

ПРОБЛЕМИ ТА ВАЖЛИВІСТЬ ОПТИМІЗАЦІЇ КОДУ В РОЗРОБЦІ ВІДЕОІГОР

А.І. Коробкова¹, О.М.Рисований²

¹ магістрант кафедри комп'ютерної інженерії та програмування, НТУ «ХПІ», Харків, Україна

alyonaserhienko99@gmail.com

² професор кафедри комп'ютерної інженерії та програмування, канд. техн. наук, НТУ «ХПІ», Харків, Україна

У сучасній індустрії розробки відеоігор питання продуктивності програмного забезпечення має вирішальне значення. Оптимізація коду є одним із ключових етапів, що безпосередньо впливає на швидкість, стабільність і якість ігрового процесу. Зі зростанням вимог до графіки, фізики, штучного інтелекту та мережевих можливостей постає потреба в ефективному використанні ресурсів системи, особливо під час розробки мультиплатформних проєктів [1].

Відсутність належної оптимізації призводить до зниження частоти кадрів, перевантаження процесора та графічного процесора, збільшення часу відгуку і, як наслідок, погіршення користувацького досвіду [2-3]. Водночас надмірна або неправильна оптимізація може ускладнювати підтримку коду, зменшувати його гнучкість і збільшувати ризик помилок.

Крім того, важливо враховувати баланс між продуктивністю та підтримуваністю коду. Надмірна оптимізація може призвести до втрати зрозумілості структури програми, що ускладнить подальші зміни чи командну роботу. Тому при розробці ігор варто дотримуватися принципів чистого коду, розділення логіки та візуалізації, а також застосовувати профайлер для контролю продуктивності під час усього циклу розробки.

Метою цього дослідження є створення невеликої 2D-гри з акцентом на написання компактного, читабельного та оптимізованого коду без витоків пам'яті. У межах роботи здійснюється аналіз і практичне застосування підходів до підвищення ефективності виконання скриптів у середовищі Unity, зокрема раціональне використання компонентів рушія, оптимізація об'єктів сцени, роботи з оновленнями та управління пам'яттю.

Дослідження спрямоване на визначення практик, що дозволяють поєднати якісну архітектуру коду, стабільну роботу гри та ефективне використання ресурсів системи. Отримані результати можуть бути корисними для інді-розробників та початківців у сфері геймдеву, які працюють із Unity та прагнуть створювати продуктивні й підтримувані проєкти.

Список літератури:

1. Yan H., Sui Y., Chen S., Xue J. Automated memory leak fixing on value-flow slices for c programs; Proceedings of the 31st Annual ACM Symposium on Applied Computing; Pisa, Italy. 4–8 April 2016; pp. 1386–1393.

2. Santos RES, Magalhães CVC, Capretz LF, Correia-Neto JS, Da Silva FQB, Saher A (2018) Computer games are serious business and so is their quality: particularities of software testing in game development from the perspective of practitioners. Proc 12th ACM/IEEE Int Symp Empir Softw Eng Meas [Internet]. ACM, Oulu, pp 1–10.

3. Ma X., Yan J., Wang W., Yan J., Zhang J., Qiu Z. Detecting Memory-Related Bugs by Tracking Heap Memory Management of C++ Smart Pointers; Proceedings of the 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE); Melbourne, Australia. 15–19 November 2021; pp. 880–891.