МИНИСТРЕРСТВО 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  
по курсу: «Качество программного обеспечения и тестирование»  
на тему: «Работа с ветками»  
Вариант №15

Выполнил:  
ст. гр. ПИ-20а  
Коржов А. В.

Проверил:

Филипишин Д. А.

ДОНЕЦК – 2024

**Тема разработки:** тетрис, с возможностью игры по сети.

**Ссылка на проект:**

https://github.com/AntonDarkness/PPPI\_Block\_Game/tree/main

**Ветка master (основная)**

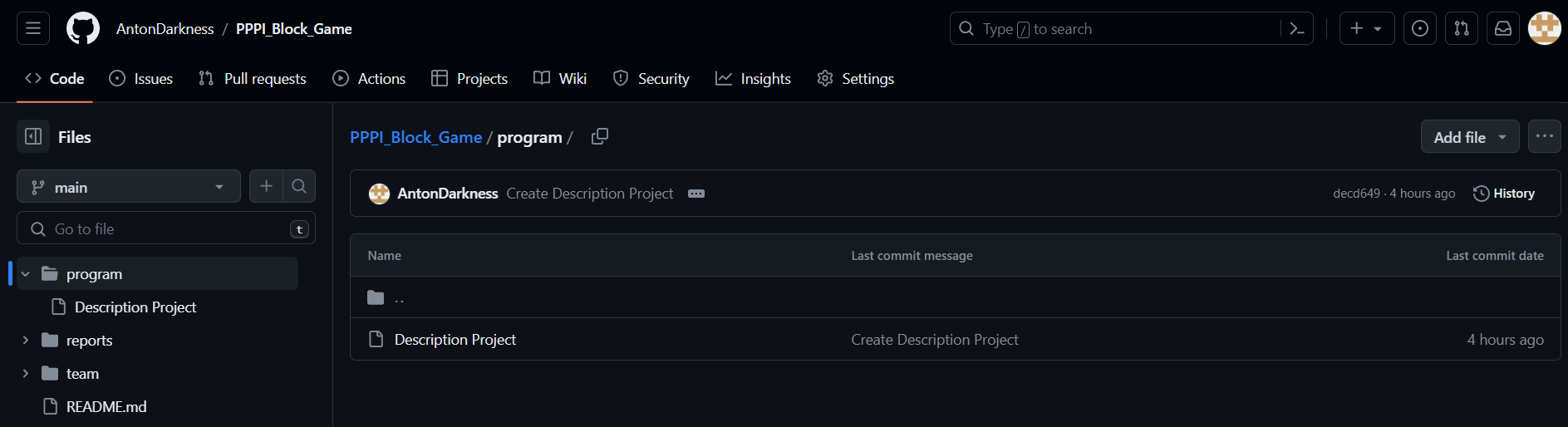


Рисунок 1 — Основная ветка

**Разработка веток**

При рассмотрении проекта было принято решение создать 4 ветки, которые будут носить название основных модулей: сервер, клиент, интерфейс, игровая логика.

Рассмотрим данные ветки:

- server — ветка, в которой будет рассмотрена работа над моделированием серверной составляющей проекта: управление сессиями, синхронизация пользователей, обмен сигналами и др.;

- client — ветка, в которой промоделируем классы, отвечающие за работу клиентского приложения (переключение между окнами, загрузка полей, подключение к серверу и т.д.);

- interface — ветка, в котором будет происходить работа над моделированием классов, отвечающих за интерфейс игры, а также за все элементы и их логику работы;

- game-logic — ветка, в которой рассмотрим классы, которые отвечают за всю игровую логику нашего проекта.

**Структура веток и файлов, лежащих в них**

Примечание: файлы кода будут содержать некоторые методы классов, однако описано всё будет комментариями, полного описания кода не будет приведено в классах, как и методов, которые можно опустить или сложно продумать сразу.

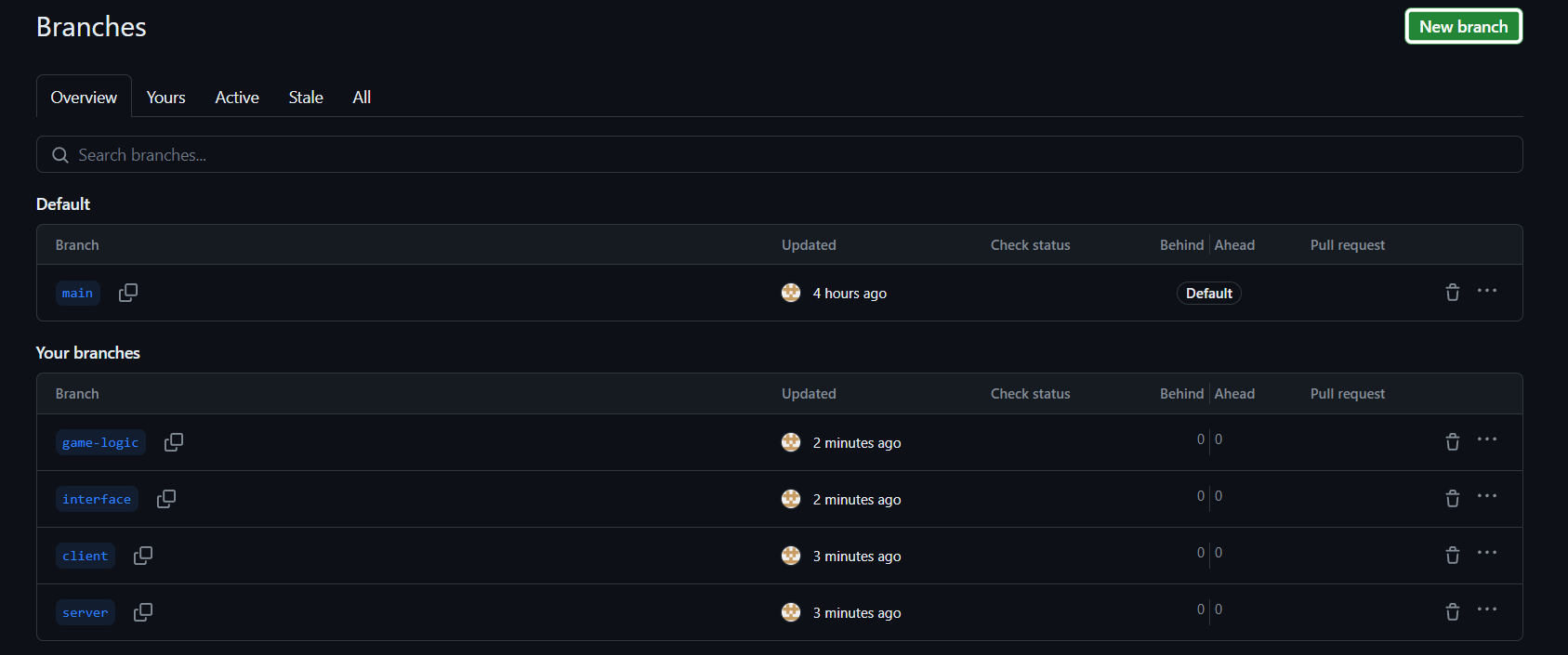


Рисунок 2 — Созданные ветки

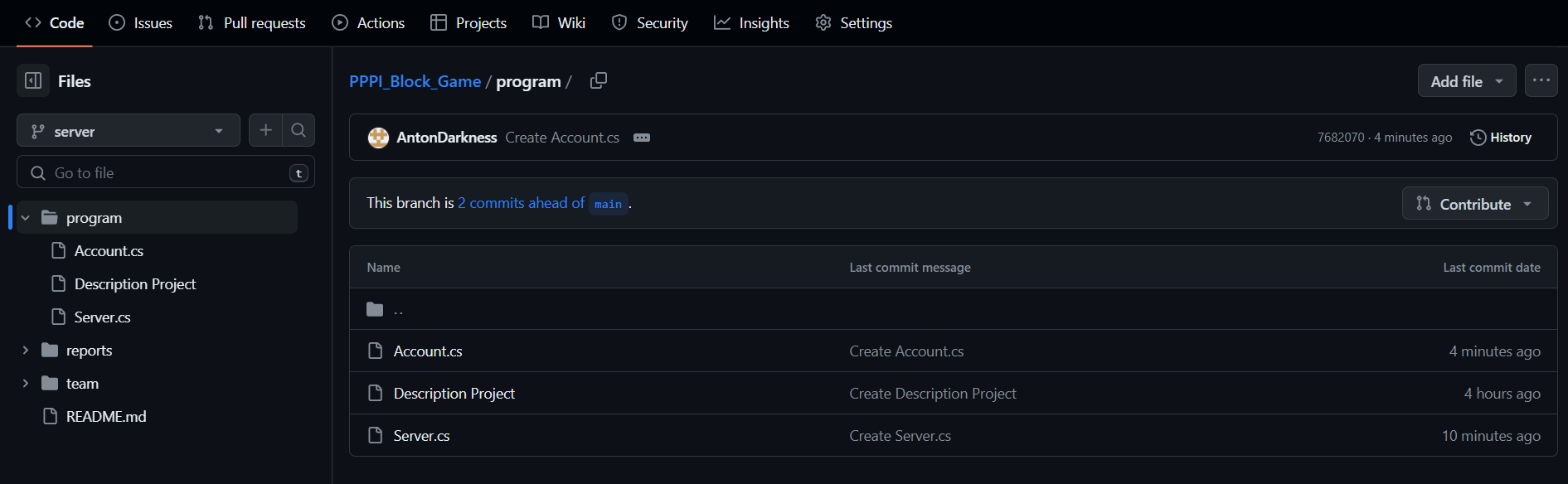


Рисунок 3 — Ветка server

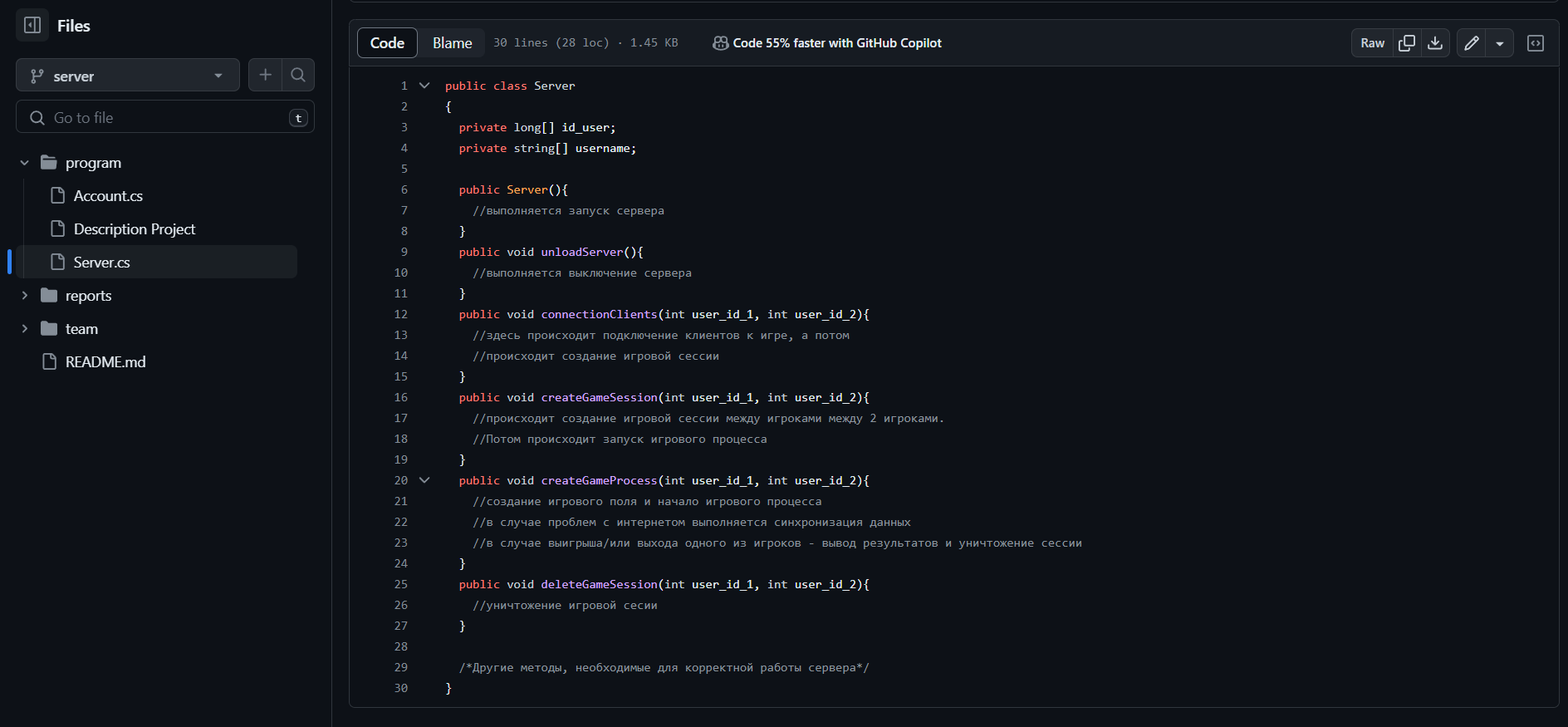


Рисунок 4 — Ветка server (файл Server.cs)

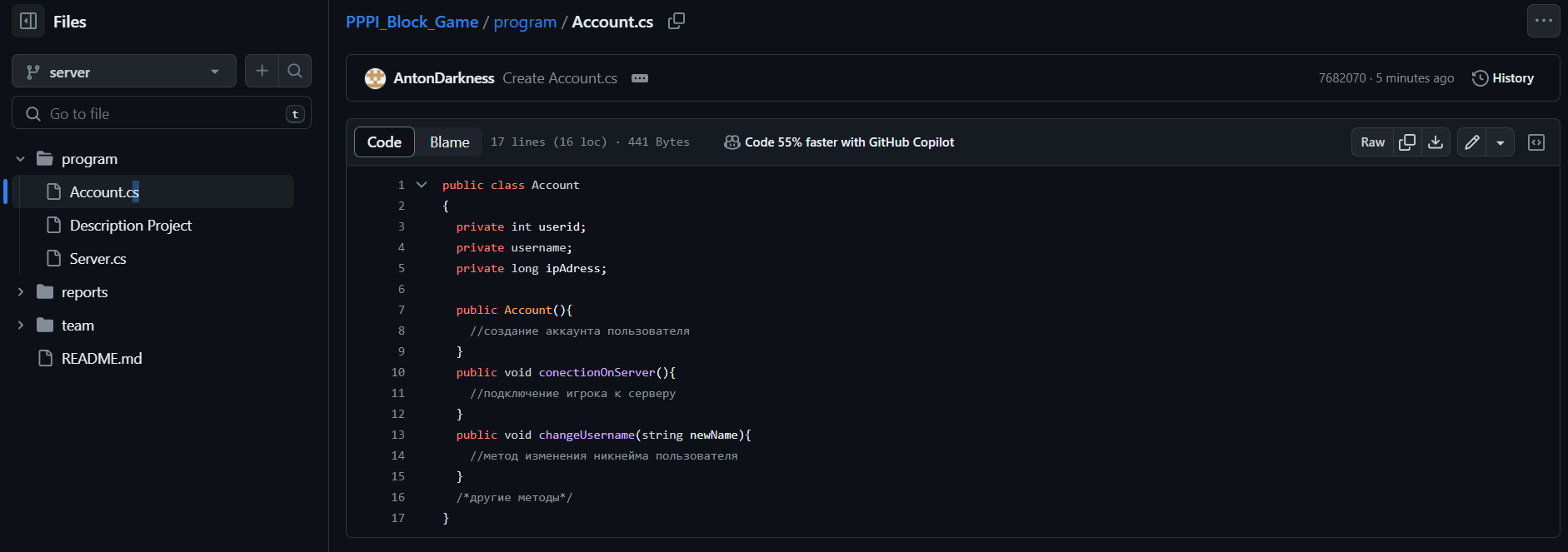


Рисунок 5 — Ветка server (файл Account.cs)

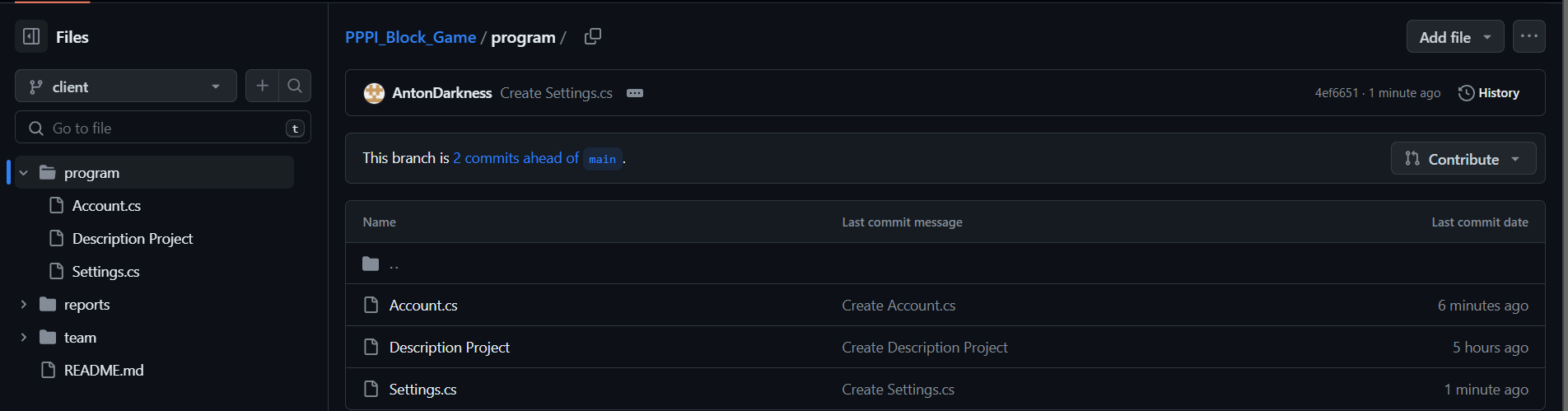


Рисунок 6 — Ветка client

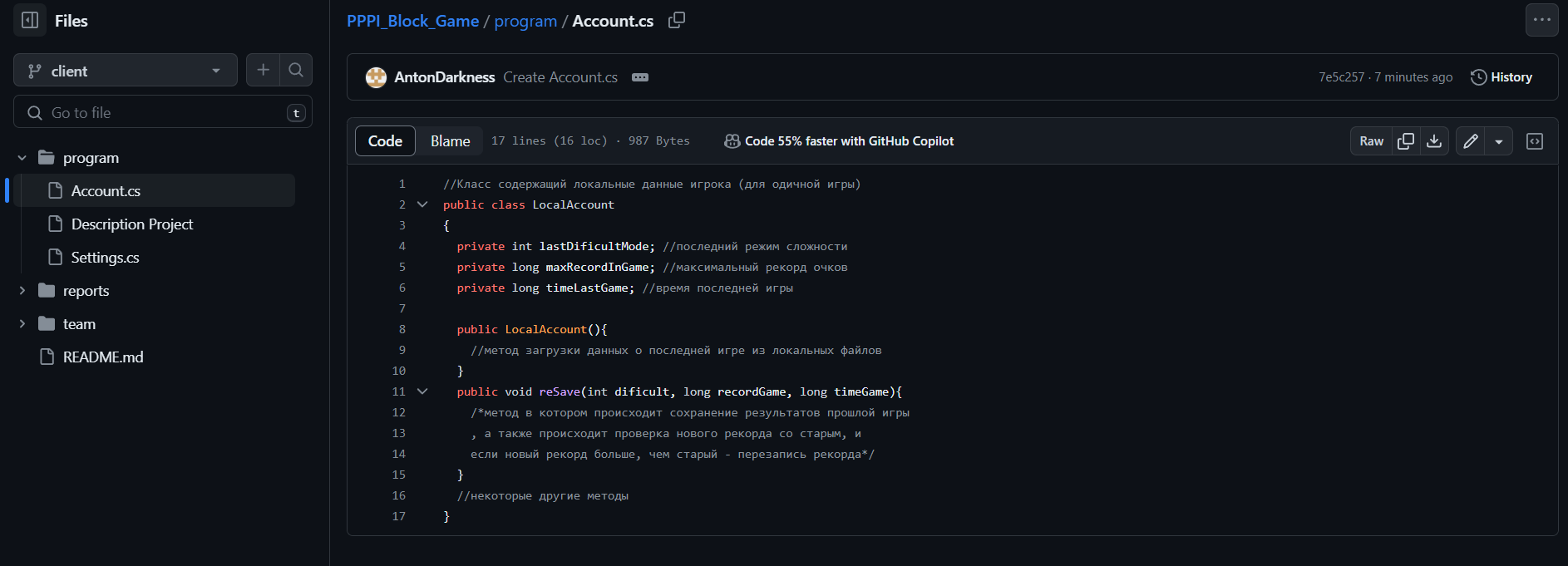


Рисунок 7 — Ветка client (файл Account.cs)

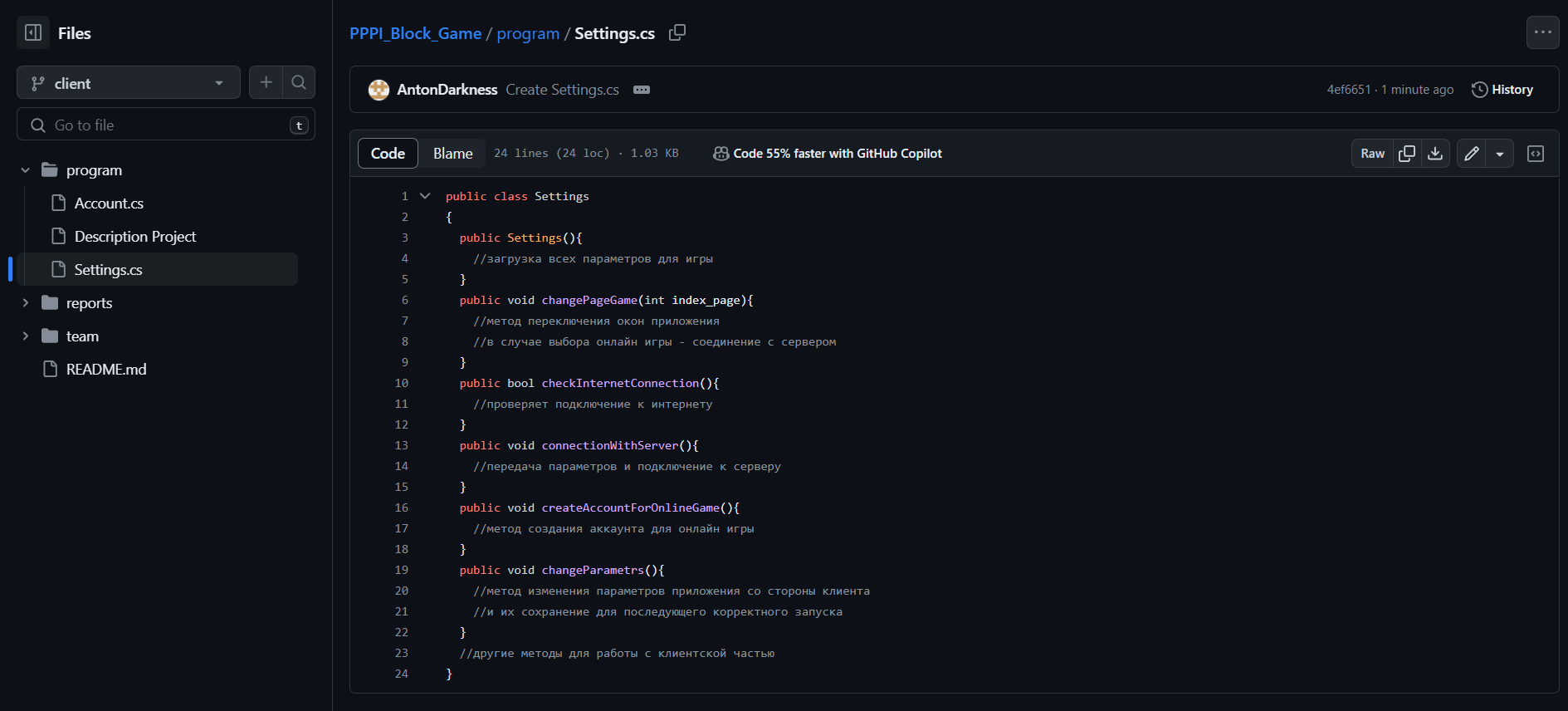


Рисунок 8 — Ветка client (файл Settings.cs)

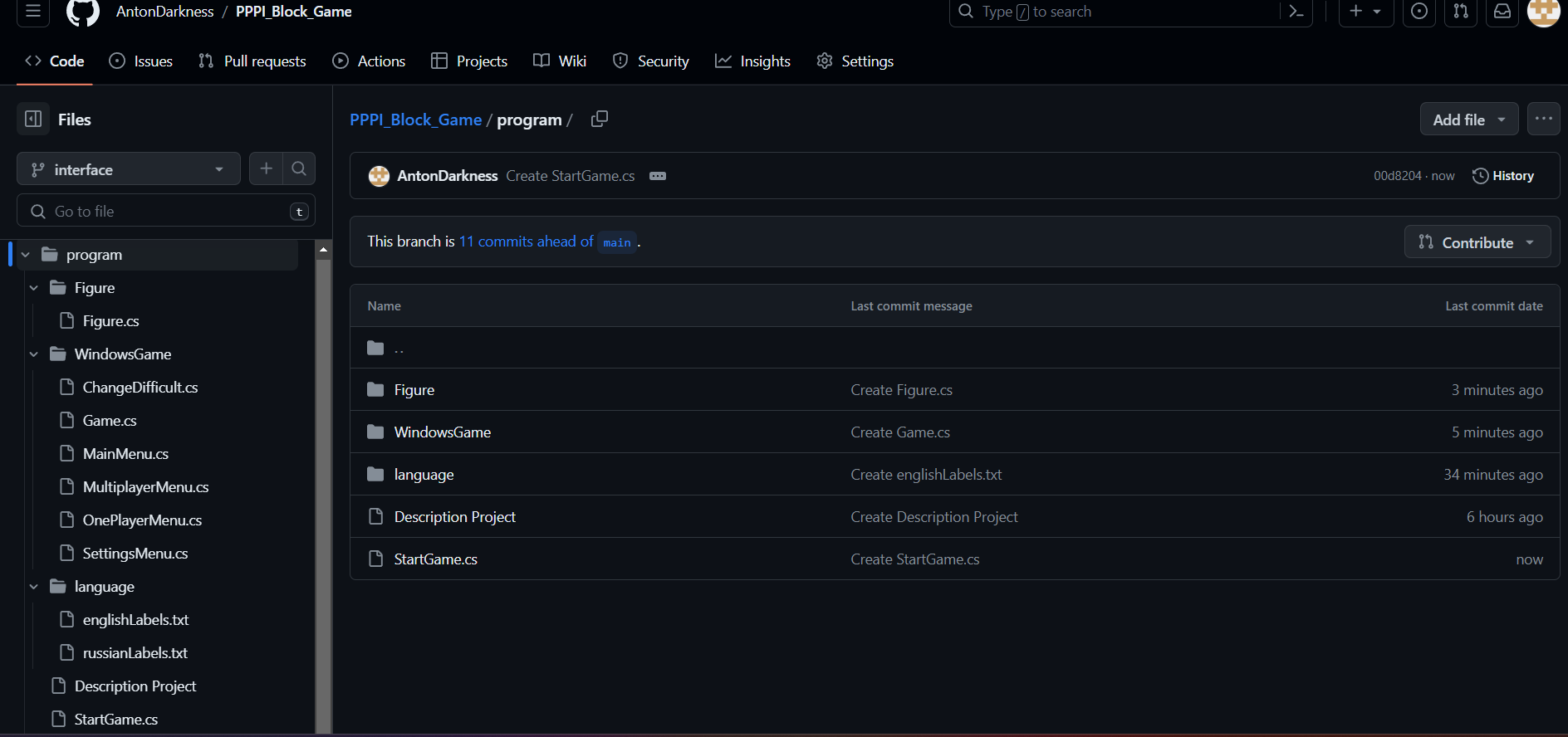


Рисунок 9 — Ветка interface

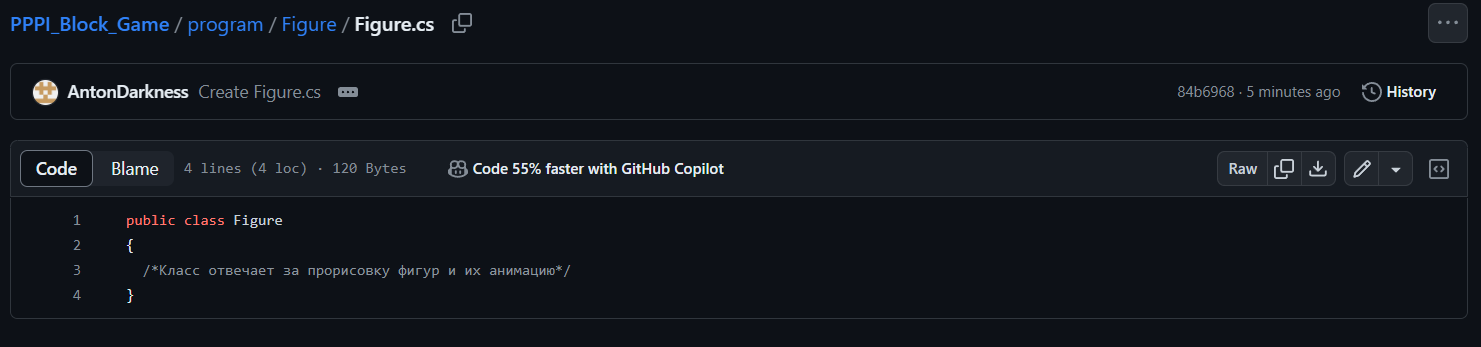


Рисунок 10 — Ветка interface (файл Figure.cs)

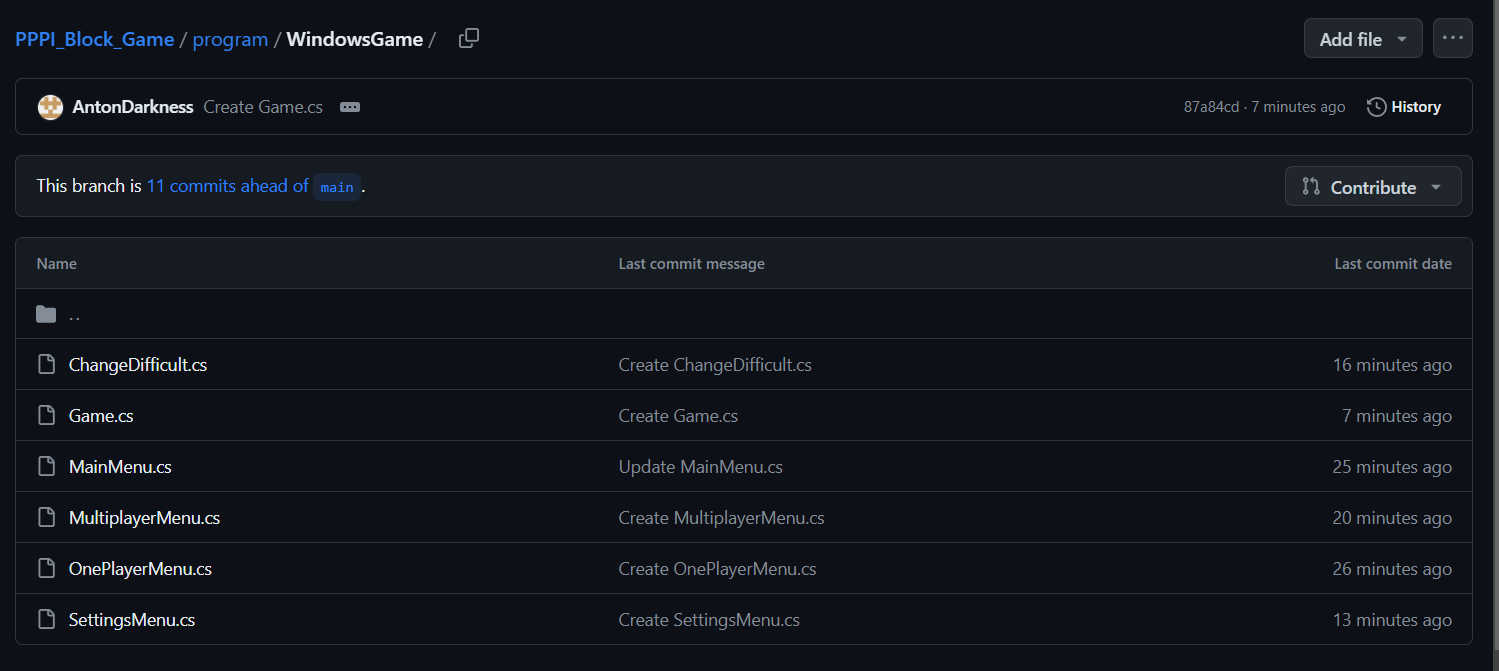


Рисунок 11 — Ветка interface (папка WindowsGame)

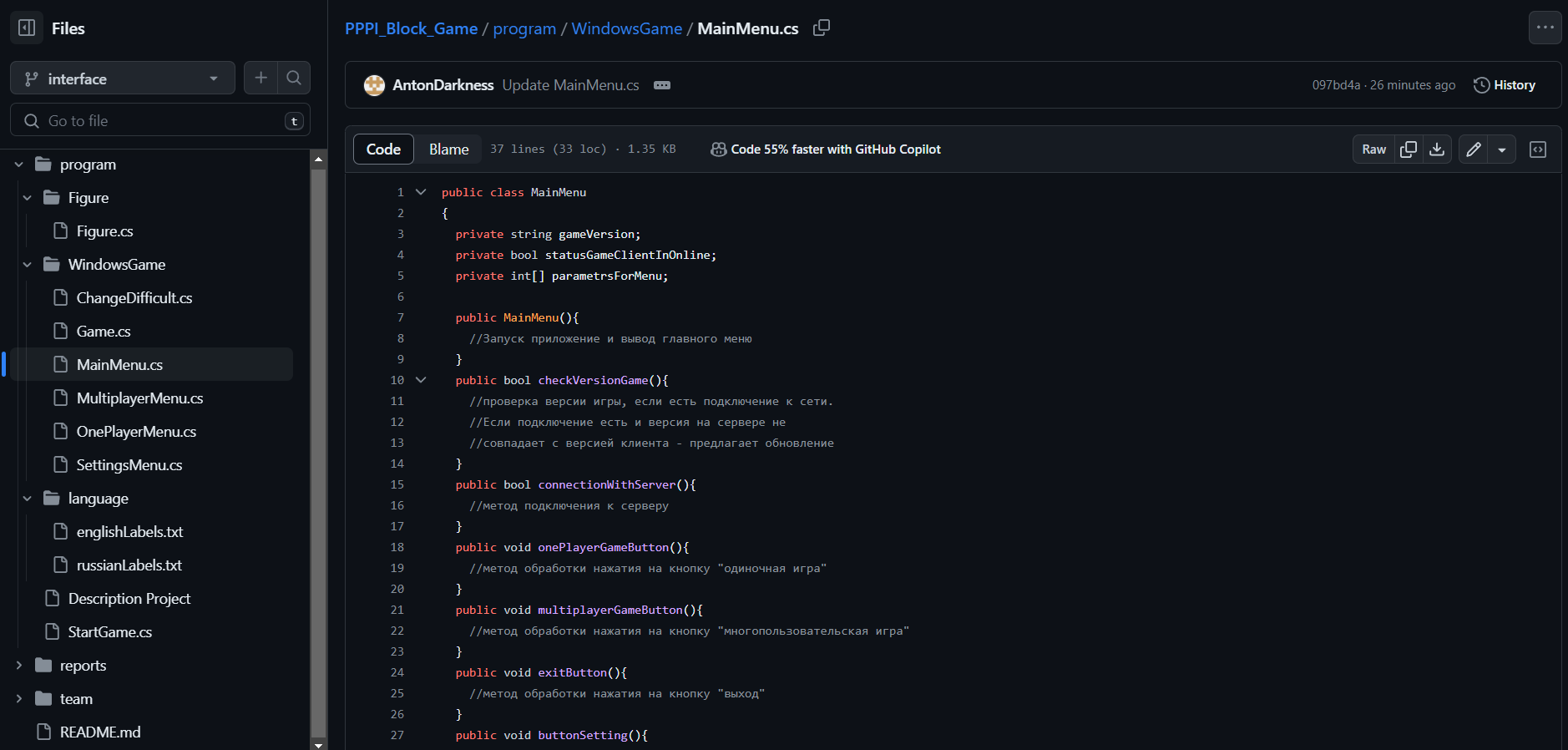


Рисунок 12 — Ветка interface (файл MainMenu.cs)

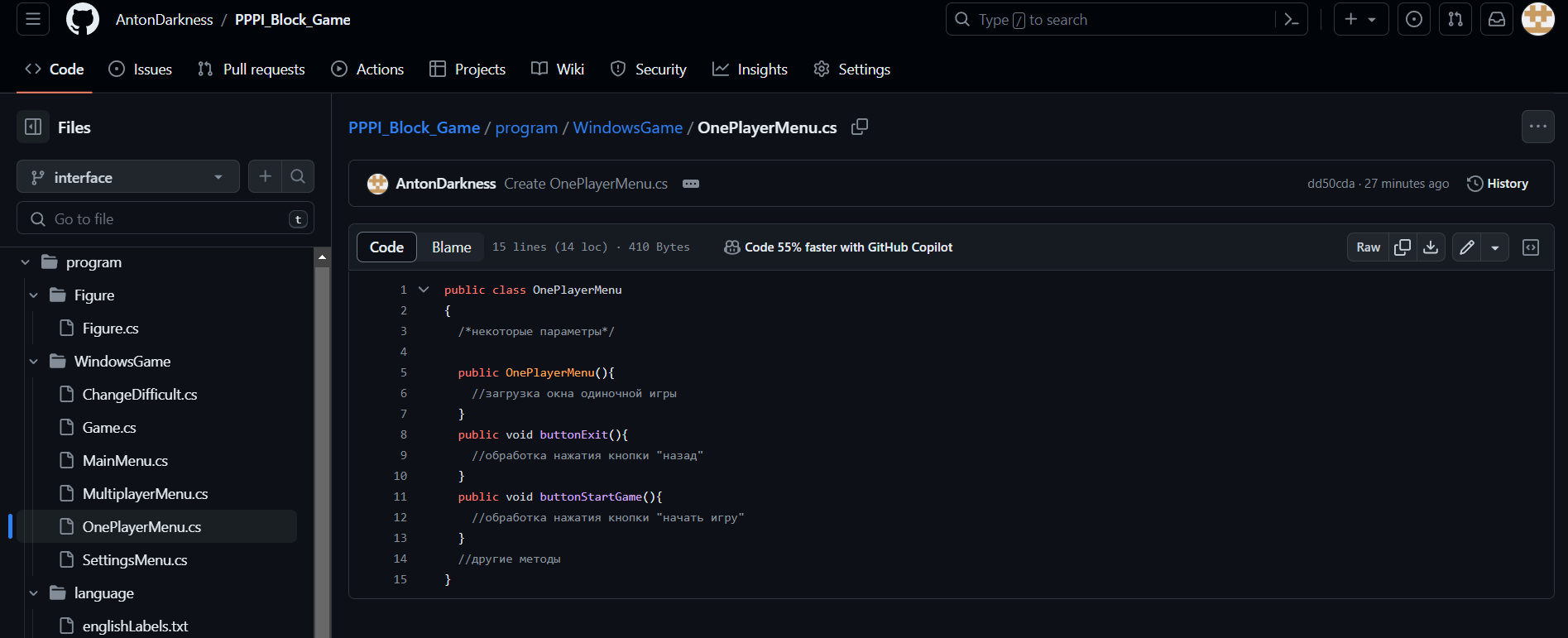


Рисунок 13 — Ветка interface (файл OnePlayerMenu.cs)

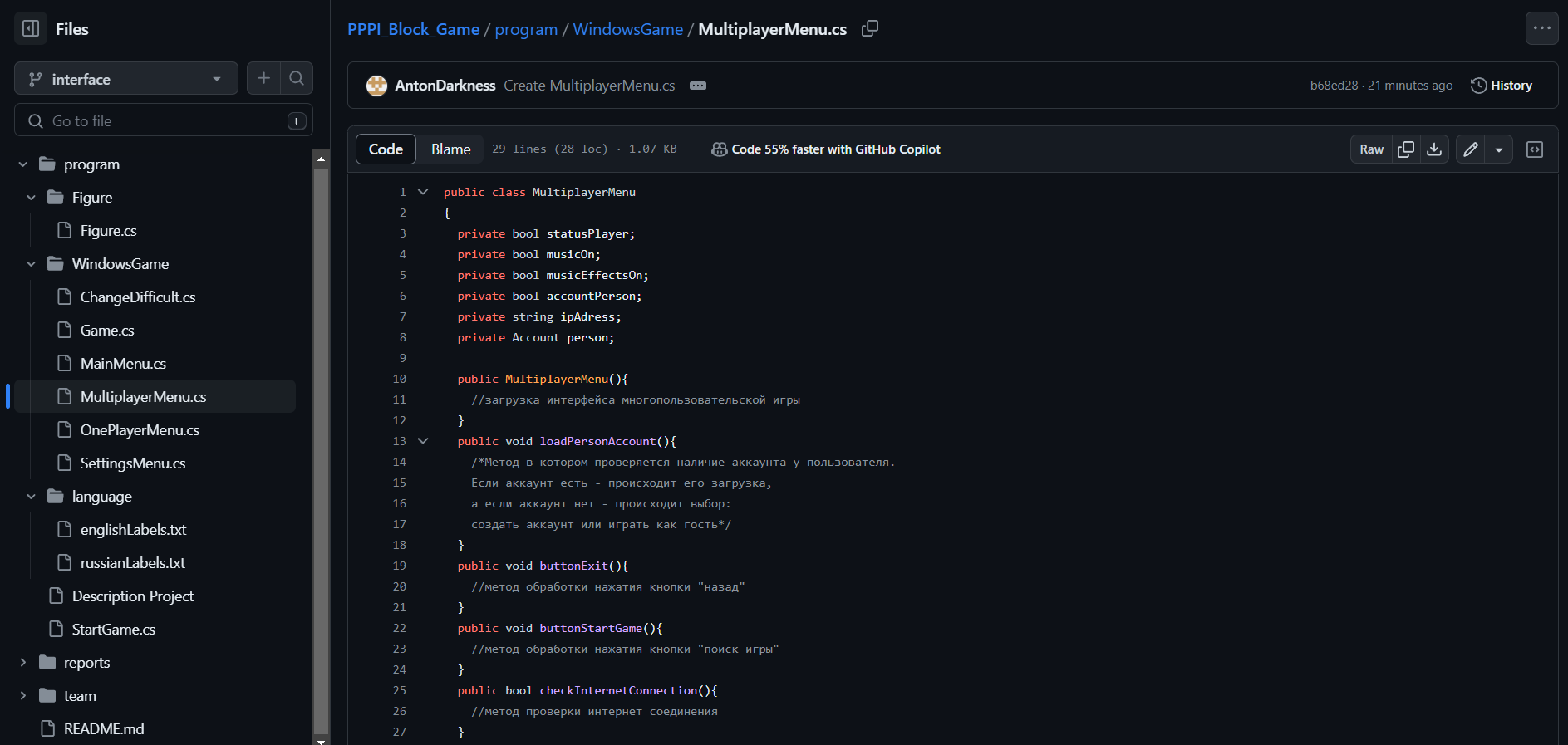


Рисунок 14 — Ветка interface (файл MultiplayerMenu.cs)

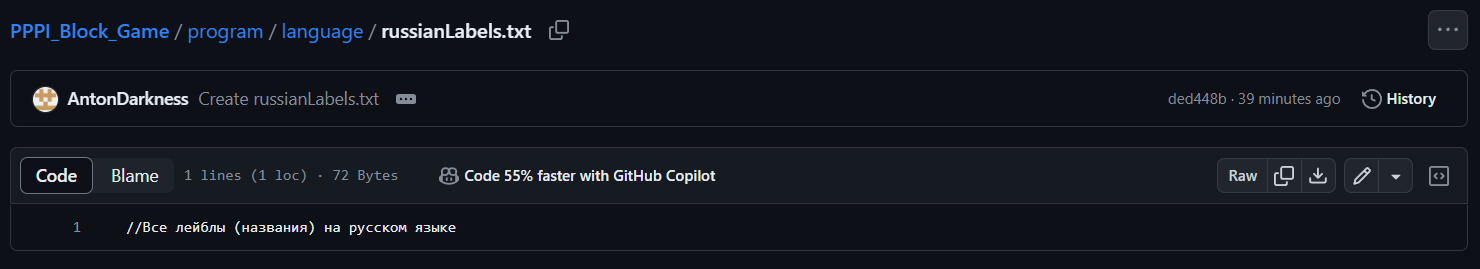




Рисунок 15 — Ветка interface (файлы russianLabels.txt и englishLabels.txt)

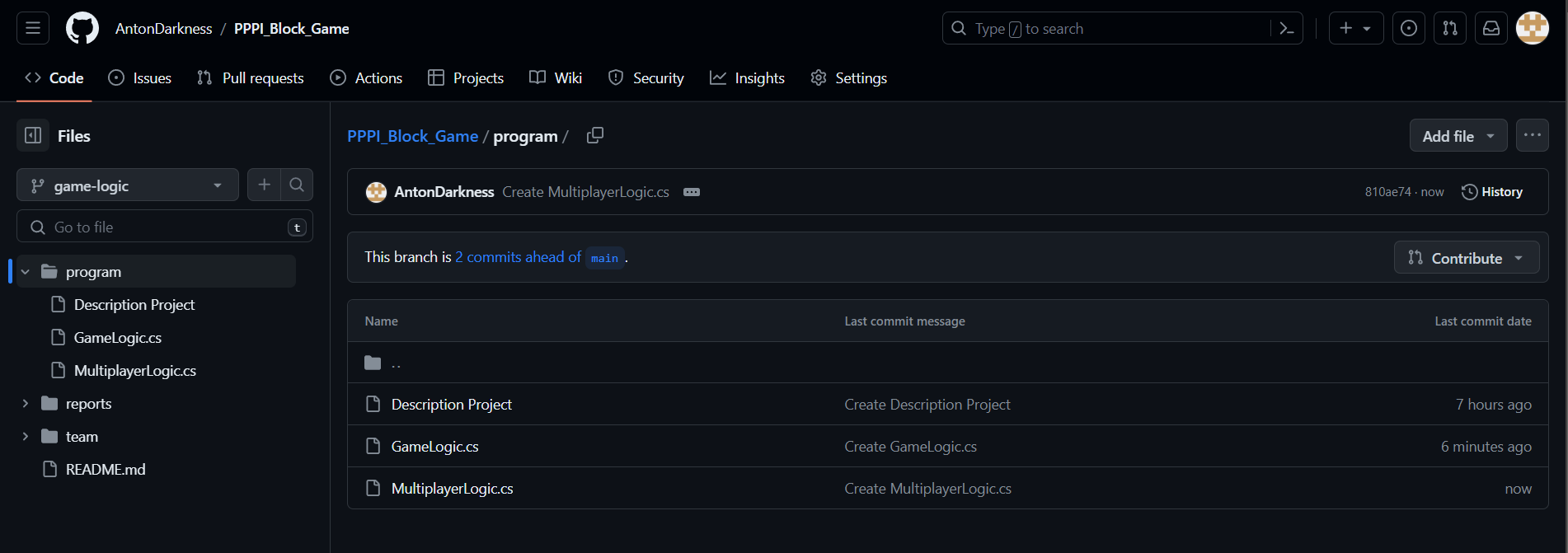


Рисунок 16 — Ветка game-logic

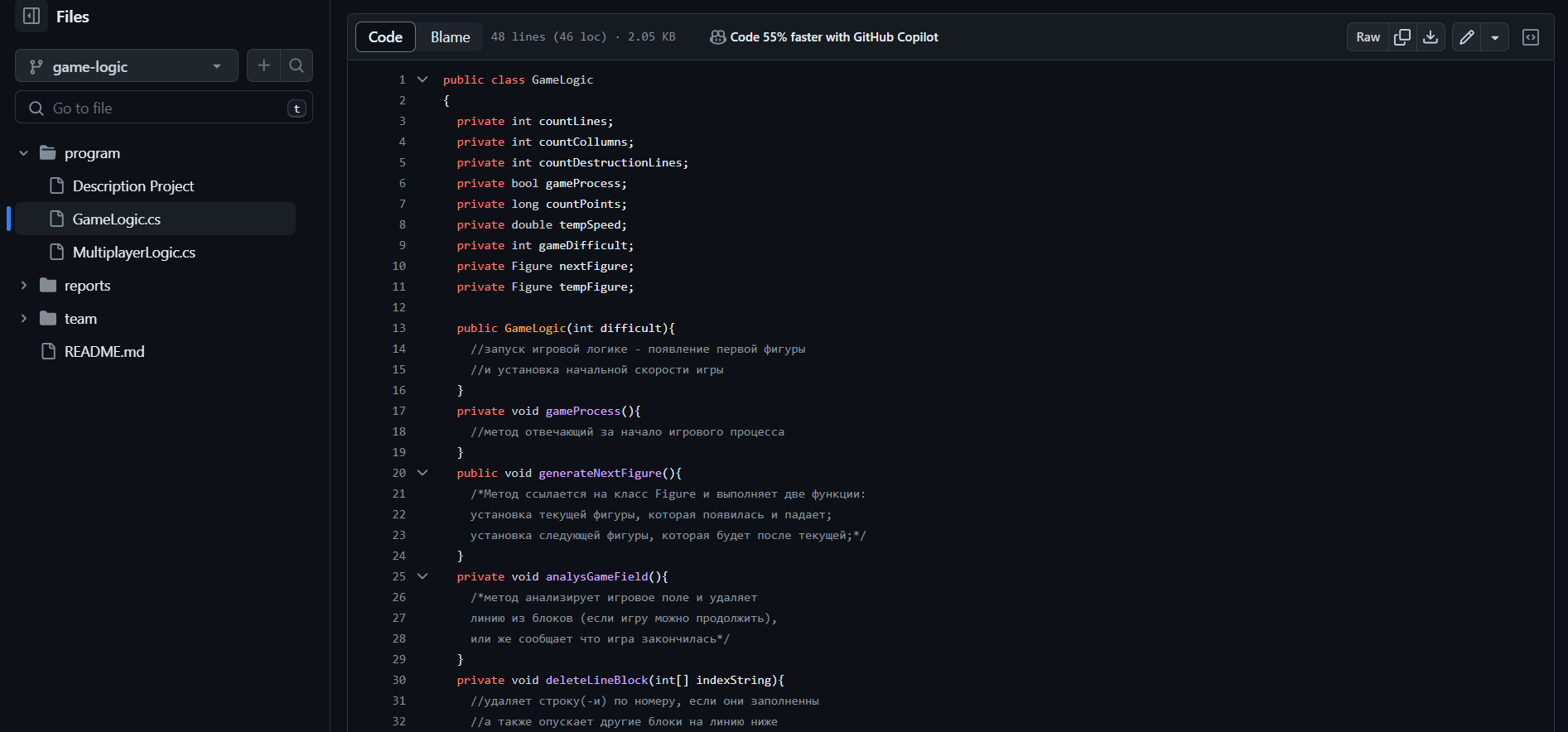


Рисунок 17 — Ветка game-logic (файл GameLogic.cs)

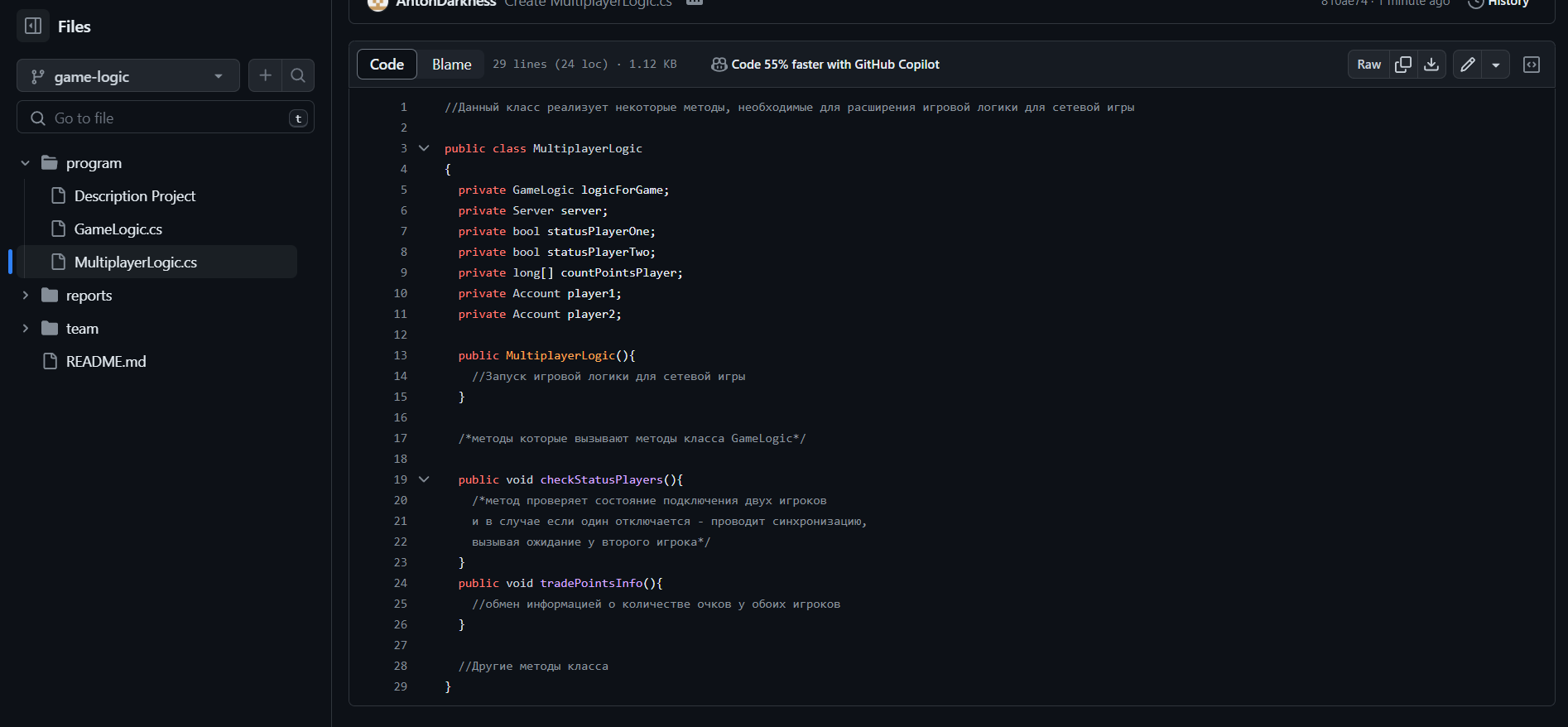
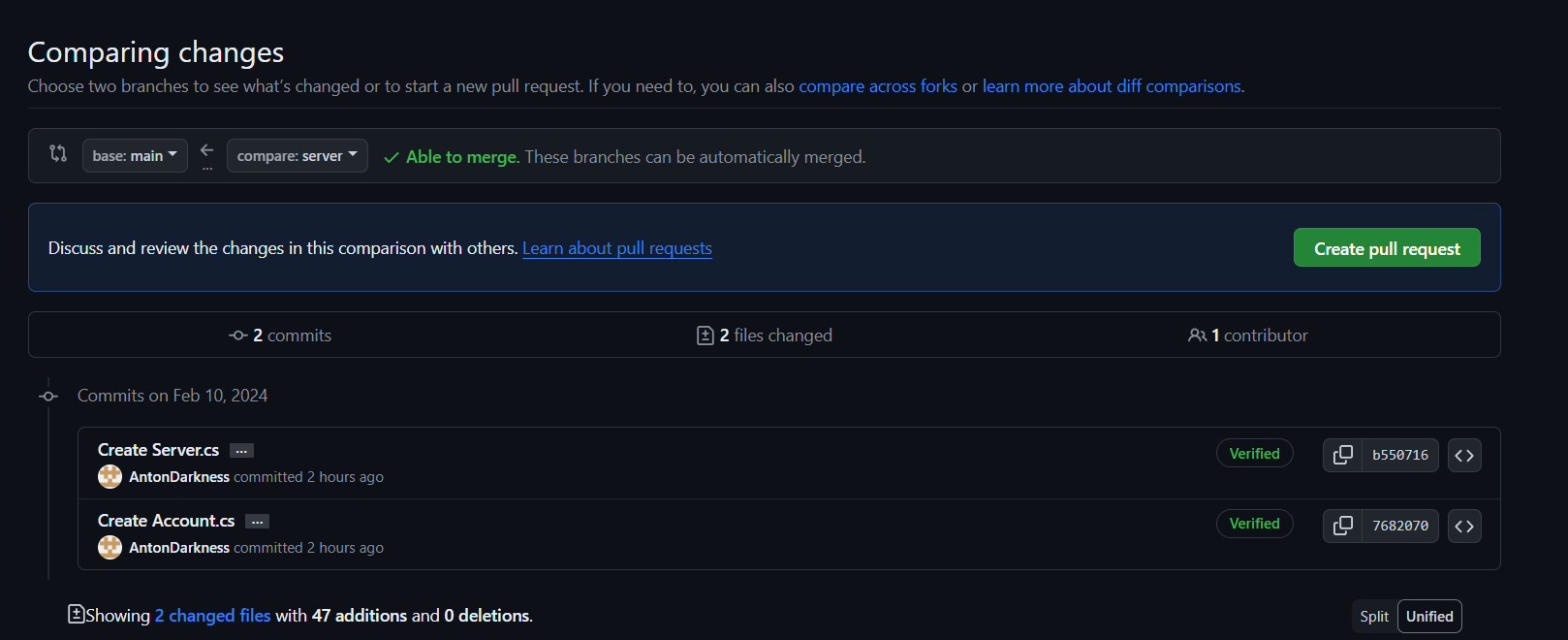


Рисунок 17 — Ветка game-logic (файл MultiplayerLogic.cs)



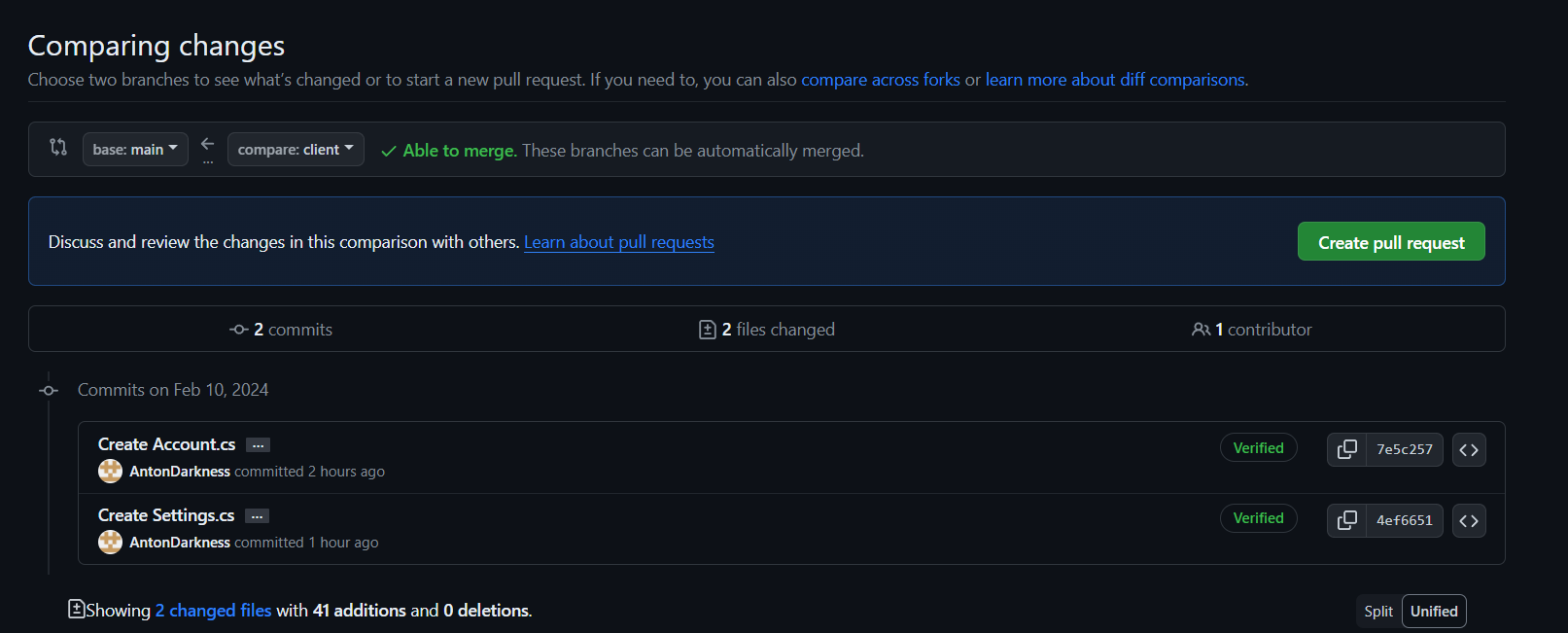
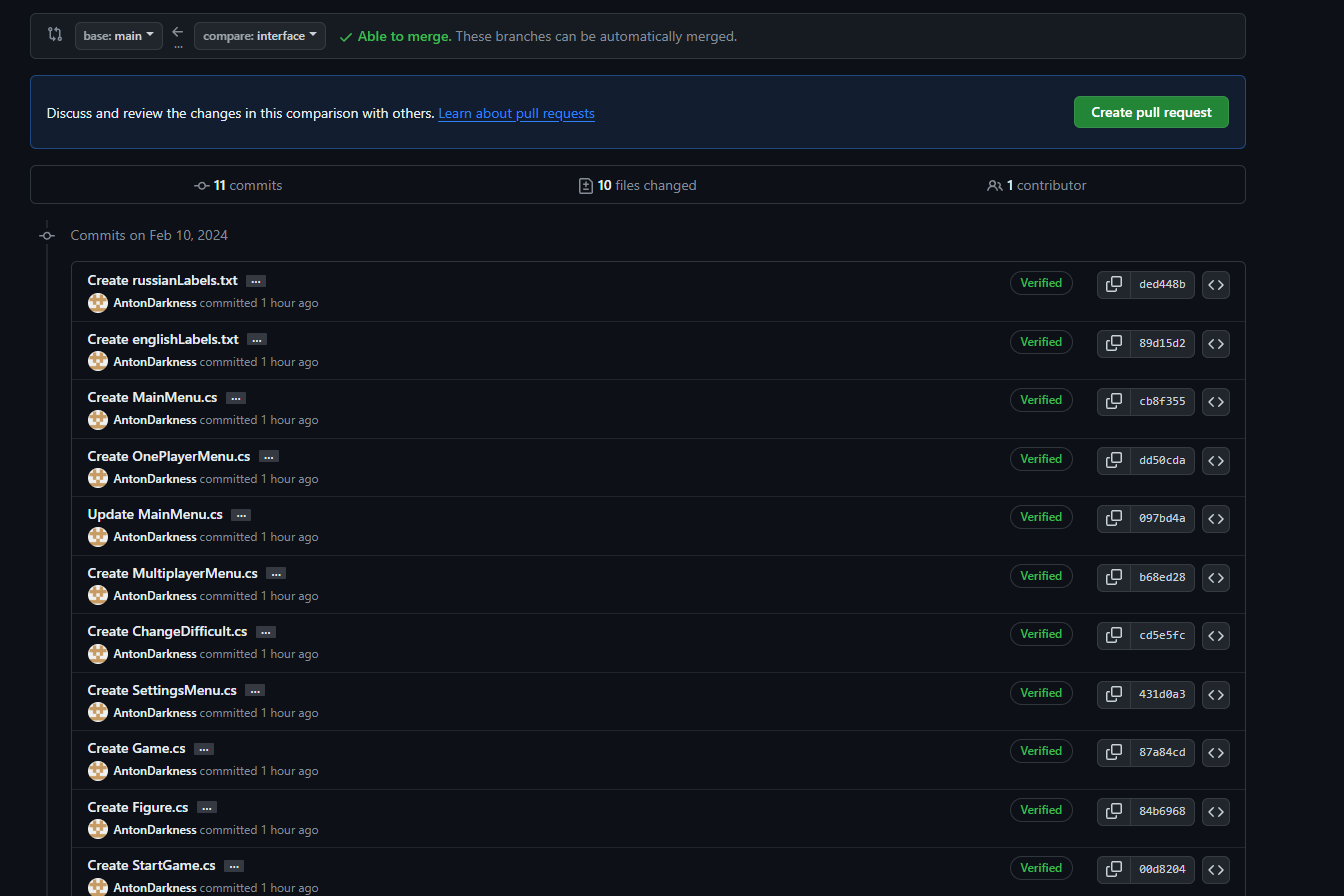


Рисунок 18 — История коммитов (ветки server и client)



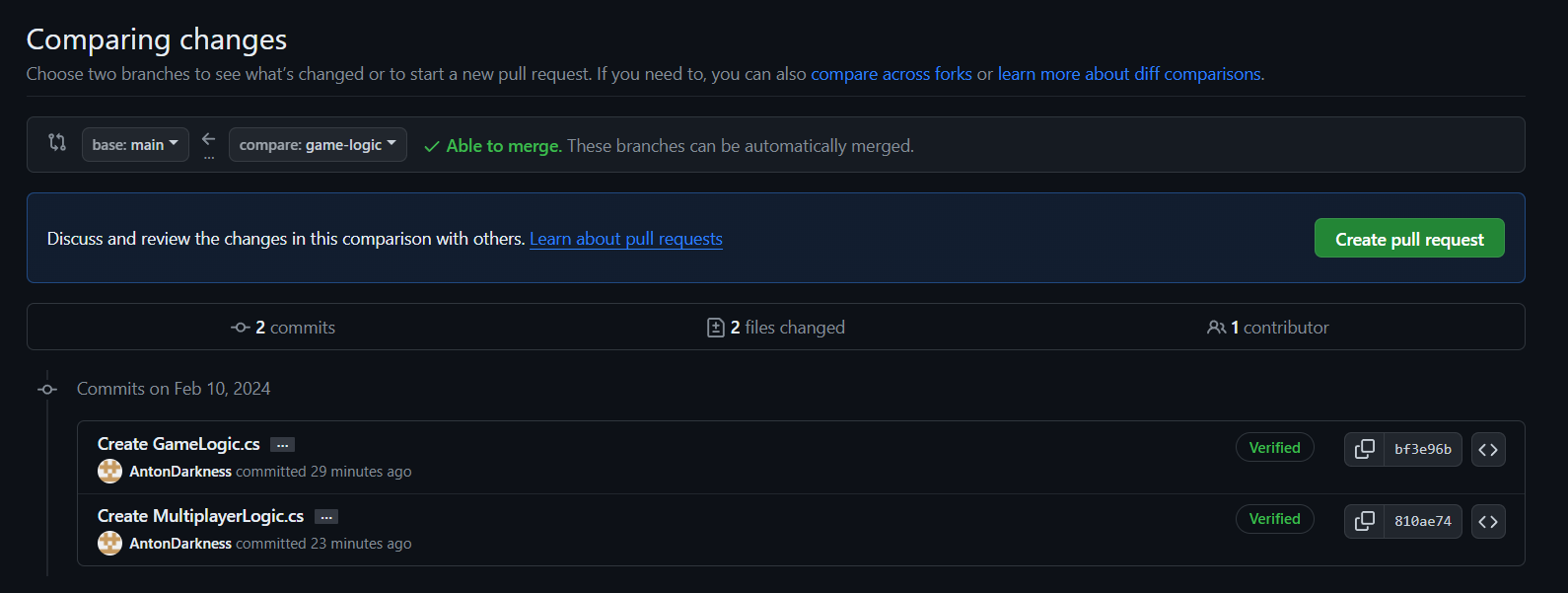


Рисунок 19 — История коммитов (ветки interface и game-logic)

**Вливаем все ветки в одну (мерж), создание конфликтной ситуации**

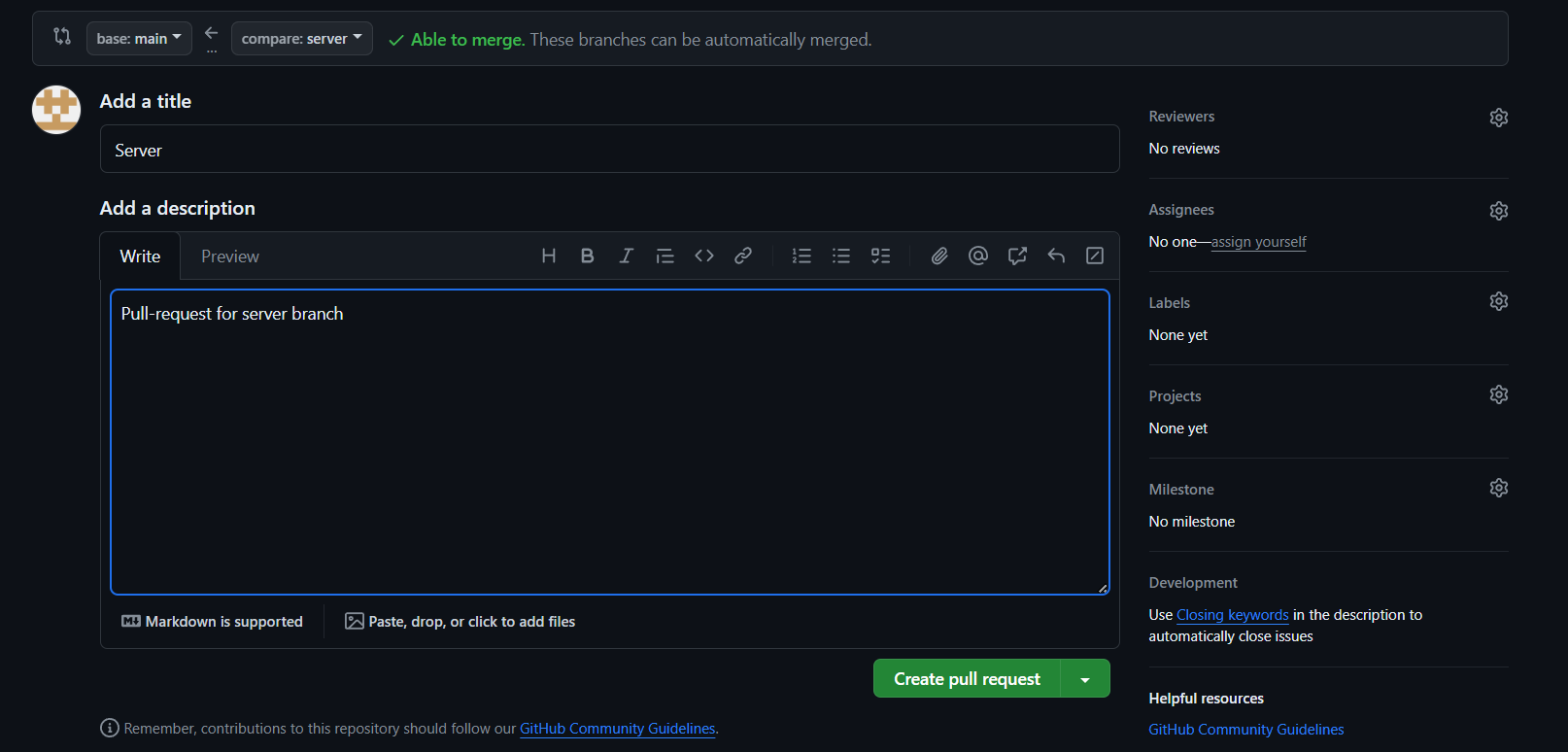


Рисунок 20 — Создаем pull-request для ветки server

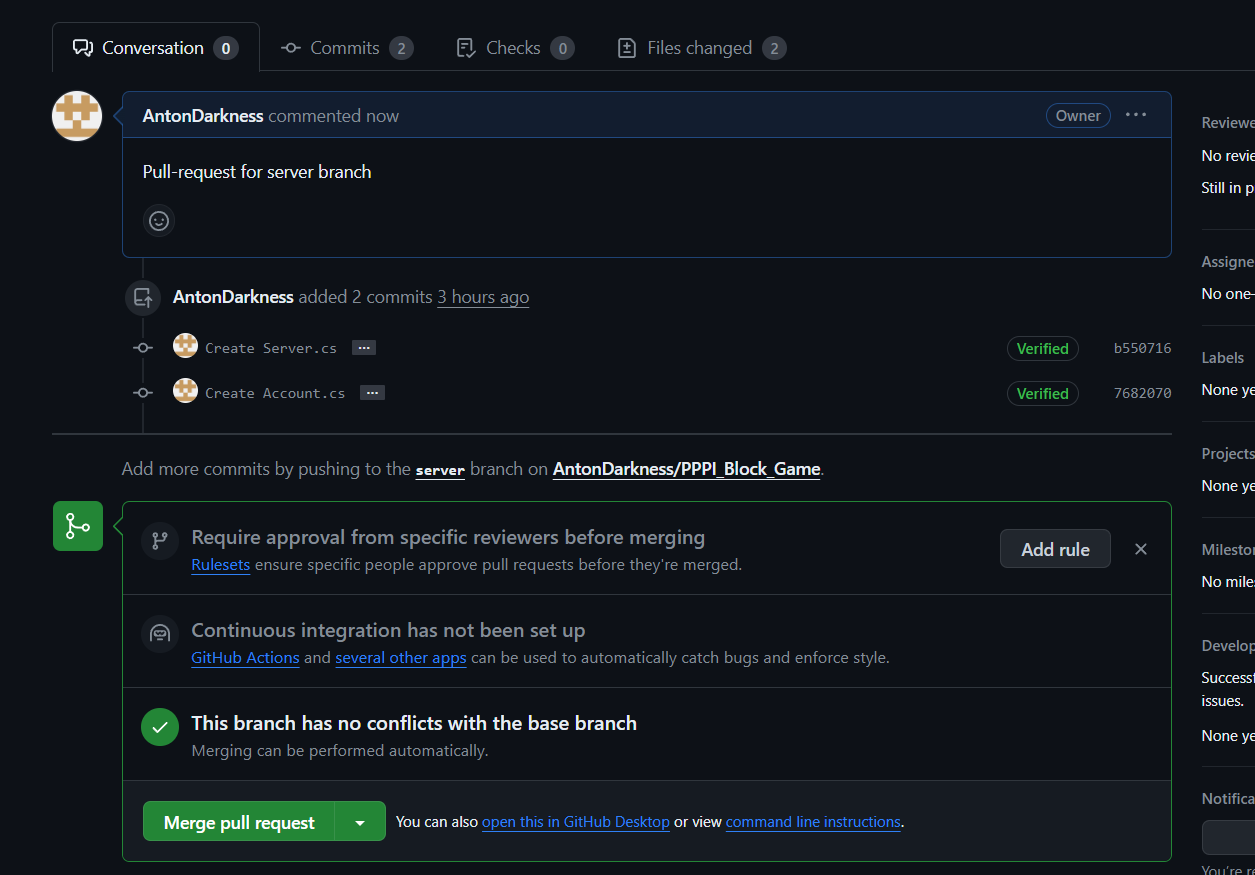


Рисунок 21 — Информация о возможности слияния



Рисунок 22 — Удачное слияние веток и удаление ветки

Теперь будем проделывать тоже самое для ветки client, однако можно сразу заметить, что ветка содержит конфликт. Это вызвано тем, что в ветке server, которую мы слили с основной был файл Account.cs, такой-же файл содержится в ветке client.

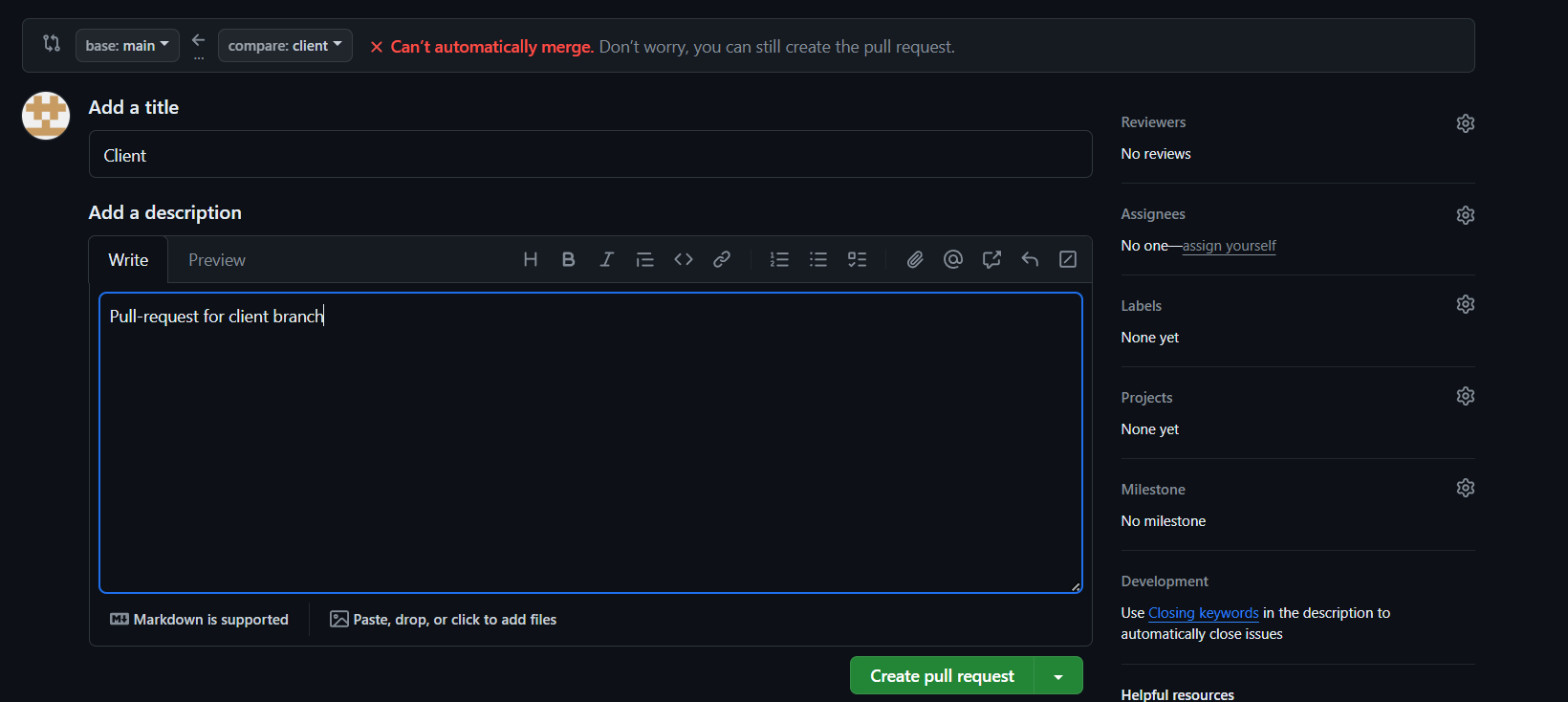


Рисунок 23 — Создаем pull-request для ветки client

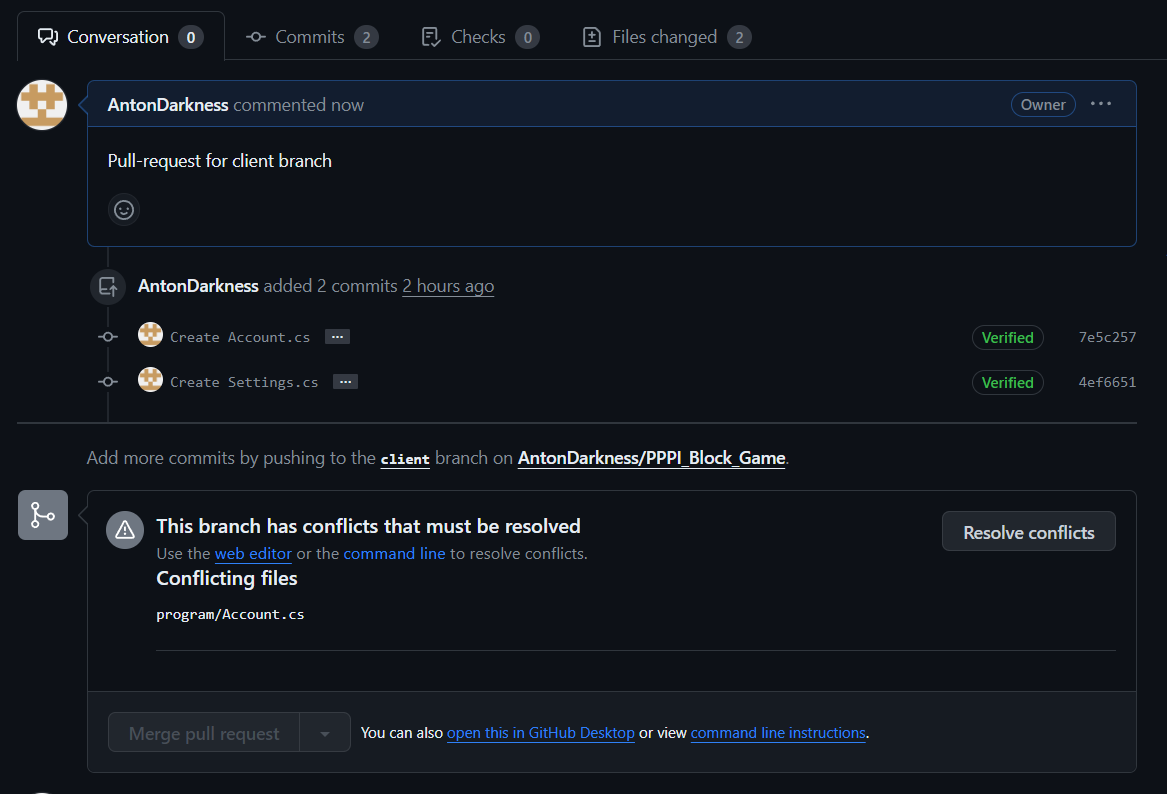


Рисунок 24 — Попытка слияния (видно, что требуется ручное решение конфликта)

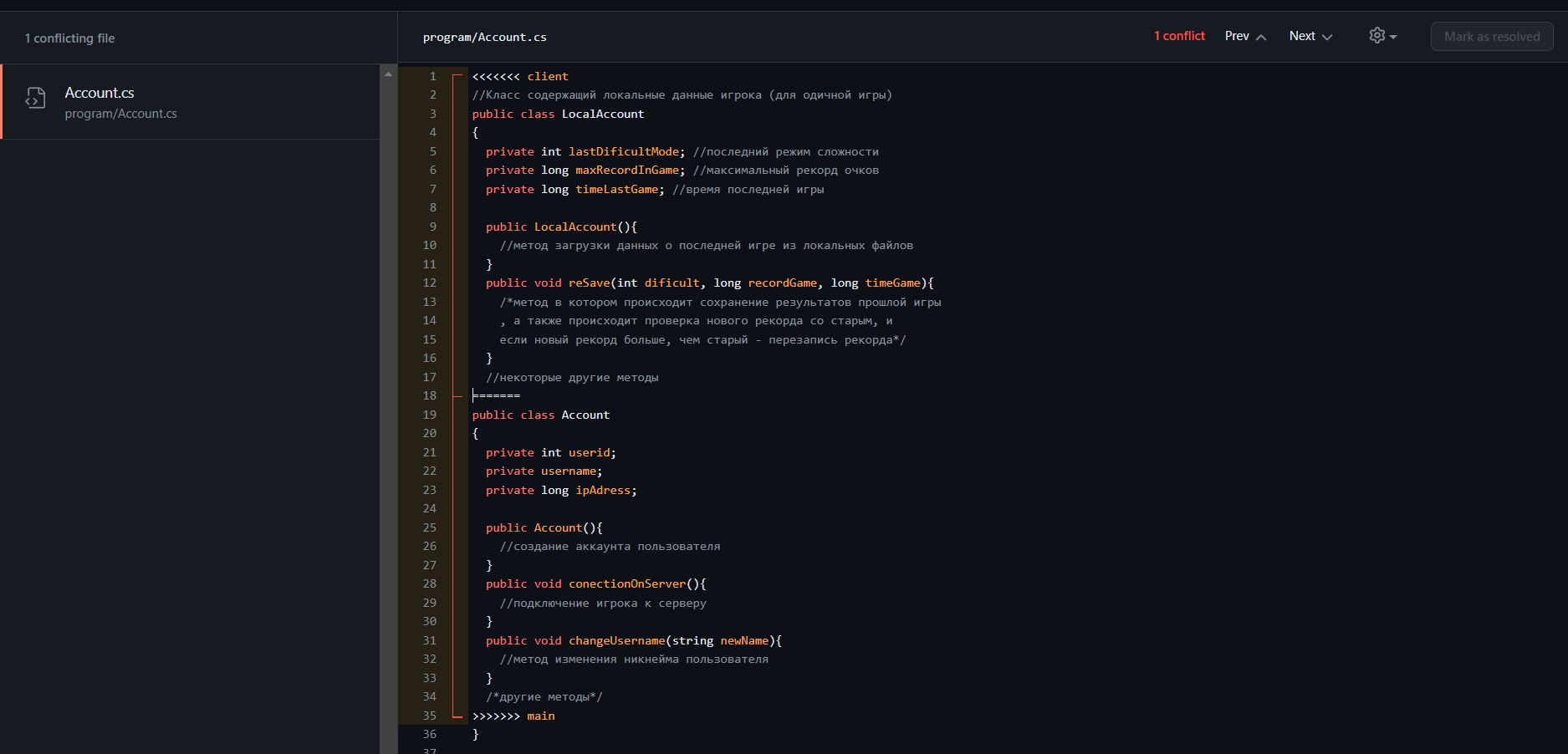


Рисунок 25 — Разница в строках двух файлов при попытке слияния веток

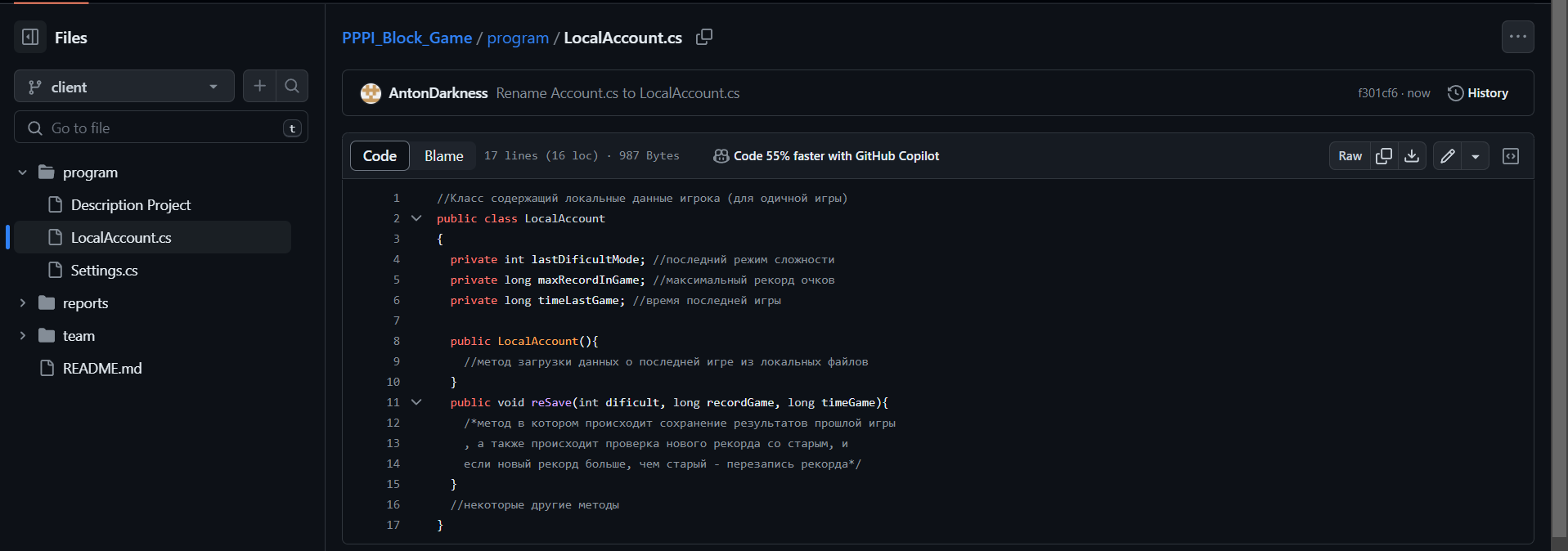


Рисунок 26 — Решение переименовать файл в ветке client на LocalAccount.cs (однако можно просто оставить один файл, слив функционал в общий)

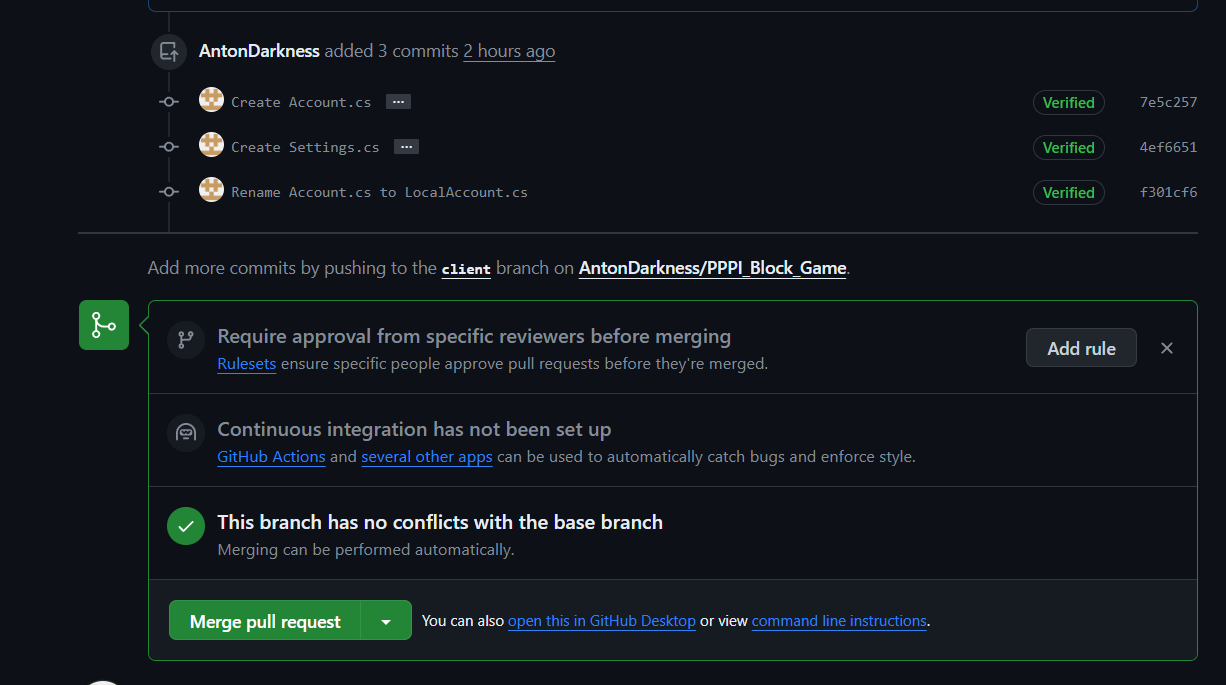


Рисунок 27 — Возможность слияния веток теперь доступна

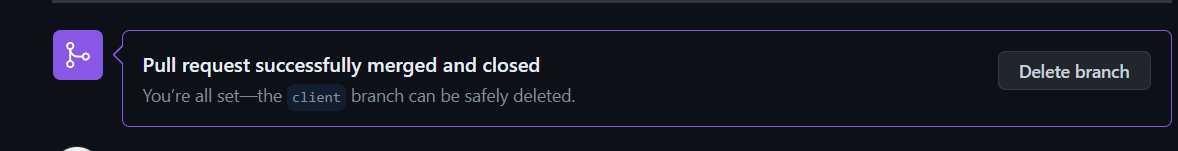


Рисунок 28 — Результат слияния

Данное действие повторяем для оставшихся веток, попутно проверяя на конфликты. Конфликтов при слиянии других веток – не возникло.

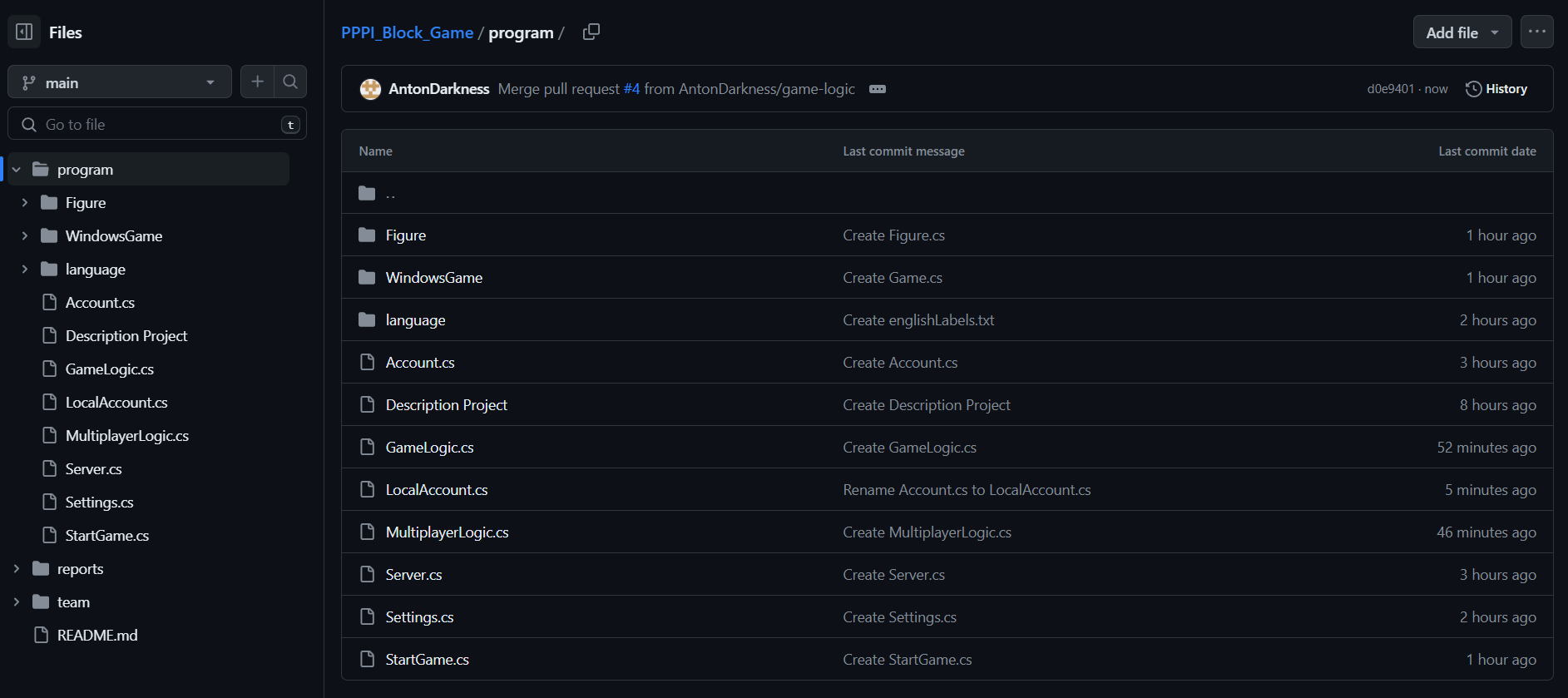


Рисунок 29 — Структура основной ветки в папке program после всех pull-request

Удаляем все ветки, кроме основной (main/master).

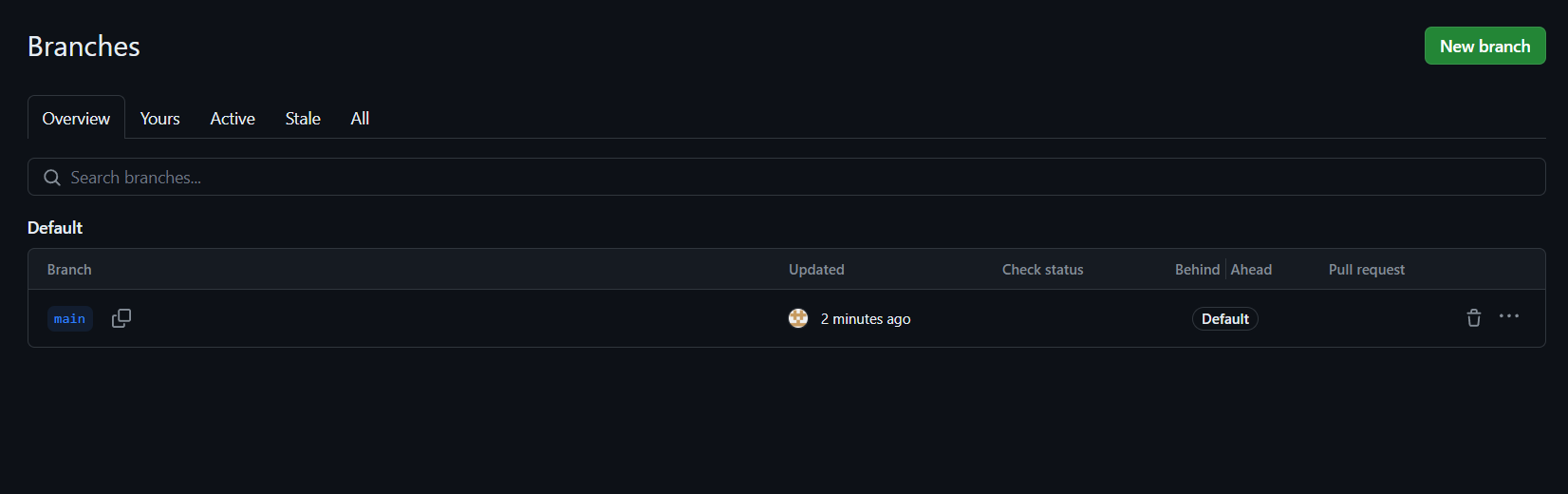


Рисунок 30 — Результат после удаления веток