

ГБОУ “Школа №2098 им. Героя Советского Союза Л.М.Доватора”

Infoboy: Революция в обучении программированию

Участники:

ученик 11 “Т” класса ГБОУ
Школа №2098

Новиков Виктор Николаевич

ученица 11 “Т” класса ГБОУ
Школа №2098

Саргаева Анна Сергеевна

Руководители:

педагог ГБОУ Школа №2098

Гоптарь Евгений Андреевич

педагог ГБОУ Школа №2098

Малевин Дмитрий Сергеевич

1. Обзор Проекта

Infoboy — это геймифицированная платформа, предназначенная для обучения программированию с помощью интерактивной стратегии. Основные компоненты:

- Игровая карта: абстрактное игровое поле, символизирующее различные области знаний в программировании.
- Команды: Игроки объединяются в команды для соревнования.
- Раунды: игра делится на раунды, включающие ответы на вопросы и стратегическое планирование.
- Управление ресурсами: очки, полученные за ответы и захват территорий, используются для усиления позиций.
- Интерактивный Интерфейс: Удобный и привлекательный интерфейс для пользователей.

2. Технические Требования

2.1. Языки и Технологии

- Python: основной язык программирования для реализации логики игры и серверной части.
- Customtkinter: Библиотека для создания графического интерфейса пользователя (GUI).
- Tkinter: Используется для создания основного окна приложения и управления событиями.
- Pillow (PIL): Библиотека для обработки и отображения графических элементов игровой карты.

2.2. Архитектура

Проект следует модульной архитектуре. Основные модули:

- GUI (графический пользовательский интерфейс): отвечает за отображение игры и взаимодействие с пользователем.
- Логика (Игровая логика): содержит правила игры, обработку ответов, управление ресурсами и расчёт результатов.

- Управление данными (Data Management): модуль для работы с данными (например, вопросы, ответы, карта, состояние игры).
- AI (Искусственный интеллект): планируемый модуль для управления неигровыми персонажами (ботами).
- Networking (Сетевое Взаимодействие): Модуль для будущей многопользовательской версии.

2.3. База Данных

- Вопросы и ответы должны храниться в структурированном формате (например, JSON или CSV).
- База данных может быть расширена для хранения пользовательских профилей и статистики.

3. Детализация Компонентов

3.1. GUI (Графический Интерфейс)

- Главное меню: должно предоставлять доступ ко всем основным функциям: начало игры, настройки, выбор карты, выход.
- Игровая карта: отображение карты с территориями, командами и их ресурсами.
- Окна ввода: формы для ввода результатов раундов и управления ресурсами.
- Окна Статистики: Отображение результатов игр и индивидуальных достижений.

3.2. Logic (Игровая Логика)

- Обработка Ответов: Проверка правильности ответов на вопросы и начисление очков.
- Управление ресурсами: распределение очков между командами и использование для «атак».
- Расчет результатов: определение победителя на основе захваченных территорий и набранных очков.

- Стратегическое планирование (частично): реализация механизмов для планирования действий и «атак».

3.3. Data Management (Управление Данными)

- Загрузка данных: загрузка вопросов, ответов и данных игровой карты из внешних файлов.
- Сохранение Данных: Хранение текущего состояния игры.
- Управление Пользовательскими Данными: Запись пользовательских профилей и статистики.

4. Этапы Разработки

4.1. Текущая Версия (V1.0):

- Реализация однопользовательской версии с базовым функционалом.
- Разработка базового интерфейса и игровой логики.
- Создание тестового набора вопросов.
- Тестирование и отладка.

4.2. Планируемые Версии:

- V2.0:
 - Добавление новых карт и вопросов.
 - Расширение функционала для стратегического планирования.
 - Улучшение пользовательского интерфейса.
- V3.0:
 - Внедрение искусственного интеллекта (AI) для управления ботами.
 - Разработка многопользовательского режима.
 - Адаптация под различные платформы (веб, мобильные).

5. Планируемые Технологии (Будущие Версии)

- WebSocket или аналоги: Для сетевого взаимодействия в многопользовательском режиме.
- TensorFlow или PyTorch: для реализации алгоритмов искусственного интеллекта.
- Веб-фреймворки (Django, Flask): для создания веб-версии игры.

6. Документирование и Тестирование

- Код должен быть четко документирован в соответствии с PEP8.
- Необходимо создавать юнит-тесты для каждого модуля.
- Регулярно проводить пользовательское тестирование.

7. Коммуникация

- Регулярные встречи для обсуждения прогресса и решения проблем.
- Использование системы контроля версий (Git).
- Создание и поддержание документации.

8. Заключение

Проект Infoboy обладает значительным потенциалом для изменения подхода к обучению программированию. Успешная реализация проекта зависит от качественной работы команды инженеров и их внимания к деталям. Инженеры должны стремиться к созданию надёжной, удобной и масштабируемой платформы, которая будет привлекательной для пользователей.

Приложение

- Список используемых библиотек и их версий.
- Структура данных для вопросов и ответов.
- Чертежи и схемы архитектуры проекта.

Эта документация содержит основные сведения о проекте и служит руководством для инженеров на всех этапах его разработки. Регулярное обновление документации является важной частью работы.

Ссылки на литературу:

- Документация языка Python /
<https://docs.python.org/3/>
- Документация языка гипертекстовой разметки HTML /
<https://developer.mozilla.org/ru/docs/Web/HTML>
- Документация Фреймворка Flask /
<https://flask.palletsprojects.com/en/3.0.x/>
- Документация языка JavaScript /
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Документация формального языка декорирования и описания внешнего вида документа CSS /
<https://developer.mozilla.org/ru/docs/Web/CSS>
- Документация модуля в Python (SQLAlchemy) /
<https://docs.sqlalchemy.org/en/20/>