GitHub project management report



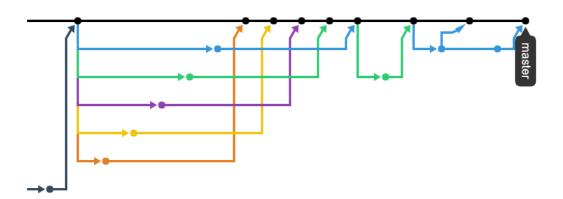
이경규 20123217 최종윤 20124055 박준영 20125753 지영민 20133138 조성욱 20133191 김정진 20133539

* Address of github project repository

- https://github.com/derekjychoi/SoftwareProject

* Project progress history

- Simple Merge 프로그램을 크게 GUI 부분, Compare 부분, Merging 부분으로 나누어 작업하였습니다. 작업 도중 Compare 기능 구현을 맡은 팀원들이 기본적인 GUI를 만들어야 할 필요가 있어서 이 부분을 함께 작업했습니다. 먼저 기본 기능을 갖고있는 GUI 와 Compare 를 GitHub 에 Commit 하였습니다. 이후 Merging 기능을 구현할 때 기존 Commit 한 소스가 구조적으로 MVC 모델에 맡지 않는 것을 확인하였고 GitHub 에 올린 소스를 MVC 모델에 맞추어 GUI를 view package 에 넣어서 구조화 하였고 기존 Component package 에 일부 GUI 기능과 Compare 기능이 합쳐 있었는데 이를 새로 구조화 하여 작업하였습니다. 이 작업후 Model(Data.java) / View(MainJFrame.java, LoadJFrame.java, SaveJFrame.java) /Controller(controller.java) 로 모델링하였습니다. View 부분을 GitHub 에 Commit 하고 Merge 구현 후 마지막으로 최종 Commit 하였습니다.



- 위 사진에서 맨 처음 Git 프로젝트에 Commit 한 부분이 최초 GUI와 Compare 입니다. 이후 MVC 모델에 맞추어 완전히 다른 구조로 소스 구조가 바뀌어 이를 작업 후 주황색 라인부터 하늘색 라인까지 Model(Data.java) /

View(MainJFrame.java, LoadJFrame.java, SaveJFrame.java) /Controller(controller.java) 파일을 새로 올렸습니다. 처음 Commit 한 소스와는 구조가 달라지면서 파일 자체가 새로워져서 우려했던 Conflict 는 발생하지 않았습니다. 마지막으로 Compare 와 Merge 를 테스트하면서 크고 작은 버그와 오류들을 수정하여 마지막 최종 작업을 마무리 하였습니다.

* Experience

- GitHub으로 프로젝트 작업 중 나오는 여러 자바 파일의 버전들을 관리하여야 하는데 초기 작업 결과물에서 MVC 모델이 맞지 않아 거의 새로 작업하는 수준으로 파일이름과 내용, 구조가 변경되었습니다. 이 과정에서 GitHub으로 버전관리를 하는 장점을 많이 챙기지 못하였습니다. 하지만 이번 프로젝트를 GitHub와 연동하여 진행하면서 각자 작업한 소스파일을 공유하여 파일들이 중구난방으로 돌아다니는 것이 아니라 한군데 모아서 작업을 할 수 있어서 작업면에서 발생할 수 있는 혼동이 많이 줄었다고 생각합니다.