Государственное бюджетное образовательное учреждение дополнительного образования Самарской области «Самарский областной центр детско-юношеского технического творчества»
СП Центр цифрового образования детей «IT-Куб»

Областной конкурс креативных педагогических решений «ВЕКТОР»

МЕТОДИЧЕСКАЯ РАЗРАБОТКА

«ИИ-соавтор в создании образовательной игры: от методической идеи до тренажёра по таблице умножения»

Автор: Немцев Артём Алексеевич
Должность: педагог дополнительного образования
Образовательная организация: СП «Дом детского творчества», ГБОУ СОШ № 1 «ОЦ»

с. Большая Глушица, 2026

ОГЛАВЛЕНИЕ

1. Цель применения искусственного интеллекта 3
2. Используемые инструменты ИИ и их внедрение 4
3. Практическая реализация и архитектура цифрового продукта 6
3.1. Функциональное устройство игры 6
3.2. Система очков и мотивации 6
3.3. Встроенное интерактивное обучение (Туториал) 7
3.4. Архитектурная надежность для школьных реалий 8
3.5. Инновационный модуль адаптивной сложности 8
3.6. Модуль предиктивной диагностики и карта дефицитов 8
4. Результаты применения 9
5. Выводы и рекомендации коллегам 10
ПРИЛОЖЕНИЯ 11

1. Цель применения искусственного интеллекта

Таблица умножения — основа школьной математики, а её освоение требует многократных повторений, которые детям быстро надоедают. Цифровой тренажёр с игровой мотивацией помогает превратить рутину в увлекательную деятельность. Однако педагогу, не имеющему команды разработчиков, создать подобный продукт обычно не по силам: требуются навыки программирования, дизайна и сложной математической балансировки.

Цель: продемонстрировать эффективную практику, при которой педагог без профильного ИТ-образования с помощью инструментов искусственного интеллекта (ИИ) может самостоятельно создать архитектурно надежный, педагогически выверенный цифровой тренажёр и успешно применить его на занятиях.

Задачи:

1. Определить и распределить роли ИИ на разных этапах создания ПО: от проектирования игровых механик до написания кода и генерации графики.
2. Реализовать в архитектуре игры гибкую, педагогически обоснованную систему мотивации, встроенного обучения и адаптивной сложности.
3. Провести апробацию цифрового продукта с обучающимися 10–11 лет.
4. Сформулировать рекомендации для педагогического сообщества по эффективному и безопасному взаимодействию с генеративными моделями.

2. Используемые инструменты ИИ и их внедрение

В проекте использовались три профильных ИИ-инструмента. Каждый сервис подбирался под специфические задачи, исходя из его сильных сторон:

1. *Claude Code* — использовался как ведущий ИИ-инженер для написания чистого, отказоустойчивого кода игры, реализации логики управления экранами и связки с облачной платформой.

2. *DeepSeek* — применялся для математического моделирования, генерации формул и калибровки весов в системе начисления очков и штрафов.

3. *Qwen* — выполнял задачи по проектированию пиксельных макетов животных и генерации визуального контента.

Таблица 1

Распределение ролей в совместной разработке

Задача / Модуль	**Профильный инструмент ИИ**	**Роль ИИ**	**Роль педагога**
Игровая логика и код	Claude Code	Написание кода и исправление ошибок по техническому описанию	Постановка логических задач, интеграция в среду разработки Unity и проверка
Математика и баланс	DeepSeek	Расчёт формул прогрессивного штрафа, бонусов за скорость и серии	Экспертное утверждение весов очков и педагогических лимитов
Графический контент	Qwen	Генерация идей для пиксель-арта, подготовка описаний	Нарезка текстур, импорт в проект, адаптация и оптимизация

Стратегия внедрения строилась по принципу «ИИ предлагает и пишет код — педагог модерирует и проверяет» и включала пять итераций:

1. *Педагогическая постановка.* Зафиксирован тренируемый навык (нахождение неизвестного множителя) и ключевой триггер мотивации (геймифицированная закраска скрытого пиксельного изображения).

2. *Проектирование.* Совместно с ИИ спроектирована система управления состояниями (меню, игра, пауза, результат), что позволило изолировать логику вычислений от интерфейса.

3. *Итеративная разработка.* Код создавался модулями. ИИ генерировал алгоритмы, а педагог внедрял их в проект, возвращая ИИ на доработку любые логические нестыковки.

4. *Создание медиа.* Сгенерированы референсы животных. Программа динамически считывает пиксели изображения, преобразуя их в игровое поле для закраски.

5. *Апробация и полировка.* По результатам тестов с детьми ИИ помог добавить элементы визуальной обратной связи (анимации кнопок, плавное появление карточек), которые сделали игру яркой, отзывчивой и приятной визуально.

Важнейшая методическая особенность - концепция изолированной верификации:

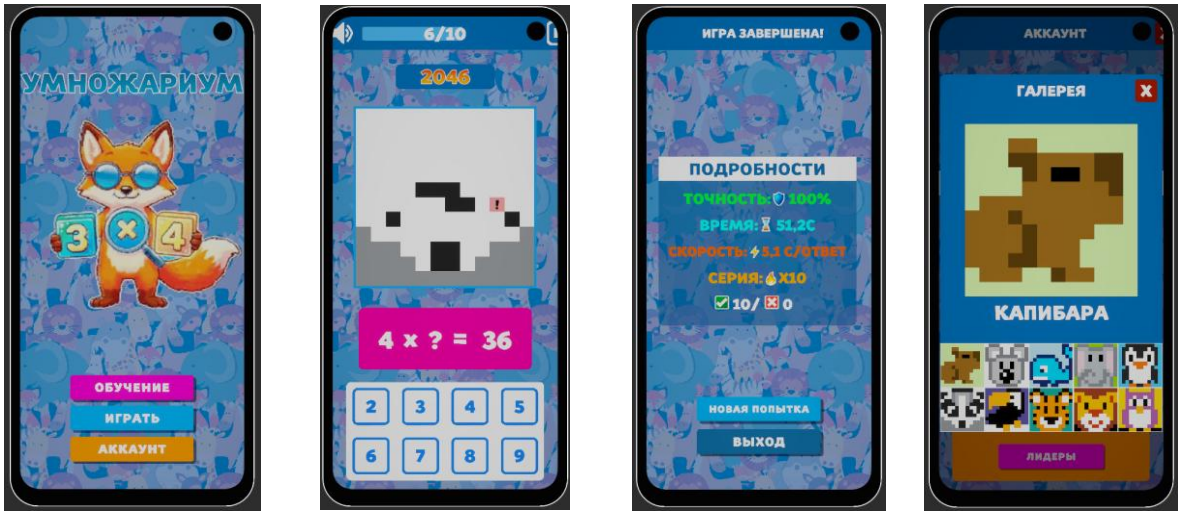
- уравнения и математическая логика в игре генерируются строгим программным кодом на основе классического алгоритма случайной выборки;
- правильный ответ всегда вычисляется математически непосредственно на устройстве;
- большие языковые модели полностью исключены из процесса генерации самих примеров в реальном времени. Это на 100% гарантирует отсутствие «галлюцинаций» и ошибок в примерах, что критически важно для доверия детей и родителей к образовательному продукту.

Промпты и примеры работы с искусственным интеллектом приведены в приложении А.

3. Практическая реализация и архитектура цифрового продукта

3.1. Функциональное устройство игры

После запуска игры пользователь попадает в главное меню (рис. 1а). При запуске раунда программа обращается к фабрике примеров, которая на основе списков разрешенных множителей формирует пул из 10 уникальных задач. В верхней части экрана отображается счетчик прогресса (например, "6/10") и текущий счет игрока (например, "2046"). Игрок видит пример с пропущенным компонентом на яркой цветной плашке (например, "4 × ? = 36") и панель ввода вариантов ответов — кнопки с числами от 2 до 9, расположенные в две строки (рис. 1б). За каждый правильный ответ система мгновенно окрашивает целевую ячейку сетки в истинный цвет скрытого изображения. При ошибке срабатывает анимация покачивания, а на экране появляется подсказка с правильным ответом. По окончании каждого раунда пользователю демонстрируется расширенная статистика (рис. 1в). Игра содержит 10 уникальных пиксельных животных. Если ребенок завершает раунд с точностью выше 70%, животное разблокируется в личной галерее (рис. 1г). Умный алгоритм выдачи приоритетно показывает тех животных, которые ребенок еще не открыл, стимулируя продолжать занятия.



а) б) в) г)

Рис. 1. – Интерфейс приложения

3.2. Система очков и мотивации

Чтобы игра не превращалась в бездумное «закликивание», совместно с ИИ спроектирована многофакторная система оценки:

1. *Базовые очки:* зависят от сложности примера (от 50 за простые до 200 за сложные).

2. *Бонус за скорость:* если ребенок отвечает быстрее норматива (4 сек.), он получает бонус. Реализована защита: если время превышает норму, бонус обнуляется.

3. *Серии ответов:* за правильные ответы подряд начисляются дополнительные очки, а за серии из 5 и 10 ответов предусмотрены крупные бонусы.

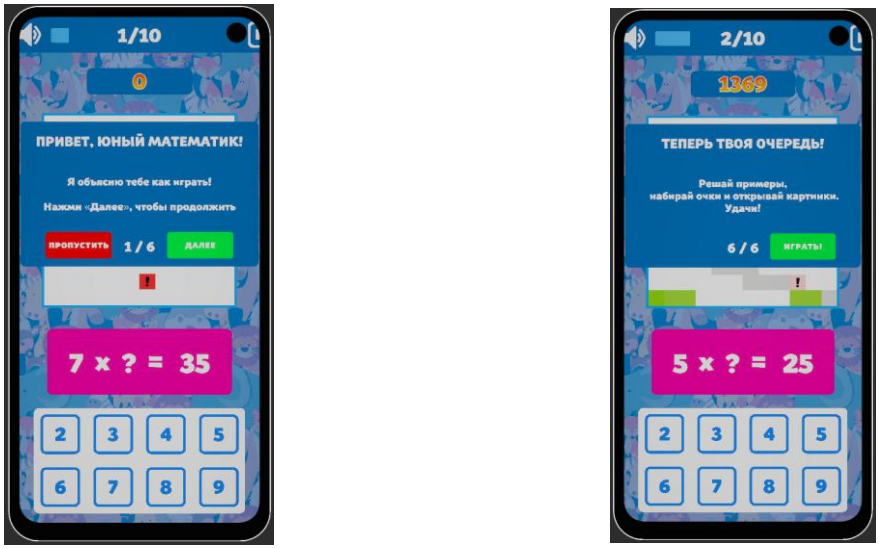
4. *Бонус за точность:* масштабируется вплоть до +750 очков при 100% правильных ответов.

5. *Прогрессивный штраф:* ошибки наказываются строго, но установлен жесткий «потолок» — штраф не может превысить 30% от базовых очков. Это защищает учеников от демотивации и ухода в «глубокий минус».

Все параметры отображаются на экране результатов после завершения раунда. Параметры системы оценивания и констант баланса приведены в приложение Б.

3.3. Встроенное интерактивное обучение (Туториал)

Чтобы минимизировать участие учителя в объяснении интерфейса, в игру внедрен пошаговый управляемый туториал, доступный через кнопку "ОБУЧЕНИЕ" на главном меню. Он состоит из 6 фаз: от приветствия (рис. 2а) и выделения текущего примера пульсирующей рамкой до блокировки неверных кнопок с текстовым предупреждением («Попробуй еще раз!»). Шаг не завершится, пока не будет нажат верный ответ. Это реализует принцип scaffolding (педагогических лесов), мягко проводя ребенка от незнания к самостоятельной игре (рис. 2б).



а) б)

Рис. 2 – Интерфейс обучающего модуля

3.4. Архитектурная надежность для школьных реалий

В проект заложены стандарты отказоустойчивости, критически важные для школ:

1. *Умный Offline-режим:* если сеть в школе нестабильна, игра автоматически подменяет облачные сервисы на автономные. Приложение продолжает работать локально, не зависая.
2. *Безопасное слияние данных:* при восстановлении сети профили объединяются по принципу сохранения максимума. Облачное сохранение никогда не затрет локальный прогресс, достигнутый ребенком во время оффлайн-сессии.

3.5. Инновационный модуль адаптивной сложности

Контроллер сложности работает по принципу обратной связи, реализуя психологический принцип «ситуации успеха»:

1. *Повышение уровня:* при точности ≥ 90% и скорости ответа в пределах нормы уровень повышается. Важно: включен педагогический фильтр — уровень не повысится, если ребенок тратил слишком много времени (защита от механического подсчета на пальцах или черновике).
2. *Понижение уровня:* при точности ≤ 60% приложение мягко снижает сложность, возвращая ребенка в ситуацию успеха и снижая математическую тревожность.

3.6. Модуль предиктивной диагностики и карта дефицитов

Для объективного контроля динамики результатов интегрирован автономный диагностический модуль, решающий проблему отсутствия классических замеров «до» и «после»:

1. *Входной фильтр:* первая игровая сессия автоматически переходит в режим «Диагностика» (без подсказок и эффектов). Фиксируется чистая скорость и точность.
2. *Цифровая карта дефицитов:* на основе диагностики формируется матрица. Примеры, в которых ребенок ошибся или думал дольше норматива, получают повышенный приоритет выдачи в следующих раундах до тех пор, пока навык не стабилизируется.
3. *Скрытый контроль:* каждая 5-я игра анализируется модулем. Педагог получает автоматический лог прогресса без проведения стрессовых для детей контрольных работ.

4. Результаты применения

Апробация проведена на занятиях кружка. Дети 10–11 лет проверяли знание таблицы умножения и получали награды (приложение Г). Сценарий применения на занятии:

- 5 минут — вводное слово педагога и демонстрация маскота;
- 10 минут — самостоятельная работа детей на ПК;
- 5 минут — обсуждение результатов и открытых животных.

Таблица 2

Ключевые показатели эффективности разработки

Показатель	Результат
Общий объем выборки (апробация)	Более 60 детей в возрасте 10–11 лет
Среднее количество сессий на ребенка	3-7 игр (возвращались ради улучшения рекордов)
Медиа-охват образовательного события	Более 1500 просмотров постов о занятиях во ВКонтакте
Процент добровольного повторного обращения	70 % (тренировка выполнялась по личной инициативе)
Текущий статус дистрибуции	Продукт опубликован на платформе Яндекс Игры

Что изменилось в образовательном процессе: применение тренажёра переломило отношение детей к проверке знаний. Процесс заучивания превратился из принудительного в соревновательный. Ученики самостоятельно совершали повторные сессии, чтобы улучшить положение в таблице лидеров.

Экономия времени и ресурсов педагога: благодаря делегированию рутины ИИ, весь цикл разработки от идеи до публикации занял всего 14 дней (около 28 рабочих часов). Аналогичный продукт традиционным методом занял бы не менее 3–4 месяцев. ИИ сэкономил более 80% времени, позволив педагогу сфокусироваться на методике и дидактике.

Верифицированные данные диагностики: после 4–5 сессий среднее время поиска пропущенного множителя в «трудных» примерах снизилось с 6.8 секунд до 3.2 секунд (при нормативных 4 с), а общая точность стабилизировалась на уровне 88–92%. Это доказывает высокую дидактическую эффективность адаптивного алгоритма.

Визуальная привлекательность интерфейса (яркие цвета, анимированный маскот-лиса, пиксельные животные) значительно повысила вовлеченность детей. Ученики активно обсуждали, какое животное откроют следующим, и стремились заполнить всю галерею.

5. Выводы и рекомендации коллегам

Выводы:

1. Генеративный ИИ позволяет педагогу преодолеть «технологический барьер» и создавать сложные цифровые продукты без привлечения программистов.
2. Наивысшую эффективность дает строгое разделение труда: педагог — архитектор смыслов и методист, ИИ — исполнитель рутинных операций.

Практические рекомендации для коллег:

1. *Идите от дидактической задачи.* Прежде чем открывать чат с ИИ, распишите на бумаге правила игры и микронавык. ИИ — идеальный исполнитель, но плохой педагог.
2. *Разбивайте разработку на микро-итерации.* Не просите ИИ написать «сразу всю игру». Проектируйте систему пошагово: от одной ячейки до сложных систем.
3. *Изолируйте вычисления от нейросетей.* Никогда не поручайте языковым моделям генерацию точных математических данных в реальном времени. Используйте ИИ для написания жестких алгоритмов, где ответ вычисляется кодом нативно.
4. *Закладывайте аппаратную устойчивость.* При создании приложений для школ всегда требуйте от ИИ реализации оффлайн-режимов, чтобы урок не был сорван из-за падения школьного Wi-Fi.
5. *Тестируйте на целевой аудитории как можно раньше.* Первые сборки нужно давать детям сразу. Их интерфейсные затруднения — лучший источник данных для доработки туториала.
6. Соблюдайте цифровую гигиену: не передавайте ИИ персональные данные детей и обязательно проверяйте сгенерированный контент на соответствие возрастным нормам.

Перспективы развития:

Ближайший вектор — проведение окружного дистанционного конкурса на знание таблицы умножения. Облачное хранилище позволит формировать глобальную таблицу лидеров. Игра доступна по ссылкам из приложения Г.

Интеграция предиктивной диагностики открывает перспективы для регионального внедрения. Накопленная статистика (цифровые карты дефицитов школьников) позволит сформировать единую региональную тепловую карту типичных затруднений при изучении математики.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Промпты и примеры работы с искусственным интеллектом

Пример 1. Проектирование архитектурной логики (Система загрузки)

Задача: разработать безопасную схему инициализации игры, учитывающую сбои сети в школе и необходимость работы в автономном режиме.

Инструмент: Claude Code

Промпт: «Напиши архитектурный модуль пайплайна инициализации. Он должен принимать список шагов, сортировать их и выполнять асинхронно. Заложи поддержку токена отмены. Если шаг завершается с флагом сбоя (например, оффлайн при таймауте платформы), логируй предупреждение, но не прерывай загрузку. Код должен быть чистым, без использования тяжелых компонентов движка».

Результат и доработка: ИИ сгенерировал отказоустойчивую систему классов. Педагог добавил интеграцию кастомного логгера для записи логов загрузки в текстовый файл на устройстве для последующего анализа сбоев.

Пример 2. Моделирование математического баланса очков

Задача: сформировать алгоритм расчета очков, стимулирующий отвечать быстро и аккуратно, с прогрессивным, но безопасным для психики наказанием за ошибки.

Инструмент: DeepSeek

Промпт: «Спроектируй класс расчета очков. Базовые очки должны зависеть от сложности примера (от 50 до 200). Напиши алгоритм бонуса за скорость относительно норматива в 4 секунды. Реализуй прогрессивную шкалу штрафов: за 1-ю ошибку — 30, за 2-ю — 70, за последующие — по 50. Штраф должен иметь жесткое ограничение — не более 30% от базового счета, чтобы не уходить в глубокий минус».

Результат и доработка: ИИ предоставил выверенный математический алгоритм. Педагог добавил логику начисления «утешительной звезды» для детей, ответивших верно хотя бы на 50% вопросов, для поддержки мотивации отстающих учеников.

Пример 3. Разработка контроллера адаптивной сложности

Задача: создать модуль автоматического изменения сложности на основе успешности ребенка в реальном времени.

Инструмент: DeepSeek

Промпт: «Напиши класс, управляющий уровнем сложности. Если точность игрока выше 90% и его средняя скорость ответа не превысила норму, увеличивай стрик побед и повышай уровень. Если точность ниже 60%, увеличивай стрик поражений и через 2 игры снижай сложность. Заложи очистку подписок при выходе в главное меню».

Результат и доработка: ИИ создал рабочий реактивный компонент. Педагог доработал логику сброса накопленных серий при выходе в меню, чтобы новая учебная сессия всегда начиналась со стабильной, сохраненной в профиле сложности.

Пример 4. Проектирование пиксель-арт маскота и персонажей для игрового поля

Задача: разработать строгие правила и промпты для генерации изображений животных, которые программа сможет однозначно считать и разбить на цветовую сетку 10×10 без артефактов. Отдельно — создать маскота игры (лису в очках с кубиками «3 × 4») для главного экрана.

Инструмент: Qwen

Промпт для маскота: «Составь детальное текстовое описание (промпт) для генерации пиксель-арт изображения лисы — главного персонажа (маскота) образовательной игры "Умножариум". Лиса должна стоять анфас, держать в лапках два кубика с цифрами "3" и "4" и знаком умножения между ними. На лисе надеты круглые очки в синей оправе. Размер изображения — 64×64 пикселя. Палитра: оранжевый (#FF8C42), белый (#FFFFFF), чёрный (#1A1A1A), тёмно-коричневый (#5C3317), синий для очков (#4A90E2). Стиль: дружелюбный ретро 8-бит, подходящий для детей 8–10 лет. Фон — прозрачный или однородный светло-голубой (#B8D4E8). Критически важно: избегай градиентов, сглаживания (anti-aliasing) и полупрозрачных пикселей, чтобы программный код мог однозначно считывать цвет каждой ячейки.»

Промпт для игровых животных (капибара, кит, лев и др.): «Составь промпт для генерации пиксель-арт изображения капибары размером 16×16 пикселей. Изображение должно иметь чёткие контуры, ограниченную палитру из 4–5 контрастных цветов (коричневый, тёмно-коричневый, чёрный, бежевый) и строго прозрачный фон. Стиль: ретро 8-бит,

дружелюбный. Критически важно: избегай градиентов, сглаживания и полупрозрачных пикселей. Животное должно занимать центральную часть кадра, оставляя по краям не менее 1 ячейки пустого фонового пространства.»

Результат: ИИ выдал структурированные промпты для генератора изображений, а также таблицу рекомендуемой цветовой палитры с HEX-кодами для обеспечения единообразия всех 10 животных в игре и маскота.

Доработка педагога: В требования к промпту было добавлено условие симметрии и проверки контрастности цветов для обеспечения инклюзивности (корректное восприятие детьми с особенностями цветовосприятия, например, дейтеранопией). Также добавлено правило: все животные должны быть повёрнуты анфас для единообразия галереи.

Пример 5. Разработка концепции и промптов для UI-элементов интерфейса

Задача: создать визуальный стиль интерфейса (кнопки главного меню, панель примера, экран результатов), который будет гармонировать с пиксельными животными и соответствовать санитарным нормам (СанПиН) для детских приложений.

Инструмент: Qwen

Промпт для кнопок главного меню: «Разработай концепцию и текстовые промпты для генерации набора UI-кнопок в стиле пиксель-арт для главного меню образовательной игры. Необходимы три кнопки: "ОБУЧЕНИЕ" (фиолетовая, #C77DFF), "ИГРАТЬ" (голубая, #00B4D8), "АККАУНТ" (оранжевая, #F4A261). Каждая кнопка должна иметь три состояния: обычное, наведение (hover, на 10% светлее), нажатие (pressed, на 10% темнее). Требования: крупный пиксельный шрифт без засечек (стилизация под "Press Start 2P"), скруглённые пиксельные углы, тёмная пиксельная обводка толщиной 2 пикселя. Размер каждой кнопки — 200×48 пикселей.»

Результат: ИИ предложил описание стилей, пропорции элементов и конкретные промпты для генерации ассетов, разделив их на состояния (normal, hover, pressed).

Доработка педагога: Уточнил требование к шрифтам: добавил в промпт условие использования крупного шрифта без засечек и увеличил контрастность между текстом и фоном панели для снижения зрительной нагрузки. Также попросил ИИ проверить цветовую палитру на соответствие требованиям WCAG 2.1 AA для контрастности текста.

ПРИЛОЖЕНИЕ Б

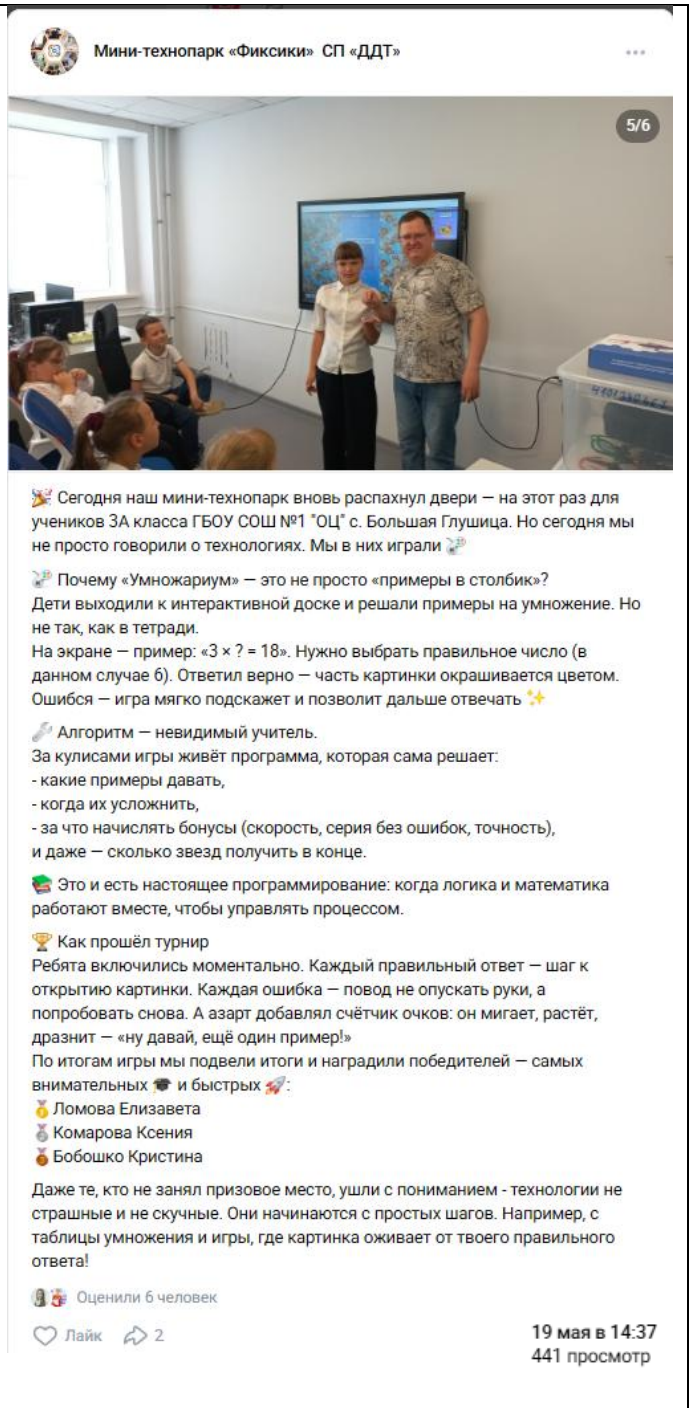
Параметры системы оценивания и констант баланса

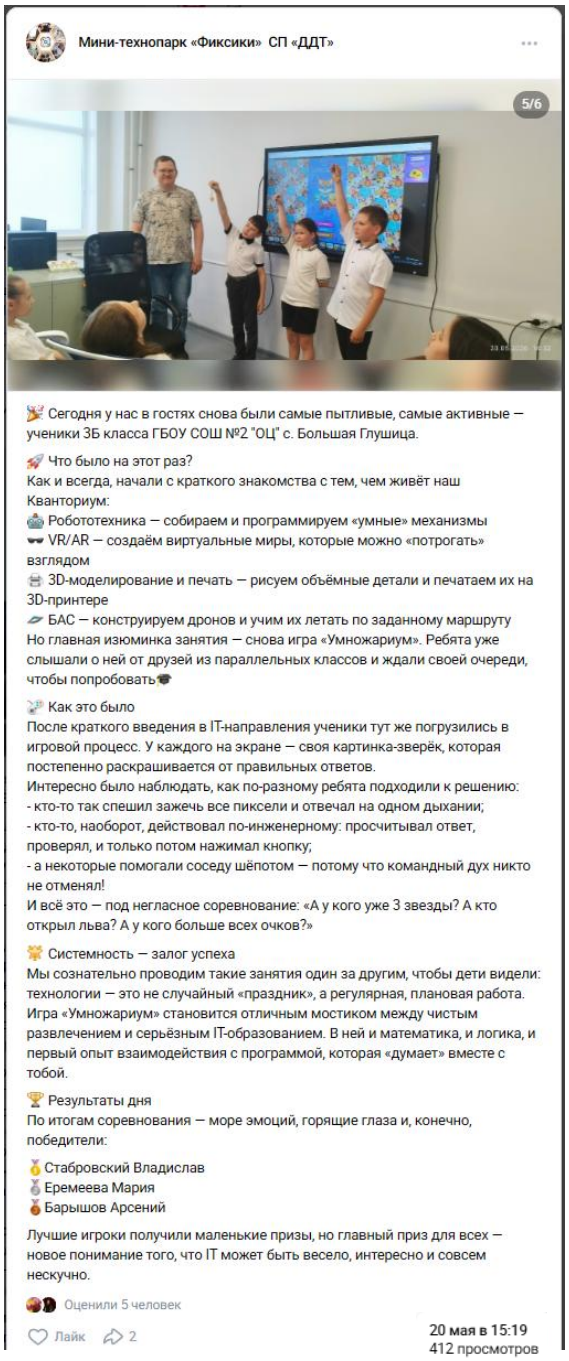
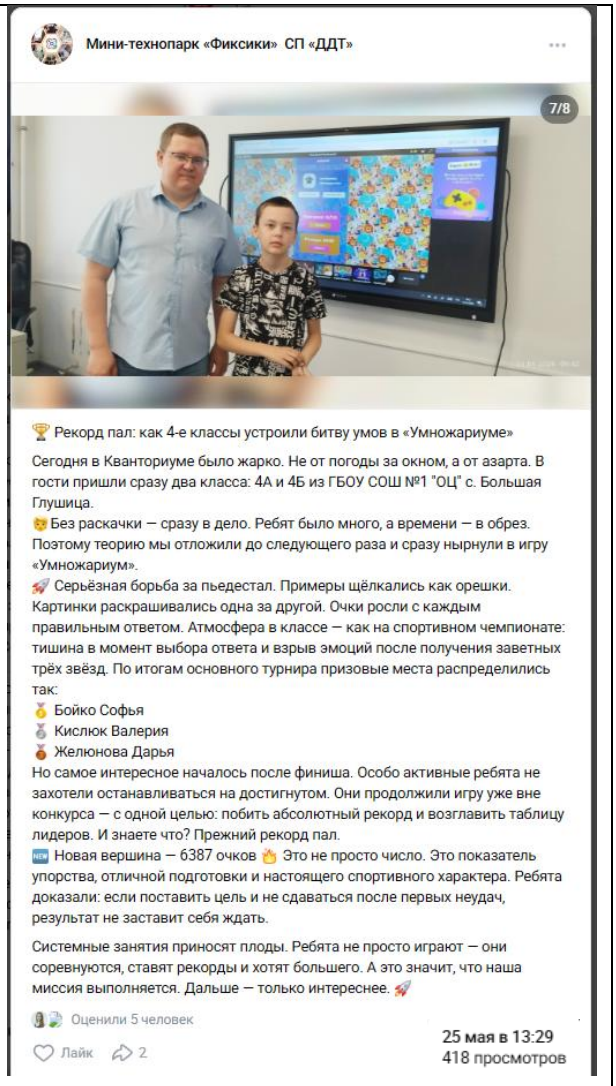
(Извлечено из программных конфигурационных файлов ScoreConfig.cs и GameplayConstants.cs)

Имя переменной в коде проекта	**Смысловое педагогическое значение параметра**	**Утвержденное значение**
MinPointsPerEquation	Стартовые (базовые) очки за решение простейшего уравнения	50 очков
MaxPointsPerEquation	Максимальные базовые очки за самые сложные уравнения (9x9)	200 очков
MaxTimeBonusPerQuestion	Верхний лимит дополнительного бонуса за мгновенный ответ	200 очков
StreakThreshold	Порог непрерывной серии верных ответов для включения множителя	Начиная с 3 ответов подряд
StreakPointsPerLevel	Очки за каждый последующий шаг серии выше установленного порога	50 очков
StreakMilestone5Bonus	Единоразовый крупный бонус за удержание серии из 5 ответов	200 очков
StreakMilestone10Bonus	Единоразовый триумфальный бонус за серию из 10 ответов	500 очков
AccuracyBonusThreshold	Минимальный порог точности раунда для начисления бонуса	70% верных ответов
MaxAccuracyBonus	Максимальный бонус за идеальную точность раунда (100% результат)	750 очков
FirstMistakePenalty	Фиксированный штраф, накладываемый за совершение первой ошибки	30 очков
SecondMistakePenalty	Повышенный штраф, накладываемый за совершение второй ошибки	70 очков
AdditionalMistakePenalty	Штраф за каждую последующую ошибку (начиная с третьей)	50 очков
MaxPenaltyPercent	Жесткий лимит («потолок») штрафа от общего счета (защита профиля)	30% от базовых очков
PerfectionBonusMax	Супер-бонус за сочетание 100% точности и скорости опережения нормы	500 очков
UnlockProgressValue	Процент верных ответов, необходимый для разблокирования животного	Более 70%
DefaultEquationCount	Фиксированное количество математических примеров в одном раунде	10 уравнений

ПРИЛОЖЕНИЕ В

Промпты и примеры работы с искусственным интеллектом

	
Ссылка	Ссылка

	
Ссылка	Ссылка

ПРИЛОЖЕНИЕ Г

QR-код для доступа к игре



ССЫЛКА для доступа к игре