

Генерация развивающей веб-игры для дошкольников методами промпт-инжиниринга на примере нейросети Qwen Studio

Анишкова Анастасия Сергеевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается процесс разработки браузерной обучающей игры для дошкольников «Собери слово из кубиков» с использованием большой языковой модели Qwen Studio. Исследование сосредоточено на особенностях проектирования детских интерактивных приложений: реализации механизма перетаскивания объектов, процедурном синтезе звуковой обратной связи и построении педагогически корректной системы визуальных подсказок. Особое внимание уделено итеративному характеру взаимодействия с нейросетью, где каждая доработка требует от разработчика чёткой формулировки не только технической, но и методической задачи. Результаты показывают, что языковые модели способны самостоятельно выстраивать многоуровневую игровую логику и адаптировать интерфейс под возрастную аудиторию, однако проектирование обучающей обратной связи потребовало нескольких итераций уточнения.

Ключевые слова: Qwen Studio, нейросети, промпт-инжиниринг, вайб-кодинг.

Generating an educational web game for preschoolers using prompt engineering methods, demonstrated using the Qwen Studio neural network

Анишкова Анастасия Сергеевна

Sholom-Aleichem Priamursky State University

Student

Absract

The article examines the process of developing a browser-based educational game for preschoolers called “Assemble a Word from Cubes” using the large language model Qwen Studio. The study focuses on the specifics of designing children’s interactive applications: implementing a mechanism for dragging objects, procedurally synthesizing sound feedback, and building a pedagogically sound system of visual cues. Special attention is paid to the iterative nature of interaction with the neural network, where each revision requires the developer to clearly formulate not only a technical but also a methodological task. The results show that language models are capable of independently constructing multi-level game logic and adapting the interface to the target age group; however, designing the learning feedback required several iterations of refinement.

Keywords: Qwen Studio, neural networks, prompt engineering, vibe coding.

1 Введение

1.1 Актуальность

Раннее развитие детей дошкольного возраста всё чаще опирается на цифровые инструменты. Интерактивные обучающие приложения становятся повседневным элементом домашнего и дополнительного образования, помогая осваивать чтение, счёт и логику в игровой форме. Однако рынок детских веб-приложений сталкивается с двумя противоречивыми тенденциями: с одной стороны, коммерческие продукты часто перегружены рекламой и требуют подключения к интернету, с другой – бесплатные решения редко отличаются продуманной педагогической методикой и качественной обратной связью.

Разработка по-настоящему хорошего детского приложения требует одновременного решения нескольких задач: интуитивного интерфейса, учитывающего моторные особенности маленьких пользователей, корректной методической подачи материала, приятной визуальной и звуковой обратной связи. Традиционно такая работа занимает недели и требует участия методиста, дизайнера и программиста. Возникает вопрос: способна ли большая языковая модель, получив структурированное описание педагогической задачи, самостоятельно спроектировать приложение, сочетающее игровую механику и обучающую ценность?

1.2 Обзор исследований

Базовые принципы построения интерактивных веб-продуктов и игр с использованием исключительно нативных HTML5-технологий, исключающих внешние зависимости, подробно изложены в работах С.В.Рогатина и В. А. Антошкина [1]. Эффективность применения больших языковых моделей для анализа HTML-структур и оптимизации процессов тестирования веб-приложений наглядно показана в исследовании А.М.Титеева [2]. Методологию вайб-кодинга как актуальный инструмент цифровой педагогики и инженерной практики, значительно расширяющий функциональные горизонты разработчиков, детально рассматривает А.Б.Рахмонов [3]. Роль генеративных искусственных интеллектов как стратегического фактора, обеспечивающего конкурентное преимущество и радикальное ускорение ИТ-разработки, эмпирически подтверждается в трудах Д. С. Пашенко [4]. Потребность в жестком инженерном надзоре и внедрении парадигмы Spec Driven Development для качественной верификации кода, созданного нейросетями в режиме вайб-кодинга, аргументированно обосновывает И. В. Елманов [5].

1.3 Цель исследования

Цель работы – оценить возможности большой языковой модели Qwen Studio при создании развивающей веб-игры для дошкольников, сочетающей механику перетаскивания объектов, процедурный синтез звуков и многоуровневую систему визуальной обратной связи.

2 Материалы и методы

Исследование построено на методе итеративного промпт-инжиниринга с использованием платформы Qwen Studio. Всё приложение должно размещаться в едином HTML-файле, использующем исключительно нативные HTML, CSS и JavaScript без внешних библиотек, шрифтов и медиафайлов.

Для механики перетаскивания букв выбран стандартный HTML5 Drag-and-Drop API, обеспечивающий работу как на настольных браузерах, так и на сенсорных устройствах. Звуковая обратная связь реализуется через Web Audio API с процедурным синтезом: звуки генерируются математически, без использования готовых аудиофайлов, что полностью соответствует концепции автономного приложения.

Ключевой особенностью методологии стало разделение ролей: нейросеть выступает в роли исполнителя, реализующего техническую часть, а разработчик формулирует не только функциональные, но и педагогические требования – как именно приложение должно реагировать на ошибки ребёнка, какие подсказки давать, как поддерживать мотивацию.

3 Результаты

Первый этап заключался в формулировании комплексного промпта, описывающего архитектуру игры и её педагогическую логику.

Промт для генерации: «Создай развивающую веб-игру для дошкольников «Собери слово из кубиков» со следующими характеристиками:

Цель игры: Ребёнок должен составить слово из букв-кубиков в правильном порядке, глядя на картинку-подсказку.

Функциональные требования:

1. Три уровня сложности:

- Лёгкий: слова из 3 букв
- Средний: слова из 4 букв
- Сложный: слова из 5 букв

2. Игровой процесс:

- Показывается картинка и подсказка со словом
- Буквы перемешаны и представлены в виде ярких кубиков
- Ребёнок перетаскивает буквы в правильном порядке
- При правильном ответе – анимация успеха и переход к

следующему слову

- При ошибке – мягкая подсказка

3. Дизайн:

- Яркие, дружелюбные цвета
- Крупные элементы для маленьких детей
- Анимации и визуальная обратная связь

4. Технические требования:

- Один HTML-файл
- Работает локально без сервера
- Плавные анимации

5. Дополнительные возможности:

- Счётчик правильных ответов
- Кнопка подсказки
- Возможность выбора уровня сложности

Создай полностью рабочую игру с минимум 9 словами для каждого уровня сложности».

Результат генерации: https://disk.yandex.ru/d/YFUzzpBVmf9_iA

В результате первичной генерации получен работоспособный прототип (рис. 1). Нейросеть реализовала три уровня сложности, механику перетаскивания букв через Drag-and-Drop API, систему подсказок и счётчик правильных ответов. Интерфейс выполнен в ярких цветах с крупными элементами, что соответствует принципам детского UX.



Рисунок 1. Интерфейс приложения

Ключевым достижением первичной генерации стала корректная реализация игровой логики: нейросеть самостоятельно разделила слова по уровням сложности, реализовала перемешивание букв, настроила анимации успеха и ошибки. CSS-анимации (подпрыгивание картинки, пульсация

сообщения, тряска при ошибке) добавили приложению живости, важной для удержания внимания ребёнка.

Однако при тестировании выявились две проблемы. Во-первых, нейросеть допустила содержательные ошибки: на уровне «4 буквы» попались слова из пяти букв (КНИГА, ПТИЦА) (рис. 2), а на уровне «5 букв» – слова из шести букв (ДОРОГА, МАШИНА, СОЛНЦЕ, ДЕРЕВО, ЦВЕТOK, МОЛОКО, ПЕЧЕНЬЕ, ЯБЛОКО) (рис. 3). Для дошкольной игры такая неточность критична: ребёнок, видя несоответствие между заявленной длиной слова и фактическим количеством букв, теряет доверие к приложению.

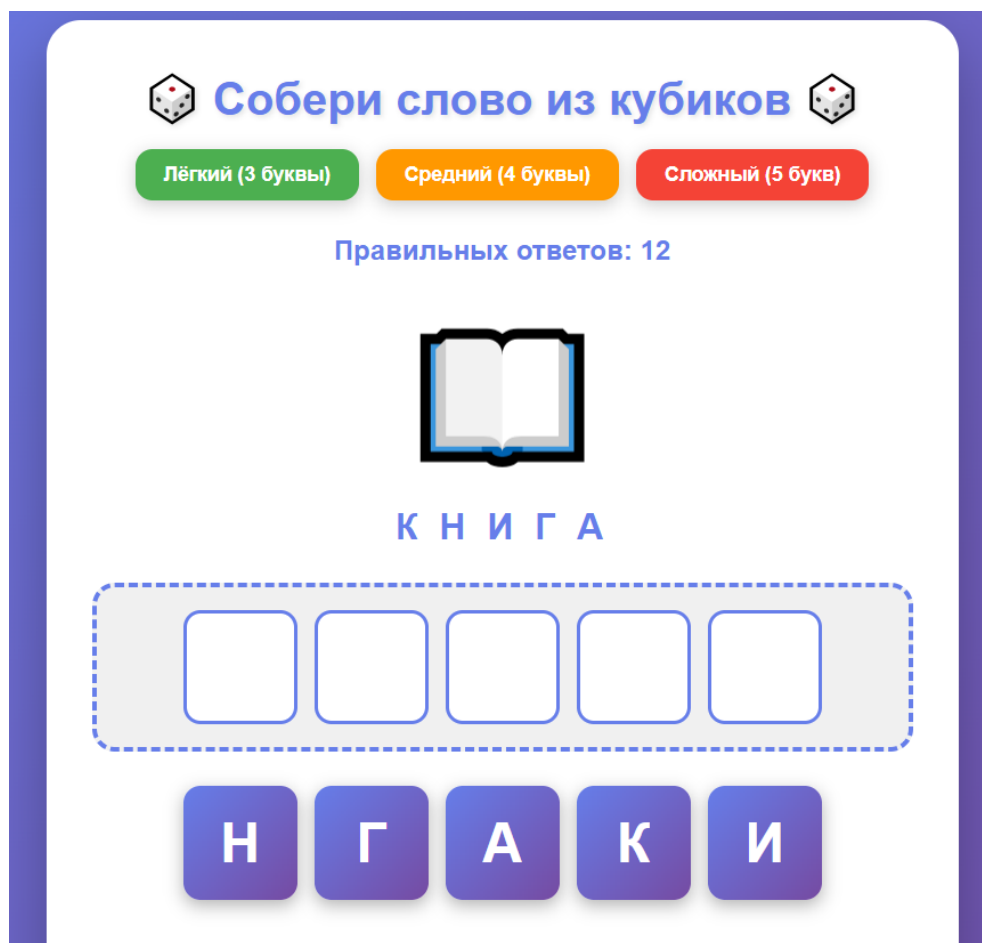


Рисунок 2. Уровень «4 буквы»



Рисунок 3. Уровень «5 букв»

Во-вторых, приложение не имело звуковой обратной связи. Для детской игры это существенный недостаток: звук усиливает эмоциональное восприятие успеха и смягчает реакцию на ошибку.

Для решения выявленных проблем сформулирован уточняющий промт.

Промт для доработки: «1. На втором уровне сложности (слова из 4 букв) несколько слов из 5 букв, а на третьем уровне сложности (слова из 5 букв) несколько слов, состоящих из 6 букв, замени их на другие слова, подходящие по количеству букв.

2. Добавь звуковые эффекты при правильно и неправильно собранном слове».

Результат генерации: <https://disk.yandex.ru/d/O95-UbkvIhfR5w>

Нейросеть заменила все слова на корректные по количеству букв. На среднем уровне теперь: ЛУНА, РЫБА, СЛОН, ЗВОН, ГРИБ, МЫЛО, ТОРТ, РУКА, НОГА (рис. 4). На сложном: КОШКА, ШКОЛА, КНИГА, ТРАВА, МЫШКА, МИШКА, ШАРИК, ЗАЙКА, ДОМИК (рис. 5).

Звуковая обратная связь реализована через Web Audio API с процедурным синтезом. Нейросеть самостоятельно выбрала музыкальные

решения: для успеха – восходящая мелодия из трёх нот, создающая ощущение победы; для ошибки – мягкий нисходящий звук пилообразной формы, не пугающий ребёнка, но ясно сигнализирующий о необходимости повторить попытку.



Рисунок 4. Уровень «4 буквы»



Рисунок 5. Уровень «5 букв»

Важно, что звуки генерируются математически в момент воспроизведения, без использования внешних аудиофайлов. Это полностью соответствует методологии исследования и позволяет приложению работать полностью автономно. Частоты и огибающие подобраны так, чтобы звуки приятными и не резкими – критичное требование для детской аудитории.

Третий этап оказался наиболее интересным с методической точки зрения. Исходная логика приложения при ошибке просто сбрасывала все буквы и предлагала начать заново. Такой подход педагогически неоптимален: ребёнок, правильно собравший часть слова, теряет свой прогресс и вынужден повторять уже сделанное. Это демотивирует и не помогает обучению.

Сформулирован промт, описывающий более тонкую логику обратной связи.

Промт для доработки: «Добавь визуал неправильного ответа (собранное слово подсвечивается красным)».

Результат генерации: <https://disk.yandex.ru/d/8p-3LkAnGaoZkQ>

Нейросеть реализовала значительно более сложную логику, чем можно ожидать из краткого запроса. При ошибке приложение анализирует каждую позицию отдельно:

- Правильно поставленные буквы остаются зелёными – ребёнок видит, что он уже знает.
- Неправильно поставленные буквы подсвечиваются красным с пульсирующим свечением и анимацией тряски (рис. 6).
- Через две секунды красные буквы автоматически сбрасываются, а зелёные остаются на местах.
- Соответствующие кубики возвращаются в набор, и ребёнок может попробовать поставить их правильно (рис. 7).



Рисунок 6. Ошибка в ответе



Рисунок 7. Вторая попытка

Такая реализация превращает ошибку из неудачи в обучающий момент. Ребёнок получает чёткую обратную связь: «вот здесь ты угадал, а вот здесь нужно подумать ещё». Это соответствует принципам формирующего оценивания в педагогике, где ошибка рассматривается не как повод для наказания, а как информация для коррекции.

Технически задача оказалась нетривиальной: потребовалось ввести систему уникальных идентификаторов для каждого кубика, чтобы отслеживать связь между слотом и буквой, и реализовать частичный сброс состояния без потери прогресса. Нейросеть справилась с этой задачей, хотя в первоначальном промте эти детали не прописаны явно – модель самостоятельно вывела необходимую архитектуру из общего описания желаемого поведения.

4 Вывод

Проведённое исследование демонстрирует, что большие языковые модели способны успешно создавать развивающие веб-приложения для дошкольников самостоятельно, при этом человек выступает лишь методистом и тестировщиком. Эксперимент по генерации игры «Собери слово из кубиков» выявил несколько важных закономерностей.

Нейросеть хорошо справляется с технической реализацией игровой механики: Drag-and-Drop, анимации, адаптивный интерфейс, процедурный синтез звуков – все эти компоненты реализованы корректно и в соответствии с парадигмой автономного приложения.

Содержательные ошибки подтверждают необходимость верификации сгенерированного контента человеком. Для детской игры такие ошибки критичны, так как подрывают доверие к приложению.

Процедурный синтез звуков через Web Audio API подтвердил свою эффективность: приложение получило эмоциональную окраску без привлечения внешних ресурсов, что особенно важно для автономных детских приложений.

Полученные результаты позволяют сформулировать практическую рекомендацию: при создании детских обучающих приложений с помощью ИИ разработчик должен выступать в роли методиста, формулируя не только технические, но и педагогические требования. Нейросеть в этом тандеме берёт на себя техническую реализацию, но не заменяет человека в вопросах возрастного соответствия и обучающей ценности.

В перспективе описанная методология может быть расширена на другие возрастные группы и предметные области: обучение счёту, геометрии, иностранным языкам. Итеративный подход к промпт-инжинирингу, продемонстрированный в данном исследовании, универсален и применим к широкому классу образовательных веб-приложений.

Библиографический список

1. Рогатин С. В., Антошкин В. А. Разработка компьютерных игр по технологии html5 //Информатика и прикладная математика. 2015. №. 21. С.12-16. URL: <https://elibrary.ru/item.asp?id=25030492>
2. Титеев А. М., Оптимизация процесса создания автотестов веб-приложений с использованием LLM и структурного анализа HTML //Моделирование и анализ данных. 2025. Т. 15. №4. С. 87–103. URL: https://psyjournals.ru/journals/mda/archive/2025_n4/Titeev
3. Рахмонов А. Б. Вайб-кодинг как инструмент цифровой педагогики: функциональные возможности и механизмы реализации //The Lingua Spectrum. 2025. Т. 12. №. 2. С. 552-556. URL: <https://lingvospektr.uz/index.php/lnbsp/article/view/1329>
4. Пащенко Д. С. Использование ИИ-инструментов в конкурентном развитии ИТ-компаний: прикладное исследование 2026 //Информатика. Экономика. Управление-Informatics. Economics. Management. 2026. Т. 5. №. 3. С. 2012-2024. URL: <https://oajiem.com/index.php/24/article/view/297>
5. Елманов И. В. Инженерный подход к ИИ: использование spec driven development для валидации vibe coding //Вестник науки. 2025. Т. 4. №. 12 (93). С. 716-723. URL: <https://cyberleninka.ru/article/n/inzhenernyy-podhod-kii-ispolzovanie-spec-driven-development-dlya-validatsii-vibe-coding>