**Государственное бюджетное общеобразовательное учреждение города Москвы “Образовательный центр Протон”**

**“WEB-приложение:**

**Мониторинг посещаемости учеников школы ”**

Участник:

Ученик 10 “Т” класса

Школы ГБОУ ОЦ “Протон”

Мищенко Дмитрий Юльевич

Руководитель:

Педагог ГБОУ ОЦ “Протон”

Лукичев Евгений Сергеевич

**Москва, 2025**

**Оглавление:**

1. [Введение](#Введение)
   1. Актуальность работы
   2. Обоснование выбора темы
   3. Постановка проблемы
2. [Цель и задачи работы](#Цель_и_задачи)
   1. Цель
   2. Задачи
3. [Обоснование используемых программных средств](#Исп_материалы)
   1. Язык
   2. Библиотеки
   3. Программы
4. [Методика выполнения работы](#Методика)
5. [Результаты работы и их проверка](#Результаты_работы)
6. [Описание продукта и перспективы развития](#Описание_продукта)
   1. Основная концепция
   2. Основные механики
   3. Дополнительные элементы
7. [Инструкция к игре](#Инструкция_к_игре)
   1. Управление
   2. Механика ввода и вывода
   3. Механика конвейеров
   4. Механика рецептов
8. [Ссылки](#Ссылки)
9. [Список литературы](#Список_литературы)

**Введение:**

1. Актуальность работы:

В школах ученикам, учителям, нужно больше информации о посещении учеников в школу. Такой как подсчёт опозданий учеников, посещение учеников определенных предметов, возможность сравнения этих данных между друг другом.

1. Обоснование выбора темы:

Я выбрал данную тему из-за того что у моей школы есть потребность в данном проекте, аналоги моего проекта предоставляют не настолько подробную информацию или предназначены не для большой аудитории.

1. Постановка проблемы:

Потребность школ в более понятной, упрощённой, и разнообразной информации о посещении учеников в учреждение.

**Цель и задачи работы:**

1. Цель:

Разработать web-приложение, которое предоставляет членам учреждения подробную и понятную информации о посещаемости, опозданиях школьников.

1. Задачи:
   1. Изучить основы разработки web-приложений.
   2. Изучить функционал библиотек, который понадобится при реализации проекта.
   3. Продумать какие данные будут предоставляться, стиль приложения и главную идею проекта.
   4. Нарисовать текстуры и спрайты для игры.
   5. Разработать код видеоигры.
   6. Протестировать web-приложения для отладки ошибок.
   7. Применение на практике и сдача проекта.

**Обоснование используемых программных средств:**

Язык:

Для создания проекта использовался язык Python. У него есть несколько преимуществ и недостатков:

Преимущества:

1. Лёгкость освоения, простота синтаксиса и динамическая типизация.
2. Большая популярность и обилие обучающих материалов в интернете. Столкнувшись с какой-то проблемой, можно быстро найти решение в документациях или на форумах программистов.
3. Кроссплатформенность. Программу можно запустить на любой ОС. Главное, чтобы был установлен актуальный Python. Есть возможность собрать проект в исполняемый файл, и тогда будет не обязательно наличие Python на компьютере.

Недостатки:

1. Медленность языка. Python – интерпретируемый язык. Это означает, что код выполняется по одной команде за раз. Также динамическая типизация, которая является очень удобной, снижает производительность.

Для написания web-приложения я использовал фреймворк Flask

Преимущества:

1. Лёгкость. Flask достаточно легкий и простой фреймворк для реализации небольших проектов.
2. Обширная документация и большое сообщество.

Библиотеку языка python openpyxl для чтения и работы с данными из файлов с расширением xlsx.

Библиотеку языка python datetime для работы с датами и временем

Библиотеку языка python SQLAlchemy для работы с базой данных:

SQLAlchemy – это библиотека для работами с реалиционными базами данных на языке python. SQLAlchemy позволяет при работе с базами данных, вместо SQL-запросов использовать методы удобные разработчику.

Программы:

Visual code studio – приложение предоставляющее большой функционал благодаря своим расширениям, на нем удобно разрабатывать проект состоящий из разных форматов файлов.

**Методика выполнения работы:**

1. Изучение основы разработки web-приложений на Flask: (сентябрь - октябрь 2024)
2. Изучение необходимых частей библиотек SQLAlchemy, datetime и openpyxl через их документацию и различные видеоматериалы: (октябрь – ноябрь 2024)
3. Составление основной функциональности приложения: (ноябрь – декабрь 2024)
4. Написание web-приложения: (январь – февраль 2025)
5. Тестирование, выявление багов и ошибок: (февраль 2025)
6. Сдача проекта: (февраль 2025)

**Результаты работы и их проверка:**

Как итоговый продукт была получена рабочая видеоигра. Через запуск продукта на разных устройствах было выявлено, что на ОС Windows игра работает наиболее стабильно, также были выявлены рекомендуемые системные требования:

1. CPU – AMD Ryzen 5 3600
2. GPU – Nvidia GeForce RTX 2060
3. RAM – 4 GB
4. Disk memory – 50 MB

Высокие системные требования связаны с особенностями обработки графики и логики игры на Python, а также недостаточной оптимизацией на данном этапе.

**Описание продукта и перспективы развития:**

Описание конечного продукта:

1. Основная концепция.

Factory Zone – песочница в жанре автоматизации, в которой игроку предстоит создание большого завода. Игрок появляется в пустом мире с открытыми жилами ресурсов. Цель игрока – собрать полное хранилище всех ресурсов (2048 штук каждого предмета), а для этого нужно построить хитрую и оптимизированную цепь производств.

1. Основные механики:
   1. Перемещение. Игрок может свободно перемещаться по карте. И даже за картой.
   2. Строительство. Игрок может поставить любую постройку в любой точке на карте и так же её убрать.
   3. Конвейеры. Предметы можно удобно перемещать при помощи конвейеров.
   4. Окна. При нажатии на интерактивное строение открывается окно взаимодействия. С его помощью можно посмотреть информацию о строении и изменить его параметры.
   5. Входы и выходы. Загружать и выгружать предметы в строения можно только при помощи коннекторов ввода и вывода.
2. Дополнительные элементы:
   1. Система сохранений, основанная на [PyYAML](#yayaml). Игрок может сохранить свою карту, а при желании очистить её (изменения вступают в силу после перезагрузки).

Структура файла сохранения. На графе 1 конечными точками показаны объекты, данные которых нужно сохранить.

1. Изображение выглядит как снимок экрана, диаграмма, линия, дизайн

   Контент, сгенерированный ИИ, может содержать ошибки.Ground cells и Auxiliary cells – это клетки поверхности и вспомогательные клетки. Они имеют одинаковые по структуре данные:

Граф 1

[“layer\_type”, “cell\_type”, “direction”, “position”]

1. Building cells – клетка второго слоя, слоя построек (строений). Она имеет следующую структуру данных:

[“layer\_type”, “cell\_type”, “direction”, “position”, “cell\_inventory”, “selected\_recipe”]

1. Player – данные о игроке, хранилищах и камере.

[“type”, “position”, “camera\_offset”, “storage\_inventory”]

Перспективы развития:

1. Выпуск обновлений и насыщение игры контентом. Новые механики, предметы, строения и логистические элементы.
2. Сформировать более интересную цель для игрока и реализовать возможность прохождения игры.
3. Улучшение оптимизации и стабильности.
4. Портирование игры на Linux и стабилизация порта на MacOS.

Движок игры сделан с расчётом на будущее. Его структура позволяет, не меняя кода основных механик, добавлять новые предметы, руды, блоки и строения. Поэтому дальнейшая разработка будет значительно проще.

**Инструкция к игре:**

1. Управление:
   1. Перемещение: “W” – вверх, “A” - влево, “S” - вниз, “D” – вправо.
   2. Выбор строения: “ЛКМ” пустым курсором по иконке строения в хотбаре.
   3. Отмена выбора строения / закрытие окна взаимодействия: “Escape”.
   4. Вращение выбранного строения: “R”.
   5. Установка строения: “ЛКМ”.
   6. Уничтожение строения: Удерживание “ПКМ” на строении.
   7. Открытие окна взаимодействия: “ЛКМ” пустым курсором по строению.
   8. Выбор рецепта: “ЛКМ” пустым курсором по иконке предмета в окне взаимодействия.
   9. Сохранение игры: “P”.
   10. Очистка сохранения: Удерживание “O”. (вступает в силу после перезапуска игры)
2. Механика ввода и вывода:

Изображение выглядит как текст, снимок экрана, Прямоугольник, прямоугольный

Контент, сгенерированный ИИ, может содержать ошибки.У каждой постройки, которая может взаимодействовать с коннекторами, при установке показываются оранжевые и зелёные стрелки. Эти стрелки указывают на ввод и вывод ресурсов соответственно. Для работы строения необходима подача ресурсов со стороны входа (оранжевой стрелки) коннектором ввода и вывод продукции со стороны вывода (зелёной стрелки) коннектором вывода. (см. рисунок 1)

Рисунок 1

1. Механика конвейеров:

Изображение выглядит как прямоугольный, Прямоугольник, шаблон, Симметрия

Контент, сгенерированный ИИ, может содержать ошибки.Конвейеры – это простой способ перемещения ресурсов из точки А в точку Б. Конвейеры нужно подключать последовательно, можно изменить направление потока ресурсов повернув конвейер в сторону (см. рисунок 2). Если предмет доедет до края конвейера, и там не будет строений, которые могут его принять, то он упадёт.

Рисунок 2

**Ссылки:**

Проект на GitHub – <https://github.com/FNTnemo/FactoryZone>

Страница игры на itch.io – <https://fntnemo.itch.io/factory-zone>

Видео-инструкция – <https://youtu.be/Pc2DhIfv00I?si=JmJmRbcQpSyvuu_d>

**Список литературы:**

Документация Python – <https://docs.python.org/3/index.html>

Документация Pygame – <https://pygame.readthedocs.io/_/downloads/en/latest/pdf/>

Документация PyYAML – <https://pyyaml.org/wiki/PyYAMLDocumentation>