**Final Report**Simple Merge

|  |  |
| --- | --- |
| **Team 11** | |
| **20151532강영은** | kye0925@naver.com |
| **20155445김규리** | kkyuree@nate.com |
| **20154648김준성** | keeka2@naver.com |
| **20164736박기철** | rlarla423@naver.com |
| **20150596이세영** | tpdud3131@naver.com |
| **20165584이융** | nngeet010@naver.com |

**Contents**

1. **Team Introduction**
2. **Program Description**

1) Summary

2) Main Functionality

1. **Development Environment**
2. Development OS
3. Development Tool
4. **Design & Implementation**
5. Development Process Model
6. View Design
7. Controller Design
8. **Execution Screen**
9. **GitHub Project Management**

**1. Team Introduction**

|  |  |
| --- | --- |
| Name | Role |
| 강영은, 김규리 | JUnit & EasyMock Test |
| 박기철, 박준영 | MVC, Implement Controller |
| 이세영 | Implement GUI |
| 이융 | Implement Content Service, algorithm |

**2. Program Description**

1) Summary

이 프로젝트는 ‘SimpleMerge’ 프로그램을 개발하는 것이다. ‘SimpleMerge’는 두 파일을 불러와 Viewing할 수 있고, 각 각의 파일들을 편집하고 저장할 수 있다. 두 파일을 비교해서 서로 다른 부분을 명시해줘서 사용자로 하여금 명확하게 차이점을 알 수 있도록 해준다. 또한, 사용자는 두 파일을 비교한 후 한 파일을 다른 파일로 Merge할 수 있다.

이 ‘SimpleMerge’ 프로그램은 JavaFX를 통해 구현했다. Testing Frameworks을 JUnit and EasyMock을 이용했고**,** GUI components들은 MVC architecture pattern을 이용하여 구현했다.

2) Main Functionality

(1) Functionality of viewing/editing/saving files

먼저, 프로그램을 실행하면 메인화면 맨 위에 "Load", "Edit", "Save" 버튼이 있다.

"Load" 버튼을 누르면, 파일을 찾아서 선택하여 edit panel에 load할 수 있어 Viewing이 가능하다.

"Edit" 버튼을 누르면, 사용자는 edit panel에서 파일을 편집할 수 있다.

"Save" 버튼을 누르면, 프로그램은 파일을 저장한다.

(2) Functionality of comparing two files

"Compare" 버튼을 눌러 두개의 파일을 비교할 수 있다. 이 버튼을 누르면, 두 개의 파일의 다른 문장들을 line by line으로 비교할 수 있다. 두 파일의 다른 부분은 다른 색으로 표시한다. 두 파일의 비교를 통해 추가되거나 삭제된 부분을 알 수 있다.

(3) Functionality of merging two files

“Merge”버튼을 눌러 두개의 파일을 병합할 수 있다. 비교한 후나 병합 후의 차이점 및 병합한 부분을 알 수 있다. 병합은 두 개의 버튼이 존재하는데, “Copy to Right” 와 “Copy to Left” 가 있다. 먼저 병합할 부분을 block으로 정한 뒤 “Copy to Right” 버튼을 누르면 왼쪽 panel에 있는 파일이 오른쪽 penal에 있는 파일과 병합된다. “Copy to Left” 은 그와 반대로 오른쪽 panel에 있는 파일이 왼쪽 penal에 있는 파일과 병합된다.

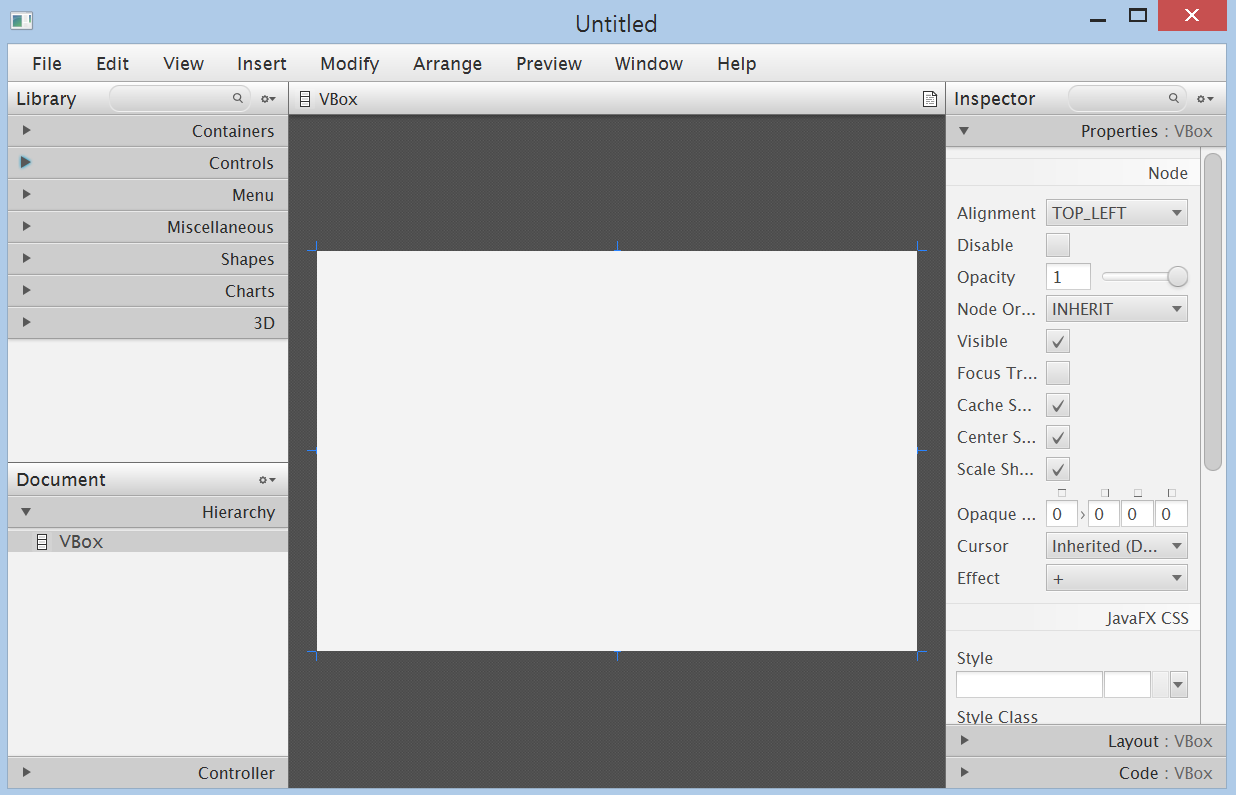
**3. Development Environment**

1)Development OS

개발은 windows8, windows10 환경에서 진행되었습니다.­

2)Development Tool

개발 도구는 eclipse를 사용하고 JAVA 언어로 구현했으며 사용된 JDK는 1.8.0\_111버전입니다. GUI는 javaFX의 fxml을 사용했으며 JavaFXSceneBuilder2.0을 사용하였습니다. 팀프로젝트 진행은 Github에서 코드를 공유하고 변경사항을 확인하며 작업했습니다.

scenebuilder 실행 화면

**4. Design & Implementation**

1)Development Process Model

저희는 development process model로 Waterfall Model을 사용했습니다. 우선 만들고자 하는 제품의 요구사항이 명확히 주어졌으므로 프로그래밍 하는 과정에서 큰 변화가 없을 것 이라고 생각했습니다.

먼저 계획하는 단계에서는 어떻게 프로그램을 만들지 조사하며 프로그래밍 언어, 개발 도구 등을 정했습니다. 요구 분석 단계에서는 SRS를 작성하여 제품의 전체적 기능과 설명, UseCase, 시스템 특징, 기능의 요구 사항, 외부 인터페이스 요구사항, 제한 조건 등을 정리하였습니다.

2)View Design

뷰는 JavaFX SceneBuilder2.0로 fxml파일을 열어서 디자인을 했습니다. 우선 화면을 크게 2개로 나눠줘야 하므로 VBox를 2개를 생성하여 pane을 만들고 각각 내부에 글을 받아오거나 쓸 TextArea를 생성하였습니다. 그리고 edit, load, save, merge, compare 등 기능이 되는 버튼을 만들고 각각의 버튼에 onAction을 설정하여 이벤트가 발생하면 컨트롤러에 전달할 수 있게 했습니다.

3)Controller Design

프로젝트의 규모가 크지 않다고 생각하여 컨트롤러는 하나만 만들었습니다. 뷰에서 onAction 이벤트가 발생하면 컨트롤러에 각각 해당하는 eventlistener로 들어가서 실행이 되게끔 했습니다.

**5. Test**

1) Functional Unit Test

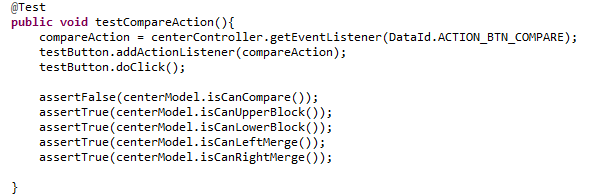
Controller의 경우 대부분의 private로 이루어진 메소드가 대부분이다. 따라서 private메소드를 호출하는 public메소드를 테스트한다. 케이스별로 버튼을 생성하고 ActionListener를 달아 메소드를 테스트 한다.

1. CenterControllerTest

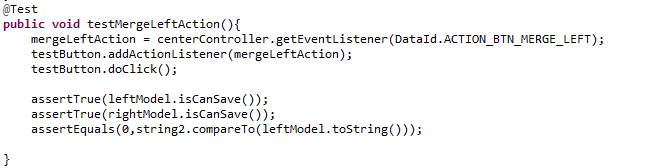
* 조건과 변수를 각각 초기화 한 후, 비교 action을 테스트한다. Left&right 모델로 각각 merge하는 테스트와 비교 후 인덱스를 이동시키는 액션도 테스트 하는 함수가 있다.
* 각각의 테스트를 모두 통과 하였다.



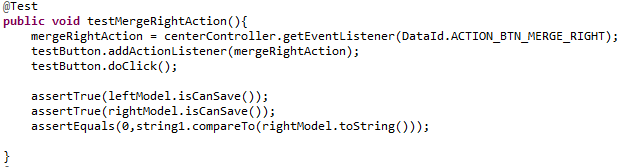
1. testConpareAction



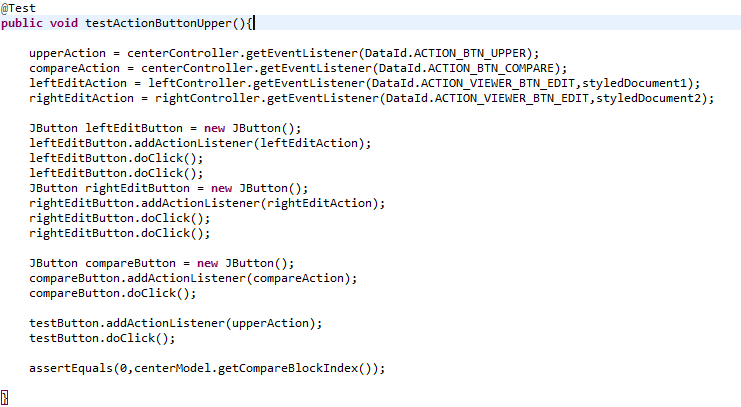
1. testMergeLeftAction



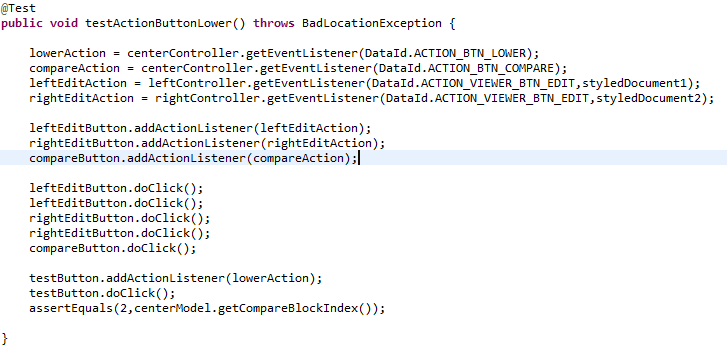
1. testMergeRightAction



1. testActionButtonUpper



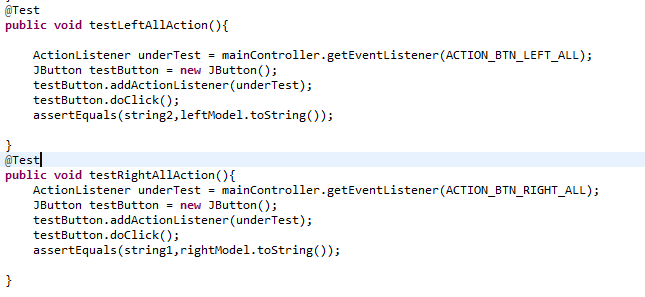
1. testActionButtonLower



1. MainController

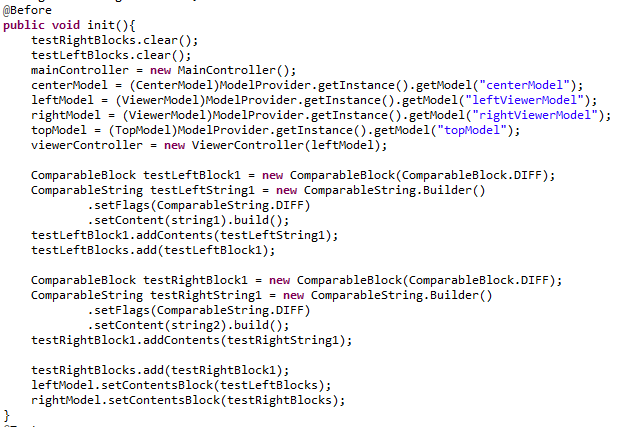
* 메인컨트롤러에서는 양모델의 수정이 끝난 뒤 비교한다음 왼쪽으로 / 오른쪽으로 merge하는 액션을 테스트 하였다.
* 모든 테스트를 통과 하였다.

1. testLeftAllAction / (2) testRightAllAction

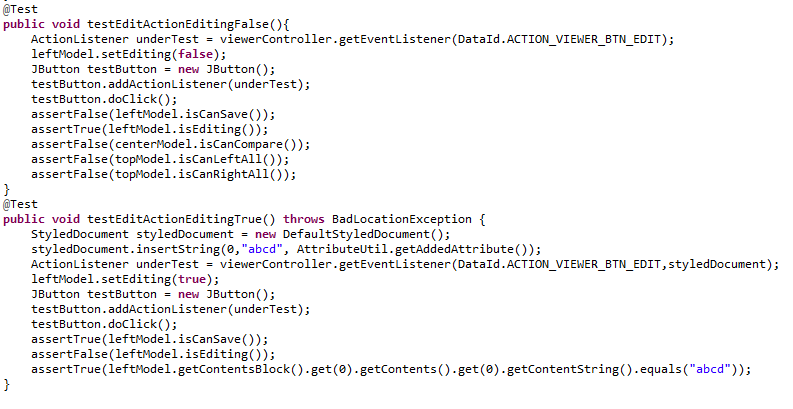


1. ViewController

* 수정정보를 테스트 한다. 이전의 두 테스트와 비슷한 방식으로 진행된다.
* 다른 모델의 상태를 잘 변화 시켰는지 테스트 하는 함수이다.
* 테스트를 통과 하였다.

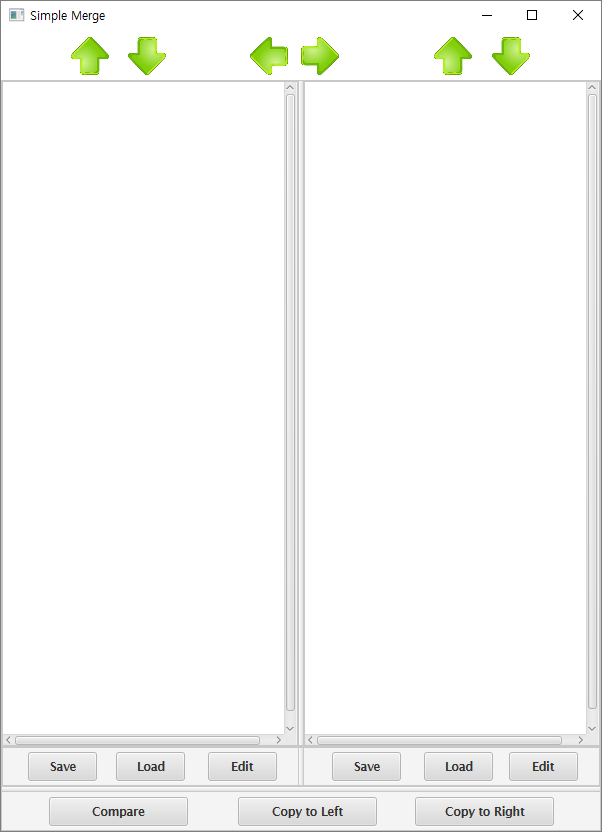


1. testEditActionEditingFlase/True



**6. Execution Screen & Explanation**

처음 실행 시켰을 때 화면 입니다



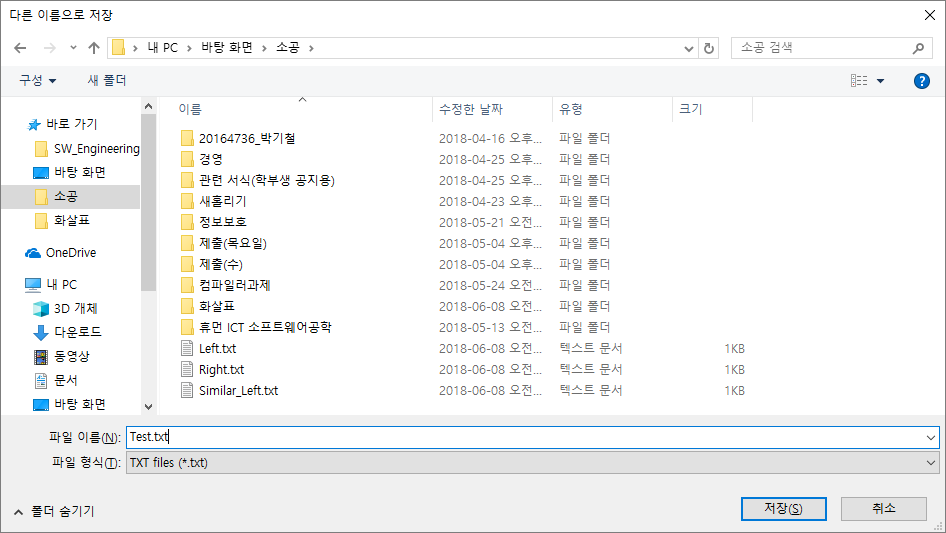
버튼

줄을 선택 할 수 있게 해줍니다. 위 방향 화살표 버튼을 누르면 선택하는 줄이 올라가고 아래 방향 화살표를 누르면 선택하는 줄이 내려갑니다.

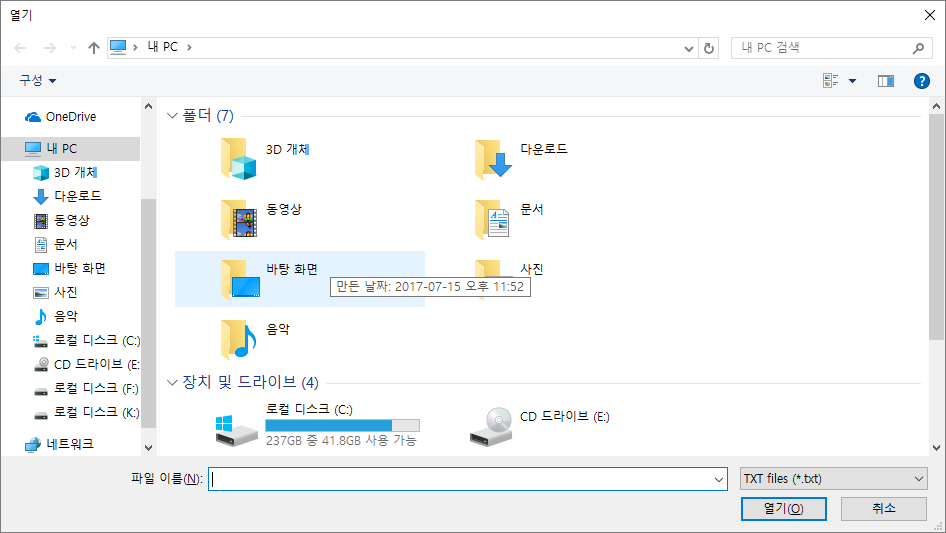
버튼

Copy to Left, Copy to right 버튼으로 누르면 선택한 줄이 선택한 방향으로 복사됩니다.

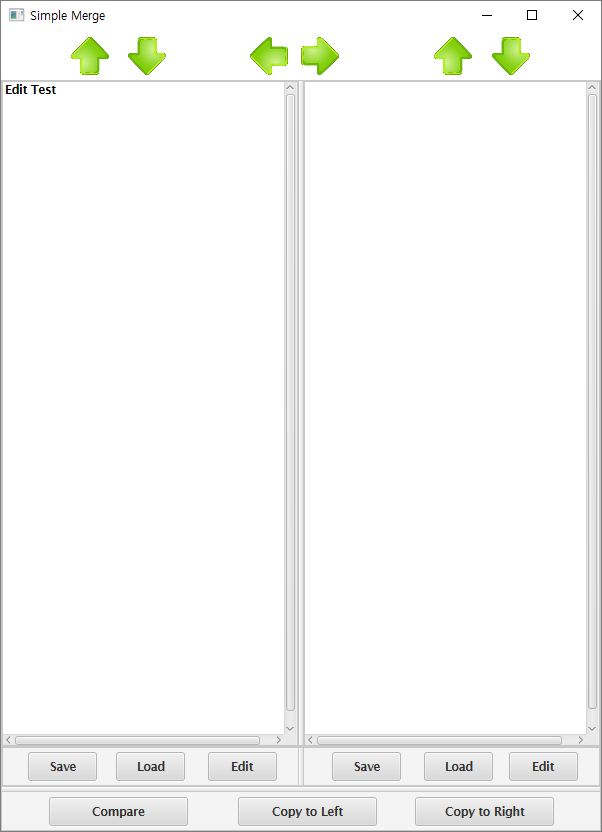
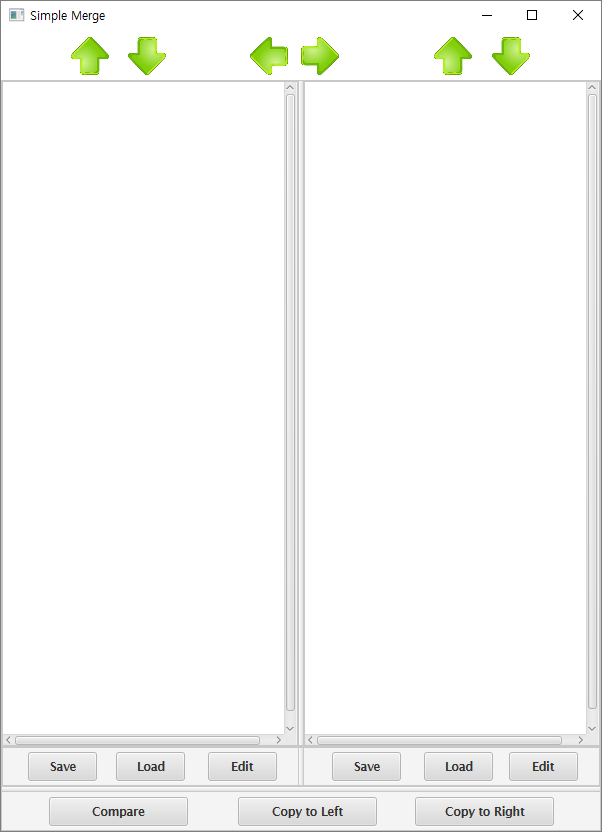
Save버튼-누르면 저장하는 화면이 아래와 같이 나옵니다. 파일 이름을 설정하고 원하는 위치에 저장을 시킬 수 있습니다.



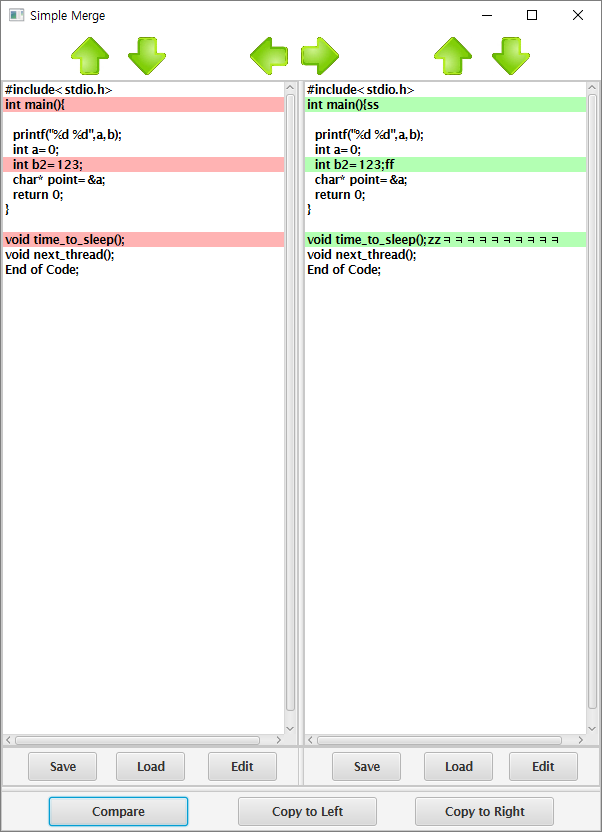
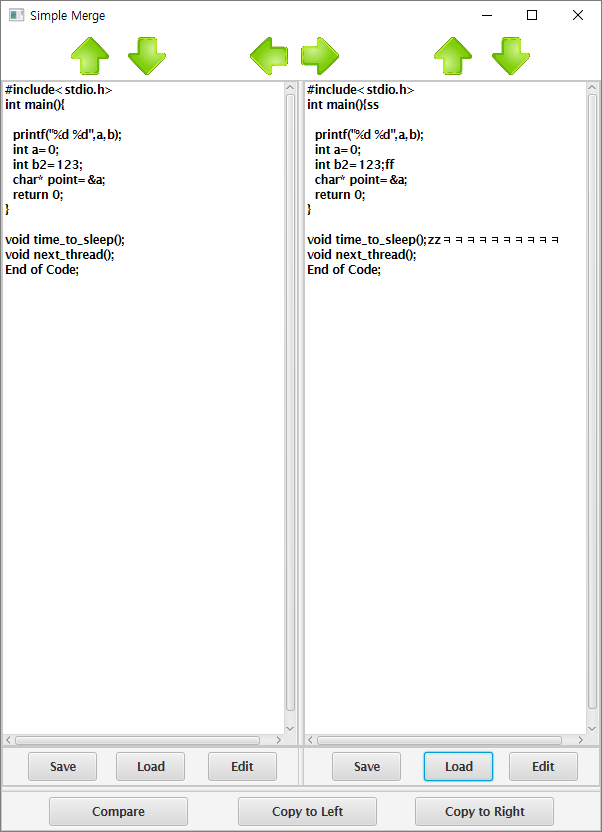
Load 버튼-누르면 아래와 같이 불러올 텍스트 파일을 선택할 수 있는 창이 나옵니다.



Edit버튼-누르면 글을 수정하거나 새로 쓸 수 있습니다.

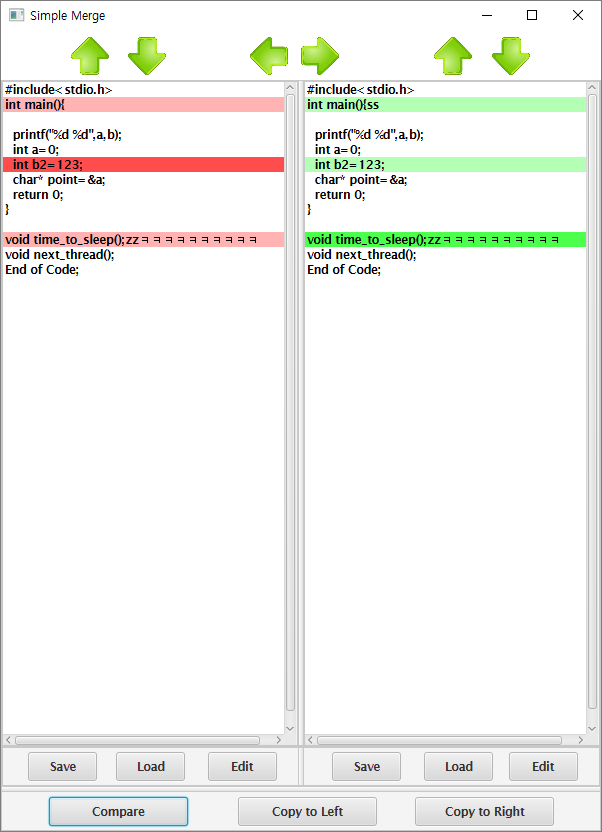
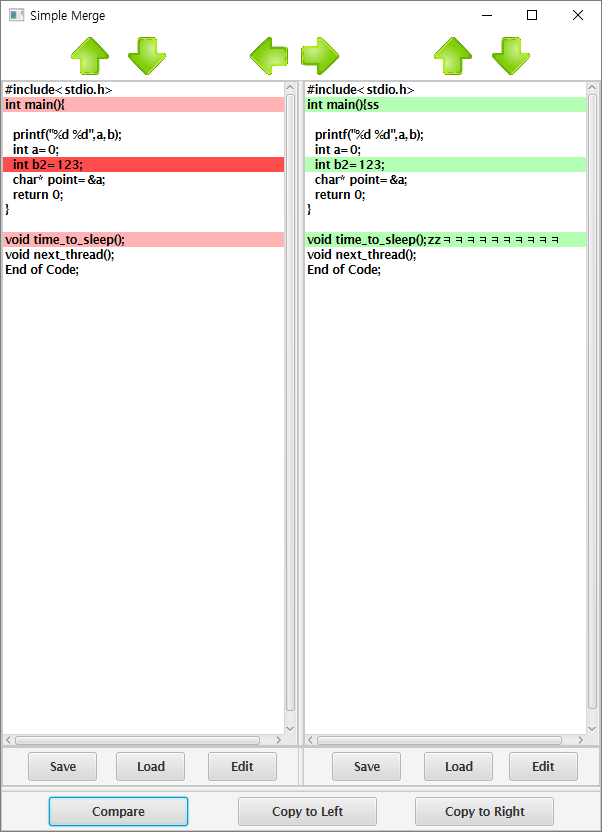


Compare버튼- 누르면 다른 부분이 색칠돼서 나타납니다.

****



Copy to right & Copy to left

****

11번째 줄에서 copy left 버튼인 왼쪽 화살표를 눌렀을 때 나타나는 화면입니다.

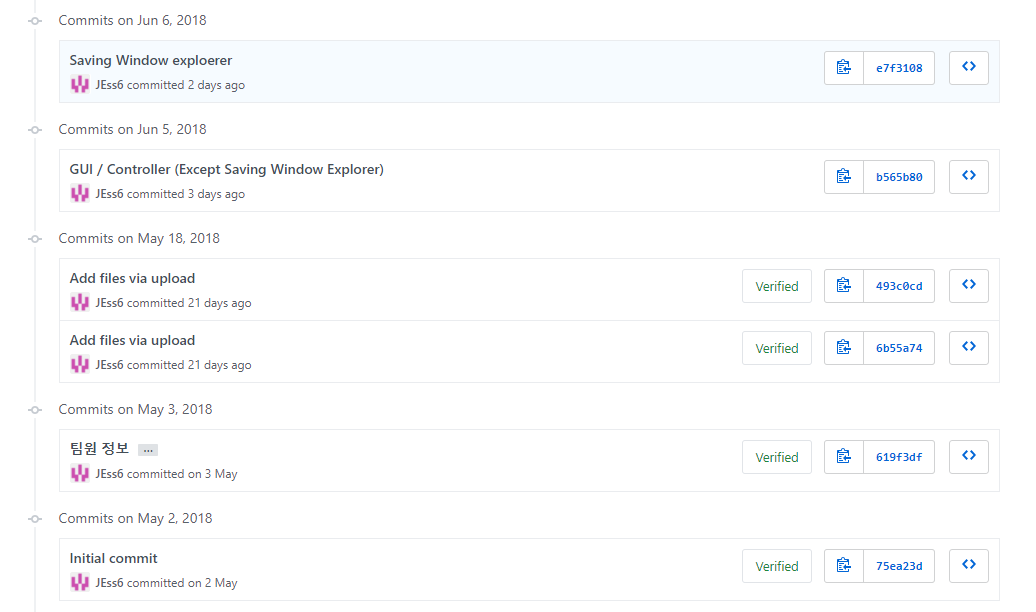
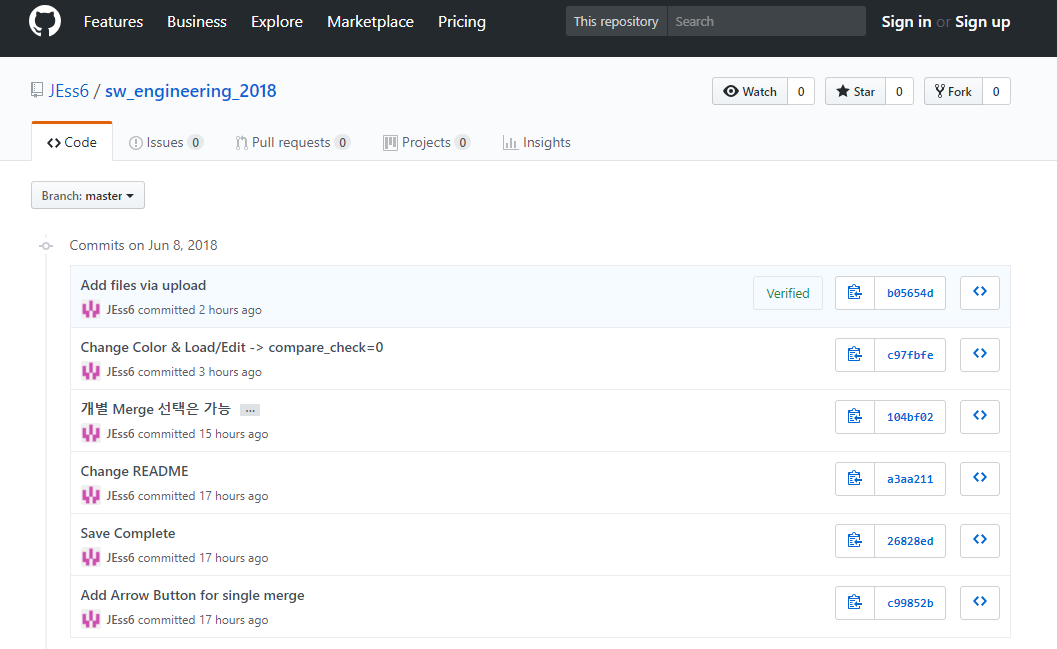
**7. Project Management**

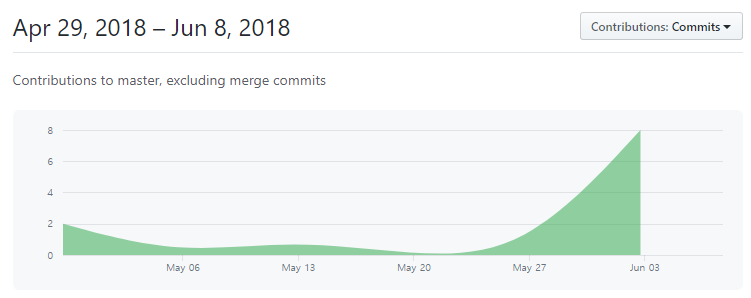
1) GitHub

GitHub address : <https://github.com/JEss6/sw_engineering_2018>

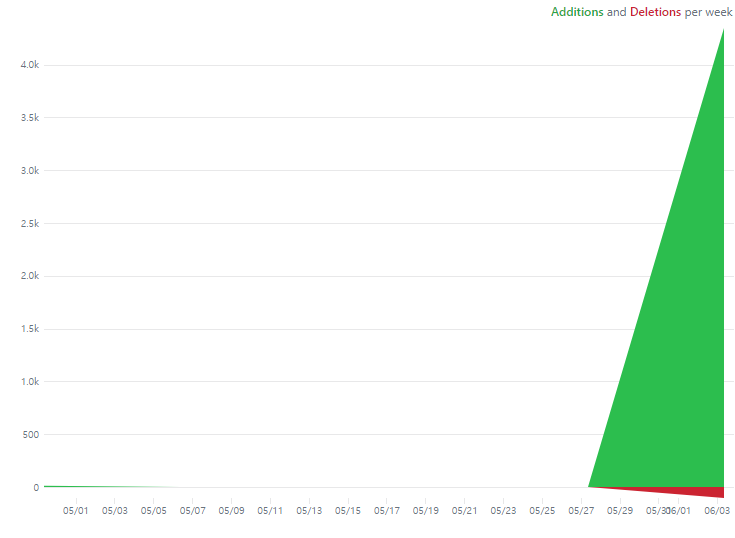
GitHub는 분산 버전 관리 툴인 Gut을 사용하는 프로젝트를 지원하는 웹호스팅 서비스로, Git이 텍스트 명령어 입력 방식인데 반해, 깃허브는 화려한 그래픽 유저 인터페이스(GUI)를 제공한다. 깃허브는 페이스트빈과 유사한 서비스인 Gist와 위키를 각 저장소마다 운영하고 있으며, 깃 저장소를 통해 고칠 수 있다.

2)Project Progress History by GitHub

(i)Commits

(ii)Contributors

(iii)Code frequency



3) Team Experience

|  |  |
| --- | --- |
| 강영은 | 처음으로 Git을 사용하게 되어 처음엔 사용법이 익숙하지 않아 많이 헤메었지만, 함께 작업하기 편리하다는 점이 소프트웨어 개발단계에서 정말 큰 이점으로 다가왔다. 이론으로만 배우던 소프트웨어 개발단계를 실제로 차근차근 밟아가며 과제를 진행하니 왜 그런 모델들을 만들었는지 비로소 이해가 되었다. |
| 김규리 | 이번에 프로젝트를 진행하면서 Git을 알게 되었고 이를 사용하는 것이 편리하다는 것을 알게 되었다. 소스코드들을 체계적으로 관리하여 동기화돼서 작업할 수 있다는 점에서 좋았다. 이번 소프트웨어공학 수업을 수강하면서 배웠던 것 MVC Concept을 직접 적용해보고 응용해볼 수 있어서 좋은 경험이었다. 이번 프로젝트를 통해서 협업을 배웠고 차후에 취업을 하여 프로젝트를 진행할 때 큰 밑거름이 될 것이라 생각한다. |
| 김준성 | 여러 사람과 한 프로그램을 만들 때는 소통을 하는 것이 가장 중요하다는 것을 알게 되었습니다. |
| 박기철 | Git을 처음 사용해보았는데 협업은 마냥 힘든 것인 줄만 알았던 나의 생각을 바꿀 수 있는 계기가 되었다. 따로 코드를 합칠 필요 없이 한번에 코드를 비교해서 합쳐주는 git덕분에 협업을 보다 수월하게 할 수 있었다.  또한 매번 사용하던 Swing이 아닌 JavaFx을 사용하여 GuI를 구현하였는데, 보다 직관적이고, 편리하게 GUI를 작성할 수 있었던 것 같다. |
| 이세영 | 소프트웨어 프로젝트를 통해 협력과 소통을 알게 된 것 같다. 다음엔 더 많은 도움이 될 수 있도록 열심히 공부해서 노력해야겠다. |
| 이융 | 이번 소프트웨어 공학 프로젝트를 진행하면서, 그 전에 했던 팀플들에 비해 git을 이용하여 보다 체계적이고 정교하게 프로그래밍을 하게 된 계기가 되어 좋은 경험이었고, 개인 프로젝트 할 때의 자신만 이해 할 수 있는 코드를 작성 하는 것이 아니라, 모두가 이해 할 수 있는 코드를 작성해야 하기 때문에 난잡하게 프로그래밍 하는 습관을 줄이게 된 계기가 되어 좋았다. |