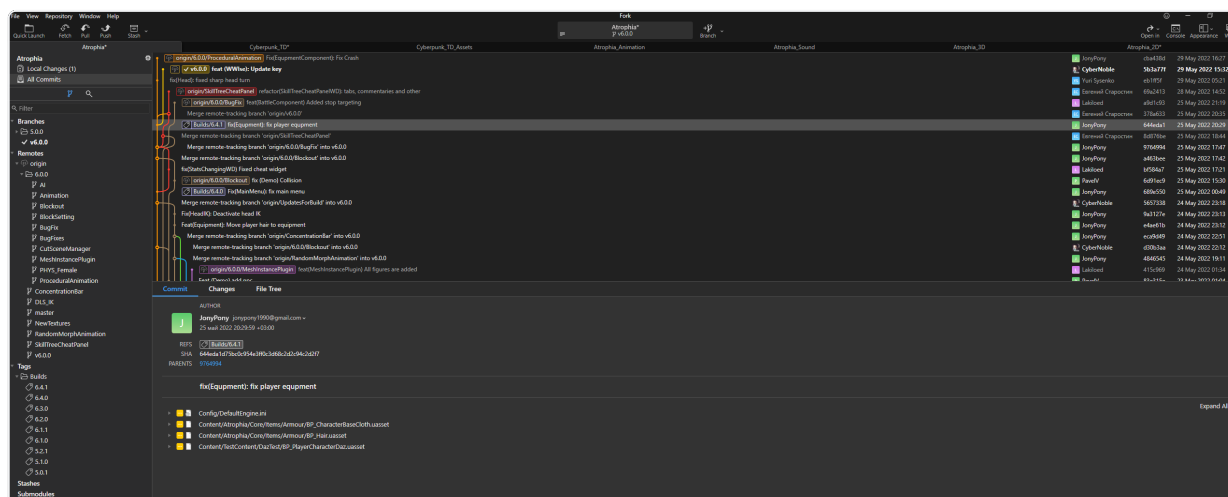


Гайд по работе с Fork

0. Введение

После того, как вы установили форк и скачали нужный вам репозиторий - вы будете работать в интерфейсе данной программы.



0.1 Локальные и серверные изменения

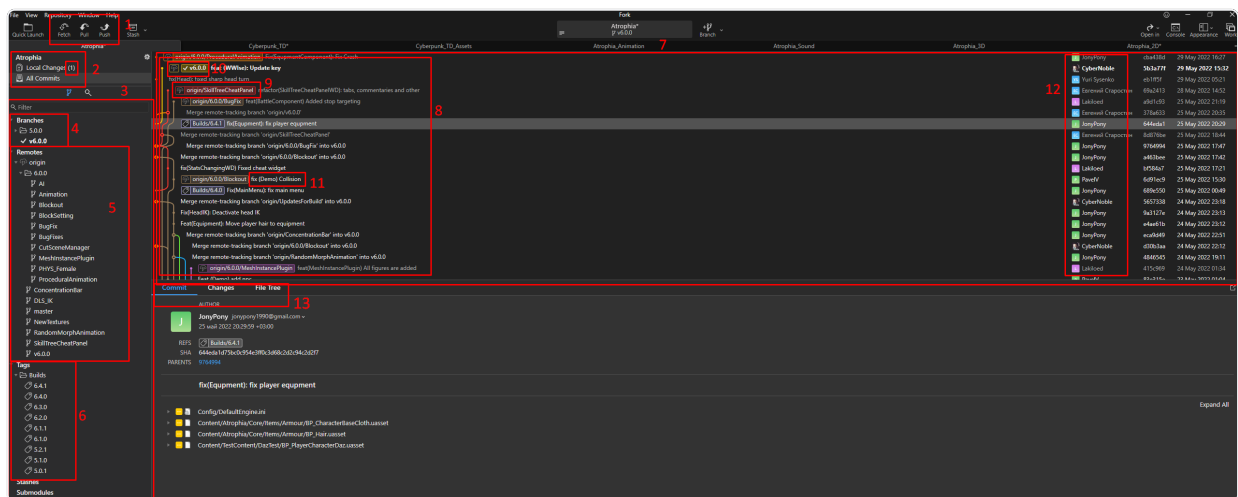
Гит работает по следующему принципу - вы скачиваете копию проекта с сервера и можете изменять ее локально у себя на ПК, чтобы ваши изменения попали на сервер, вам нужно их запустить.

 Чтобы узнать как - смотри  [Гайд по работе с Fork](#)

Любые ваши изменения, которые вы делаете с файлами - никак не затрагивают других людей, пока вы их не запустите и эти изменения не заберет себе другой человек

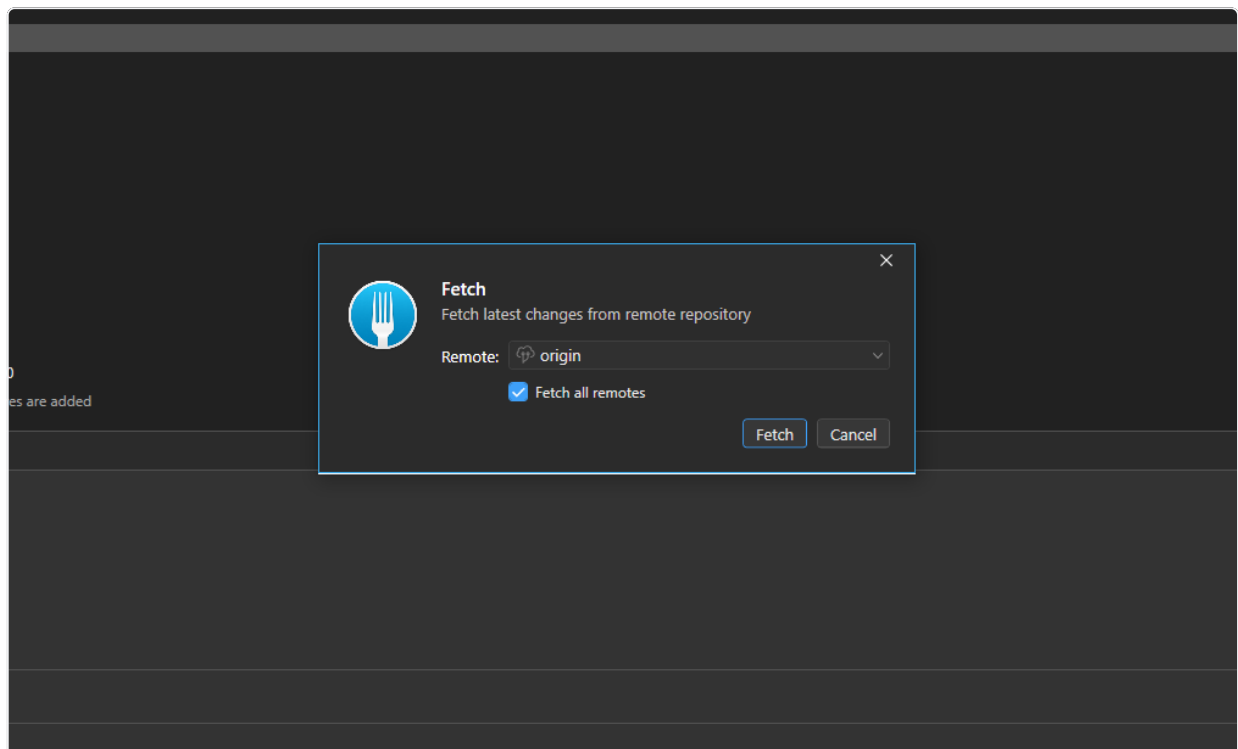
1. Интерфейс

Давайте разберемся что означают разные блоки



1. Fetch/Pull/Push

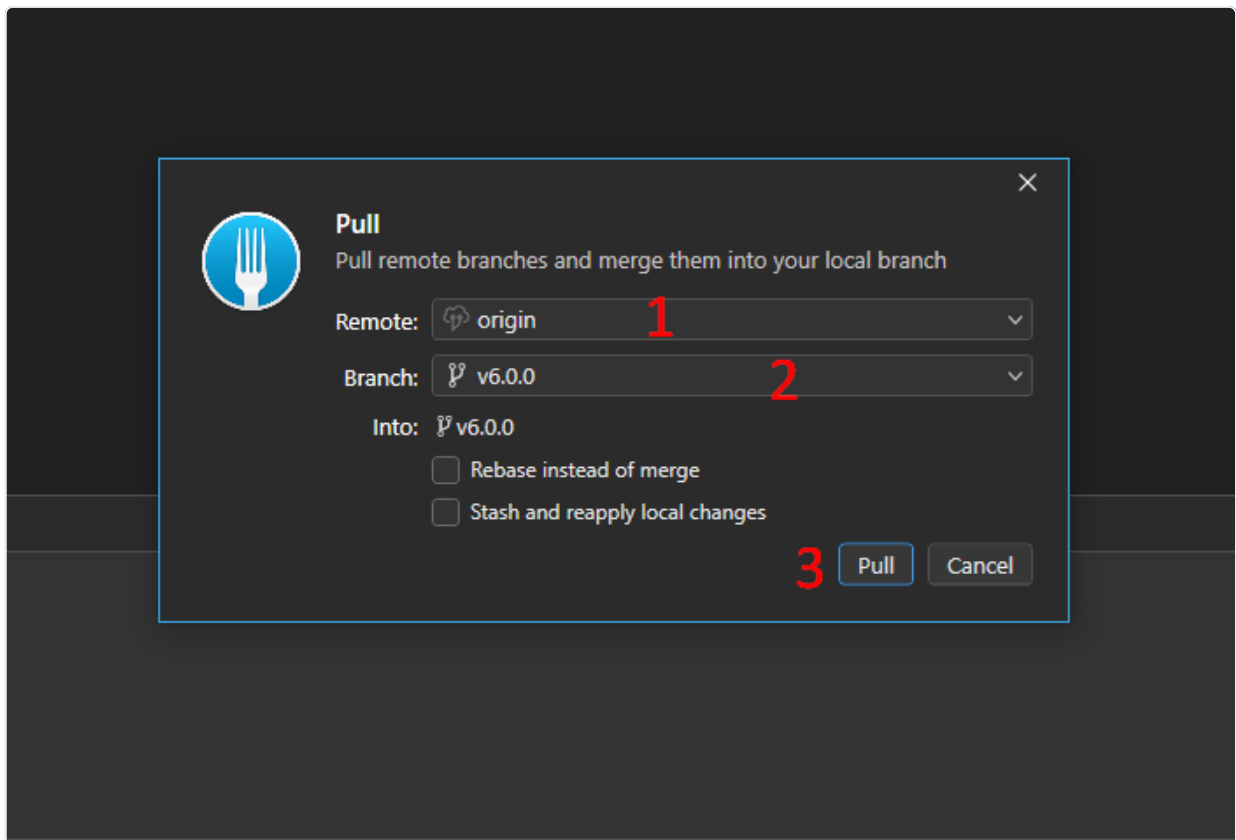
- Fetch - это обновление репозитория. Ее нужно всегда тыкать, если вы хотите что-то скачать или загрузить, чтобы актуализировать дерево из пункта 7.
 - Эту кнопку всегда нажимаем перед тем, как пушить или пулить
 - Тыкать на кнопку каждые 5 секунд после любого клика по форку - не нужно, достаточно 1 раз нажать и забыть об этом, в следующий раз нажмете, если пройдет, хотя бы, день или, если вас попросили.
 - Этой кнопкой ничего не возможно сломать, поэтому не бойтесь
 - Сам интерфейс кнопки. Не забудьте тыкнуть галочку Fetch all remotes



- Pull - это скачивание определенных изменений. Нужна, когда вы хотите в свою ветку "подтянуть" чужие изменения файлов.

❗ Например. Я сделал арт природы, но мне нужен медведь, который делал Миша. Он свою работу уже залил, но у меня ее не появилось, потому что я на другой ветке. Я могу скачать его медведя в свою ветку и добавить на арт

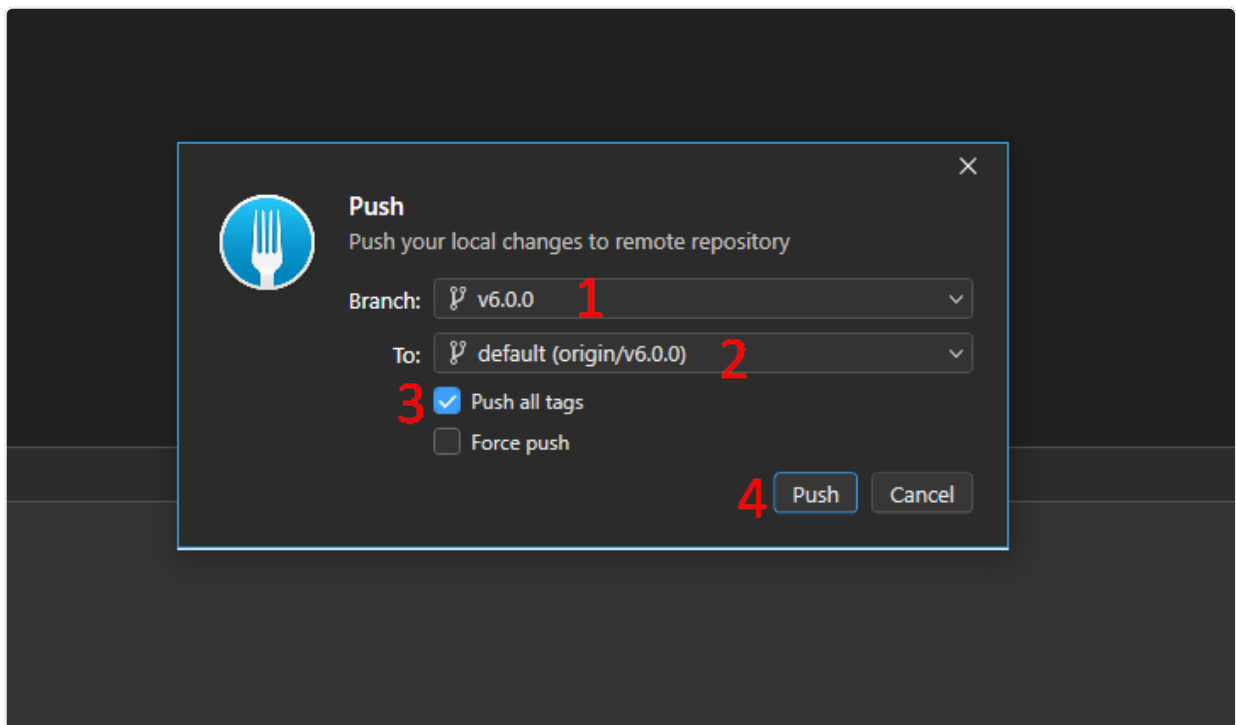
- Интерфейс кнопки pull



- 1 - Тут ничего не трогаем. Это то, из какого репа мы будем скачивать файлы
- 2 - Из какой ветки мы будем брать файлы (ветка Миши). Можно тыкнуть и вам покажется выпадающий список со всеми ветками на сервере
- 3 - Сама кнопка пул, жмите и получите изменения.

❗ Возможно конфликты, подробнее в разделе [🎯 Гайд по работе с Fork](#)

- Кнопка Push - Отправка своих изменений на сервер.
 - Интерфейс пуша



- 1 - Ветка, из которой вы хотите отправить изменения. Обычно, это ваша ветка
- 2 - Ветка, в которую вы хотите отправить изменения. Обычно, тоже ваша.
Разница между первым и вторым - это в том, что первая строка отвечает за локальную ветку, а вторая - за серверную.
 - Также, в этом списке можно выбрать самый нижний вариант - Custom Branch. После его выбора - вам откроется окно, в котором вы можете создать новую ветку прямо при пуше. Обычно используется, когда вы не создали ветку для работы и вам лень переходить на новую ветку. Новую ветку можно отправить вместе с изменениями.
При выборе этого пункта - создается ветка на сервере и ваши изменения попадают уже туда

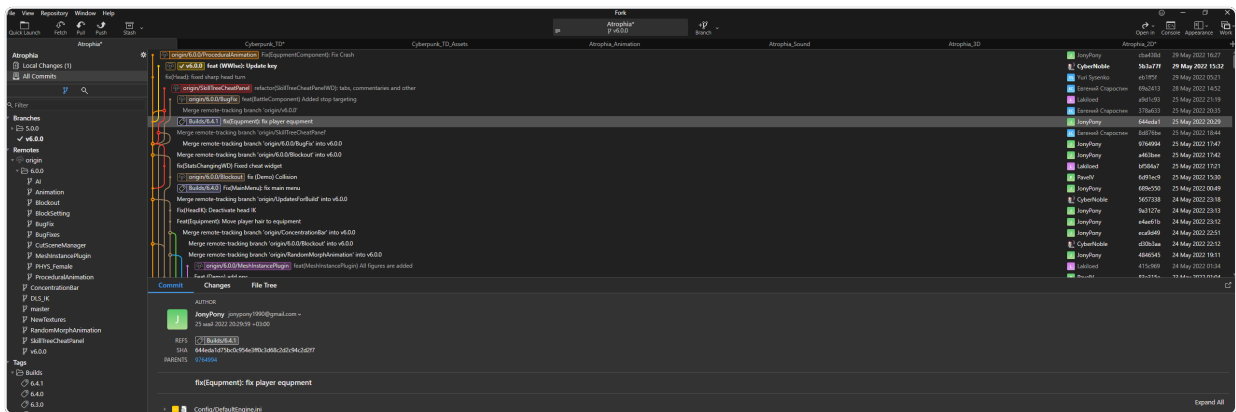
i Подробнее об локальных и серверных изменениях - смотри в разделе

- 3 - Ставим галочку. Это позволит пушить теги. Тег - это просто метка на коммите. С помощью тегов можно делать подписи. Обычно используется, чтобы быстро найти нужный коммит, если он затеган
- 4 - Сама кнопка пуш.

i Чтобы узнать как пушить файлы - смотри раздел

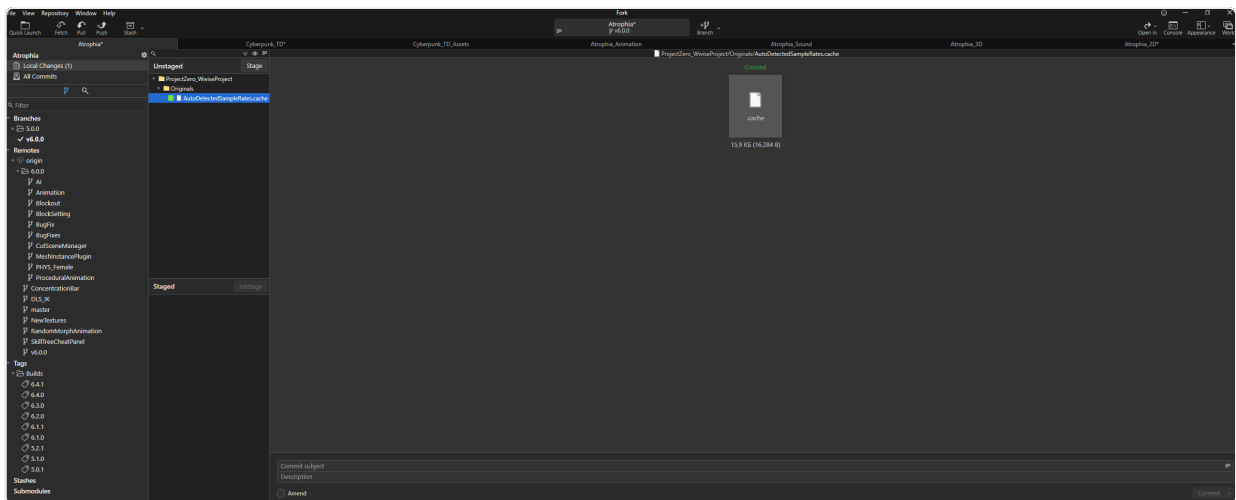
2. All commits / Local Changes

Данный блок позволяет переключаться между деревом



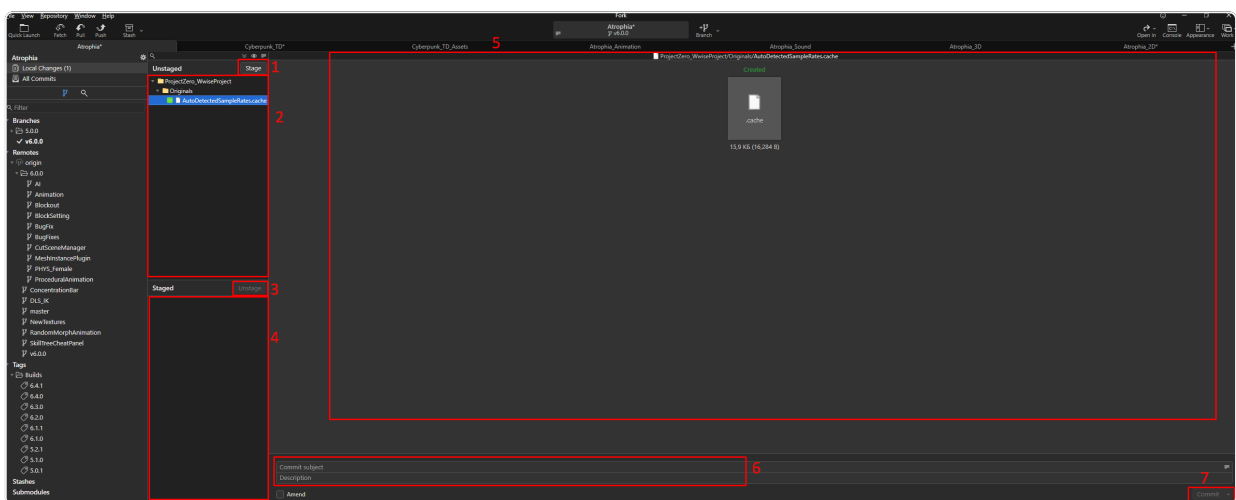
All Comits

И вашими изменениями



Local Changes

2.1 Интерфейс Local Changes



1. Кнопка, которая переводит файлы из области 2 в область 4

- а. Чтобы перекинуть файлы - их нужно выделить. Для этого достаточно ткнуть по любому файлу из списка и нажать ctrl + a
2. Список файлов, которые были изменены/удалены/добавлены в репозиторий (локально)
3. Кнопка, которая переводит файлы из области 4 в область 2
 - а. Чтобы перекинуть файлы - их нужно выделить. Для этого достаточно ткнуть по любому файлу из списка и нажать ctrl + a
4. Список файлов, которые были выбраны для созданий коммита
 - а. Сюда можно не все файлы из области 2 кидать, а лишь некоторые, возможно, что вы для себя - что-то удалили или заменили, чтобы вам было удобней работать, но это не должно попасть всем. В таком случае - вы, на этапе выбора файлов для коммита - удаляете из 4 области все файлы из списка, кроме тех, которые вам нужны
 - і. Файлы удаляться только из области 4. Локально они никуда не денутся, но на сервер отправлены не будут. На сервер отправляются только стейджнутые файлы (файлы из списка 4)
5. Подробное описание изменений в файлах. При выборе любого файла в списке 2 или 4 - тут будет отображены его свойства, если формат файла это позволяет
6. Поле для названия коммита. То, как будет называть коммит.
 - а. Опционально - можно заполнить еще и нижнюю строку. Это описание. Сюда, обычно пишут чуть более детально что было сделано
7. Кнопка коммита - после нажатия на нее - собирается коммит, который потом можно запустить на сервер

3. Ветки

В данном блоке расположены локальные и серверные ветки.

4. Локальные ветки

Это такие ветки, которые вы скачали или создали у себя на компе. Галочкой помечена текущая ветка

5. Серверные ветки

Это ветки, которые находятся на сервере и их можно скачать себе на комп, подтянуть из них изменения или запустить в них изменения

6. Теги

Пометки для себя, никак не влияют на файлы, нужны для лучше ориентации в коммитах

7. Рабочая область

Это блок, в котором вы можете увидеть как мержились ветки (объединялись их изменения), какие ветки где находятся и что было сделано

8. Коммиты

Изменения, которые делались людьми на сервере. Одна строка - один коммит

9. Серверная ветка

Ветка, которая находится на сервере. Ее можно определить по облачку рядом с названием и начало названия с origin/название ветки

10. Локальная ветка

Ветка, которая скачена или создана на вашем ПК. В примере выше - там есть еще и облачко, это означает, что ваша ветка никак не отличается от серверной и находится на одном месте

11. Название коммита

Тут указывается кратко то, что было сделано

12. Логин пользователя

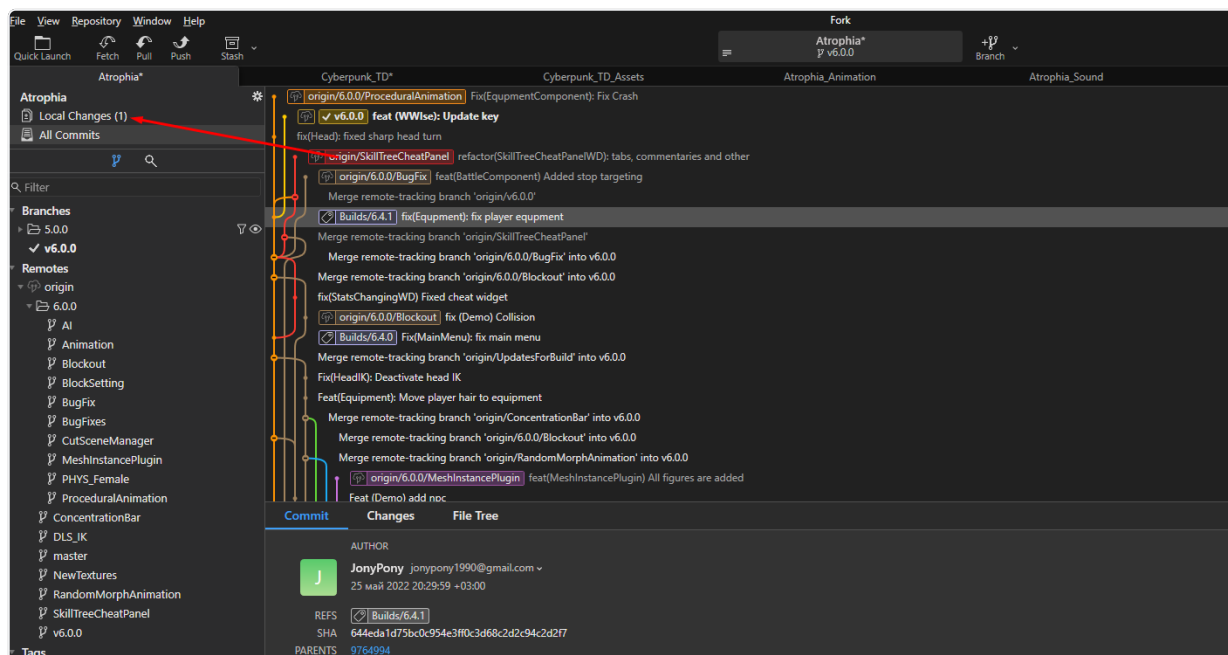
Тут отображается тот, кто сделал данный коммит

13. Файлы

В данных 3 вкладках можно посмотреть разные варианты отображения того, что было изменено в коммите

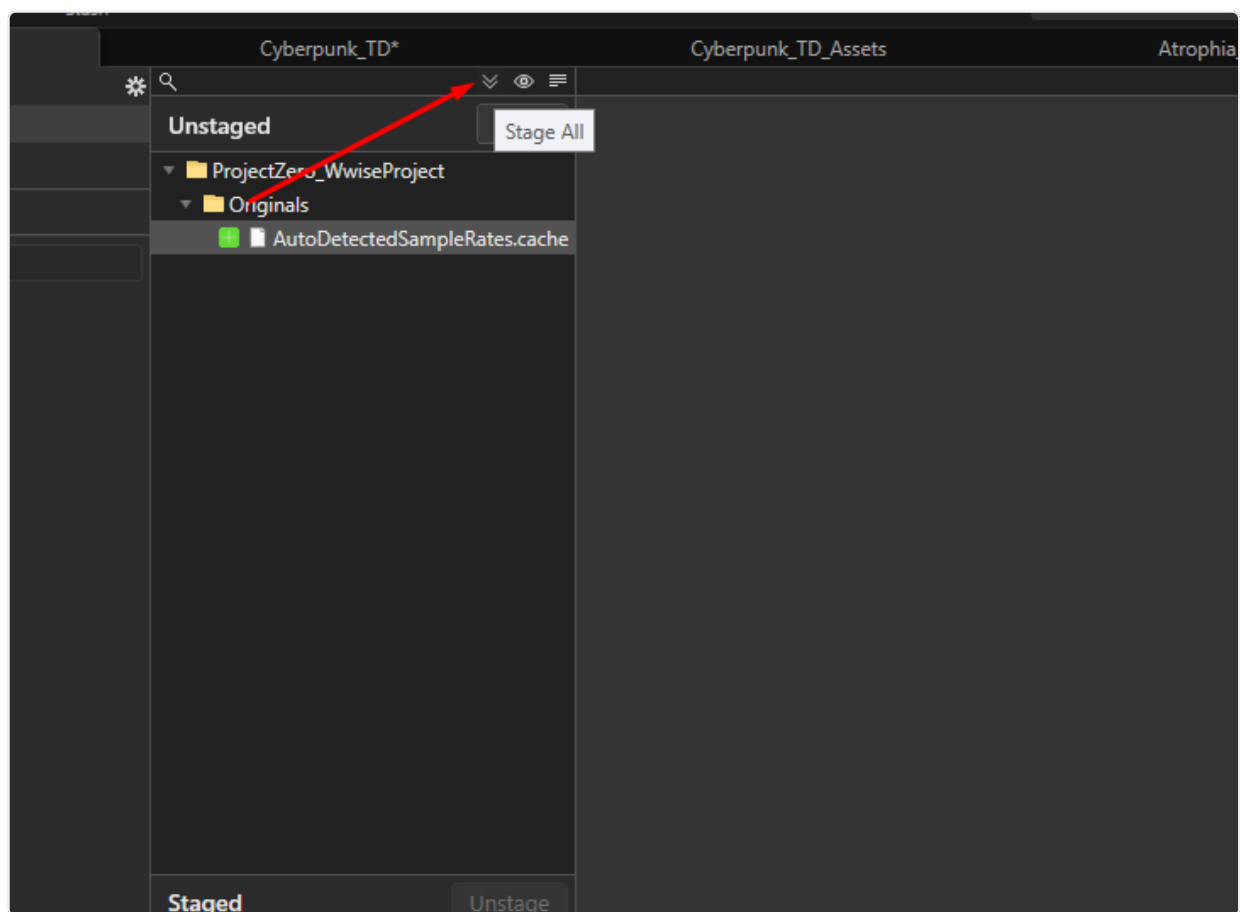
2. Как пользоваться форком

После того, как вы поработали в движке или с папками репозитория у себе на компе - у вас появится N-ное количество измененных файлов, их число можно увидеть тут

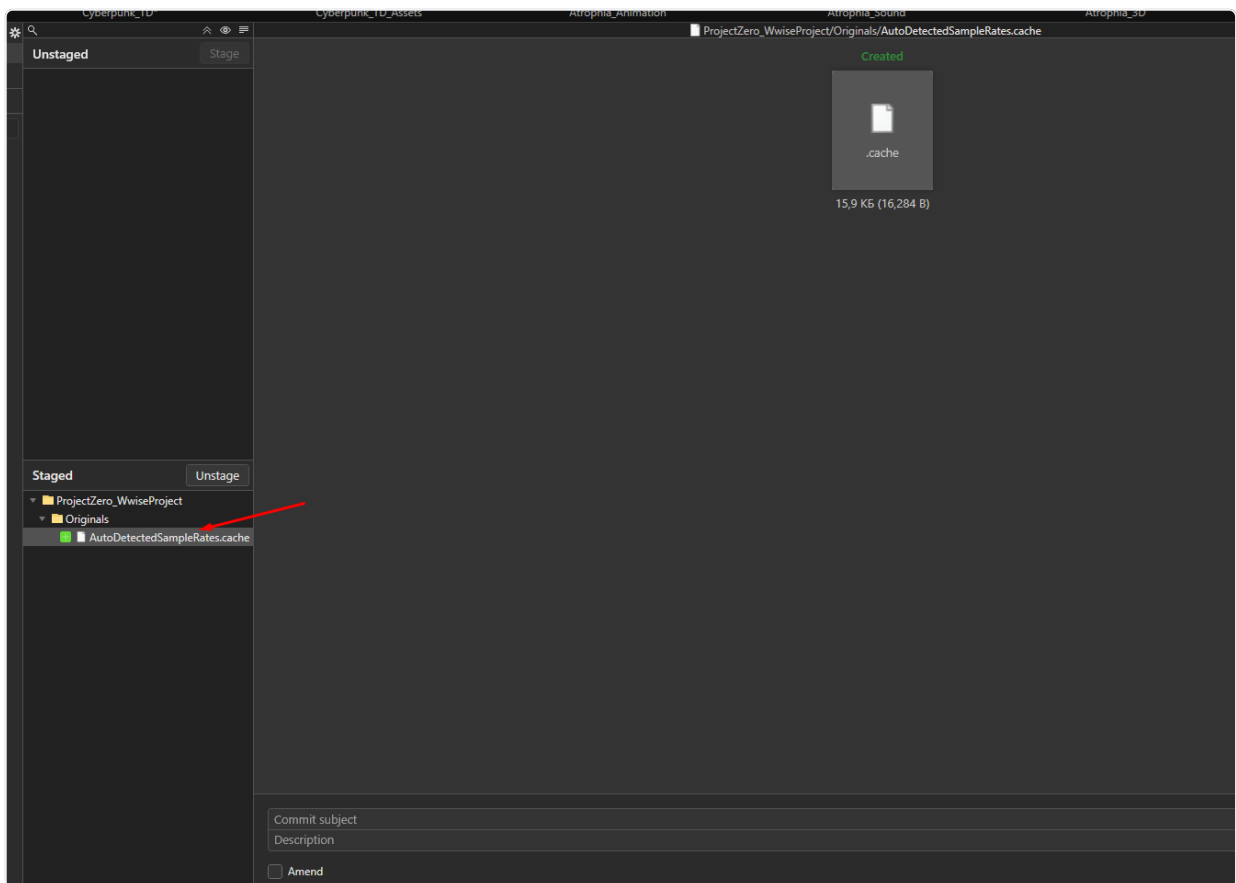


Чтобы запустить эти файлы в реп, вам нужно сделать следующее:

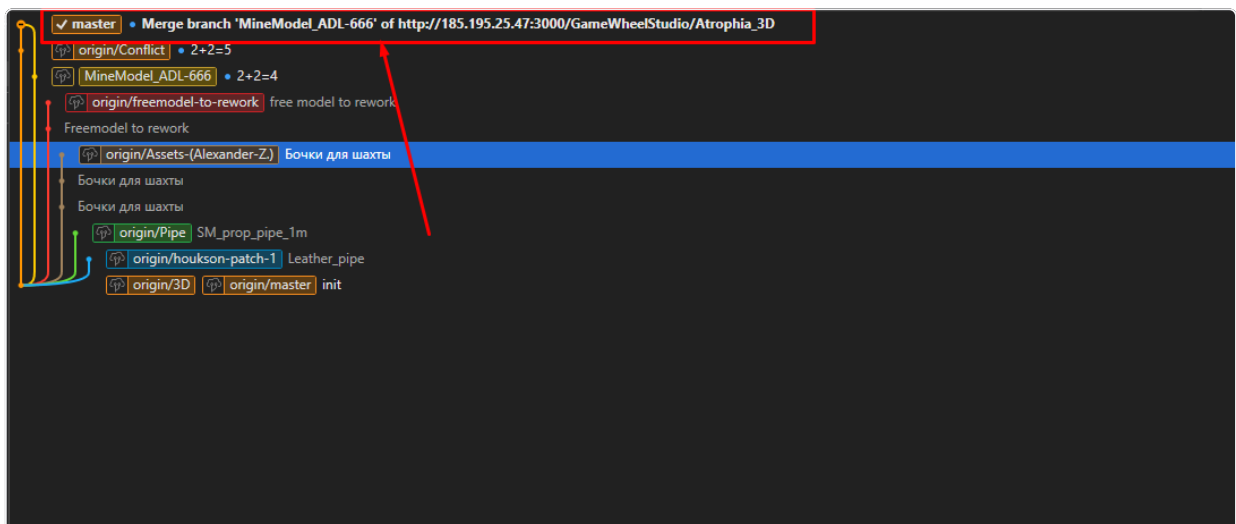
- Тыкнуть на вкладку Local Changes
- Выбрать файлы, которые вы изменили И ВАМ НУЖНО, чтобы эти файлы попали в общий реп
 - Если вы ради прикола добавили фотку обезьяны в папку с репом, то не нужно ее загружать туда, просто потому, что она показывается как изменение
- Чтобы выбрать файлы - выделяем их через CTRL или тыкаем CTRL + A (Выделятся все файлы), либо можно нажать на кнопку STAGE ALL (если вы хотите все изменения запустить и там нет ненужных файлов)



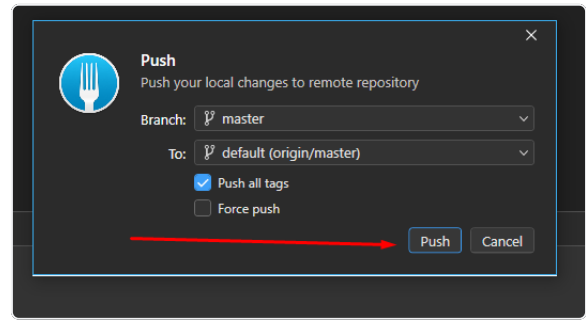
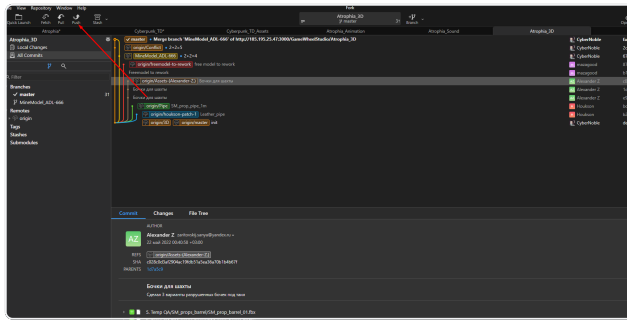
- После того, как вы застейджили файлы - они появятся в блоке ниже



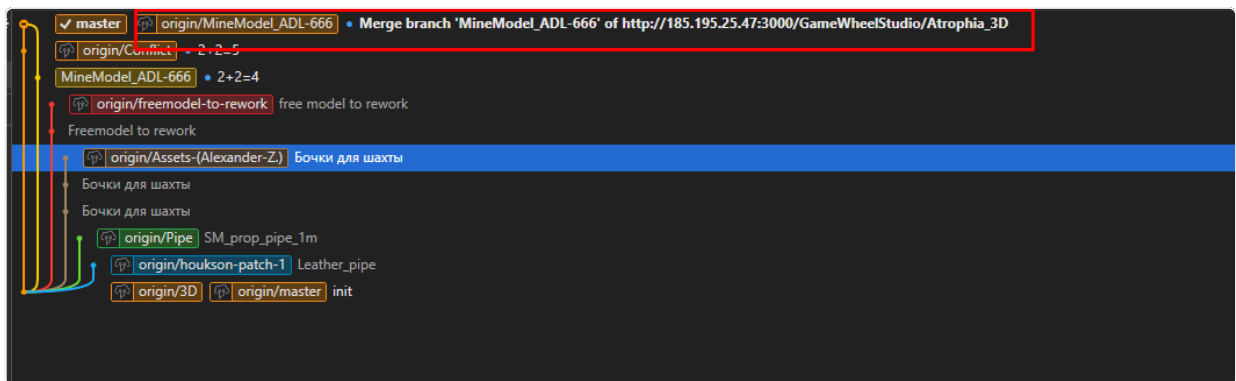
- Теперь даем название вашему коммиту, согласно спецификации [Гайд по работе с Fork](#)
- Затем переходим во вкладку All Commits



- Как вы видите, ваш коммит появился в дереве, но рядом с ним нет облачка. Чтобы оно появилось, вам осталось только кликнуть на кнопку push



- Если после этого рядом с вашей веткой появилось облачко, то вы все сделали правильно



3. Нейминг коммитов и веток

3.1 Нейминг коммитов

Название коммита составляется из нескольких частей, которые разделены нижним п:

1. Тип
2. Область
3. Конкретная цель работы
4. Номер задачи с ютрека

Для Atrophia (Dev команды)

Рассмотрим пример, где я сделал скульпт модели ГГ

Пример. Char_MainHero_Sculpt_ADL-666

Теперь разберем по пунктам:

1. Тип работы, которую я делал, в данном случае - это персонаж, поэтому пишу Char
2. Затем пишу область, уточнение, что это за персонаж, поэтому указал MainHero
3. Затем конкретно то, что я делал, я скульптил, поэтому пишу sculpt

4. Моя задача находится на ютреке под номером ADL-666, поэтому я указываю именно это в названии коммита. Не нужно писать всю ссылку, только номер задачи

✗ Важное уточнение. В форке забудьте о русском языке и пробелах, только английский, некоторые спец. символы и куча слов вместе, которые разделены заглавными буквами, как в примере выше с MainHero

3.2 Нейминг веток

При создании названия ветки - указывается то, что вы собираетесь делать и номер задачи с ютрека.

i Пример. Я делаю задачу на ретоп пушистика
<https://gamewheel.myjetbrains.com/youtrack/issue/ADL-1147>
Исходя из этого, я называю ветку так, чтобы было понятно, что это ретоп пушистика, например - PushistikRetop
Далее - я добавляю номер задачи на ютреке, не всю ссылку, а только номер.
Получается вот так: PushistikRetop_ADL-1147

4. Правила работы с ветками

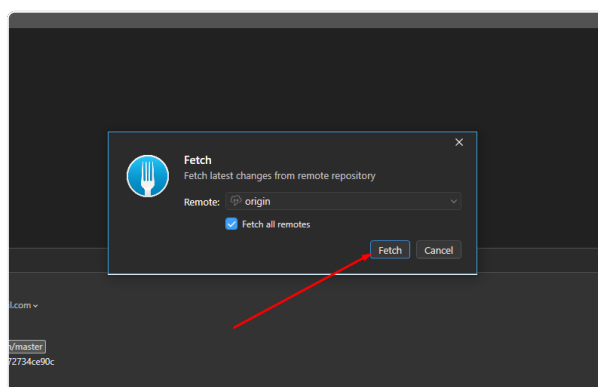
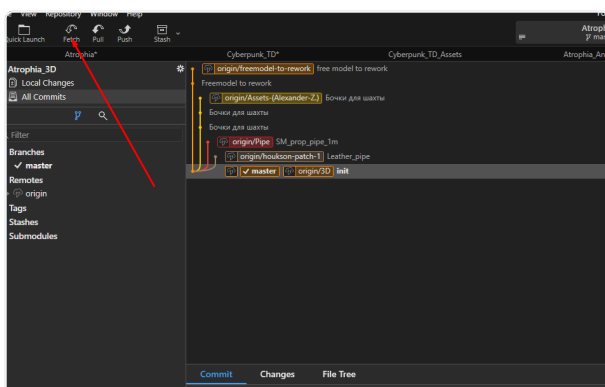
Как только вы взяли задачу, которая подразумевает загрузку в репозиторий - вы создаете ветку с правильным названием и пушите ее на сервер

Пример. Я получил задачу под номером 666 на создание модели шахты. Первым делом я иду в форк и создаю ветку от ветки мастера.

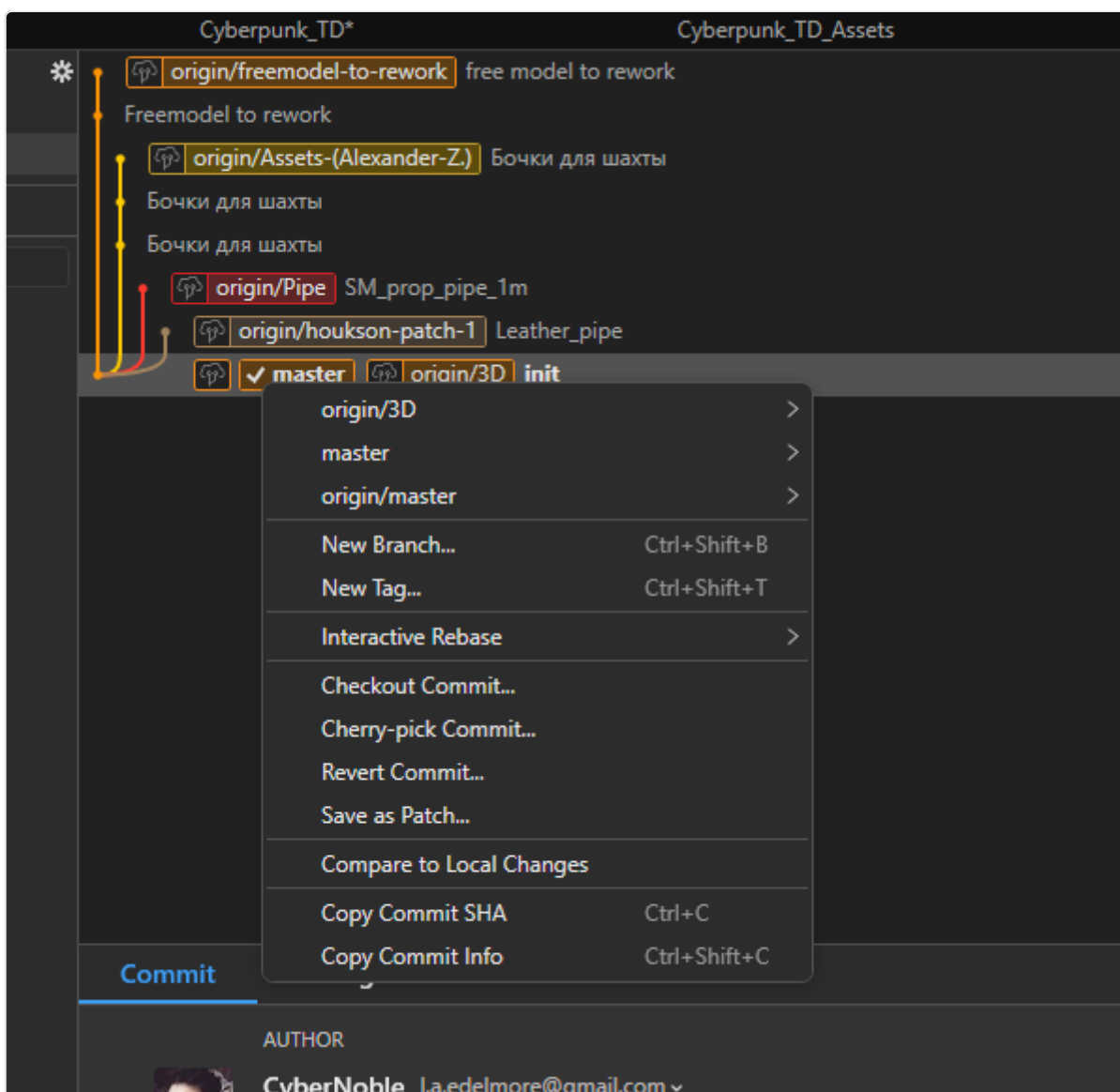
! Специально для Atrophia (Dev команды) - создаем ветку не от мастера, а от последней билдовой. Билдовая ветка называется так - v6.0.0 (цифры могут изменяться)

Чтобы это сделать, нужно:

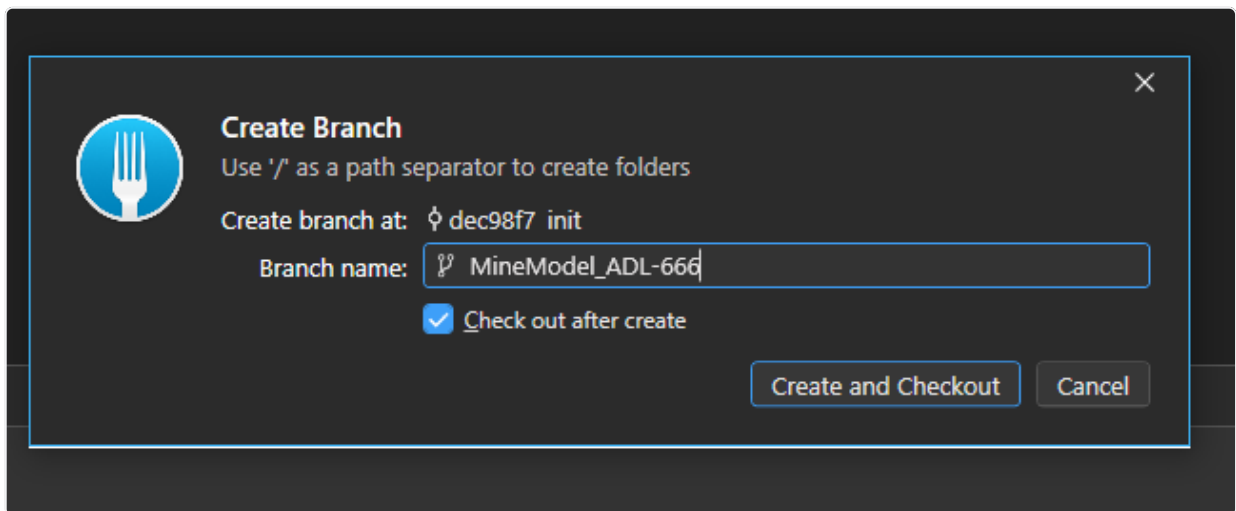
- Открыть форк и зайти на страницу нужного вам репозитория
- Обновить репозиторий с помощью кнопки fetch



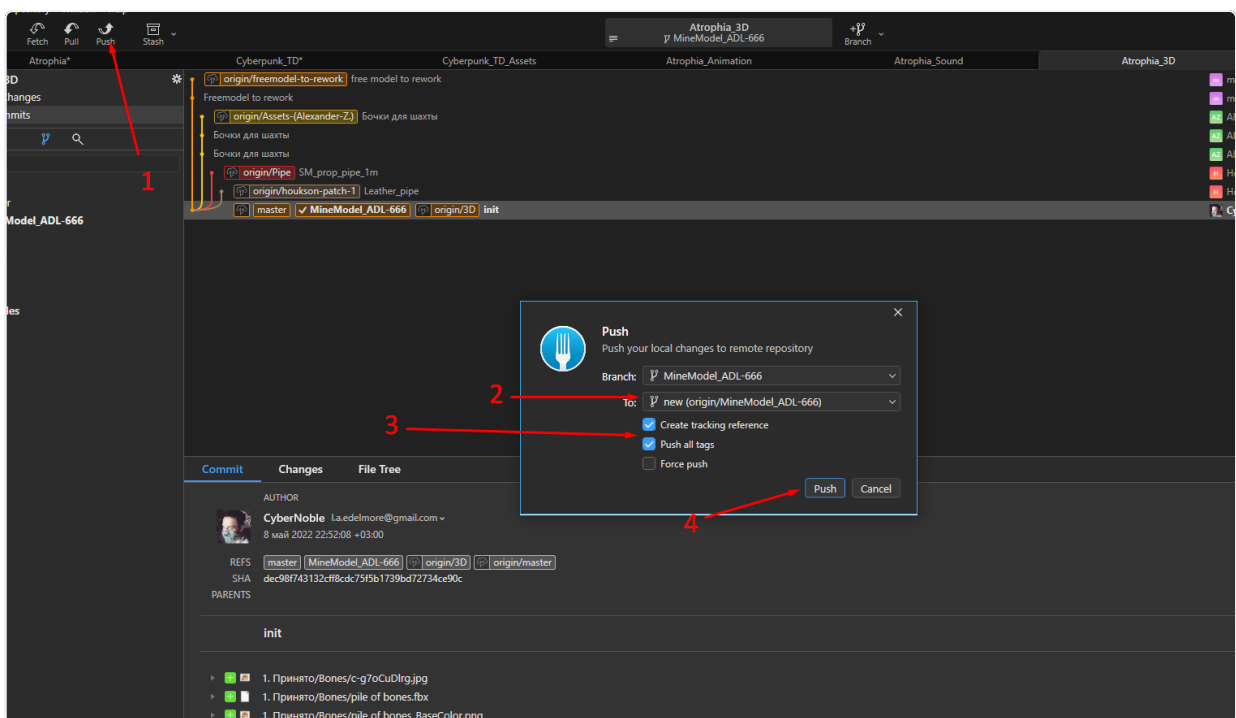
- Нажать ПКМ на коммите, где находится ветка мастера



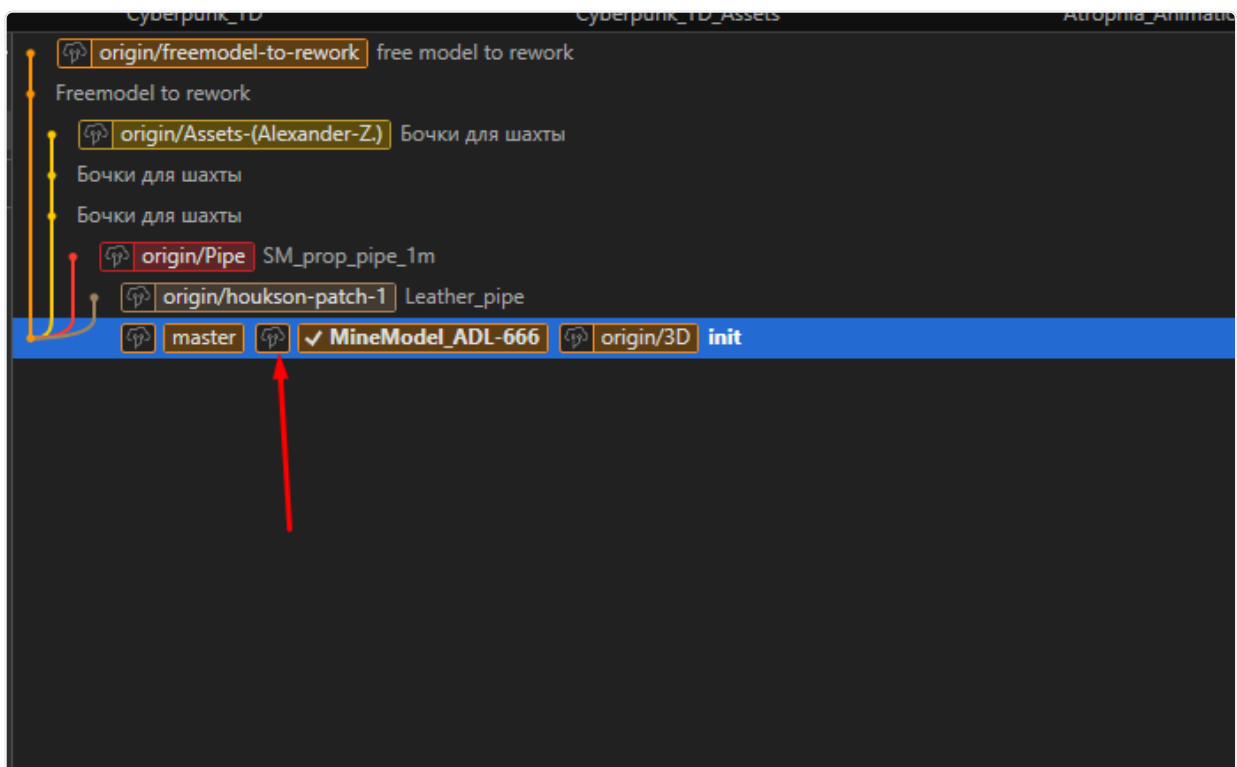
- Выбрать пункт New Branch и указать правильное название ветки. В моем примере - это




- Тыкаю галочку и нажимаю создать
- Затем нажимаю на кнопку пуша и загружаю ветку на сервер



- 1 - Нажимаем на кнопку
- 2 - Тут выбирается ветка, по умолчанию, если вы создали новую, то автоматически форк предложит создать такую же на сервере, можно ничего не менять. Если же вы хотите сделать пуш в другую ветку, то тогда нужно выбрать из выпадающего списка существующую ветку
- 3 - Ставим две галочки (первая галочка доступна только, если вы создали новую ветку)
- 4 - Кнопка пуша - отправляет вашу ветку на сервер
 - Если у вас, рядом с названием вашей ветки появилась иконка облачка, то поздравляю, вы справились!

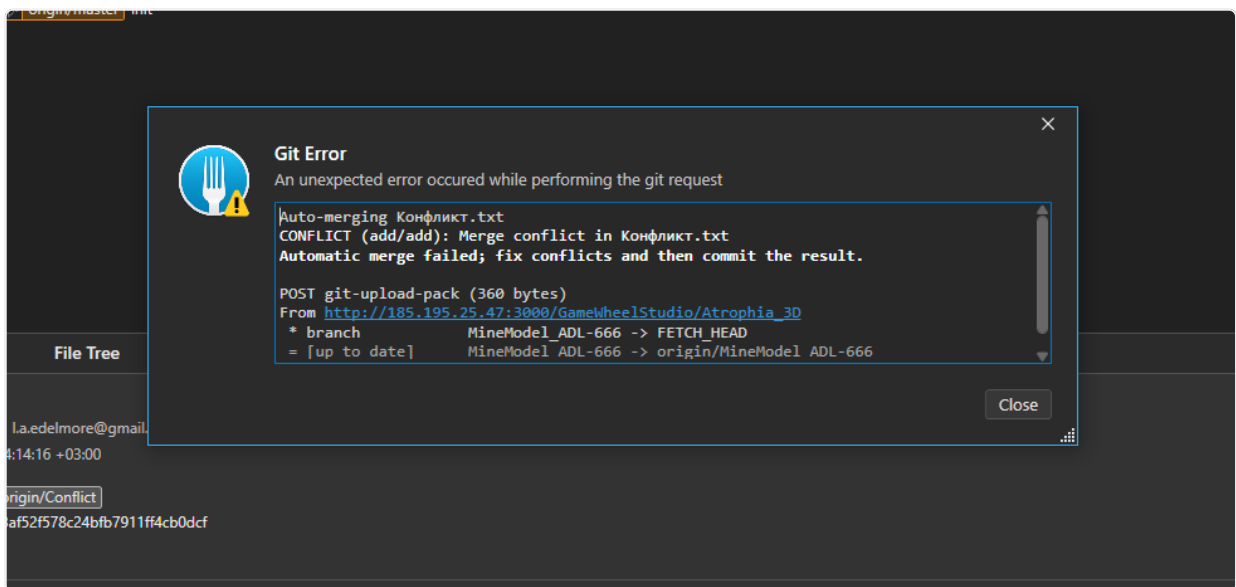


- Теперь, когда task рассмотрен - вы загружаете изменения, согласно пайплайну 
[Гайд по работе с Fork](#)
- После того, как вы все загрузили и положили все в правильные места, QA все проверили - подписываетесь вашему лиду, чтобы он заморозил изменения в мастер. Как только это сделано - ветка удаляется вами или лидом

5. Конфликты

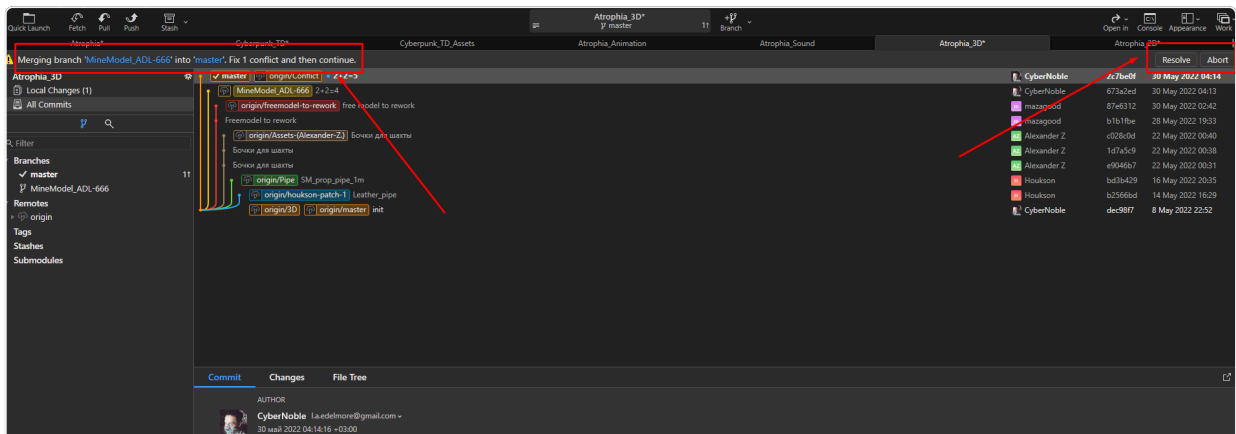
Иногда, когда вы мержите в себя или себя в другую ветку - возникают конфликты. Это такие ситуации, когда файл был изменен два раза и двумя разными способами. Форк не может определить какой из двух файлов важный, поэтому конфликты нужно решать руками.

Пример интерфейса конфликта

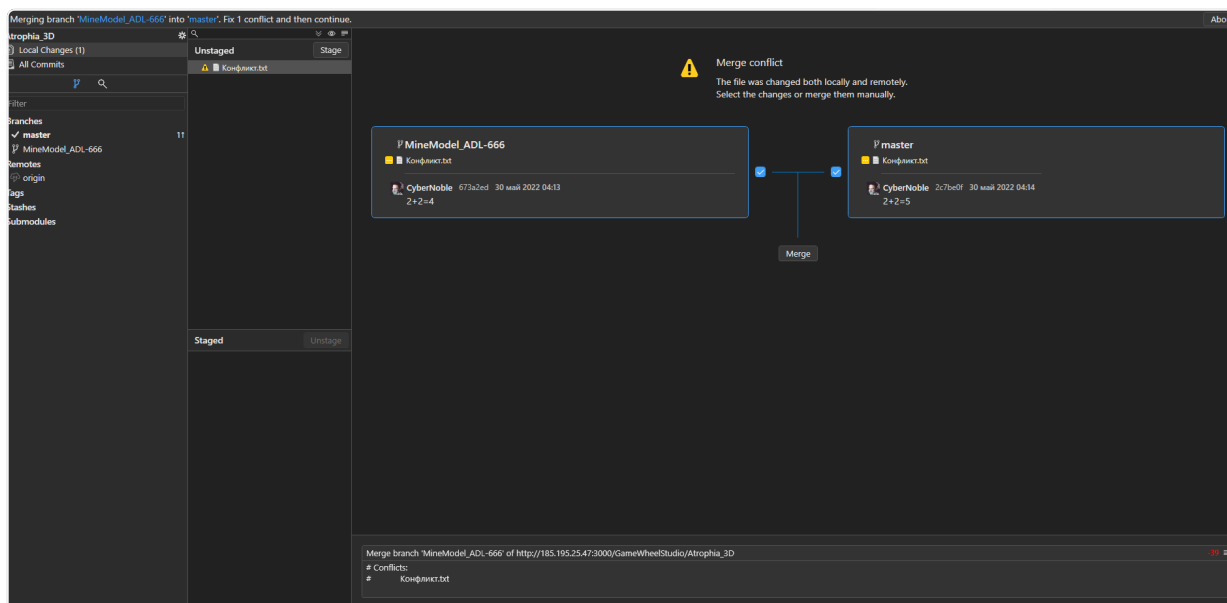


Флоу решения конфликтов:

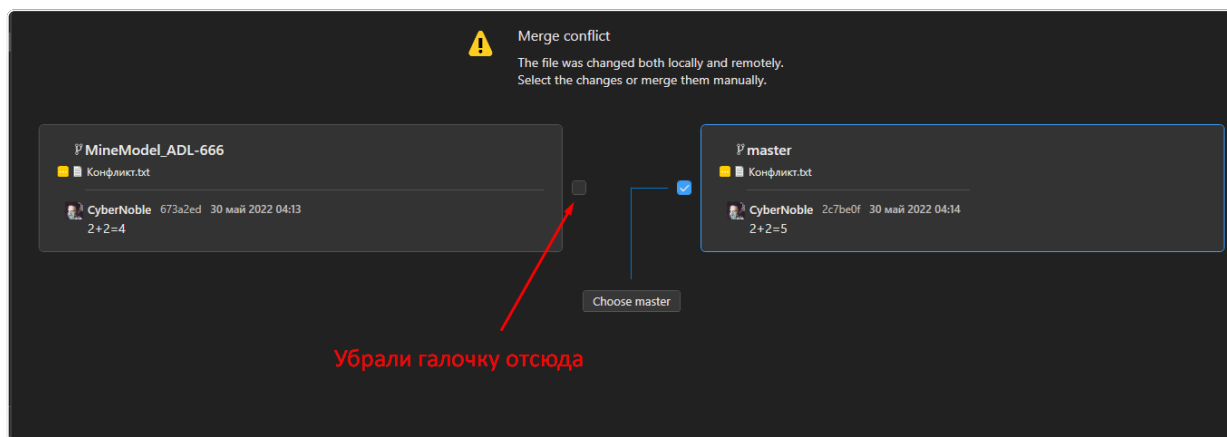
- Тыкаем кнопку Close и в форке в верхнем правом углу - появится две кнопки
 - Resolve - перенаправляет на страницу решения конфликта
 - Abort - отменить ваше действие, тыкайте, если не знаете как решить конфликт и сообщите вашему лиду



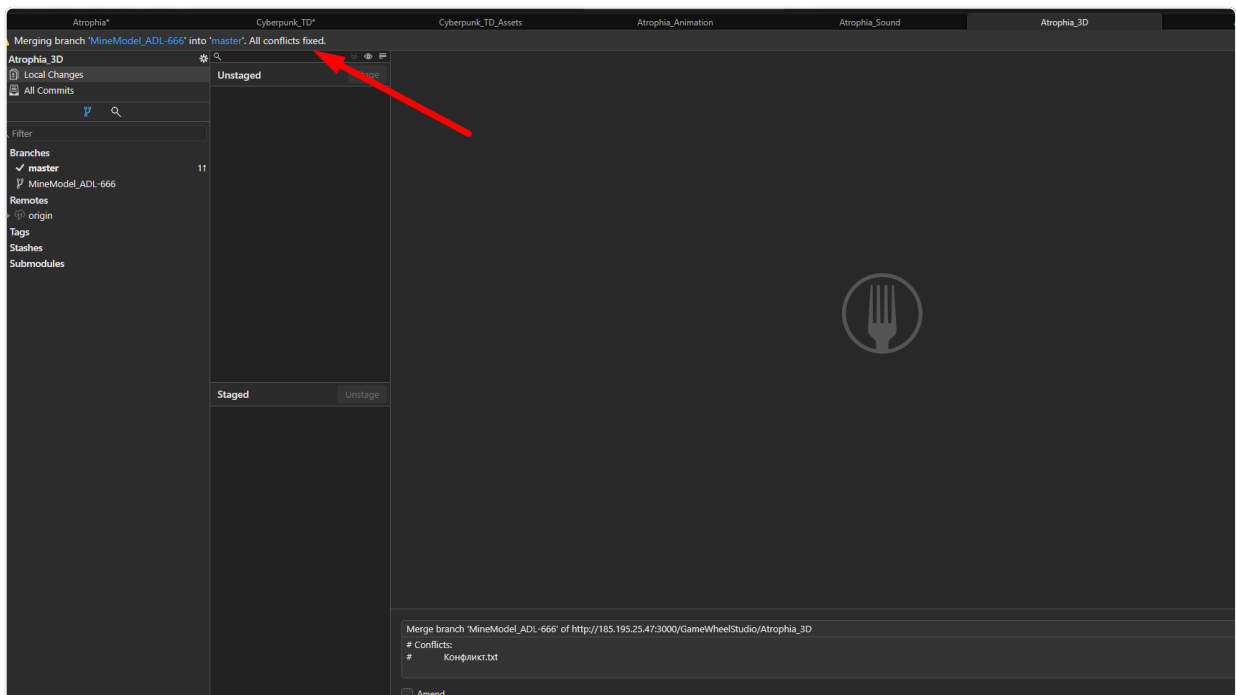
- Мы пойдем решать конфликт, поэтому тыкаем на resolve
- Тут у нас два файла, в которых были изменения, но разные. Вы должны решить какой из файлов при мерже вам нужнее



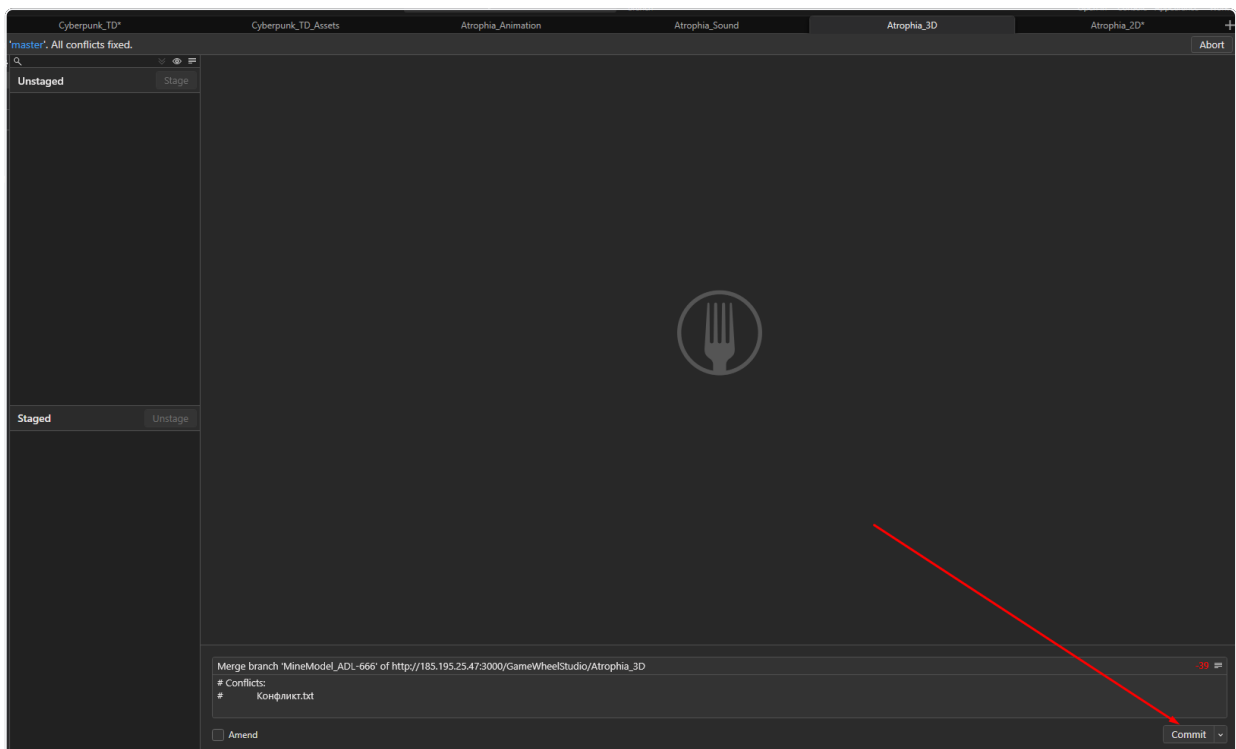
- Возможно, что один из файлов устаревший или человек, который делал изменения в вашем файле - случайно его затронул
- Тут банальный пример с текстовым файлом. В одном написано $2+2=4$, в другом $2+2=5$. Поскольку, второй вариант - это верное математическое утверждение - мы выбираем его
- Чтобы выбрать файл - на нем должна остаться галочка, нам нужно лишь убрать галочку с первого файла, где написан бред про $2+2=4$. Вот итог нашего выбора



- Теперь тыкаем кнопку между двумя файлами
- Если конфликты еще остались - форк автоматически предложит следующий конфликт, если вы все решили, то сверху слева надпись изменится на All Conflict Fixed



- А вам останется лишь сделать коммит ваших изменений. Форк автоматически собрал название коммита, файлы и описание, ничего писать не нужно, только тыкнуть сюда



- Теперь в дереве коммитов у вас появился новый коммит, его осталось только запустить, согласно этому флору [🎯 Гайд по работе с Fork](#)

