Міністерство освіти і науки України

Національний університет «Запорізька політехніка»

Кафедра програмних засобів

Лабораторна робота № 1

з дисципліни: «**Якість програмного забезпечення**»

на тему: **«Основи GIT»**

Виконав:

ст. групи КНТ-126 В.Л. Самойлов

Прийняла:

Т.І. Каплієнко

2019

1. **ЛАБОРАТОРНА РОБОТА № 1**

**«Основи GIT»**

**1.1 Мета роботи**

1.1.1 Вивчити основні команди по роботі з локальним репозиторієм GIT

**1.2 Завдання на виконання лабораторної роботи**

**Варіант 2**

Ініціалізувати каталог з майбутнім проектом в git. Створити файл .gitignore з описом ігнорованих файлів (\*. [oa], \* ~). Користувач вводить свої даних, створює папку проекту, створює в ній файл із завданнями. Виконує комміт для фіксації внесених змін. Створює нову гілку, відкриває файл і виконує завдання, виконує комміт. Повертається в гілку майстер, додає аліаси для branch, checkout, status, history. За допомогою алиасів переглядає всі гілки проекту, його поточний статус та історію коммітів.

**1.3 Виконання лабораторної роботи**

Спочатку ініціалізуємо каталог з майбутнім проектом в git. Та створемо файл .gitignore з описом ігнорованих файлів (\*. [oa], \* ~).

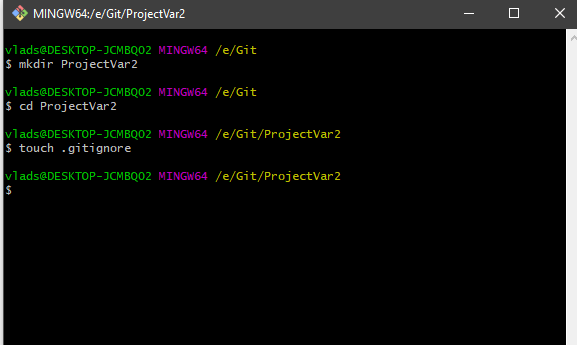


Рисунок 1.1 – Створюємо каталог з майбутнім проектом та файл .gitignore

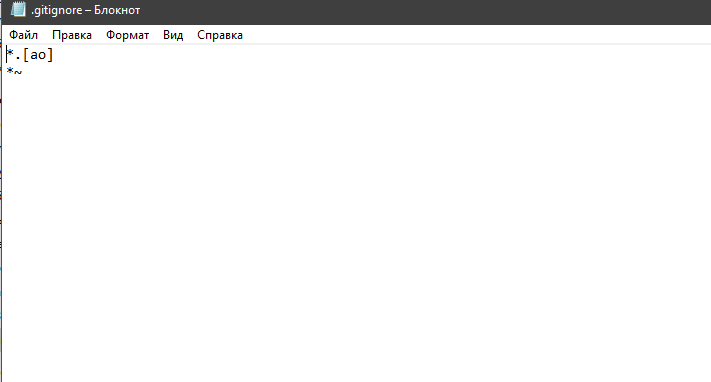


Рисунок 1.2 – Файл .gitignore з описом ігнорованих файлів

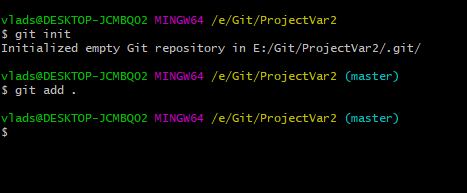


Рисунок 1.3 – Створюємо репозиторій та додаємо у нього файли

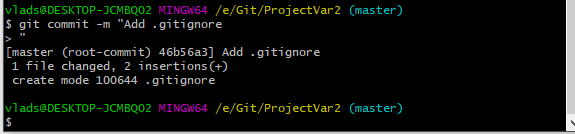


Рисунок 1.4 – Робимо комміт

Потім водимо дані користувача (його ім’я та електронний адрес). А також створимо папку проекту, а в ній створюємо файл з переліком завдань.

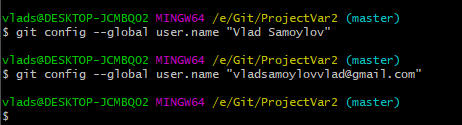


Рисунок 1.5 – Водимо дані користувача

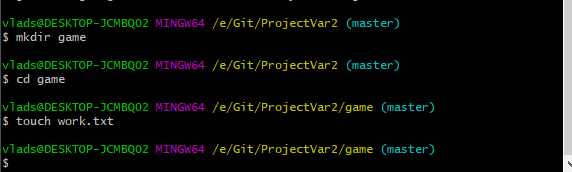


Рисунок 1.6 – Створюємо папку проекту та файлу з завданням

Далі робимо комміт для фіксації внесених змін.

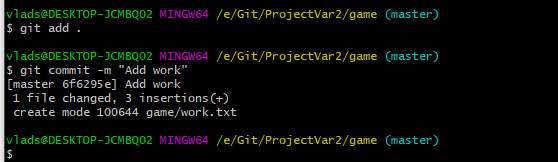


Рисунок 1.7 – Робимо комміт

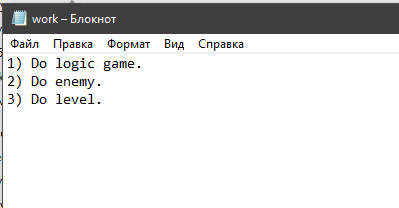


Рисунок 1.8 – Файл, що містить перелік робіт

Наступним кроком буде створення нової гілки, виконання завдання та виконання комміту для фіксації змін.

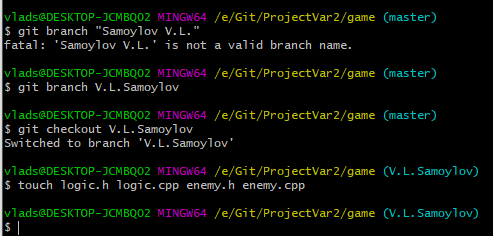


Рисунок 1.9 – Створення нової гілки та переходу до неї

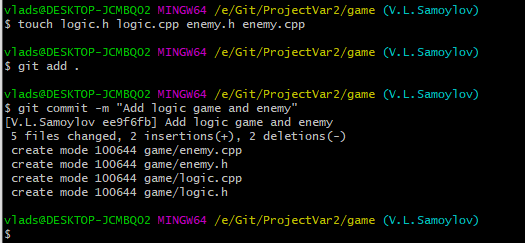


Рисунок 1.10 – Виконання завдання та комміту

Останнім кроком буде перегляд, за допомою алиасів всіх гілок проекту, їх поточного стану та історію коммітів.

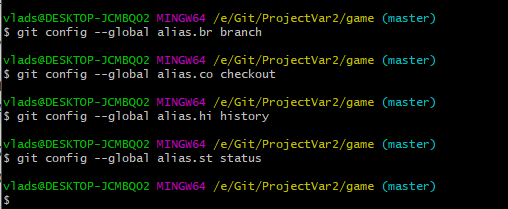


Рисунок 1.11 – Додавання алиасів

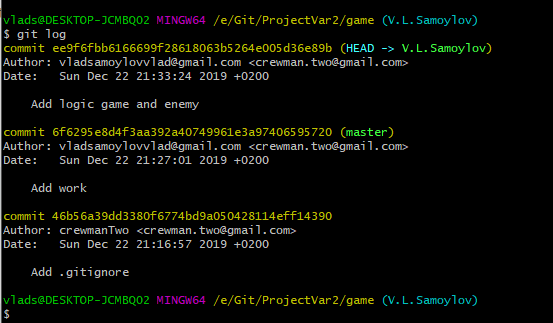


Рисунок 1.12 – Перегляд історії коммітів

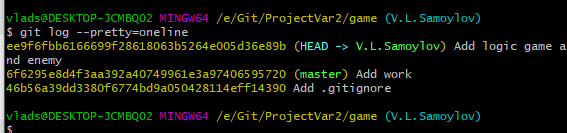


Рисунок 1.13 – Перегляд історії коммітів, використовуючи одностроковий формат

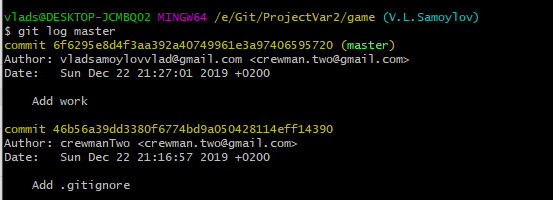


Рисунок 1.14 – Перегляд історії коммітів в гілці master

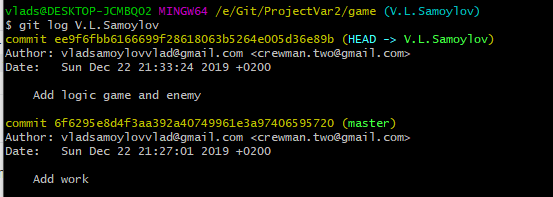


Рисунок 1.15 – Перегляд історії коммітів в гілці V.L.Samoylov

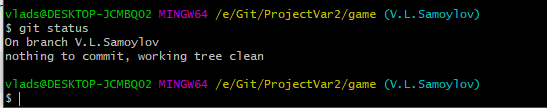


Рисунок 1.16 – Використання команди status в гілці master

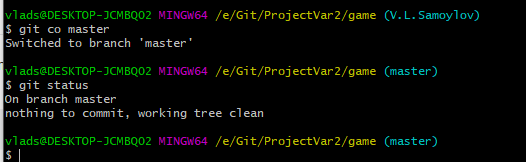


Рисунок 1.17 – Використання команди status в гілці V.L.Samoylov

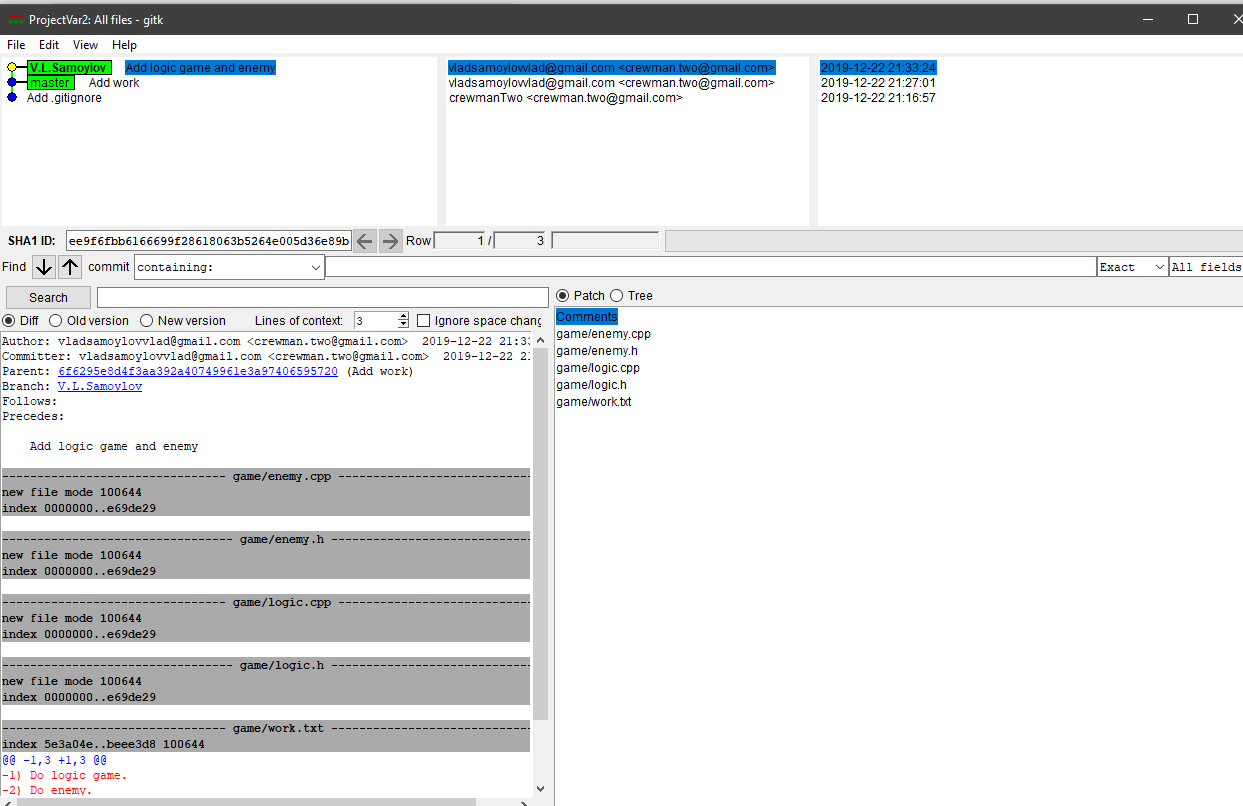


Рисунок 1.18 – Використання команди gitk

* 1. **Висновки**

Вивчили основні команди по роботі з локальним репозиторієм. Щоб створити гіт репозиторій потрібно використати команду “git init”. Щоб додати під версійний контроль усі файли треба використати команду “git add .”. Файли в репозиторії Git можуть знаходитися у трьох станах: зафиксований, змінений, підготовлений. Файл .gitignore створюється для опису ігнорованих файлів. Командою “git commit” виконується фіксація змін. Для видалити файл з Git-репозиторію треба використати команду «git rm». Щоб переглянути історію коммітів треба скористатися командою “git log”. Команда “git branch” створює нову гілку проекту. Щоб об’єднати гілки треба скористатися командою “git merge”.

Git потрібен для того, щоб вести зручну спільну розробку, також вона додає можливість повернутися до попередніх версій проекту та надає можливості відстежити помилки.