Министерство образования Российской Федерации  
Пензенский Государственный Университет  
Кафедра САПР

**ОТЧЁТ**о выполнении учебной (эксплуатационной) практики  
на тему: «Основы работы с системой контроля версий Git на примере разработки игры на Unity»

Выполнил студент группы 20вв4:

Горбунов Н.А.

Принял: Тимонин А.Ю.

Пенза 2022 г.

ОГЛАВЛЕНИЕ

[Оглавление 2](#_Toc76062557)

[Аннотация 3](#_Toc76062558)

[Работа с ситемой контроля версий 4](#_Toc76062559)

[Разработка игры 11](#_Toc76062566)

[Вывод 14](#_Toc76062567)

АННОТАЦИЯ

В данной пояснительной записке приведено описание работы с системой контроля версий git на примере разработки веб-сайта. Для работы с системой контроля версий использовался Git Bash. В данной записке продемонстрированы и описаны основные команды Git. Каждый этап проиллюстрирован скриншотами.

РАБОТА С СИСТЕМОЙ КОНТРОЛЯ ВЕРСИЙ

Создание репозитория

Git репозитория инициализируется в папке с проектом:

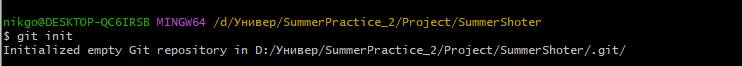


Рисунок — Инициализация репозитория

Настраивается пользователь Git на локальном компьютере: вводится имя пользователя и адрес электронной почты:

Команда git status позволяет определить в каком состояние какие файлы находятся. В данном случае найдены новые файлы, но они ещё не отслеживаются.

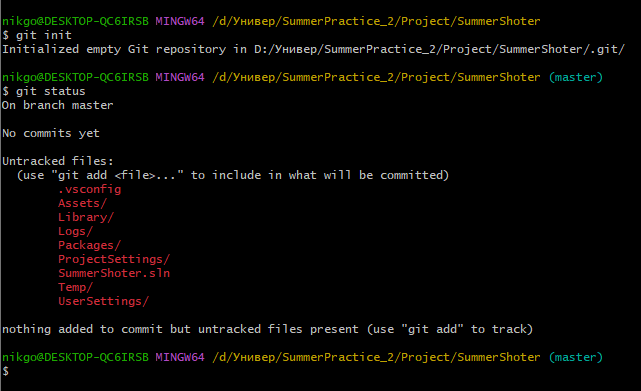
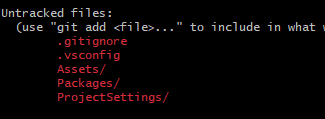
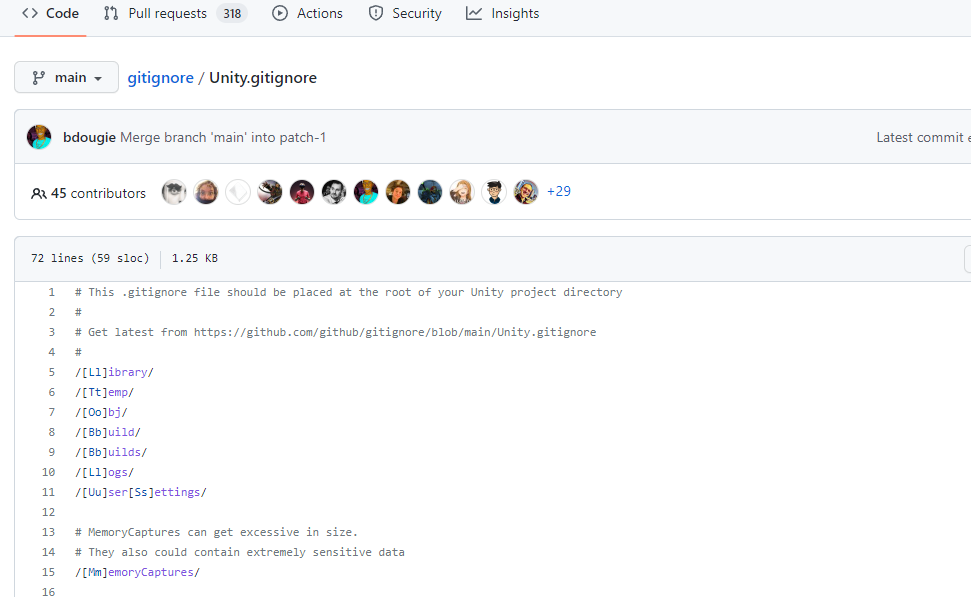


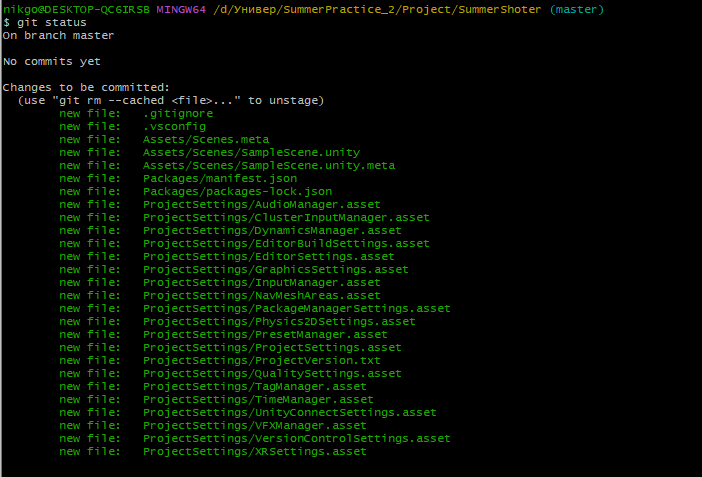
Рисунок 2 — Команда git status

Для отслеживания файлов и последующего сохранении в репозитории (commit) используется команда git add.



Рисунок 3 — Команда git add

Но прежде чем добавлять файлы проекта UNITY, нужно добавить файлы для игнора с помощью отдельного файла .gitignore. Специальный файл .gitignore для unity можно найти в интернете или добавить на сайте GitHub.  
 



Теперь можно сохранить текущую версию проекта в репозитории (commit). Для этого используется команда git commit. Метка -m позволяет описать сохраняемую версию.

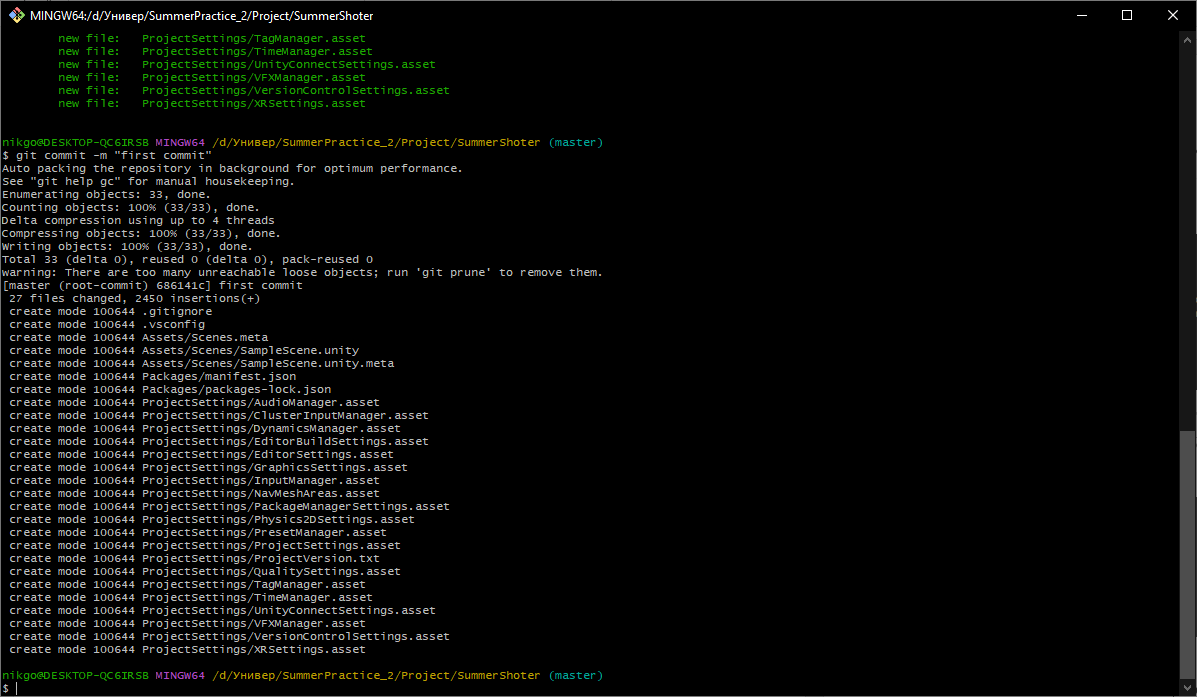
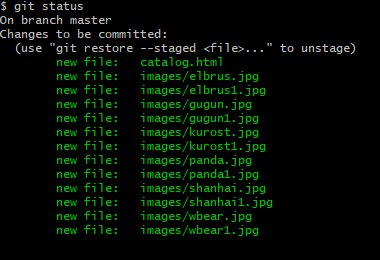
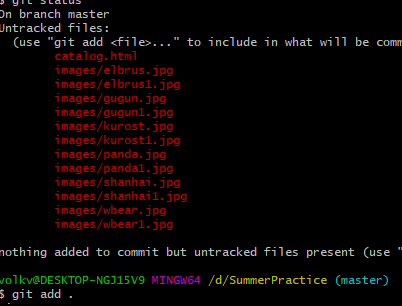


Рисунок 4 — Команда git commit



Удаление файлов

Удаление файлов производится с помощью команды git rm. Удалить можно из индекса (отслеживания), когда командой git add был добавлен лишний файл, или из директории. На примере файл был удалён из директории.

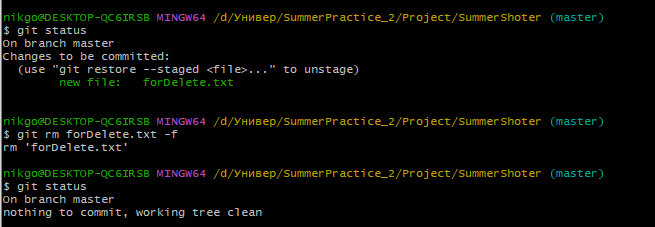


Рисунок 5 —Удаление файла из директории

Перемещение файлов

Для перемещения файлов используется команда git mv. Также эта команда удобна для переименования файлов.

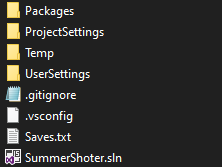


Рисунок 6 — Добавление нового файла Saves.txt

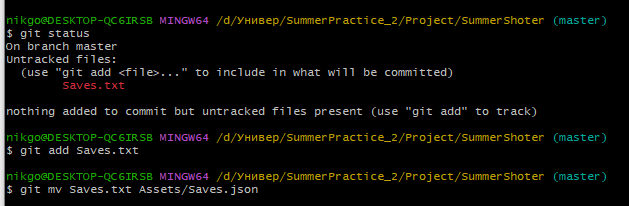


Рисунок 7 — Перемещение и переименование файла

В данном примере файл «Move.txt» одновременно переносится в папку img и меняет название на «move.html».

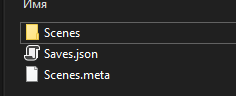


Рисунок 8 — Перемещённый и переименованный файл

Игнорирование файлов

Иногда бывает необходимо не сохранять некоторые файлы в систему контроля версий. Например, временные файлы среды разработки или файлы, которые нужны только локально на компьютере (как в примере сверху). Чтобы игнорировать какие-то файлы, необходимо создать файл «.gitignore», в котором указывается, что должно быть проигнорировано. В процессе разработки проекта был необходим файл .gitignore.

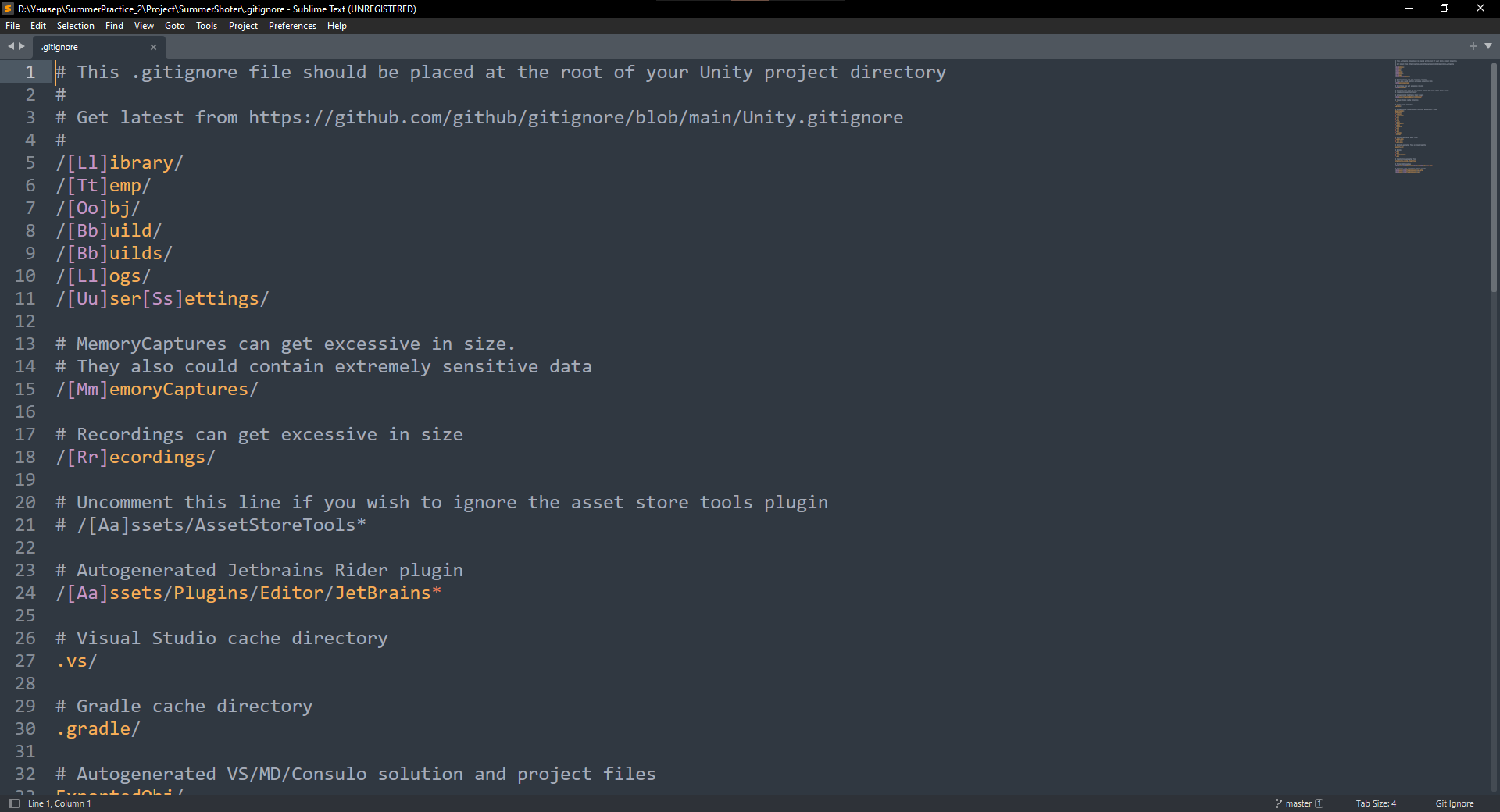
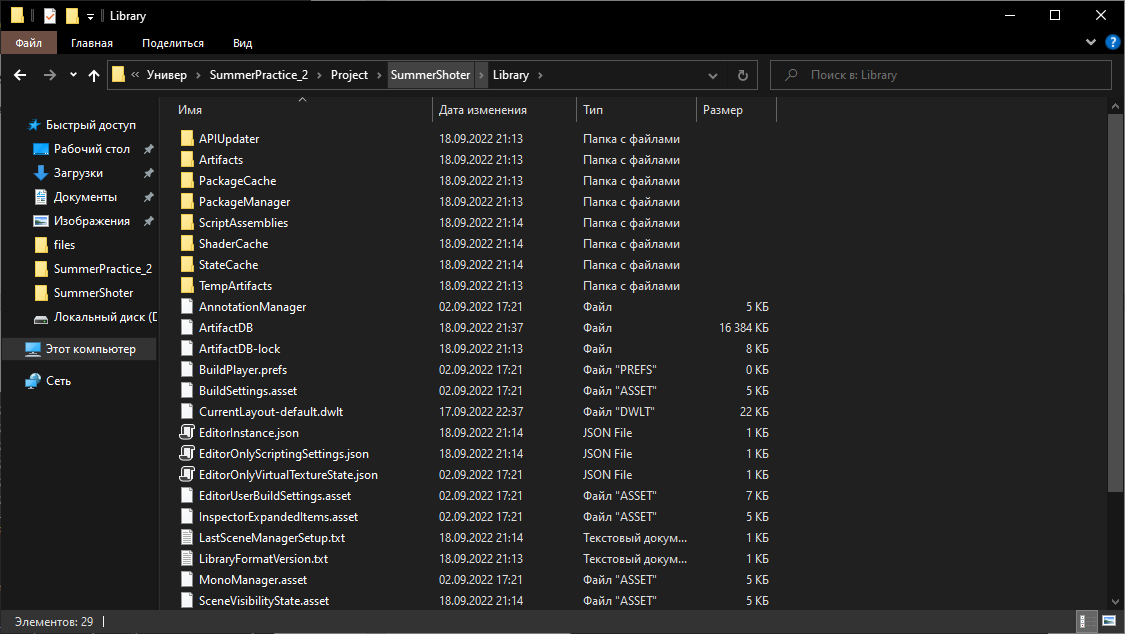


Рисунок 9 — Файл. gitignore и игнорируемые файлы

.



Удалённый репозиторий

Файлы из локального репозитория загружаются на удалённый с помощью команды git push.



Рисунок 15 — Загрузка локального репозитория на удалённый

Загрузка на удалённый репозиторий github.com прошла успешно.

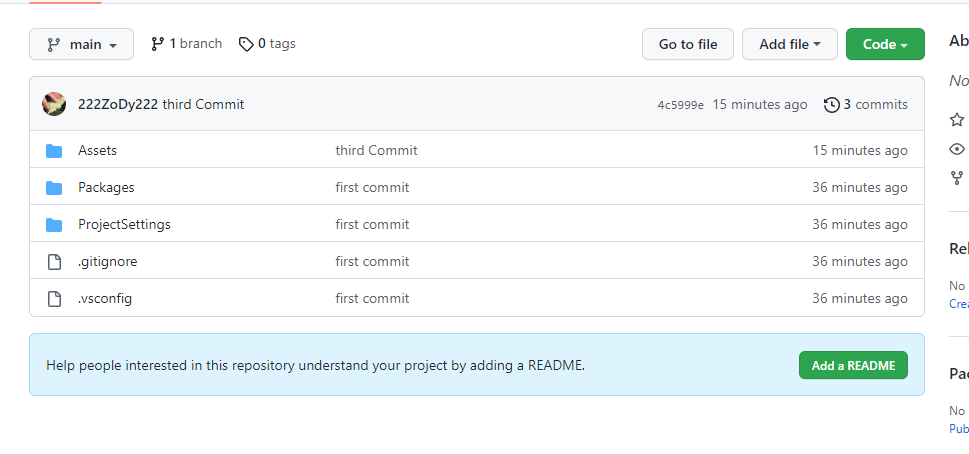


Рисунок 11 — Удалённый репозиторий

Также можно скопировать файлы обратно в папку с проектом или в другое место. Для этого используется команда git clone.

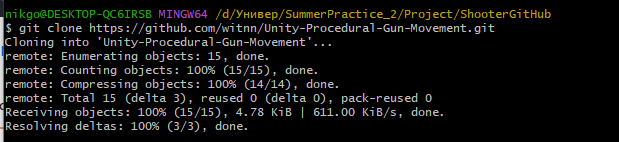


Рисунок 12 — команда git clone

Папка с проектом Unity-Procedural-Gun-Movement из репозитория на github была скопирована на компьютер.

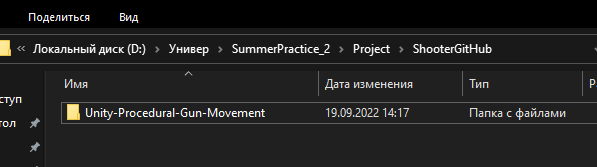


Рисунок 13 — Клонированный репозиторий

Если на удалённом репозитории находится более новая версия проекта, чем на локальном, можно «обновить» проект на локальном. Эта операция представляет собой загрузку файлов и объединение с текущим проектом. Вызывается командой git pull.

Для демонстрации этой функции был создан новый commit, файлы были загружены на удалённый репозиторий, а после на локальном этот commit был отменён.

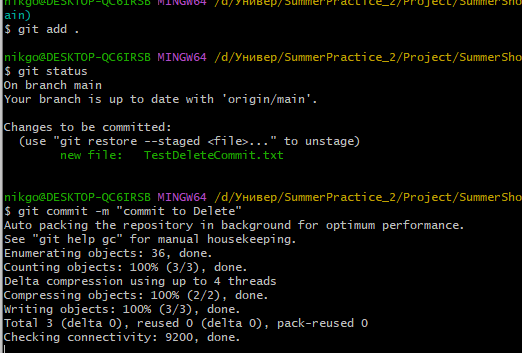


Рисунок 14 — Новый commit

С помощью команды git pull удалось восстановить удалённый commit на локальном репозитории.

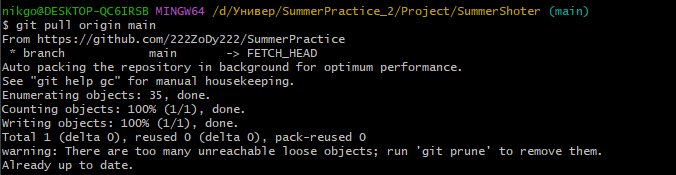


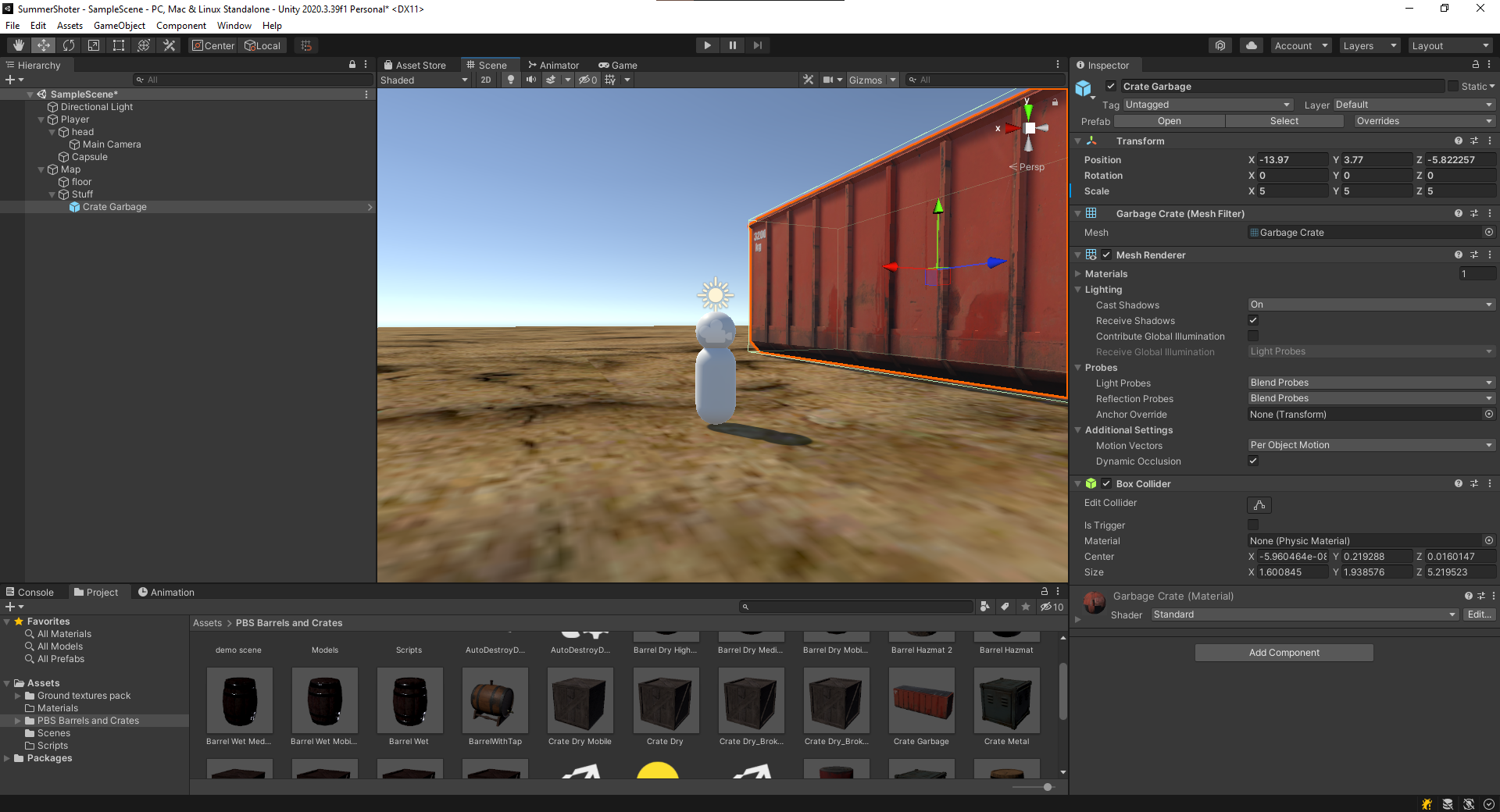
Рисунок 15 — Команда git pull

РАЗРАБОТКА ИГРЫ

Для примера была выбрана игра. В процессе разработки использовались игровой движок Unity, Язык C#.

Игра разрабатывалась в несколько этапов, каждый из которых был занесён в систему контроля версий git.

Первым делом создали карту, добавили материалы и главного героя.



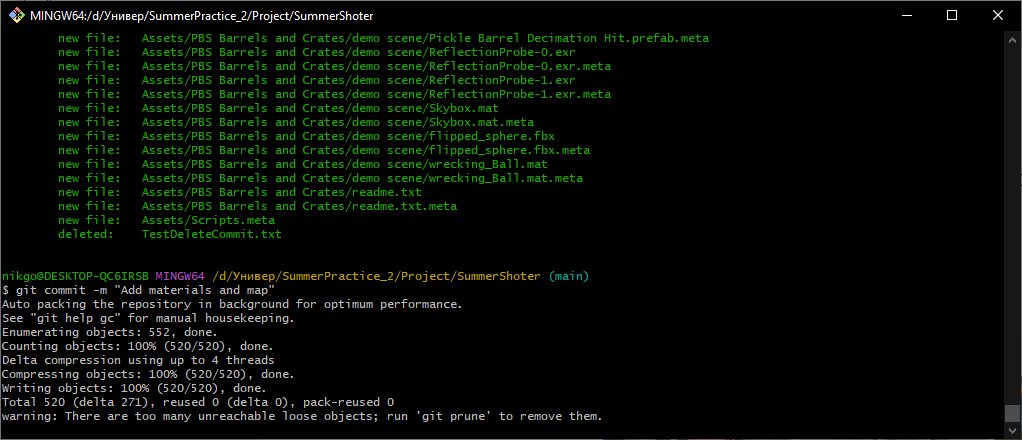
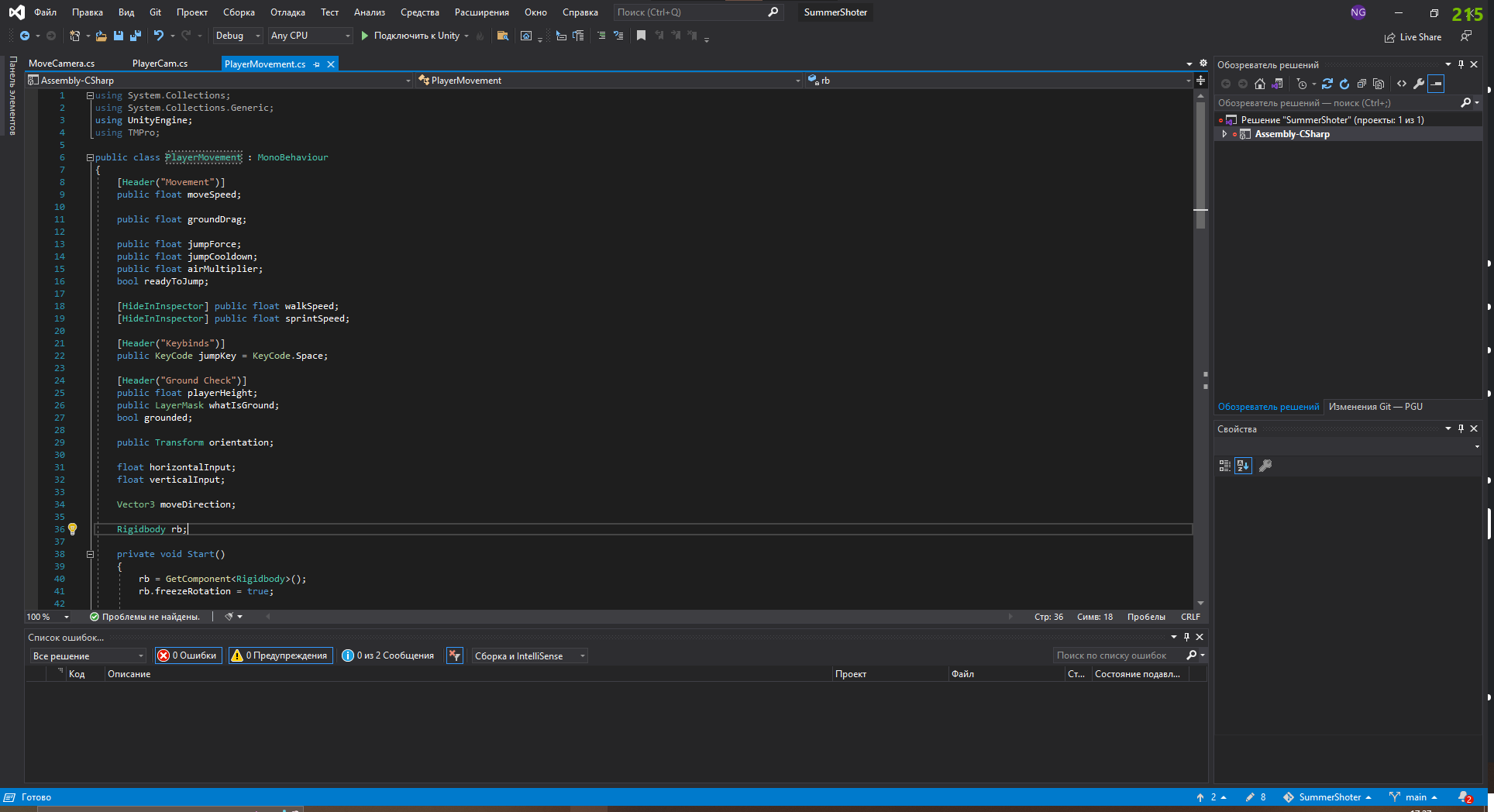
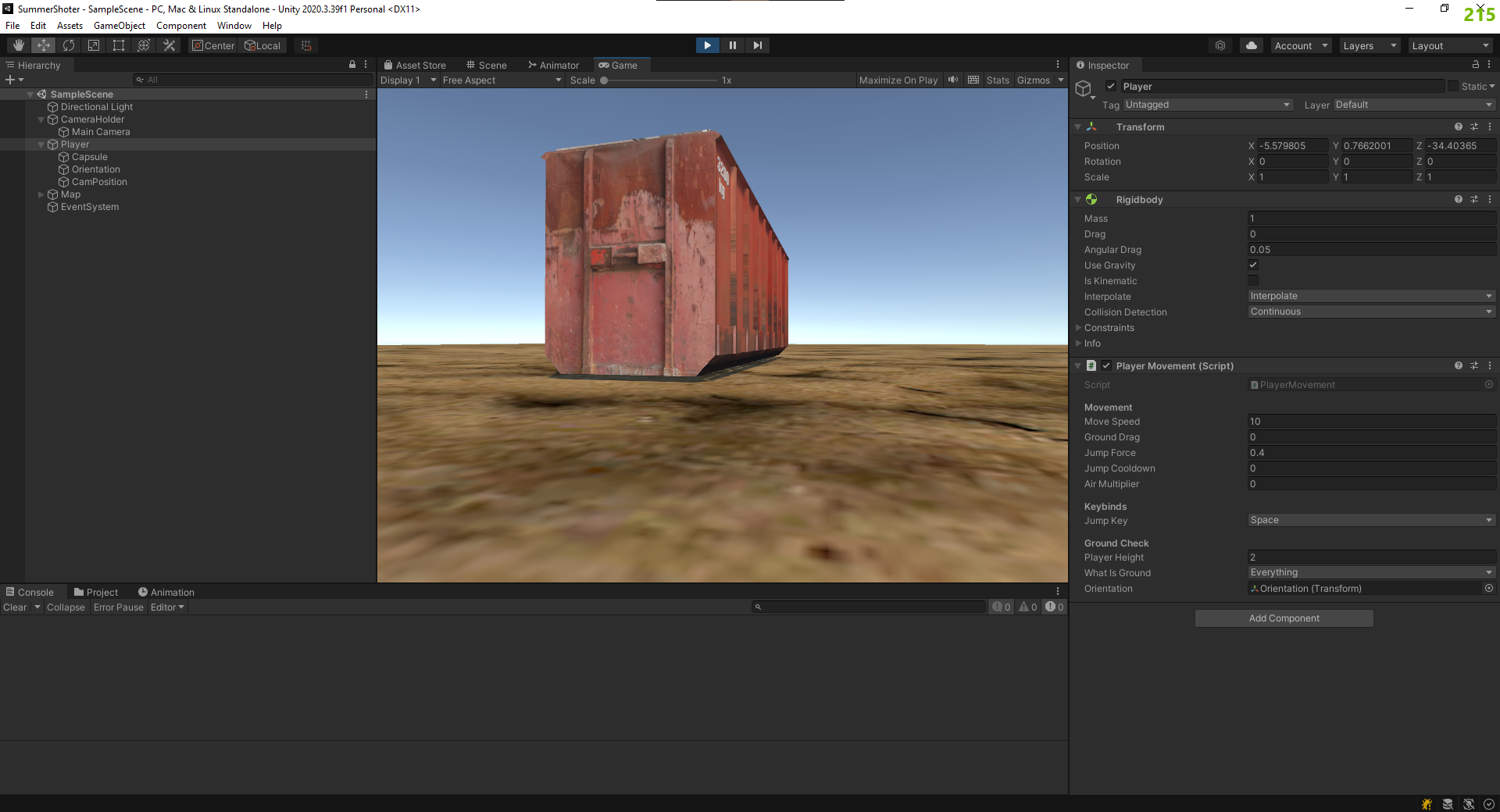


Рисунок 16 — Создание карты и добавление материалов

Далее был разработан скрипт для передвижения персонажа и камеры.





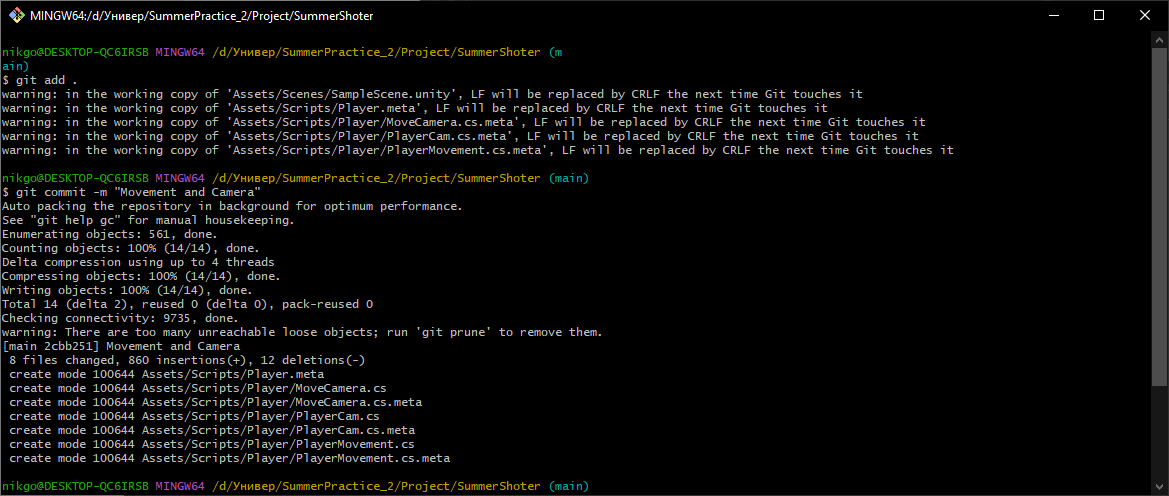
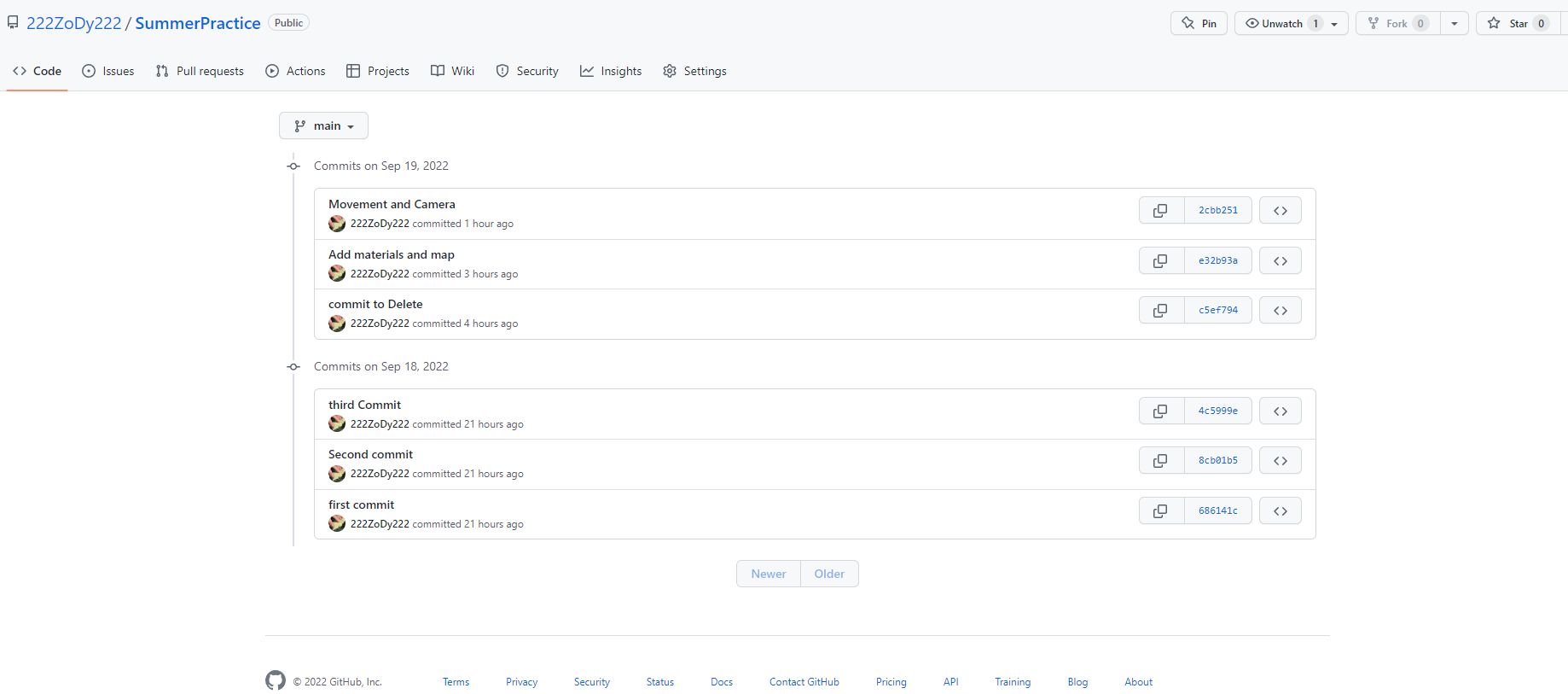


Рисунок 17 – Создаём коммит с добавленным движением игрока

Рисунок 18 — история commit`ов

**Вывод**

В процессе выполнения данной практической работы была разработана игра, этапы создания которой были зафиксированы в системе контроля версий git.