ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ РЕАЛИЗАЦИИ ИНТЕРАКТИВНОЙ ИГРЫ НА ПЛАТФОРМЕ UNITY

Сабадаш Д.А., студент гр. 281071

Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь

Середа И.А. – маг. техн. наук, старший преподаватель.

В проекте разработано браузерное игровое программное средство GetRich в жанре idle/clicker с микросервисной архитектурой. Цель проекта – создание масштабируемого решения, обеспечивающего активный и пассивный фарминг внутриигровой валюты, систему карточек, бустеров, сундуков, реферальную программу и таблицу лидеров.

Современная индустрия интерактивных развлечений демонстрирует устойчивый рост браузерных игр, благодаря мгновенному доступу без необходимости установки. Жанр idle/clicker сочетает простоту управления с глубокой прогрессией и пассивным накоплением ресурсов, привлекая широкую аудиторию. Технологии WebGL и WebAssembly обеспечивают производительность, сопоставимую с нативными приложениями, позволяя интегрировать игровые движки в веб-среду [1]. Разрабатываемое программное средство GetRich направлено на реализацию современных игровых механик с применением микросервисной архитектуры, обеспечивающей масштабируемость и надёжность системы [2].

Разработанное игровое программное средство реализует комплекс игровых механик, обеспечивающих вовлечённость пользователей. Система пассивного фарминга автоматически начисляет доход за время отсутствия игрока на основе дохода в час и активных бустеров с ограничением по размеру хранилища. Активный фарминг реализован через систему динамического мультипликатора, увеличивающего получаемый доход с каждым кликом. Система карточек позволяет игрокам улучшать параметры прогрессии через экспоненциальную шкалу стоимости улучшений. Система бустеров предоставляет три направления улучшения: длительность, мультипликатор и скорость перезарядки, с транзакционной безопасностью через Repeatable Read уровень изоляции.

Архитектура программного средства базируется на микросервисном подходе и включает пять основных компонентов. Backend на Go обрабатывает основную игровую логику, включая пассивный и активный фарминг, управление карточками, систему сундуков и реферальную систему [4]. Сервис администрирования на TypeScript управляет удалённой конфигурацией системы. Сервис бустеров на TypeScript изолирует логику временных усилений с транзакционной безопасностью. Frontend на React служит оболочкой для Unity клиента, обеспечивая JSBridge коммуникацию. Unity клиент на движке Unity 6 реализует визуальную часть игры с использованием Universal Render Pipeline для оптимизированного рендеринга [4].

На рисунке 1 представлен главный игровой экран, где отображается персонаж-рабочий, собирающий монеты с игровой платформы. Интерфейс включает текущий баланс монет, шкалу энергии, мультипликатор и навигационную панель для переключения между экранами.

Рисунок 1 – Главный игровой экран

На рисунке 2 представлен экран прогресса уровня профиля, где игрок может просматривать своё текущее звание, количество монет до следующего ранга и все доступные уровни профиля в игре.

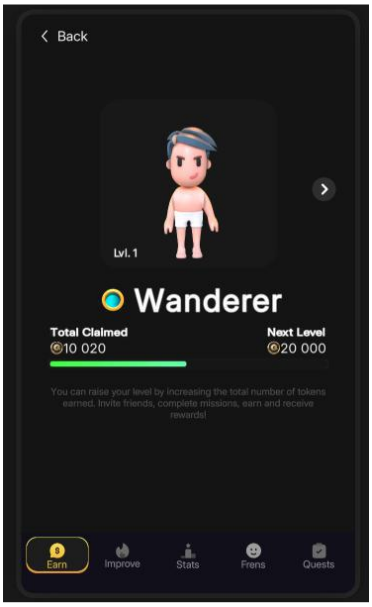

Рисунок 2 – Экран прогресса уровня профиля

Для хранения данных используется PostgreSQL с оптимизированными индексами для частых запросов, что обеспечивает быструю выборку данных при операциях с пользователями, карточками, бустерами и таблицей лидеров [5]. Для обеспечения целостности данных при параллельных операциях применяется уровень изоляции транзакций Repeatable Read, предотвращающий аномалии параллельного доступа. Оптимизация запросов достигается через стратегическое индексирование (B-tree, хэш-индексы, композитные индексы) и пакетную загрузку данных через JOIN-операции для предотвращения проблемы N+1 запросов.

Redis применяется для кеширования таблицы лидеров, ограничения количества запросов к API, а также для управления пользовательскими сессиями и удалённой конфигурацией. Аутентификация реализована через JWT токены с сроком жизни 30 дней, обеспечивающими stateless-верификацию запросов. Для оптимизации сетевого взаимодействия применяется алгоритм батчинга кликов, группирующий множественные события касания в единый пакет перед отправкой на сервер, что снижает нагрузку на сетевое соединение [4].

Тестирование программного средства включало функциональное, модульное, интеграционное, нагрузочное и тестирование безопасности. Модульное тестирование подтвердило корректность алгоритмов расчёта пассивного дохода, динамического мультипликатора и стоимости улучшений карточек. Тестирование безопасности не выявило критических уязвимостей: JWT аутентификация с алгоритмом HS512 обеспечивает надёжную защиту от подделки токенов, а все запросы к базе данных используют параметризованные запросы, полностью предотвращающие SQL-инъекции. Нагрузочное тестирование показало, что Backend стабильно обрабатывает 1000 одновременных запросов со средним временем отклика 45 миллисекунд для простых операций, а кеширование в Redis снижает нагрузку на PostgreSQL на 70 процентов.

Разработанное браузерное игровое программное средство GetRich представляет собой готовое к внедрению решение с микросервисной архитектурой, обеспечивающее высокую производительность и масштабируемость. Применённые архитектурные решения позволяют:

– независимо масштабировать нагруженные компоненты в зависимости от текущей нагрузки;
– обрабатывать высокий объём конкурентных запросов без деградации производительности;
– гибко добавлять новый функционал без нарушения работы существующих модулей;
– обеспечивать отказоустойчивость за счёт изоляции сервисов друг от друга.

Список использованных источников:

1. Хокинг, Дж. Unity в действии. Мультиплатформенная разработка на C# / Дж. Хокинг. – 3-е изд. – СПб. : Питер, 2023. – 448 с.
2. Ньюман, С. Создание микросервисов / С. Ньюман. – 2-е изд. – СПб. : Питер, 2022. – 624 с.
3. Цукалос, М. Golang для профи : работа с сетью, многопоточность, структуры данных и машинное обучение с Go / М. Цукалос. – СПб. : Питер, 2023. – 720 с.
4. Шелл, Дж. Геймдизайн. Как создать игру, в которую будут играть все / Дж. Шелл. – М. : Альпина Паблишер, 2021. – 640 с.
5. Рогов, Е. В. PostgreSQL 16 изнутри / Е. В. Рогов. – М. : ДМК Пресс, 2024. – 664 с.