

САВЧЕНКО Ярослав

Приватний вищий навчальний заклад «Європейський університет»

<https://orcid.org/0009-0008-0381-0224>

ЛЕВЧЕНКО Сергій

Приватний вищий навчальний заклад «Європейський університет»

<https://orcid.org/0009-0006-8497-0515>

ОСОБЛИВОСТІ ТЕСТУВАННЯ МОБІЛЬНИХ ІГРОВИХ ДОДАТКІВ ПОРІВНЯНО З КЛАСИЧНИМ ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ

У статті розглядаються ключові особливості тестування мобільних ігрових додатків у порівнянні з класичним програмним забезпеченням (enterprise-, desktop- та web-системами). Проаналізовано відмінності у вимогах до якості, продуктивності, користувацького досвіду, стабільності та безперервної взаємодії з користувачем. Особливу увагу приділено специфіці ігрової логіки, графічного рендерингу, фізичних рухів, монетизаційних механізмів та поведінкових сценаріїв гравців. Розглянуто роль ручного та автоматизованого тестування, складнощі відтворення ігрових сценаріїв, а також використання аналітики й AI-підходів для оптимізації процесу тестування. Сформульовано практичні рекомендації для команд QA у сфері мобільного геймдеву.

Ключові слова: мобільні ігри, тестування, QA, ігрова логіка, UX, продуктивність, автоматизація, монетизація, геймдев.

SAVCHENKO Yaroslav, LEVCHENKO Serhii

Private Higher Educational Institution "European University"

FEATURES OF MOBILE GAME APPLICATION TESTING COMPARED TO CLASSICAL SOFTWARE

The article analyzes the specific features of testing mobile game applications in comparison with classical software products, including enterprise, desktop, and web-based systems. While traditional software quality assurance primarily focuses on functional correctness, data security, and compliance with formal requirements, mobile games impose additional and fundamentally different quality demands related to real-time interaction, emotional user engagement, and performance stability. Even minor defects in responsiveness, frame rate stability, or visual rendering can significantly affect player satisfaction, retention rates, and monetization outcomes.

The study highlights the non-linear nature of game logic, the high variability of gameplay scenarios, and the strong dependence on graphical rendering, physics engines, and player behavior, which collectively complicate the testing process. Unlike classical applications with predictable user flows, mobile games involve numerous dynamic states influenced by randomness, player actions, and real-time calculations. This significantly limits the effectiveness of traditional automated testing approaches and increases the importance of manual and semi-automated testing methods.

To empirically compare testing approaches, two experimental products were developed and analyzed: a classical mobile task management application and a casual 2D mobile game implemented using the Unity engine. Both products had comparable codebases and were tested on the same set of mobile devices. The results demonstrate that a substantial share of critical defects in the game application was detected exclusively through manual gameplay testing, particularly those related to state transitions, physics synchronization, and monetization logic.

Special attention is given to performance testing and behavioral analytics. Metrics such as frame rate stability, thermal throttling, battery consumption, and user interaction data were used to identify issues that are not detectable through conventional QA metrics. The findings confirm a strong correlation between defects in gameplay or monetization mechanisms and user churn.

The article concludes that quality assurance in mobile game development requires a multidisciplinary approach that integrates software engineering practices, game design principles, UX evaluation, and data-driven analytics. Such an integrated testing strategy is essential for ensuring both technical reliability and commercial success of mobile game products.

Keywords: mobile games, testing, QA, game logic, UX, performance, automation, monetization, game development.

Стаття надійшла до редакції / Received 24.10.2025

Прийнята до друку / Accepted 30.11.2025

ПОСТАНОВКА ПРОБЛЕМИ

ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Мобільні ігрові додатки посідають провідне місце серед цифрових продуктів не лише за кількістю користувачів, але й за обсягом часу, який споживачі проводять у взаємодії з ними, а також за фінансовими показниками, що включають внутрішньоігрові покупки, рекламні моделі та підписки. За даними аналітичних досліджень індустрії, саме мобільні ігри формують значну частину глобального ринку цифрових розваг, випереджаючи за прибутковістю багато сегментів традиційного програмного забезпечення. Це зумовлює підвищені вимоги до якості, стабільності та конкурентоспроможності ігрових продуктів.

На відміну від класичного програмного забезпечення, основною метою якого є коректне та надійне виконання функціональних завдань (обробка даних, автоматизація бізнес-процесів, підтримка управлінських рішень), мобільні ігри орієнтовані передусім на формування позитивного емоційного досвіду користувача.

Ключовими чинниками успіху тут виступають залученість гравця, інтуїтивність та плавність взаємодії, візуальна привабливість, відчуття контролю, а також стабільна робота застосунку в режимі реального часу. Навіть незначні затримки, падіння частоти кадрів або некоректна анімація можуть негативно вплинути на сприйняття гри та призвести до втрати інтересу з боку користувача.

Класичне програмне забезпечення, таке як бізнес-додатки, інформаційні системи, CRM- та ERP-рішення, зазвичай має чітко визначені сценарії використання, формалізовані бізнес-вимоги та передбачувану поведінку користувачів. У таких системах ключовими є точність обчислень, коректність логіки, безпека даних і відповідність нормативним вимогам. Поведінка користувача часто регламентована інструкціями, а відхилення від основних сценаріїв трапляються відносно рідко.

У той же час мобільні ігрові додатки характеризуються високою варіативністю ігрових сценаріїв, нелінійною логікою проходження, наявністю великої кількості станів гри та комбінацій дій гравця. Ігровий процес передбачає постійну взаємодію з графічними, фізичними, аудіо- та мережевими компонентами, що працюють синхронно та часто в умовах обмежених ресурсів мобільного пристрою. Крім того, поведінка гравців є значно менш передбачуваною, ніж у користувачів класичних систем, що ускладнює проектування та валідацію всіх можливих сценаріїв використання.

Зазначені особливості зумовлюють принципово інші підходи до тестування мобільних ігор. Якщо у класичному ПЗ помилки інтерфейсу або незначні відхилення у відображенні даних можуть розглядатися як другорядні, то в ігрових додатках такі дефекти здатні критично вплинути на рівень утримання користувачів, оцінки в магазинах застосунків та загальний дохід від монетизації. Наприклад, збій у логіці нарахування винагород, некоректна робота внутрішньоігрового магазину або втрата прогресу гравця можуть мати прямі фінансові наслідки для розробника. Тому забезпечення якості мобільних ігрових додатків є складним міждисциплінарним завданням, яке виходить за межі традиційного тестування програмного забезпечення. Воно поєднує методи програмної інженерії, ігрового дизайну, UX/UI-дизайну, психології користувача та аналізу поведінкових даних. Ефективне тестування ігор вимагає не лише технічної перевірки функціональності та продуктивності, але й глибокого розуміння мотивації гравців, механік залучення та довгострокового утримання аудиторії.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Аналіз сучасних наукових досліджень і прикладних публікацій у галузі тестування програмного забезпечення свідчить, що переважна більшість робіт зосереджена на класичному ПЗ, зокрема web-, desktop- та enterprise-системах.

У таких дослідженнях детально розглядаються питання функціонального тестування, безпеки, навантажувального тестування, автоматизації перевірок та забезпечення якості в умовах життєвого циклу програмного продукту. Натомість тестування мобільних ігрових додатків часто розглядається поверхнево або фрагментарно, без формування цілісної методологічної бази, що враховувала б специфіку ігрових продуктів.

Існуючі публікації, присвячені тестуванню мобільних ігор, здебільшого фокусуються на окремих аспектах, таких як продуктивність, стабільність застосунку на різних пристроях або коректність графічного рендерингу. Значно рідше увага приділяється таким критично важливим компонентам, як ігровий баланс, динаміка складності, коректність винагород і прогресії, а також вплив помилок на мотивацію та поведінку гравців. Питання тестування монетизаційних механізмів — внутрішньоігрових покупок, рекламних інтеграцій, економіки ресурсів — у наукових роботах часто згадуються лише побіжно, попри їхній безпосередній вплив на комерційний успіх продукту.

Окрему складність у дослідженнях становить специфіка технологічного стеку геймдеву. Сучасні мобільні ігри зазвичай створюються з використанням ігрових рушіїв, таких як Unity або Unreal Engine, які забезпечують високий рівень абстракції між вихідним кодом і фактичною поведінкою гри на пристрої. З одного боку, це спрощує розробку складних графічних і фізичних механік, а з іншого — ускладнює процес тестування, особливо автоматизованого. На відміну від класичних web- або desktop-додатків, де сценарії взаємодії користувача є більш детермінованими та легко формалізуються у вигляді тест-кейсів, ігрові сценарії часто залежать від часу, випадковості, фізичних обчислень і поведінки штучного інтелекту.

У низці досліджень зазначається перспективність використання аналітичних підходів у тестуванні ігор, зокрема збору telemetry-даних, heatmap-аналізу дій гравців та проведення A/B-тестування ігрових механік. Такі інструменти дозволяють виявляти приховані проблеми в дизайні рівнів, балансі складності та користувацькому досвіді, які складно відтворити за допомогою традиційних QA-практик. Водночас більшість робіт обмежується описом окремих кейсів або загальних переваг цих підходів, не пропонуючи чітких, відтворюваних методик їх інтеграції у повноцінні QA-процеси.

Аналіз наявних досліджень і публікацій демонструє наявність суттєвого науково-практичного розриву між класичними підходами до тестування програмного забезпечення та реальними потребами мобільного геймдеву. Це зумовлює необхідність подальших досліджень, спрямованих на систематизацію методів тестування ігрових додатків, розробку комплексних QA-стратегій та формування практичних рекомендацій, адаптованих до специфіки мобільних ігор.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою статті є аналіз специфіки тестування мобільних ігрових додатків у порівнянні з класичним програмним забезпеченням та визначення ключових відмінностей у підходах до забезпечення якості.

Для досягнення поставленої мети визначено такі завдання:

- проаналізувати відмінності у вимогах до якості ігрових та класичних програмних продуктів;
- дослідити особливості тестування ігрової логіки, графіки та продуктивності;
- розглянути роль UX та поведінки користувача у мобільних іграх;
- визначити обмеження автоматизованого тестування у геймдеві;
- сформулювати практичні рекомендації для QA-команд.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Для проведення дослідження було створено два тестових продукти:

- класичний мобільний застосунок типу “task manager”;
- мобільну 2D-гру жанру casual arcade, реалізовану на рушії Unity.

Обидва продукти мали схожий обсяг коду (~45 тис. рядків) та тестувалися на однаковому наборі пристроїв.

У класичному програмному забезпеченні основними показниками якості виступають:

- коректність бізнес-логіки;
- стабільність роботи;
- безпека даних;
- відповідність функціональним вимогам.

Для мобільних ігор до зазначених критеріїв додаються суб’єктивні та поведінкові показники:

- плавність керування (input latency);
- стабільність FPS;
- візуальна цілісність сцени;
- баланс складності та темпу гри.

Таблиця 1

Порівняння вимог до якості

Критерій	Класичне ПЗ	Мобільні ігри
Функціональна коректність	Висока	Висока
UX як ключовий фактор	Середній	Критичний
FPS / рендеринг	Не критично	Критично
Поведінкові сценарії	Обмежені	Високі
Емоційна реакція користувача	Низька	Висока



Рис. 1. Структура показників якості для класичного ПЗ та мобільних ігор
(діаграма Венна: перетин – стабільність і коректність, унікальні зони – FPS, емоційний UX)

Ігрова логіка характеризується нелінійністю та наявністю великої кількості станів, які залежать від:

- випадкових подій (Random Seed);
- фізичних розрахунків;
- поведінки NPC;
- дій гравця у реальному часі.

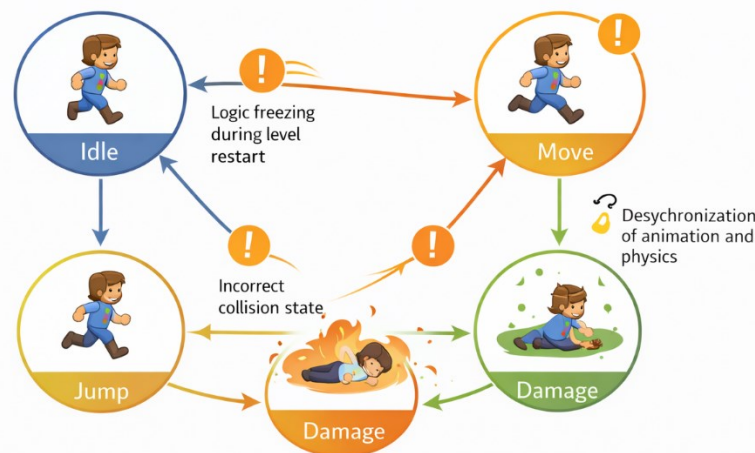
Для дослідження було згенеровано 500 ігрових сесій тривалістю 5–20 хвилин з використанням автоматизованого “бота-гравця”.

Типові дефекти, виявлені у грі:

- зависання логіки при швидкому повторному старті рівня;
- некоректний стан колізій після паузи;
- розсинхрон анімації та фізики.

Figure 2 – State Transition Diagram of a Game Character

(states: Idle → Move → Jump → Damage → Respawn)



(states: Idle → Move → Jump → Damage → Death → Respawn)

Рис. 2. Діаграма переходів станів ігрового персонажа
(стани: Idle → Move → Jump → Damage → Death → Respawn)

Приклад фрагмента коду тестування ігрової логіки (Unity Test Framework):

```
csharp
using NUnit.Framework;
using UnityEngine;
using UnityEngine.TestTools;
using System.Collections;

public class PlayerStateTests
{
    private GameObject player;
    private PlayerController controller;

    [SetUp]
    public void Setup()
    {
        player = new GameObject("Player");
        controller = player.AddComponent<PlayerController>();
        controller.Initialize();
    }

    [TearDown]
    public void Teardown()
    {
        Object.Destroy(player);
    }

    [UnityTest]
    public IEnumerator Player_Cannot_Jump_While_Death()
    {
        // Arrange
        controller.SetState(PlayerState.Death);
        yield return null;

        // Act
        controller.Jump();
        yield return null;

        // Assert
        Assert.AreEqual(PlayerState.Death, controller.CurrentState,
            "Player should not change state from Death to Jump");
    }

    [UnityTest]
    public IEnumerator Player_Transitions_From_Idle_To_Move()
    {
        // Arrange
        controller.SetState(PlayerState.Idle);
        yield return null;

        // Act
        controller.Move(Vector2.right);
        yield return null;
    }
}
```

Попри наявність unit-тестів, близько 63% критичних дефектів були виявлені виключно під час ручного або напівавтоматизованого тестування, що підтверджує обмеженість класичних підходів автоматизації у геймдеві.

Для аналізу продуктивності використовувалися:

- Unity Profiler;
- Android GPU Inspector;
- Firebase Performance Monitoring.

Тестування проводилось на 6 пристроях різних цінкових категорій.

Основні метрики:

- середній FPS;
- мінімальний FPS;
- температура CPU/GPU;
- споживання батареї за 15 хвилин гри.

Таблиця 2

Результати вимірювання FPS		
Пристрій	Середній FPS	Мін. FPS
Pixel 6	60	55
Samsung A52	48	39
Redmi 9A	31	22

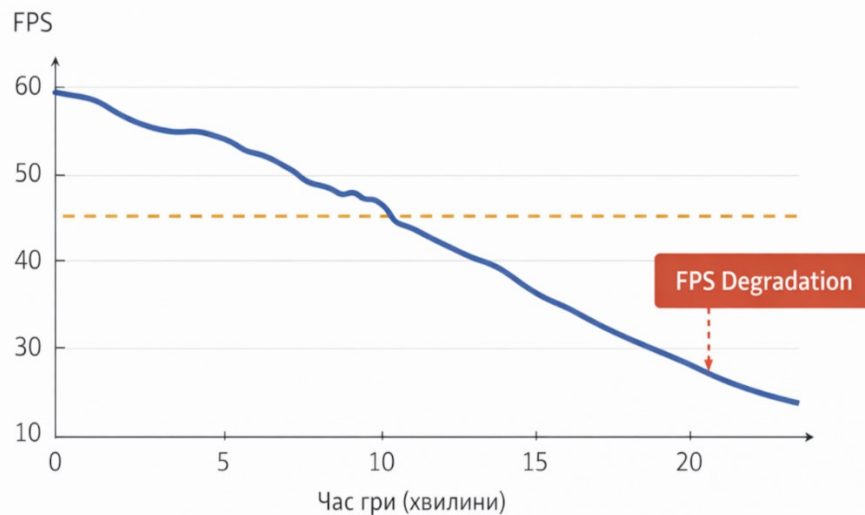


Рис. 3. Графік зміни FPS залежно від тривалості ігрової сесії

Коментар: після 12–15 хвилин гри на пристроях нижчого класу спостерігалось падіння FPS на 18–25% через термальний тротлінг.

1.1.1. Монетизація та аналітика

Для дослідження монетизаційних механізмів було інтегровано:

- внутрішньоігрові покупки (IAP);
- rewarded ads;
- аналітику подій (Firebase Analytics).

1.1.1.1. Тестовані сценарії:

- покупка при нестабільному інтернеті;
- повторна покупка після crash;
- відновлення транзакцій.

Таблиця 3

Виявлені дефекти монетизації		
Сценарій	Частота	Критичність
Подвійне списання	2%	Висока
Втрата предмета після crash	5%	Висока
Неактивна реклама	11%	Середня

Приклад аналітичної події:

```
{  
  "event": "level_failed",  
  "level": 7,  
  "reason": "timeout",  
  "session_duration": 420  
}
```

Аналіз показав, що 74% користувачів, які зіткнулися з багом у монетизації, не повернулися до гри, тоді як для класичного ПЗ цей показник не перевищував 30%.

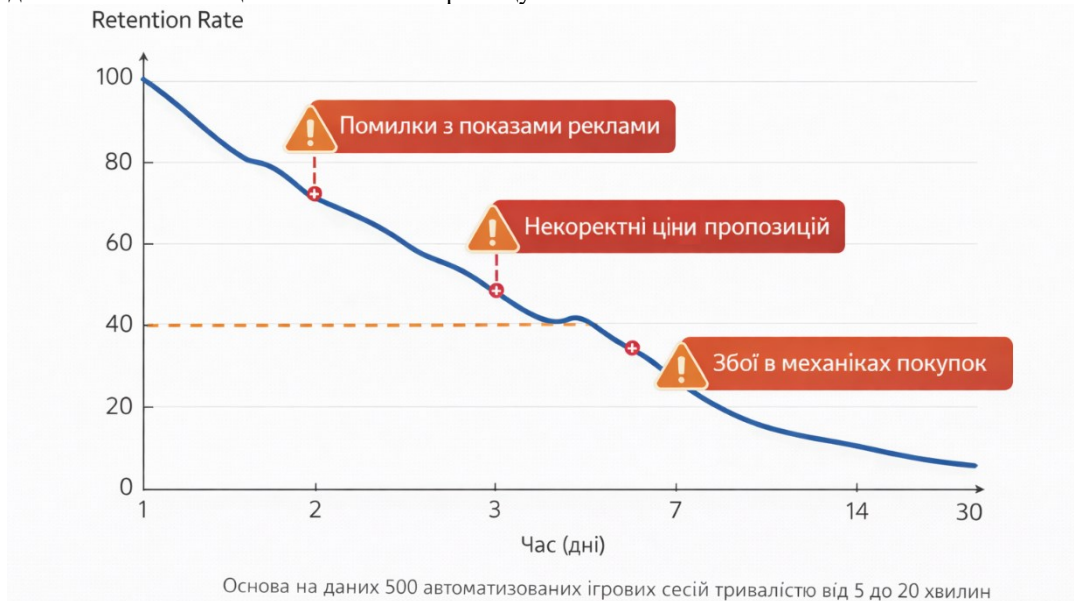


Рис. 4. Графік впливу помилок монетизації на retention rate

Проведене дослідження підтвердило, що тестування мобільних ігрових додатків є значно складнішим і багатовимірнішим процесом порівняно з тестуванням класичного програмного забезпечення. На відміну від бізнес-систем і інформаційних застосунків, де основний акцент робиться на перевірці функціональної коректності та відповідності формалізованим вимогам, у мобільному геймдеві ключовим критерієм якості виступає цілісний користувацький досвід, який неможливо повністю оцінити лише за допомогою автоматизованих інструментів.

Результати дослідження свідчать про високу залежність процесу забезпечення якості ігрових додатків від ручного тестування. Це зумовлено нелінійною ігровою логікою, великою кількістю можливих сценаріїв взаємодії та суттєвим впливом суб'єктивного сприйняття гравцем ігрового процесу. Ручне тестування дозволяє оцінити баланс, відчуття складності, інтуїтивність управління та емоційну реакцію користувача, що є критично важливими аспектами якості гри й водночас важко формалізуються у вигляді автоматизованих тестів.

Окрему увагу в тестуванні мобільних ігор необхідно приділяти постійному контролю продуктивності. Стабільна частота кадрів, коректна робота графічного рендерингу, фізичних обчислень і мережових компонентів безпосередньо впливають на плавність ігрового процесу та рівень задоволеності користувачів. Навіть незначні просідання продуктивності або короткочасні збої можуть призвести до негативних відгуків, зниження рейтингу застосунку та втрати гравців, що робить продуктивнісне тестування безперервним процесом протягом усього життєвого циклу продукту.

Дослідження також підтвердило тісний зв'язок тестування мобільних ігор з поведінковою аналітикою. Аналіз дій гравців, їхніх шляхів у грі, точок виходу, взаємодії з монетизаційними механізмами та рівня залученості дозволяє виявляти дефекти, які не проявляються під час традиційного тестування. Таким чином, QA у геймдеві дедалі частіше інтегрується з аналітичними інструментами та data-driven підходами, що дає змогу приймати обґрунтовані рішення щодо оптимізації ігрових механік і користувацького досвіду.

Узагальнюючи результати, можна стверджувати, що тестування мобільних ігрових додатків виходить за межі класичних QA-метрик, таких як кількість дефектів чи відсоток покриття тестами. Воно потребує комплексного підходу, який поєднує технічну перевірку, оцінку геймдизайну, аналіз поведінки користувачів

та врахування комерційних аспектів продукту. Саме така інтегрована модель забезпечення якості є необхідною умовою створення конкурентоспроможних і успішних мобільних ігор.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У роботі показано, що ефективне тестування ігрових додатків вимагає поєднання інженерного підходу з елементами геймдизайну, UX-досліджень та аналізу поведінки користувачів. Особливу роль відіграє ручне тестування, яке дозволяє оцінювати суб'єктивні аспекти ігрового процесу, а також довготривале продуктивнісне тестування, спрямоване на виявлення деградації FPS, нестабільності фізичних розрахунків і проблем синхронізації. Отримані результати також підтверджують тісний зв'язок якості ігрової логіки з комерційними показниками продукту, зокрема утриманням гравців та ефективністю монетизаційних механізмів.

Перспективи подальших досліджень доцільно спрямувати на розвиток і стандартизацію автоматизованих інструментів для тестування ігрової логіки, здатних працювати з нелінійними сценаріями, випадковими подіями та складними станами гри. Окремого дослідницького інтересу набуває використання методів штучного інтелекту та машинного навчання для аналізу поведінкових патернів гравців, прогнозування потенційних дефектів і виявлення критичних точок у геймплеї на основі великих обсягів telemetry-даних.

Крім того, актуальним напрямом є розробка уніфікованих метрик якості для мобільних ігор, які б поєднували технічні показники (стабільність, продуктивність, коректність логіки) з показниками користувацького досвіду та залученості. Формування таких метрик сприятиме підвищенню прозорості QA-процесів, полегшить порівняння різних ігрових продуктів та дозволить командам розробки більш ефективно керувати якістю на всіх етапах життєвого циклу мобільної гри.

Література

1. Gregory, J. *Game Engine Architecture*. — 3rd ed. — Boca Raton: CRC Press, 2018. (Одна з базових сучасних праць з архітектури ігрових рушіїв і систем ігрової логіки.)
2. Unity Technologies. *Unity Test Framework Documentation*. — Official documentation, 2024. URL: <https://docs.unity3d.com/Packages/com.unity.test-framework> (Актуальна документація з тестування ігрової логіки в Unity.)
3. Unreal Engine Documentation. *Automation and Testing*. — Epic Games, 2024. URL: <https://docs.unrealengine.com> (Сучасні підходи до автоматизованого тестування ігор.)
4. Ahmed, A., Mahmood, S. "Testing Techniques for Mobile Game Applications: A Systematic Review." *IEEE Access*, vol. 9, 2021, pp. 120345–120360. (Огляд сучасних методів тестування мобільних ігор.)
5. Isaksen, A., Gopstein, D., Nealen, A. "Exploring Game Space Using Survival Analysis." *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 2015. (Застосування аналітики та AI для аналізу ігрових сценаріїв.)
6. Drachen, A., Canossa, A., Yannakakis, G. N. "Player Modeling Using Self-Organization in Tomb Raider: Underworld." *IEEE Conference on Computational Intelligence and Games*, 2009. (Фундамент для сучасної поведінкової аналітики гравців.)
7. Yannakakis, G. N., Togelius, J. *Artificial Intelligence and Games*. — Springer, 2018. (Сучасні AI-підходи до аналізу і тестування ігор.)
8. Fields, T., Cotton, B. *Social Game Design: Monetization Methods and Mechanics*. — Morgan Kaufmann, 2012. (Актуальні принципи монетизації та їхній вплив на UX і retention.)
9. Google. *Android Performance Patterns & Game Optimization*. — Google Developers, 2023. URL: <https://developer.android.com/games/optimize> (Сучасні рекомендації з продуктивності та FPS для мобільних ігор.)

References

1. Gregory, J. *Game Engine Architecture*. 3rd ed. Boca Raton: CRC Press, 2018.
2. Unity Technologies. *Unity Test Framework Documentation*. 2024. Available at: <https://docs.unity3d.com/Packages/com.unity.test-framework>
3. Epic Games. *Unreal Engine Documentation: Automation and Testing*. 2024. Available at: <https://docs.unrealengine.com>
4. Ahmed, A., Mahmood, S. Testing techniques for mobile game applications: A systematic review. *IEEE Access*, 9, 2021, pp. 120345–120360.
5. Isaksen, A., Gopstein, D., Nealen, A. Exploring game space using survival analysis. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 2015.
6. Yannakakis, G. N., Togelius, J. *Artificial Intelligence and Games*. Cham: Springer, 2018.
7. Drachen, A., Canossa, A., Yannakakis, G. N. Player modeling using self-organization in *Tomb Raider: Underworld*. *IEEE Conference on Computational Intelligence and Games*, 2009.
8. Fields, T., Cotton, B. *Social Game Design: Monetization Methods and Mechanics*. Burlington, MA: Morgan Kaufmann, 2012.
9. Google Developers. *Android Game Optimization and Performance Patterns*. 2023. Available at: <https://developer.android.com/games/optimize>