**Приложение**

Национальный исследовательский университет   
«Высшая школа экономики»

Лицей

Индивидуальная выпускная работа

Отчёт о проекте

**Решайка к ЕГЭ**

*Выполнил Возмитель Андрей Анатольевич*

**(опционально)** Консультант:

Фамилия Имя Отчество

Москва 2023

**Представление продукта**

<https://github.com/VozmitelAndrew/Project_NIU/tree/main> - ссылка на гитхаб со всеми файлами проекта

Я решил делать IT проект, потому что я хотел испытать себя в сфере информационных технологий. Изначально я очень хотел провести проект в области изучения нейронных сетей, но для проекта обязательно необходимо 7 типов функциональных особенностей, что почти невозможно для таких продуктов. Поэтому я был вынужден писать приложение другого формата. Мой выбор пал на инструменты, которые помогут ученикам подготавливаться к ЕГЭ. И вот я здесь.

**Обоснование актуальности, целевая аудитория**

Многие ученики сталкиваются с проблемой нехватки времени и желания для постоянной подготовки к ЕГЭ по математике. Тем не менее, успешное прохождение экзамена является важным шагом для дальнейшей учёбы или карьеры. Именно поэтому я создал приложение, которое поможет множеству учеников подготовиться к ЕГЭ по математике с возможностью расширения для других предметов.

Мой продукт поможет решить проблему нехватки времени на подготовку к ЕГЭ по математике. С приложением, ученики смогут изучать материал наилучшим способом, пропуская те уроки, которые уже хорошо знают, а также дообучаться в слабых местах. То есть, конечный продукт является идеальным помощником для каждого студента.

Нехватка времени для подготовки к ЕГЭ является актуальной проблемой среди многих школьников. Несколько причин, влияющих на это, включают участие во внеклассных мероприятиях, работу или семейные обязательства. Кроме того, современное общество ставит перед учащимися большое количество задач, что часто становится причиной отсутствия достаточного количества времени для подготовки к экзамену. Академическая подготовка также играет важную роль, и многие студенты чувствуют сложности с материалом в выбранных ими предметах. Решением этих проблем может стать мое приложение, которое учитывает время и мотивацию студентов, несмотря на их занятость. Оно содержит несколько типов задач с необходимым материалом для подготовки к ЕГЭ по математике и располагает функциями для сохранения прогресса и мотивирования студентов на ежедневные / еженедельные тренировки.

Также это приложение могут использовать ученики, которых затрудняются с выбором качественного материала для тренировки или которые хотят более эффективно использовать свое свободное время.

Мое приложение имеет ряд функций, среди которых:

1. Систему классификации по типам задач, которые встречаются в ЕГЭ по математике.

2. Возможность проходить тесты и проверять ответы.

3. Награды за прохождение через определенные этапы «игры».

**Видение продукта/Проблемы**

В начале своего проектного пути я натолкнулся на приложение AimLab's. Задача данного приложения - улучшить механические способности игрока, путём тренировок, которые потом можно будет применять в FPS-е своего выбора. Я решил использовать игровой двидок под названием "Unity". В моём приложении, ключевые идеи будут очень схожими с AimLab-овскими. У ученика будет возможность выбора всевозможных заданий для подготовки, после которого он будет получать счет, на основе эффективности/точности своих решений. После чего, этот счет будет попадать в таблицу лидеров. Для удержания игроков будут использоваться различные приёмы, такие как: искусственная прогрессия (новые фоны, задания, прогресс по таблице лидеров) за решение задач, всевозможные достижения, которые будутмотивировать учеников работать дальше.

Я хотел, чтобы пользователь, зашедший на мой продукт, получил большое количество заданий, но при этом удобно организованных. Это должно было удержать пользователей в приложении и эффективно подгатавливать их к экзаменам, ведь ученик сам определяет уровень сложсти, с которым ему придется столкнуться.

Хотелось бы также добавить несколько других функция для того, чтобы удерживать игроков в приложении. Для этого я предполагал отойти от AimLab-овского подхода в пользу своего видения. Однажды, мои друзья школьные друщья играли в приложение «Ударения». Оно не предполагало участия двух и более игроков, но мои одноклассники соревновались за наибольший счет, и соревновались с большим интересом. Тогда то я и понял, насколько это важно – добавлять соревнующуюся среду вне экрана в любом обучащем приложении. Поэтому, я хотел добавить систему глобального чата, систему друзей и кланов, а также возможность для решений заданий в группах. Это бы позволило еще раз добавить интереса и мотивации к решению заданий, к примеру, ради того, чтобы показать другу со школы самую редкий фон. Или же устроить «соревнование» или «поединок» прямо в здании школы. Это бы добавило интереса к моему продукту а также создало бы дополнительные точки взаимодействия между коллективом обучающихся.

Таким образом, я представлял, как мой продукт решит 2 главные проблемы при подготовке к ЕГЭ: небольшая мотивация для долгосрочных занятий, недостаток времени у учеников.

**Вышедший продукт**

К сожалению, продукт несколько видоизменился. Количество заданий оказалось не таким уж и большим, хотя количество решенных заданий можно увеличивать с сервера, что по своей сути уже очень хорошо.

Систему групповых решений тоже пришлось отодвинуть до лучших дней, она требовала очень большого количества ресурсов и знаний. К сожалению, ни того ни другого у меня не было – проект я делал с абсолютного нуля в программировании и на Godot и на Unity.

Поэтому фокус решения пришлось сместить в сторону быстрой подготовки за несколько недель/месяцев до самого экзамена, банально потому что такую систему сделать было легче. Конечно, небольшие куски из соревновательной части я оставил – к примеру, таблицу лидеров.

**Этапы разработки**

Моя работа прошла множество итераций. Возможно чрезчур большое... Но о всем по порядку.

1. Изучить основы движка "Godot"
2. Поиск материалов по интересным способам решения задач
3. Сортировка каждых заданий по уровням сложности
4. Имплементирование регистрации в аккаунт приложения
5. Имплементация над таблицей лидеров, которая приводила к множественным ошибкам. Путей решения я не обнаружил, как я не старался, на контакт со мной никто не вышел. Я вообще заметил, что в коде мной допускатеся огромное количество багов, а фиксить их было практически невозможно, документации ни к Godot, ни к Nakama не было – оба проекта open source. В момент растерянности, мне нужно было дейстовать. Самым быстрым способом в долгосрочной перспективе было перейти на более популярный движок – Unity, ведь за ним лежит огромное камьюнити indie games разработчиков
6. Пришло понимание, что нужно переделывать тот огромный кусок работы, которыыя я уже сделал на совсем новый движок, где каждый элемент двух движков были непохожими друг на друга. Я начинал проект с очередного нуля
7. Изучить основы движка "Unity"
8. Изучать средства работы с "Lootlocker"
9. Создание инвентаря
10. Начало работы с префабами
11. Реализация разделения всех Assets для отображения в инвентаре
12. Реализация отображения вопросов
13. Работа с таймером
14. Вызов функций и измение свойст обьекта через OnClick() event
15. Регистрация/Логин игрока
16. Выдача опыта
17. Введение soft currency
18. Создание системы магазина
19. Работа с фильтрами
20. Работа с тригерами
21. Реализация системы выборов фона/"аватарок"
22. Интегрирование таблицы лидеров

И другие.

**Выбор относительно БД, движка разработки**

Изначально мой выбор пал на Godot. Я был мало знаком ни с одним движком разработки, но из многочисленных видео на Youtube, многие говорили о том, что Godot это один из самых перспективных и развивающихся игровых движков на всей инди-сцене огромного разнообразия игровых движков. Поэтому я и решил работать на нём. А для БД я решил выбрать Nakama на CockroachDB, просто потому что выбора-то и другого толком и не было. Либо Lootlocker, либо Nakama. Но, со временем, для совсем неопытного в геймдев индустрии меня, я понял насколько критическим оказался недостаток материалов и документации. Прогресс был и был неплохим, но чем дальше я шел, тем сильнее этот недостаток информации отнимал у меня времени. Все дошло до того, что на реализацию магазина я потратил больше 14 часов, так его и не доделав.

Поэтому, попав на первую комисиию, я принял решение – менять движок. Я перешел на Unity – самый популярный у инди разработчиков движок. Тут меня окружало огромное количество людей с похожими проблемами, поэтом, вся техника, которая не касалась связи с сервером решалась достаточно просто. Вот в этом тоже просматривается один момент. Lootlocker оказался несколько сложнее, чем я изначально предпологал. Выбрал я его, кстати, очевидно, узнав о всех его преимуществах в Godot, узнав, что Lootlocker действует и в Unity. Возвращаясь к сложности. Документации и видеороликов в этот раз оказалось действительно больше. Хотя, это смотря как сказать. Неплохих обзоров было всего 2 штуки: один из которых выпущен официальными разработчиками, а второй – 1 девушкой с несколькими тысячами подписчиков на YouTube. Вот их ссылки на канал:

https://www.youtube.com/@ZygerGFX

https://www.youtube.com/@LootLocker

И тут не было золотой серидины. Девушка хорошо рассказала о начальных функциях, а официальный сайт не давал хорошей середины, очень разплывчиво описывал работу с функциями Lootlocker-а, заходя сильно вперед (особенно чувствовалось на системе магазина и выдачи Assets). В общем, пришлось мне не сладко, но свой выбор я уже не менял.

**Рефлексия**

В силу небольшого количества времени скажу так – большинсво всех своих проблем и сложностей я указывал в предыдущих пунктах. Особенно(*Выбор относительно БД, движка разработки*). Но действительно, мои опасения насчет моей полной неопытности и отсутствия адекватных источников информации сбылись. Из-за этого я и попал на 3-ю комиссию. Да и в целом, код и там, и там давался мне с огромнейшим трудом. Много вещей ломались просто на ровном месте. И на любое своё минутное действие имплементации нового функционала я мог потратить до 4-5 часов на дебаг. Мой проект хоть и не выглядит внушительным, стоил мне огромных жертв сил и ментального здоровья.

Но не без хорошего: Я научился работать на достаточном уровне и на Unity, и на Godot. Теперь у меня будет больше уверенности в том, чтобы начинать работу в этих сферах. Я смог сравнить эти два движка, посмотреть на пути решения одинаковых задач разными путями. К примеру, В Godot мне очень понравился их встроенный построитель Ui. Он интуитивно понятен, требует минимального количества кода. С другой стороны Unity имеет два разных инструмента для написания UI, и оба из них на достаточно плохом уровне, требует много кодирования. В этом и заключается главное преимущество и недостаток Unity – почти каждый элемент можно, а больше даже нужно кастомизировать и подстраивать под нужны своего проекта.

Улучшить свой продукт я хочу за счет имплементации систем, которые позволили бы расширить пространство для взаимодействий внутри игроков – добавить систему глобального чата, систему друзей и кланов, а также возможность для решений заданий в группах. Это бы позволило еще раз добавить интереса и мотивации к решению заданий. Я бы работал над полной модульностью всех блоков заданий – для русского, информатики и т.д. и т.п. вместо частичной модульности – только для заданий ЕГЭ по математике.