МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Донецкий национальный технический университет»

Факультет ИСП

Кафедра ПИ им Л.П.Фельдмана

Лабораторная работа № 3

на тему: «Восстановление данных из ранних ревизий»

по дисциплине «Профессиональная практика программной инженерии»

Выполнил: студент группы ПИ-20а

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Синяева Э.В.

(подпись) «\_\_\_» \_\_\_\_\_\_\_2021г.

Принял:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Незамова Л.В.

(подпись) «\_\_\_» \_\_\_\_\_\_\_2024г.

2024г.

**Цель работы:** получить практические навыки в совместной разработке программного средствами системы контроля версий Git, а также использование инструмента восстановления предыдущих ревизий.

1. Демонстрация целостности данных:

Переключиться на ветку module1 с помощью команды **git checkout module1**

Команда **git log --oneline** выводит краткий список коммитов в репозитории, отображая только их хеши и сообщения коммитов на одной строке каждый (Рисунок 1 и Рисунок 1.1)

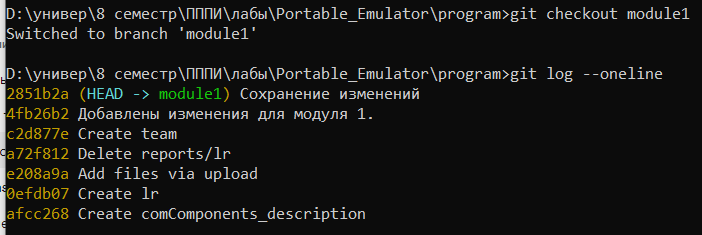
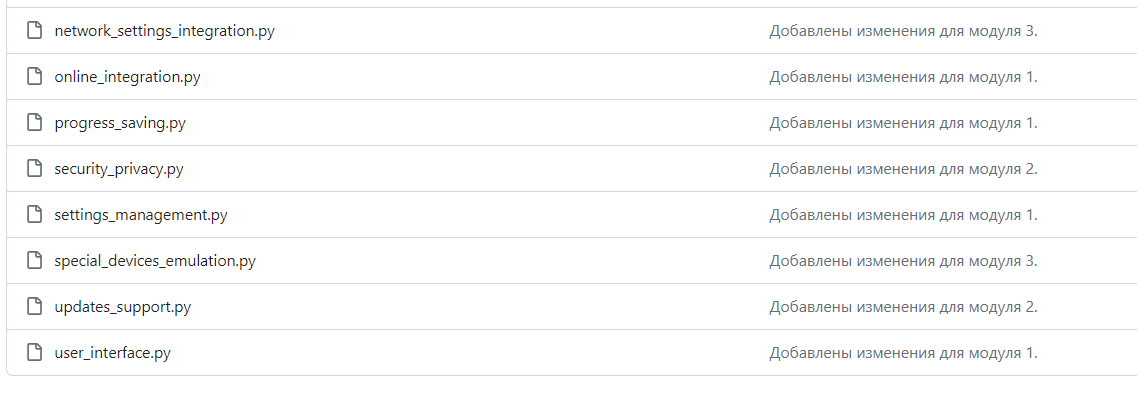
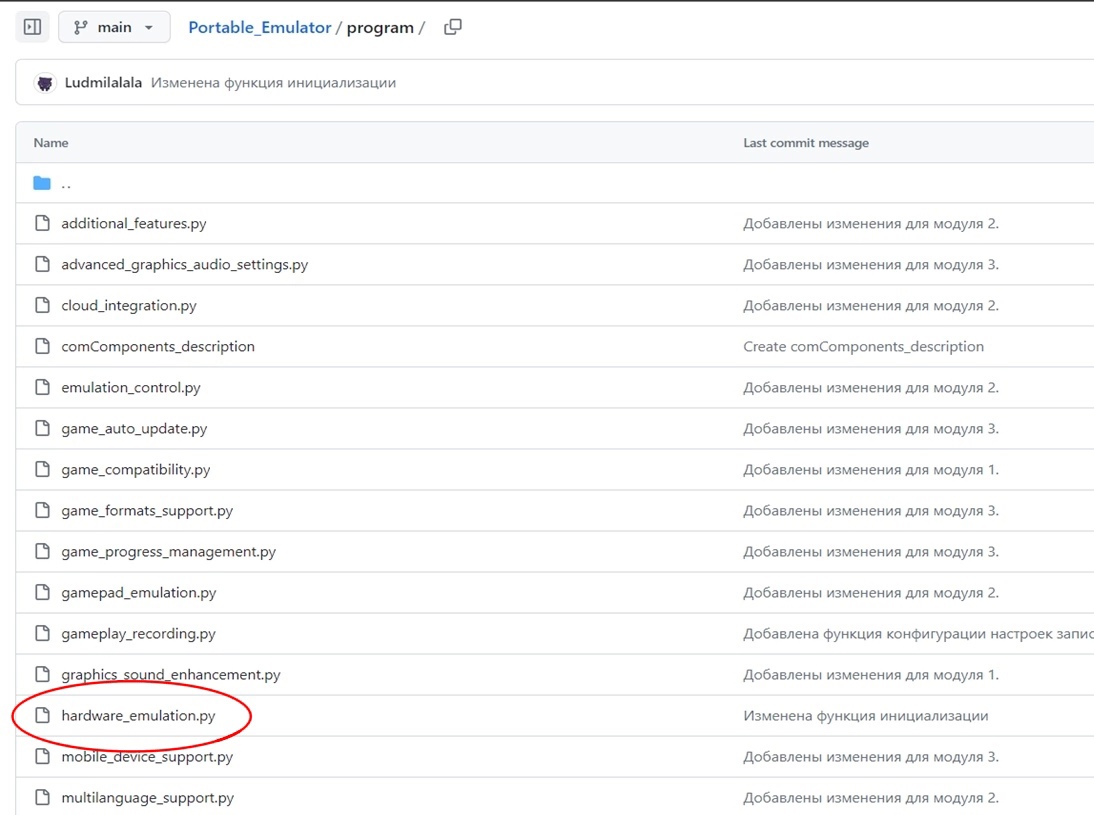


Рисунок 1 - Результат целостности данных

Рисунок 1.1 - Результат целостности данных

1. Внесение непоправимых изменений:

Вносим непоправимые изменения в проект, удаляя в ветке module1 важный файл hardware\_emulation.py. Данный файл содержит код, отвечающий за эмуляцию аппаратной части системы. Удаление этого файла может серьезно повлиять на функциональность всего модуля и привести к ошибкам в работе программы.

Для этого используется команда **git rm hardware\_emulation.py**, которая удаляет файлы из индекса Git и рабочего дерева. Это вызвало потерю важного компонента программы и нарушение целостности данных в репозитории. После удаления файла мы фиксируем изменения в репозитории с помощью следующих команд:

**git commit -m "Удаление файла hardware\_emulation.py"**: Фиксирует изменения в репозитории с комментарием "Удаление файла hardware\_emulation.py".

Таким образом, файл hardware\_emulation.py будет удален из проекта и зафиксирован в истории коммитов (Рисунок 2, Рисунок 2.1).

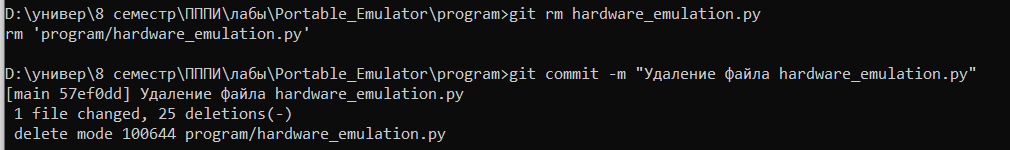


Рисунок 2 – Удаление файла hardware\_emulation.py

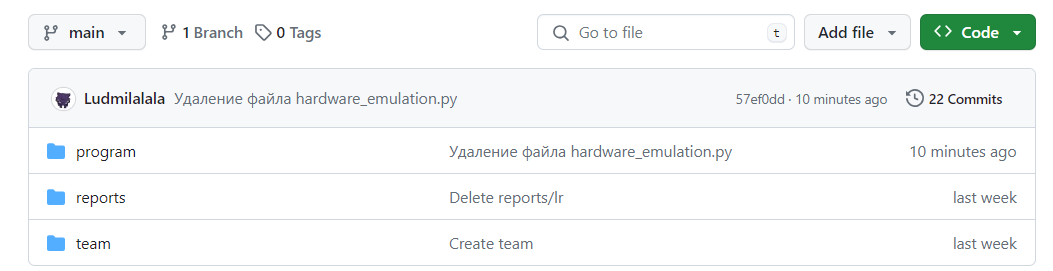
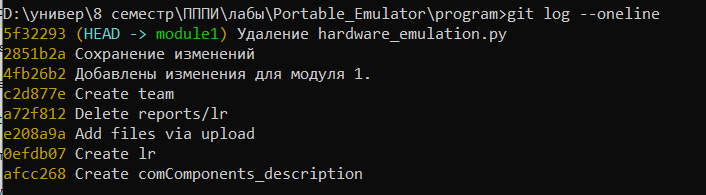


Рисунок 2.1 – Удаление файла hardware\_emulation.py

1. Демонстрация поврежденных данных

Для этого используется команда **git log --oneline**, которая выводит историю коммитов в сокращенном формате, отображая только хеш коммита и его сообщение. После удаления файла hardware\_emulation.py и фиксации этого изменения, коммит, отражающий это действие, будет присутствовать в истории коммитов (Рисунок 3 и Рисунок 3.1).



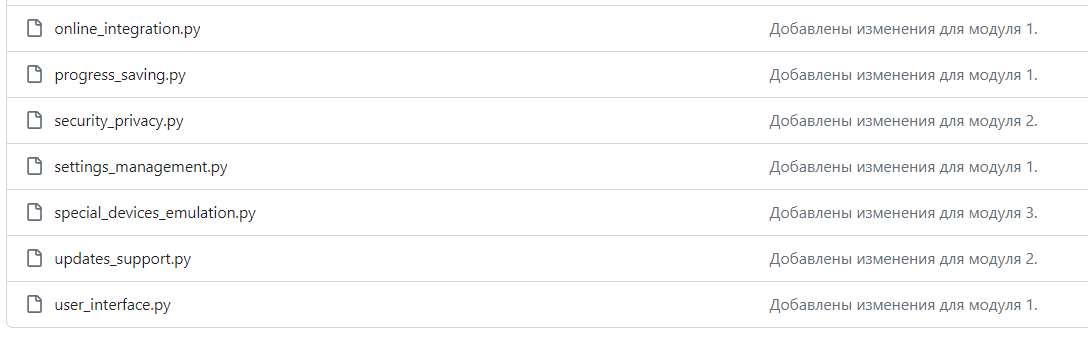
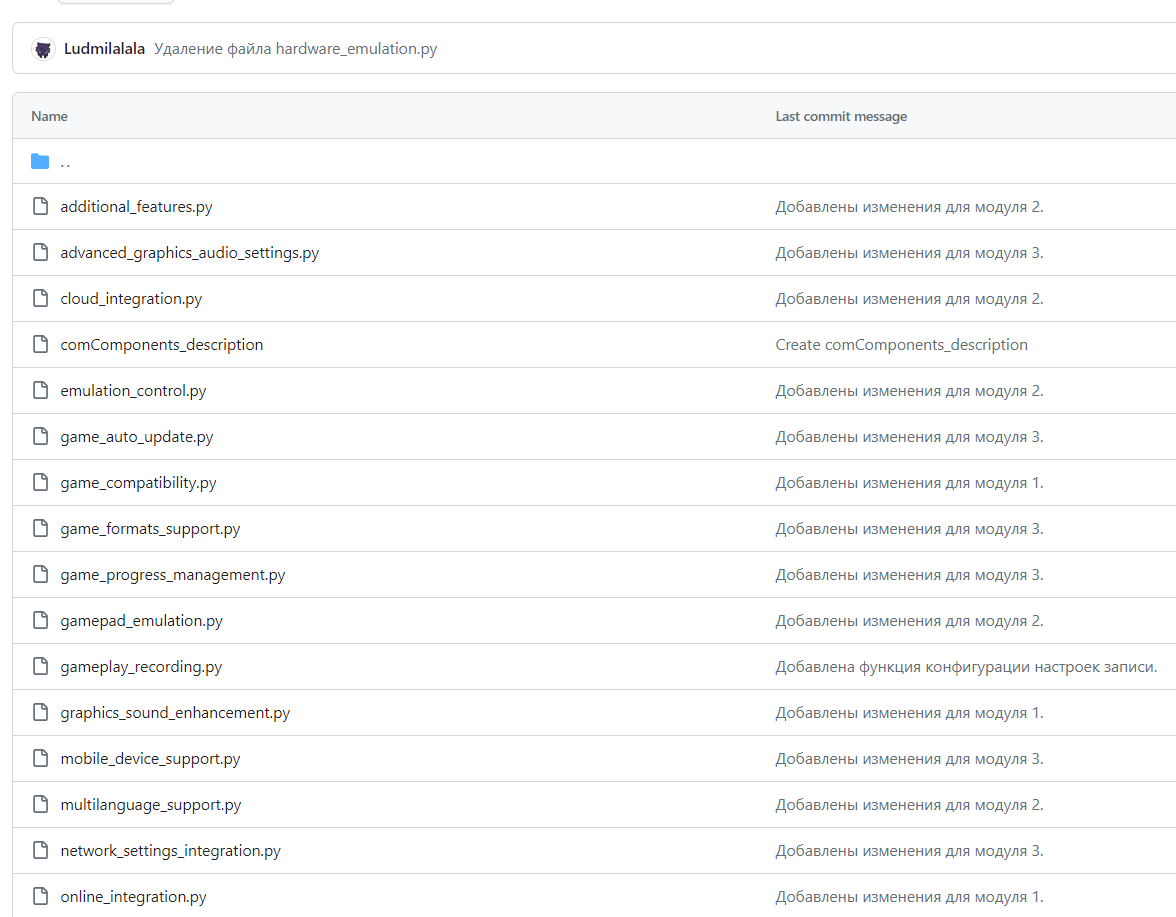
Рисунок 3 – История коммитов

Рисунок 3.1 – Отсутствие файла hardware\_emulation.py

1. Восстановление предыдущей ревизии:

Для восстановления предыдущей ревизии используем команду **git reflog**, которая позволяет просмотреть историю перемещений указателя HEAD и действий, совершенных в репозитории. Находим хеш предыдущего коммита, который указывает на состояние репозитория до удаления файла (Рисунок 4).

Нужный хеш 2851b2a

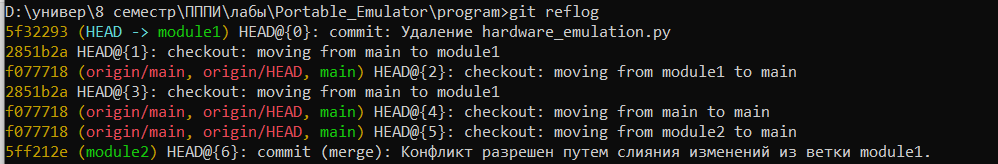


Рисунок 4 – История совершенных действий

После того как найден нужный хеш коммита, выполняем команду **git reset --hard <хеш\_коммита>**. Эта команда сбрасывает HEAD и текущую ветку на указанный коммит, удаляя все изменения, сделанные после этого коммита. В результате файла hardware\_emulation.py восстанавливается в репозитории к состоянию, которое было до его удаления (Рисунок 5, Рисунок 5.1)

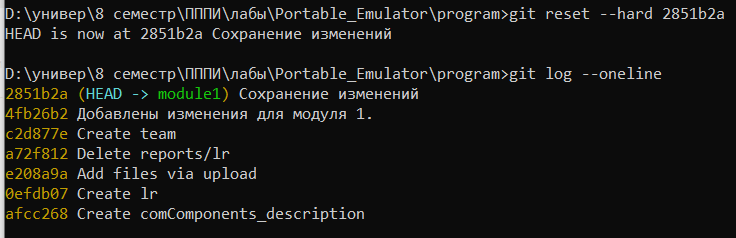


Рисунок 5 – Удаление изменений и восстановление файла

Рисунок 5.1 – Восстановление файла

