

«Национальный исследовательский университет

«Высшая школа экономики»

Лицей

Индивидуальная выпускная работа

Byte Scrap - Игра про создание электронных схем.

Выполнил Антоненко Артём Викторович

Москва 2025

I. Введение

Этот проект представляет собой симулятор сборки электронных схем. Игрок должен проходить уровни, собирая схемы и выполняя задания этого уровня. Я решил делать этот проект, так как немного увлекаюсь электроникой, и в последнее время занимаюсь разработкой игр.

II. Проблемное поле

Изучение электроники и программирования часто начинается со сложной теории и профессиональных программ. Эта игра делает процесс весёлым и наглядным для всех, кто хочет начать, и дает пространство для творчества.

Эта игра может облегчить понимание основ в области электроники. А так же разнообразить опыт людей уже разбирающихся в этой теме. Здесь не нужны профессиональные знания, так как сборка схем сильно упрощённая. Она научит основам сборки схем и откроет простор для творчества, позволяя создавать уникальные схемы.

Игры, которые были исследованы, не могут дать возможности создавать схемы простым путём. Поэтому в игре будет возможность:

- Сборки различных схем различной сложности.
- Следовать целям и подсказкам для прохождения уровня
- Реализовать свои схемы в творческом режиме.

III. Целевая аудитория

Игра может заинтересовать широкую аудиторию игроков, предлагающих жанр песочницы и электронику. Основные группы пользователей будут:

- Студенты факультетов компьютерных наук
- Радиолюбители
- Подростки от 14 до 18 лет

IV. Функциональные требования

Проект будет представлять собой компьютерную игру. Это 3D-игра в жанре песочницы с элементами головоломок. Основной режим игры - Компания.

- Управление клавиатурой и мышью с видом сверху.
- Строительство схем в сетке и симуляция работы.
- 4-6 обучающих уровня со сборкой различных схем.
- Помощь игроку (Вывод подсказки в интерфейсе).
- Детали открывающиеся по мере прохождения уровней.
- Возможность сохранять уровень локально и в облако.
- Творческий режим с доступом ко всем компонентам.
- Сохранение/загрузка схем, созданных разными игроками (Аналог Steam Workshop).
- Система целей на уровне, которые игрок должен выполнять для завершения уровня.

V. Стек технологий

Движок для игры был выбран Unity, так как он достаточно популярный и опыт работы с ним у меня примерно 4 года. Ещё использовались различные ассеты для удобства и ускорения разработки, такие как: дебаг консоль в игре, Hot Reload, и несколько самописных скриптов.

Для бэкенда был выбран Python с фреймворком FastAPI, так как он простой, у него есть хорошая документация, и с ним так же есть опыт работы раньше. В качестве базы данных был выбран PostgreSQL, потому что я привык с ним работать. Весь бэкенд собирается в Docker контейнер. Он развёртывается на моём сервере через GitHub Actions.

VI. Этапы работы над проектом

- Проектная заявка (ноябрь - декабрь 24)
- Первый прототип и базовый бэкенд (январь)
- Система интерфейсов и базовая логика (апрель - май)
- Несколько прототипов системы симуляции схем (июнь - июль)
- Система сохранения (конец июля)
- Окончательная версия системы симуляции схем (август)
- Система целей (сентябрь)
- Бэкенд и интеграция его с игрой (начало октября)
- Первый показ кода (начало октября)
- Второй показ кода (конец октября)
- CI для бэкенда и сборки игры (Начало ноября)

VII. Анализ похожих продуктов

Minecraft – Использование вокселей в строительстве накладывает некоторые ограничения, но сильно уменьшает количество багов. Поэтому была взята механика красного камня, как концепция для системы схем.

Shortcuit - Игра находится ещё в альфа версии, но уже предоставляет симуляции электронных схем. В этом виде эта система планировалась быть изначально.

Scrap Mechanic - В этой игре удобнее всего реализованы интерактивные элементы и инструменты. Так же есть модификации, которые сильно расширяют

их функционал. Эта игра в жанре выживания-песочницы, и инструментарий больше подходит для строительства транспорта, а не создания небольших и электрических схем. Частично взята механика сохранения построек в облако.

VIII. Рефлексия

Самая большая проблема проекта была главная часть игры - система сборки и симуляции схем. Изначально хотелось реализовать так: компоненты располагаются свободно и провода подключаются к пинам. Но возникла проблема зацикливания цепи и было решено сделать систему так: компоненты располагаются в сетке, а сигналы передаются между сторонами. Другая проблема была с внедрением эмуляции терминала linux и программированием. Хоть и идея как реализовать этот функционал была, но ресурсов на реализацию не осталось (система схем заняла слишком много времени).

В этом проекте много чего можно улучшить: графику, звуки, уровни. Но самое главное - это архитектура. Изначально я пытался сделать проект, так чтобы он удобно разрабатывался. И я пытался использовать разные паттерны, но очевидно они не помогли, потому что это просто решение или упрощение какой-либо абстрактной проблемы, а не архитектура. В итоге получается код, с которым, чем сложнее проект, тем сложнее с ним работать.

Работая над проектом, я понял, что очень важно не переоценивать свои силы и полностью не полагаться на изначальное планирование (особенно если делать что-то в первый раз). Так же я понял, что для игр необходима модульная архитектура. В процессе разработки этого проекта у меня появлялась масса идей, как реализовать и дополнить то, что планировалось, и этот подход сильно бы облегчил разработку. Я думаю, что продолжать дальнейшую разработку нет смысла, потому что этот проект был плохо спроектирован изначально.