

Лісневський Д. Є.,
Криворізький національний університет
Музика І. О.,
к.т.н., доцент, Криворізький національний університет

РОЗВАЖАЛЬНА ОНЛАЙН ПЛАТФОРМА З ЗАСТОСУВАННЯМ ТЕХНОЛОГІЇ NODE.JS

Проаналізовано перспективи застосування засобів для створення веб-додатків, таких як JavaScript та Flash. Описано технологію Node.js та принцип роботи деяких ігор, які входять до складу створюваної розважальної онлайн платформи.

Браузерні ігри хоч і не такі круті, як ігри для ПК та консолей, проте вони не гірше приковують до себе увагу гравця. Причин цього доволі багато: вправна робота гейм-дизайнерів, котрі створюють «дозований» ігровий процес, відсутність необхідності завантажувати та встановлювати гру на пристрій, можливість грати будь-де, будь-коли, та з будь-якого пристрою, котрий має веб-браузер та підключення до мережі Інтернет.

Багато браузерних ігор було створено на Flash, який працює на скриптовій мові програмування ActionScript. Flash – це програмна платформа для створення анімацій, ігор та різних інтернет-застосунків. Проте наразі популярність цієї платформи сильно знижується з декількох причин:

- припинення підтримки та розповсюдження платформи у 2020 році;
- проблеми з безпечністю продукту та недовіра користувачів до платформи Flash [1];
- наявність на ринку платформ та мов програмування, які забезпечують більш гнучкий функціонал, вищу продуктивність та мають меншу кількість вразливостей.

Тому наразі набирає популярності створення веб-ігор за допомогою мови JavaScript, котра дає можливість зберігати інформацію про гравця в змінних та обробляти графіку за допомогою елемента HTML5 Canvas. Проте без бази даних та розробки серверної частини така гра не матиме можливості постійно зберігати прогрес

гравця, а продовжити грати з будь-якого збереження на іншому пристрої без спеціальної синхронізації буде неможливо. Також буде відсутня можливість грати через мережу з кимось, що одразу зменшить популярність такої гри. Тому розробники використовують серверні мови програмування для вирішення цих проблем.

Враховуючи те, що основною грою створюваної платформи буде клікер, потрібно використовувати таку технологію, котра буде пристосована для створення високопродуктивних мережових застосунків, щоб максимальна кількість клієнтів могли синхронізувати дані з БД за найменший проміжок часу, адже для створення комфортного процесу гри необхідно буде часто зберігати прогрес гравця. Також треба врахувати, що на платформі будуть доступні такі ігри: «Покер», «Аркада» та «Платформер», котрі також потребують частий обмін інформацією з сервером.

Гра, котра найменше потребує високої продуктивності сервера є «Покер», адже для початку гри необхідно мінімум два гравці, для котрих сервер повинен випадковим чином обрати по 2 карти на гравця та 5 на стіл з 52 доступних у колоді, а потім надсилати результат свого вибору клієнтам у відповідний час.

У свою чергу гравці мають можливість робити ставки, мінімальне та максимальне значення яких має контролюватись сервером, та не перевищувати кількість наразі доступних очок, зароблених у грі «Клікер», значення яких сервер повинен розіслати всім клієнтам, котрі знаходяться в виділеній кімнаті (гравцям та спостерігачам).

Отже, сервер повинен обрати від 9 до 23 випадкових карт та розіслати відповідні дані про 7 карт (2 карти гравця та 5 карт на стіл) кожному клієнту з урахуванням виділеного часу на ходи гравців.

Грою, яка дуже вибаглива до продуктивності сервера є «Платформер», адже сервер повинен згенерувати рівень, розставити ворогів та контролювати їх. Також сервер повинен контролювати виконання дій персонажу гравця, щоб унеможливити нечесний вплив на ігровий процес.

Враховуючи вищезазначене, можна зробити наступний висновок: необхідно використовувати таку платформу, котра надасть функціонал для реалізації подієвої системи синхронізації даних з сервером. На цю роль ідеально підійде платформа Node.js [2], яка є

асинхронно подієвим JavaScript оточення, котре дозволяє програмувати клієнтську та серверну частини однією мовою. До того ж ця технологія забезпечує майже на чверть вищу швидкість у порівнянні з PHP [3].

ВИСНОВКИ

Отже: функціонал мови програмування JavaScript в поєднанні з HTML5 дають можливість створювати клієнтські веб-застосунки будь-якої складності, а при додаванні технології Node.js, котра призначена для створення високопродуктивних мережових додатків та надає можливості використання операцій неблокуючого операційного введення/виведення інформації, можливо створити будь-який клієнт-серверний веб-додаток.

ЛІТЕРАТУРА

1. Про безпечність Flash Player [Електронний ресурс] – Режим доступу до ресурсу: <https://ichip.ru/tekhnologii/pravda-ili-mif-rochemu-flash-player-nebezopasen-225982>.
2. Офіційний сайт Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org>.
3. PHP vs Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/273259/>.

Пузіно М. В.

Криворізький національний університет

Рибальченко О. Г.

старший викладач, Криворізький національний університет

АВТОМАТИЗАЦІЯ РОЗГОРТАННЯ WEB-ДОДАТКІВ З ВИКОРИСТАННЯМ КОНТЕЙНЕРІВ DOCKER

У доповіді були порівняні основні методи розгортання веб-додатків та висвітлені переваги автоматизації цього процесу за допомогою програмного забезпечення Docker.

Розвиток технологій створення веб-додатків, їх супроводу та необхідність автоматизації цих процесів призвели до виникнення низки актуальних питань, пов'язаних з розробкою, доставкою та експлуатацією готового продукту. Основною з них є проблема передачі додатку клієнту та користувачеві, а також забезпечення його