|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Корпоративных Информационных Систем | |
|  | |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ОТЧЕТ ПО ПРОЕКТНОЙ РАБОТЕ № 1**  на тему  **«Создание структуры проекта в системе контроля версий»** | | | |
| **по дисциплине** | | | |
| **«**Системная и Программная Инженерия**»**  **Тема проекта: «Многопользовательская игра «Tower battle»»** | | | |
|  | | | |
| Выполнили студенты группы  ИКБО-09-18,  ИКБО-08-18,  ИКБО-17-18. | | *Валяев Д.А,*  *Андреев И.К.,*  *Ахметов Д.Б.,*  *Корчиков М.Д.* | |
| Принял | | Копылова А.В. ассистент кафедры | |
| Отчет выполнен | «\_\_»\_\_\_\_\_\_\_2020 г. | |  | |
|  |  | |  | |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2020 г. | |  | |

Москва 2020

**Описание проекта**

Наша команда представляем вашему внимаю проект «Многопользовательская игра «Tower battle»». Не секрет, что есть много операторов связи и то, что конкуренция между ними довольно высока, однако чтобы привлечь дополнительную аудиторию, на наш взгляд, было бы целесообразно создать многопользовательскую игру с оригинальной тематикой. В данной игре игрокам предстоит за определенное время занять наибольшее количество вышек телекоммуникационной связи. Помимо простого захвата вышек, в игре присутствует система бонусов, которая позволяет игрокам увеличивать скорость передвижения и скорость захвата вышек. На наш взгляд подобная концепция должна разнообразить геймплей и заставить игроков придумывать разнообразные стратегии победы.

Почему именно мы решили разработать игру «Tower battle», с экономической точки зрения:

1. Экономия на продвижении, так как игры подобного формата становятся вирусными и игроки с удовольствием делятся игрой с друзьями.
2. Экономия на разработке, так как все используемые технологии бесплатны.

Последующее описание проекта не является полностью утвержденным, перечень каких-либо деталей и функций могут быть изменены в период доработки и разработки каких-либо новых вещей, но данное описание позволяет примерно понять, что из себя представляет игра в данный момент времени.

**Роли в проекте и их описание**

Всегов проекте участвуют 4 человек, у каждого из них есть своя роль и определенные задачи, которые вовремя должны быть выполнены. Но никто не исключает того момента, что каждый из участников может помогать другому участнику в решении какой-либо из задач, если такая необходимость всё-таки появиться.

**Состав команды и роли:**

1) Андреев Иван Константинович - Team Leader, Tester. В обязанности входит: руководить всей командой; отслеживать сроки выполнения задач; оперативно реагировать, если начинаются проблемы с качеством; мотивировать команду работать более эффективно; принимает решения, когда возможны альтернативные варианты при выборе развития продукта в ходе разработки; рисует график прогресса проекта; разработка плана проекта; оценка рисков, контроля функциональных и стоимостных рамок и ежедневная работа с командой на проекте, тестирование программного продукта.

2) Ахметов Данила Борисович – QA (Quality assurance), Technical Writer, Tester. В обязанности входит: тестирование программного продукта; тестирования; техническое исследование программы для получения информации о её качестве с точки зрения определённого круга заинтересованных лиц. А так же основная задача Technical Writer — написание документа, который бы удовлетворял определённым требованиям. Требования могут определяться как нормативными актами, существующими в отрасли применения продукта, так и различными целями, которые организация-разработчик ставит перед собой.

3) Валяев Данила Андреевич – Tech Leader, Developer, User Interface Designer. В обязанности входит: заниматься архитектурой; ревью кода; решением технических проблем; координацией работы; написание кода; рецензирование кода за другими; принимать решение, как конкретно будет реализована та или иная функциональность в приложении.

4) Корчиков Михаил Дмитриевич – Database Administrator, Project Manager, Tester. В обязанности входит: выработка требований к базе данных, её проектирование, реализацию, эффективное использование и сопровождение; проектирование базы данных; оптимизация производительности базы данных; обеспечение безопасности в базе данных; обеспечение целостности баз данных; принимать решение, как конкретно будет реализована та или иная функциональность в приложении; определять весь ход создания и развития продукта; принимать решение, что должно быть сделано, оценивать, соответствует ли реализация концепции продукта или нет; тестирование программного продукта.

**Используемые в проекте технологии**

Какие технологии участвовали в создании игры:

1. Был использован Unity, как игровой движок.
2. Модели башен и игрока были сделаны в Blender.
3. Подключен Photon PUN для сетевого режима.

Теперь подробно рассмотрим каждую из технологий.

1. Unity(игровой движок) — межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Выпуск Unity состоялся в 2005 году и с того времени идёт постоянное развитие. Основными преимуществами Unity являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. К недостаткам относят появление сложностей при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек .На Unity написаны тысячи игр, приложений, визуализации математических моделей, которые охватывают множество платформ и жанров. При этом Unity используется как крупными разработчиками, так и независимыми студиями.
2. Blender  — профессиональное [свободное и открытое программное обеспечение](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%B8_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) для создания [трёхмерной компьютерной графики](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%91%D1%85%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0), включающее в себя средства моделирования, скульптинга, анимации, симуляции, [рендеринга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BD%D0%B4%D0%B5%D1%80%D0%B8%D0%BD%D0%B3), постобработки и монтажа видео со звуком, компоновки с помощью «узлов» ([Node Compositing](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%B8%D0%BD%D0%B3" \o "Композитинг)), а также создания 2D-анимаций. В настоящее время пользуется большой популярностью среди бесплатных 3D-редакторов в связи с его быстрым стабильным развитием и технической поддержкой.
3. Какие особенности Photon PUN

### *Photon PUN – Облако В Реальном Времени*

КАЛАМБУРНЫЕ игры размещаются в нашем глобально распределенном фотонном облаке, чтобы гарантировать низкую задержку и кратчайшее время туда и обратно для ваших игроков по всему миру.

### *Многопользовательский режим*

Соединяйтесь, сопоставляйте и играйте: Каламбур-это прочная основа для любого типа многопользовательской игры на основе комнаты в Unity 3D. вы концентрируетесь на создании отличного названия, в то время как мы заботимся о бэкэнде игры.

### Межплатформенный

Экспорт на мобильные, настольные, веб-или консольные устройства: Photon-это стандартный кроссплатформенный многопользовательский сервис и мировой рекорд № 1 для игр Unity.

### *Высочайшая Масштабируемость*

Игры, построенные с помощью каламбура, масштабируются легко и автоматически в фотонном облаке: от нескольких до десятков тысяч одновременных пользователей. Справедливое и прозрачное ценообразование включено в стоимость. Доказанный. Успешно.

### *API для подбора партнеров*

Вы подбираете игроков в игры случайным образом или с помощью параметризованного поиска. Или вы создаете список открытых комнат и позволяете игрокам выбрать одну. Это работает. Надежно. На мобильных телефонах, ПК, консолях и в интернете.

### *Настройка*

Какую многопользовательскую игру вы хотите построить? FPS или аркадный шутер? RPG или RTS? Гонщик или спортсмен? Моба или ТД? Еще один вид экшена? PUN поддерживает любой тип комнатных игр на любой популярной игровой платформе.

### *Клиент-Сервер*

Это просто работает. Никаких сквозных проблем, будь то с надежными UDP, TCP, HTTP или Websockets. Высокоскоростная клиент-2-серверная архитектура Photon - это самая прочная основа индустрии для ваших игр.

### *Непревзойденная гибкость*

Фотон очень гибкий: вы можете сопоставить пользователя iOS Game Center с кем-то, кто использует сервисы Google Play, аутентификация пользователя через Facebook или добавление пользовательской аутентификации, используйте игровой сервис или крюк в вашем собственном бэкэнде.

### *Беззаботное Ценообразование*

Справедливое и прозрачное: 100% использование на основе параллельных пользователей (CCU). Гибкость: начните с бесплатного плана, повышайте или понижайте рейтинг в любое время. Беззаботность: отсутствие немедленного ограничения при превышении CCU с любым ежемесячным планом. Идеально подходит для запуска вашей игры.

**Ход работы**

1. Распределение ролей в команде, а так необходимо завести канбан-доску для полного отслеживания задач каждого члена команды.

Для чего нужно распределение обязанностей между членами команды? Все очень просто, распределение обязанностей – это необходимая часть работы при создании какого-либо проекта. Так как каждый член команды понимает, что ему нужно делать на определенном этапе. Следовательно, можно сделать вывод, что это решение уменьшает количество затраченного времени и энергии, а так позволяет любому из участников команды помочь другому участнику.

#### С распределением ролей в команде можно ознакомиться в разделе “Роли в проекте и их описание”. Для отслеживания задач и этапов проекта, наша команда решила воспользоваться канбан-доской на github. Создается канбан-доска в данном случае с названием “[Tele2SolutionsDays](https://github.com/Reality-Ctrl/Tele2SolutionsDays/projects/1)“, на которой все члены команды могут вносить свои изменения, предложения, а так же отслеживать план работ.

Наша доска на данном этапе проекта состоит из 3 колонок ( to do – делать, In progress – в процессе, Done – сделано ). В колонке to do – сейчас 3 пункта, In progress – 5 пунктов, Done – 6 пунктов. Пока в проекте столько задач, но их количество постепенно будет только расти.

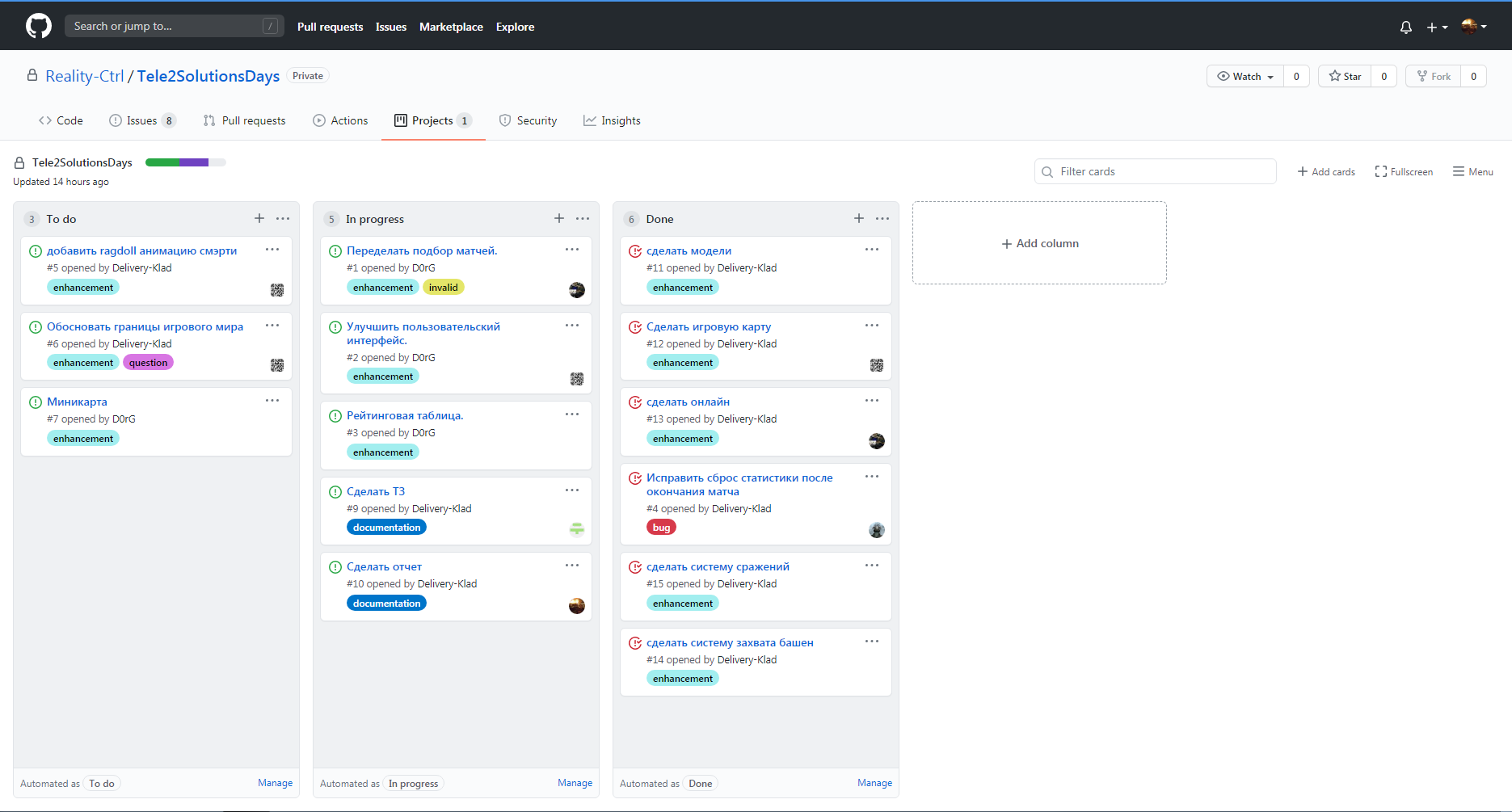


Рис.1 Общий вид канбан-доски

При нажатии на один из пунктов (задач), справа появляется подробная информация об этой задаче.

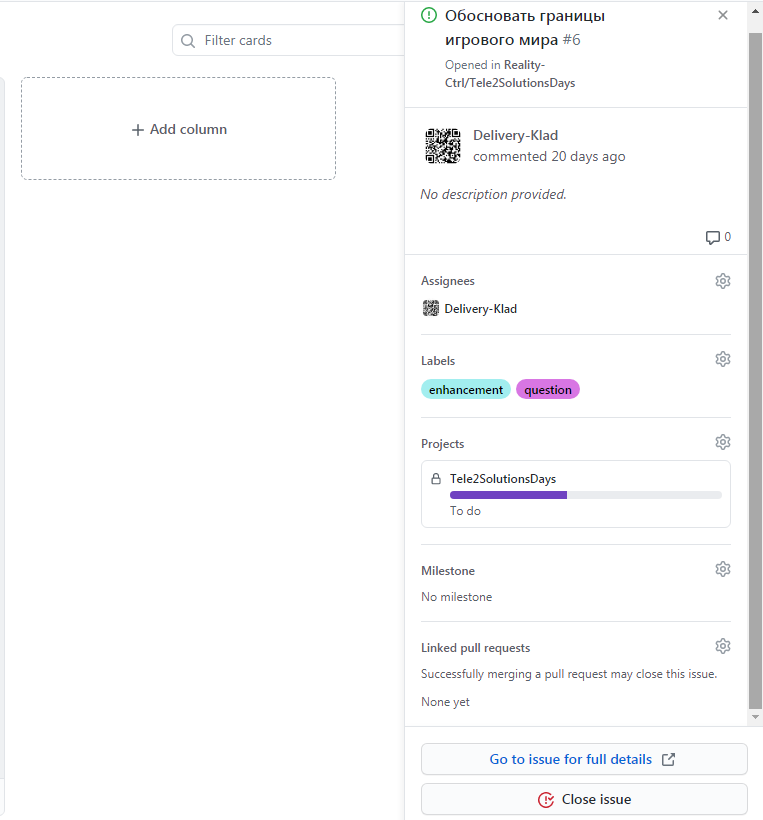


Рис.2 Подробная информация

1. Создание Git-репозитория проекта и обеспечение доступом к репозиторию всем членам команды.

К сожалению, репозиторий был создан еще на ранней стадии разработки проекта, поэтому его начальное создание наша команда не сможет показать. Сам репозиторий создал один из нас, а так же был предоставил доступ каждому члену команды.

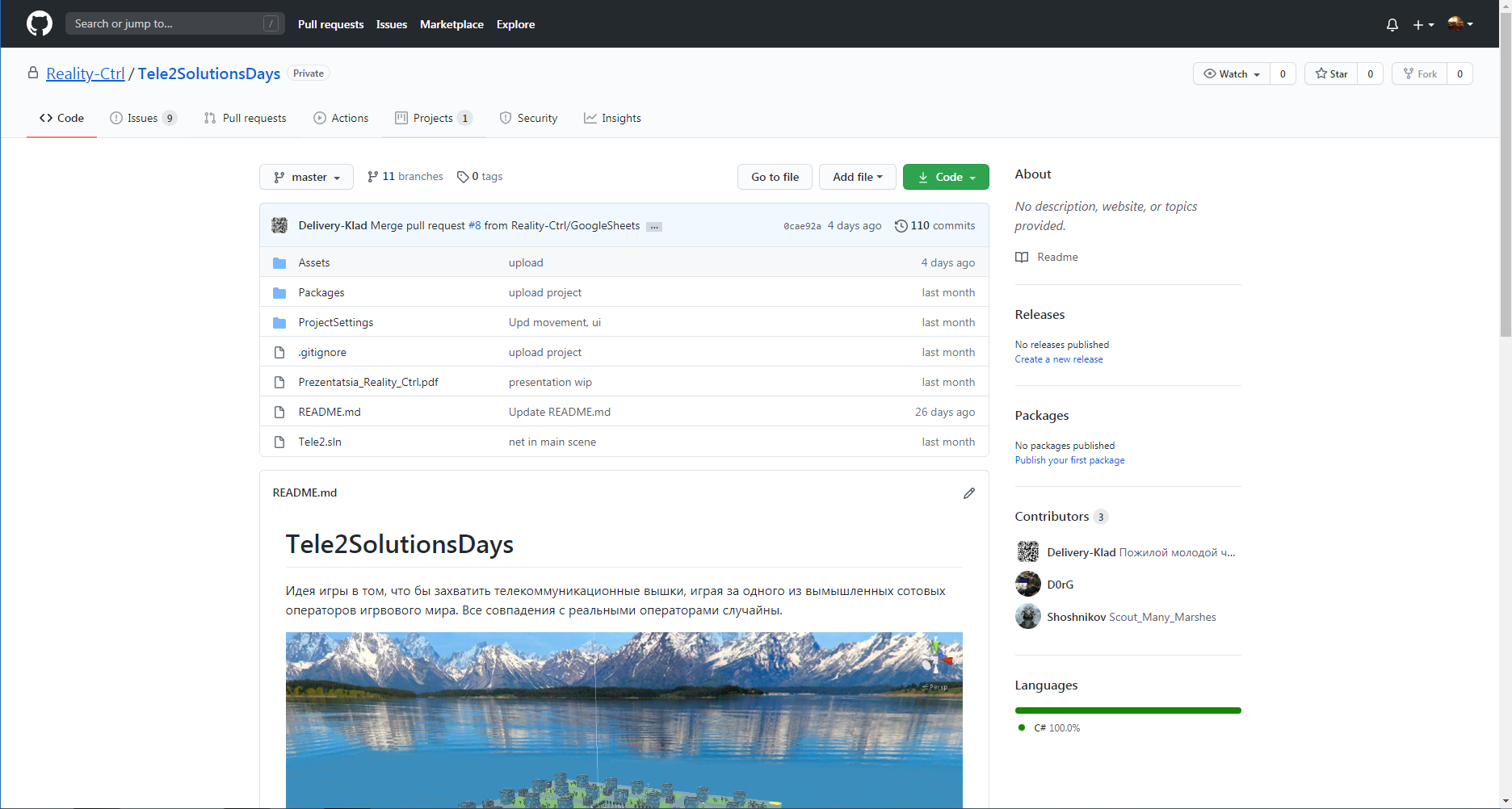


Рис.3 Страница репозитория на GitHub

Дальше команда начала создавать ветки, их всего 11.

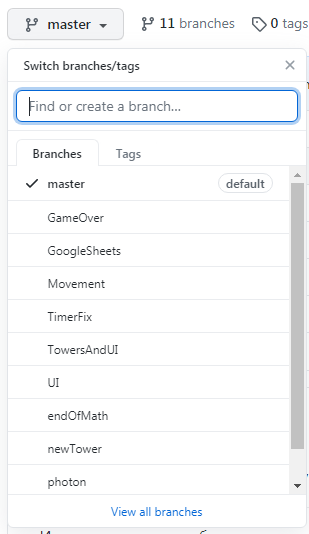


Рис. 4 − 11 веток

1. Загрузка кода проекта в Git-репозиторий

Так как наша команда разрабатывает многопользовательскую игру, нами было принято решение создать 11 веток в репозитории из-за огромного масштаба нашего проекта.

Описание веток:

* 1. master – является основной веткой, в ней расположены главные функции(основной код).
  2. GameOver - реализована система окончания игры.
  3. GoogleSheets – реализовано добавление таблиц гугл.
  4. Movement – реализовано передвижение игроков.
  5. TimerFix – реализовано исправление работы таймера.
  6. TowersAndUI- реализовано исправление башен и интерфейса.
  7. UI - работа с интерфейсом.
  8. endOfMath - окончание игры (меню).
  9. newTower – добавление новой модели.
  10. photon – работа с онлайном.
  11. Speed – скорость игры.

Дальше загружаем весь наш код, в данные ветки. Весь этот код был залит в репозиторий месяц назад, поэтому саму процедуру загрузки мы показать не сможем.

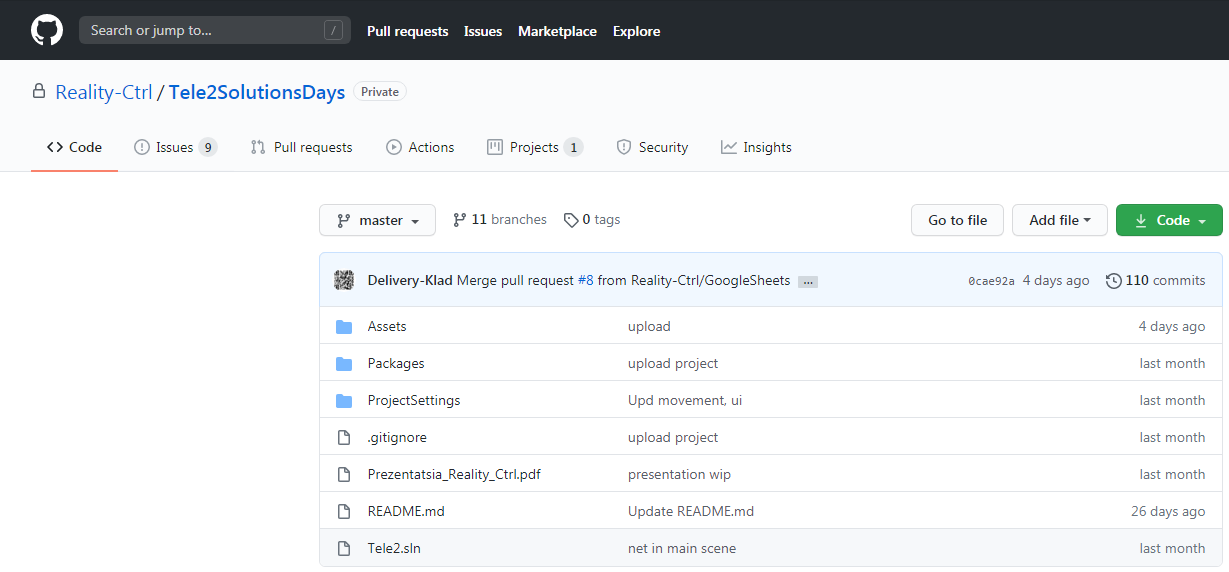


Рис. 5 − Загруженный код в Git-репозиторий в ветку master

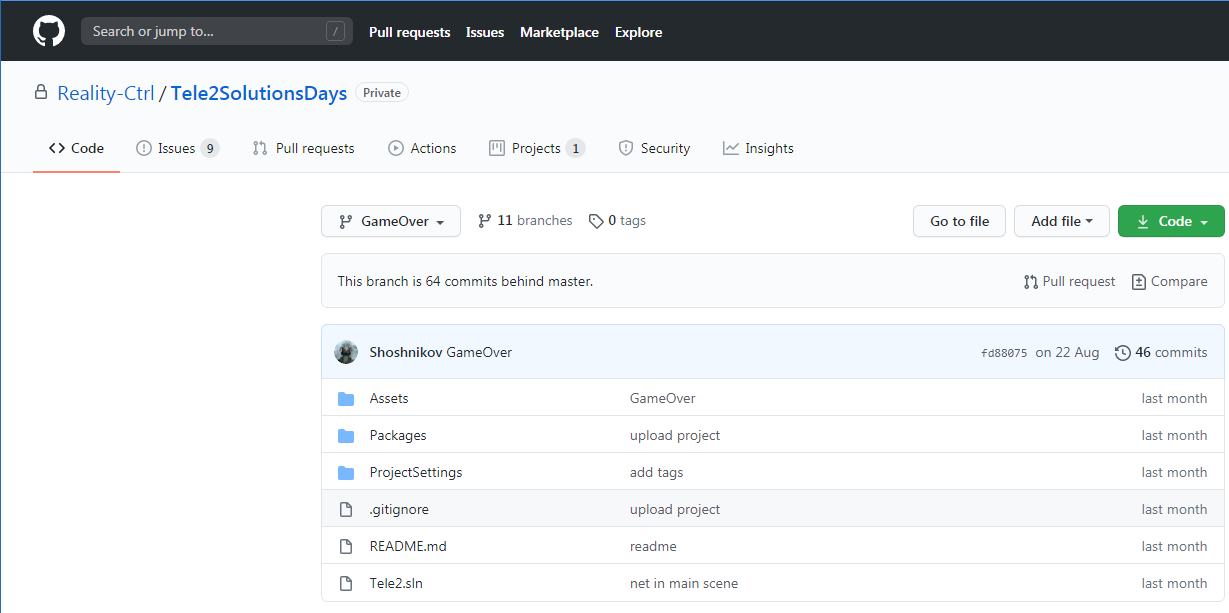


Рис. 6 − Загруженный код в Git-репозиторий в ветку GameOver

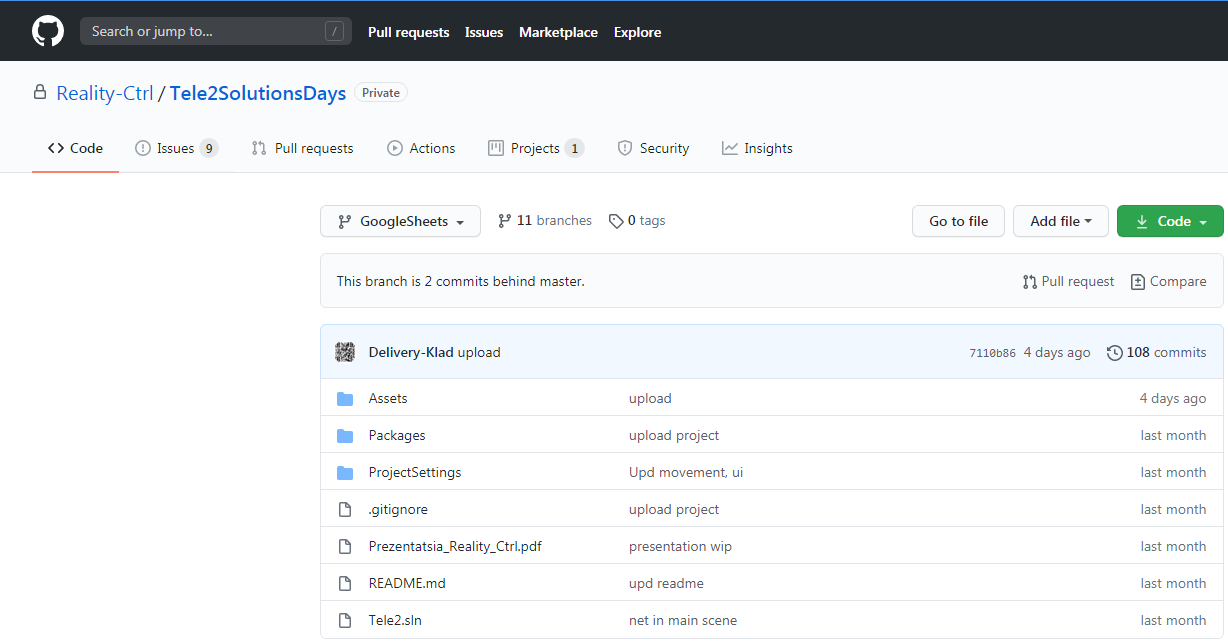


Рис. 7 − Загруженный код в Git-репозиторий в ветку GoogleSheets

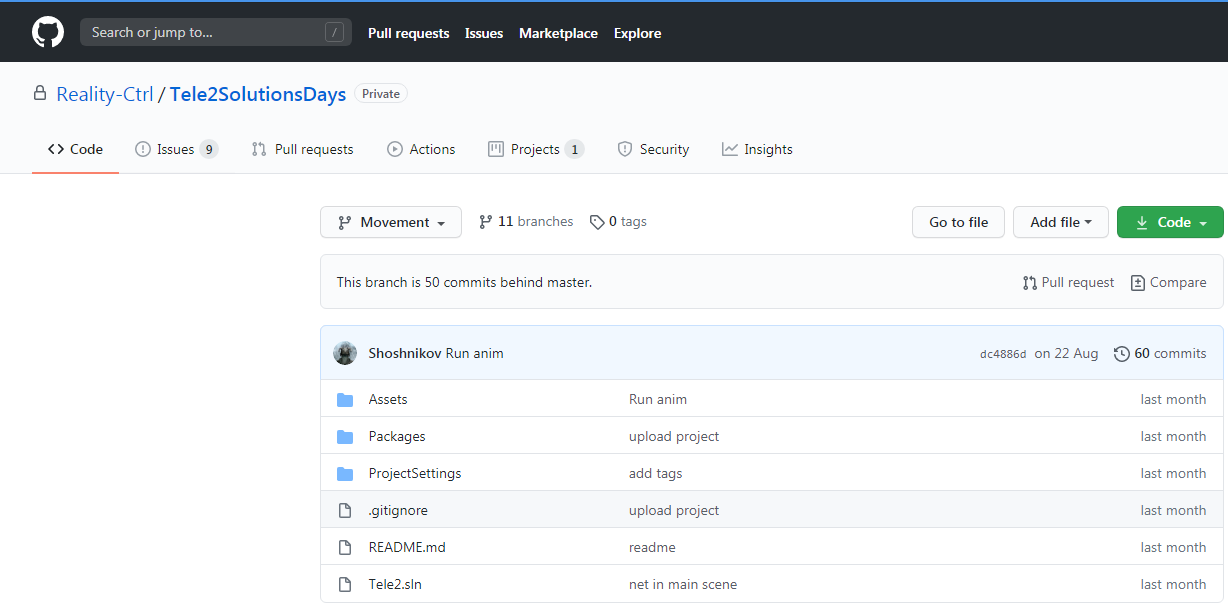


Рис. 8 − Загруженный код в Git-репозиторий в ветку Movement

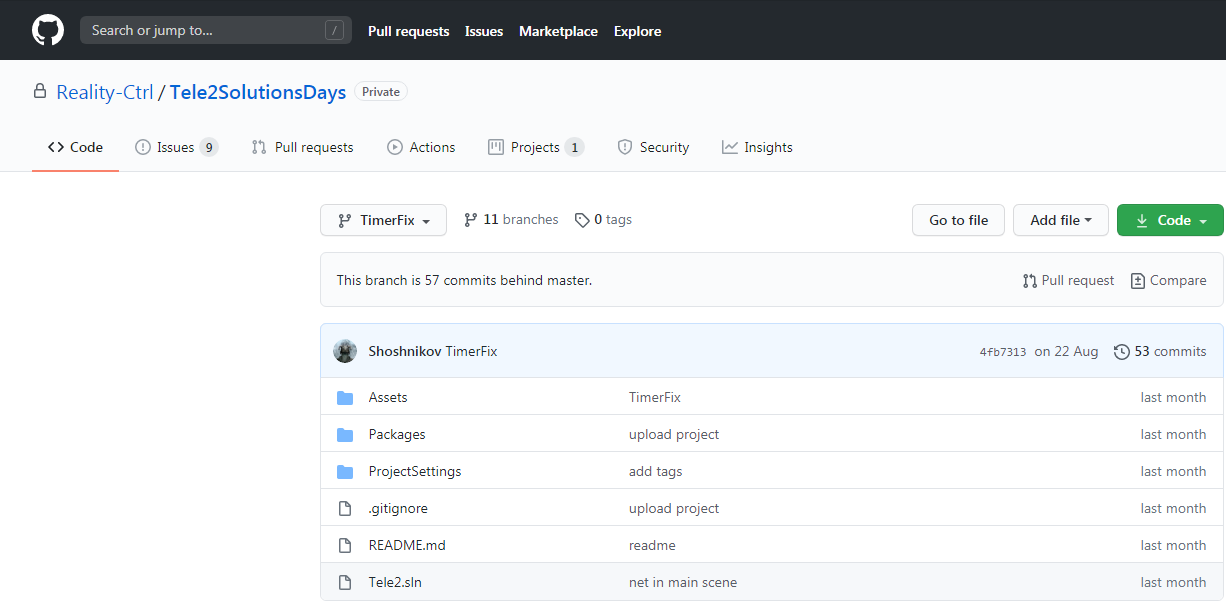


Рис. 9 − Загруженный код в Git-репозиторий в ветку TimerFix

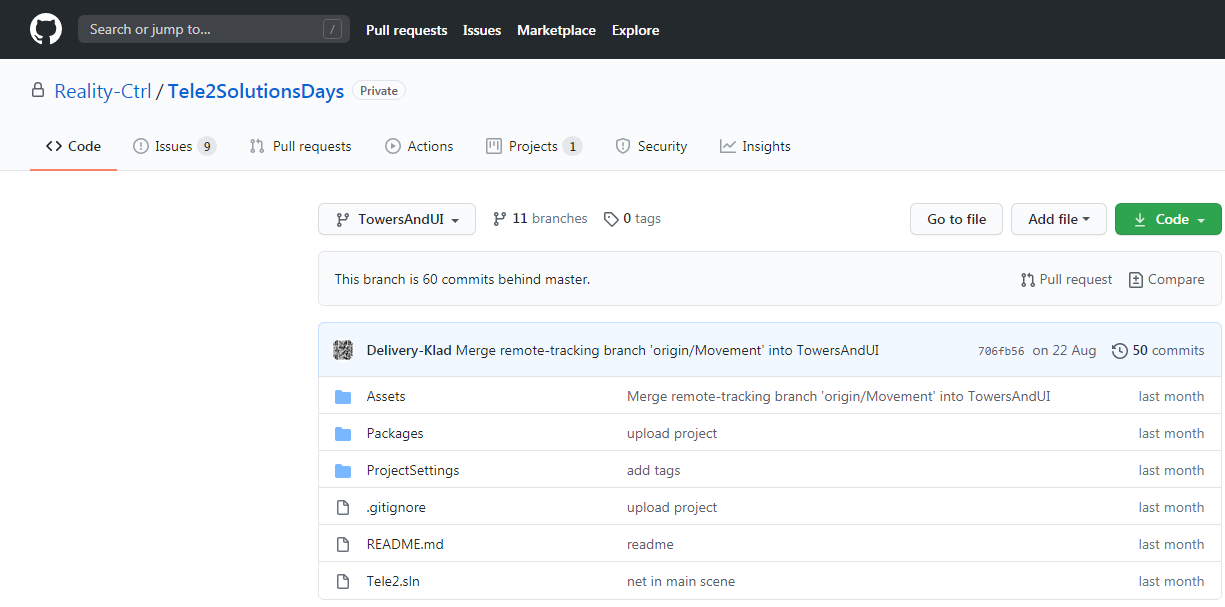


Рис. 10 − Загруженный код в Git-репозиторий в ветку TowersAndUI

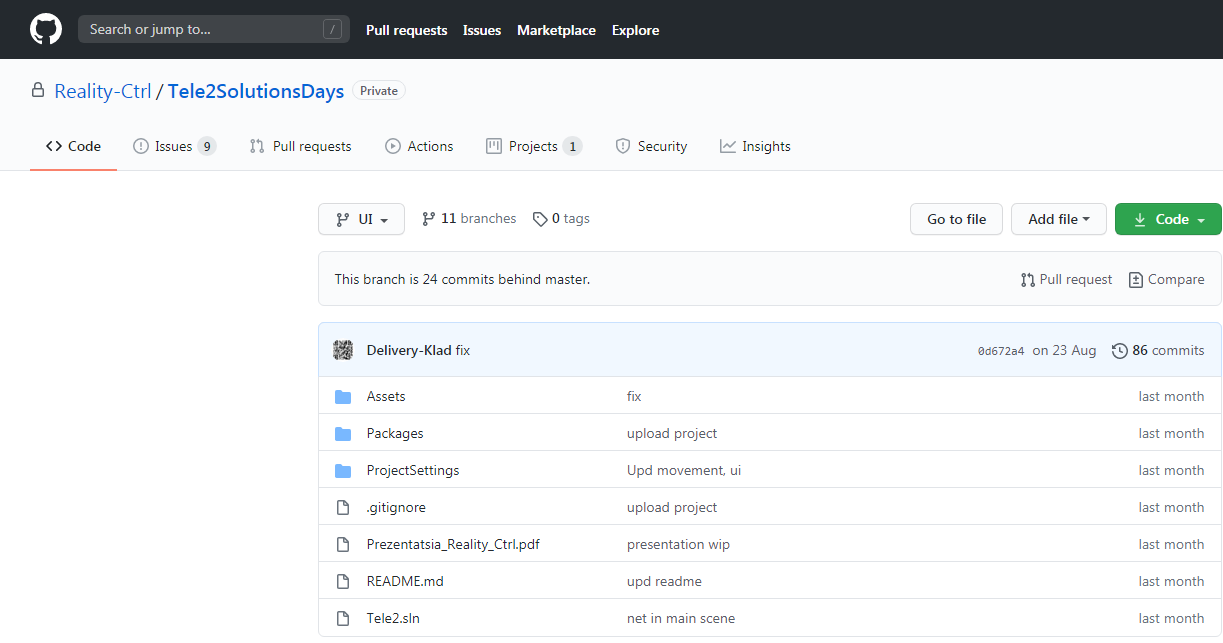


Рис. 11 − Загруженный код в Git-репозиторий в ветку UI

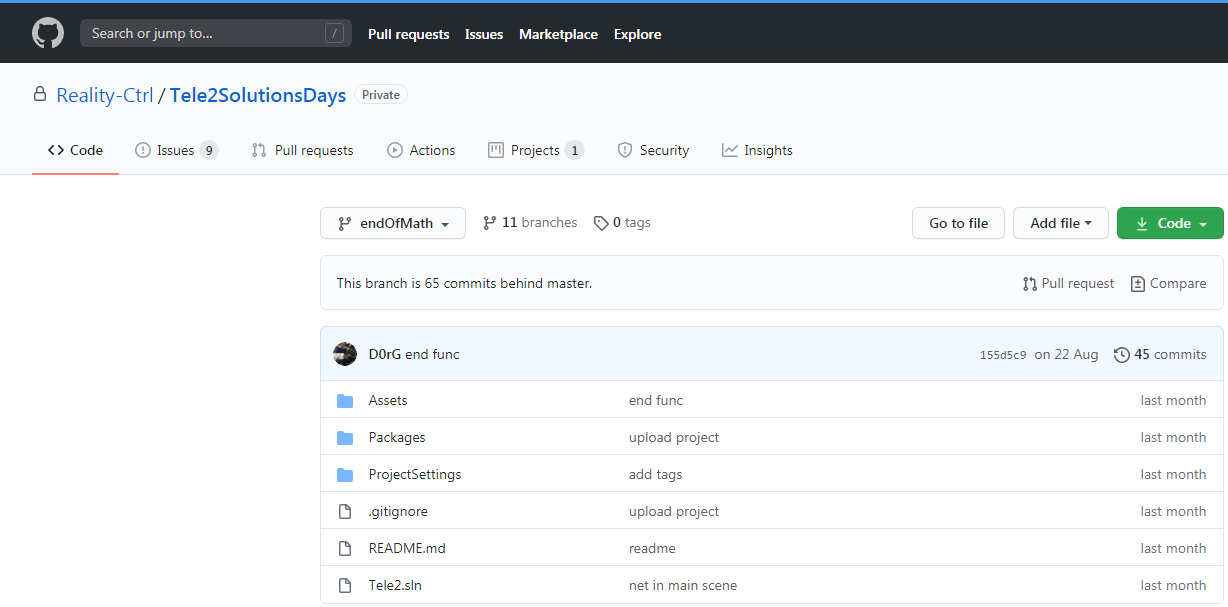


Рис. 12 − Загруженный код в Git-репозиторий в ветку endOfMath

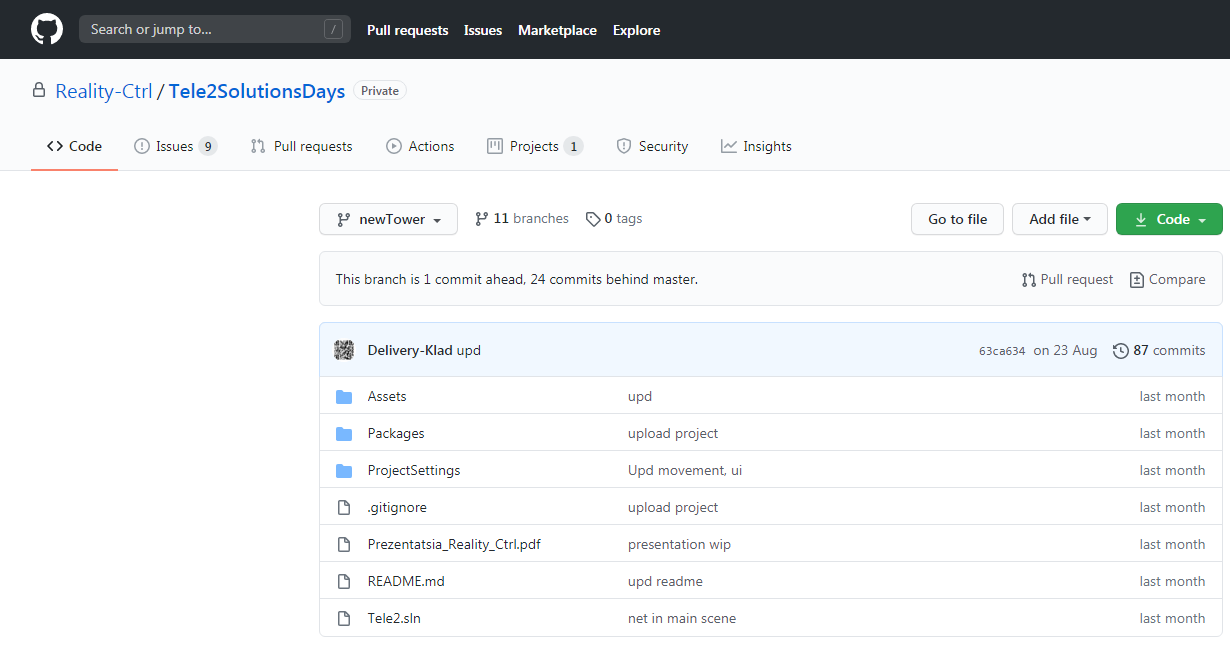


Рис. 13 − Загруженный код в Git-репозиторий в ветку newTower

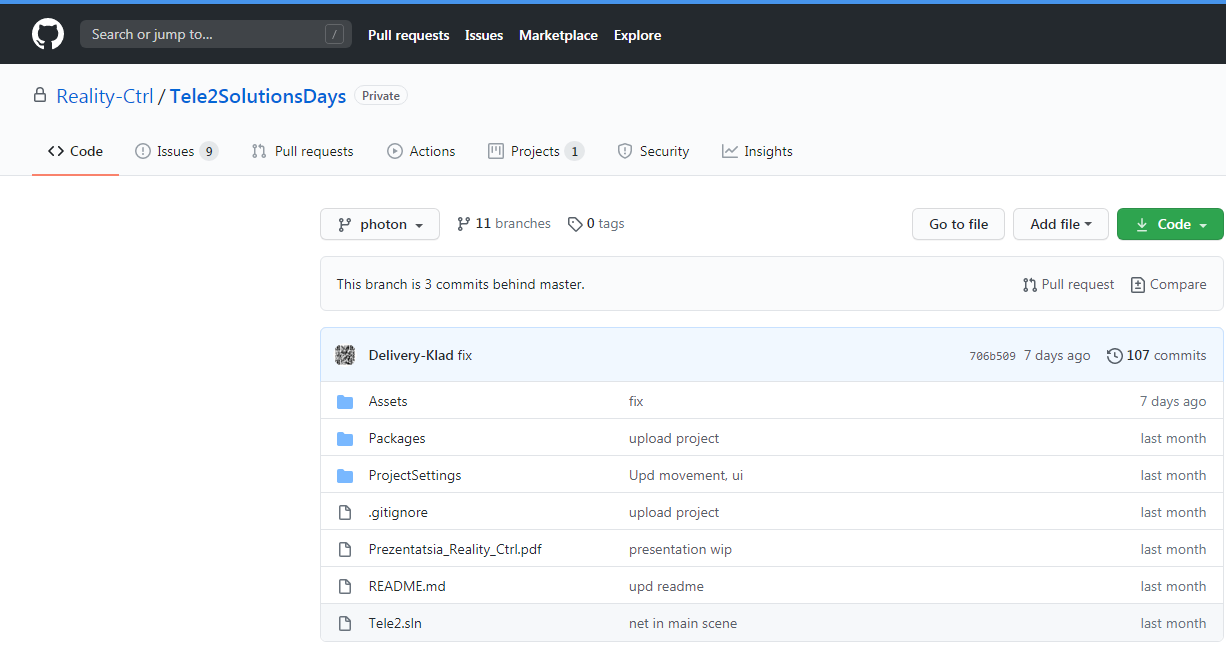


Рис. 14 − Загруженный код в Git-репозиторий в ветку photon

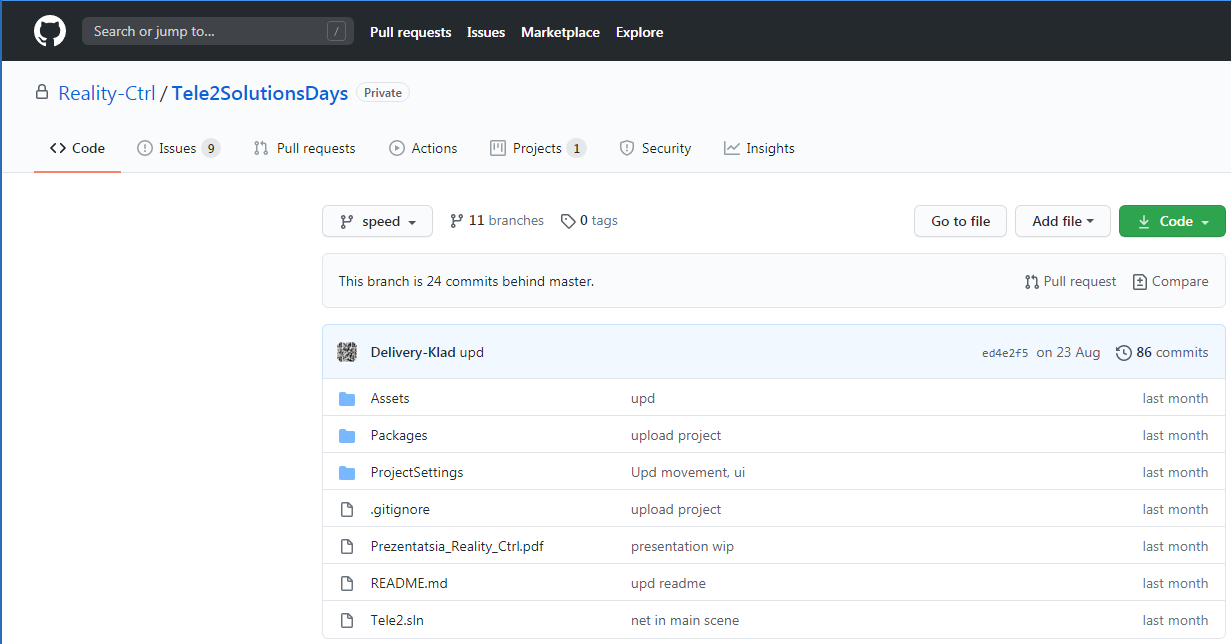


Рис. 15 − Загруженный код в Git-репозиторий в ветку speed

Для примера, рассмотрим, как отображается код в ветке master.

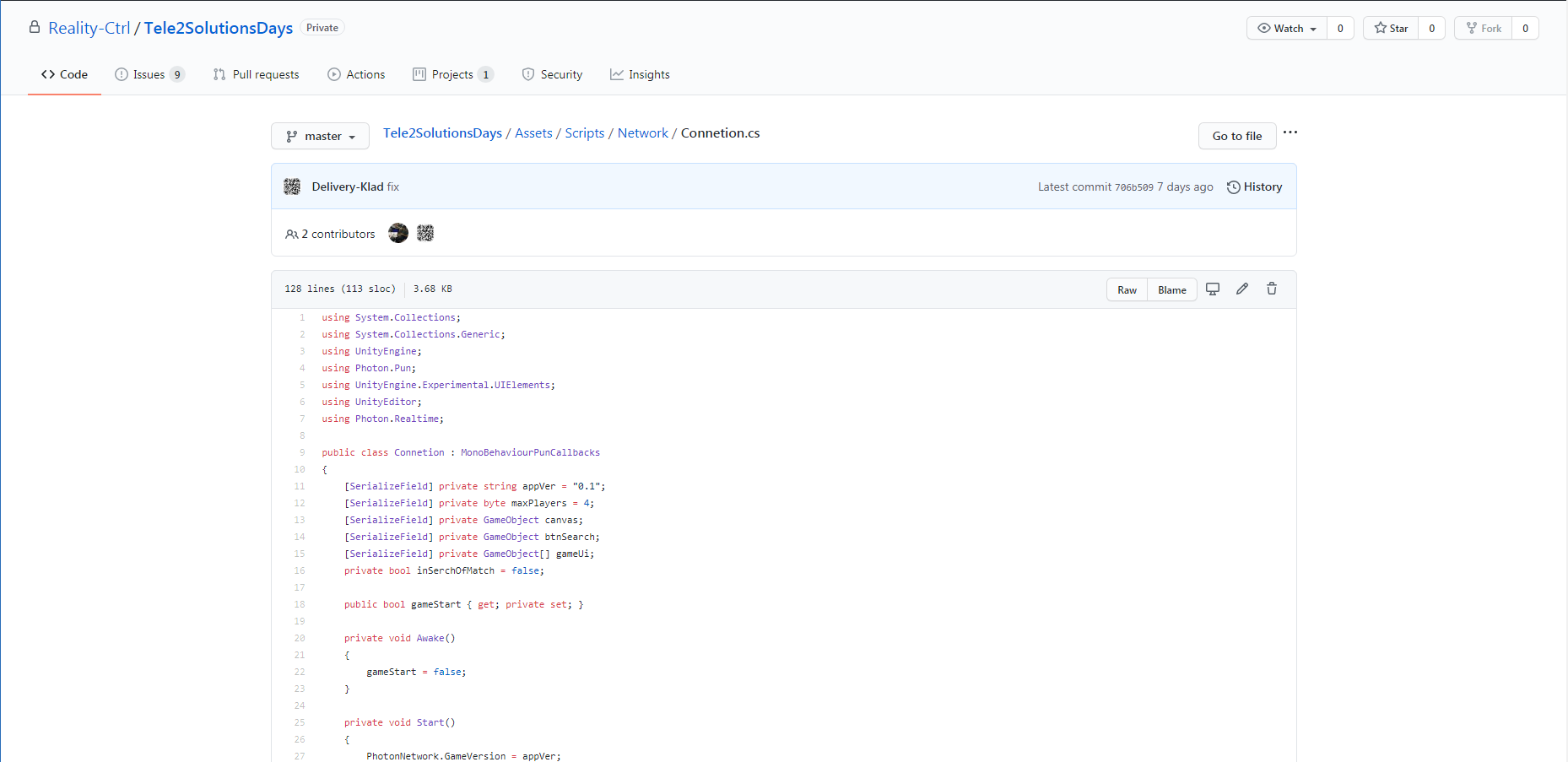


Рис. 16 − Код в Git-репозитории master

1. Добавление новых файлов в одну из веток после доработки.

Сейчас рассмотрим добавление файла в одну из веток после доработки, для этого.

За эти две недели мы доработали некоторые файлы и добавили их ветку GoogleSheets в папку Google Sheets to Unity.

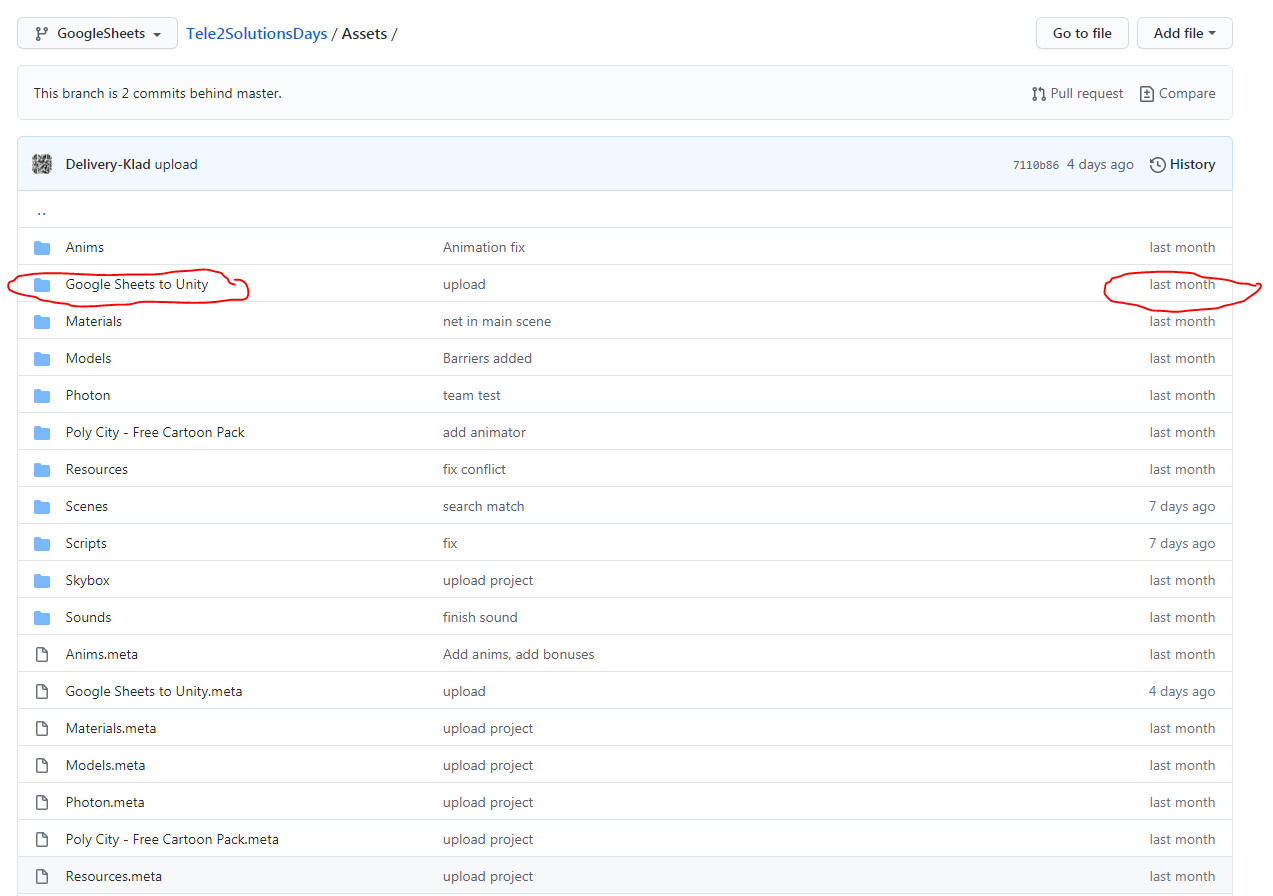
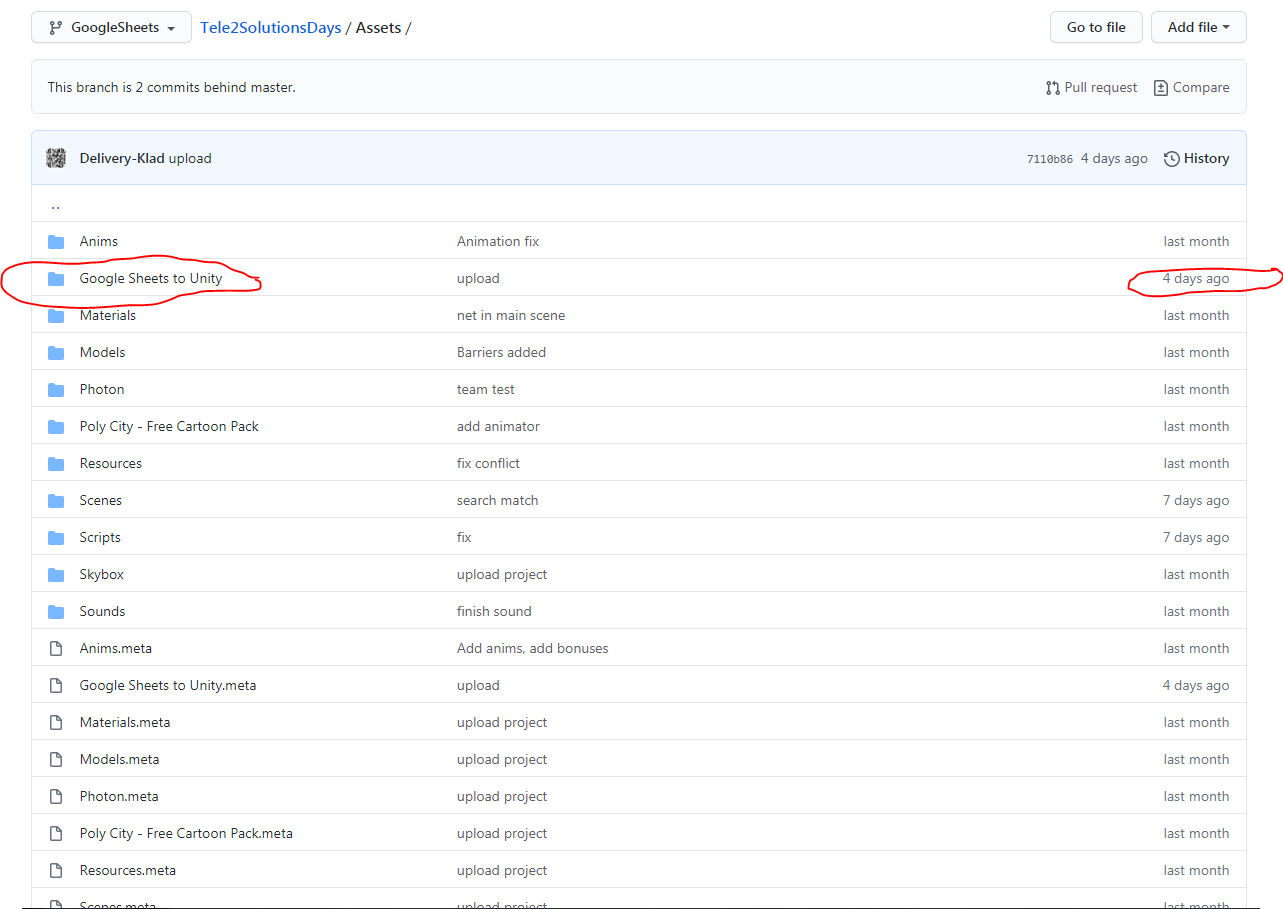


Рис. 17 - Как выглядела папка Google Sheets to Unity до добавления переработанных файлов(2 недели назад).



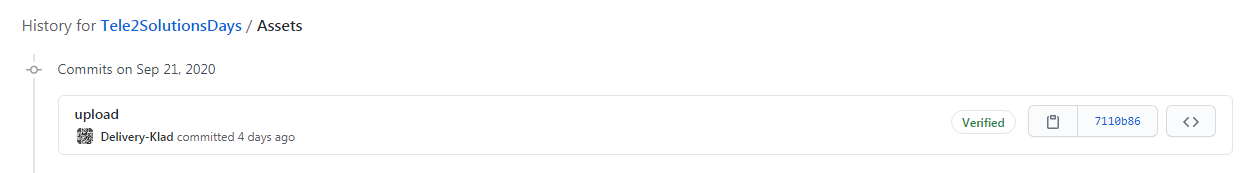


Рис. 18 - Как выглядела папка Google Sheets to Unity после добавления переработанных файлов(4 дня назад).

Изменение произошли 4 дня назад.

**Дальнейшие действия в разработки проекта**

Цели, которые мы перед собой поставили, после данного этапа

* Отслеживание этапов решения задач, каждого члена команды.
* Повторный анализ нашего проекта, для добавления дополнительных компонентов.
* Разработка алгоритмов.

**Вывод**

В данной практической работе мы распределили роли между членами команды, а так же создали доску нашего проекта на GitHub, которая поможет в дальнейшем членам команды лучше взаимодействовать друг с другом, а так же экономить свое время и энергию. Помимо этого был создан репозиторий, с которым может работать любой член команды, вносить какие-то изменение в любое время (загружать файла, оставлять комментарии).В общем и целом мы побольше мы изучили спектр действий, которые можно выполнять на GitHub. В дальнейшем данные навыки нам помогут.

