

УДК 004.4

АРХІТЕКТУРНІ ТА ТЕХНОЛОГІЧНІ РІШЕННЯ ПРИ ПРОЄКТУВАННІ БАГАТОКОРИСТУВАЦЬКОЇ ОНЛАЙН-ІГРИ

Захожий А.С., студент

Київський національний університет технологій та дизайну

Волівач А. П., кандидат технічних наук, доцент

Київський національний університет технологій та дизайну

Ключові слова: багатокористувацькі онлайн-ігри, клієнт-серверна архітектура, GodotEngine, GDScript, мікросервіси, бази даних.

Сучасна індустрія інтерактивних ігор характеризується високими вимогами до продуктивності, масштабованості та якості програмних продуктів. Одними з найскладніших інформаційних систем є багатокористувацькі онлайн-ігри, оскільки вони потребують одночасної обробки великих обсягів даних у режимі реального часу та забезпечення стабільної взаємодії значної кількості користувачів. Це зумовлює необхідність розроблення та впровадження ефективних архітектурних і технологічних рішень для створення сучасних ігрових платформ.

Основою побудови надійної багатокористувацької системи є використання клієнт-серверної архітектури із застосуванням концепції авторитетного сервера (AuthoritativeServer). У такій моделі клієнтський застосунок відповідає переважно за рендеринг графіки, відтворення 3D-анімації, візуальних ефектів та обробку дій користувача. Водночас уся критично важлива бізнес-логіка, зокрема розрахунок кількості ушкоджень, обробка колізій, синхронізація станів об'єктів і контроль ігрових характеристик, виконується виключно на стороні сервера [1]. Такий підхід забезпечує захист від несанкціонованих змін клієнтських даних, мінімізує можливість використання читів та гарантує узгодженість стану ігрового середовища для всіх учасників.

Для оптимізації процесу розробки клієнтської частини доцільно використовувати сучасні ігрові рушії з відкритим вихідним кодом. Зокрема, застосування рушія Godot дозволяє ефективно реалізовувати вузлову архітектуру сцен та інтегрувати складні візуальні системи й скелетну анімацію. Важливим підходом до підвищення якості коду ігрової логіки є використання принципурозподілу ключових підсистем між окремими менеджерами. Це сприяє зменшенню залежностей між компонентами, спрощує підтримку програмного коду та запобігає формуванню монолітної кодової бази, забезпечуючи чітке логічне розділення функціональності відповідно до її призначення.

У високонавантажених системах, якими є MMORPG, монолітна серверна архітектура швидко досягає меж своєї продуктивності та масштабованості. Для вирішення цих обмежень застосовується мікросервісний підхід, зокрема у вигляді Microservicesarchitecture. Ігровий бекенд у такій моделі поділяється на незалежні компоненти, кожен з яких

виконує окрему функціональність: сервери авторизації відповідають за автентифікацію користувачів, сервери чату та соціальної взаємодії забезпечують комунікацію між гравцями, а світові сервери (WorldServers) обробляють ігрову логіку, фізику та переміщення в межах окремих ігрових локацій. Це дозволяє підвищити відмовостійкість системи та забезпечує горизонтальне масштабування шляхом динамічного розподілу обчислювальних ресурсів відповідно до концентрації гравців у різних зонах віртуального світу.

Особливої уваги потребує підсистема збереження даних. Ігрові персонажі генерують значний обсяг динамічних параметрів, що постійно змінюються, зокрема показники здоров'я, стан інвентарю, прогрес виконання завдань. Для забезпечення високої швидкодії та надійності доцільним є застосування комбінованого підходу до проектування баз даних [2]. Реляційні системи управління базами даних використовуються для надійного збереження критичної інформації про облікові записи користувачів, тоді як NoSQL-рішення забезпечують ефективне збереження та обробку тимчасових даних поточного стану гравця під час активної ігрової сесії. Додатково важливим аспектом є реалізація алгоритмів мережевої синхронізації, що забезпечують узгоджене оновлення стану ігрового світу між клієнтською та серверною частинами. Застосування таких технологій дозволяє підтримувати стабільність ігрового процесу навіть за умов нестабільного інтернет-з'єднання та мережевої затримки.

Для реалізації розглянутих підходів було розроблено архітектуру багатокористувацької онлайн-гри ZerixWord (рис. 1).

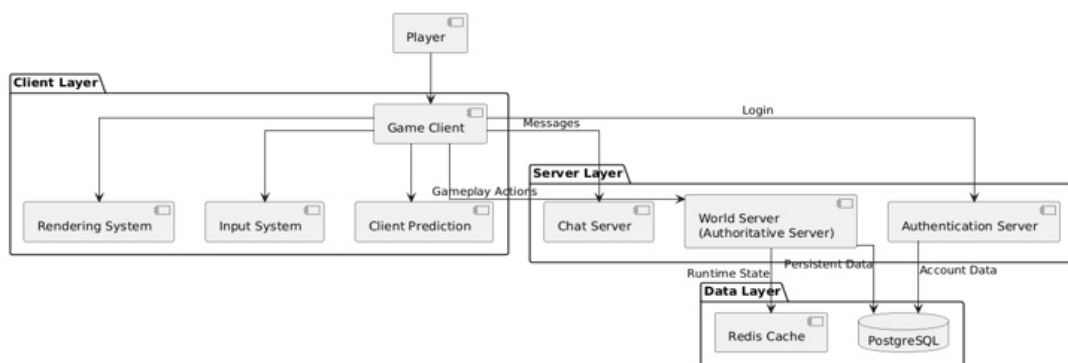


Рисунок 1 – Архітектура багатокористувацької онлайн-гри ZerixWord

У запропонованій архітектурі застосовано принцип авторитетного сервера (AuthoritativeServer), за якого важлива ігрова логіка обробляється виключно на стороні сервера, тоді як клієнтська частина відповідає переважно за візуалізацію, обробку введення користувача та забезпечення інтерфейсу взаємодії з грою.

Список використаних джерел

1. Glazer J., Madhav S. Multiplayer Game Programming: Architecting Networked Games // Addison-Wesley Professional. – 2015. – 304 p.
2. Gregory J. Game Engine Architecture, Third Edition // CRC Press. – 2018. – 1240 p.