Github Desktop

Дата выполнения: 06.12.2025
Выполнил: Калинин Игорь Вячеславович
Группа: ИВТ-2

1. Введение

GitHub Desktop — это графическое приложение для работы с системой контроля версий Git. Оно позволяет управлять репозиториями без использования командной строки, что упрощает работу с Git для начинающих пользователей.

GitHub Desktop предоставляет интуитивно понятный интерфейс для выполнения всех основных операций с Git, таких как создание репозиториев, клонирование, коммиты, работа с ветками и синхронизация с удаленным сервером. Это отличный инструмент для студентов и начинающих разработчиков, которые хотят освоить систему контроля версий без необходимости запоминать сложные команды.

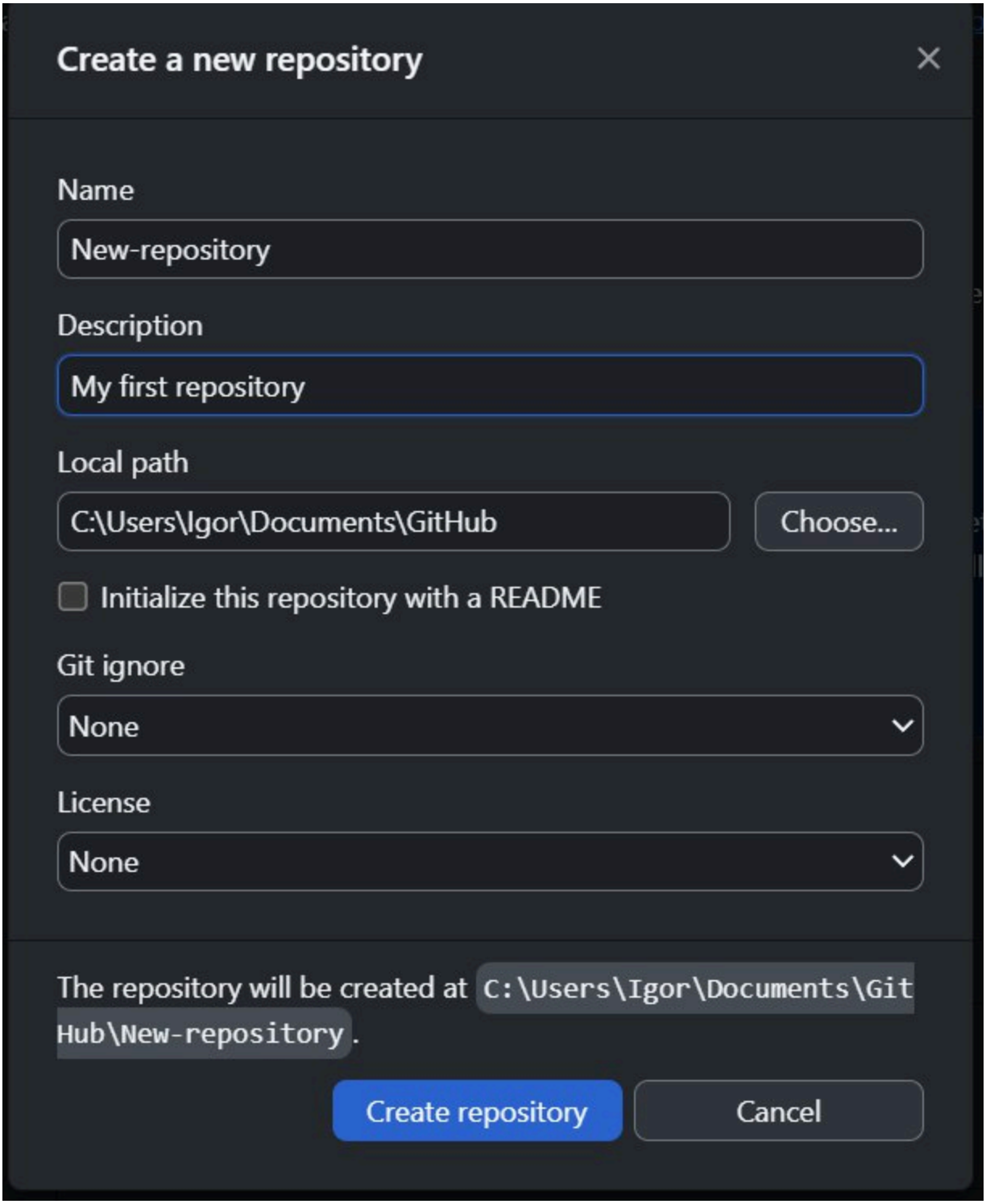
2. Основные возможности GitHub Desktop

2.1. Работа с репозиториями

Создание нового репозитория:

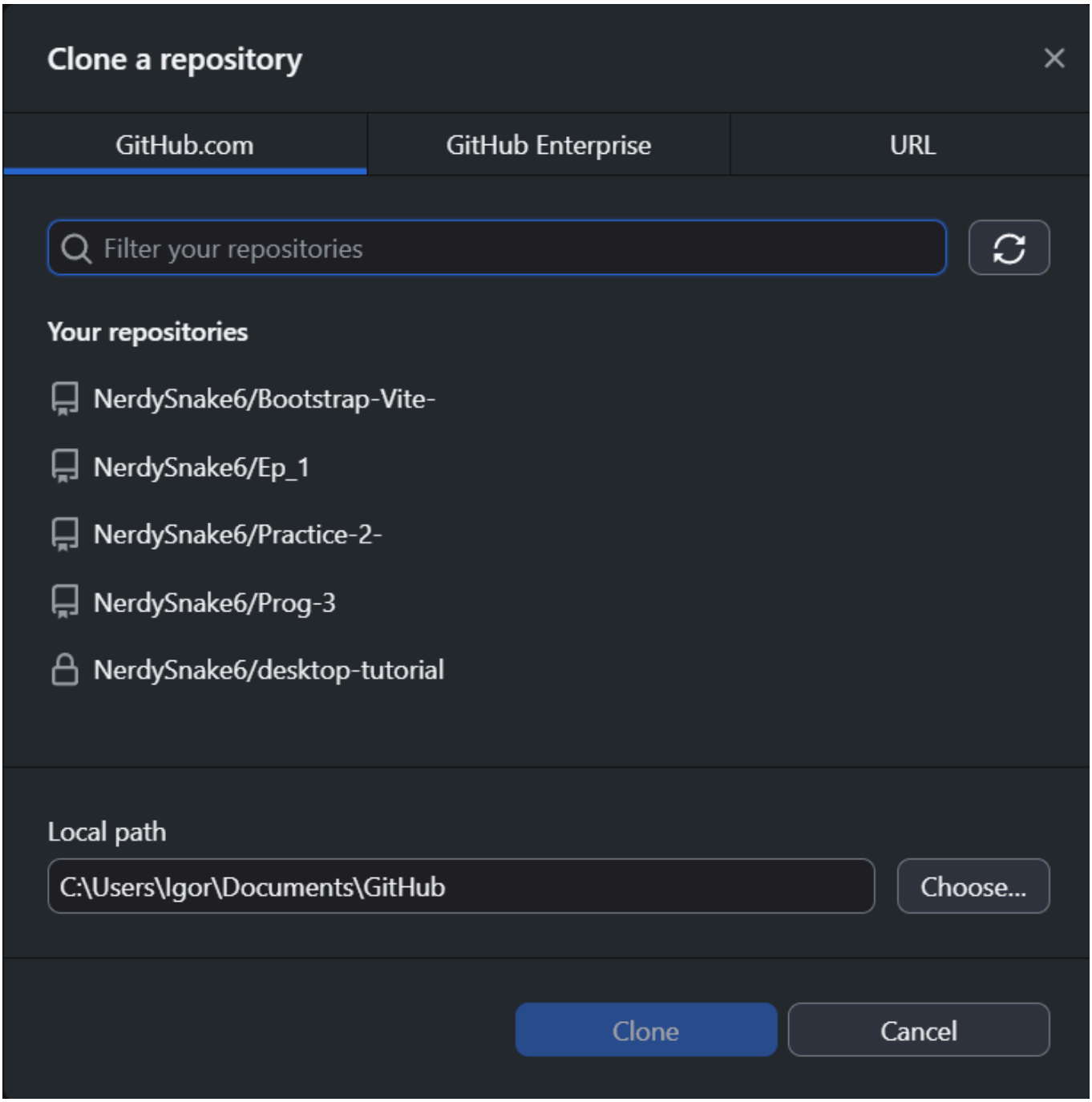
- **File → New Repository (Ctrl + N)**
- Задание имени и описания
- Выбор локального пути

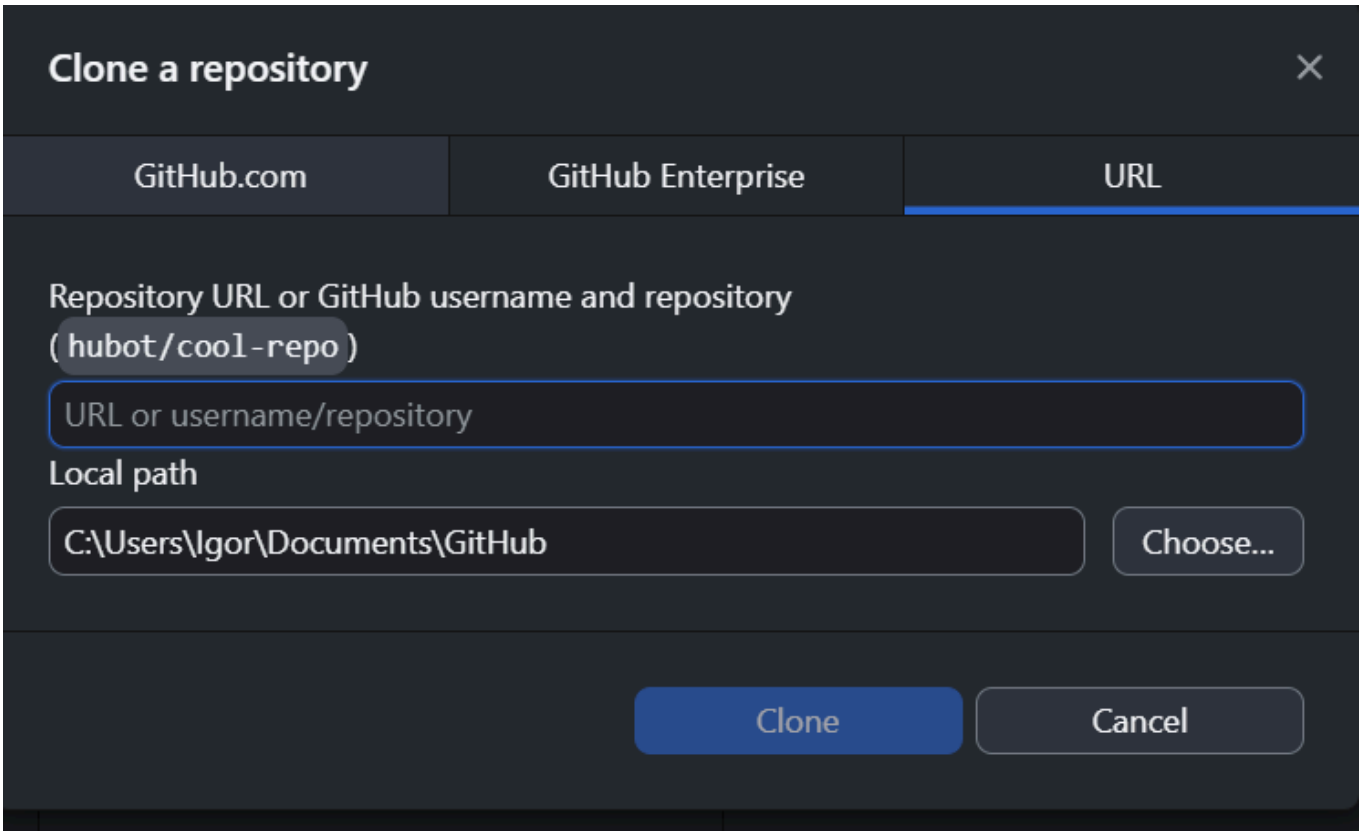
- Настройка README, .gitignore и лицензии



Клонирование существующего репозитория:

- **File → Clone Repository (Ctrl + Shift + O)**
- Выбор из списка своих репозиториев GitHub
- Клонирование по URL


Clone a repository
GitHub.com
GitHub Enterprise
URL
Filter your repositories
Your repositories
NerdySnake6/Bootstrap-Vite-
NerdySnake6/Ep_1
NerdySnake6/Practice-2-
NerdySnake6/Prog-3
NerdySnake6/desktop-tutorial
Local path
C:\Users\Igor\Documents\GitHub
Choose...
Clone
Cancel

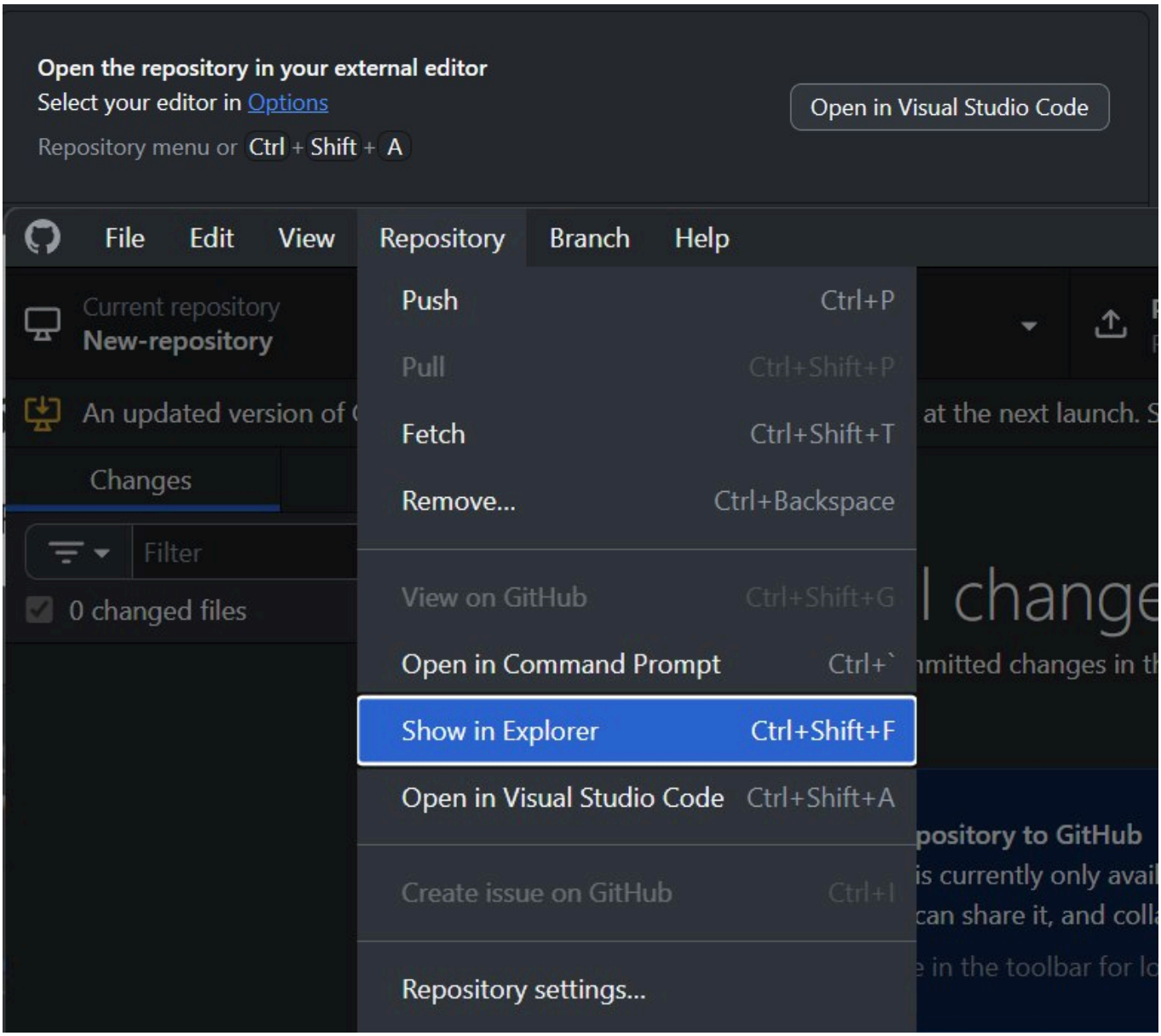


Клонирование позволяет работать с уже существующими проектами. Можно выбрать репозиторий из списка своих проектов на GitHub или ввести URL любого публичного репозитория. После выбора локального пути для сохранения, репозиторий будет скопирован на компьютер и готов к работе.

2.2. Работа с изменениями

В разделе **Changes** отображаются все измененные файлы. Можно видеть, какие строки кода были добавлены или удалены.

3. Работа с репозиториями



GitHub Desktop интегрируется с популярными редакторами кода, такими как Visual Studio Code. Это позволяет быстро открыть проект в предпочитаемом редакторе для внесения изменений. Также доступны быстрые команды для открытия репозитория в проводнике, командной строке или просмотра на GitHub.

4. Работа с изменениями

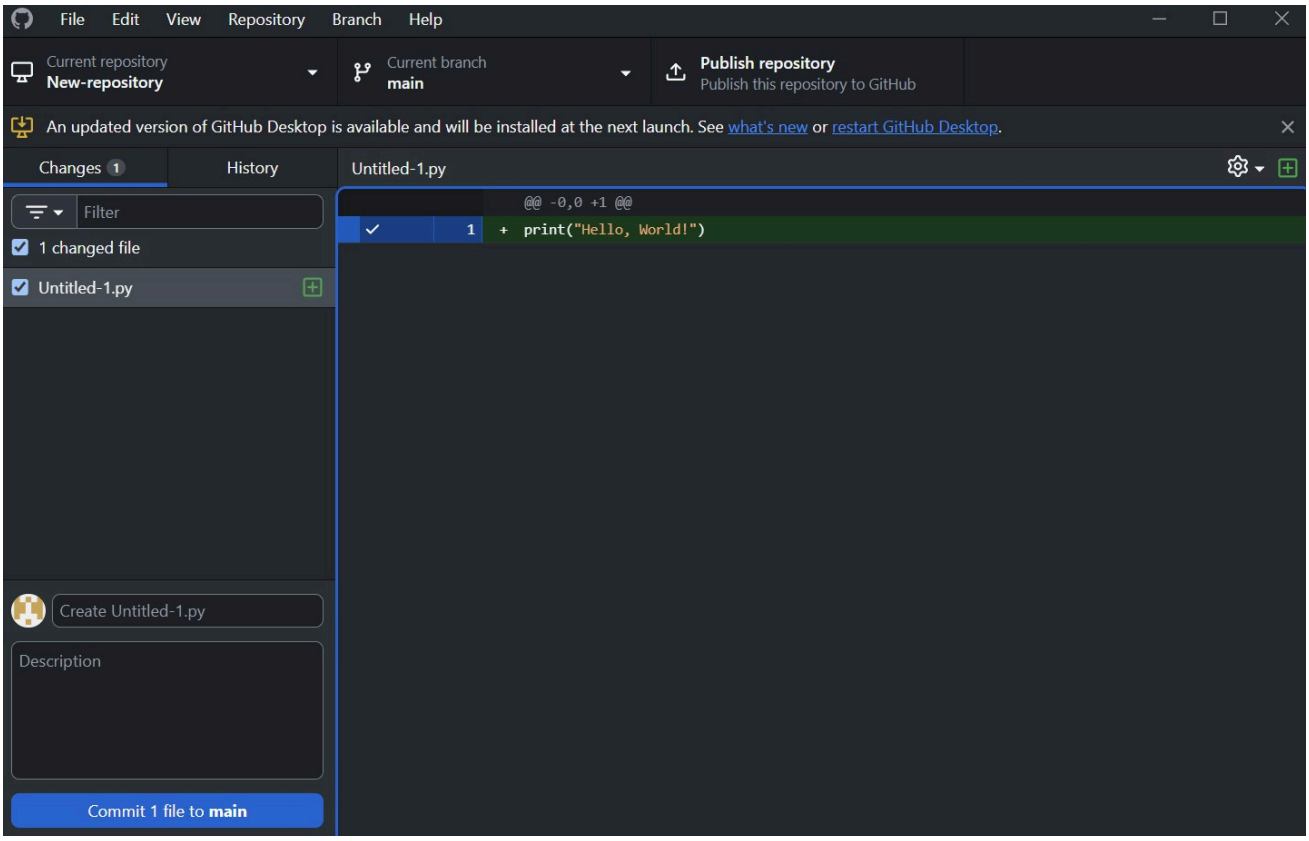
В разделе **Changes** отображаются все измененные файлы. Можно видеть, какие строки кода были добавлены или удалены.

4.1 Создание коммитов

Для сохранения изменений:

1. Выбрать файлы для коммита
2. Добавить заголовок и описание

3. Нажать "Commit to [ветка]"

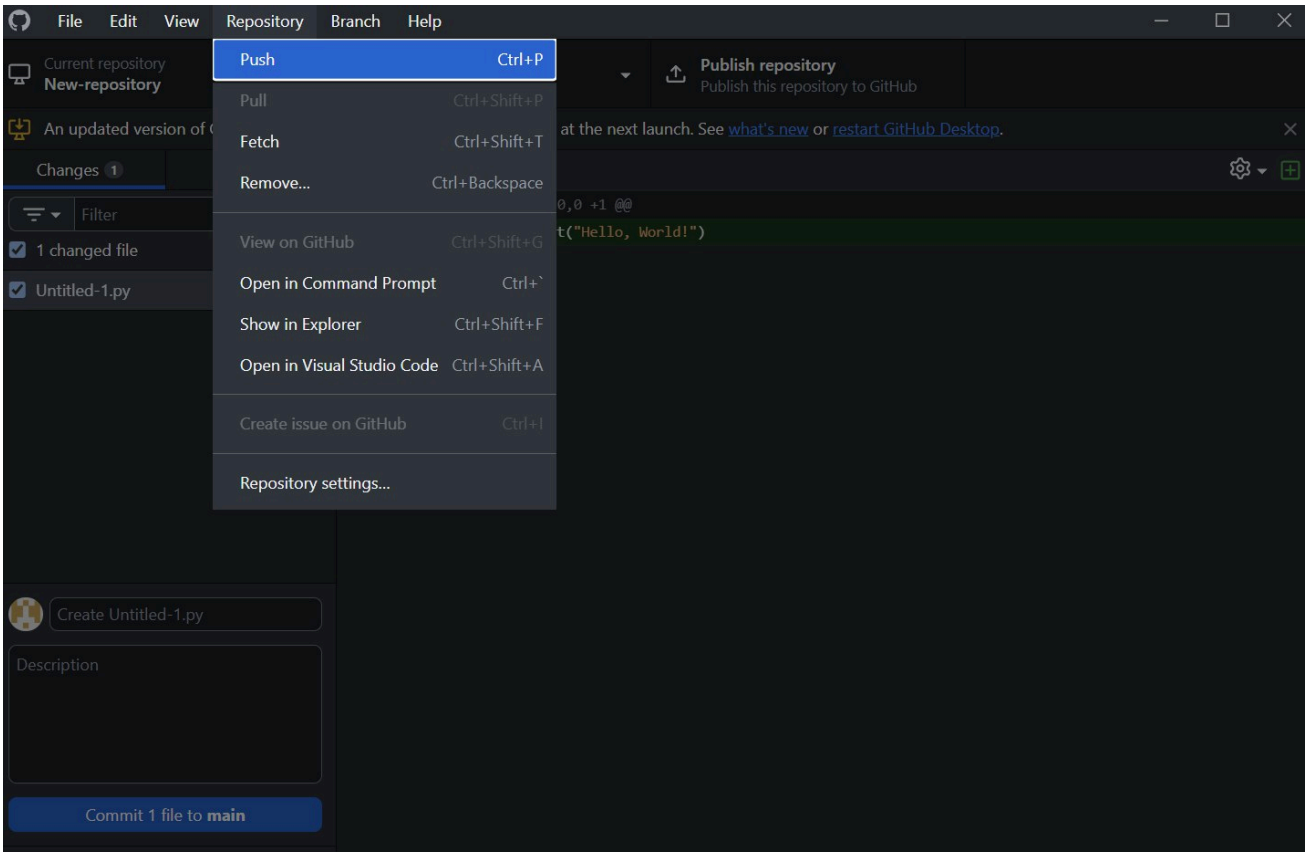


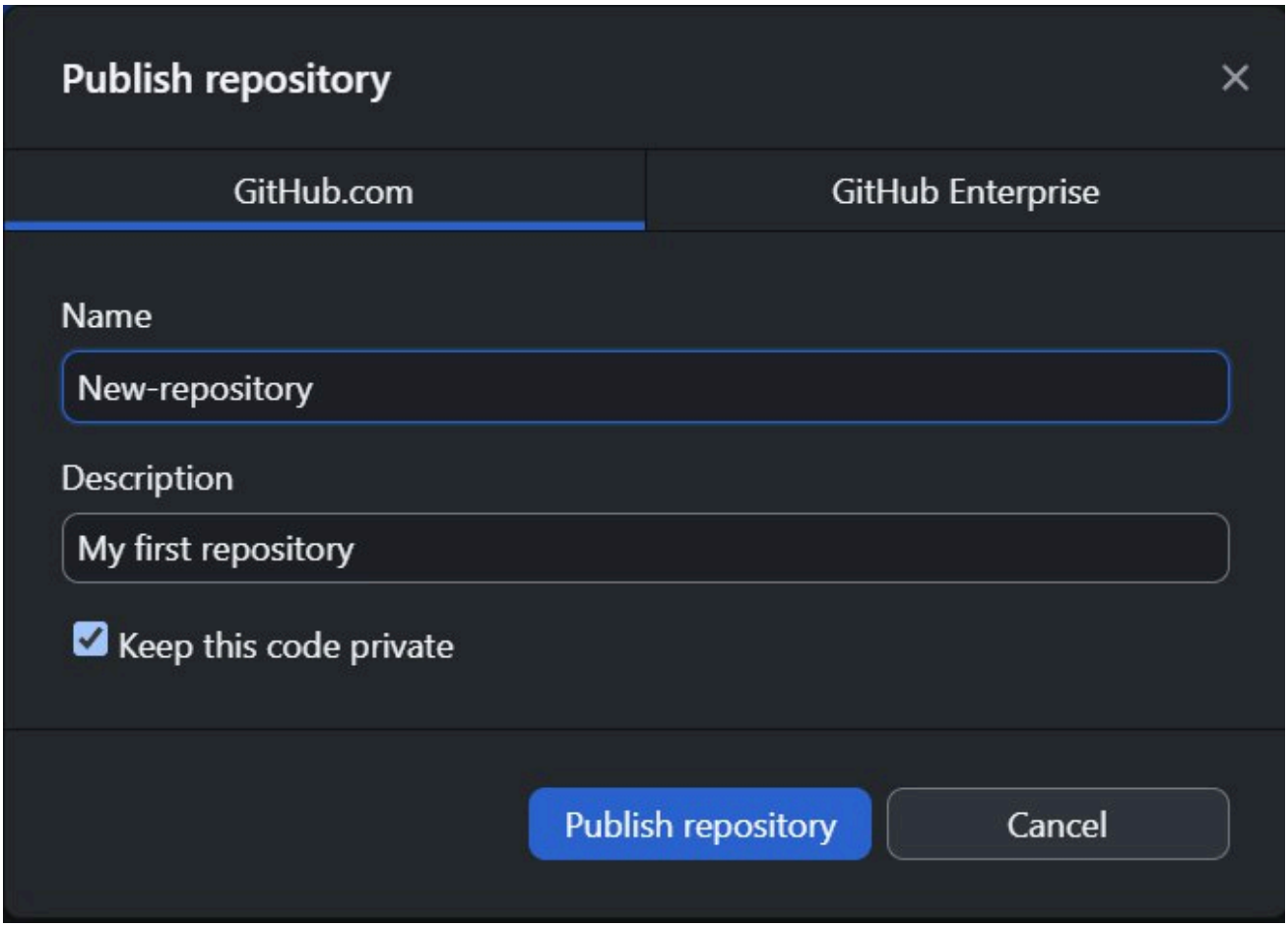
4.2 Синхронизация с GitHub

Push (отправка изменений):

- **Repository → Push (Ctrl + P)**

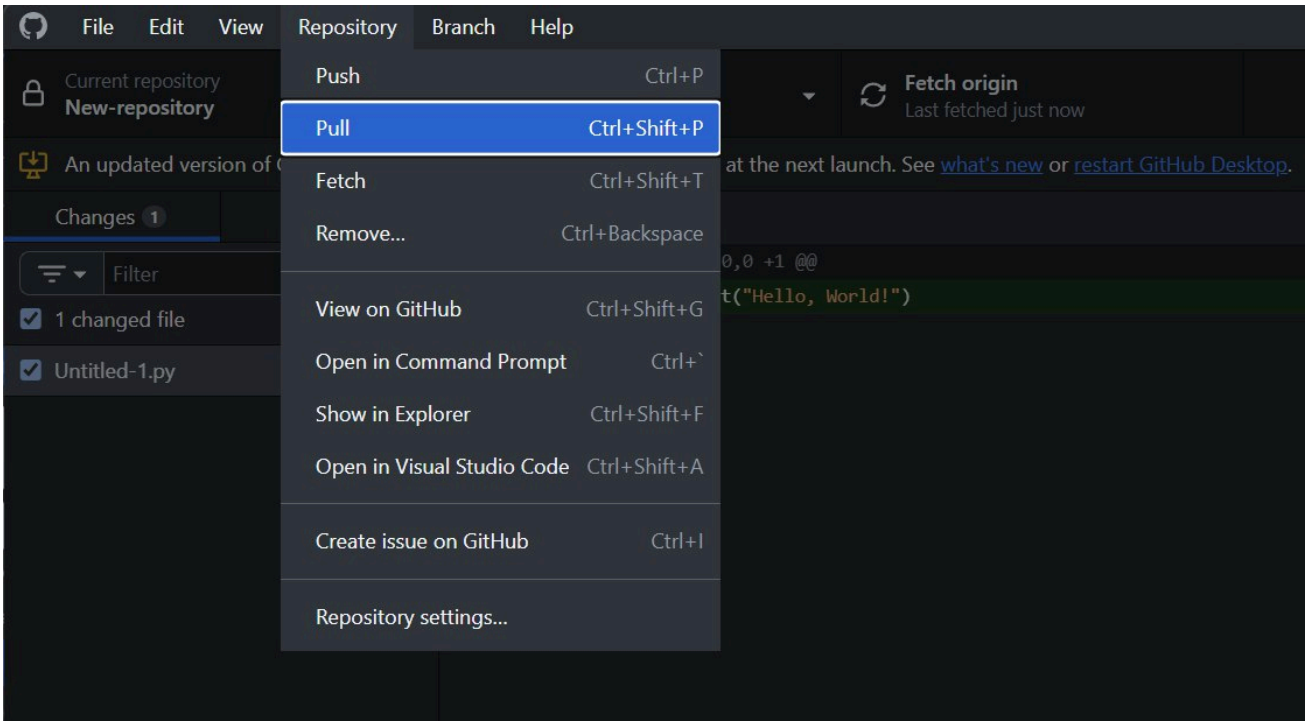
- Отправка локальных коммитов на GitHub



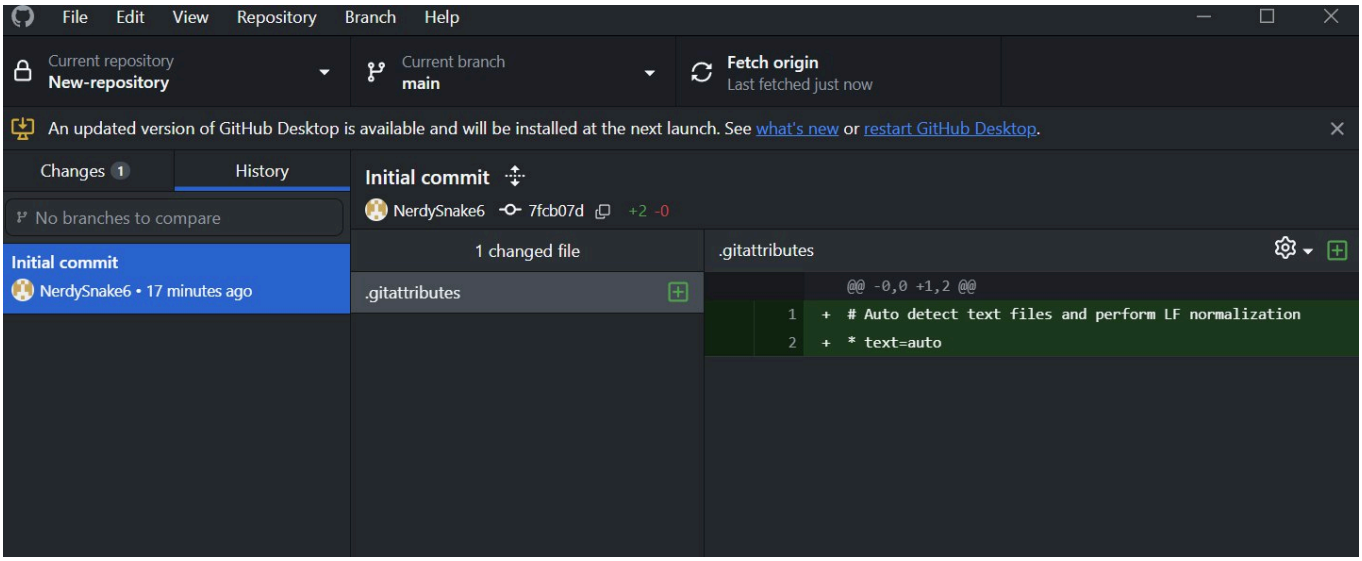


Pull (получение изменений):

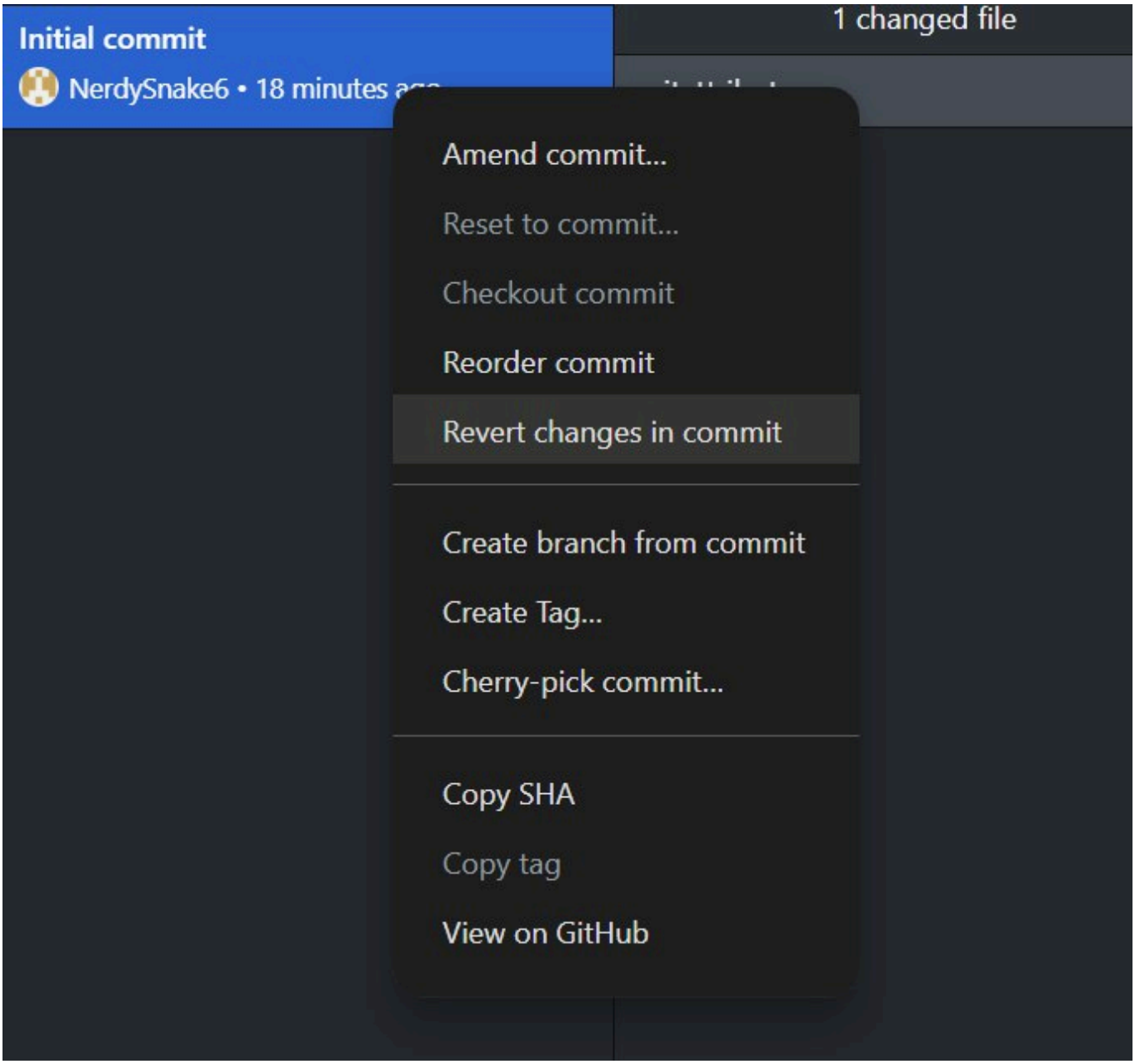
- **Repository → Pull (Ctrl + Shift + P)**
- Загрузка изменений с GitHub на локальный компьютер



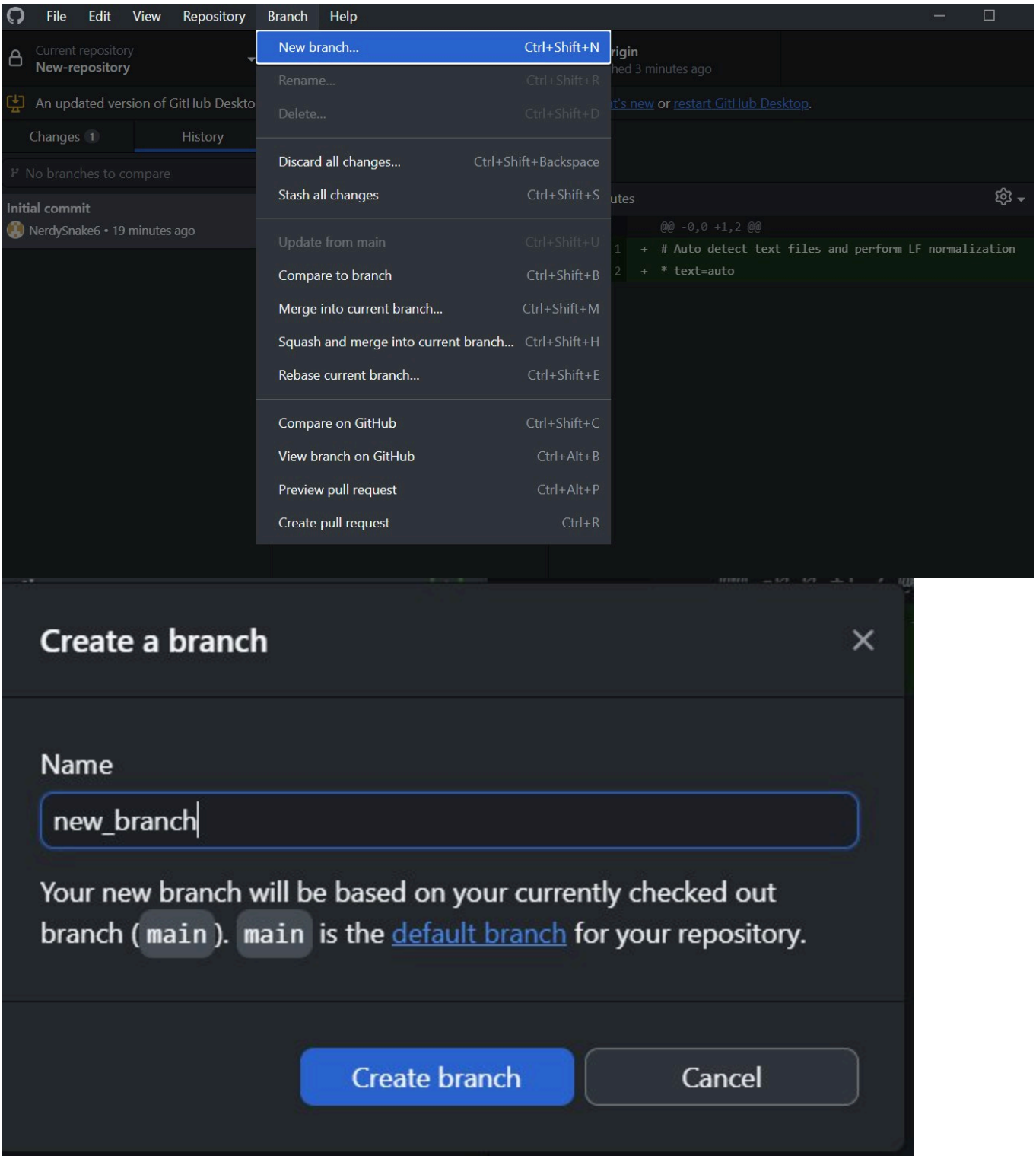
После создания коммитов на локальном компьютере, их нужно отправить на удаленный сервер GitHub с помощью команды Push. Это сохраняет изменения в облаке и делает их доступными для других участников проекта.



Через **History** можно просмотреть историю всех коммитов в репозитории и при необходимости **вернуться к любой из предыдущих версий** файлов. Это удобно для отката изменений или восстановления удалённого кода.



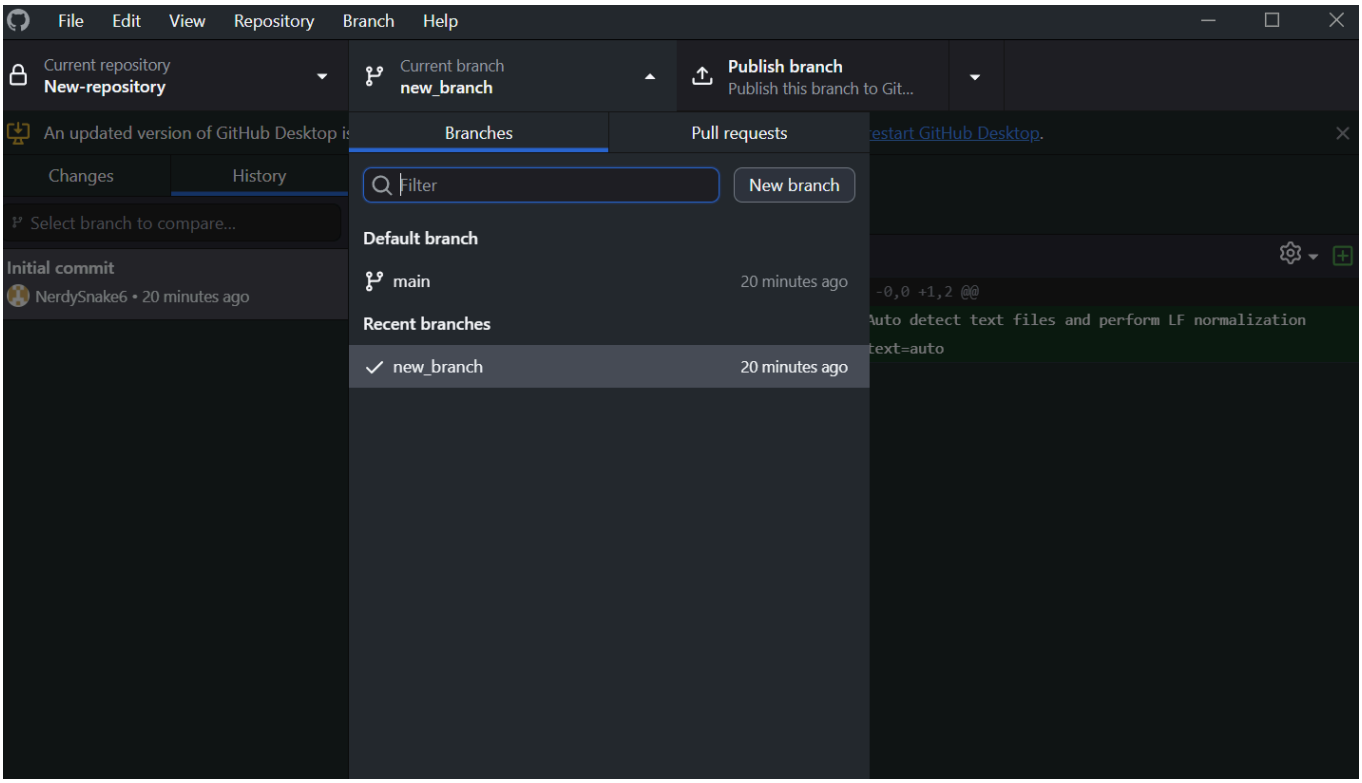
Кнопка **Revert changes in commit** позволяет отменить изменения, внесённые в выбранном коммите, создавая новый коммит с обратными правками. Это безопасный способ «откатить» конкретный коммит, не удаляя его из истории.



Создание новой ветки:

- **Ветка → Новая ветка (Ctrl + Shift + N)**
- Указание имени новой ветки
- Выбор базовой ветки

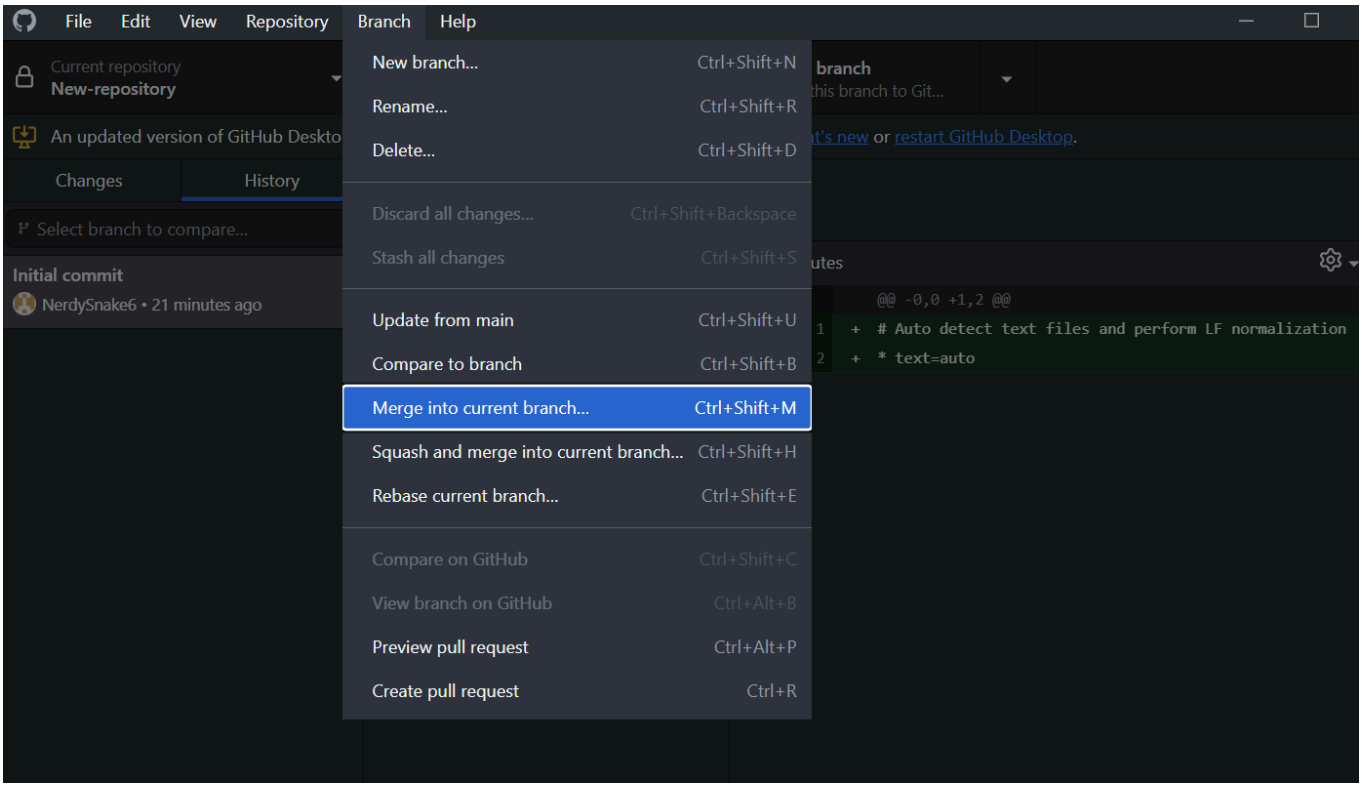
Новая ветка создается на основе текущей активной ветки. Это позволяет начать работу над новой функцией или исправлением, не влияя на основную версию проекта.



Объединение веток:

- **Ветка → Объединить с текущей веткой... (Ctrl + Shift + M)**
- Выбор ветки для слияния

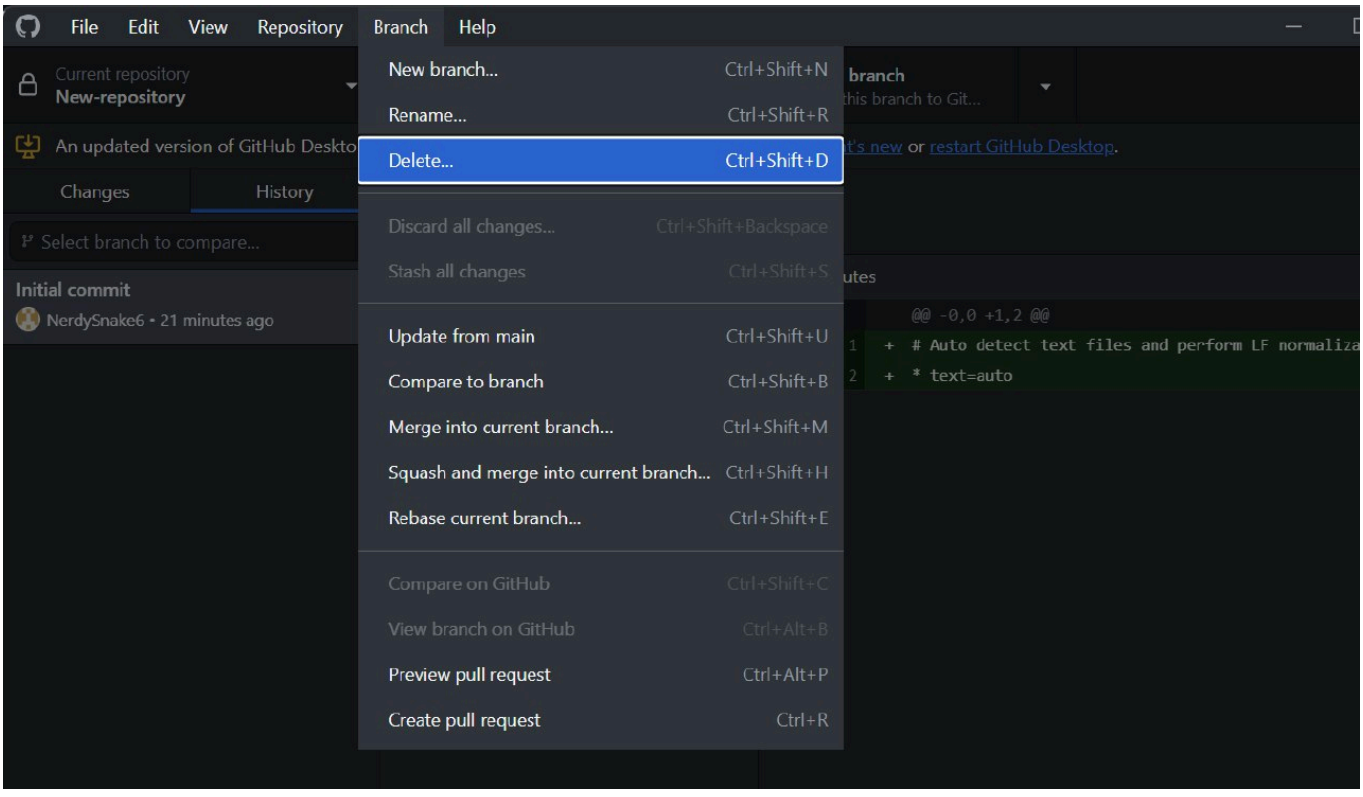
После завершения работы над веткой, ее изменения можно объединить с основной веткой. GitHub Desktop предоставляет инструменты для сравнения изменений и разрешения возможных конфликтов.

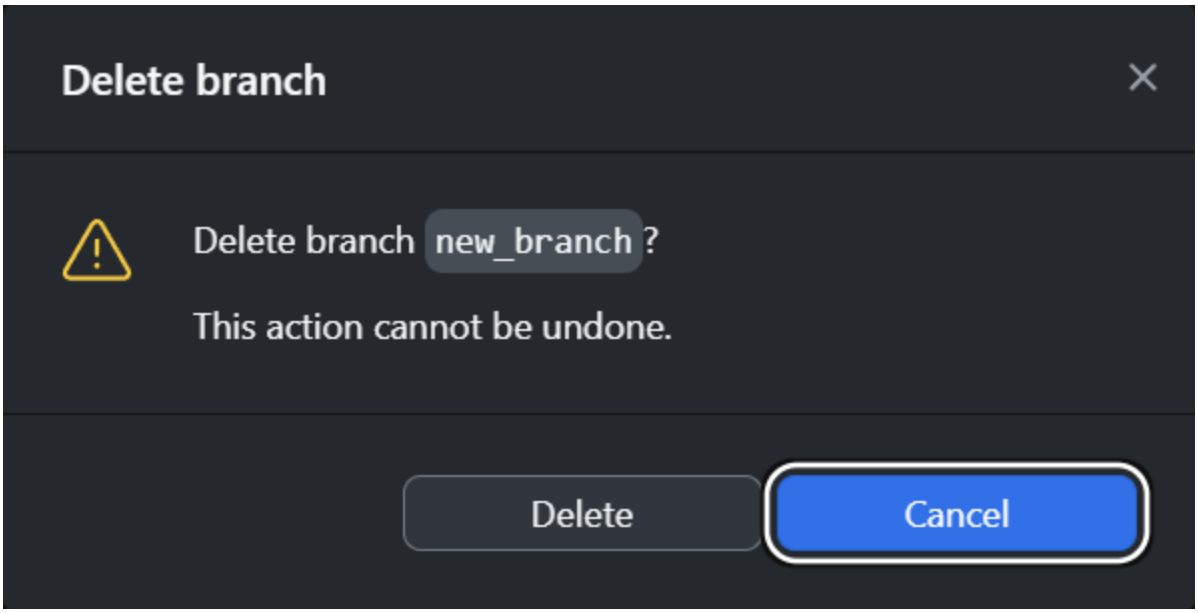


Удаление веток:

- **Ветка → Удалить... (Ctrl + Shift + D)**
- Подтверждение удаления

После успешного слияния ветки с основной, ненужные ветки можно удалить, чтобы поддерживать чистоту репозитория. GitHub Desktop запрашивает подтверждение перед удалением, чтобы избежать случайной потери данных.





6. Преимущества использования GitHub Desktop

1. **Простота использования** - интуитивный интерфейс, не требующий знания команд Git
2. **Визуальное представление** - наглядное отображение изменений и истории коммитов
3. **Интеграция с редакторами** - быстрый доступ к проекту в VS Code и других редакторах
4. **Безопасность** - подтверждение важных операций для предотвращения ошибок
5. **Бесплатность** - полностью бесплатный инструмент для личного и образовательного использования

7. Вывод

GitHub Desktop является отличным инструментом для начинающих разработчиков и студентов, которые хотят освоить систему контроля версий Git. Он предоставляет все необходимые функции для работы с репозиториями в понятном графическом интерфейсе.

Использование GitHub Desktop значительно упрощает процесс работы с Git, позволяя сосредоточиться на написании кода, а не на запоминании команд. Это делает его идеальным выбором для учебных проектов и первых шагов в разработке программного обеспечения.

Для более сложных рабочих процессов в профессиональной разработке может потребоваться переход к использованию командной строки Git, но GitHub Desktop служит отличной отправной точкой для изучения основных концепций системы контроля версий.