МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Донецкий национальный технический университет»

Факультет ИСП

Кафедра ПИ им Л.П.Фельдмана

Лабораторная работа № 2

на тему: «Работа с ветками»

по курсу: «Профессиональная практика программной инженерии»

Проверил:

асс. каф. ПИ им. Л.П.Фельдмана Филипишин Д.А.

Выполнил:

ст. гр. ПИ-20а

Стеблий Д.Г.

Донецк-2024

Цель работы – познакомиться с основами использования веток в системе контроля версий Git.

Вариант 20. Видеоигру шутер, с реализацией механик, не менее 30-ти.

Аккаунт: danny31311, ссылка:

<https://github.com/danny31311>

1. Распределим модули по веткам и укажем модули, принадлежащие к каждой ветке:

1. Стрельба и механики персонажа – shut\_and\_mech\_pers:
   1. "Персонаж" – pers.py;
   2. "Инициализация персонажа" – pers.py;
   3. "Обновление персонажа" – pers.py;
   4. "Движение" – pers.py;
   5. "Инициализация движения" – pers.py;
   6. "Обновление движения" – pers.py;
   7. "Движение без возможностей остановиться" – pers.py;
   8. "Игровая камера" – pers.py;
   9. "Инициализация игровой камеры" – pers.py;
   10. "Обновление игровой камеры" – pers.py;
   11. "Курсор" – shoot.py;
   12. "Инициализация курсора" – shoot.py;
   13. "Обновление курсора" – shoot.py;
   14. "Стрельба" – shoot.py;
   15. "Инициализация стрельбы" – shoot.py;
   16. "Изменение стрельбы" – shoot.py;
   17. "Показывание вектора стрельбы" – shoot.py;
2. Начало игры и захват - start\_game\_and\_capture:
3. "Запуск игры" - start\_game.py;
4. "Захват территорий" - capture\_teritory.py;
5. "Выбор бонуса за захват" - capture\_teritory.py;
6. "Начало захвата территорий" - capture\_teritory.py;
7. "Начало запуска игры" - start\_game.py;
8. Уникальные механики - unicle\_mechanics:
9. "Кража внешнего вида других существ игры" - change\_model.py;
10. "Начало кражи" - change\_model.py;
11. "Смена модели персонажа" - change\_model.py;
12. "Разрушение мира и окружения" - destroying\_world.py;
13. "Начало разрушения мира и окружения" - destroying\_world.py;
14. "Изменение мира" - destroying\_world.py;
15. "Система точек сохранения" - save\_point.py;
16. "Начало системы точек сохранения" - save\_point.py;
17. "Сохранения точек сохранения" - save\_point.py;
18. "Изменение сохранённых точек сохранения" - save\_point.py.

2. Создадим ветки на сайте GitHub (рис. 1-3).

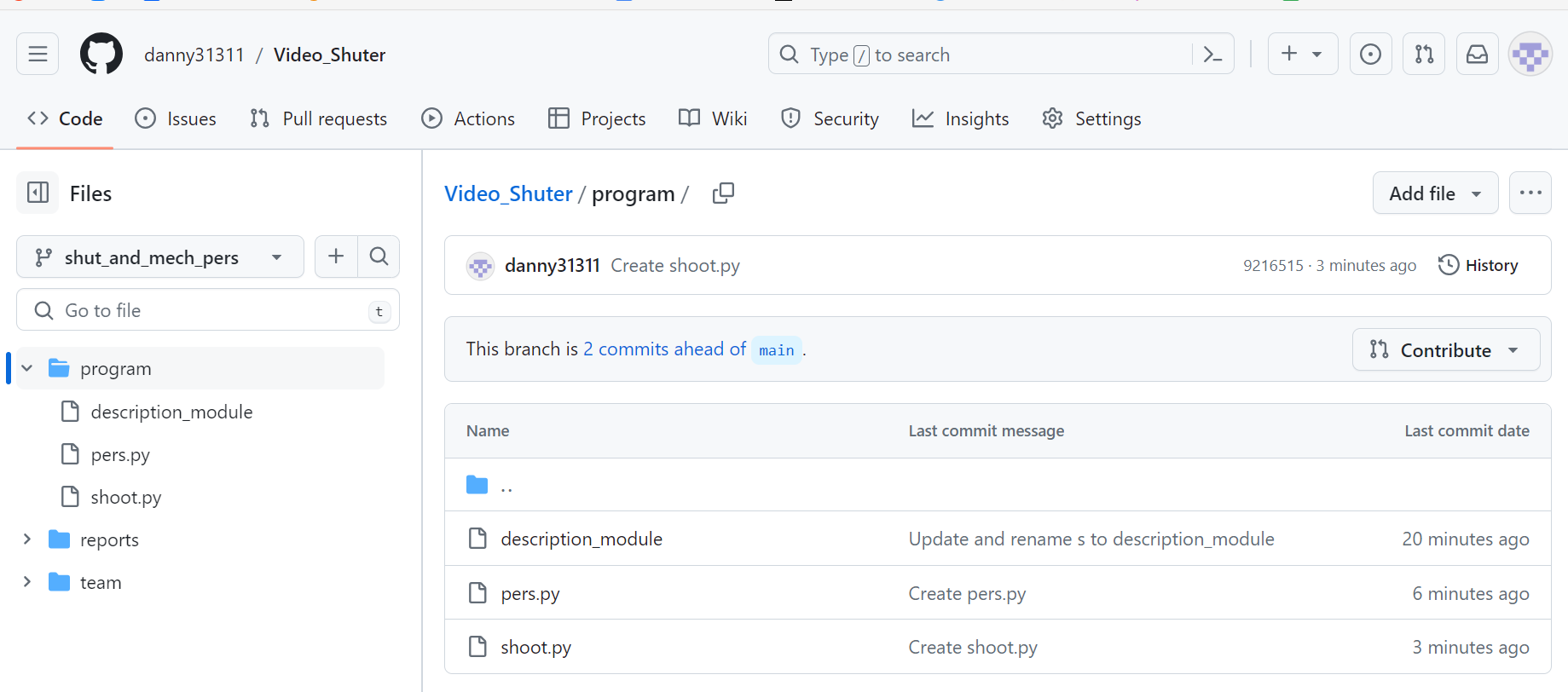


Рисунок 1 – Ветка «Стрельба и механики персонажа»

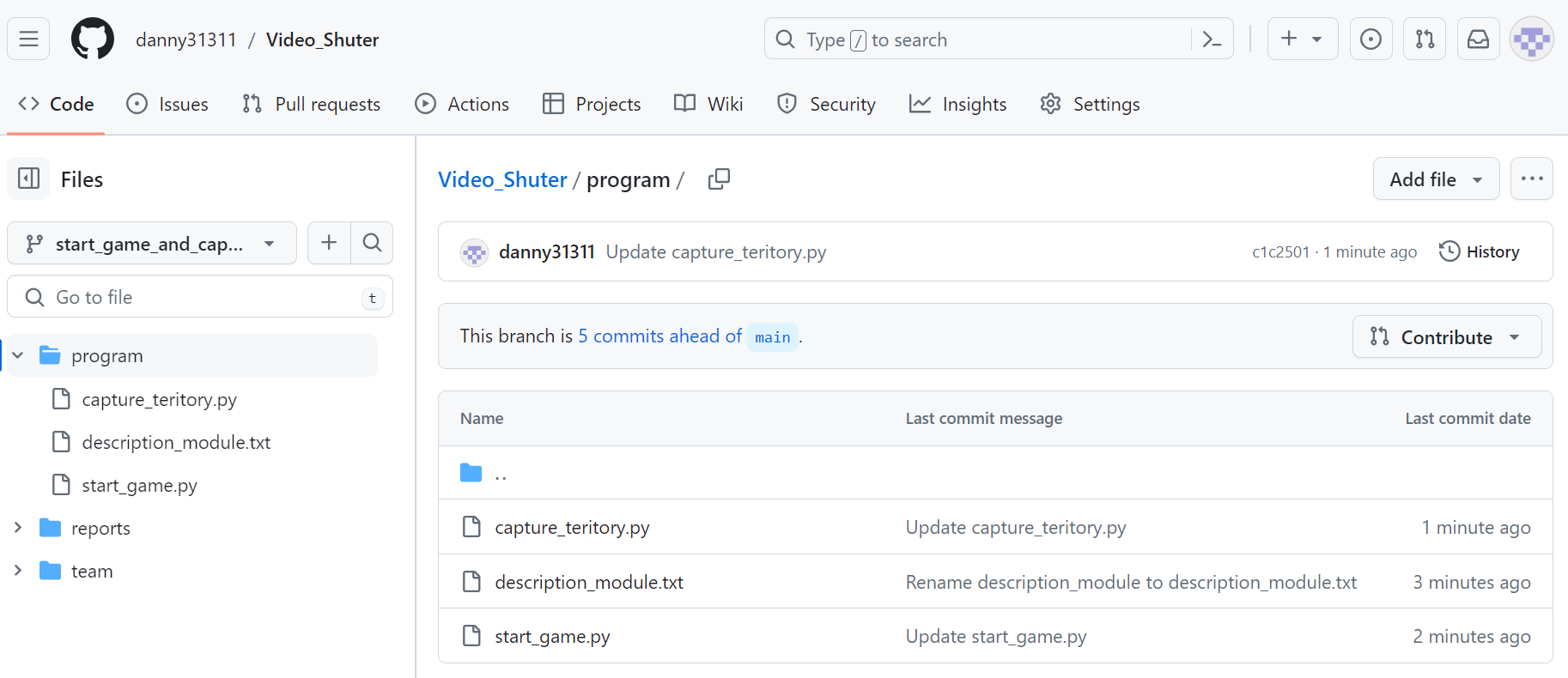


Рисунок 2 – Ветка «Начало игры и захват»

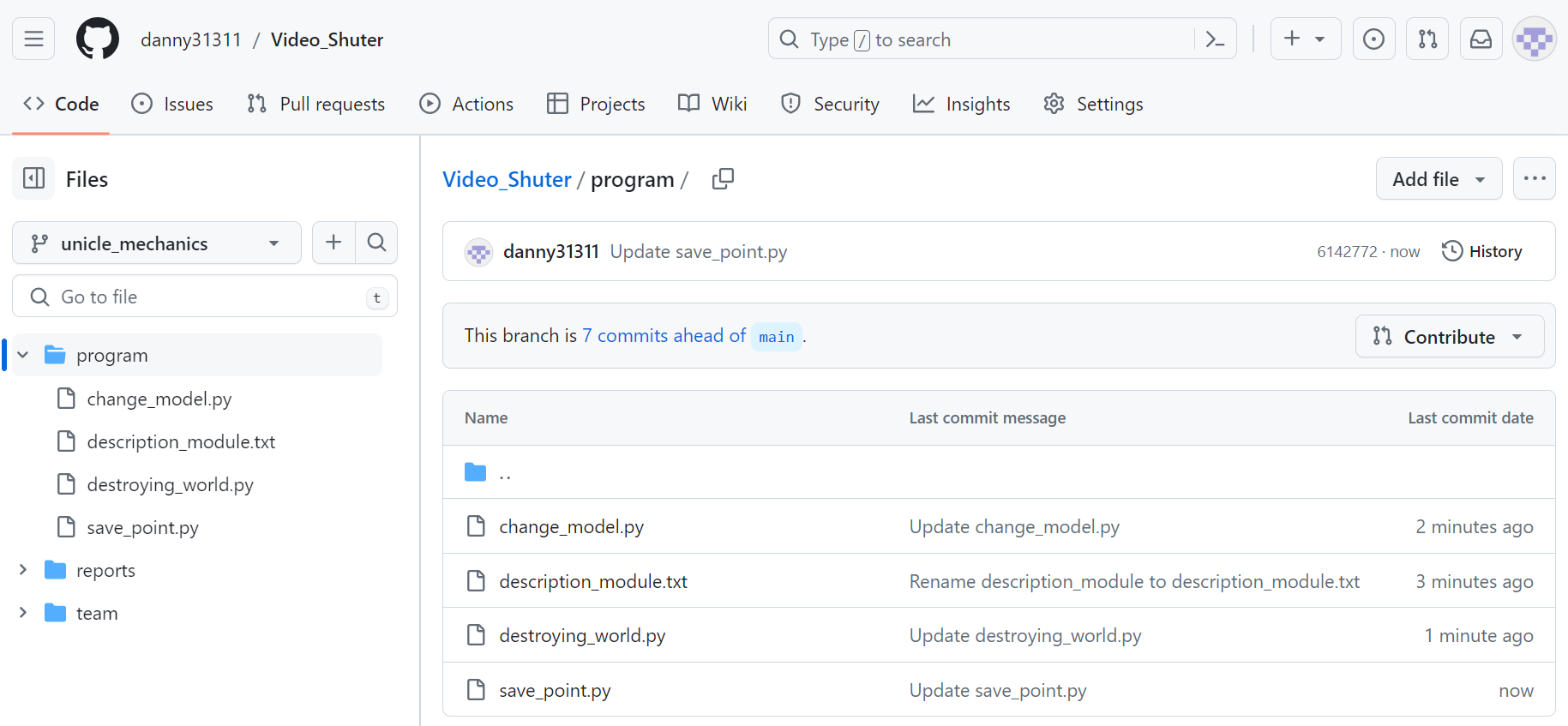


Рисунок 3 – Ветка «Уникальные механики»

На рисунках 4-10 представлены файлы в ветках.

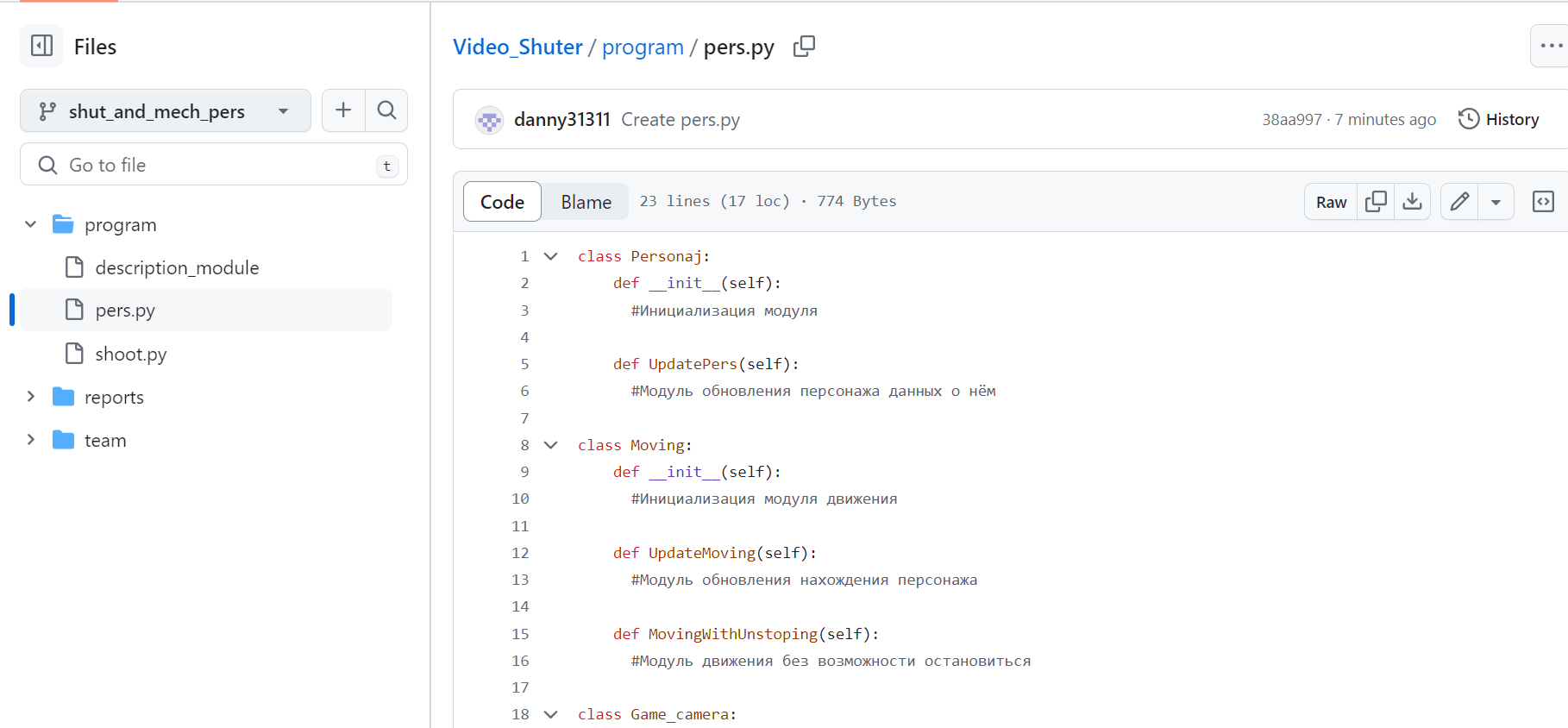


Рисунок 4 – Ветка «Стрельба и механики персонажа», файл pers.py

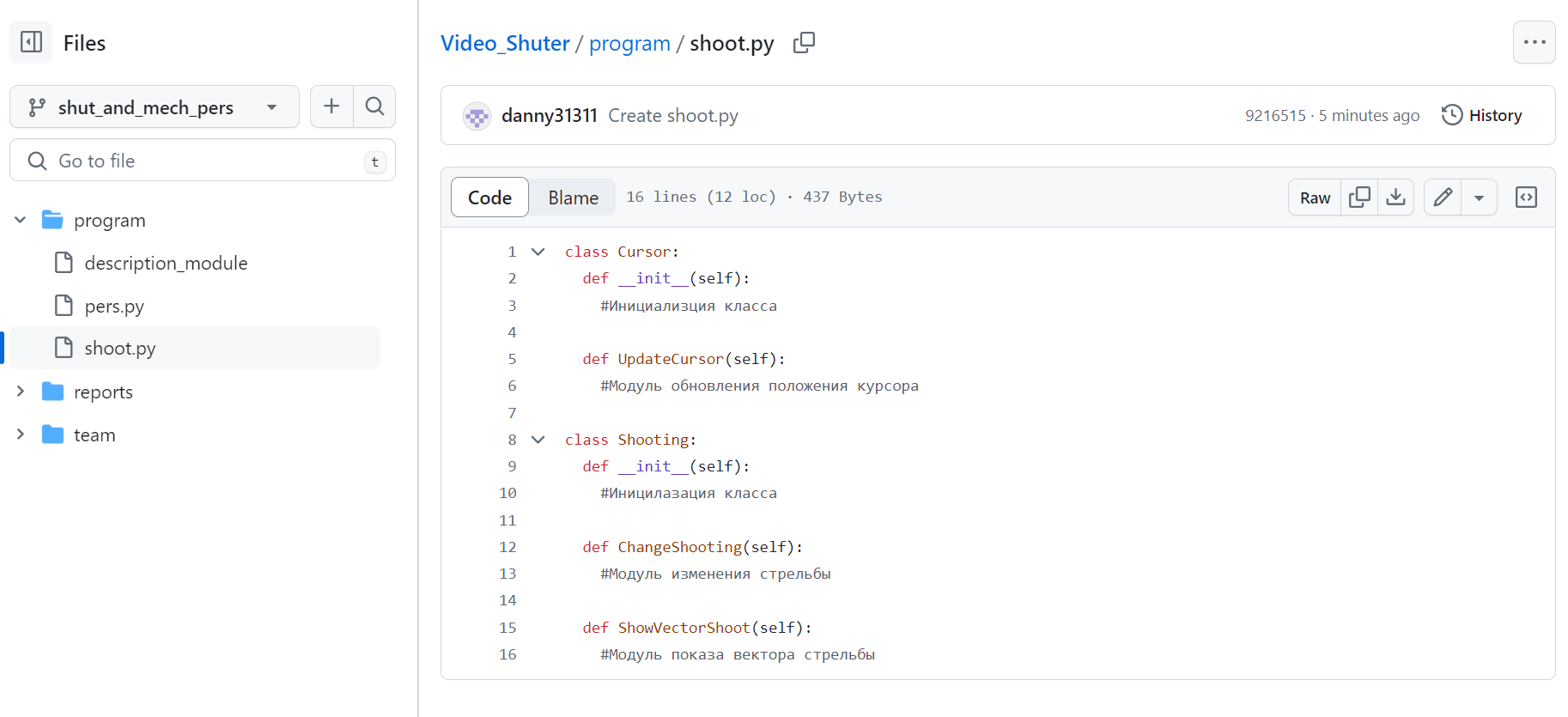


Рисунок 5 – Ветка «Стрельба и механики персонажа», файл shoot.py

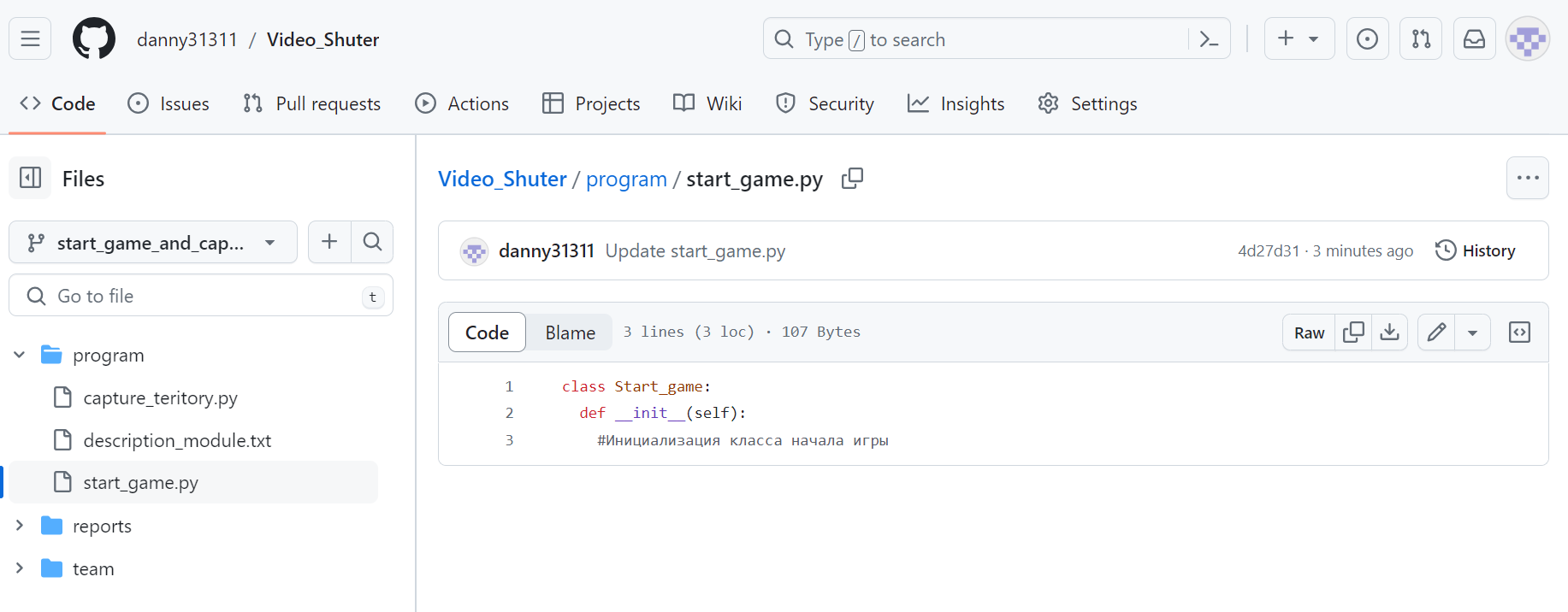


Рисунок 6 – Ветка «Начало игры и захват», файл start\_game.py

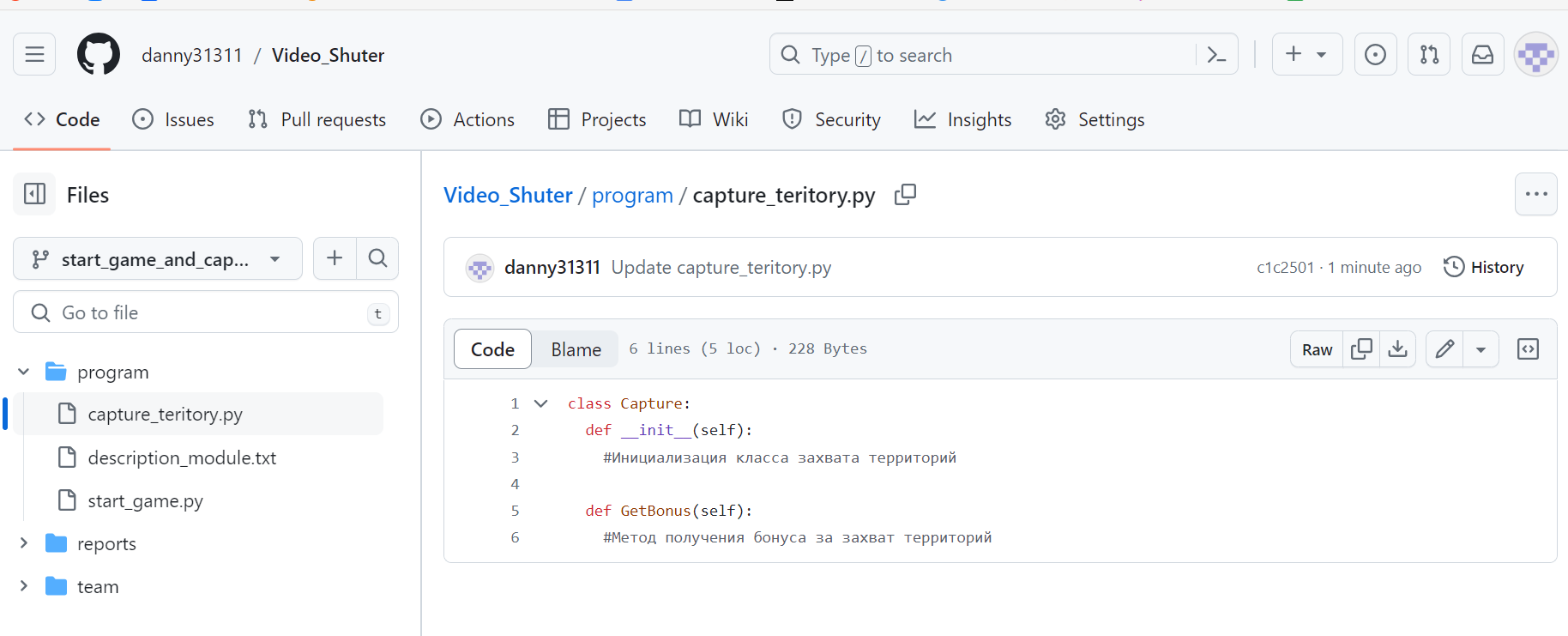


Рисунок 7 – Ветка «Начало игры и захват», файл capture\_teritory.py

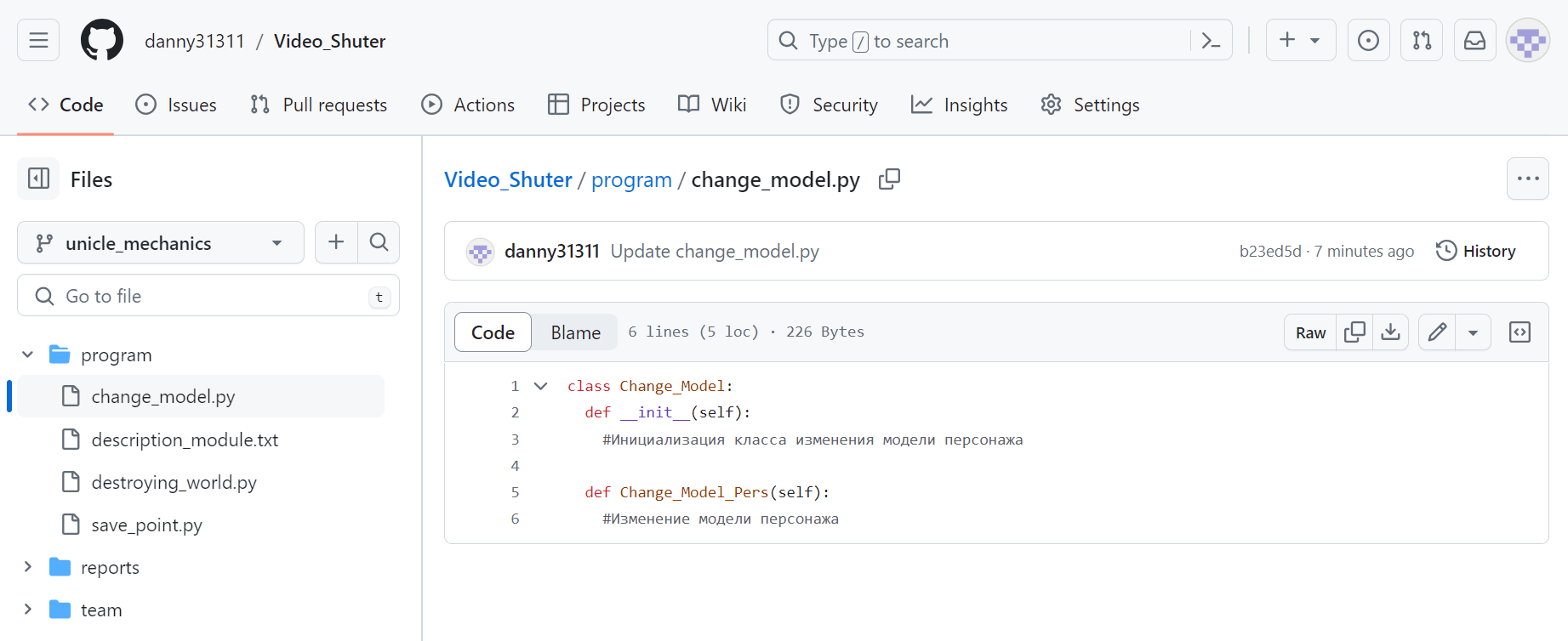


Рисунок 8 – Ветка «Уникальные механики», файл change\_model.py

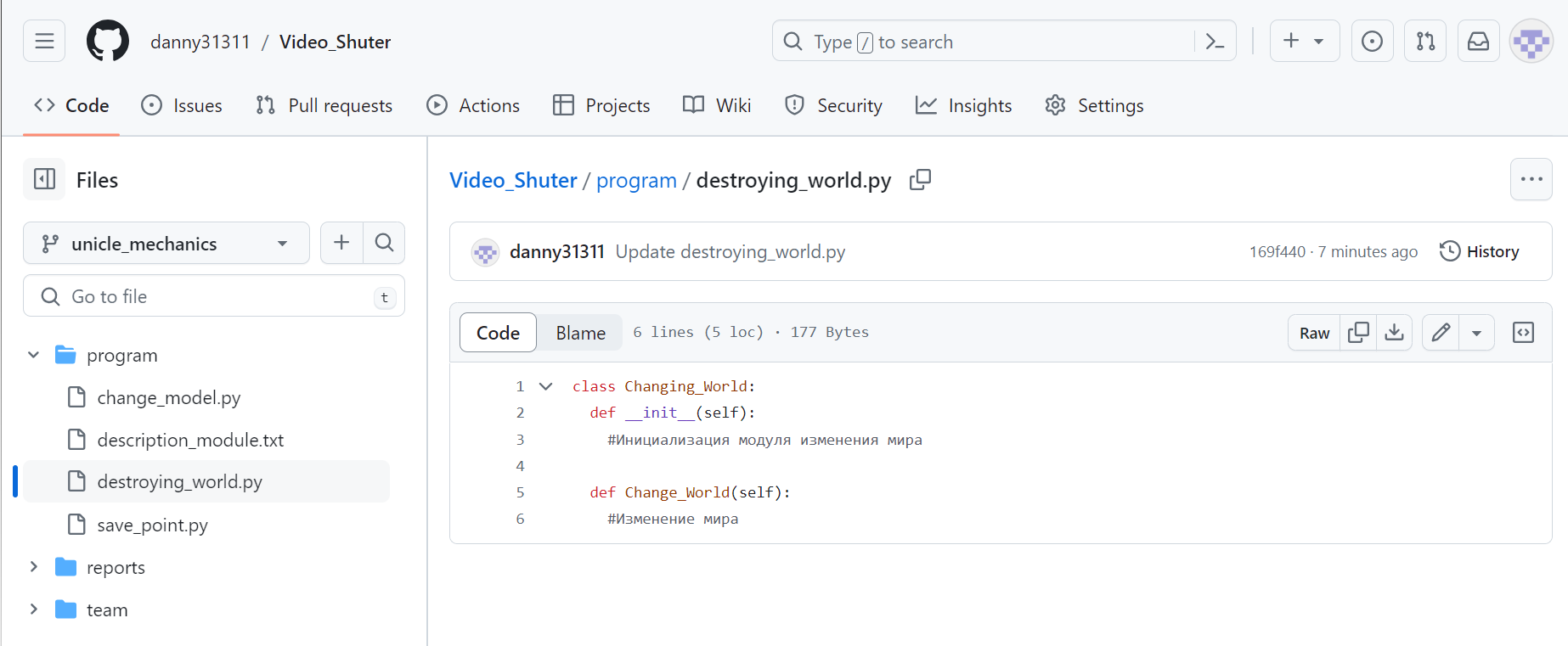


Рисунок 9 – Ветка «Уникальные механики», файл destroying\_world.py

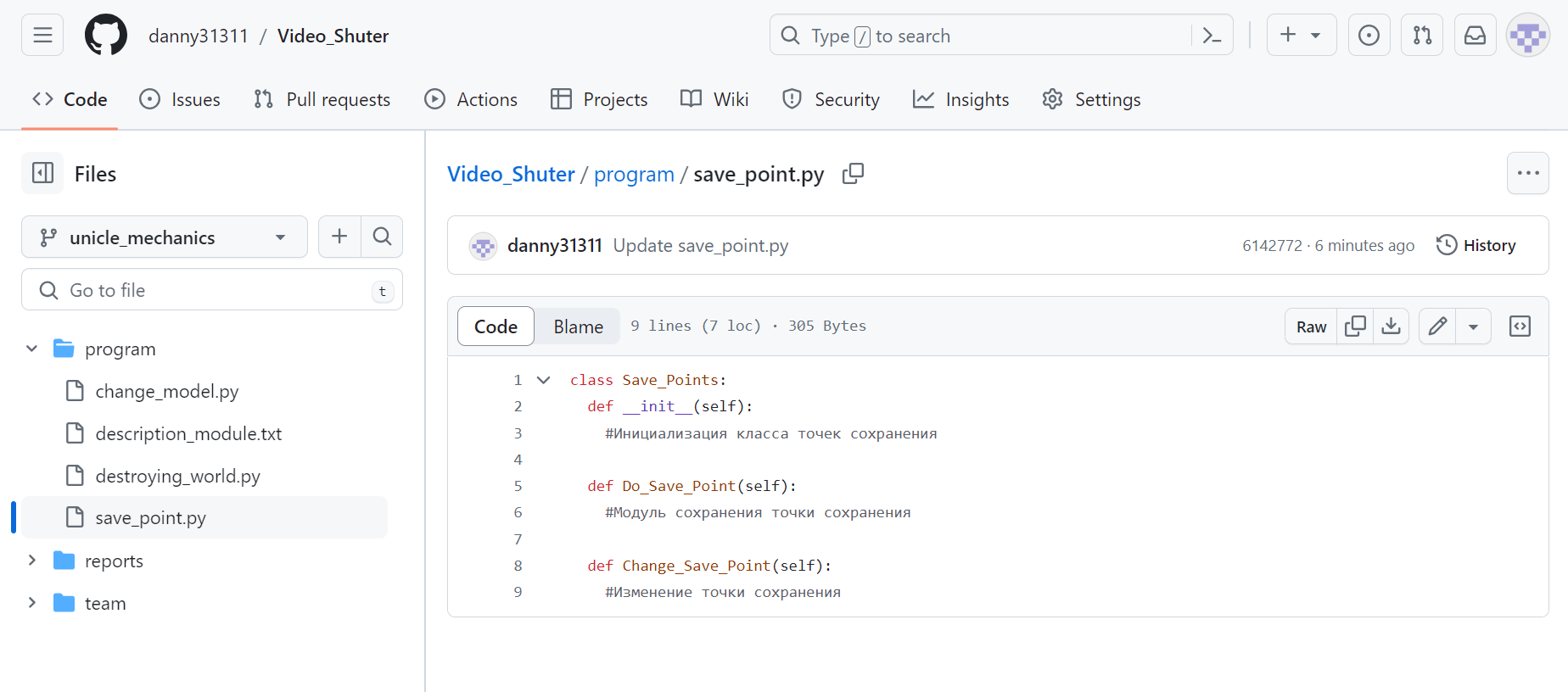


Рисунок 10 – Ветка «Уникальные механики», файл save\_point.py

На рисунках 11-13 представлены истории коммитов для каждой ветки.

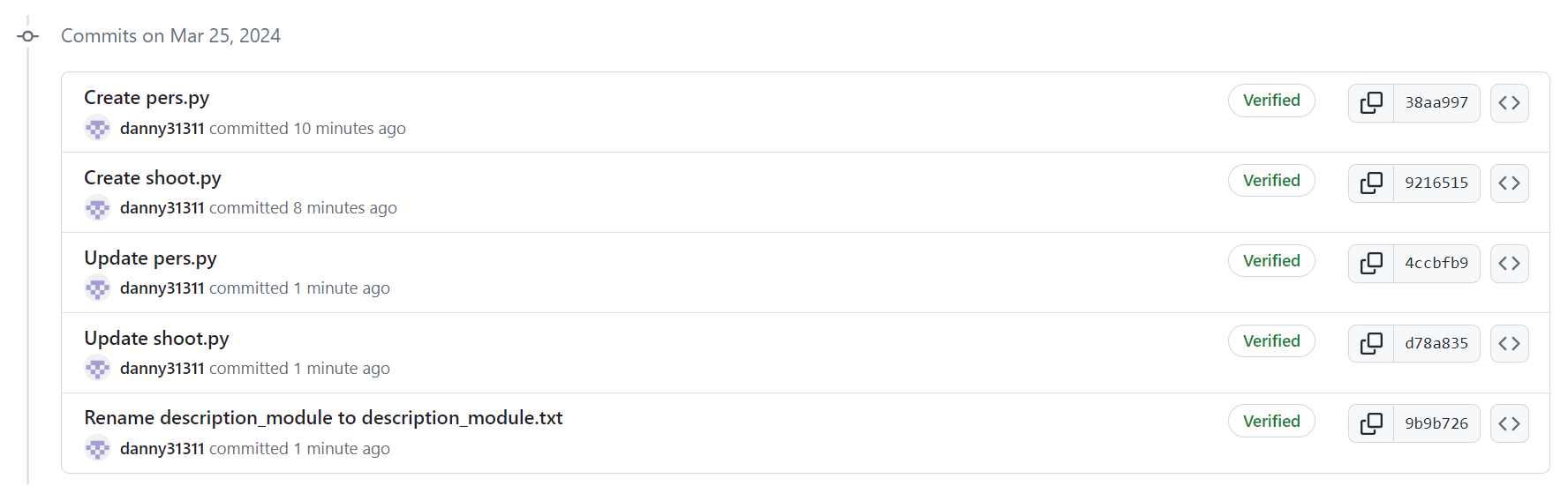


Рисунок 11 – История коммитов ветки «Стрельба и механики персонажа»

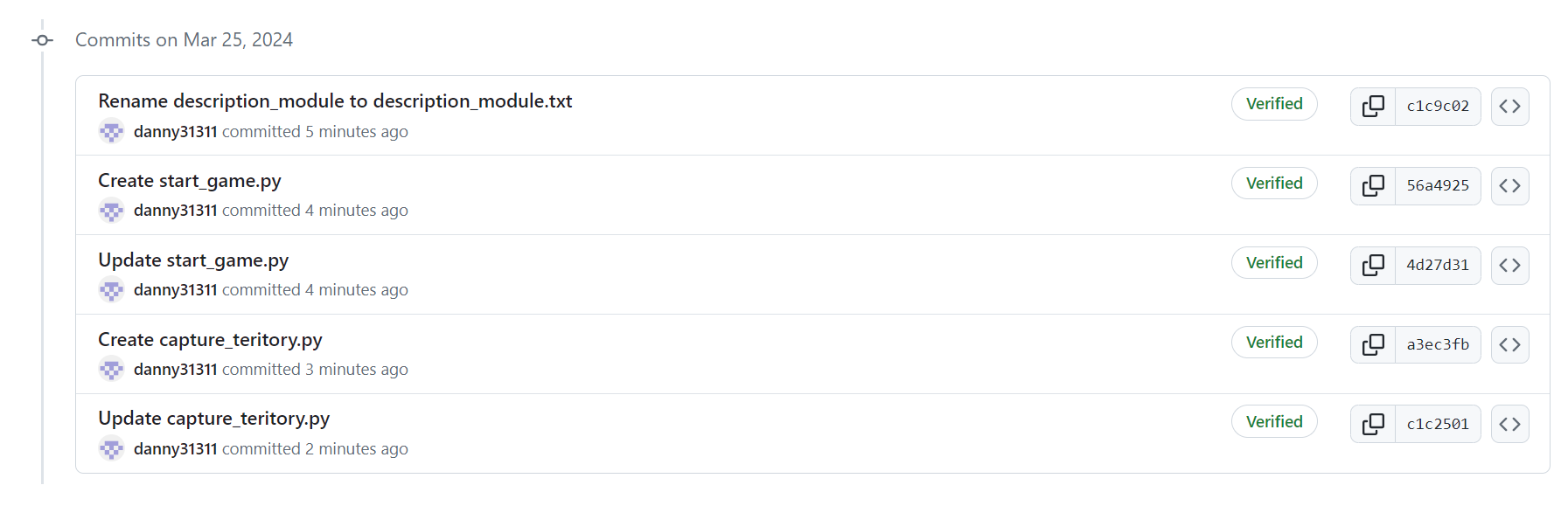


Рисунок 12 – История коммитов ветки «Начало игры и захват»

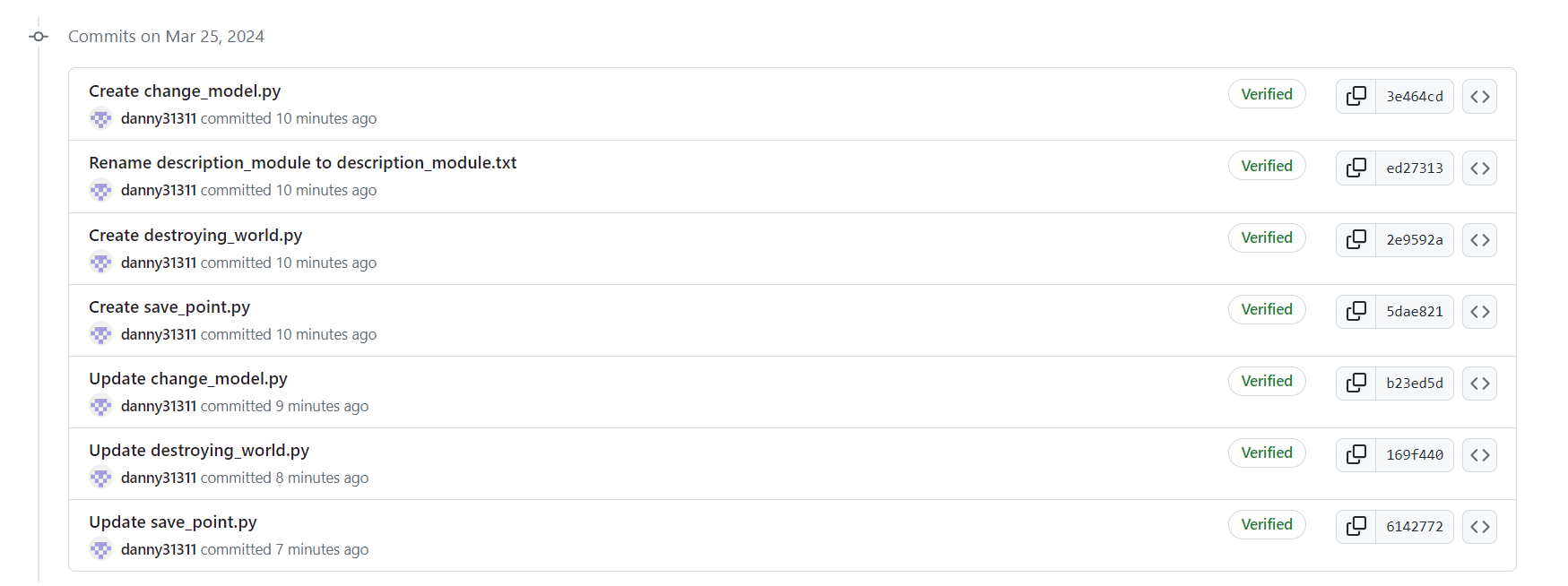


Рисунок 13 – История коммитов ветки «Уникальные механики»

Далее произведём merge ветки main с веткой shut\_and\_mech\_pers. Создадим pull request (рис. 14).

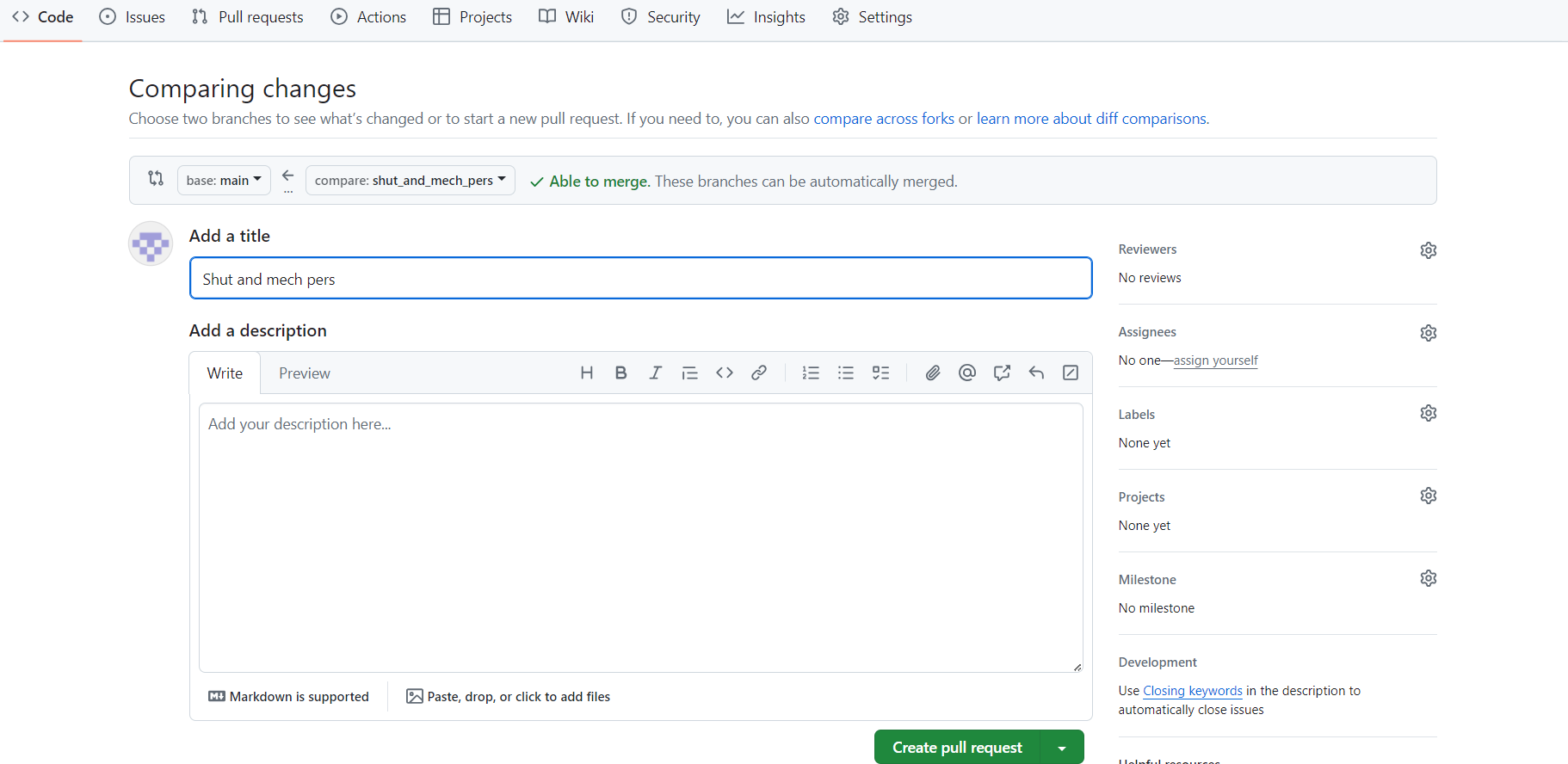


Рисунок 14 – Создание pull request

После нажатия на кнопку «Create pull request» на экране появляется информация об отсутствии конфликтах между ветками – это означает, что merge(слияние) возможно (рис. 15).

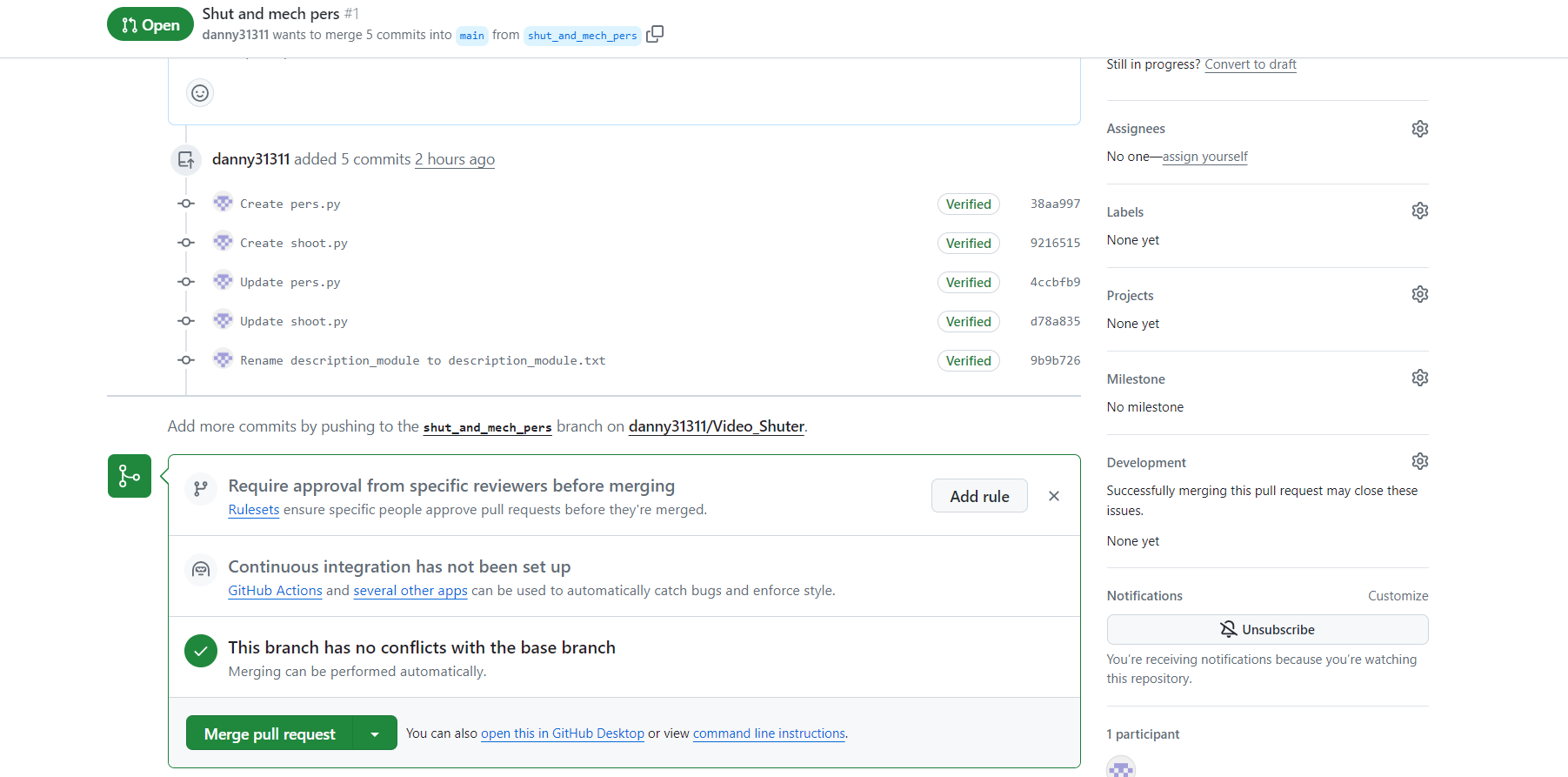


Рисунок 15 – Уведомление о возможности merge

После нажатия на кнопку «Merge pull request» и подтверждения действия, merge будет успешно выполнен (рис. 16).

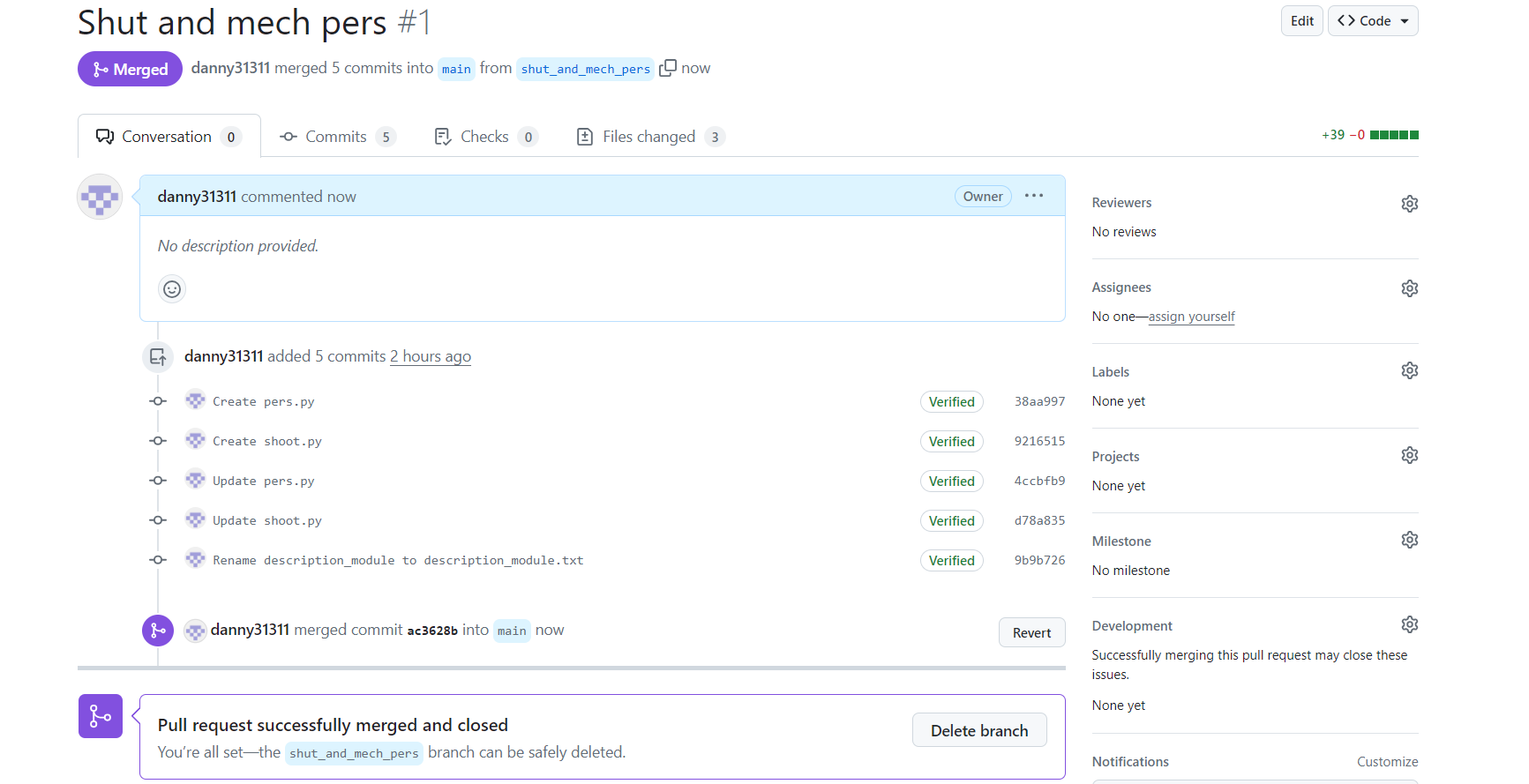


Рисунок 16 – Успешное выполнение merge

Создадим конфликтную ситуацию – в ветку start\_game\_and\_capture добавим файл pers.py (этот файл ранее был в ветке shut\_and\_mech\_pers, которую включили в ветку main).

Попробуем произвести merge ветки main с веткой start\_game\_and\_capture. Уже при создании pull request можно заметить сообщение **«Can’t automatically merge»** (рис. 17).

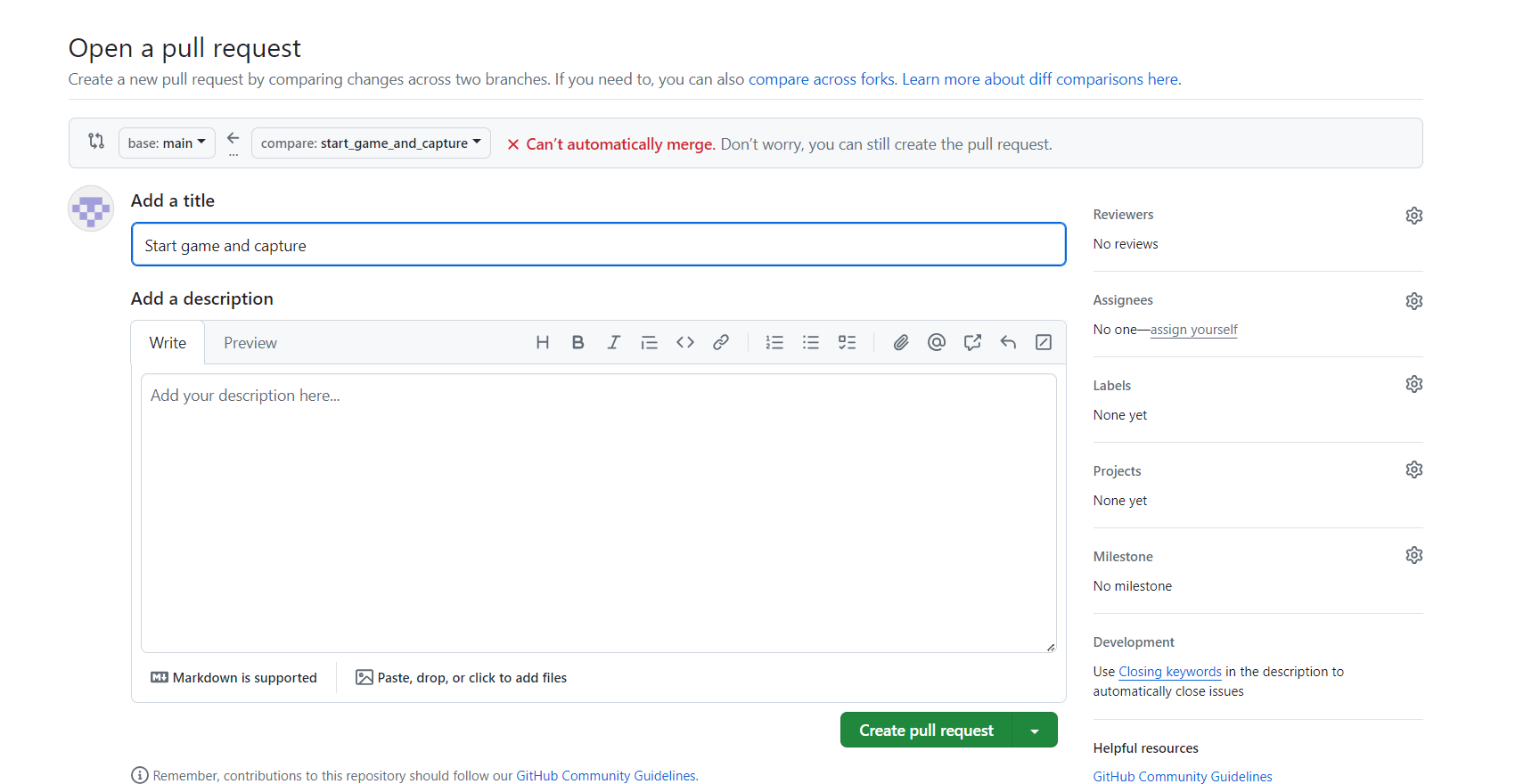


Рисунок 17 – Создание pull request с несовпадающими данными

После нажатия на кнопку «Create pull request» на экране появляется сообщение о наличие конфликта в ветке, для успешного merge этот конфликт должен быть решён (рис. 18).

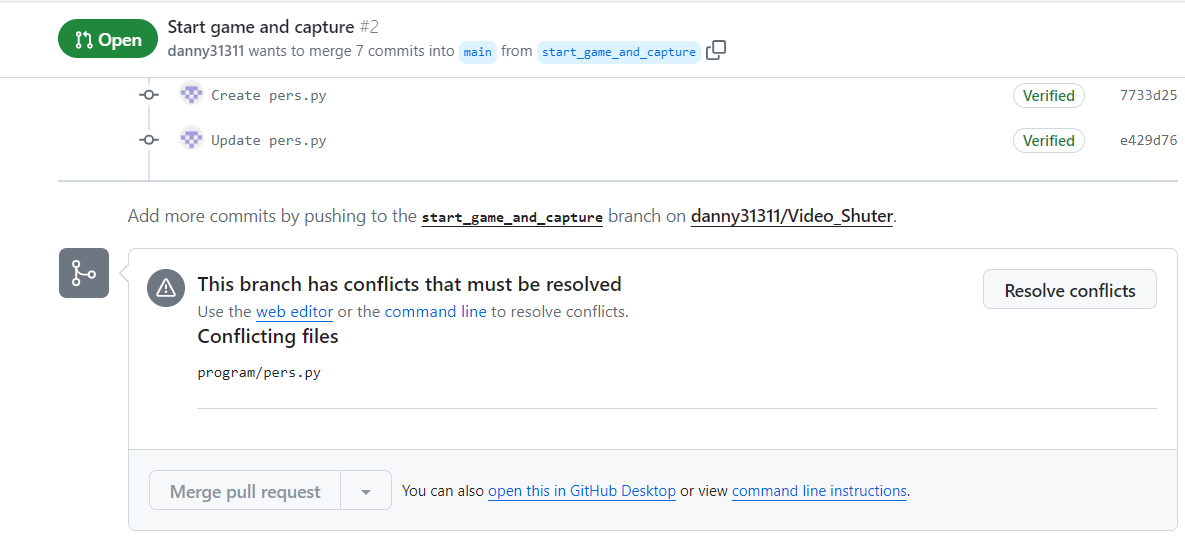


Рисунок 18 – Сообщение о наличии конфликта

Решение конфликта ручными способом предполагает выбор нужной части кода (рис. 19).

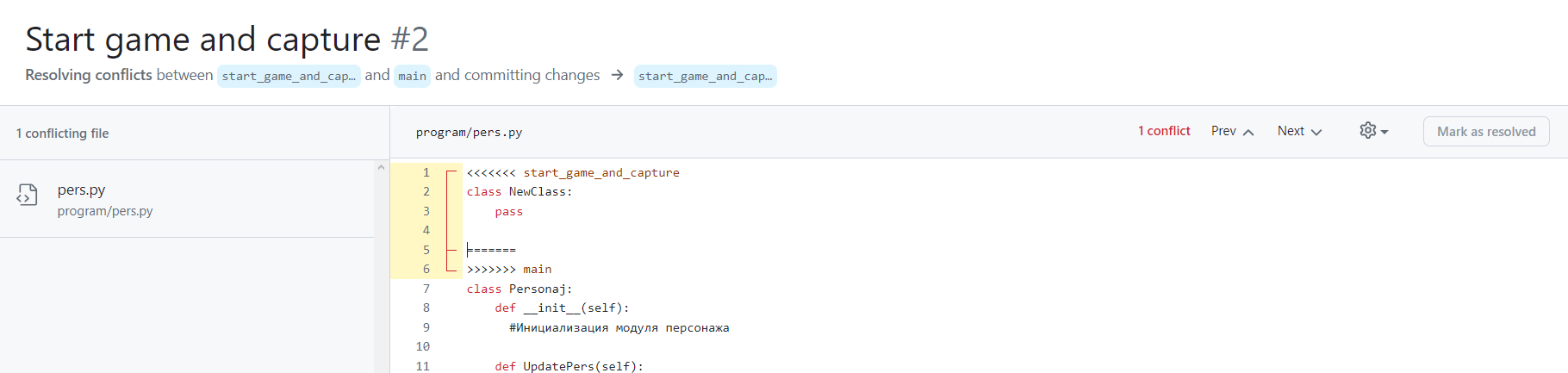


Рисунок 19 – Конфликт в коде

В данном случае, выберем вторую часть кода и нажмём на кнопку «Mark as resolved» (рис. 20).

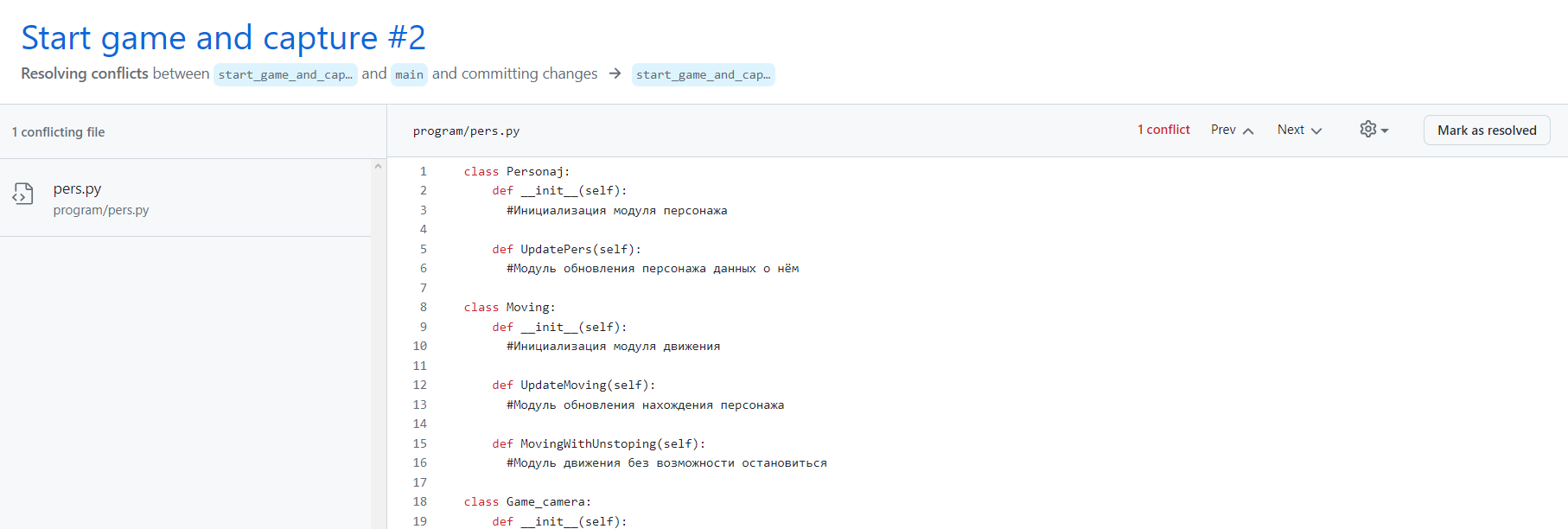


Рисунок 20 – Решение конфликта

Попробуем выполнить новый commit и получим успешный результат (рис. 21).

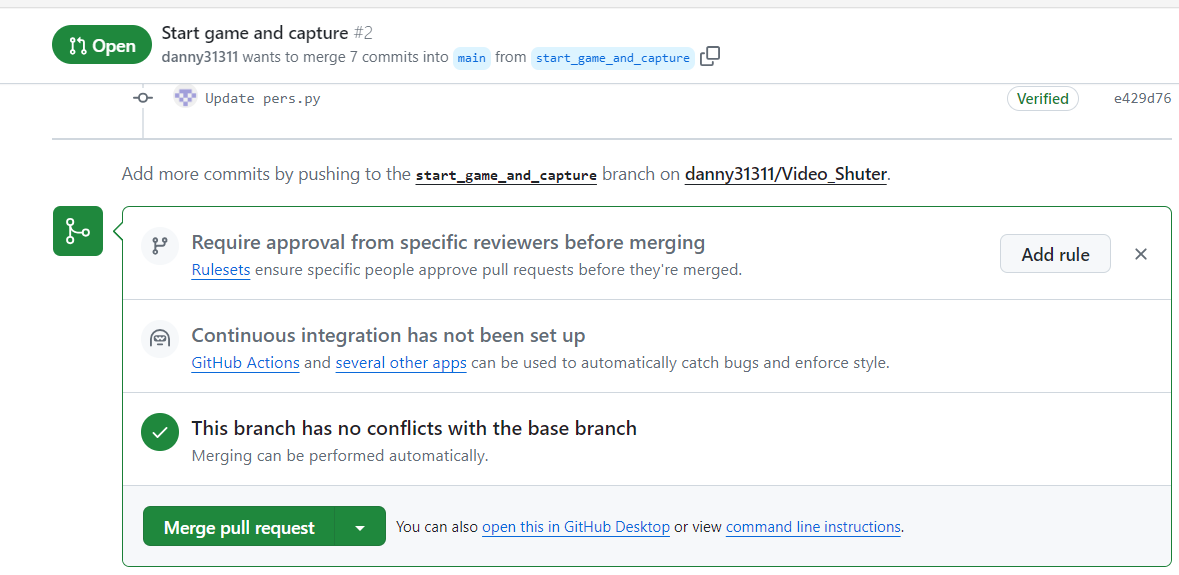


Рисунок 21 – Успешное решение конфликта

По тому же принципу объединим ветку unicle\_mechanics с main и получим результат – все ветки были объединены в main (рис. 22).

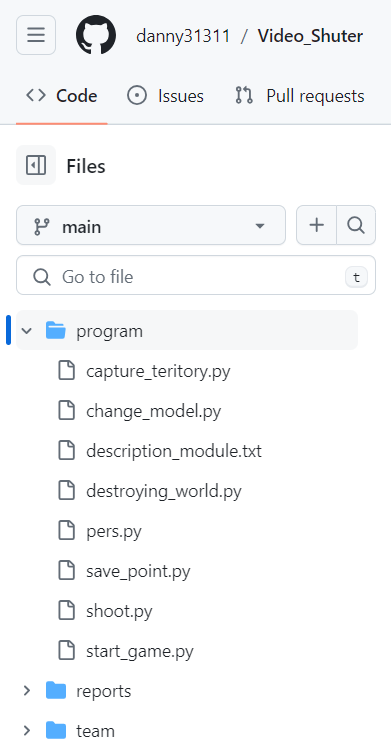


Рисунок 22 – Результат объединения всех веток