1. **ПРЕДМЕТНАЯ ОБЛАСТЬ**

Doodle Jump (рис. 1) – это 2д игра с видом сбоку. Жанр: аркада с элементами платформера. Был разработан студией Lima Sky, вышел на мобильные телефоны в 2009 году. Цель игры – подняться как можно выше по платформам, которые располагаются случайным образом и получить как можно больше очков. С течением времени сложность увеличивается. Также во время игры персонаж сталкивается с преградами в виде монстров. Но игрок может стрелять в них и таким образом оказывать сопротивление.

Doodle Jump стала популярной в 2009 году благодаря простому, но затягивающему геймплею, уникальному визуальному стилю с рисованной графикой, удобному управлению (наклон устройства), и доступности на первых iPhone, которые только набирали популярность.

Основная аудитория – подростки и молодёжь в возрасте от 12 до 25 лет, но игра также подойдёт для более широкой аудитории, благодаря простоте механик. Может заинтересовать любителей казуальных игр, которые предпочитают короткие игровые сессии с возможностью достижения высоких результатов. Поскольку игра не требует длительного обучения, она будет интересна как для опытных игроков, так и для новичков.

****Рисунок 1 – Doodle Jump

Применение для получения прибыли:

1. Реклама**:** Встраивание рекламы (например, баннеров или вознаграждаемых видео, которые дают бонусы игроку)
2. Платное отключение рекламы**:** Разрешение игрокам купить премиум-версию игры без рекламы

Основные механики:

1. Движение персонажа**:**
   * Управление движением влево/вправо
   * Автоматическое движение вверх по мере прыжков
2. Генерация платформ**:**
   * Платформы появляются случайным образом
   * Типы платформ:
     + Обычные (стационарные)
     + Разрушаемые (разваливаются после одного прыжка)
     + Движущиеся (перемещаются влево-вправо)
     + Ускоряющие (дают толчок выше)
3. Препятствия**:**
   * Враги (например, летающие монстры, препятствующие движению)
   * Ловушки (шипы, взрывные платформы)
4. Бонусы и усиления**:**
   * Предметы для временных эффектов:
     + Ракетный ранец (мгновенное ускорение вверх)
     + Пружина (увеличение высоты прыжка)
   * Монеты / очки для прогресса
5. Прогрессия**:**
   * Счётчик очков (зависит от высоты, на которую поднялся игрок)
   * Постепенное увеличение сложности (больше препятствий, меньше безопасных платформ)
6. Система жизней**:**
   * Если персонаж падает за пределы экрана или сталкивается с врагом, игра заканчивается

**2. ОБЗОР АНАЛОГОВ**

1. Doodle Jump - Lima Sky, 2009 год: <https://play.google.com/store/apps/details?id=com.lima.doodlejump&hl=ru&pli=1>



Рисунок 2 – Скриншот интерфейса игры «Doodle Jump»

Doodle Jump (рис. 2) - это оригинальная версия игры. Была разработана студией Lima Sky 15 марта 2009г на android телефоны. Простая, но затягивающая механика сделала игру одной из самых популярных мобильных игр своего времени. **Платформы**: iOS, Android, Windows Phone, Java (раньше).

Преимущества:

1. Простая, но затягивающая механика
2. Симпатичный и запоминающийся визуальный стиль
3. Разнообразие тем

Недостатки:

1. Однообразие игрового процесса на длинных сессиях
2. Простой ИИ врагов
3. Doodle Jump, фанатский клон, интернет версия: <https://doodle-jump.co>

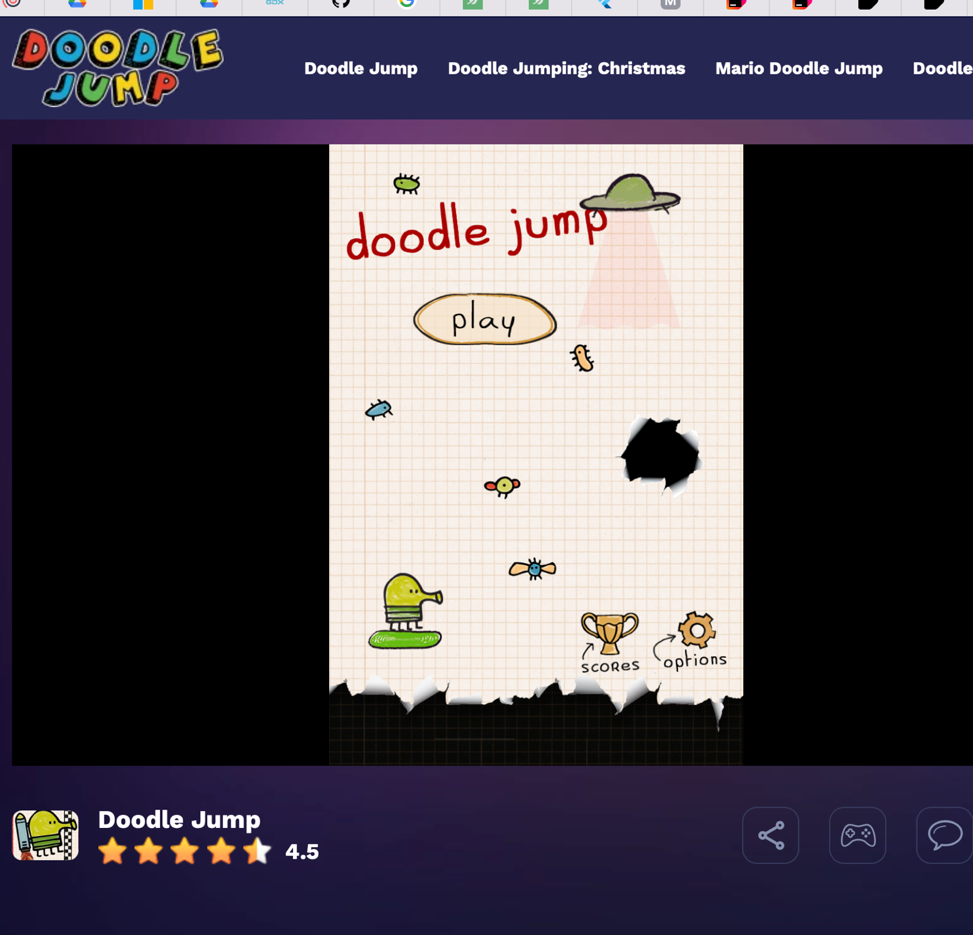


Рисунок 3 – Скриншот интерфейса интернет-версии игры Doodle Jump

Клон оригинального Doodle Jump, адаптированный для игры в браузере (рис. 3). Геймплей идентичен оригиналу: игрок прыгает по платформам, уклоняется от препятствий и врагов. **Платформы**: Браузер.

Преимущества:

1. Доступность (не требует скачивания)
2. Простая, но проверенная временем механика

**Не**достатки:

1. Менее плавная анимация по сравнению с мобильной версией
2. Возможны лаги и задержки в управлении
3. Helix Jump, Voodoo, Android/iOS, 2018: <https://helixjump2.com>

  
Рисунок 4 – Скриншот интерфейса игры Helix Jump

В отличие от Doodle Jump, в Helix Jump (рис. 4) управление не основано на прыжках персонажа, а на вращении башни, по которой падает шарик. Игрок должен направлять шарик вниз, избегая опасных зон. **Платформы**: Android, iOS.

Преимущества:

1. Оригинальная механика с вращением башни
2. Затягивающий процесс с увеличивающейся сложностью
3. Минималистичный, но приятный визуальный стиль

**Недостатки**:

1. Может быстро наскучить из-за однообразия
2. Сильно зависим от физики, что делает управление иногда непредсказуемым
3. Навязчивая реклама в мобильной версии
4. Stack Jump, Voodoo, Android/iOS, 2017: <https://yaksgames.com/games/stack-jump-voodoo/G0624686A6>

  
Рисунок 5 – Скриншот интерфейса игры Stack Jump

**В** Stack Jump **(рис.5) игрок управляет персонажем, который прыгает вверх по движущимся блокам, формируя башню. Главное — вовремя нажимать на экран, чтобы персонаж приземлялся на блоки. Платформы**: Android, iOS.

Преимущества:

1. Простой, но затягивающий геймплей
2. Разные персонажи и темы оформления
3. Хорошая физика и приятная анимация

Недостатки:

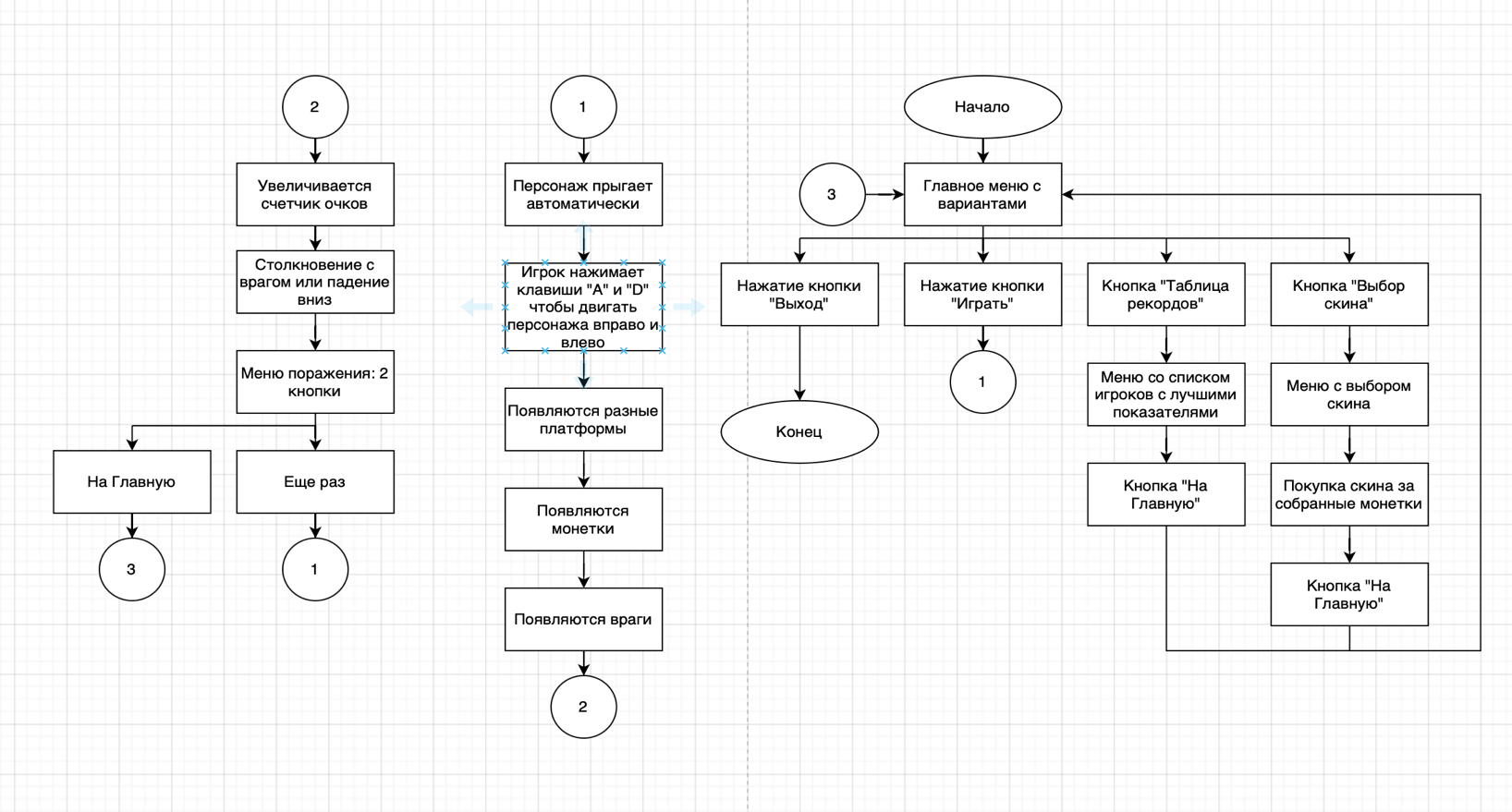
1. Однообразный геймплей, особенно на длинных сессиях
2. Большая зависимость от случайности (не всегда предсказуемые движения блоков)
3. Навязчивая реклама

**Вывод**

В результате анализа аналогов сформировалось примерное представление, видение конечного продукта, который должен содержать: меню игры, функционал движения персонажа, индикацию прогрессии, алгоритм процедурной генерации платформ, препятствий, бонусов и усилений. Необходимо также реализовать покупку внутриигровых скинов за полученные во время игры монетки.

Особое внимание требуется уделить процедурной генерации игровых объектов, т.к. именно на этом у игрока будет завязываться интерес к игре.

**3. АЛГОРИТМЫ ИСПОЛЬЗОВАНИЯ ПРОГРАММЫ**

  
Рисунок 6 – Алгоритм использования программы в виде Блок-Схемы

Запуск игры – при открытии приложения отображается заставочный экран с логотипом игры, после чего пользователь попадает в главное меню. В главном меню доступны кнопки: «Играть», «Настройки», «Магазин», «Таблица рекордов» и «Выход» (на ПК).

Начало игры – после нажатия кнопки «Играть» загружается игровой уровень, где персонаж автоматически начинает прыгать вверх. Управление осуществляется наклоном устройства или клавишами (в зависимости от платформы).

Игровой процесс – на пути игрока встречаются различные платформы: стандартные, разрушающиеся, движущиеся и другие. Также могут появляться бонусы, такие как ускорители, щиты или двойные прыжки. Враги представляют собой препятствия, которые могут двигаться, атаковать или появляться в неожиданных местах.

Проигрыш – если игрок падает за границы экрана или сталкивается с врагом без защиты, игра завершается. После этого появляется экран с результатами, на котором отображаются текущий рекорд, очки за попытку и кнопки «Играть снова», «В меню» и «Поделиться».

Прогресс и награды – игрок может зарабатывать очки или монеты, которые тратятся в магазине на новые скины, способности или бонусы. Также могут разблокироваться достижения за выполнение определенных условий.

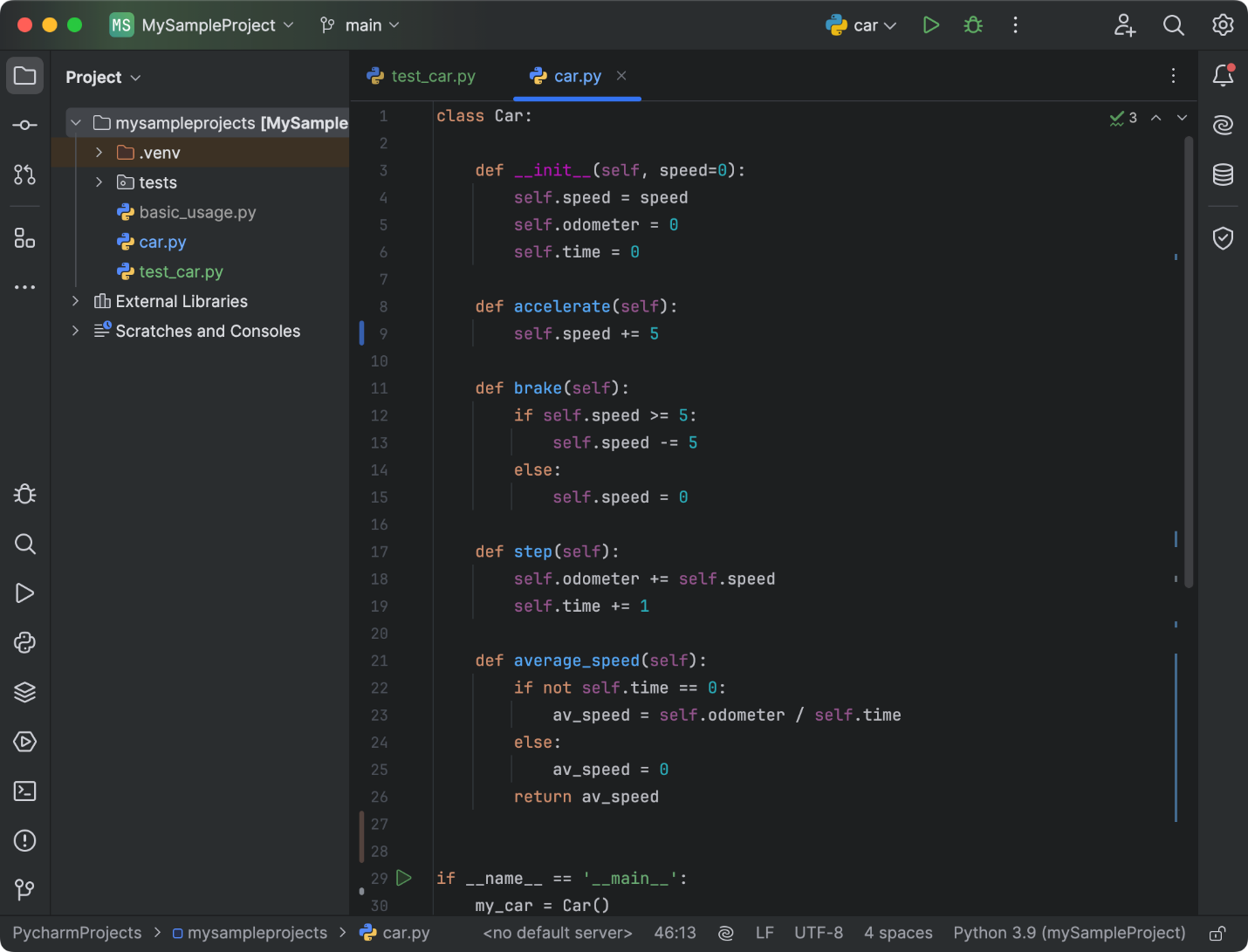
Выход из игры – при закрытии приложения прогресс автоматически сохраняется. При следующем запуске пользователь может продолжить игру или начать новую попытку.

**4. ВЫБОР ЯЗЫКА ПРОГРАММИРОВАНИЯ И НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ**

Python - это интерпретируемый, интерактивный, объектно-ориентированный язык программирования высокого уровня, созданный Гвидо ван Россумом в начале 1990-х годов. Он назван в честь телепередачи BBC «Летающий цирк Монти Пайтона». Его часто сравнивают с языками программирования Tcl, Perl, Scheme, Java и Ruby. Один из самых популярных языков программирования в мире. Python не требует явного объявления переменных, является регистро-зависим объектно-ориентированным языком. Он широко применяется в разработке веб-приложений и прикладного программного обеспечения, а также в машинном обучении и обработке больших данных.

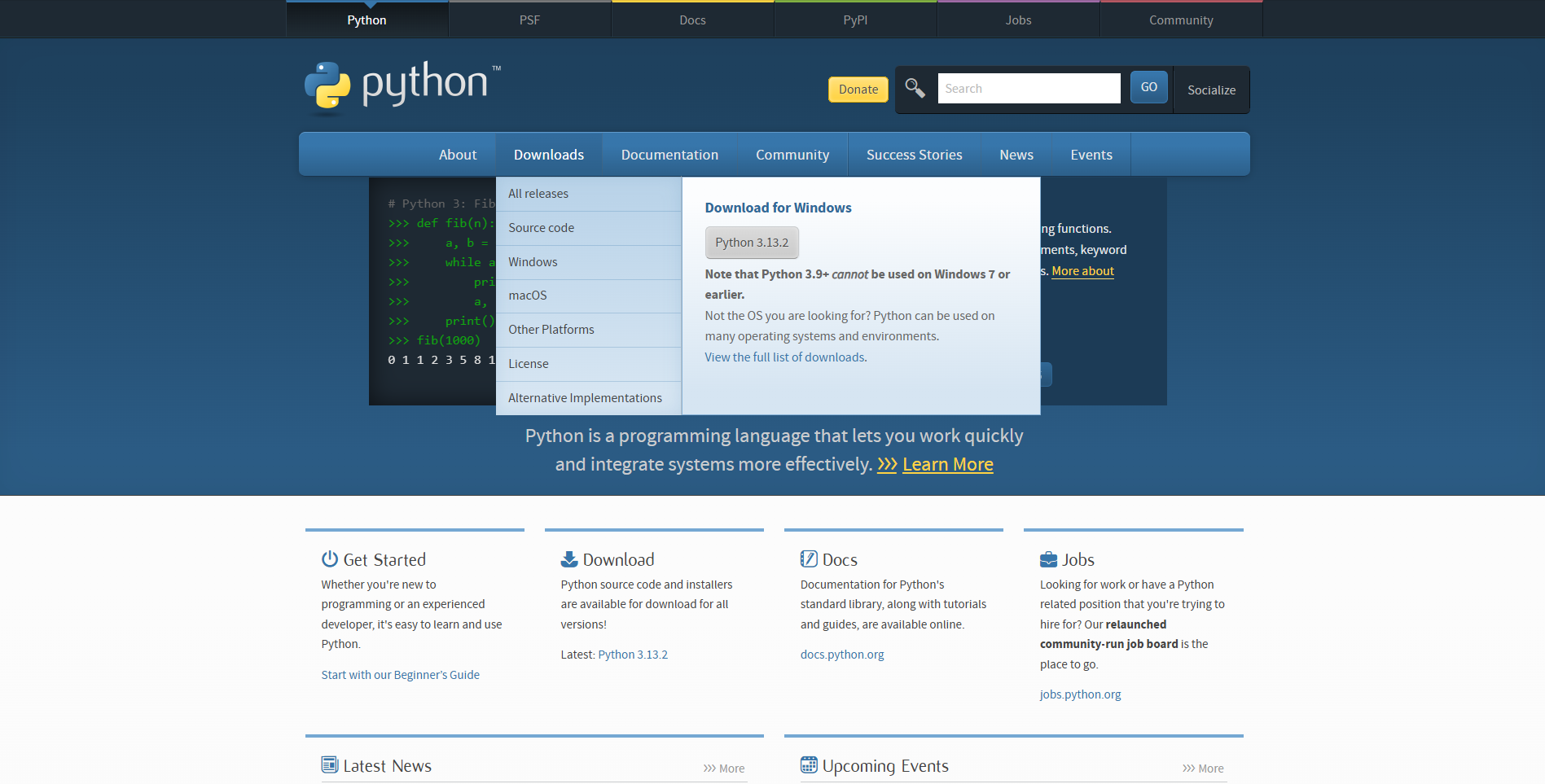
PyCharm — это мощная интегрированная среда разработки (IDE) для языка Python, разработанная компанией JetBrains. Она обеспечивает удобную и продуктивную работу за счёт интеллектуальной поддержки кода, автоматического дополнения, отладки и встроенных инструментов тестирования. PyCharm поддерживает различные парадигмы программирования, включая объектно-ориентированное, процедурное и функциональное программирование, что делает его универсальным инструментом для разработки.

PyCharm включает визуальную надстройку над некоторыми командами (для работы с GitHub, инициализация проекта), мощный отладчик, профайлер и поддержку удалённой разработки, что делает его отличным выбором как для начинающих, так и для опытных программистов. Благодаря продвинутому анализу кода и рефакторингу, PyCharm помогает писать чистый и эффективный код, минимизируя ошибки и повышая продуктивность работы.

  
Рисунок 7 – Интерфейс среды разработки PyCharm

В разработке игры было принято решение использовать именно язык программирования Python, потому что он обеспечивает быстрый и очень простой процесс разработки. В отличие от Pascal. Данный язык программирования также имеет большой арсенал различных библиотек, что может быть очень удобным при разработке.

Для разработки приложения потребуется установить на персональный компьютер некоторое программное обеспечение. Далее рассмотрим каждое из них. Интерпретатор языка Python, необходимый для работы интерпретируемого языка. Данное ПО устанавливается с официального сайта - <https://www.python.org/> (рис. 8). Необходимо навести курсор на вкладку «Downloads» и нажать кнопку «Python 3.13.2» - здесь указана актуальная версия на 13.02.2025.

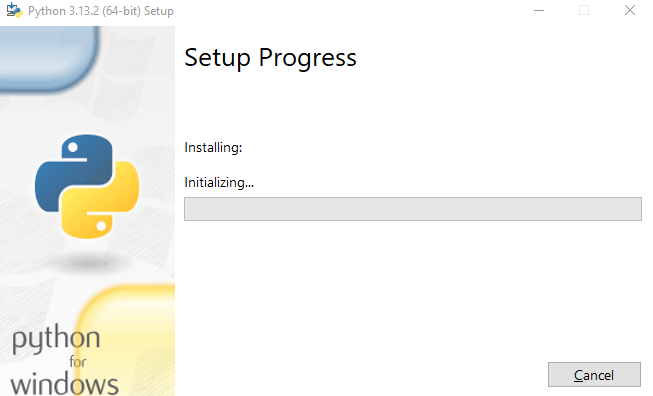
  
Рисунок 8 – Офф. сайт Python

После нажатия кнопки файл установщика сохраниться в папке «Загрузки» на компьютере. Открыв этот установщик, нас встречает экран установки Python (рис. 9)

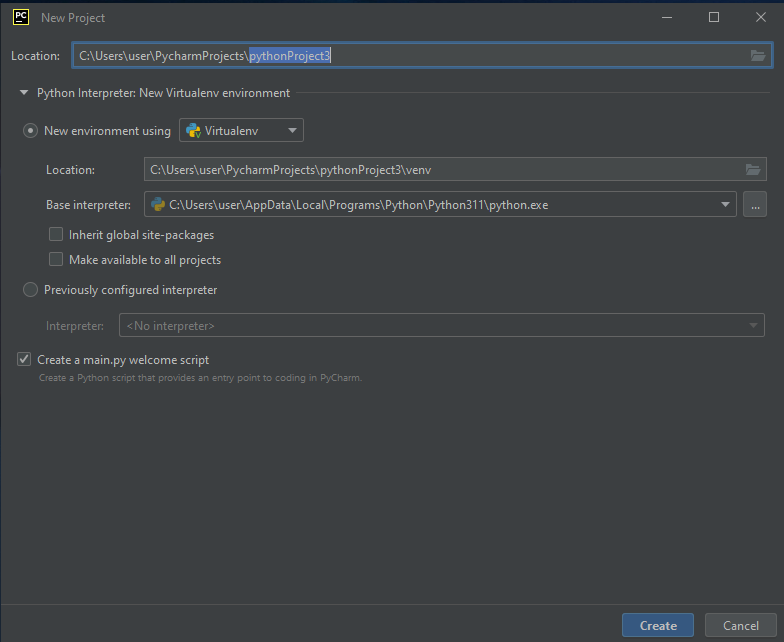
Изображение выглядит как текст, снимок экрана, Операционная система, веб-страница

Автоматически созданное описание  
Рисунок 9 – Экран установки Python

Кнопка «Install Now» - установить с рекомендуемыми настройками, без лишних действий, а «Customize installation» - установка с кастомизацией некоторых настроек. Для большинства подойдет вариант с установкой по-умолчанию. Далее нас встречает процесс установки (рис. 10).

  
Рисунок 10 – Процесс установки Python

Далее необходимо создать новый проект в PyCharm. После выбора «New Project» всплывет диалоговое окно, в котором можно необходимо указать название проекта, папку расположения проекта, а также дополнительную настройку виртуального окружения. Виртуальное окружение (virtual environment) в Python – это инструмент, позволяющий изолировать зависимости проекта от глобального окружения и предотвращать возможные конфликты между библиотеками. Настройка виртуального окружения потребует указать версию интерпретатора, который был установлен ранее.

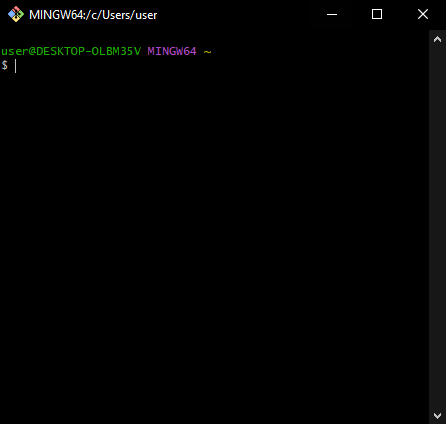
  
Рисунок 11 – Создание нового Python проекта

Для удобной разработки, управлением версиями при разработке, а также для возможности работать над этим проектом с любого компьютере: дома за компьютером или в любом другом месте за ноутбуком потребуется установить систему контроля версий Git. Git - Git - это консольная утилита, для отслеживания и ведения истории изменения файлов, в проекте. Чаще всего используют для кода, но можно и для других файлов. С помощью Git-a можно откатить свой проект до более старой версии, сравнивать, анализировать или загружать свои изменения в репозиторий.

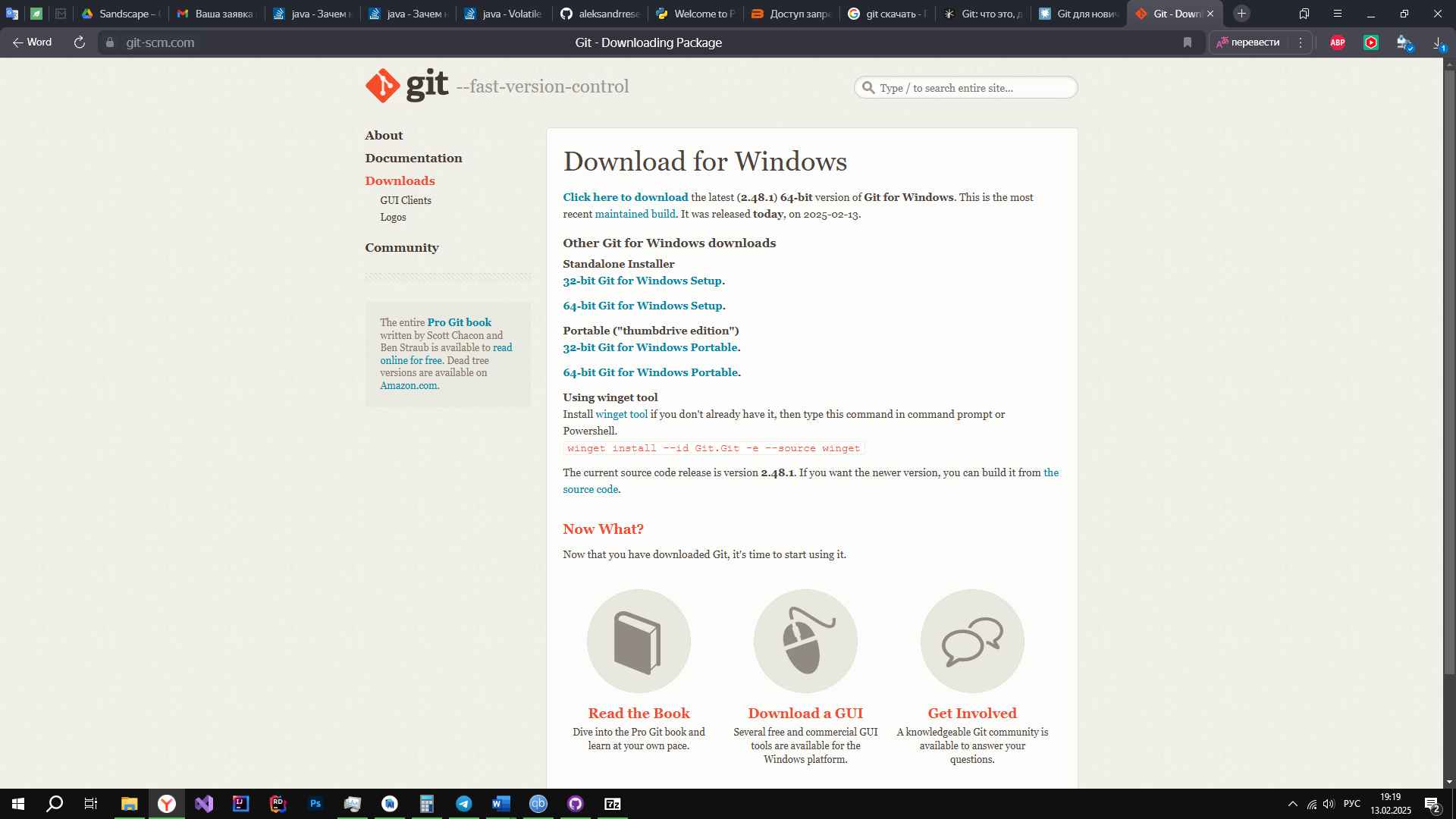
Репозиторием называют хранилище вашего кода и историю его изменений. Git работает локально и все репозитории хранятся в определенных папках на жестком диске. Так же ваши репозитории можно хранить и в интернете, для этого можно воспользоваться сервисом GitHub.

Каждая точка сохранения проекта носит название коммит (commit). У каждого commit-a есть hash (уникальный id) и комментарий. Из таких commit-ов собирается ветка. Ветка – это история изменений. У каждой ветки есть свое название. Репозиторий может содержать в себе несколько веток, которые создаются из других веток или вливаются в них.

Работать с Git и его командами необходимо из специального терминала – Gitbash (Рис. 12) или же из терминала, установленного в среде разработки.

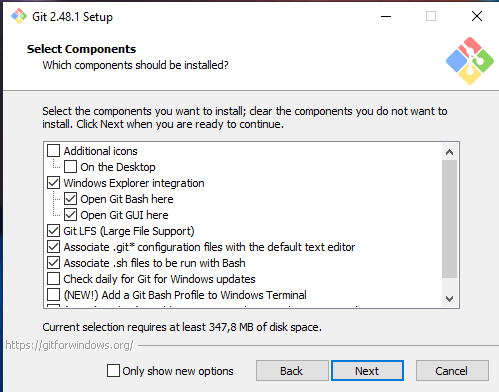
  
Рисунок 12 – Интерфейс GitBash

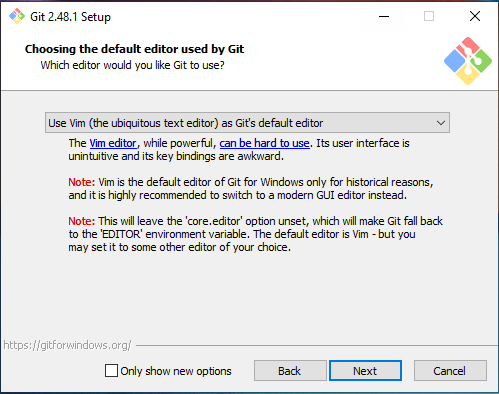
Теперь можно приступить к установке системы контроля версий. Git устанавливается с официального сайта - <https://git-scm.com/downloads/win> (Рис. 13). На сайте необходимо нажать кнопку скачивания с выбором ОС, установленной на компьютере. После этого начнется скачивание установочного файла в папку «Загрузки» на вашем компьютере.

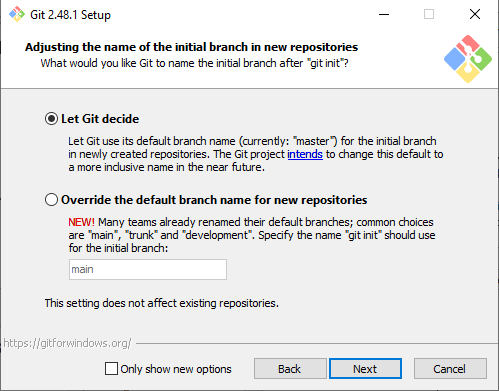
  
Рисунок 13 – Страница загрузки на офф. сайте Git

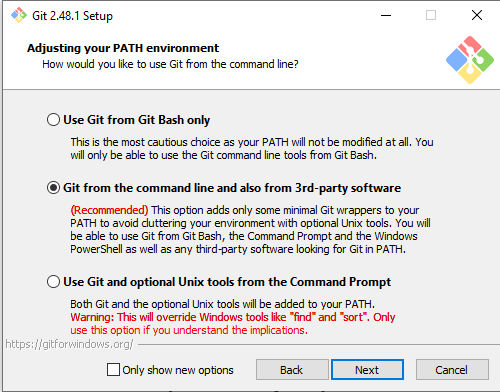
После запуска программы установщика Git (рис. 14) можно выбрать выбор всех параметров установки по умолчанию (рис 15 – 25).

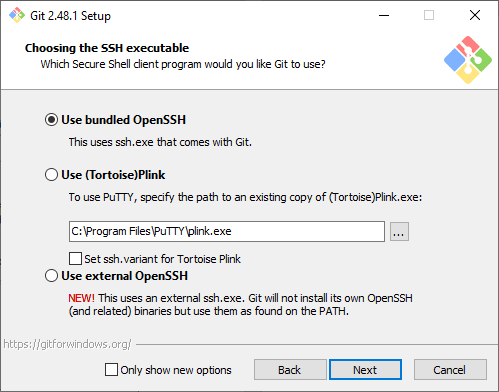
  
Рисунок 14 – Установка Git

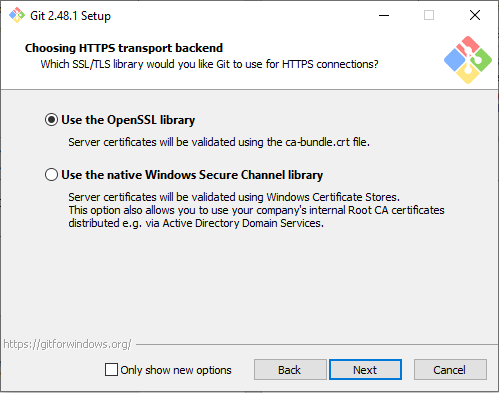
  
Рисунок 15 – Настройки установки Git

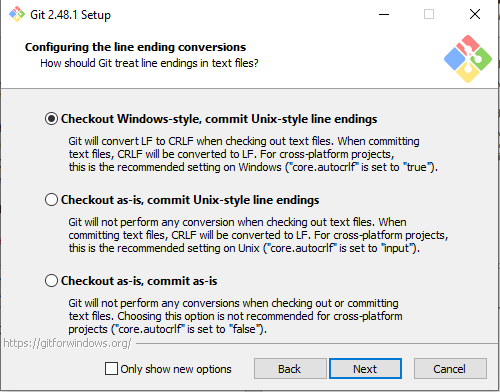
  
Рисунок 16 – Настройки установки Git

  
Рисунок 17 – Настройки установки Git

  
Рисунок 18 – Настройки установки Git

  
Рисунок 19 – Настройки установки Git

  
Рисунок 20 – Настройки установки Git

  
Рисунок 21 – Настройки установки Git

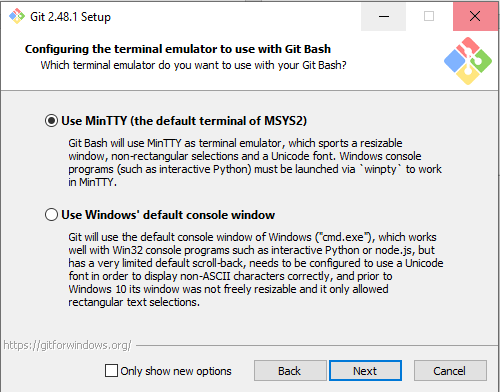
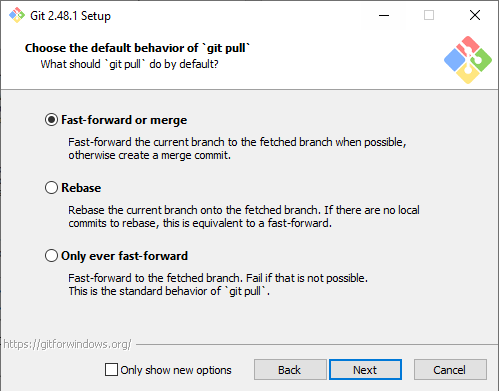


Рисунок 22 – Настройки установки Git

  
Рисунок 23 – Настройки установки Git

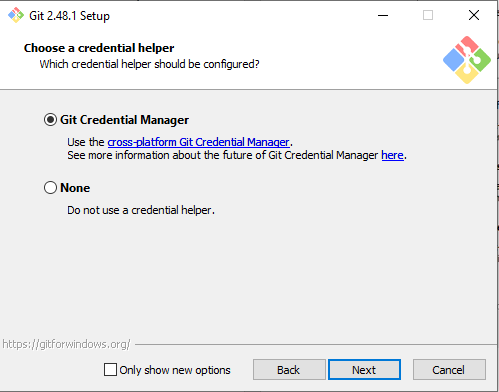
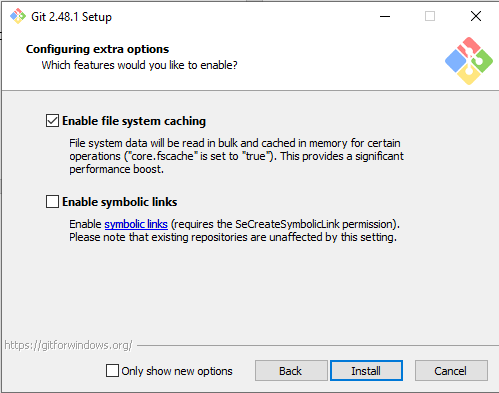
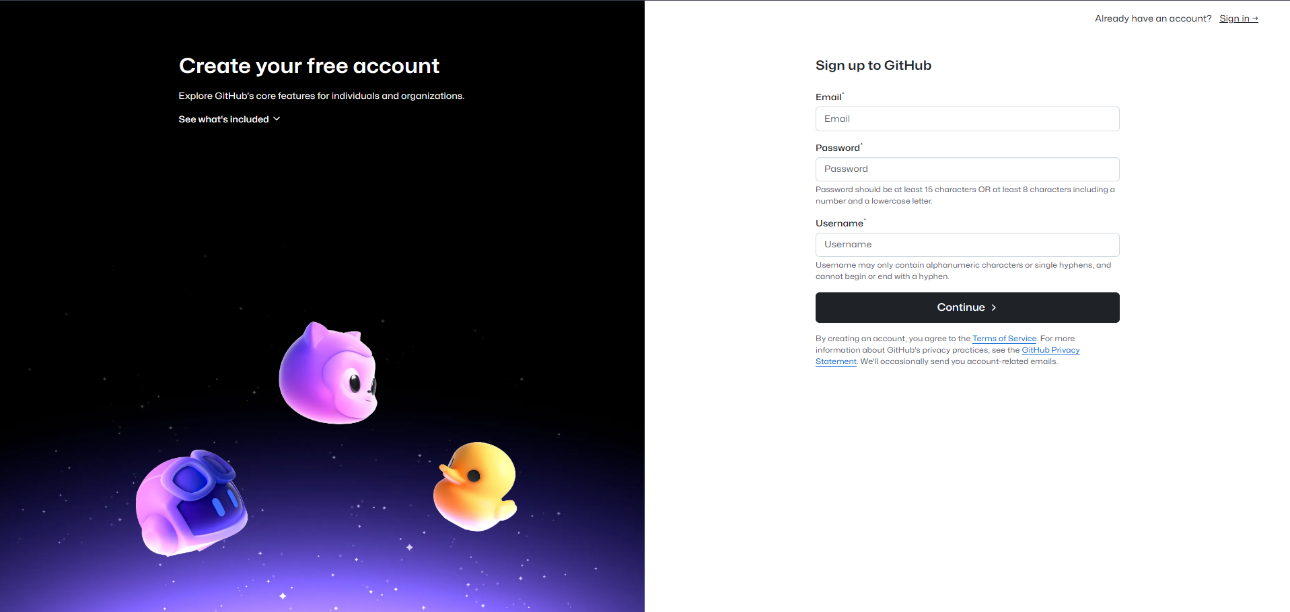


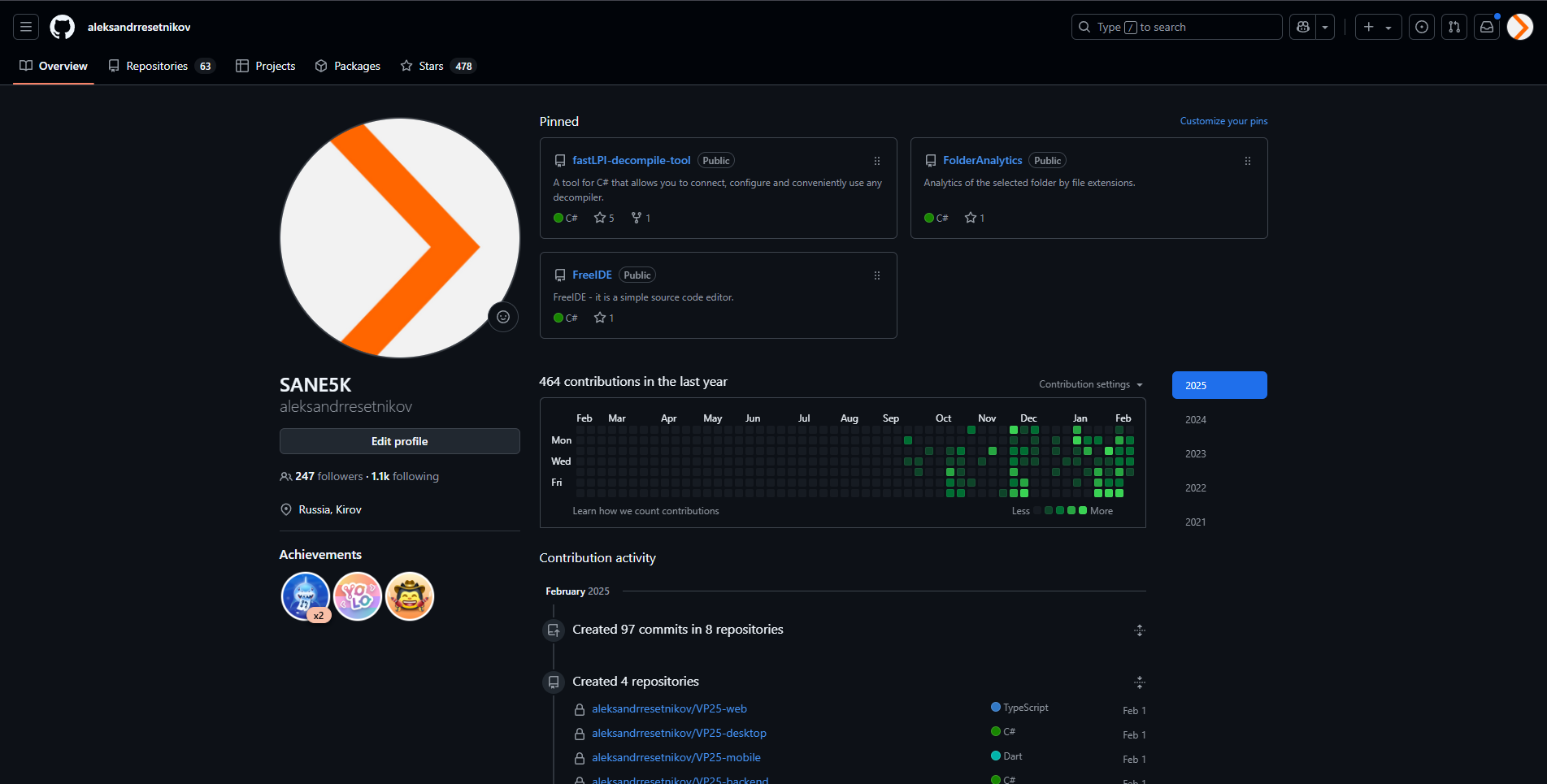
Рисунок 24 – Настройки установки Git

  
Рисунок 25 – Настройки установки Git

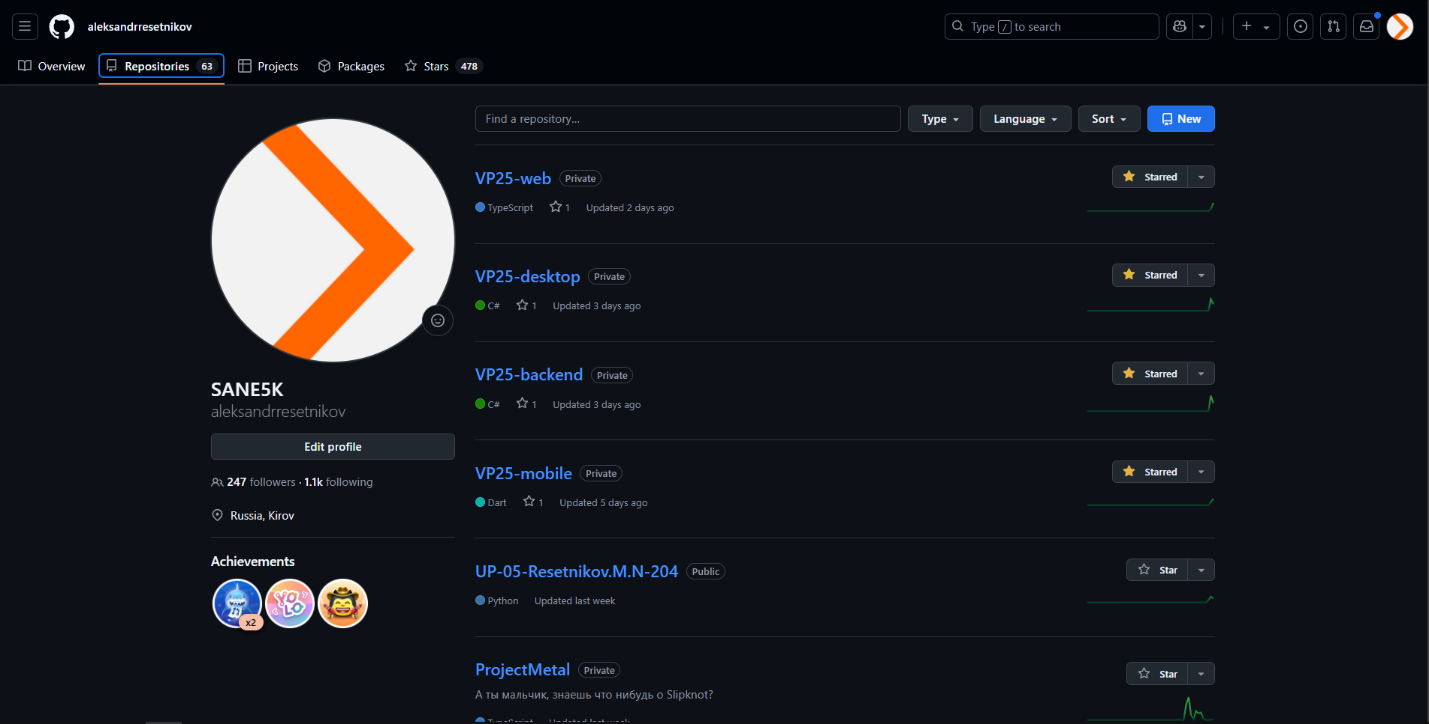
Как уже было упомянуто ранее – в ходе работы над проектом будет использоваться онлайн-сервис GitHub – облачная платформа, упрощающая работу с Git, предоставляющая удобный веб-интерфейс, инструменты для командной работы, систему управления задачами и многое другое. Перед началом работы с Github необходимо зарегистрироваться на официальном сайте - <https://github.com/signup> (Рис. 26).

  
Рисунок 26 – Регистрация на офф. сайте GitHub

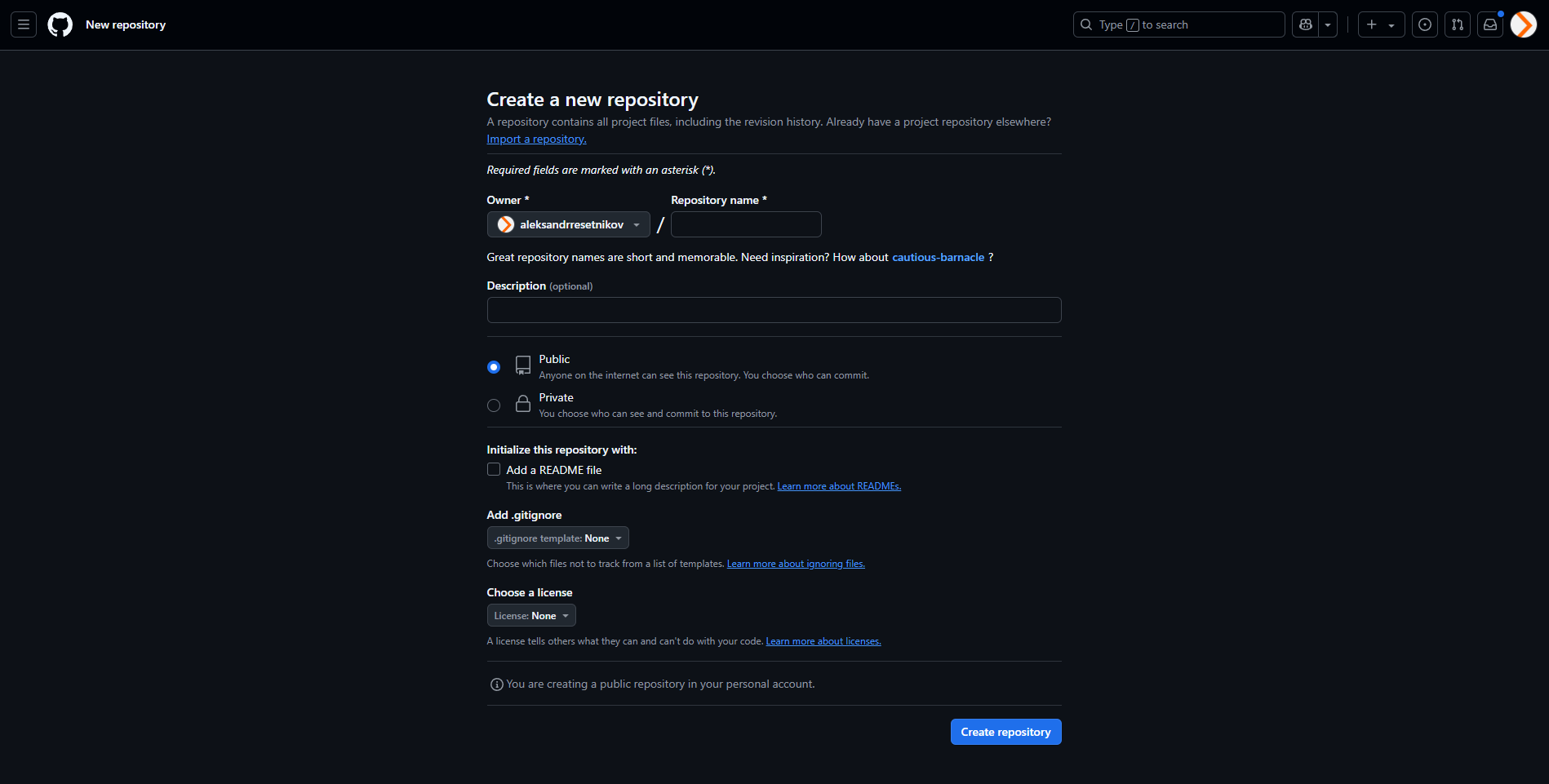
После регистрации на сайте GitHub, пользователя встречает страница профиля (рис. 27), на которой отображено: пользовательское имя, никнейм, аватар, список «ачивок», закрепленные репозитории, график активности пользователя.

  
Рисунок 27 – Страница пользовательского профиля

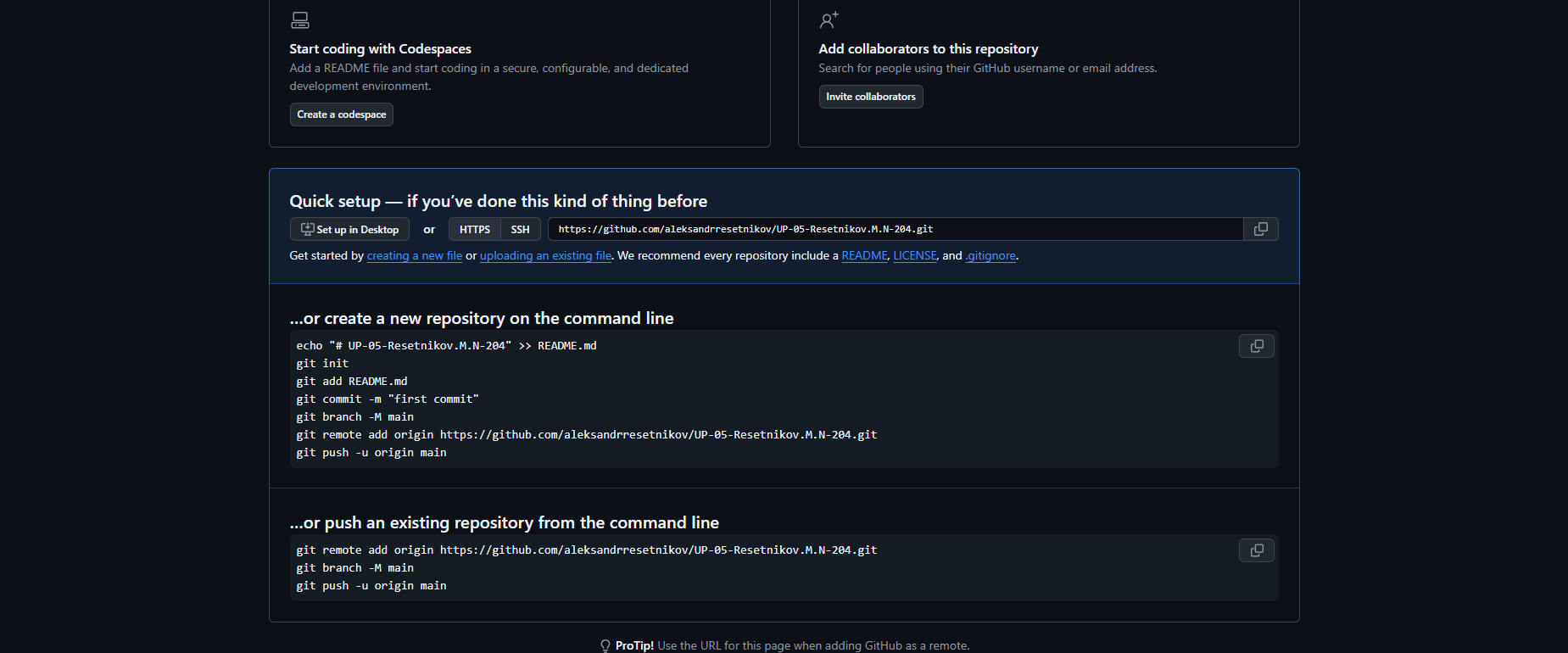
Для создания нового репозитория необходимо перейти во вкладку с репозиториями «Repositories» (рис. 28).

  
Рисунок 28 – Вкладка репозиториев в профиле

Далее необходимо нажать кнопку «New» - для создания нового репозитория, после чего пользователя встречает вкладка (рис. 29), где необходимо указать владельца нового репозитория, название репозитория, уровень доступа репозитория (приватный или открытый).

  
Рисунок 29 – Создание нового репозитория

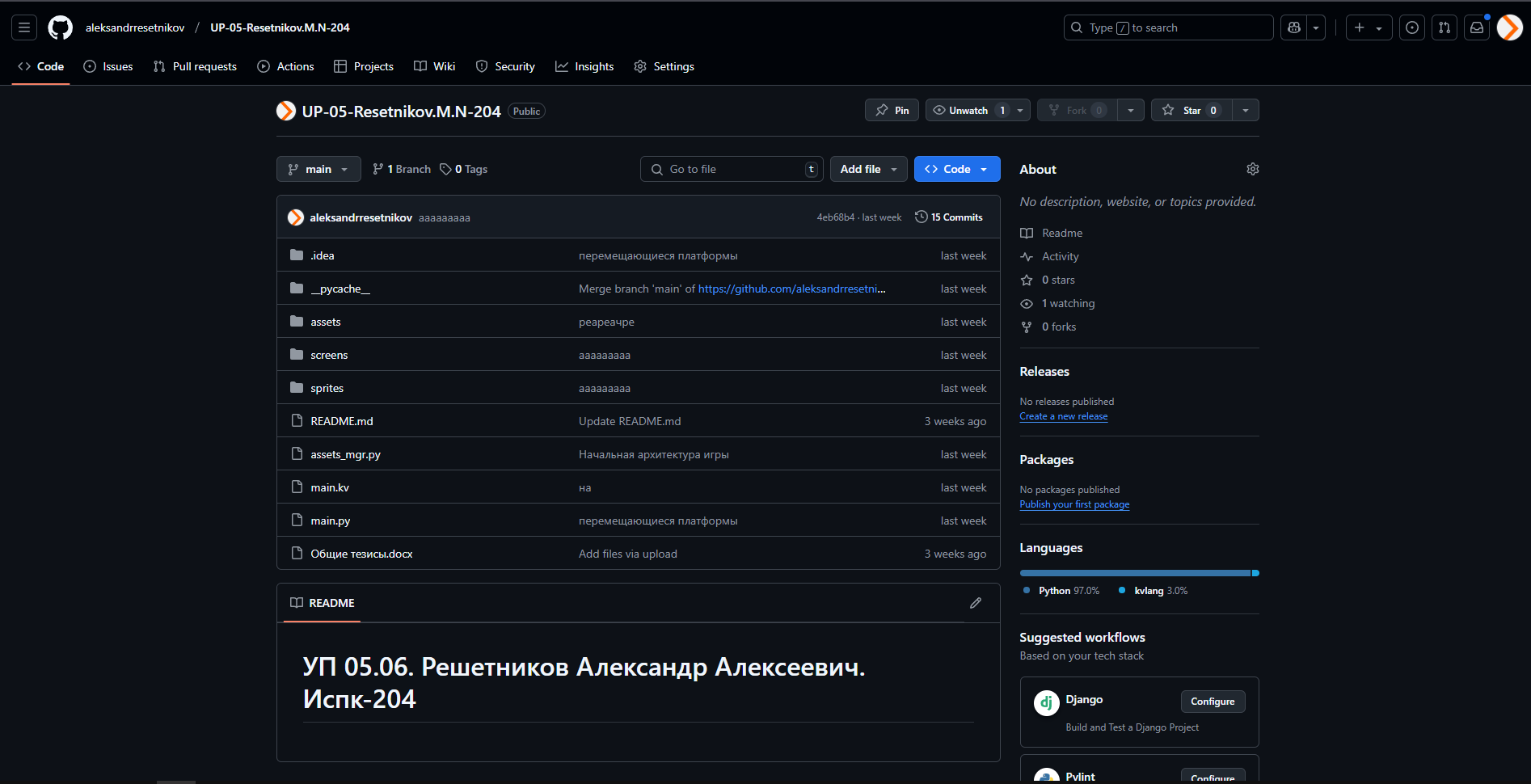
После создания нового репозитория откроется вкладка с подробной инструкцией (рис. 30), как привязать свой локальный репозиторий к удаленному онлайн-репозиторию.

  
Рисунок 30 – Инструкция подключения к удаленному репозиторию

Чтобы подключить локальный репозиторий к удаленному, необходимо выполнить несколько команд в терминале среды разработки:

1. git init – инициализация локального репозитория.
2. git add . – добавление всех изменений в список подготовленных к коммиту.
3. git commit -m "сообщение" – создание коммита, фиксирующего изменения с возможностью оставить поясняющее сообщение. В дальнейшем это сообщение будет отображаться в GitHub.
4. git branch -M main – создание (или переименование) основной ветки main.
5. git remote add origin <ссылка на удаленный репозиторий> – привязка локального репозитория к удаленному.
6. git push -u origin main – отправка изменений из локального репозитория в удаленный в ветку main.

После выполнения этих команд структура проекта и созданный коммит появятся в удаленном репозитории на GitHub (Рис. 31).

  
Рисунок 31 – Связанный локальный репозиторий с удаленным

В проекте было принято решение использовать Kivy в качестве библиотеки для отрисовки графики игры. Это мощная, простая и гибкая библиотека для разработки кроссплатформенных приложений на языке программирования Python. Она позволяет создавать приложения, которые могут работать на различных операционных системах, таких как Windows, macOS, Linux, iOS и Android. Установка Kivy осуществляется следующим образом: в терминале проекта необходимо ввести команду «pip install kivy».

**Вывод**

В ходе работы над созданием рабочего окружения был выбран язык программирования Python за счет его удобства и скорости написания кода на нем. В качестве среды разработки была выбрана программа PyCharm. В качестве графической библиотеки была выбрана Kivy за счет ее простоты, многофункциональности и кроссплатформенности. Также был настроен удалённый репозиторий на GitHub для хранения и управления кодом проекта, а локальный репозиторий был подключён к нему, что позволяет эффективно отслеживать изменения, работать в команде и обеспечивать безопасность данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Kivy.org: офф. сайт с документацией по kivy. URL: <https://kivy.org/doc/stable/gettingstarted/installation.html>
2. Skypro: онлайн-университет Skypro от Skyeng для обучения и развития профессиональных навыков в сфере IT. URL: <https://sky.pro/wiki/python/sozdanie-prostogo-prilozheniya-na-kivy/>
3. Яндекс практикум: Яндекс Практикум — сервис онлайн-образования. URL: <https://practicum.yandex.ru/blog/chto-takoe-git-i-dlya-chego-nuzhen/>
4. Habr: русскоязычный веб-сайт в формате системы тематических коллективных блогов. URL: <https://habr.com/ru/articles/541258/>
5. Git: офф. сайт с документацией и загрузкой системы Git. URL: <https://git-scm.com/downloads/win>
6. РОССТАТ : офиц. сайт Москва, Федеральная служба государственной статистики. URL: <https://rosstat.gov.ru/folder/313/document/63640>