МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

КАФЕДРА ПРОГРАМНОЙ ИНЖЕНЕРИИ

Кафедра ПИ

Лабораторная работа №2

«Работа с ветками»

Проверил:

Незамова Л.В.

Выполнил студент

группы ПИ-20Б

Шатохин И.И.

Донецк – 2024

1. В проекте имеются следующие ветки:

PlayerLogic: Отвечает за логику работы главного игрока, а так же оружия которыми он пользуется

WorldInteractions: Отвечает за взаимодействие с внешним миром, а так же рендер

OnlineFeatures: Отвечает за онлайн составляющую проекта

Storytelling: Отвечает за сюжетную составляющую игры

Все ветки представлены на рисунке 1

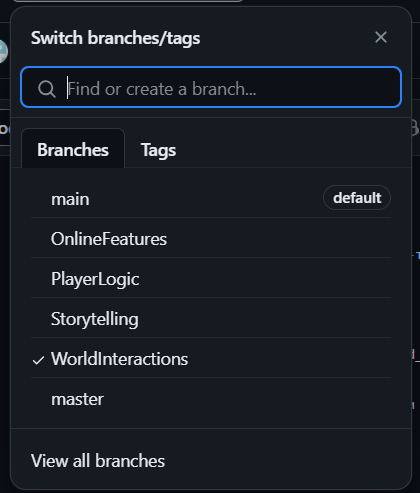


Рисунок 1 – Перечень веток репозитория

1. Создание веток происходит следующим образом



Рисунок 2 – Создание ветки PlayerLogic

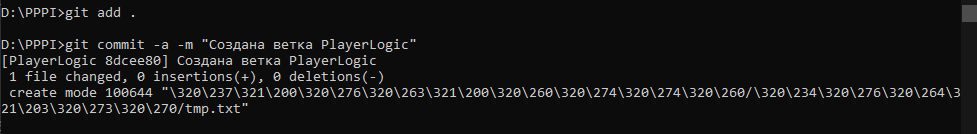


Рисунок 3 – Добавление изменений в ветку

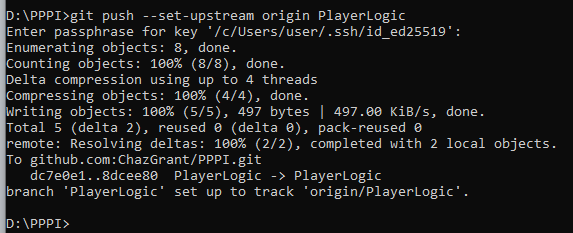


Рисунок 4 - Обновление информации на сервере

3.1 Файл Player.py находится в ветке PlayerLogic и реализует следующие модули:

* Перемещение персонажа
* Добавление способности парирования, переката
* Перекат
* Отдых у костра
* Заклинания
* Сопротивление элементальному урону
* Очки здоровья, концентрации и выносливости

3.2 Файла Weapon.py находится в ветке PlayerLogic и реализует следующие модули:

* Улучшение оружия
* Заточка оружия
* Починка оружия
* Получение текущей прочности оружия

3.3 Файл Interface.py находится в ветке WorldInteractions и реализует следующие модули:

* Вывод сообщений
* Отображение фона локации
* Отображение атмосферных явлений
* Отображение карты
* Подбираемые предметы

3.4 Файл World.py находится в ветке WorldInteractions и реализует следующие модули:

* Получение достижения
* Коллизии
* Урон от болот
* Создание врага
* Создание костра
* Помещение боссов на карту
* Возрождение всех врагов
* Взаимодействие с объектами
* Подбор душ
* Повышение уровня персонажа
* Нанесение стихийного урона

3.5 Для того чтобы произвести слияния веток используется следующие синтаксис “git merge название ветки”. Слияние происходит с той веткой, на которой мы в данный момент находимся. Для перехода на нужную ветку используется команда “git checkout –b название\_ветки”

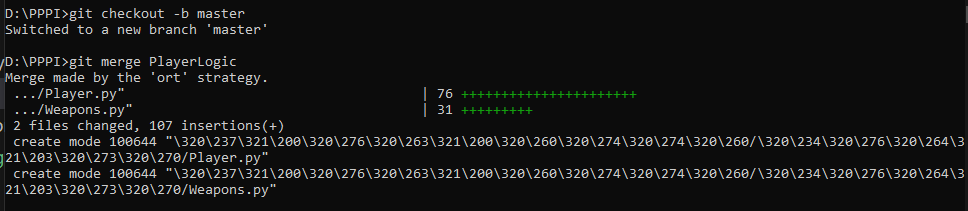


Рисунок 5 – Слияние ветки PlayerLogic с главной веткой master

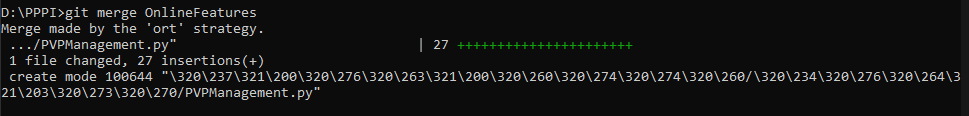


Рисунок 6 – Слияние ветки OnlineFeatures с веткой master

С остальными ветками логика слияния такая же

Для получения конфликтной ситуации, были изменены некоторые участки кода в следующих файлах

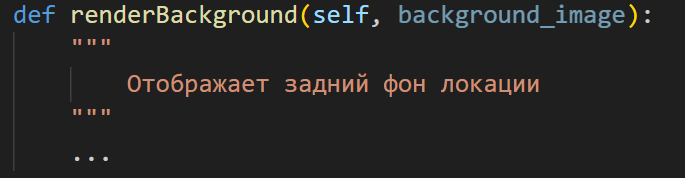


Рисунок 7 - Ветка WorldInteractions. Файл Interface.py

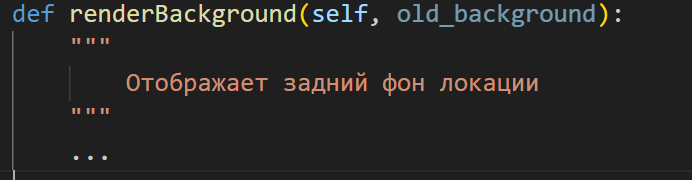
****

Рисунок 8 - Ветка master. Файл Interface.py

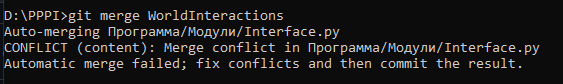
****

Рисунок 9 – Сообщение об ошибке при слиянии

После выполнения команды git merge в редакторе кода на ветке master появились ошибки слияния, которые можно решить оставив предыдущее состояние или обновить новым

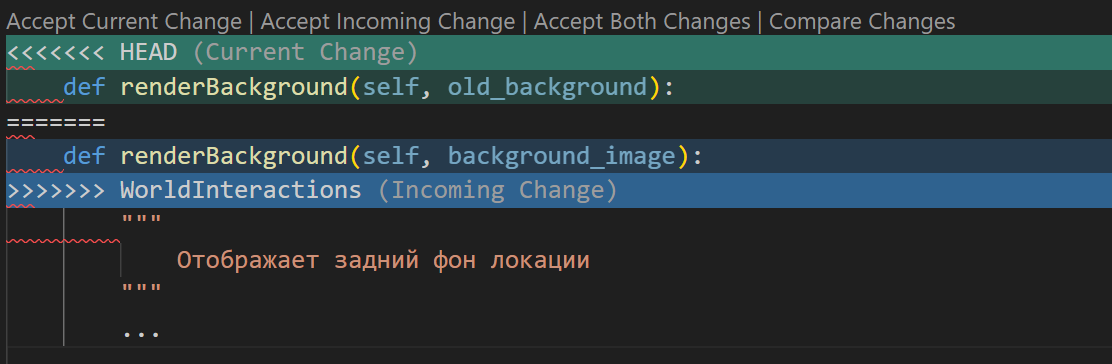
****

Рисунок 10 – Предложение решить конфликт в редакторе кода

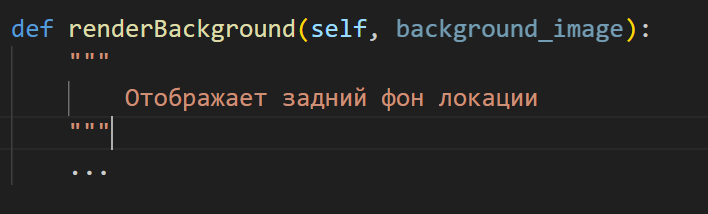
****

Рисунок 11 – Итоговая часть кода Interface.py в ветке master

На рисунке 12 показано что конфликтов слияний больше нет

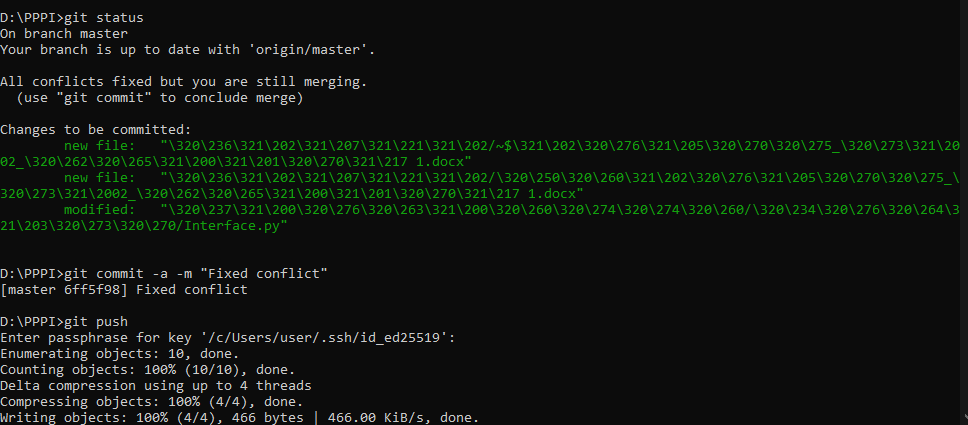


Рисунок 12 – Обновление информации на сервере

Для того чтобы удалить ветку, необходимо написать “git push –d название\_удаленного\_адреса название\_ветки”. На рисунке 13 показано удаление всех веток, созданных в этой лабораторной работе.

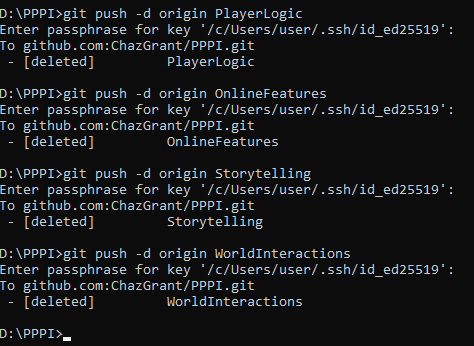


Рисунок 13 – Удаление всех веток