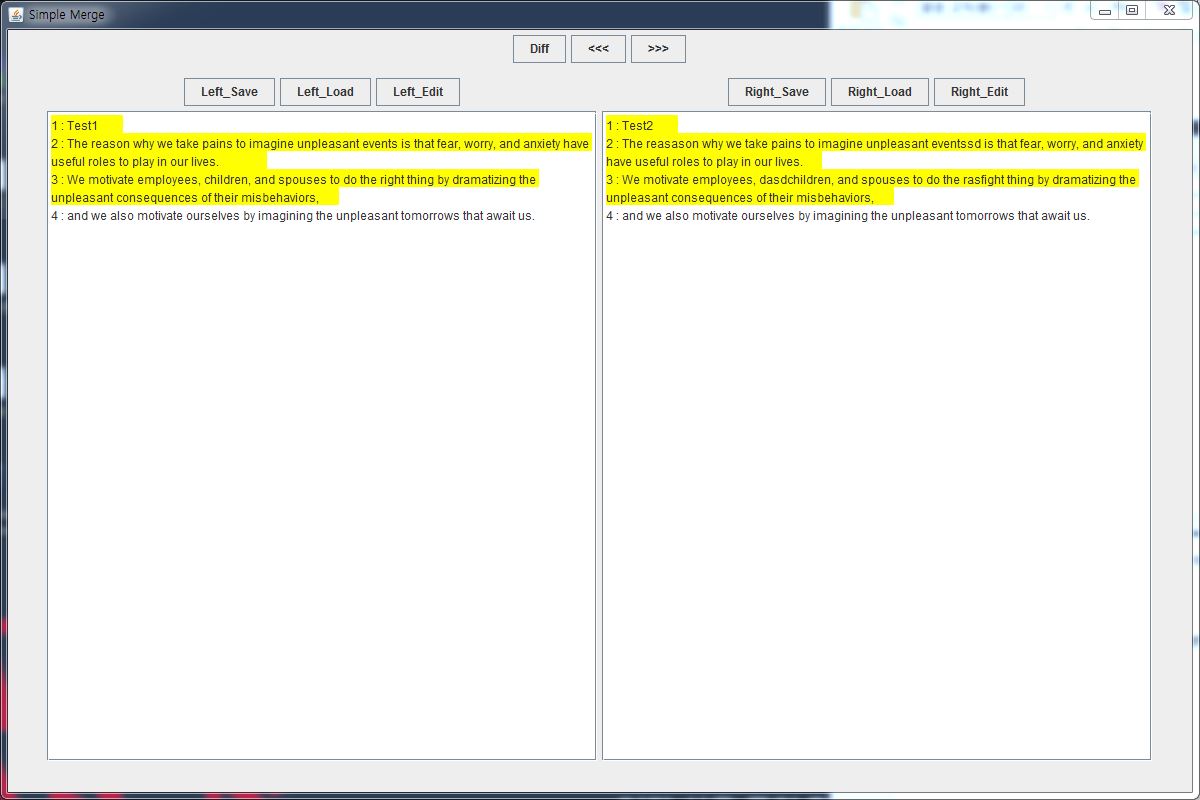


여기서는 저장된 파일을 불러와 보자.



왼쪽에는 test1파일을 오른쪽에는 test2파일을 불러왔다.

불러온 파일들



Diff를 눌러보았다.

왼쪽 패널과 오른쪽 패널에서 각 라인별로 일치하는 라인을 찾고, 불일치하는 라인은 노란색으로 색칠된다.

예시의 내용이 너무 극단적인거 같아서 글을 바꿔보자.

1. Project Management