

УДК 004.514

## **ДОСЛІДЖЕННЯ МЕТОДІВ ОПТИМІЗАЦІЇ ТА ПРОЕКТУВАЛЬНИХ РІШЕНЬ ДЛЯ СТВОРЕННЯ СКЛАДНИХ ІГРОВИХ ПРОГРАМНИХ СИСТЕМ**

Косенко Б. А.

Науковий керівник – к.т.н., доц. Мар'їн С. О.

Харківський національний університет радіоелектроніки, каф. ПІ,  
м. Харків, Україна

e-mail: [borys.kosenko@nure.ua](mailto:borys.kosenko@nure.ua)

This research is dedicated to the exploration of optimization methods and design solutions in the development of complex gaming software systems. The research aims to investigate contemporary trends in the field of game system development and identify optimal optimization strategies to ensure high performance and gameplay quality. The Unity development environment will be utilized for this research, given its extensive built-in tools for project exploration and analysis. The robust customization capabilities of Unity also allow for the development of new tools. Additionally, the research plans to examine how various design solutions impact the development speed and fault tolerance of the gaming system.

Ігрова індустрія, нинішній гігант із оборотом у 200 мільярдів доларів, можливо, є сьогодні одним із найважливіших та інноваційних секторів технологій. Його важливість для культури, соціальних мереж та розваг неможливо недооцінити. Термін «індустрія розваг» більше не використовується тільки для Голлівуду та кіноіндустрії, оскільки ігри тепер надають одну з найбільш захоплюючих форм розваг більш ніж трьом мільярдам людей по всьому світу. Сучасна ігрова індустрія, створена постійно зростаючими ігровими компаніями, розширює межі технологій, створюючи передові та захоплюючі розваги.

Зі збільшенням кількості гравців зростає і конкуренція у сфері розробки ігор, що у свою чергу сприяє появі нових ігрових програм. В наш час практично всі можливі ігрові механіки вже були створені кимось і реалізовані в одному з тисяч проектів. Однак, просто надати якісну гру вже не достатньо для завоювання користувальницької уваги.

Сьогодні гравців необхідно дивувати, пропонуючи їм щось унікальне та захоплююче. Це призводить до того, що ігри стають все більш масштабними та амбітними у своїх ідеях та реалізації. Збільшується складність як процесу розробки, а й подальшої підтримки таких гігантських ігрових проектів.

Розробка та підтримка АА та ААА проектів є викликом для багатьох розробників, особливо в умовах зростаючої конкуренції та вимогливих смаків гравців[1]. Відсутність якісної оптимізації може призвести до невдалого випуску гри на ринок.

Отже оптимізація продуктивності ігрового проекту — один із вирішальних факторів для масштабування, оскільки різні пристрої мають різні апаратні можливості та обмеження. Продуктивність означає, наскільки плавно та ефективно працює гра, без затримок, збоїв та розрядки акумулятора. Щоб оптимізувати продуктивність гри необхідно використовувати різні методи, такі як скорочення кількості викликів малювання, оптимізація розміру та якості ресурсів, використання ефективних алгоритмів та структур даних, а також реалізація кешування та об'єднання в пули [2]. Також необхідно використовувати інструменти профілювання для вимірювання та аналізу продуктивності проекту на різних платформах та виявлення потенційних вузьких місць та проблем.

Підтримка таких проектів також є значущим аспектом. Після випуску розробники повинні надавати постійні оновлення, виправляти помилки та взагалі удосконалювати гру, щоб утримати та залучати аудиторію [3].

Швидкість впровадження нових механік та вирішення наявних помилок залежить від обраного комплексу проектних рішень. Це охоплює всі частини проекту: створення та налаштування ігрових об'єктів, їх управління, впровадження залежностей, підхід до розподілу бізнес-логіки, управління ігровим циклом та станами, складність ієрархії класів, абстракції ігрових об'єктів, отримання доступу до інших компонентів у об'єкті чи сцені, складність компонентів ігрових об'єктів, характеристики об'єктів у грі, модифікації, доступ до тимчасових та постійних даних, серіалізація даних та їх безпека, передача даних між об'єктами у сцені та між сценами, покриття тестами та інше.

При цьому вибір одних рішень може перекривати можливість використання інших. Наприклад, якщо бізнес-логіка буде в ігрових компонентах, прикріплених до об'єктів на сцені, то це викличе труднощі для покриття такої логіки автоматичними текстами. У такому разі доведеться розділити логіку та візуальну модель, що у свою чергу збільшить час, необхідний на впровадження нових механік, але зменшить ризик виникнення помилок.

Також при проектуванні потрібно розробляти рішення (правила, підходи, гайдлайни) не тільки для ігрової логіки, але і для всього, з чим проект пов'язаний: назва файлів і папок, структура зберігання асетів, структура збірок, утиліти для розробки, тестування, робота із системою контролю версії, білд проекту та його деплой, робота зі сторями та вилівка хотфіксів, моніторинг метрик, помилок та реагування на них [4].

Розробка додаткових сервісів та шарів абстракції, кастомних утиліт для редактора, покриття тестами може зайняти значний час розробки та окупити його лише у довгостроковій перспективі. Іноді цілі команди розробників займаються проектуванням заради проектування там, де це зовсім не потрібно забуваючи, що в першу чергу вони розробляють гру.

Саме тому важливо знаходити баланс між дійсно необхідними проектними рішеннями та тими, що на даний момент не несуть практичної користі.

Отже, грамотний підхід до проектування продукту може сприяти швидкому впровадженню нових механік, уникненню основної проблеми швидкої розробки – появи великої кількості багів. У такому випадку розробники матимуть можливість витратити більше часу на вдосконалення ігрових механік проекту.

Оптимізація продуктивності має безпосередній вплив на фінансовий успіх проекту, оскільки вона визначає кількість пристроїв, на яких гра зможе оптимально функціонувати. Профайлінг проекту, хоча і є тривалим і складним процесом, є необхідною складовою розробки, яку необхідно враховувати впродовж всього проекту, а не залишати на останній момент.

Якісно розроблений оптимізаційний план із систематичним усуненням вузьких місць у проекті буде значно ефективнішим, ніж хаотичні виправлення. Це сприятиме вдосконаленню продукту та забезпечить його стабільну роботу, що відіграє ключову роль у забезпеченні задоволення користувачів та, в кінцевому рахунку, фінансовому успіху проекту.

Список використаних джерел:

1. Jason Gregory. Game Engine Architecture. 3rd edition. CRC Press, 2018. 1240 p.
2. Andrew Rollings, Dave Morris. Game Architecture and Design. 2nd edition. New Riders, 2004. 926 p.
3. Julian Gold. Object-oriented Game Development. Pearson Education, 2004. 426 p.
4. Daniel Sánchez, Crespo Dalmau. Core Techniques and Algorithms in Game Programming. New Riders, 2004. 854 p.