**Описание языка программирования**

**Python** — **высокоуровневый язык программирования, отличающийся эффективностью, простотой и универсальностью использования**. Он широко применяется в разработке веб-приложений и прикладного программного обеспечения, а также в машинном обучении и обработке больших данных.

**Модули, использованные для написания программы**

tkinter - это библиотека, предоставляющая интерфейс для создания графического пользовательского интерфейса (GUI) в Python. С помощью Tkinter можно создавать окна, кнопки, ползунки, текстовые поля и другие виджеты, которые позволяют взаимодействовать с пользователем. Tkinter предоставляет простые и интуитивно понятные методы для управления виджетами и обработки событий.

**Pydub** — это **библиотека на Python для работы с аудиофайлами**. Она поддерживает различные форматы и позволяет выполнять множество операций, таких как обрезка, склеивание и изменение громкости.

В процессе работы все файлы загружались на Github в созданный в нем репозиторий.

Github — **веб-платформа для совместной использования и разработки кода программ**. Она даёт инструменты, которые помогают программистам работать над проектами вместе в режиме реального времени и отслеживать изменения.

**Репозиторий** — папка с проектом, где лежат файлы, иконки программы, разные картинки. В Github у каждого репозитория есть страница с описанием проекта.

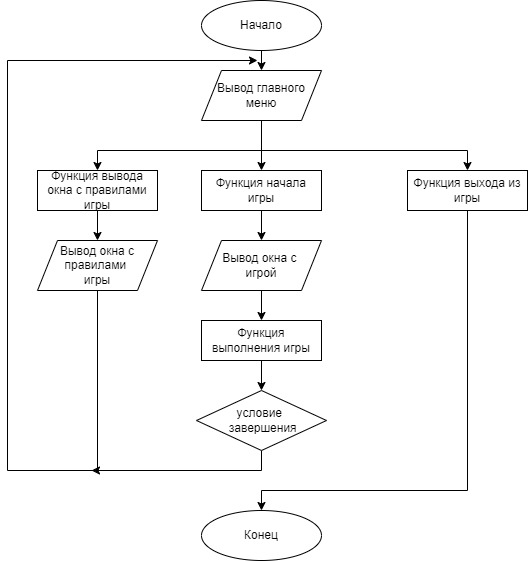
**Ветка** (Branch) — копия проекта в рамках одного репозитория. Есть главная ветка проекта, или main. Но любой разработчик может скопировать проект в свою ветку и работать над его частью, не трогая исходный код и не мешая другим разработчикам. Ветки независимы друг от друга, но их можно объединять, мёржить (от англ. merge — слияние), даже если есть разница в коде.

Для создания схем алгоритмов использовалось приложение draw.io.

**Draw.io** — это **приложение, которое позволяет создавать и редактировать диаграммы, различные схемы и прочие визуальные представления данных**. Также он доступен онлайн.

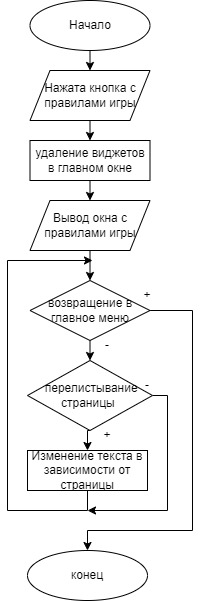
**Описание алгоритма главного меню**

1. Создаётся главное меню: задаётся размер окна, фон, его название.
2. В открытом окне располагаются 3 кнопки: начать, правила игры, правила игры, выход, выполняющая закрытие окна.



**Описание алгоритма меню с правилами игры**

1. Открывается окно, на котором прописаны правила игры.
2. На окне располагается 3 кнопки: перелистнуть вправо, перелистнуть в лево, вернуться в главное меню.



**Описание алгоритма выполнение игры.**

1. Открывается окно игры.
2. Задаётся начальное здоровье персонажа и его уровень.
3. Запуск таймера.
4. Сравнивается прожитое время в игре с временем победы. Если время игры равно с временем победы то останавливается таймер и выводится окно о выигрыше, очищается экран и возвращает пользователя в главное меню. Если время не равно, то сравнивается здоровье игрока.
5. Сравнивается здоровье игрока с минимальным. Если здоровье меньше минимального то останавливается таймер выводится окно о проигрыше, очищается экран и возвращает пользователя в главное меню. Если время не равно, то сравнивается здоровье игрока.
6. Количество противников на карте. Если их меньше максимального то по прожитому времени определяется появление противников с 0 сек. Появляется «монстр1» с 1:30 мин. появляется «монстр2» с 3:00 мин. появляется «монстр3» и сравнивается текущий уровень с максимальным. Если их больше чем максимальное то сравнивается текущий уровень с максимальным.
7. Сравнивается текущий уровень с максимальным. Если их меньше максимального то проверяется количество опыта для получения нового уровня с требуемым: если равен требуемому то выполняется функция выбора новой способности; если меньше то ничего не происходит. Если текущий уровень равен максимальному то ничего не происходит.
8. Выполняется поиск ближайшего противника.
9. Проверяется наличие способностей. Если способность есть, то проверяется наличие перезарядки у способности, если нет перезарядки у способности то наносится урон противнику; если есть перезарядка у способности или этой способности нет то проверяется здоровье противников.
10. Проверка здоровья противников. Если противник имеет меньше 1 здоровья то он исчезает игроку даётся опыт, если у него здоровье больше чем 1 то он перемещается в сторону игрока.
11. Проверяется наличие противников около игрока. Если около игрока есть противник то игроку наносится урон. Если нет то ничего не происходит.
12. Выполняется отображение всех изменений на экране и сравнивается прожитое время.

