

созданию более интуитивного и персонализированного пользовательского опыта. Сайт корректно отображается на любом устройстве, повышается его удобство и визуальная согласованность, что напрямую влияет на эффективность взаимодействия и увеличивает вероятность совершения целевого действия. Таким образом, применение искусственного интеллекта в адаптивном веб дизайне представляет собой шаг к созданию интеллектуальных, гибких цифровых решений нового поколения [8].

Заключение

Проведённое исследование позволяет сделать вывод о том, что адаптивный дизайн прошёл значительный путь эволюции, от решения узких технических задач по корректному отображению контента на различных устройствах до становления полноценной концепции, определяющей стандарты современной веб разработки. Его развитие закономерно связано с ростом разнообразия мобильных устройств, экранных форматов и способов взаимодействия пользователей с цифровыми платформами. На современном этапе адаптивный дизайн характеризуется переходом от фиксированных шаблонов к интеллектуальным системам, способным динамически подстраиваться под индивидуальные потребности пользователя и контекст использования. В стратегической перспективе адаптивный дизайн рассматривается как универсальный принцип проектирования цифровых продуктов, обеспечивающий их гибкость, устойчивость и соответствие актуальным технологическим тенденциям.

Список литературы

1. Koishigawa K. A Brief History of Responsive Web Design. URL: <https://www.freecodecamp.org/news/a-brief-history-of-responsive-web-design/> (дата обращения: 15.12.2025).
2. Лемешко К. Г., Калайда А. В. Современные тенденции веб-дизайна и адаптивной верстки // Научный аспект. 2019. № 2. С. 1634-1639. [Электронный ресурс]. URL: https://na-journal.ru/pdf/nauchnyi_aspekt_2-2019_t13_web.pdf#page=74 (дата обращения: 15.12.2025).
3. Joni Korpi. Adaptive web design. As applied to the design process of a web application. [Электронный ресурс]. URL: https://www.theseus.fi/bitstream/handle/10024/40696/Korpi_Joni.pdf (дата обращения: 11.12.2025).
4. Эволюция веб-дизайна: от HTML-страниц к адаптивным системам. [Электронный ресурс]. URL: <https://sky.pro/wiki/profession/istoriya-i-razvitiye-veb-dizajna/> (дата обращения: 15.12.2025).
5. meta viewport. Как он работает? [Электронный ресурс]. URL: <https://itchief.ru/html-and-css/viewport-meta-tag> (дата обращения: 18.12.2025).
6. Егорова И. Н., Толстая Е. А. Исследование принципов адаптивного веб-дизайна. URL: <https://cyberleninka.ru/article/n/issledovanie-printsipov-adaptivnogo-veb-dizajna/viewer> (дата обращения: 15.12.2025).
7. Руководство по реализации отзывчивого дизайна в 2023 году. [Электронный ресурс]. URL: <https://habr.com/ru/companies/ruvds/articles/718700/> (дата обращения: 17.12.2025).
8. Использование ии для создания адаптивного веб-дизайна. [Электронный ресурс]. URL: <https://sмм-agentstvo.ru/blog/ispolzovanie-ii-dlia-sozdaniia-adaptivnogo-veb-dizajna/> (дата обращения: 10.12.2025).

9. Стрельченко О. В., Олейник Н. П. Адаптивный дизайн и производительность / Материалы XVII Международного студенческого научного форума, 2025. [Электронный ресурс]. URL: <https://scienceforum.ru/2025/article/2018037504> (дата обращения: 15.12.2025).

АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ БАЛАНСА ИГРОВЫХ ХАРАКТЕРИСТИК В АВТОМАТИЗИРОВАННЫХ АНАЛОГАХ НАСТОЛЬНЫХ ИГР

Сушинский И. В., Лебедев В. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: v.lebedev@melsu.ru

Научный руководитель: Лебедев В. А.

Введение

Игровая индустрия последних лет демонстрирует рост интереса к мобильным платформам и мессенджерам. Текстовые квесты и адаптации настольных игр на базе чат-ботов (в том числе в Telegram) привлекают пользователей доступностью: нет необходимости устанавливать дополнительное ПО, игра интегрирована в привычную среду общения [1]. Особенностью данного сегмента является низкий порог входа для разработчиков и мгновенное распространение через механизмы вирусного маркетинга в групповых чатах.

Перенос классических настольных игр (например, «Бункер» – игра на выживание и социальную дедукцию) в цифровой формат создаёт технические сложности. В физической версии баланс и уникальность игровых ситуаций достигаются конечным набором карт в колоде. Взятая карта покидает набор, повторное её появление невозможно.

Механика физической колоды обеспечивает несколько критически важных свойств игрового процесса. Конечность ресурсов создаёт информационное преимущество для внимательных игроков: запоминание выбывших карт позволяет вычислять вероятности появления оставшихся элементов. Детерминированность распределения исключает статистические аномалии – невозможна ситуация, когда в десяти последовательных раздачах доминирует одна категория карт.

В программной реализации со стандартными генераторами псевдослучайных чисел (ГПСЧ) возникает риск нарушения этого принципа [2]. Без алгоритмического контроля возможны дублирование уникальных ролей (два персонажа с одинаковой биологией или профессией) или искажения вероятностного распределения характеристик. Такие ошибки разрушают игровую логику и снижают вовлечённость пользователей.

Стандартные генераторы псевдослучайных чисел, основанные на алгоритмах линейного

конгруэнтного метода или вихря Мерсенна, обеспечивают статистическую случайность последовательности, но не гарантируют уникальность элементов в ограниченной выборке. При наивной реализации вероятность коллизии растёт экспоненциально с увеличением числа участников. Для колоды из N карт вероятность дублирования при k выборках описывается парадоксом дней рождения и достигает 50 процентов уже при k примерно равном корень квадратный из N [7].

Цель исследования – разработать и описать алгоритм процедурной генерации игрового контента, который обеспечивает математически строгий баланс характеристик и воспроизводит механику «выборки без возвращения» в Telegram-боте.

Поставленная цель предполагает решение комплекса взаимосвязанных задач. Необходимо обеспечить изоморфизм вероятностных свойств между физической и цифровой реализацией, то есть для любого подмножества карт вероятность его появления в цифровой версии должна совпадать с вероятностью выбора соответствующих физических карт. Важно спроектировать архитектуру хранения состояния, устойчивую к сбоям и обеспечивающую консистентность данных при параллельных запросах от игроков одной сессии.

Материалы и методы исследования

Программная архитектура чат-бота основана на методах алгоритмизации стохастических процессов и технологиях работы с структурированными данными в Python. Методологическая база: встроенные механизмы сериализации объектов для управления состоянием сессии, применение регулярных выражений для синтаксического анализа и преобразования строк. Реализована логика конечных автоматов для корректного переключения фаз игры и целостности данных между запросами.

В качестве базовой методологии выбран принцип чистых функций с иммутабельными структурами данных. Каждая операция генерации создаёт новое состояние игровой сессии, не модифицируя предыдущее, что упрощает отладку и позволяет реализовать механизм отката транзакций при ошибках. Для управления конкурентным доступом применяется блокировка на уровне файловой системы, предотвращающая состояние гонки при одновременных запросах игроков.

Материал исследования – корпусы текстовых данных в JSON-структурах с массивами категориальных переменных: профессий, биологических характеристик, карт здоровья и предметов. Данные имеют жёсткие перекрёстные зависимости, требующие валидации при генерации. Для создания игровых профилей используется выборка без возвращения через ди-

намическое исключение элементов из списков. Этот подход с условной логикой корректировки параметров (синхронизация возраста и стажа) моделирует замкнутую экономику ресурсов, исключая коллизии и противоречия в контенте.

Структура данных организована по принципу нормализованных реляционных таблиц, адаптированных к формату JSON. Каждая категория характеристик представлена массивом объектов с атрибутами. Карты здоровья имеют иерархическую структуру: корневой объект описывает заболевание, вложенный массив содержит возможные стадии с различной степенью тяжести. Данные прошли экспертную валидацию на предмет логической непротиворечивости.

Результаты исследования и их обсуждение

Архитектура программного комплекса – событийно-ориентированная система, интегрированная в Telegram. Основная логика генерации контента вынесена в отдельный модуль на Python. В отличие от стандартных чат-ботов без сохранения состояния, данная реализация фиксирует контекст каждой игровой сессии.

Событийная модель построена на паттерне наблюдатель-подписчик. Каждое действие пользователя генерирует событие, которое обрабатывается соответствующим обработчиком без блокирования основного цикла приложения. Разделение логики генерации контента в отдельный модуль следует принципу единственной ответственности: модуль взаимодействия с API мессенджера отвечает только за маршрутизацию сообщений, модуль генерации – исключительно за создание игрового контента.

Модель данных: каждая активная партия представлена уникальным JSON-объектом. Этот файл – динамическая база данных, хранящая состояние «колоды» игровых карт. Принципы построения: генерация общей среды и персонафикация игроков.

JSON-объект сессии включает метаданные игры: уникальный идентификатор партии, временную метку создания, список участников с их Telegram ID, текущую фазу игрового процесса. Динамическая часть содержит актуальное состояние колоды – массивы доступных карт для каждой категории. Важным аспектом является атомарность операций: каждое изменение состояния сопровождается немедленной записью на диск, что гарантирует консистентность даже при аварийном завершении процесса.

1. Алгоритмическая реализация генерации среды

Функция `create_bunker` инициализирует игровой мир и создаёт контекст выживания. Алгоритм загружает список карт из файла-шаблона, выбирает случайные элементы и удаляет их

из доступных опций (ключевой момент для баланса) [4]. Программная реализация этапа:

Детализация алгоритмической логики включает несколько критических этапов. Загрузка файла сессии выполняется с обработкой исключений: если файл повреждён или содержит невалидный JSON, система создаёт новый объект сессии с предупреждением администратору. Выборка катастрофы реализована через метод `random.choice`, использующий вихрь Мерсенна в качестве базового ГПСЧ [5]. После выбора элемент немедленно удаляется из массива методом `remove`.

Загрузка состояния: система читает файл `game_id.json`.

Выборка катастрофы: метод `random.choice()` определяет тип угрозы, выбранный элемент удаляется методом `remove()`. Это исключает коллизии условий в партии.

Формирование ресурсов: в цикле выбираются предметы для бункера. Исключение выбранных элементов на каждой итерации моделирует физическое убывание, повышая ценность оставшихся предметов.

Такой подход обеспечивает уменьшение множества `Sstore` с каждой итерацией, предотвращая дубликаты предметов.

$$|S_{i+1}| = |S_i| - 1$$

Математическая формализация процесса: пусть исходное множество `S` содержит `N` элементов. После `k` выборов мощность множества становится равной `N` минус `k`. Вероятность выбора конкретного элемента на `i`-й итерации равна единица делённая на `N` минус `i` плюс единица, что точно соответствует вероятности извлечения карты из физической колоды с уменьшающимся количеством карт.

Важно отметить, что применение метода выборки без возвращения обеспечивает не только уникальность элементов, но и сохраняет статистические свойства распределения, соответствующие физической реализации игры. Это критически важно для поддержания справедливости игрового процесса и доверия пользователей к системе.

2. Комплексная генерация ролевых характеристик

Наиболее сложный этап – распределение характеристик персонажа с учётом логических зависимостей между параметрами. Реализованная логика:

Сложность данного этапа обусловлена необходимостью согласования множественных ограничений целостности. Каждый атрибут персонажа существует в отдельном семантическом пространстве, но между пространствами установлены жёсткие связи. Профессия накладывает ограничения на минимальный возраст,

биологические характеристики влияют на допустимые виды хобби, состояние здоровья коррелирует с возрастными группами.

Атомарная выборка: профессия, биологические характеристики, хобби и факты выбираются без возвращения. Это гарантирует уникальность игроков в партии.

Процедурная валидация: для реализма стаж зависит от возраста. Алгоритм собирает строку биологии, извлекает возраст регулярными выражениями, генерирует стаж в диапазоне `k0`, Возраст – 18]. При нулевом стаже генерируется значение в месяцах с помощью функции склонения.

Алгоритм извлечения возраста использует регулярное выражение, сканирующее строку биологии на наличие паттерна «число лет». После извлечения числового значения выполняется валидация диапазона: возраст должен находиться в интервале от 18 до 80 лет согласно правилам игры. Специальная обработка нулевого стажа предусматривает конвертацию в месяцы: генерируется значение от 1 до 11, затем применяется функция склонения существительного «месяц» в соответствии с правилами русского языка.

Иерархическая генерация здоровья: метод `pop()` извлекает объект болезни целиком, затем выбирается конкретная стадия. Это предотвращает ситуацию, когда два игрока получают одну болезнь с разными стадиями (противоречие механике уникальности).

Метод `pop` обеспечивает атомарность операции выборки: элемент извлекается из массива и одновременно удаляется, что исключает необходимость отдельного вызова `remove` и предотвращает условие гонки при параллельных запросах. После извлечения объекта болезни система обращается к вложенному массиву стадий. Выбор стадии выполняется независимым вызовом `random.choice`, поскольку стадии одного заболевания не являются уникальными между партиями.

Процесс генерации характеристик построен таким образом, чтобы обеспечить максимальное разнообразие игровых ситуаций при сохранении логической целостности. Каждый персонаж представляет собой уникальную комбинацию параметров, что способствует повышению вовлечённости и стратегической глубины игрового процесса.

Комбинаторная сложность пространства возможных персонажей достигает астрономических значений. При типичной конфигурации игры с 50 профессиями, 40 биологическими характеристиками, 30 хобби и 25 картами здоровья количество теоретически возможных уникальных комбинаций превышает 10 в степени 9. Это обеспечивает практически бесконечную играбельность: вероятность получить идентичную комбинацию в двух независимых партиях пренебрежимо мала.

3. Оценка эффективности и игрового баланса

Тестирование проведено на выборке из 1000 симуляций игровых сессий. Критерии оценки: отсутствие дубликатов уникальных ролей и семантическая корректность профилей.

Методология тестирования включала автоматизированную генерацию игровых сессий с различным количеством участников: от минимального состава в 4 игрока до максимального в 24 человека. Каждая симуляция выполняла полный цикл создания бункера и распределения ролей, затем результаты проходили серию проверок на уникальность множеств, валидацию перекрёстных зависимостей и анализ семантической корректности.

Анализ результатов: метод выборки без возвращения обеспечивает «входную случайность». Вероятность P получения профессии x для k -го игрока при общем количестве профессий N динамически изменяется:

$$P(x_k) = \frac{1}{N - (k - 1)}$$

где $(k - 1)$ – количество выданных ролей. С каждым игроком пространство вариантов сужается, но вероятность получения любой из оставшихся ролей равномерна, как при физической раздаче карт.

Данное свойство алгоритма является прямым следствием теоремы о гипергеометрическом распределении. В отличие от схемы выборки с возвращением, где вероятности остаются константными, предлагаемый метод моделирует конечную популяцию без пополнения. Эмпирическая верификация теоретических расчётов выполнена методом Монте-Карло: для каждой профессии вычислялась частота появления в 1000 симуляциях, затем применялся критерий хи-квадрат для проверки гипотезы о равномерности распределения.

Система устойчива к логическим ошибкам: в 100% случаев числовые параметры (возраст, стаж, длительность хобби) находились в допустимых диапазонах, что подтверждает эффективность процедурных ограничений.

Дополнительно было проведено сравнительное исследование производительности различных алгоритмов генерации, которое подтвердило оптимальность выбранного подхода. Время обработки одного запроса составляет менее 50 мс, что обеспечивает мгновенный отклик системы с точки зрения пользователя.

Сравнительный анализ включал три альтернативных подхода к генерации [3]. Первый – наивная реализация с проверкой дубликатов через множество: каждая выборка добавляла элемент в набор, при коллизии выборка повторялась. Второй подход – предварительное перемешивание всей колоды алгоритмом

Фишера-Йейтса с последовательной раздачей. Предлагаемый метод с выборкой без возвращения показал оптимальное соотношение времени выполнения и потребления памяти.

4. Практические аспекты внедрения и оптимизации

Интеграция с мессенджером Telegram осуществляется через официальный Bot API с использованием механизма вебхуков по протоколу HTTPS [6]. В отличие от метода длительного опроса, данная архитектура инициирует передачу данных только при наступлении события, что существенно снижает холостую нагрузку на сервер и сетевой трафик. Обработка входящих запросов происходит асинхронно, что позволяет обслуживать множество параллельных игровых сессий без блокировки основного потока выполнения.

Архитектура вебхуков требует настройки обратного прокси-сервера, обеспечивающего терминацию SSL и переадресацию запросов к приложению Python. Каждый входящий запрос от API Telegram содержит JSON-объект с информацией о событии: тип события, идентификатор пользователя, текст сообщения, метаданные чата. Приложение парсит этот объект, определяет текущую фазу игровой сессии для данного чата, вызывает соответствующий обработчик.

Нагрузочное тестирование в условиях эксплуатации проводилось на виртуальном сервере с конфигурацией: 4 ядра CPU и 16 ГБ RAM. Результаты показали высокую эффективность оптимизации кода: загрузка процессорных мощностей не превышала 30% при одновременной поддержке 50 активных игровых сессий. Среднее время отклика системы оставалось в пределах комфортных значений UX, что подтверждает масштабируемость решения и корректность выбранного архитектурного подхода.

Сценарий нагрузочного тестирования моделировал реалистичное поведение пользователей: создание новых игровых комнат с различными интервалами, одновременные запросы генерации персонажей от всех участников партии.

Заключение

Разработано и программно реализовано алгоритмическое обеспечение для автоматизированного аналога настольной игры «Бункер». Метод выборки без возвращения с хранением состояния колоды в JSON эффективен для решения проблемы дублирования игровых сущностей.

Функции `create_bunker` и `create_person` решают задачу цифровой имитации физических компонентов игры: обеспечивают уникальность ситуаций и реиграбельность, логическую непротиворечивость характеристик через перекрёстную валидацию параметров, математиче-

ски обоснованный баланс для соревновательной механики социальной дедукции.

Достигнутые результаты имеют методологическое значение для проектирования аналогичных систем. Продемонстрирована применимость классических алгоритмических подходов теории вероятностей к современным задачам разработки игрового программного обеспечения. Архитектурное решение с хранением состояния в JSON-файлах демонстрирует оптимальный баланс между простотой реализации, производительностью и надёжностью для класса приложений с умеренной нагрузкой.

Результаты показывают целесообразность применения таких алгоритмических подходов при разработке чат-ботов для настольных игр: качественные цифровые адаптации с минимальными требованиями к клиентскому ПО.

Список литературы

1. Гордеев Д. А., Ковалева Т. М. Разработка чат-ботов для мессенджера Telegram на языке Python // E-Scio. 2021. № 4 (55). С. 526-533.
2. Пименов М. А. Практики балансирования компьютерных игр // International Journal of Open Information Technologies. 2020. Т. 8, № 11. С. 53-60.
3. Сысоева А. А. Анализ алгоритмов процедурной генерации уровней в видеоиграх // StudNet. 2021. Т. 4, № 6. С. 1245-1252.
4. Епифанов А. С. Гибридный алгоритм процедурной генерации подземелий для 2D-игр // Инновации и инвестиции. 2021. № 5. С. 235-238.
5. Кекеев К. В. Анализ генераторов псевдослучайных чисел // Вестник современных исследований. 2019. № 1.14 (28). С. 22-25.
6. Рубцов А. А. Разработка чат-бота для мессенджера Telegram: практические аспекты // Вестник науки. 2021. Т. 4, № 9 (42). С. 53-57.
7. Гмурман В. Е. Теория вероятностей и математическая статистика: учебное пособие для вузов. 12-е изд. М.: Юрайт, 2024. 479 с.

ЦИФРОВЫЕ ТЕХНОЛОГИИ В СФЕРЕ ОБРАЗОВАНИЯ ЗАПОРОЖСКОЙ ОБЛАСТИ

Тарасенко И. В.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ilyatarasenko2020@gmail.com*

Научный руководитель: Ступницкий В. С.

Введение

Современный этап развития общества характеризуется стремительной цифровизацией всех сфер жизнедеятельности, и в соответствии с Распоряжением Правительства РФ от 2 декабря 2021 г. № 3427-р «Об утверждении стратегического направления в области цифровой трансформации образования, относящейся к сфере деятельности Министерства просвещения РФ» [1], образование не является исключением. Вне-

дрение цифровых технологий в образовательную среду открывает множество возможностей улучшения не только имеющихся педагогических методик, но и развитие новых, а также повышает качество предоставляемых образовательных услуг.

В рамках исследования в данной статье будут рассмотрены ключевые направления цифровизации, выявлены основные проблемы и препятствия, а также предложены конкретные рекомендации по оптимизации этого процесса с учетом специфики региона. Особое внимание будет уделено сравнению опыта цифровизации в Запорожской области с другими новыми регионами, что позволит выявить наиболее эффективные стратегии и практики, направленной на повышение конкурентоспособности региона и обеспечение равных возможностей для всех обучающихся.

Цель исследования – на основе анализа имеющихся данных по цифровизации образовательной сферы Запорожской области разработать новую стратегию внедрения, более простую для пользователей.

Материалы и методы исследования

В основе предлагаемой стратегии лежит собранная статистика в период с марта 2025-го года по декабрь 2025-го года. В данный период вводилось большинство цифровых технологий в образовательную среду с целью интеграции Запорожской области в общероссийскую систему образования. Определение ключевых моментов внедрения, анализ проблем, с которыми сталкивались образовательные учреждения в данный период времени, который поможет в формировании новой стратегии цифровизации образовательной системы.

Результаты исследования и их обсуждение

Более лояльная и ориентированная на пользователей стратегия внедрения цифровых технологий в сферу образования в Запорожской области с учётом текущей ситуации в регионе. Результат разработки стратегии – пошаговый план по внедрению, представленный в графическом виде.

На рис. 1. представлены основные региональные исполнители во внедрении цифровых технологий и их пользователи.

Главными целями разрабатываемой стратегии являются:

- Лояльность населения к цифровым нововведениям;
- Обеспечение более простого и постепенного внедрения цифровых технологий со стороны конечных исполнителей во внедрении;
- Определение и устранение проблем в процессе внедрения.