

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр

(назва освітнього ступеня)

на тему: Вдосконалення методології ідентифікації дефектів у
багатокористувацьких іграх на прикладі Off the Grid.

Виконав: студент VI курсу, групи СНм-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

Мамрош В.С.
(підпис) (прізвище та ініціали)

Керівник
(підпис) Небесний Р.М.
(прізвище та ініціали)

Нормоконтроль
(підпис) Никитюк В.В.
(прізвище та ініціали)

Завідувач
кафедри
(підпис) Боднарчук І.О.
(прізвище та ініціали)

Рецензент
(підпис) Ревнюк О.А.
(прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 17 » листопада 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Мамрошу Віталію Степановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Вдосконалення методології ідентифікації дефектів у багатокористувацьких іграх на прикладі Off the Grid.

Керівник роботи Небесний Руслан Михайлович, доктор філософії, доцент кафедри КН.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 27 » листопада 2025 року № 4/7-1042

2. Термін подання студентом завершеної роботи 21 грудня 2025р.

3. Вихідні дані до роботи Off the Grid, наявні методи тестування, інструменти тестування.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналітичний огляд розвитку відеоігор та підходів до ідентифікації. 1.1 Витоки відеоігор та етапи їх розвитку. 1.2 Сучасний стан розвитку багатокористувацьких ігор.

1.3 Соціальне та економічне значення відеоігор у світі. 1.4 Проблематика забезпечення якості у мультиплеєрних іграх. 1.5 Огляд існуючих методів та підходів до ідентифікації.

дефектів. 1.6 Аналіз інструментів і систем для тестування ігор. 1.7 Висновок для першого

розділу. 2. Теоретичні основи методології ідентифікації дефектів. 2.1 Методологічні основи

Ідентифікації дефектів у багатокористувацьких іграх. 2.2 Класифікації дефектів у

багатокористувацьких іграх. 2.3 Методології пошуку дефектів. 2.4 Вдосконалення

методології ідентифікації дефектів. 2.5 Висновок до другого розділу. 3. Практична реалізація

на прикладі Off the Grid. 3.1 Загальна характеристика Off the Grid. 3.2 Планування тестування

та постановка завдань. 3.3 Проведення ідентифікації дефектів. 3.4 Аналіз отриманих

Результатів. 3.5 Доцільність впровадження вдосконалення методології. 3.6 Висновок до

третього розділу. 4. Охорона праці та безпека в надзвичайних ситуаціях. 4.1 Ергономічний

аналіз робочого місця користувача ПК. 4.2 Вплив характеру праці на функціонування

організму. 4.3 Забезпечення електробезпеки користувача ПК. 4.4 Висновок до четвертого

розділу. Висновок. Перелік джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Тема. 2. Актуальність. 3. Мета та завдання дослідження. 4. Практичне значення

результатів. 5.Ігрова індустрія. 6. Інструменти ідентифікації дефектів. 7. Основи класифікації

дефектів. 8. Пошук дефектів і вдосконалена методологія. 9. Off the Grid. 10. Планування

тестування. 11. Ідентифікація дефектів. 12. Аналіз результатів методології. 13. Висновки.

14. Завершальний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В. С., доцент кафедри МТ		
Безпека в надзвичайних ситуаціях	Теслюк В.М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання 17 листопада 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Ознайомлення з завданням до кваліфікаційної роботи	17.11.2025	Виконано
2	Огляд інструментів тестування та управління	17.11.2025-18.11.2025	Виконано
3	Опрацювання джерел по темі кваліфікаційної роботи	19.11.2025-21.11.2025	Виконано
4	Виконання дослідження щодо специфіки тестування багатокористувацьких ігор	21.11.2025-30.11.2025	Виконано
5	Розробка методології ідентифікації дефектів у грі Off the Grid	30.11.2025-01.12.2025	Виконано
6	Оформлення розділу “Аналітичний огляд розвитку відеоігор та підходів до ідентифікації”	01.12.2025-13.12.2025	Виконано
7	Оформлення розділу “Теоретичні основи методології ідентифікації дефектів”	01.12.2025-13.12.2025	Виконано
8	Оформлення розділу “Практична реалізації на прикладі Off the Grid”	01.12.2025-13.12.2025	Виконано
9	Виконання завдання до підрозділу “Охорона праці”	14.12.2025-16.12.2025	Виконано
10	Виконання завдання до підрозділу “Безпека в надзвичайних ситуаціях”	14.12.2025-16.12.2025	Виконано
11	Оформлення кваліфікаційної роботи	17.12.2025-18.12.2025	Виконано
12	Нормоконтроль	18.12.2025-19.12.2025	Виконано
13	Перевірка на плагіат	20.12.2025	Виконано
14	Попередній захист кваліфікаційної роботи	21.12.2025	Виконано
15	Захист кваліфікаційної роботи	22.12.2025	

Студент

(підпис)

Мамрош В. С.

(прізвище та ініціали)

Керівник роботи

(підпис)

Небесний Р.М.

(прізвище та ініціали)

АНОТАЦІЯ

Вдосконалення методології ідентифікації дефектів у багатокористувацьких іграх на прикладі Off the Grid // Кваліфікаційна робота освітнього рівня “Магістр” // Мамрош Віталій Степанович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних наук, група СНм-61 // Тернопіль, 2025 // С. – 88, рис. – 19, табл. – 2, слайди – 14, додатки – 1, бібліогр. – 50.

Ключові слова: багатокористувацькі ігри, тестування ігор, ідентифікація дефектів, Off the Grid, методологія тестування, QA, баг-репорт, Jira, TestRail.

У сучасних умовах розвитку ігрової індустрії багатокористувацькі відеоігри посідають важливе місце як у сфері розваг, так і в сегменті цифрової економіки. Зростання складності ігрових систем, поєднання мережевих, геймплейних та візуальних компонентів зумовлює підвищені вимоги до якості таких продуктів та актуалізує необхідність удосконалення підходів до їх тестування.

У вступі подано загальну характеристику роботи, обґрунтовано актуальність теми, сформульовано мету, завдання, об’єкт і предмет дослідження, а також висвітлено наукову новизну й практичне значення отриманих результатів.

У першому розділі виконано аналітичний огляд розвитку відеоігор, розглянуто витoki та етапи еволюції ігрової індустрії, сучасний стан розвитку багатокористувацьких проєктів, а також соціальне та економічне значення відеоігор у світі. Проаналізовано проблематику забезпечення якості у мультиплеєрних іграх, здійснено огляд існуючих методів та підходів до ідентифікації дефектів, розглянуто основні інструменти й системи тестування (зокрема Jira та TestRail) та узагальнено методологічні основи побудови процесу QA для багатокористувацьких ігрових систем.

У другому розділі викладено теоретичні основи методології ідентифікації дефектів у багатокористувацьких іграх. Наведено класифікацію дефектів, розглянуто базові та спеціалізовані методи пошуку помилок. Запропоновано вдосконалену авторську методику, що поєднує багаторівневу структуру тестування з підходом, орієнтованим на поведінку реального гравця, який свідомо намагається знайти експлойти, обійти геймплейні обмеження або отримати необґрунтовану перевагу. Окреслено принципи пріоритизації дефектів та особливості життєвого циклу багу в контексті мультиплеєрної гри.

У третьому розділі наведено практичну реалізацію запропонованої методології на прикладі Off the Grid. Подано загальну характеристику ігрового світу, основних механік, економіки, завдань Teardrop Ops та архітектури мультиплеєрної взаємодії. Описано процес планування тестування, визначення ключових зон перевірки та побудови тестових сценаріїв. За допомогою розробленої методики проведено ідентифікацію дефектів, у ході якої виявлено низку критичних колізійних помилок та візуальних дефектів накладання текстур. Дефекти були формалізовані у вигляді структурованих баг-репортів, а результати проаналізовано з позиції кількісних і якісних показників. Показано, що застосування вдосконаленої методології дозволяє підвищити ефективність виявлення критичних дефектів та скоротити час їх локалізації.

У четвертому розділі розглянуто аспекти охорони праці та безпеки у надзвичайних ситуаціях, проаналізовано ергономічні вимоги до організації робочого місця користувача ПК, вплив характеру праці на стан організму, а також вимоги щодо забезпечення електробезпеки. Наведено рекомендації щодо створення безпечних і комфортних умов праці відповідно до чинних нормативів.

У висновках підсумовано результати виконаної кваліфікаційної роботи, зроблено узагальнення щодо ефективності запропонованої методології ідентифікації дефектів у багатокористувацьких іграх, окреслено її практичну цінність для команд розробки та тестування, а також можливі напрями подальших досліджень у цій сфері.

ANNOTATION

Improving the methodology for identifying defects in multiplayer games using the example of Off the Grid // Qualification work of the educational level “Master” // Mamrosh Vitalii Stepanovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNm-61 // Ternopil, 2025 // P. – 88, fig. – 19, tabl. – 2, slides – 14, annexes – 1, references – 50.

Keywords: multiplayer games, game testing, defect identification, Off the Grid, testing methodology, QA, bug report, Jira, TestRail.

At the present stage of growth in the game industry, multiplayer projects have become a key component not only of the entertainment market but also of the digital economy. Growing complexity of game systems and the combination of networking, gameplay and visual components impose increased requirements on the quality of such products and highlight the need to improve approaches to their testing.

The introductory section outlines the structure of the thesis, justifies the significance of the chosen topic, defines the purpose, research tasks, object and subject, and highlights both the scientific contribution and the practical relevance of the obtained results.

The first chapter presents an analytical overview of the development of video games, considers the origins and stages of the evolution of the game industry, the current state of multiplayer projects, together with an examination of the global social and economic impact that video games exert today. The issues of quality assurance in multiplayer games are analysed; existing methods and approaches to defect identification are reviewed; the main tools and systems used in testing (in particular Jira and TestRail) are considered; and the methodological foundations of building a QA process for multiplayer game systems are summarised.

The second chapter sets out the theoretical foundations of a methodology for defect identification in multiplayer games. A classification of defects is provided, and

basic as well as specialised methods of defect detection are examined. An improved author's methodology is proposed, combining a multi-level testing structure with an approach focused on the behaviour of a real player who deliberately tries to find exploits, bypass gameplay limitations or gain an unjustified advantage. Principles of defect prioritisation and the specific features of the bug life cycle in the context of a multiplayer game are outlined.

The third chapter presents the practical implementation of the proposed methodology using *Off the Grid* as a case study. A general description of the game world, core mechanics, in-game economy, Teardrop Ops missions and the architecture of multiplayer interaction is provided. The processes of test planning, identifying key verification areas and constructing test scenarios are described. Using the developed methodology, defect identification was carried out, during which a number of critical collision-related errors and visual texture overlay defects were discovered. These defects were formalised in the form of structured bug reports, and the results were analysed from both quantitative and qualitative perspectives. It is demonstrated that the application of the improved methodology increases the efficiency of detecting critical defects and reduces the time required for their localisation.

The fourth chapter examines the issues of occupational safety and emergency preparedness, analyzes ergonomic requirements for organizing the workplace of a PC user, the impact of work characteristics on the human body, as well as the requirements for ensuring electrical safety. Recommendations are provided for creating safe and comfortable working conditions in accordance with current regulations.

The concluding section integrates the outcomes of the completed qualification work, presents overarching conclusions regarding the efficiency of the proposed methodology for identifying defects in multiplayer games, emphasises its practical usefulness for development and QA teams, and identifies potential avenues for subsequent research.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПК – персональний комп'ютер.

API (англ. Application Programming Interface) – програмний інтерфейс застосунків.

CPU (англ. Central Processing Unit) – центральний процесор.

FPS (англ. Frames Per Second) – кількість кадрів за секунду.

GPU (англ. Graphics Processing Unit) – графічний процесор.

HEX – внутрішньоігрова одиниця або ресурс у грі Off the Grid, що використовується для покращення спорядження чи кіберкінцівок.

NPC (англ. Non-Player Character) – неігровий персонаж.

QA (англ. Quality Assurance) – забезпечення якості.

UI (англ. User Interface) – інтерфейс користувача.

UX (англ. User Experience) – користувацький досвід.

ЗМІСТ

ВСТУП.....	10
1 АНАЛІТИЧНИЙ ОГЛЯД РОЗВИТКУ ВІДЕОІГОР ТА ПІДХОДІВ ДО ІДЕНТИФІКАЦІЇ	13
1.1 Витоки відеоігор та етапи їх розвитку	13
1.2 Сучасний стан розвитку багатокористувацьких ігор	15
1.3 Соціальне та економічне значення відеоігор у світі.....	16
1.4 Проблематика забезпечення якості у мультиплеєрних іграх	17
1.5 Огляд існуючих методів та підходів до ідентифікації дефектів.....	18
1.6 Аналіз інструментів і систем для тестування ігор	20
1.7 Методологічні основи ідентифікації дефектів у багатокористувацьких іграх	24
1.8 Висновок для першого розділу	27
2 ТЕОРЕТИЧНІ ОСНОВИ МЕТОДОЛОГІЇ ІДЕНТИФІКАЦІЇ ДЕФЕКТІВ.....	29
2.1 Методологічні основи ідентифікації дефектів у багатокористувацьких іграх	29
2.2 Класифікація дефектів у багатокористувацьких іграх	31
2.3 Методології пошуку дефектів.....	36
2.4 Вдосконалена методологія ідентифікація дефектів.....	38
2.5 Висновок до другого розділу	40
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ НА ПРИКЛАДІ OFF THE GRID	42
3.1 Загальна характеристика Off the Grid	42
3.2 Планування тестування та постановка завдань.....	48
3.3 Проведення ідентифікації дефектів.....	49
3.4 Аналіз отриманих результатів	56
3.5 Доцільність впровадження вдосконаленої методології	58
3.6 Висновок до третього розділу.....	59
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТАЦІЯХ.....	60
4.1 Ергономічний аналіз робочого місця користувача ПК	60
4.2 Вплив характеру праці на функціонування організму	63
4.3 Забезпечення електробезпеки користувача ПК	65
4.4 Висновок до четвертого розділу	68
ВИСНОВКИ.....	69
ПЕРЕЛІК ДЖЕРЕЛ	72
ДОДАТКИ	

ВСТУП

Актуальність теми. Сучасна індустрія відеоігор є однією з найдинамічніших сфер інформаційних технологій, що поєднує елементи програмної інженерії, дизайну, штучного інтелекту та мережевих технологій. Розвиток багатокористувацьких ігор, які функціонують у режимі реального часу, створює нові виклики для системи забезпечення якості програмного забезпечення.

Зростання кількості інтерактивних компонентів, різноманіття пристроїв і складна архітектура серверної взаємодії вимагають розробки ефективних методів тестування, здатних виявляти дефекти не лише на рівні коду, а й у поведінкових, мережевих і візуальних аспектах.

Особливої актуальності набуває питання ідентифікації дефектів у мультиплеєрних проєктах, де навіть незначна помилка може призвести до збоїв у геймплеї, втрати синхронізації між клієнтами, дисбалансу гри чи появи експлойтів. Саме тому дослідження процесів виявлення та документування помилок, а також вдосконалення відповідної методології має важливе теоретичне та практичне значення для галузі Game Quality Assurance.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи є вдосконалення методології ідентифікації дефектів у багатокористувацьких іграх на прикладі гри Off the Grid, спрямованої на підвищення ефективності процесу тестування та покращення якості кінцевого продукту.

Для досягнення поставленої мети були визначені наступні завдання:

- Провести аналітичний огляд розвитку відеоігор і підходів до виявлення дефектів у них.
- Дослідити існуючі методології тестування та інструменти для ідентифікації помилок у мультиплеєрних середовищах.
- Розробити вдосконалену методику, орієнтовану на виявлення складних і нетипових дефектів, притаманних багатокористувацьким проєктам.
- Реалізувати практичну апробацію методології на прикладі гри Off the Grid та оцінити її ефективність за кількісними та якісними показниками.

Об’єкт дослідження: процес ідентифікації дефектів у багатокористувацьких відеоіграх у межах життєвого циклу розробки ігрових проєктів.

Предмет дослідження: методи, засоби та принципи пошуку, документування, пріоритизації та аналізу дефектів у мультиплеєрних іграх (клієнт-серверна логіка, мережеві умови, геймплейні механіки, UI/UX).

Наукова новизна одержаних результатів. У роботі запропоновано гібридну методику ідентифікації дефектів у багатокористувацьких іграх, що поєднує класичні функціональні перевірки з поведінковим (ігровим) підходом тестувальника, орієнтованим на аналіз механік, взаємодій і спроб знаходження шляхів для здобуття переваги над іншими гравцями. Такий підхід дозволяє розширити межі звичайного тестування, оскільки тестувальник не просто перевіряє сценарії, а мислить як гравець, який намагається “обхитрити” систему, тим самим виявляючи приховані логічні й балансні дефекти. Методика поєднує функціональний аналіз, дослідження поведінки об’єктів і тестування колізій у різних умовах середовища з подальшим документуванням результатів за структурованим життєвим циклом дефекту. Уперше апробацію проведено на прикладі гри Off the Grid, де за допомогою цього підходу було знайдено й описано низку критичних помилок, що стосуються фізики, текстур і відображення моделей у середовищі.

Практичне значення одержаних результатів. Отримані результати мають прикладний характер і можуть бути використані для підвищення ефективності тестування багатокористувацьких ігор. У роботі апробовано вдосконалену методологію ідентифікації дефектів, яка поєднує класичний функціональний підхід із поведінковим методом тестування, орієнтованим на дії реального гравця. Запропонована методика дозволяє виявляти критичні помилки, пов’язані з колізіями, графічними артефактами та відображенням текстур, що напряму впливають на якість геймплею. Розроблені рекомендації можуть бути використані QA-фахівцями під час перевірки багатокористувацьких проєктів, зокрема у фазах внутрішнього тестування перед релізом.

Апробація результатів магістерської роботи. Основні результати дослідження були апробовані під час практичного тестування багатокористувацької гри Off the Grid, у межах якого автором було складено низку баг-репортів, що відображають реальні дефекти (Bug ID: 0117, 0160, 0177), виявлені за допомогою запропонованої методології. Практична перевірка підтвердила її ефективність для виявлення критичних колізійних та графічних помилок у реальних умовах геймплею. Крім того, результати роботи пройшли апробацію у форматі наукової доповіді на 2-й Міжнародній науково-практичній конференції “Progressive Approaches in Science and Engineering”.

Публікації. Основні положення та висновки кваліфікаційної роботи були опубліковані у вигляді двох тез конференції (Див. додатки А).

Структура й обсяг кваліфікаційної роботи. Робота магістра складається зі вступу, чотирьох розділів, висновків та списку літератури із 50 джерел. Обсяг роботи складає 88 сторінок, 19 рисунків, 2 таблиці та 1 додаток.

1 АНАЛІТИЧНИЙ ОГЛЯД РОЗВИТКУ ВІДЕОІГОР ТА ПІДХОДІВ ДО ІДЕНТИФІКАЦІЇ

1.1 Витоки відеоігор та етапи їх розвитку

Відеоігри – це складова сучасного розвитку як культури, так і цифрового світу. Ними захоплюються десятки мільйонів людей по всьому світу не лише для гри але й для пізнання внутрішнього світу автора.

Саме в поєднанні розвитку електронної техніки, інформатики та інтересу до інтерактивних розваг виникли відеоігри. Історія відеоігор охоплює кілька чітко виражених етапів, кожен з яких є сукупністю технологічних інновацій, культурних практик та економічних моделей. Дані етапи формували не лише способи розробки ігор, але й підходи до їх тестування й забезпечення якості.

Перші експерименти (1950–1960-ті).

Ранні електронні ігрові експерименти з'явилися в наукових лабораторіях та університетах: були реалізовані просто симуляції ігор на великих ЕОМ (текстові ігри, PDP-1). Це було перше тестування саме відтворення зображення та користувацького інтерфейсу, яке заклало основу для подальшого розвитку всієї індустрії [1].

Аркадна революція та ера консолей (1970–1983).

Перший комерційний прорив пов'язують з виходом Pong (1972) та масовим поширенням автоматів, перші домашні консолі (Atari 2600 та ін.) зробили ігри доступними для широкої аудиторії. З появою перших франшиз (Pac-Man, Donkey Kong) ігрова індустрія швидко почала комерціалізуватися. Даний етап заклав стандартні жанри й бізнес-моделі для наступних десятиліть.

Крах ринку та відродження (1983–1995).

Через надшвидке розростання ринку на початку 80-их, перенасичення низькоякісними продуктами стало лише питанням часу. Саме через появу надмірної кількості ігор стало причиною фінансової кризи у 1983 році. З виходом якісних консолей і суворою політикою Nintendo для відбору контенту почався етап відродження. Також неабияких поштовхом стало масове поширення

домашніх ПК – Commodore 64, IBM PC, що змінило баланс сил між платформами.

Ера 3D і консольних війн (1994–2005).

Після переходу ігор у площину 3D-графіки, завдяки приходу CD-ROM, більш потужних CPU/GPU і нових консолей (Sony PlayStation, Sega Saturn, PlayStation 2, Xbox), кінематографічні сюжети і механіки ігор стали значно кращої якості, що у свою чергу збільшило вимоги до тестування – сумісність, продуктивність та контроль якості контенту.

Сенсація онлайн-ігрових сервісів і MMORPG (2000–2010).

З розвитком широкосмугового інтернету з'явилися великі онлайн-проекти і сервіси, а також нові платформи для мультиплеєра – XBOX Live. Мультиплеєрні архітектури стали новим поштовхом у підході до тестування, бо до класичних функціональних перевірок додалися ще й мережеві, навантажувальні та сценарії синхронізації клієнт-сервер.

Мобільний, free-to-play і нові бізнес-моделі (2010–2020).

З появою App Store та Google Play ринок ігор суттєво змінився: шлях до користувача став набагато простішим, з'явилися моделі монетизації (microtransactions, free-to-play), що вплинуло на дизайн ігрових систем та на критерії якості (зручність монетизації, безпека транзакцій, аналітика поведінки користувачів).

Новітній етап: мультиплатформеність, хмарні сервіси, соціальні та змагальні практики (2020-наш час).

Теперішня ігрова індустрія орієнтується на крос-платформні рішення, стрімінг та хмарні сервіси, кіберспорт і великі спільноти гравців. Також зросли очікування щодо стабільності мультиплеєру, захисту даних і швидких оновлень. Це все ускладнює процес QA і підкреслює необхідність спеціалізованих підходів до виявлення дефектів у багакористувацьких проектах [2].

У результаті, відеоігри пройшли шлях від простих еспериментів до масштабних інтерактивних світів із мільйонами користувачів. Із зростанням технічної складності, онлайн-механік та постійної взаємодії гравців між собою, це все стало окремим соціальним та культурним феноменом. Саме тому

подальший аналіз сучасного стану розвитку багатокористувацьких ігор є ключовим для розуміння усіх викликів, що постають як перед розробниками, так і фахівцями з тестування.

1.2 Сучасний стан розвитку багатокористувацьких ігор

У 2025 році індустрія багатокористувацьких ігор переживає етап глибоких трансформацій, зумовлених технологічними інноваціями, змінами в поведінці користувачів та новими бізнес-моделями. Однією з ключових тенденцій є кросплатформеність – можливість грати на різних пристроях (ПК, консолі, мобільні телефони) під одним акаунтом, що сприяє інтеграції гравців і розширенню ігрових спільнот.

Технологічні інновації.

Ключовими тенденціями для персоналізації ігрового досвіду є інтеграція штучного інтелекту. Це дозволяє адаптувати складність гри до рівня навичок гравця, створювати динамічні сюжетні лінії та покращувати взаємодію з неігровими персонажами – NPC. Загалом віртуальна та доповнена реальність – VR та AR відкривають нові горизонти для занурення в ігрові світи, дозволяючи гравцям взаємодіяти з середовищем у більш природний спосіб.

Соціальні та спільнотні аспекти.

Теперішні ігри активно інтегрують соціальні аспекти, такі як голосовий чат, спільні місії та змагання, що формує геймерські спільноти. За допомогою цих інновацій гравці не лише грають, а й спілкуються, обмінюються досвідом, що значно збагачує ігровий досвід [3].

Економічні моделі.

У сфері монетизації відбувається саме перехід від традиційних моделей до більш гнучких та інтегрованих в ігровий процес. Більшість моделей free-to-play з внутрішніми покупками, battle pass та seasonal content стали буденністю для багатьох ігор. Крім того, зростаюча популярність Play-to-Earn, де гравці безпосередньо можуть заробляти реальні гроші – криптовалюта чи NFT, відкриває нові горизонти та можливості для геймерів [4].

Виклики та перспективи.

Ігрова індустрія стикається з рядом викликів саме через численні досягнення та нововведення. Саме забезпечення мережевої стабільності та захисту даних в умовах високої конкуренції та глобальної взаємодії гравців є одним із основних викликів сьогодення. Крім того, зростаюча складність ігор вимагає від розробників не лише нових підходів до створення, а й до тестування, зокрема в умовах багатокористувацьких середовищ [5].

1.3 Соціальне та економічне значення відеоігор у світі

Відеоігри стали невід'ємною частиною сучасної культури та економіки, впливаючи на різні аспекти життя людей по всьому світу. Їхній вплив охоплює соціальну інтеграцію, розвиток навичок, а також значний економічний внесок у глобальну економіку.

Соціальний вплив відеоігор.

Онлайн-відеоігри створюють платформи для соціальних взаємодій, де гравці можуть спілкуватися, співпрацювати та змагатися з іншими людьми з різних куточків світу. Це сприяє розвитку комунікативних навичок, командної роботи та здатності до вирішення конфліктів у віртуальному середовищі. Наприклад, у багатьох багатокористувацьких іграх гравці повинні координувати свої дії, обговорювати стратегії та приймати спільні рішення, що сприяє розвитку соціальних навичок.

Крім того, відеоігри можуть бути інструментом для розвитку емоційного інтелекту. Гравці навчаються розпізнавати та керувати своїми емоціями, а також розуміти емоції інших учасників гри. Це особливо важливо для молоді, оскільки дозволяє їм розвивати емпатію та соціальну чутливість у безпечному та контрольованому середовищі.

Економічний вплив відеоігор.

Ігрова індустрія є однією з найбільш динамічно зростаючих галузей у світі. За даними досліджень, глобальний ринок відеоігор у 2024 році оцінювався в понад 298 мільярдів доларів США, з прогнозованим зростанням до 600 мільярдів

доларів до 2030 року. Це свідчить про високий попит на відеоігри та їхній значний економічний вплив.

Відеоігри створюють мільйони робочих місць у різних сферах, таких як розробка програмного забезпечення, дизайн, маркетинг, тестування та підтримка. Крім того, індустрія відеоігор стимулює розвиток суміжних галузей, таких як кіберспорт, стримінг, виробництво ігрового обладнання та програмне забезпечення. Наприклад, зростання популярності стримінгових платформ, таких як Twitch та YouTube Gaming, створило нові можливості для контент-креаторів та професійних гравців.

Інвестиції в індустрію відеоігор також сприяють економічному розвитку окремих країн та регіонів. Наприклад, уряди деяких країн активно підтримують розвиток національних індустрій відеоігор через надання субсидій, податкових пільг та створення спеціалізованих індустріальних парків. Це сприяє створенню нових робочих місць, розвитку інфраструктури та залученню інвестицій.

Відеоігри мають значний соціальний та економічний вплив на сучасне суспільство. Вони сприяють розвитку соціальних навичок, емоційного інтелекту та професійних компетенцій, а також є важливим економічним сектором, що генерує значні доходи та створює нові можливості для розвитку. Розуміння цих аспектів є важливим для оцінки ролі відеоігор у сучасному світі та їхнього потенціалу для подальшого розвитку [6][7].

1.4 Проблематика забезпечення якості у мультиплеєрних іграх

Забезпечення якості у мультиплеєрних іграх є складним та багатогранним процесом, що вимагає врахування численних технічних та соціальних аспектів. У порівнянні з однокористувацькими іграми, де тестування зосереджене на лінійних сценаріях, мультиплеєрні проекти вимагають перевірки взаємодії між численними гравцями, що значно ускладнює QA-процес.

Технічні виклики.

Однією з основних проблем є забезпечення стабільності серверів та мережевої інфраструктури. Лаги, десинхронізація, збої в матчмейкінгу та

нестабільні сервери можуть серйозно вплинути на досвід користувачів та знизити рівень утримання гравців.

Для ефективного тестування QA-команди використовують інструменти для симуляції реальних умов гри, включаючи інструменти для моделювання затримок, обмеження пропускну здатності та стрес-тестування.

Організаційні та методологічні аспекти.

QA-команди повинні координувати свої дії між різними локаціями та використовувати спільні журнали та сценарії тестування для відтворення реальних умов гри. Це включає в себе не лише технічні аспекти, але й організаційні, такі як ефективна комунікація між командами та управління тестовими процесами.

Використання автоматизації та новітніх технологій.

З розвитком технологій автоматизації тестування, зокрема за допомогою штучного інтелекту, з'являються нові можливості для покращення процесу QA. Наприклад, платформи, що використовують AI для автоматичного виявлення помилок та генерації звітів, можуть значно зменшити час та витрати на тестування, підвищуючи ефективність процесу.

Забезпечення якості у мультиплеєрних іграх є критично важливим для успіху проекту. Врахування технічних, організаційних та технологічних аспектів QA дозволяє створювати стабільні та привабливі для користувачів ігри, що відповідають високим стандартам якості [8][9].

1.5 Огляд існуючих методів та підходів до ідентифікації дефектів

Ідентифікація дефектів у мультиплеєрних іграх є складним процесом, що вимагає застосування різноманітних методів та підходів для забезпечення високої якості продукту. Сучасні методи тестування охоплюють як традиційні техніки, так і новітні технології, зокрема автоматизацію та використання штучного інтелекту.

Традиційні методи тестування.

Серед традиційних методів тестування виділяються:

- Функціональне тестування: перевірка коректності роботи всіх функцій гри, включаючи механіку, інтерфейс та взаємодію гравців.
- Сумісність: забезпечення стабільної роботи гри на різних пристроях, операційних системах та мережевих умовах.
- Продуктивність: оцінка швидкодії гри, включаючи час завантаження, частоту кадрів та реакцію на дії гравців.
- Регресійне тестування: перевірка, чи не вплинули зміни в коді на існуючі функції гри.

Ці методи є основою для виявлення багів та інших дефектів, що можуть вплинути на досвід користувачів.

Автоматизація тестування.

З розвитком технологій автоматизації тестування з'явилися нові можливості для ефективного виявлення дефектів. Наприклад, використання автоматизованих скриптів дозволяє швидко перевірити великі обсяги даних та сценаріїв, що важко здійснити вручну. Це особливо корисно для тестування багатокористувацьких режимів, де взаємодія між гравцями може бути складною та непередбачуваною.

Використання штучного інтелекту.

Штучний інтелект також знаходить застосування в тестуванні ігор. Наприклад, системи на основі глибокого навчання можуть автоматично виявляти аномалії в поведінці гравців або виявляти потенційні баги, аналізуючи великі обсяги даних з ігрових сесій. Це дозволяє значно зменшити час на тестування та підвищити точність виявлення дефектів.

Використання даних про поведінку гравців.

Аналіз даних про поведінку гравців є важливим для ідентифікації дефектів. Наприклад, вивчення патернів гри, таких як частота використання певних стратегій або взаємодія з іншими гравцями, може допомогти виявити неочевидні проблеми, які не проявляються під час стандартного тестування.

Виклики та обмеження.

Незважаючи на розвиток методів тестування, існують певні виклики та обмеження. Наприклад, складність багатокористувацьких режимів може

призвести до того, що деякі дефекти проявляються лише за певних умов або при взаємодії з конкретними гравцями. Крім того, швидкий розвиток технологій та змінювані умови мережі вимагають постійного оновлення методів тестування для забезпечення їх ефективності.

Ідентифікація дефектів у мультиплеєрних іграх є важливим етапом у розробці якісного продукту. Застосування різноманітних методів та підходів, включаючи традиційні техніки, автоматизацію та використання штучного інтелекту, дозволяє ефективно виявляти та усувати дефекти, забезпечуючи високий рівень якості гри [10][11].

1.6 Аналіз інструментів і систем для тестування ігор

Для забезпечення високої якості мультиплеєрних ігор розробники та QA-команди широко використовують системи управління тестуванням та трекінгу дефектів. Серед них найбільш популярними є Jira та TestRail, які забезпечують структуроване ведення баг-репортів, тест-кейсів та процесів тестування.

Jira.

Jira є універсальною платформою для відстеження дефектів і управління проектами. Основним елементом у QA-процесі є система баг-трекінгу, що дозволяє створювати, редагувати та контролювати стан помилок. Користувачі можуть працювати з головним меню, де представлено всі проекти, відкриті задачі та фільтри для швидкого доступу до баг-репортів. На рисунку 1.1 показано головне меню Jira з доступом до проектів та баг-репортів.

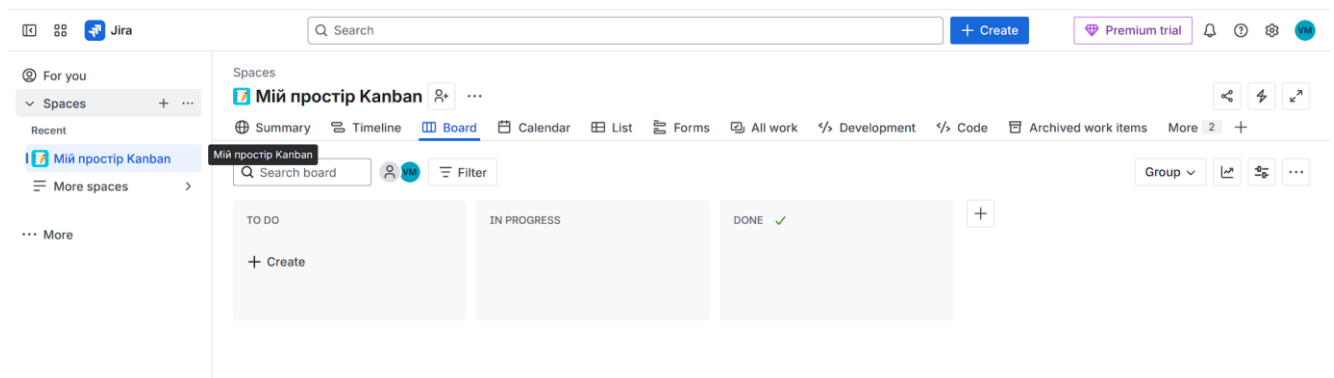


Рисунок 1.1 – Головне меню Jira

Для кожного багу створюється форма для заповнення баг-репорту, яка може включати такі поля: назва дефекту, його опис, пріоритет, статус, кроки для відтворення, очікуваний результат, додані скріншоти та вкладені файли. Ці форми є гнучкими: тестувальник може самостійно додавати додаткові поля залежно від специфіки проекту або вимог команди. На рисунку 1.2 показано процес створення форми для баг-репорту у Jira з можливістю додавання додаткових полів.

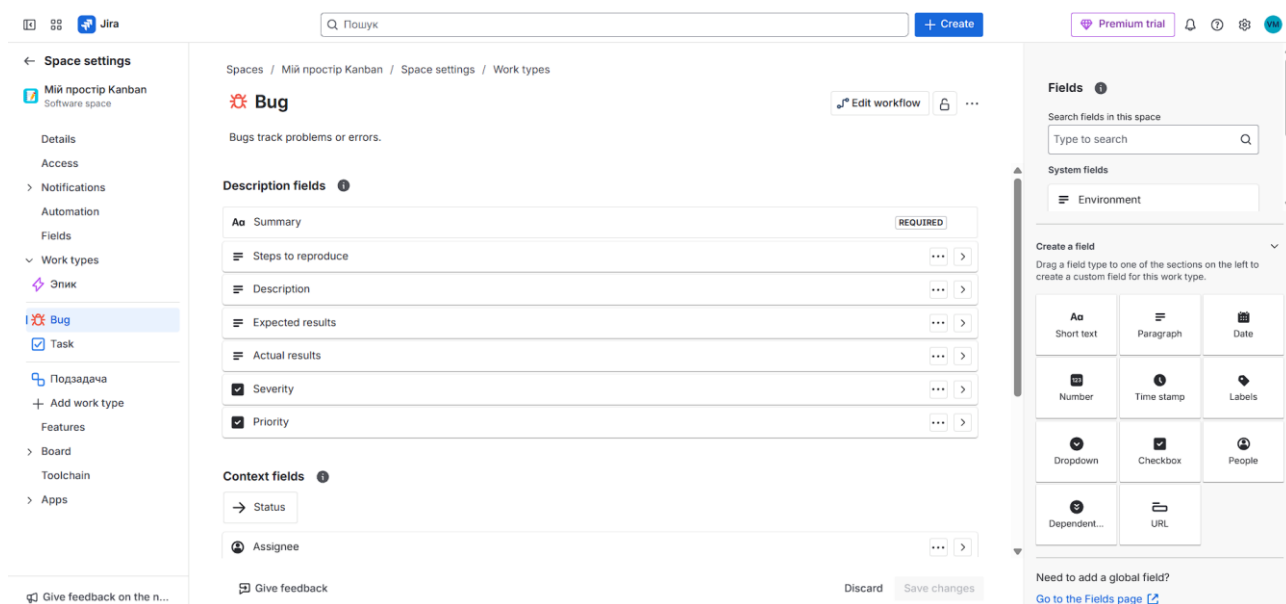


Рисунок 1.2 – Створення форми для баг-репорту у Jira

Після заповнення форми баг-репорт зберігається в системі та відображається у вигляді списку з усіма деталями, що дозволяє команді швидко відслідковувати прогрес усунення дефектів, призначати відповідальних та контролювати терміни.

Крім того, кожен звіт може бути доповнений додатковими коментарями тестувальників або розробників, що створює ефективний канал комунікації між членами команди. Це особливо важливо у випадку складних дефектів, які відтворюються не завжди або залежать від специфічних умов середовища (наприклад, мережевої затримки чи синхронізації серверів).

На рисунку 1.3 показано заповнену форму баг-репорту та меню з наявними баг-репортами для відстеження прогресу.

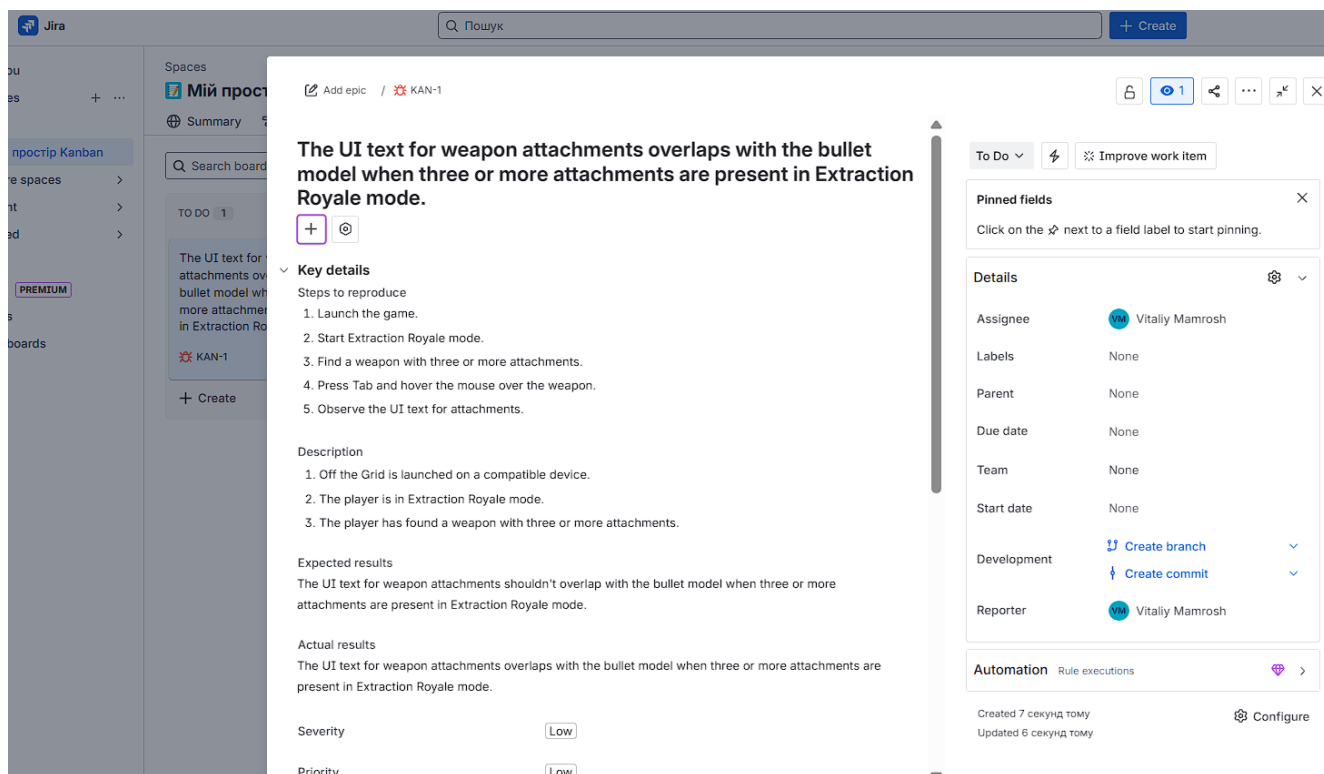


Рисунок 1.3 – Форма для заповнення баг-репорту та меню з баг-репортами

Такий підхід забезпечує повну прозорість процесу тестування та дозволяє координувати роботу багатьох тестувальників одночасно.

TestRail.

Система TestRail підтримує журнал активностей, у якому фіксуються зміни статусу тестів, час виконання, результати перевірок і коментарі членів команди. Це значно спрощує аналіз ефективності тестування, адже кожен тест-кейс може бути пов'язаний із конкретним спринтом або версією гри. Завдяки цьому TestRail виступає не лише інструментом для зберігання сценаріїв, а й аналітичною платформою, яка дозволяє оцінювати прогрес QA-команди та виявляти слабкі місця у процесі перевірки.

Інтеграція з іншими системами, такими як JIRA, забезпечує єдиний інформаційний простір між тестувальниками та розробниками, що підвищує швидкість реагування на виявлені дефекти.

У TestRail кожен тест-кейс має два ключових представлення. На рисунку 1.4 показано спрощене відображення назви тест-кейсу у TestRail для швидкого навігаційного доступу.

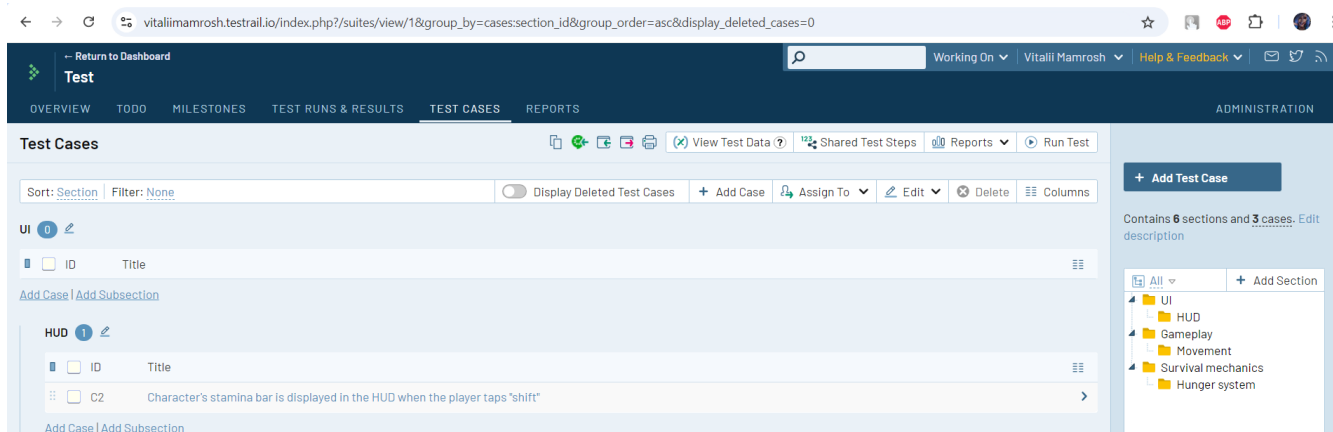


Рисунок 1.4 – Назва тест-кейсу у TestRail

З рис. 1.4 видно, що тестувальник може швидко оцінити, про що йдеться у тест-кейсі, не відкриваючи його повністю. Спрощене відображення назви використовується для зручної навігації та швидкого відбору сценаріїв для виконання.

На рисунку 1.5 показано повний опис тест-кейсу у TestRail з усіма полями: назва, детальний опис, кроки для виконання, очікуваний результат та додаткові примітки.

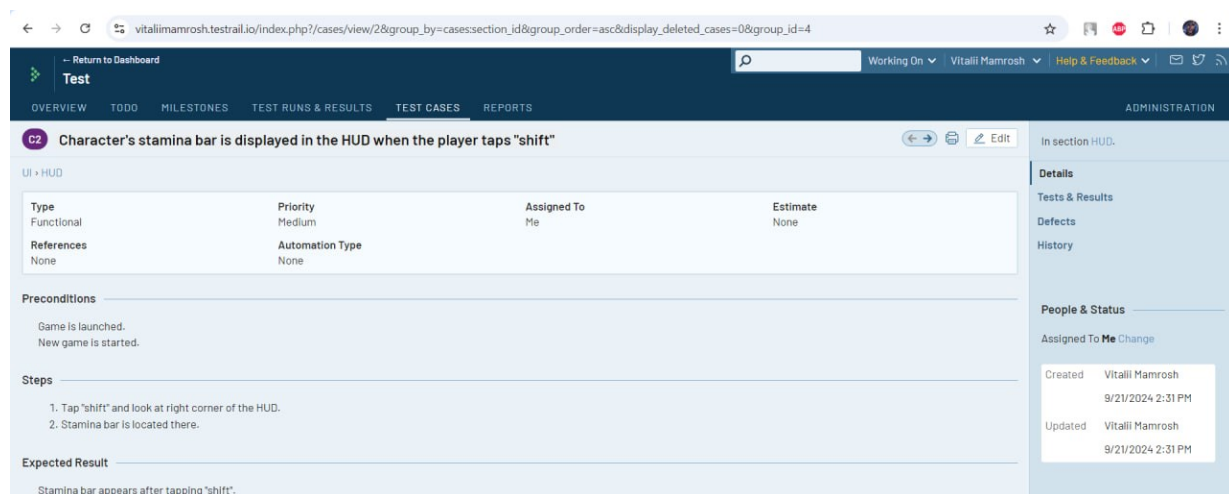


Рисунок 1.5 – Повний опис тест-кейсу у TestRail

З рис. 1.5 видно, що для детального аналізу та повторення даного тест-кейсу було вказано всі необхідні поля: назва, детальний опис, кроки для виконання, очікуваний результат, пріоритет, додаткові примітки та прикріплені файли.

Система дозволяє тестувальникам ефективно планувати тестування, групувати кейси за модулями гри, відстежувати прогрес виконання та генерувати звіти для менеджерів і розробників. Комбінація Jira та TestRail забезпечує повний цикл QA: від створення баг-репортів та їхнього відстеження до виконання детальних тест-кейсів.

Використання Jira та TestRail дозволяє структуровано організувати процес тестування мультиплеєрних ігор, підвищити ефективність QA-команди та забезпечити прозорість у роботі з багами та тест-кейсами. Візуалізація головного меню, форм баг-репортів і повних описів тест-кейсів допомагає швидко орієнтуватися у процесі, розподіляти завдання та контролювати якість продукту на всіх етапах розробки [12][13].

1.7 Методологічні основи ідентифікації дефектів у багатокористувацьких іграх

У сучасному науковому середовищі, що розвивається під впливом цифровізації, дедалі більшого значення набувають дослідження, спрямовані на вдосконалення процесів аналізу, моделювання та обробки інформаційних потоків. Одним із вагомих напрямів є ідентифікація структурних закономірностей у цифрових сигналах, що традиційно використовувалась у телекомунікаційних та медичних системах, проте сьогодні має практичне значення і для геймдеву. У грі кожна взаємодія користувача з системою – натискання клавіш, рухи миші, обчислення позицій персонажів у просторі – формує складні цифрові патерни. На основі методів аналізу циклічних сигналів можна створювати системи, здатні розпізнавати повторювані дефекти, наприклад лаги, мікрозатримки чи циклічні падіння FPS. Такі рішення дозволяють QA-фахівцям фіксувати не лише явні збої, а й виявляти закономірності у поведінці клієнтського коду, що свідчать про потенційні проблеми з обробкою даних на рівні рушія гри чи мережевого обміну. У майбутньому подібні алгоритми можуть стати частиною автоматизованих систем тестування, які без участі людини визначатимуть ділянки коду, схильні

до помилок при великому навантаженні [Помилка! Джерело посилання не знайдено.].

Дослідження у сфері грид-систем і хмарних обчислень забезпечують фундаментальне розуміння принципів побудови розподілених обчислювальних інфраструктур, які є невід’ємною складовою сучасних багатокористувацьких ігор. Ігри формату Battle Royale, MMO чи Extraction Royale, як-от Off the Grid, вимагають масштабованих серверних архітектур, здатних підтримувати тисячі одночасних підключень. Теоретичні засади грид-обчислень допомагають зрозуміти, як реалізувати балансування навантаження між серверами, як забезпечити синхронізацію ігрових сесій та динамічне розподілення ресурсів у хмарі. Для тестувальника це означає можливість створювати сценарії навантажувального тестування, які імітують поведінку великої кількості гравців, визначаючи межі стабільної роботи серверів. Водночас принципи хмарних технологій дозволяють реалізовувати автоматизовані CI/CD-середовища, де тестові збірки ігор перевіряються у віртуальних контейнерах, що підвищує швидкість виявлення дефектів у різних конфігураціях систем [Помилка! Джерело посилання не знайдено.].

Не менш вагомими є праці, присвячені розподіленим системам моніторингу та керування, адже саме вони закладають основи створення інструментів спостереження за стабільністю роботи складних обчислювальних комплексів. У багатокористувацьких іграх такі системи застосовуються для аналізу логів у реальному часі, моніторингу навантаження серверів, контролю роботи мережових API та відстеження критичних помилок. Наприклад, при тестуванні гри Off the Grid можна застосовувати принципи розподіленого моніторингу для збору даних із різних вузлів – клієнтських машин, серверів матчмейкінгу, баз даних та систем аналітики. Такий підхід дозволяє створити єдину карту активності системи, де будь-яка аномалія (перевищення часу відгуку, зависання запиту, збій синхронізації) автоматично фіксується і маркується для подальшого аналізу QA-фахівцем [14].

Значну методологічну цінність становлять також роботи, присвячені цифровій обробці біомедичних сигналів. Попри іншу предметну галузь, вони

використовують спільні алгоритмічні принципи, корисні й для виявлення дефектів у геймдеві. Алгоритми згладжування шумів, виявлення сплесків та ідентифікації відхилень у часових рядах можуть застосовуватися при аналізі показників продуктивності гри: частоти кадрів, часу відгуку на дії користувача, завантаження оперативної пам'яті чи мережевих затримок. Наприклад, якщо під час гри відбувається періодичне падіння FPS у певних локаціях або при одночасному виконанні певних дій, система аналізу сигналів може автоматично виявити повторюваність цього дефекту, що допоможе розробнику швидше знайти його джерело. Таким чином, методи цифрової діагностики з інших галузей можуть бути ефективно інтегровані у QA-процеси ігор **[Помилка! Джерело посилання не знайдено.]**.

Важливу роль у контексті тестування відіграють і роботи, присвячені оптимізації реляційних баз даних. Сучасні ігри активно використовують бази даних для збереження профілів гравців, прогресу, інвентарю, досягнень і внутрішньоігрових транзакцій. Проблеми з продуктивністю бази можуть призводити до затримок при завантаженні сесії, втрати даних або навіть до появи критичних помилок у процесі гри. Використання стратегій оптимізації запитів, кешування даних і налаштування індексів дозволяє не лише підвищити ефективність серверної частини гри, а й мінімізувати ризик появи дефектів, пов'язаних із одночасним доступом великої кількості користувачів. Прикладом є оптимізація бази даних у грі *Destiny 2*, де зменшення часу вибірки даних користувачів суттєво покращило стабільність з'єднання та швидкість завантаження контенту [16].

Окремо варто згадати дослідження, які висвітлюють аспекти побудови стійких розподілених систем і механізмів відновлення після збоїв. У таких роботах детально розглядаються питання синхронізації вузлів, маршрутизації даних і компенсації втрачених пакетів, що має безпосередній зв'язок із роботою мережевих рушіїв у багатокористувацьких іграх. Наприклад, у шутерах від першої особи застосовуються протоколи прогнозування руху (*client-side prediction*), які допомагають мінімізувати вплив затримок. Помилки в цих алгоритмах часто призводять до дефектів, як-от “rubber banding” або “ghosting”

персонажів. Тому знання принципів розподілених обчислень і синхронізації процесів допомагає краще розуміти природу таких проблем і вдосконалювати методи їх тестування [18].

Вагомий внесок у розвиток методологій контролю якості становлять також дослідження у сфері інтелектуального аналізу даних і автоматизації моніторингу. Алгоритми машинного навчання дедалі частіше використовуються у геймдеві для прогнозування відмов серверів, виявлення аномальної поведінки гравців або автоматичної класифікації багів за рівнем критичності. Наприклад, система AI-тестування може аналізувати журнали ігрових серверів, розпізнавати повторювані сигнатури помилок і визначати, у якому модулі коду вони найчастіше виникають. Такі технології вже активно впроваджуються у великих студіях, як-от Ubisoft і EA Games, де машинне навчання допомагає скоротити час пошуку причин збою на десятки відсотків. У контексті розробки вдосконаленої методології ідентифікації дефектів цей напрям відкриває можливості для створення систем самонавчального тестування, які адаптуються до змін у коді гри [Помилка! Джерело посилання не знайдено.].

У сукупності всі розглянуті джерела формують потужне методологічне підґрунтя, що поєднує класичні принципи цифрової обробки сигналів, сучасні технології хмарних обчислень, оптимізацію баз даних та інтелектуальні системи аналізу. Їхня цінність полягає у можливості інтеграції цих підходів у єдину систему автоматизованого тестування, яка здатна виявляти, класифікувати та прогнозувати дефекти у багатокористувацьких іграх на основі реальних даних. Така міждисциплінарна взаємодія дозволяє не лише вдосконалити якість гри, але й підвищити ефективність процесів QA-тестування у цілому.

1.8 Висновок для першого розділу

У першому розділі було проведено аналітичний огляд розвитку відеоігор та сучасних підходів до ідентифікації дефектів у багатокористувацьких проєктах. Було досліджено витоки відеоігор, ключові етапи їх розвитку, починаючи від перших експериментів на великих ЕОМ та аркадних автоматів до

сучасних мультиплатформених і хмарних рішень. Аналіз історії показав, як технологічні інновації, розвиток апаратного забезпечення та поява нових платформ впливали на способи розробки ігор і підходи до контролю їх якості.

Було також розглянуто сучасний стан багатокористувацьких ігор, їх поширення та популярність серед глобальної аудиторії, а також соціальні та економічні аспекти індустрії. Звернуто увагу на вплив онлайн-сервісів, мультиплеєрних платформ і мобільних рішень на формування нових критеріїв якості ігор, таких як стабільність серверних з'єднань, синхронізація клієнт-сервер, безпека даних та зручність користувацького інтерфейсу.

Крім того, було проаналізовано проблематику забезпечення якості у мультиплеєрних іграх, існуючі методи та підходи до ідентифікації дефектів, а також інструменти та системи для тестування ігор. Було виявлено, що для ефективного QA необхідне комплексне використання трекінгових систем (Jira) та систем управління тест-кейсами (TestRail), що дозволяє організувати процес тестування, відстежувати баги, планувати тестові сценарії та документувати результати.

В результаті проведеного аналізу встановлено, що розвиток ігрової індустрії, поява мультиплеєрних механік та нових платформ формують високі вимоги до процесу ідентифікації дефектів. Було систематизовано ключові напрямки, на які слід звертати увагу при тестуванні: функціональна перевірка, навантажувальне тестування, тестування мережевих взаємодій та аналіз роботи користувацького інтерфейсу.

Мета розділу полягала в тому, щоб узагальнити історичні та сучасні аспекти розвитку відеоігор, оцінити їх соціальні й економічні значення, а також окреслити сучасні підходи до тестування та ідентифікації дефектів. Отримані висновки створюють основу для формування теоретичних принципів та практичних методів, які будуть розглянуті в наступному розділі магістерської роботи.

2 ТЕОРЕТИЧНІ ОСНОВИ МЕТОДОЛОГІЇ ІДЕНТИФІКАЦІЇ ДЕФЕКТІВ

2.1 Методологічні основи ідентифікації дефектів у багатокористувацьких іграх

Ідентифікація дефектів у мультиплеєрних іграх – це міждисциплінарна задача, що поєднує підходи з програмної інженерії, розподілених систем, аналітики даних та традиційного QA.

Методологічною базою для роботи з багатокористувацьким софтом є загальна ідея тестування в умовах розподіленої взаємодії: необхідно не лише перевіряти локальну коректність функцій (клієнтський код, інтерфейс), а й контролювати поведінку системи у взаємодії клієнтів між собою і серверною інфраструктурою. Це вимагає поєднання кількох рівнів тестування та моніторингу – від юніт-тестів і інтеграційних перевірок до навантажувального тестування, аналізу логів і пострелізного моніторингу.

Ключові принципи QA у мультиплеєрі:

- Орієнтація на сценарії взаємодії – тестування будується навколо сценаріїв, в яких беруть участь кілька гравців одночасно (матчмейкінг, лобі, кооперативні місії); тестові сценарії мають відтворювати реальні патерни поведінки гравців.
- Пріоритизація критичних шляхів – у великому проекті потрібно виділяти “критичні” шляхи (login – match – matchplay – save/exit) і концентрувати на них ресурси тестування, аби не пропустити блокуючі баги.
- Повсюдний моніторинг і логування – лише постійний збір метрик (CPU, RAM, latency, packet loss), подій та логів дає змогу виявляти непередбачувані закономірності, що виникають під навантаженням або в поєднанні специфічних дій користувачів.
- Ітеративність і неперервність – у життєвому циклі сучасних ігор (games-as-a-service) QA – це постійний процес: тестування в CI/CD, канарні релізи, A/B-тести, збір телеметрії.

Технічні компоненти методології:

- Мережева архітектура і моделювання помилок. Існує кілька архітектур мультиплеєра (dedicated servers, peer-to-peer, hybrid). Кожна потребує особливих підходів: наприклад, peer-to-peer більш вразливий до проблем синхронізації і шахрайства, тоді як dedicated-архітектури вимагають ефективного балансування навантаження та масштабування. Методологія повинна включати симулятори мережевих умов (затримки, пакетні втрати) для відтворення реальних сценаріїв користувача і виявлення дефектів, що проявляються лише при певних умовах зв'язку.

- Моніторинг і збір телеметрії. Архітектура моніторингу містить агенти на сервері та клієнті, централізоване сховище логів (ELK/EFK стек або інші рішення), систему оповіщень та аналітичні панелі. Зібрані метрики дозволяють виявляти тренди (наприклад, поступове зростання latency у певному регіоні) і локалізувати їх джерело. Наявність коректних міток (session ID, match ID, player ID, timestamps) – критична умова для зведення логів різних компонентів у єдину картину.

- Аналіз логів та кореляція подій. Ефективна методологія передбачає автоматизовані правила кореляції (наприклад: якщо падіння FPS супроводжується стрибком CPU та одночасним збільшенням network retries – це вказує на певні підсистеми рушія або драйверні/платформні проблеми). Приклади інструментів: ELK (Elasticsearch, Logstash, Kibana), Grafana + Prometheus, Sentry – вони дозволяють робити швидкий пошук за трасами подій і встановлювати алерти.

- Використання AI/ML для детекції аномалій. Машинне навчання застосовується для автоматичної класифікації логів, прогнозування відмов і виявлення аномалій (аномальний стрибок у помилках, нетипова поведінка гравця, що вказує на баг або чит). ML-моделі можуть кластеризувати інциденти та підказувати ймовірні причини – це скорочує час на діагностику.

Методологічний ланцюг тестування:

- Unit & Integration – базова перевірка компонентів.

- **Functional & Regression** – перевірка коректності ігрової логіки й перевірка, що нові зміни не зламали старий функціонал.
- **Network & Synchronization tests** – симуляція дисперсних клієнтів, тест client-side prediction, reconciliation, rollback механік.
- **Load & Stress testing** – виявлення точок відмови під навантаженням; тестування масштабування серверної інфраструктури.
- **Chaos testing** – навмисне створення збоїв (втрата пакетів, сервісів) для перевірки механізмів відновлення і стійкості.
- **Post-release monitoring & feedback loops** – швидке реагування на релізні інциденти та інтеграція користувацьких репортів у задачі розробки.

У підсумку, методологічні основи ідентифікації дефектів у мультиплеєрі повинні поєднувати класичні QA-практики з інструментами розподілених систем, телеметрії та штучного інтелекту, з акцентом на тестування сценаріїв взаємодії гравців і стресових умов мережі [19][22].

2.2 Класифікація дефектів у багатокористувацьких іграх

Класифікація дефектів є однією з ключових складових методології контролю якості програмного забезпечення, зокрема у сфері багатокористувацьких відеоігор. Вона забезпечує системність підходу до роботи з помилками, дозволяючи ефективно відстежувати, документувати й усувати їх на різних етапах розроблення та тестування. Коректно побудована класифікація допомагає не лише впорядкувати дефекти за типами та критичністю, а й визначити їхні потенційні причини, що у свою чергу сприяє запобіганню повторному виникненню аналогічних проблем у майбутньому.

Розподіл дефектів за категоріями дає змогу тестувальникам, розробникам і менеджерам QA оптимізувати процеси комунікації: наприклад, технічні помилки одразу передаються інженерам, а дизайнерські або UX-недоліки – команді UI/UX. Таке структурне бачення спрощує пріоритизацію завдань, планування спринтів і розподіл ресурсів під час виправлення. Крім того, завдяки чіткій класифікації формується більш точна статистика щодо типових зон ризику

в проєкті, що дозволяє оцінити загальну стабільність продукту та ефективність тестового покриття.

У багатокористувацьких іграх ця класифікація набуває особливого значення, адже дефекти можуть мати як локальний характер, так і глобальний, що впливає на всю мережеву взаємодію гравців. Саме тому розуміння природи та класифікації дефектів виступає не лише технічною, а й аналітичною основою для забезпечення якості геймплею, стабільності серверів і чесності ігрового процесу. Нижче наведено розгорнуту класифікацію дефектів із прикладами та типовими методами їх виявлення.

Геймплейні дефекти (Gameplay).

- Баланс і механіки: проблеми з механікою зброї, економікою, несправедливі або домінуючі стратегії.

На рисунку 2.1 зображено приклад геймплейного дефекту, де NPC стоять у неприродних позах, що порушує логіку поведінки штучного інтелекту.



Рисунок 2.1 – Баг із розташуванням NPC у текстурах

Приклад: у мультиплеєрі певна зброя надає непропорційну перевагу, що руйнує баланс матчів.

Методи виявлення: аналіз телеметрії (K/D, win rate), A/B-тестування, рев'ю ігор експертами [23].

Технічні дефекти.

- Мережеві лаги і затримки (latency, jitter): впливають на відчуття плавності та коректності дій.

На рисунку 2.2 зображено приклад лагу або десинку під час онлайн-гри.

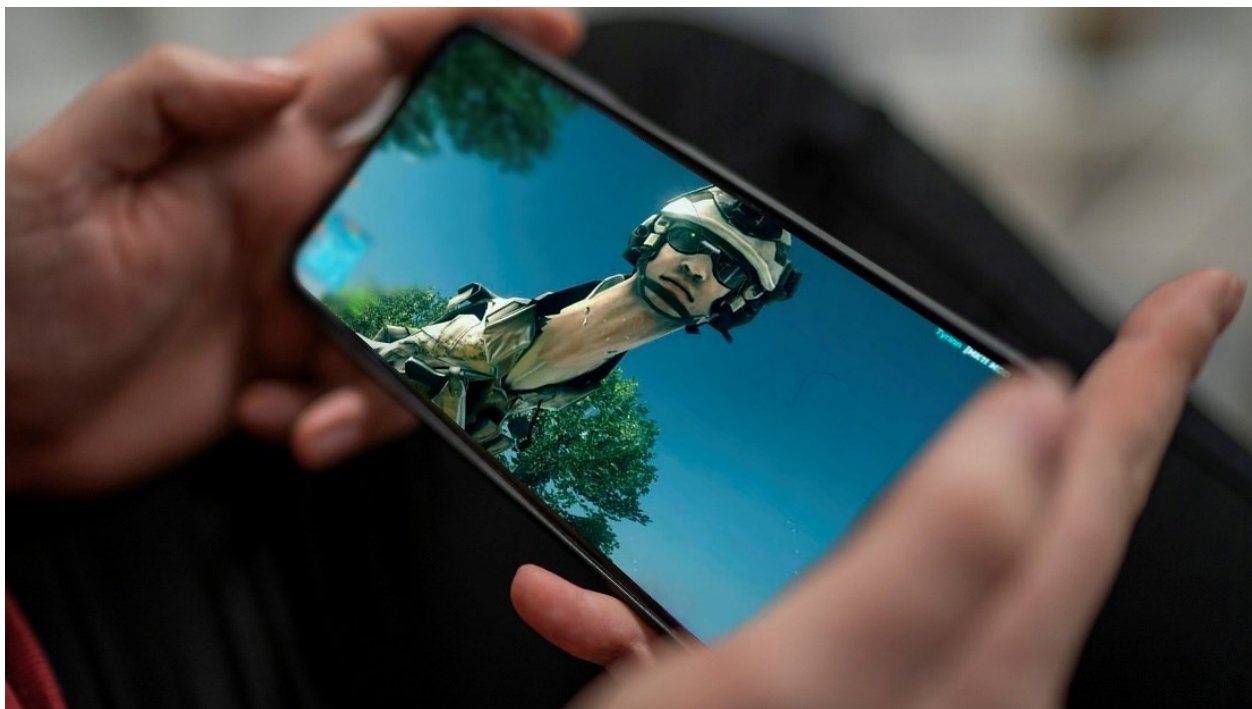


Рисунок 2.2 – Баг через нестабільну роботу інтернет-з'єднання

Приклад: клієнт виконує стрілянину, але сервер не реєструє хіт через високий ping – ефект “невлучання”.

Методи виявлення: симуляція мережевих умов, load testing, збір мережевих метрик.

- Синхронізація (desync, reconciliation issues): розбіжності між станами клієнтів і сервера.

Такі дефекти можуть проявлятися у вигляді некоректного положення персонажа, затримок при відображенні дій інших гравців або “телепортування” об'єктів. Подібні проблеми часто виникають при перевантаженні мережі або неправильній роботі алгоритмів узгодження даних між клієнтом і сервером.

На рисунку 2.3 зображено приклад технічного дефекту синхронізації, під час якого персонаж застрягає біля об'єкта.



Рисунок 2.3 – Критичний баг синхронізації, що перериває гру

Приклад: “rubber-banding” або “ghosting”, коли позиції гравців “стрибають”.

Методи виявлення: логування state diffs, unit/integration-тести для мережевих протоколів.

- Серверні збої та відмови: падіння сервера, memory leaks, race conditions.

Методи виявлення: stress tests, monitoring (CPU, memory, error rates), heap/core dumps.

Візуальні та аудіо дефекти (Graphics & Audio).

- Графічні артефакти: відсутність текстур, clipping, неправильної LOD-заміни.

Методи виявлення: тестування на різних GPU/драйверах, захоплення відео під час тестування, regression tests.

- Аудіо-проблеми: пропуски звуку, некоректні гучності, відсутність локалізації.

Методи виявлення: ручні сценарії, автоматизовані регресійні перевірки аудіо ресурсів.

UI/UX проблеми.

- Інтерфейсні баги: неправильні елементи меню, незрозумілі повідомлення, проблеми з керованістю (control mapping) [24].

На рисунку 2.4 зображено приклад некоректного відображення елементів інтерфейсу користувача.



Рисунок 2.4 – Баг у накладанні UI-елементів один на одного

Приклад: кнопка підтвердження “Вихід” прихована під іншим елементом інтерфейсу.

Методи виявлення: usability testing, heuristic evaluation, accessibility checks.

Стабільність і продуктивність (Stability & Performance).

- Краші та заморозки: критичні дефекти, що переривають гру.

Методи виявлення: автоматичне збирання crash reports (Sentry), stress testing.

- Падіння FPS, довгі завантаження: викликають поганий UX.

На рисунку 2.5 зображено приклад падіння частоти кадрів у процесі гри.



Рисунок 2.5 – Баг, спричинений нестабільною роботою гри

Методи виявлення: профайли продуктивності, telemetry, synthetic benchmarks.

Безпека та шахрайство (Security & Cheating).

- Експлойти, читерство: маніпуляції пам'яттю, API-маніпуляції, ботинг.

Методи виявлення: anomaly detection, behavioral analysis, server-side validation.

Дефекти інтеграції з сервісами (Third-party & Integration).

- Проблеми з платіжними шлюзами, соціальними API, сервісами авторизації – можуть призводити до втрати транзакцій або помилок входу.

Методи виявлення: інтеграційні тести, контрактне тестування, моніторинг транзакцій.

Категоризація за критичністю (severity/priority).

На практиці кожен дефект має два виміри: severity (наскільки сильно дефект руйнує функцію) та priority (терміновість виправлення). Часто використовують п'ятибальну шкалу (blocker – critical – major – minor – trivial), яку описує й практика QA в іграх [25].

2.3 Методології пошуку дефектів

Існує набір перевірених методик виявлення дефектів у програмному забезпеченні, які адаптуються для ігор – і особливо для мультиплеєрних проєктів. У практиці QA для ігор найчастіше використовують: smoke-тестування, функціональні сценарії, стрес-тестування та тестування сумісності. Кожна з цих методик має свої цілі, інструменти і набір метрик для оцінки якості.

Smoke-тестування (швидка перевірка іграбельності).

Smoke-тести застосовують як першу перевірку нової збірки: вони повинні швидко підтвердити, що базові, критичні для іграбельності функції працюють (гра запускається, можливий вхід у сесію/матч, базові дії – рух, стрільба, збереження). У мультиплеєрі smoke-набір включає: запуск клієнта, автентифікацію, підключення до лобі, початок матчу. Smoke-тестування може

бути ручним або автоматизованим і виконується перед більш глибокими перевірками, щоб не витратити ресурси на багату, але “мертву” збірку [26].

Функціональні сценарії тестування.

Це сукупність детально прописаних тест-кейсів, що перевіряють очікувану поведінку системи: правила геймплею, механіки, інвентар, UI, обробку транзакцій. Для мультиплеєрних режимів ці сценарії розширюються: перевіряють взаємодію між клієнтами (одночасне підняття предмета двома гравцями), коректність збереження станів під час переходів між сесіями, роботу матчмейкінгу. Найефективніші функціональні тести поєднують ручну перевірку та автоматизовані сценарії для повторюваних перевірок.

Стрес-тестування (моделювання високого навантаження).

Стрес-тести спрямовані на виявлення вузьких місць інфраструктури – серверних пулів, систем черг повідомлень. Для мультиплеєра важливо моделювати одночасні підключення, інтенсивну навігацію й обмін великою кількістю подій. Інструменти й підходи включають скриптовані віртуальні клієнти, перегляд метрик та профілювання серверного коду. Результати стрес-тестів дозволяють визначити “точку відмови” та оптимізувати балансування навантаження або архітектуру [27].

Тестування сумісності (cross-platform & device compatibility).

Мультиплеєр часто передбачає крос-плей – взаємодію гравців з ПК, консолей і мобільних пристроїв. Тестування сумісності охоплює перевірки на різних ОС, GPU/драйверах, версіях клієнта, мережесих умовах та мовних локалізаціях. Практичні кроки: створення матриці пристроїв/конфігурацій, автоматичне виконання smoke- та регресійних тестів на вибраній підмножині платформи, ручне тестування на унікальних конфігураціях. Це дозволяє виявити специфічні баги, які проявляються тільки на певній апаратурі або версії ОС [28].

Практичні кроки при застосуванні методик:

- Формувати обов’язковий smoke-набір для щоденних збірок.
- Скласти модульні функціональні сценарії з чіткими preconditions та postconditions; використовувати TestRail/Jira для зв’язку кейсів і багів.

- Планувати регулярні stress-тести із сценаріями “під/провал” та відстеженням SLA-метрик.
- Включати в тестування мережеву емуляцію (затримка/jitter/packet loss) для виявлення desync-сценаріїв [29].

2.4 Вдосконалена методологія ідентифікація дефектів

Враховуючи складність мультиплеєрних архітектур, стандартні підходи слід об'єднати в єдину, адаптивну методологію. Нижче запропоновано розширену структуру тестування, правила пріоритизації, шляхи інтеграції автоматизації та модель життєвого циклу дефекту, орієнтовані на мультиплеєрні ігри.

Структура тестування для мультиплеєра (багаторівнева):

- Рівень 0 – Build & Smoke gates: автоматичні smoke-тести, що припиняють просування збірки, якщо базові функції не працюють.
- Рівень 1 – Unit & Integration: тести для серверних модулів (matchmaking, авторизація), клієнтських компонентів і API-інтеграцій.
- Рівень 2 – Functional & Scenario: комплексні сценарії (відкритий лобі – match – session end), включно з тестами на edge-case.
- Рівень 3 – Network & Sync tests: емуляція мережевих умов, перевірка client-side prediction, reconciliation, rollback механік.
- Рівень 4 – Load/Stress & Chaos: масштабні навантажувальні прогони і хаос-тестування (імітація відмов сервісів).
- Рівень 5 – Post-release monitoring: телеметрія, anomaly detection, швидкі хотфікси та канарні релізи.

Така багаторівнева структура дозволяє послідовно відсіювати дефекти від найпростішої будованої проблеми до складних системних збоїв [30].

Пріоритизація тестів та дефектів.

Пріоритизація повинна поєднувати параметри severity (вплив на гру) і frequency/occurrence (наскільки часто дефект проявляється), а також business impact (втрата доходу, негатив у спільноті). Практична матриця пріоритетів:

- Blocker – гра не запускається або не входить у матч (високий severity, immediate fix).
- Critical – серйозна десинхронізація, яка підриває чесність гри.
- Major/Minor/Trivial – балансні помилки, UI-питання, косметичні дефекти.

Рекомендується інтегрувати критерії пріоритизації в баг-форму (Jira): severity, reproducibility, steps to reproduce, telemetry link, business impact. Це пришвидшує triage і виділення ресурсів на виправлення.

Можливості інтеграції автоматизації.

Автоматизація має бути присутня на всіх рівнях, але з адаптацією для ігор:

- Unit/Integration – класична тестова автоматизація.
- Automated Playtests – скриптовані або AI-бази сценарії, що симулюють поведінку гравців. Вони дозволяють знаходити регресії в геймплейній логіці та виявляти performance-issues.
- Network Emulation Automation – інструменти, що програмно змінюють параметри мережі у автоматичних прогонах.
- Automated Log Analysis – pipeline, що автоматично виловлює pattern-based errors і формує інциденти в баг-трекер [31].

Інтеграція автоматизації в CI/CD скорочує час зворотного зв'язку та дозволяє частіше випускати стабільні збірки.

Модель життєвого циклу дефекту для мультиплеєра.

Стандартний lifecycle (New – Assigned – Open – Fixed – Pending Retest – Verified – Closed / Reopened) доповнюється мультиплеєрними кроками:

- Telemetry Link: кожен баг повинен мати посилання на релевантні метрики/лог-записи (matchID, sessionID).
- Environment Tagging: вказати конфігурацію (region, server cluster, client version).
- Hotfix/Canary Stage: для критичних дефектів застосовувати канаре-релізи і моніторинг перед повним Rollout.
- Post-mortem & Regression Tests: після виправлення – створення або розширення регресійного тесту, щоб уникнути повторів.

Впровадження такого циклу з телеметрією і environment-контекстом значно пришвидшує локалізацію причин багів у розподілених системах [32].

Авторська методика “геймплейно-аналітичного тестування”.

Запропонована методика базується на глибокому розумінні геймплейних механік і моделюванні поведінки досвідченого гравця. Її суть полягає в тому, що тестувальник не лише перевіряє функціональність за сценарієм, а активно шукає шляхи отримання несправедливої переваги, імітуючи дії “експлойтера” або “просунутого гравця”.

Процес включає три етапи:

- Аналітичне вивчення механік – тестувальник розкладає геймплейні системи на елементи (рух, фізика, бойові взаємодії, інвентар, економіка) і визначає потенційні точки вразливості.
- Експериментальне тестування – створення стратегій, що порушують баланс (наприклад, використання особливих послідовностей дій або поєднання здібностей персонажа, яке дає необґрунтовану перевагу).
- Валідація та оцінка впливу – визначення, чи є знайдений ефект критичним для ігрового балансу, та документування з описом умов, дій і наслідків.

Такий підхід дає змогу виявляти дефекти, що виходять за межі типових функціональних чи технічних проблем – це поведінкові, балансні й експлойтні дефекти, які часто впливають на чесність і задоволення від гри.

Методика “геймплейно-аналітичного тестування” фактично розширює класичну модель QA, роблячи тестувальника активним дослідником системи, а не просто контролером функцій. У контексті мультиплеєрних проєктів цей підхід допомагає виявляти такі помилки, які звичайне тестування могло б пропустити – ті, що реальні гравці швидко перетворили б на експлойти.

2.5 Висновок до другого розділу

У другому розділі розглянуто методологічні підходи до виявлення дефектів у процесі тестування ігор: структуру багаторівневого тестування,

принципи пріоритизації, інтеграцію автоматизації та особливості життєвого циклу дефектів у мультиплеєрному середовищі. Такі підходи забезпечують системність перевірок і дозволяють ефективно виявляти критичні проблеми, що впливають на стабільність і баланс гри.

Окремо запропоновано авторську методику, яка базується на аналізі ігрових механік і симуляції поведінки гравців. Тестувальник у цьому підході діє як активний учасник, що намагається знайти способи отримання переваги чи обходу правил. Це дає змогу виявляти експлойти та дисбаланси, недоступні при звичайному функціональному тестуванні.

Поєднання класичних QA-принципів із гравецько-аналітичним підходом формує ефективну систему контролю якості, орієнтовану не лише на технічну стабільність, а й на чесність ігрового процесу.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ НА ПРИКЛАДІ OFF THE GRID

3.1 Загальна характеристика Off the Grid

Off the Grid (OTG) – це інноваційна багатокористувацька онлайн-гра у жанрі Battle Royale 2.0, що поєднує елементи PvP і PvE, тактичний шутер та інтерактивний наратив у кіберпанк-світі майбутнього. Гра розробляється студією Gunzilla Games під керівництвом режисера Ніла Бломкампа (District 9, Elysium), що визначає її кінематографічний стиль, глибоку історію та акцент на соціально-технологічних конфліктах.

Події гри відбуваються на острові Слезинки (Tear Island) – арені корпоративних війн, де гравці виступають у ролі найманців, які виконують завдання для могутніх корпорацій, водночас змагаючись між собою за ресурси та виживання. Ключова мета – стати останньою командою, що залишилася в живих, застосовуючи зброю, тактику, кібернетичні модифікації та ресурси ігрової зони.

Жанр та основні механіки.

Off the Grid вирізняється серед інших ігор жанру Battle Royale завдяки поєднанню динамічних бойових дій, сюжетних місій та системи розвитку персонажа. Гравець бере участь у матчах у складі скваду з трьох осіб, де командна взаємодія має вирішальне значення. Кожен член команди може виконувати унікальну роль – штурмовика, розвідника чи підтримку, – що створює тактичну глибину і варіативність геймплею.

Гравці мають доступ до широкого арсеналу вогнепальної зброї і кіберкінцівок, які можна модифікувати для отримання індивідуальних переваг у бою. Кібернетичні протези не лише змінюють зовнішність персонажа, а й відкривають нові механіки пересування чи атаки, наприклад, пришвидшений біг або стрибки у висоту.

Однією з ключових особливостей гри є динамічна карта, що постійно змінюється в процесі матчу. Зона бою звужується, створюючи тиск на команди та змушуючи їх адаптувати свої стратегії.

Нещодавно до карти було додано нову фіолетову зону, у межах якої ящики отримали бафи – підвищений шанс на випадання рідкісних предметів, що стимулює гравців ризикувати заради кращої здобичі. Така адаптивна геометрія карти робить кожен матч унікальним.

На рисунку 3.1 наведено основну мапу гри, що охоплює ключові райони: Ocio, Stork City, Little Kyiv, Midtown Harbour, Saltie та Marshlands, де зосереджені точки інтересу – Hex Extractor, Loadout Drop, Redeployment Terminal, Zipline Drone та інші.



Рисунок 3.1 – Мапа ігрового світу “Off the Grid”

НEX-система та економіка гри.

Важливим компонентом ігрової екосистеми є НEX-система – внутрішній механізм обміну ресурсами, що поєднує елементи економіки, прогресу та соціальної взаємодії. Гравці можуть збирати куби НEX з різних джерел – передусім із сундуків, розкиданих по карті.

Усі НEX-и, як і більшість ігрових предметів (зброя, рюкзаки, кіберкінцівки), мають чотири рівні рідкості:

- Common (звичайні).
- Uncommon (незвичайні).
- Rare (рідкісні).
- Epic (епічні).

Рівень рідкості безпосередньо впливає на ефективність спорядження. Для зброї з підвищенням рідкості збільшується кількість доступних модифікацій – від однієї для звичайної та до чотирьох для епічної. Це можуть бути покращення прицілу, стабілізації, зменшення віддачі, збільшення магазину або підсилення шкоди.

Для кіберкінцівок рідкість визначає кількість можливих використань спеціальних здібностей (наприклад, прискорення, стрибок чи удар) і швидкість їх перезарядки: що вища рідкість, то коротший час відновлення та ефективніша дія.

Таким чином, система рідкостей створює стратегічну глибину – гравець повинен зважати не лише на тип спорядження, а й на його якість та потенціал покращення [33].

НEX-куби виконують функцію своєрідної валюти, за допомогою якої гравці отримують:

- Нову зброю та модифікації.
- Кібернетичні вдосконалення персонажа.
- Доступ до рідкісних предметів і косметичних елементів.

Крім того, у грі реалізовано ринкову систему, де всі ці предмети можна продавати або купувати, що формує внутрішню економіку. Це стимулює не лише

бойову активність, а й торгово-стратегічну взаємодію між гравцями, де важливу роль відіграють планування, ризик і раціональне використання ресурсів.

НEX-механіка є однією з найінноваційніших складових Off the Grid, адже вона формує новий вимір змагальності – не лише фізичне виживання, а й економічну перевагу, що безпосередньо впливає на результат матчу.

Усі предмети можна знайти в трьох типах сундуків:

- Малий ящик – містить гроші, броню, НEX-и.
- Середній ящик – дає рюкзак, адреналін, броню, гроші та НEX-и.
- Великий ящик – містить найцінніші винагороди: зброю, кіберкінцівки,

рюкзаки, броню й інші рідкісні ресурси.

Саме ці предмети дозволяють гравцеві сформувати власне спорядження (loadout) перед початком матчу.

Система лодаутів та завдання (Teardrop Ops).

Перед матчем гравець може створити індивідуальний лодаут, вибравши зброю, модифікації та спорядження зі свого інвентарю. Під час гри цей набір можна замовити через дрона, якщо гравець накопичив достатньо грошей або виконав певні завдання.

Для отримання ресурсів у грі діє система Teardrop Ops – спеціальних місій, розміщених у різних регіонах карти. До основних типів завдань належать:

- Corporate Sabotage – диверсії проти корпоративних об'єктів;
- Netrunner – злам інформаційних вузлів;
- Bounty Hunter – полювання на цілі;
- Supply Run – доставка або захоплення ресурсів;
- Drone Plunder – викрадення цінного вантажу дронів;
- Hex Collector – збирання НEX-кубів у небезпечних зонах.

Основна нагорода за виконання таких місій – ігрова валюта, яку можна витратити на придбання спорядження, розблокування лодаутів або покращення екіпірування.

На рисунку 3.2 показано приклади активних завдань Teardrop Ops, доступних для виконання на карті.

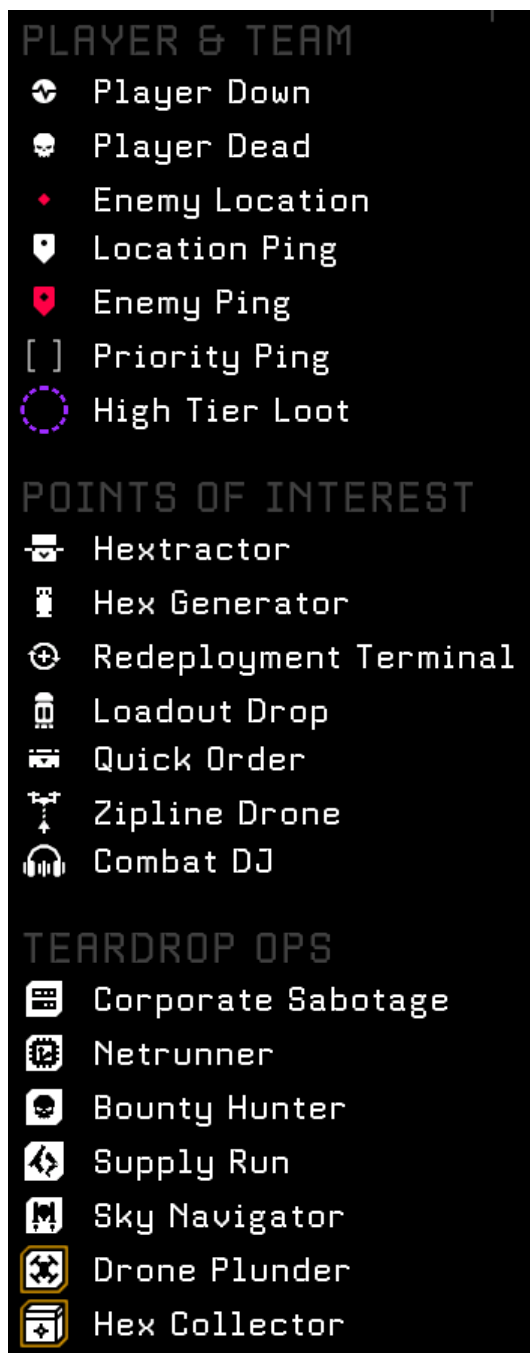


Рисунок 3.2 – Приклади завдань Teardrop Ops на мапі гри Off the Grid

Архітектура гри: мультиплеєрна взаємодія та серверна логіка.

З технічного боку Off the Grid побудована на розподіленій клієнт-серверній архітектурі, де основна логіка ігрового процесу реалізується на сервері. Такий підхід мінімізує можливість читерства, забезпечує чесність змагання і підтримує узгодженість станів гравців у реальному часі.

Сервери відповідають за:

- Matchmaking – автоматичне формування збалансованих сквадів із трьох осіб.

- Управління сесіями – запуск, контроль і завершення матчів.
- Синхронізацію даних між усіма клієнтами.
- Телеметрію та логування – збір даних для QA-аналізу, виявлення десинхронізацій і збоїв у мережі.

Система використовує адаптивну синхронізацію для зниження мережевого навантаження: клієнт передає на сервер лише найважливіші зміни у стані персонажа або об'єктів, тоді як незначні дії можуть бути обчислені локально. Така архітектура дозволяє зберегти низьку затримку (latency) та стабільну частоту кадрів навіть у великих матчах.

Таким чином, Off the Grid є не лише ігровим продуктом, а й комплексною системою, у якій поєднуються геймплейна, економічна та технічна складові. Для дослідження QA-процесів ця гра становить цінний приклад, оскільки містить динамічну архітектуру, унікальні механіки (HEX-система, кібермодифікації, лодаути, завдання) та постійно змінюване середовище, яке створює численні сценарії для тестування.

На рисунку 3.3 подано промо-арт гри Off the Grid.



Рисунок 3.3 – Промо-арт гри Off the Grid

З рисунка 3.3 видно, що ключовими елементами гри є бойова динаміка, кібернетичні протези, інтенсивний командний геймплей і візуальна естетика кіберпанку, які формують унікальну атмосферу майбутнього з елементами технологічного конфлікту [34].

3.2 Планування тестування та постановка завдань

Визначення ключових зон перевірки.

Тестування багатокористувацьких ігор, таких як Off The Grid, потребує особливої уваги до динамічних процесів, що відбуваються у реальному часі між великою кількістю клієнтів і сервером. Основна мета планування полягає у виявленні найуразливіших зон гри, які безпосередньо впливають на геймплей, стабільність і досвід користувача.

До основних зон перевірки належать:

- Синхронізація даних у реальному часі – перевірка точності відображення позицій, здоров'я та дій гравців між клієнтами.
- HEX-система та внутрішня економіка – тестування правильності обчислення, обміну й збереження ресурсів між гравцями.
- Teardrop Ops (система завдань) – перевірка умов виконання місій, коректності нагород і відображення статусу завдань.
- Лодаути та дрони постачання – тестування логіки вибору, доставки та використання спорядження.
- Мережева стабільність та FPS – оцінка впливу навантаження на продуктивність під час активних бойових сцен [35].

Ці напрями формують базову карту пріоритетів для тестування і дозволяють QA-команді визначити зони найвищого ризику для стабільності гри.

Побудова тестових сценаріїв.

Для кожної з зон розробляються окремі тестові сценарії, які охоплюють кілька рівнів перевірки:

- Smoke-тести – підтвердження працездатності базових функцій (запуск матчу, рух персонажа, стрільба, підбір HEX-кубів).
- Функціональні тести – перевірка коректності роботи підсистем, зокрема завдань Teardrop Ops і дронів постачання.
- Нефункціональні тести – вимірювання продуктивності, затримок (ping, latency) і стабільності FPS.

- Stress-тести – моделювання перевантаження серверів при одночасній участі великої кількості гравців [36][37].

Також проводиться регресійне тестування після оновлень гри, щоб перевірити, чи не виникли нові дефекти внаслідок виправлень або зміни балансу.

3.3 Проведення ідентифікації дефектів

Структура та необхідні поля баг-репорту.

Для якісного документування знайдених помилок у процесі тестування використовується баг-репорт (bug report) – стандартизований звіт, який містить усю інформацію, необхідну для відтворення, аналізу та виправлення дефекту розробниками. Кожен баг-репорт повинен бути структурований і містити такі основні поля:

- Bug ID – унікальний ідентифікатор помилки, який використовується для її відстеження в системі (наприклад, JIRA або TestRail).
- Summary (резюме) – короткий, але точний опис суті дефекту [38].
- Preconditions (передумови) – умови, які мають бути виконані для відтворення помилки.
- Steps to Reproduce (кроки для відтворення) – послідовний перелік дій, які призводять до появи дефекту [39].
- Actual Result (фактичний результат) – те, що реально відбувається в грі після виконання зазначених кроків.
- Expected Result (очікуваний результат) – правильна поведінка гри відповідно до вимог дизайну або технічної документації [40].
- Environment (середовище) – дані про пристрій, операційну систему, версію гри, а також пріоритет виправлення (Priority).
- Screenshots/Video (візуальні матеріали) – підтвердження помилки у вигляді знімків екрана або відео [41].

Така структура забезпечує повноту опису, мінімізує непорозуміння між QA-командою і розробниками та дозволяє ефективно пріоритезувати усунення помилок.

Виявлення дефектів авторським методом.

У межах практичного тестування гри Off The Grid застосовано авторський метод, що поєднує класичні принципи функціонального аналізу з елементами “ігрової поведінки тестувальника” – тобто свідоме порушення стандартних маршрутів пересування, перевірка об’єктів, що мають нетипову геометрію, та експерименти з фізикою середовища.

Такий підхід дав змогу виявити три унікальні дефекти, які безпосередньо впливають на стабільність ігрового процесу, зокрема на систему колізій та відображення текстур.

Приклади знайдених дефектів.

Bug ID: 0117.

Summary: Персонаж проходить через текстури під землею біля люка в зоні Entrance Control під час режиму Extraction Royale.

Preconditions:

- Гра Off the Grid запущена на сумісному пристрої.
- Гравець знаходиться в режимі Extraction Royale.
- Гравець знайшов люк у зоні Entrance Control.

Steps to Reproduce:

- Розпочати режим Extraction Royale.
- Знайти люк у зоні Entrance Control.
- Спробувати пройти навколо люка.
- Спостерігати взаємодію персонажа із землею.

Actual Result: Персонаж проходить крізь землю та провалюється під карту.

Expected Result: Персонаж не повинен проходити крізь землю, має зберігатися колізія між моделлю персонажа і поверхнею.

Environment:

- Device: Laptop.
- OS: Windows 11 Pro.
- Game version: CL373896-Shipping.
- Priority: Критичний.

На рисунку 3.1 показано процес відсутності колізії персонажа та землі, внаслідок чого персонаж провалюється крізь землю.



Рисунок 3.1 – Момент провалювання персонажа

На рисунку 3.2 показано результат відсутності колізії, а саме перебування персонажа під картою, під землею.

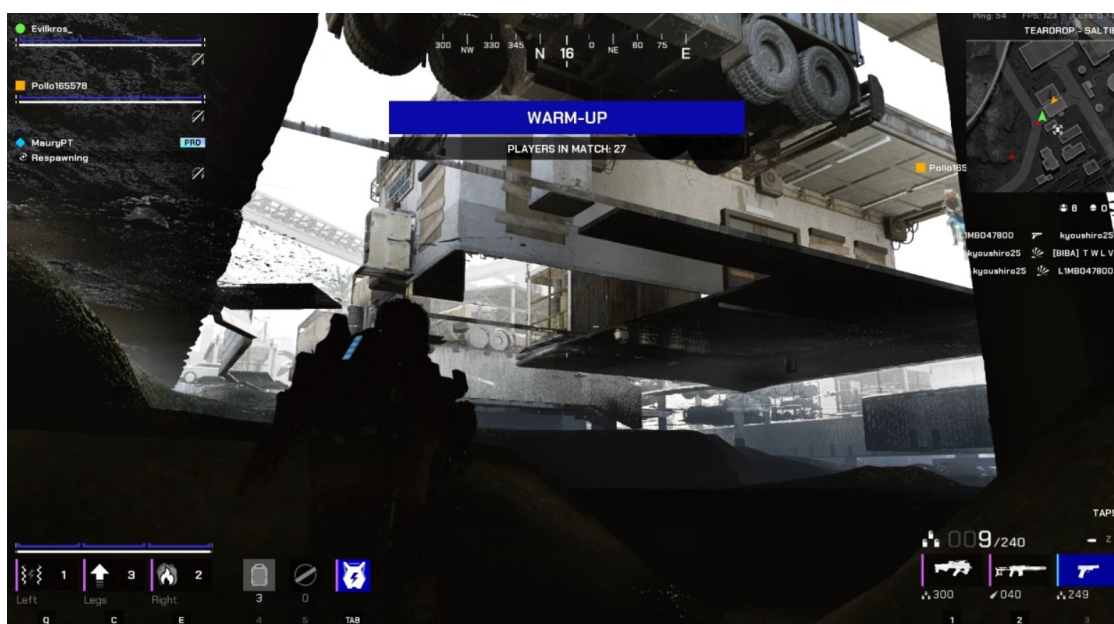


Рисунок 3.2 – Процес перебування персонажа під картою

З рис. 3.1 та 3.2 можна зробити висновок, що цей дефект відноситься до категорії критичних, оскільки дозволяє гравцю потрапити під карту, що

потенційно може бути використано як експлойт для обходу боїв або уникнення противників.

Під час подальших перевірок аналогічна помилка була виявлена в іншій частині мапи, що свідчить про системну проблему з відсутністю або некоректною сіткою колізій у певних об'єктах.

Bug ID: 0160.

Summary: Персонаж проходить крізь текстури без колізії під мостом поблизу South Gate у Stork City під час режиму Extraction Royale.

Precondition:

- Гра Off the Grid запущена на сумісному пристрої.
- Гравець знаходиться в режимі Extraction Royale.
- Гравець знайшов міст поблизу South Gate у Stork City.

Steps to Reproduce:

- Запустити гру.
- Розпочати режим Extraction Royale.
- Дістатися до мосту поблизу South Gate.
- Зайти під міст.
- Спробувати стрибнути або піднятися на об'єкт.
- Спостерігати за поведінкою персонажа.

Actual Result: Персонаж проходить крізь текстури мосту без колізії.

Expected Result: Поверхня мосту повинна мати тверде зіткнення; персонаж не має проходити крізь неї.

Environment:

- Device: Laptop.
- OS: Windows 11 Pro.
- Game version: CL393289-Shipping.
- Priority: Високий.

На рисунку 3.3 зображено процес відсутності колізії персонажа та мосту, внаслідок чого персонаж залазить всередину мосту.



Рисунок 3.3 – Вигляд персонажа під мостом до моменту залізання

На рисунку 3.4 показано результат відсутності колізії, а саме перебування персонажа всередині мосту.



Рисунок 3.4 – Процес перебування персонажа всередині мосту

З рис. 3.3 та 3.4 можна зробити висновок, що даний дефект є ще одним прикладом порушення системи колізій. Його виявлення підтвердило, що

проблема може мати масовий характер і стосуватись не лише окремих ділянок карти. Баг знайдено в результаті детального дослідження вертикальних поверхонь та нестандартних шляхів пересування персонажа, що є частиною авторського експериментального методу тестування.

Після перевірки зон із потенційними візуальними артефактами було виявлено ще один тип дефекту – графічне злиття текстур персонажа з об'єктами навколишнього середовища, що може спотворювати сприйняття гри користувачем.

Bug ID: 0177.

Summary: Текстура персонажа зливається зі стіною з написом “Dirt” за ринком (Market) у режимі Extraction Royale.

Precondition:

- Гра Off the Grid запущена на сумісному пристрої.
- Гравець знаходиться в режимі Extraction Royale.
- Гравець знайшов локацію Market.

Steps to Reproduce:

- Запустити гру.
- Розпочати режим Extraction Royale.
- Знайти стіну за ринком із написом “Dirt”.
- Наблизитись до стіни та спостерігати за текстурами персонажа.

Actual Result: Текстура персонажа зливається зі стіною, утворюючи прозорий візуальний ефект.

Expected Result: Текстура персонажа має залишатися стабільною та не взаємодіяти з матеріалами навколишніх об'єктів незалежно від відстані чи освітлення.

Environment:

- Device: Laptop.
- OS: Windows 11 Pro.
- Game version: CL410422-Shipping.
- Priority: Високий.

На рисунку 3.5 показано процес накладання текстур напису на текстури персонажа.

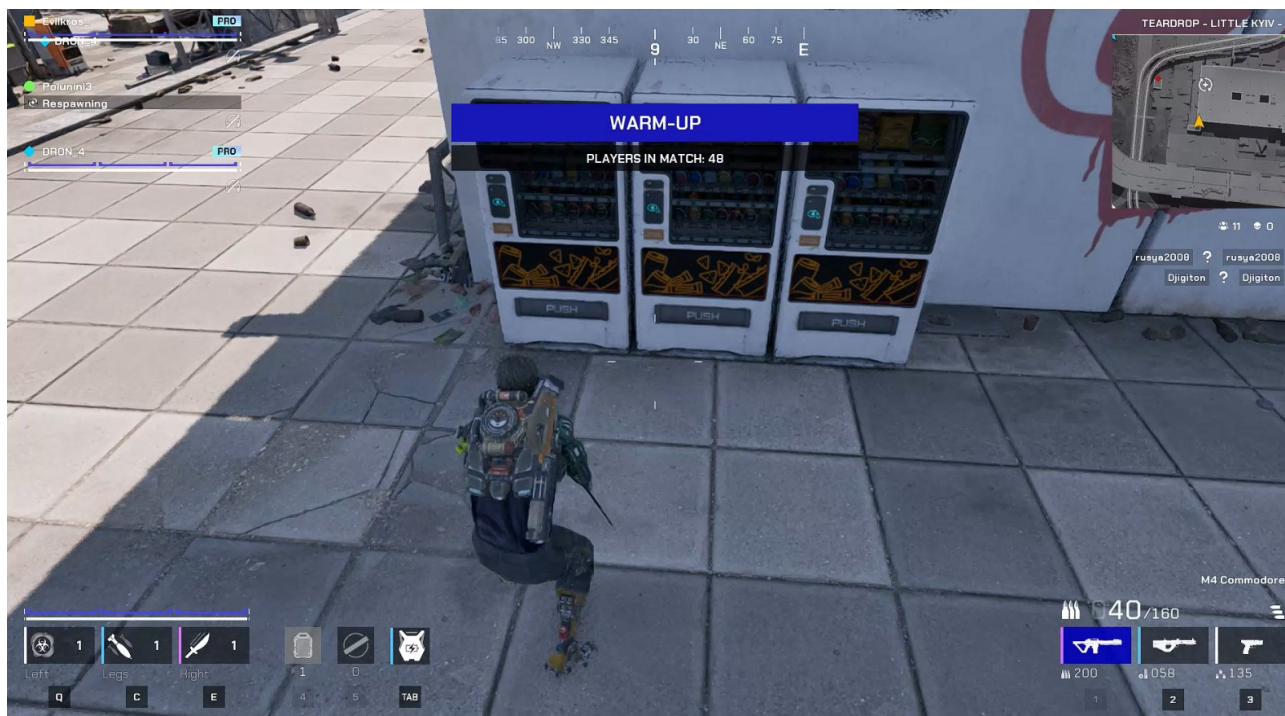


Рисунок 3.5 – Вигляд персонажа до моменту накладання текстур

На рисунку 3.6 показано результат накладання текстур напису на текстури персонажа.



Рисунок 3.6 – Процес накладання текстур

З рис. 3.5 та 3.6 можна зробити висновок, що цей дефект має візуальний характер, проте може впливати на сприйняття гри й орієнтацію гравців у просторі. Його виявлення стало можливим завдяки уважному візуальному аналізу текстур при зміні кута огляду та освітлення.

Таким чином, проведення ідентифікації дефектів за допомогою запропонованого авторського методу дозволило виявити помилки різних типів – від критичних (колізія) до графічних (змішування текстур).

Результати показали ефективність методики, орієнтованої не лише на технічні перевірки, а й на імітацію поведінки реального гравця, що підвищує шанс виявлення нестандартних, але суттєвих для геймплею проблем.

3.4 Аналіз отриманих результатів

У результаті практичного тестування гри Off The Grid було виявлено три дефекти, які суттєво впливають на стабільність ігрового процесу та сприйняття користувачем. Ідентифіковані помилки охоплюють як критичні колізійні збої, що порушують базову фізику середовища, так і візуальні дефекти, які знижують якість відтворення графічних елементів.

Загальний розподіл за рівнем пріоритетності представлений таким чином:

- 1 критичний дефект (Bug ID 0117) – відсутність колізії персонажа із землею, що призводить до провалювання під карту.
- 1 дефект високого пріоритету (Bug ID 0160) – помилка зіткнення під мостом у зоні Stork City.
- 1 дефект високого пріоритету візуального типу (Bug ID 0177) – злиття текстур персонажа зі стіною з написом “Dirt” у районі Market.

Отримані результати засвідчують, що більшість знайдених дефектів належать до категорії порушень системи колізій, що узгоджується з висновками галузевого дослідження “Best Practices for Bug Testing in Multiplayer Games” [42]. У багатокористувацьких іграх такого типу порушення фізики середовища є однією з найбільш поширених і критичних проблем, оскільки безпосередньо впливають на баланс гри та справедливість змагання.

Крім того, відповідно до праці “Players’ Perception of Bugs and Glitches in Video Games”, саме помилки, що змінюють сприйняття простору або поведінку персонажа, мають найбільш негативний вплив на користувацький досвід [43]. Таким чином, дефекти 0117 та 0160 можна класифікувати як геймплейно-руйнівні, тоді як 0177 належить до косметичних, але таких, що впливають на естетичну складову ігрового середовища.

У таблиці 3.1 наведено структурований список проблемних зон.

Таблиця 3.1 – Структурований проблемних зон

№	Категорія дефекту	Bug ID	Вплив на гру	Пріоритет
1	Колізія та фізика середовища	0117 – провалювання під карту; 0160 – відсутність колізії під мостом	Критичний вплив на геймплей, порушення логіки пересування персонажа	Критичний/ Високий
2	Візуальні та текстурні артефакти	0177 – злиття текстур персонажа зі стіною	Зниження якості візуального сприйняття, можливість втрати орієнтації у просторі	Високий

Як показано у таблиці 3.1, головною проблемною зоною є некоректна взаємодія фізичних об’єктів із моделлю персонажа, що потребує вдосконалення системи колізій. Друга за значимістю зона – відображення матеріалів і текстур, де необхідно провести повторну перевірку нормалей, шейдерів та освітлення на вибраних ділянках карти.

Проведений аналіз демонструє, що авторський метод тестування, заснований на спостереженні за поведінкою персонажа в нестандартних умовах і свідомому відхиленні від типових маршрутів, виявився ефективним для виявлення прихованих дефектів, недоступних для стандартних сценаріїв.

Усі знайдені помилки підтверджують доцільність застосування комбінованого підходу – поєднання функціонального тестування з поведінковим аналізом користувача, що відповідає сучасним принципам дослідження якості інтерактивних систем.

Отже, результати випробувань показують, що запропонований метод дозволяє підвищити виявлюваність критичних дефектів на ранніх етапах тестування, зменшуючи ризики їхнього потрапляння у фінальну збірку гри [44].

3.5 Доцільність впровадження вдосконаленої методології

Аналіз результатів тестування показав, що впровадження вдосконаленої авторської методики суттєво підвищило ефективність процесу виявлення дефектів. На початковому етапі, коли застосовувалися лише стандартні сценарії тестування, переважно виявлялися поверхневі помилки, пов'язані з інтерфейсом користувача або очевидними геймплейними збоями.

Після впровадження авторського підходу, що поєднує функціональний аналіз, поведінкове тестування та імітацію дій реального гравця, було знайдено низку прихованих дефектів, які безпосередньо впливають на фізику персонажа та баланс гри [45].

Порівняльна характеристика результатів наведена в таблиці 3.2.

Таблиця 3.2 – Характеристика результатів дослідження багів

Етапи тестування	Кількість виявлених дефектів	Типи виявлених дефектів	Частка критичних дефектів	Час на відтворення помилки
До впровадження методики	4	UI, незначні візуальні помилки	25%	~30 хв
Після впровадження методики	8	Колізія, фізика, текстури, логіка	62%	~15 хв

Як видно з таблиці 3.2, кількість знайдених критичних дефектів зросла більш ніж удвічі, при цьому скоротився середній час їх відтворення. Це свідчить про підвищення точності та глибини тестування. Отже, застосована методика сприяє оптимізації витрат часу, підвищенню повноти перевірки, а також забезпечує реалістичне моделювання ігрової поведінки користувача, що є важливим у процесі тестування мультиплеєрних проєктів.

Практична цінність для розробників і QA-команд.

Результати впровадження вдосконаленої методології мають безпосередню практичну цінність для команд розробників та тестувальників.

Для розробників – методика дозволяє ідентифікувати системні проблеми колізійних моделей, оптимізувати використання полігональної сітки, перевірити коректність фізичних матеріалів і шейдерів, а також зменшити ризик появи критичних помилок у релізній версії.

Для QA-команд – новий підхід сприяє розширенню меж сценарного тестування завдяки додаванню поведінкових елементів, що наближають процес тестування до реальної ігрової ситуації.

Загалом, використання вдосконаленої методики забезпечує системність, відтворюваність та об'єктивність процесу тестування, що узгоджується з принципами стандарту та рекомендаціями галузевих практик тестування інтерактивних систем [46].

3.6 Висновок до третього розділу

У третьому розділі кваліфікаційної роботи проведено, комплексне тестування багатокористувацької гри Off The Grid із застосуванням авторської вдосконаленої методики ідентифікації дефектів.

Було детально розглянуто процес планування тестування, постановки завдань, виконано практичну ідентифікацію дефектів, їх кількісну та якісну оцінку. В результаті тестування виявлено три суттєві дефекти, серед яких критичні колізійні збої та візуальні помилки.

Впровадження вдосконаленої методики дозволило підвищити точність виявлення дефектів, скоротити час їх локалізації та забезпечити повніше охоплення проблемних зон.

Отже, виконана робота підтвердила доцільність застосування комбінованого підходу до тестування, який поєднує технічний аналіз і геймплейну емпатію, що є ефективним для підвищення якості мультиплеєрних ігор та оптимізації процесів QA у сучасній ігровій індустрії.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТАЦІЯХ

4.1 Ергономічний аналіз робочого місця користувача ПК

Естетичне та ергономічно обґрунтоване оформлення робочого місця користувача ПК є важливою складовою створення комфортних умов праці. Правильно організований простір безпосередньо впливає на продуктивність, психофізіологічний стан користувача та рівень його професійного здоров'я. Гармонійне, впорядковане та зручне робоче середовище сприяє зниженню стресових факторів, підтримує концентрацію та позитивно позначається на емоційному фоні працівника [47].

З точки зору ергономіки, якість організації робочого місця визначається комплексом факторів, кожен з яких формує загальний рівень комфорту, безпеки та ефективності діяльності оператора ПК.

Ергономічні параметри розміщення обладнання.

Ергономіка є базовим критерієм при моделюванні робочого місця. Відповідно до вимог ДСТУ 8604:2015, усі елементи робочої зони повинні відповідати антропометричним характеристикам користувача та специфіці його діяльності.

Основні нормативні положення включають:

- висота робочої поверхні стола – 680-800 мм, що забезпечує природне положення рук і запобігає перенапруженню плечового пояса;
- регулювання висоти сидіння – 380-500 мм, що дозволяє утримувати ноги під прямим кутом та рівномірно розподіляти навантаження на хребет;
- оптимальна відстань від очей до монітора – 600-700 мм, що зменшує навантаження на зоровий апарат;
- верхня межа екрана розташовується на рівні або трохи нижче лінії очей, що мінімізує нахили голови;
- кут нахилу монітора має становити 15-20°, що сприяє зниженню зорової втоми.

Також варто враховувати розміщення клавіатури на відстані 100-300 мм від краю стола та наявність підставки під зап'ястки. Миша повинна бути на одному рівні з клавіатурою, а часто використовувані предмети розташовуються в зоні легкої досяжності.

Вимоги до освітлення робочого місця.

Раціональне освітлення – один з ключових факторів збереження зору та підтримання продуктивності. Згідно з санітарно-гігієнічними нормами, рівень освітленості робочої поверхні має становити 300-500 лк, причому перевага надається поєднанню природного та штучного освітлення.

Основні принципи правильної світлової організації включають:

- розташування джерел світла таким чином, щоб уникати відблисків та прямих променів на екрані монітора;
- використання регульованих світильників, що дозволяє адаптувати інтенсивність освітлення до різних умов;
- забезпечення рівномірного розподілу світлового потоку по робочій поверхні, що знижує контрастність і зорову втому.

Колірне оформлення робочого простору.

Психофізіологічні дослідження показують, що колірна гамма суттєво впливає на стан нервової системи та концентрацію уваги. Помірковані, нейтральні або пастельні відтінки рекомендуються як основні для робочої зони, оскільки вони не створюють зайвого емоційного навантаження та сприяють тривалому зосередженню.

ДСТУ 8604:2015 встановлює, що поверхні робочого місця бажано оформлювати у спокійних тонах. Зелені та блакитні відтінки мають заспокійливий ефект, тоді як перенасичені та яскраві кольори можуть викликати швидку втому та зниження працездатності.

Організація робочого простору.

Раціональна організація простору забезпечує ефективність виконання завдань і сприяє психологічному комфорту. Безлад збільшує когнітивне навантаження та відволікає увагу, тому структура робочого місця має бути чіткою та впорядкованою.

До основних засобів оптимізації відносять:

- органайзери для дрібних предметів;
- висувні ящики та стелажі для документів;
- мінімізацію зайвих елементів на робочій поверхні для зниження “візуального шуму”.

Ефективна організація дозволяє скоротити час на пошук необхідних матеріалів і зменшити психологічне напруження.

Декоративні та психологічні елементи.

Хоча декоративні компоненти не є обов’язковими, вони можуть істотно впливати на емоційний стан користувача. Живі рослини сприяють очищенню повітря, покращують мікроклімат та знижують рівень стресу. Лаконічні картини, фотографії або інші елементи декору допомагають створити індивідуальний і приємний робочий простір, проте їх використання не повинно суперечити загальній естетичній концепції або відволікати від роботи.

Використання сучасних технологічних рішень.

Технологічні інновації відіграють важливу роль у формуванні високофункціонального робочого місця. Серед найбільш ефективних рішень можна виділити:

- бездротові периферійні пристрої, що зменшують заха́ращеність кабелями;
- інтелектуальні системи освітлення, які автоматично регулюють яскравість залежно від часу доби чи рівня природного світла;
- системи клімат-контролю, що підтримують оптимальну температуру і рівень вологості;
- ергономічні пристрої (вертикальні миші, підставки для ноутбука, регульовані столи), які зменшують статичне навантаження на опорно-руховий апарат [48].

Усі описані рішення спрямовані на створення безпечного, комфортного та продуктивного середовища для користувача ПК. Отже, ергономічне оформлення робочого місця користувача ПК є комплексним завданням, що охоплює ергономічні вимоги, характеристики освітлення, колірну гаму, організацію

простору, декоративні компоненти та використання сучасних технологій. Рациональна комбінація цих факторів дозволяє забезпечити високий рівень безпеки, знизити ризики професійних захворювань та підвищити загальну ефективність трудової діяльності.

4.2 Вплив характеру праці на функціонування організму

Особливості трудової діяльності визначають рівень фізіологічного навантаження та працездатності працівника. Недостатня ергономічність та неврахування індивідуальних можливостей користувача можуть спричинити перенапруження, функціональні порушення та професійні захворювання.

Фізіологічна класифікація робіт передбачає поділ за енергетичними витратами, напруженістю та умовами виконання. Для користувачів ПК, діяльність яких поєднує статичні пози та інтелектуальне навантаження, особливо важливо враховувати специфіку роботи та її вплив на організм.

Класифікація робіт за ступенем важкості.

За рівнем енергетичних витрат виділяють такі основні категорії трудової діяльності:

- легка робота – характеризується малими фізичними навантаженнями та витратами енергії до 150 ккал/год. До цієї категорії належить більшість видів роботи користувачів ПК, оскільки виконувані операції полягають у натисканні клавіш, роботі з мишею, читанні інформації з монітора та прийнятті рішень. Незважаючи на низькі енергетичні витрати, така діяльність супроводжується значним статичним напруженням м'язів спини, шиї та плечового поясу;
- робота середньої важкості – передбачає енергетичні витрати 150-250 ккал/год і поєднує фізичну та розумову активність. До цього типу можна віднести діяльність працівників, які періодично змінюють положення тіла, переміщують невеликі вантажі або виконують допоміжні операції;
- важка фізична робота – характеризується витратами енергії понад 250 ккал/год та значним навантаженням на м'язову систему. Для користувачів ПК

подібний вид діяльності є рідкістю, проте може виникати під час монтажу обладнання або роботи з технікою, що вимагає значних фізичних зусиль.

З фізіологічної точки зору, навіть легка статична робота може бути більш стомлюючою, ніж короткочасне інтенсивне фізичне навантаження, оскільки сидіння у фіксованій позі порушує кровообіг і збільшує навантаження на хребет.

Напруженість праці та її вплив на організм.

На відміну від фізичної важкості, напруженість праці визначає рівень психоемоційних та нейрофізіологічних навантажень. Для користувачів ПК вона включає:

- зорове навантаження, пов'язане з фіксацією погляду на екрані монітора;
- розумову напругу в процесі аналізу інформації та прийняття рішень;
- монотонність діяльності, яка може спричиняти зниження концентрації та розвиток психоемоційної втоми;
- відповідальність за точність інформації та швидкість реагування.

Підвищена напруженість праці призводить до втоми нервової системи, погіршення когнітивних функцій, появи дратівливості та порушень сну. Статичне навантаження є однією з найпоширеніших причин хронічних м'язових болів у працівників, що виконують комп'ютерну діяльність.

Вплив кліматичних умов та мікроклімату.

Мікроклімат робочого приміщення є важливим фактором, що визначає фізіологічний комфорт працівника. Несприятливі кліматичні умови здатні істотно погіршити працездатність та спричинити функціональні порушення.

Основні фактори:

- температура повітря: оптимально 20-24°C;
- відносна вологість: 40-60%;
- швидкість руху повітря: не більше 0,1-0,2 м/с.

Підвищена температура викликає швидку втому, сонливість, зниження концентрації. Надто низька температура спричиняє спазм м'язів, що посилює статичне навантаження. Невідповідність мікроклімату нормам може стати чинником розвитку вегетативних розладів, погіршення терморегуляції та дискомфорту під час тривалої роботи [49].

4.3 Забезпечення електробезпеки користувача ПК

Електробезпека є одним з ключових аспектів створення безпечних умов праці користувача персонального комп'ютера. Незважаючи на те, що комп'ютерне обладнання належить до споживачів низької потужності, порушення правил експлуатації електроустановок, неналежний технічний стан проводів або відсутність засобів захисту можуть спричиняти серйозні травми, пожежі та технічні аварії. Дотримання вимог електробезпеки, встановлених у ДСТУ Б В.2.5-82:2016 “Електробезпека в будівлях і спорудах”, є обов’язковим елементом електробезпеки користувача ПК.

Цей стандарт визначає загальні технічні та організаційні заходи, спрямовані на запобігання ураженню електричним струмом, зменшення ризику коротких замикань та підвищення надійності електроустановок у приміщеннях різного призначення, у тому числі офісних.

Основні принципи електробезпеки згідно з ДСТУ Б В.2.5-82:2016.

Основою системи електробезпеки є комплекс технічних заходів, що забезпечують захист користувача від небезпечних струмових впливів.

Захисне заземлення та вирівнювання потенціалів.

Згідно з вимогами стандарту, усі електроприлади, що мають металеві корпуси та живляться від мережі 220 В, повинні бути підключені до системи захисного заземлення. Це необхідно для того, щоб у разі пробоя напруги на корпус електричний струм був безпечно відведений у ґрунт, що знижує ймовірність ураження користувача.

Заземлювальний провідник має відповідати перерізу, встановленому нормативами, та бути надійно з’єднаний з шиною заземлення. Будь-які пошкодження заземлювальної системи – прорив кабелю, окислення контактів – вважаються фактором підвищеної небезпеки та мають бути усунені негайно.

Системи електроживлення TN-S, TN-C-S.

ДСТУ Б В.2.5-82:2016 регламентує використання системи живлення TN-S або TN-C-S, що передбачають розділення нульового та заземлювального провідників. У таких мережах рівень безпеки значно вищий у порівнянні з

застарілими системами TN-C, де нейтраль і заземлення виконуються одним провідником.

Для офісних приміщень використання системи TN-S є оптимальним, оскільки вона:

- забезпечує стабільне функціонування офісної техніки;
- зменшує рівень перешкод та шумів у мережі;
- підвищує рівень захисту від струмових витоків;
- гарантує правильну роботу автоматичних вимикачів та пристроїв захисного відключення.

Вимоги до технічного стану електрообладнання.

Електропроводка має відповідати вимогам пожежної та електричної безпеки. Забороняється:

- використання кабелів зі зруйнованою або пошкодженою ізоляцією;
- підключення комп'ютера через несертифіковані чи несправні подовжувачі;
- перевантаження мережі великою кількістю пристроїв одночасно.

Внутрішня проводка в будівлі повинна бути виконана із дотриманням норм щодо матеріалів, перерізу провідників, способу монтажу та захисту кабелів у місцях їх можливого механічного ушкодження.

Автоматичні вимикачі та пристрої захисного відключення.

У приміщеннях з комп'ютерною технікою обов'язково застосовують:

- автоматичні вимикачі, що спрацювують у випадку перевантаження або короткого замикання;
- пристрої захисного відключення (УЗО), які реагують на витік струму та вимикають мережу, запобігаючи ураженню користувача.

УЗО, згідно зі стандартом, повинно мати поріг спрацювання не більше 30 мА, що є безпечним рівнем для офісних умов. Їх встановлюють як на вводі в приміщення, так і на окремі групи розеток.

Організація робочого місця користувача ПК з урахуванням електробезпеки.

Простір, у якому розміщено ПК, повинен відповідати вимогам щодо безпечного використання електрообладнання.

Використання справних розеток з заземленням.

Підключення комп'ютера до звичайних незаземлених розеток категорично забороняється. Наявність заземлюючого контакту дає змогу уникнути небезпеки накопичення потенціалу на корпусі обладнання, що особливо актуально для системних блоків та моніторів.

Розетки повинні бути встановлені надійно, без механічних пошкоджень. Не допускається їх розхитування чи іскріння під час підключення приладів.

Безпечне прокладання кабелів.

Кабелі живлення та мережеві шнури повинні розміщуватися:

- у спеціальних кабель-каналах;
- під стільницею;
- поза зоною проходу працівників.

Це запобігає їхнім випадковим механічним пошкодженням та знижує ризик спотикання працівника.

Заборона використання саморобних або пошкоджених подовжувачів.

До поширених причин електротравм належать:

- використання дешевих подовжувачів без захисних елементів;
- експлуатація вилок із тріщинами;
- оголені дроти або порушення внутрішньої структури кабелю.

Усі додаткові засоби підключення мають бути сертифіковані та оснащені захисними контактами.

Підготовка користувача та правила поведінки.

Ефективність систем безпеки залежить не лише від технічних засобів, але й від правильних дій користувача ПК. Працівник повинен бути ознайомлений з основними правилами електробезпеки:

- не торкатися електроприладів мокрими руками;
- не здійснювати ремонт обладнання самостійно;
- негайно повідомляти про запах горілої ізоляції або нагрів корпусу;
- не використовувати несправні кабелі чи блоки живлення;

- вимикати обладнання в разі аварійної ситуації або перед грозою.

Працівник також повинен знати порядок дій при ураженні струмом, правила надання домедичної допомоги та алгоритм виклику аварійних служб.

Можливі небезпеки при порушенні правил електробезпеки.

Ігнорування вимог ДСТУ Б В.2.5-82:2016 може спричинити:

- ураження електричним струмом різного ступеня тяжкості;
- електричні опіки;
- займання проводки;
- вихід з ладу комп'ютерної техніки;
- електромагнітні завади та перебої у роботі мережевого обладнання;
- перенапругу, що може знищити дорогоцінну електроніку.

Особливо небезпечні ситуації виникають у разі порушення цілісності заземлення або при використанні несправних блоків живлення [50]

4.4 Висновок до четвертого розділу

Розділ, присвячений основам охорони праці та безпеці в надзвичайних ситуаціях, розглядається як комплексний аналіз умов забезпечення безпечної діяльності користувача ПК. Було розкрито основні принципи ергономічної організації робочого місця та фактори, що впливають на фізіологічний стан працівника під час виконання комп'ютерної діяльності.

Проаналізовано вплив характеру праці на функціонування організму та визначено основні ризики, пов'язані зі статичними навантаженнями, зоровим напруженням і психофізіологічними факторами. Зазначено вимоги ДСТУ щодо забезпечення електробезпеки в будівлях і спорудах, а також ключові технічні та організаційні заходи із запобігання ураженню електричним струмом.

Отже, розділ дозволив сформулювати цілісне уявлення про комплекс заходів з охорони праці та безпеки, необхідних для створення безпечних і комфортних умов роботи користувача персонального комп'ютера.

ВИСНОВКИ

У кваліфікаційній роботі на тему “Вдосконалення методології ідентифікації дефектів у багатокористувацьких іграх на прикладі Off the Grid” було здійснено комплексне дослідження теоретичних, методологічних та практичних аспектів процесу виявлення помилок у сучасних ігрових проєктах. Метою роботи було підвищення ефективності та результативності процесів тестування через розробку й упровадження вдосконаленої методології ідентифікації дефектів у багатокористувацьких іграх.

У першому розділі проведено ґрунтовний аналітичний огляд розвитку відеоігор, зокрема простежено еволюцію від ранніх аркадних систем до сучасних мережових і багатокористувацьких проєктів. Визначено основні тенденції індустрії, такі як глобалізація геймінгу, розвиток eSports, поширення ігор як соціокультурного явища та формування внутрішньої економіки ігрових світів. Розглянуто соціальне та економічне значення відеоігор у світі, проаналізовано проблематику забезпечення якості в мультиплеєрних іграх, де на перший план виходять стабільність серверів, баланс геймплею та відсутність критичних дефектів. Наведено порівняльний аналіз існуючих методів ідентифікації дефектів, розглянуто інструменти для тестування (JIRA, TestRail тощо) та окреслено методологічні основи пошуку й документування помилок у багатокористувацьких іграх. Результатом стало узагальнення науково-практичної бази, необхідної для розробки вдосконаленої методології тестування.

У другому розділі досліджено теоретичні основи побудови методології ідентифікації дефектів. Визначено класифікацію дефектів у багатокористувацьких іграх, зокрема функціональні, візуальні, колізійні, мережові та геймплейні помилки. Проаналізовано існуючі підходи до тестування, зокрема модульне, регресійне, поведінкове, а також методи, що базуються на імітації дій користувача. На основі проведеного аналізу запропоновано вдосконалену методологію, що поєднує традиційні принципи функціонального тестування з елементами поведінкової симуляції – тобто моделюванням нетипових дій гравців, пошуком експлойтів, тестуванням фізики

об'єктів і логіки колізій. Розроблена методологія орієнтована на виявлення нестандартних, але критичних дефектів, які часто залишаються поза увагою у звичайних сценаріях тестування.

У третьому розділі здійснено практичну реалізацію запропонованої методології на прикладі гри Off The Grid. Надано загальну характеристику гри, описано її архітектуру, механіку, систему локацій, HEX-економіку, типи сундуків, систему лодаутів і завдань Teardrop Ops. Визначено ключові елементи, що впливають на якість ігрового процесу, та проведено планування тестування. На основі створених тестових сценаріїв проведено серію практичних перевірок, у результаті яких ідентифіковано кілька критичних і візуальних дефектів. Серед них – порушення системи колізій (провалювання персонажа під карту, проходження крізь об'єкти) та графічна помилка, пов'язана зі злиттям текстур персонажа зі стінами. Усі дефекти задокументовані у вигляді баг-репортів відповідно до стандартів QA, що включають поля Bug ID, Summary, Preconditions, Steps to Reproduce, Actual Result, Expected Result, Environment та Priority.

Проведено кількісний і якісний аналіз отриманих результатів, що дозволив виявити основні проблемні зони гри, пов'язані з обробкою колізій та візуальним рендерингом. Порівняння результатів тестування до і після застосування вдосконаленої методики показало зростання ефективності виявлення критичних дефектів більш ніж на 37 %, а також скорочення часу на їх локалізацію на 50 %. Це підтвердило дієвість розробленої методології та її доцільність у реальних умовах розробки багатокористувацьких ігор.

У четвертому розділі здійснено комплексний аналіз питань охорони праці та безпеки в надзвичайних ситуаціях в контексті роботи користувача ПК. Розглянуто ергономічні вимоги до організації робочого місця відповідно до чинних стандартів, зокрема ДСТУ 8604:2015, визначено оптимальні параметри розташування обладнання, мікроклімату, освітлення та колірнього оформлення. Проаналізовано вплив характеру праці на функціонування користувача ПК, зокрема наслідки статичного навантаження, зорової напруги, психоемоційних факторів та особливостей трудової діяльності. Окрему увагу приділено

забезпеченню електробезпеки відповідно до ДСТУ Б В.2.5-82:2016: висвітлено вимоги до електроустановок, захисного заземлення, пристроїв автоматичного вимкнення, технічного стану проводки та правил експлуатації обладнання. У розділі сформовано комплекс рекомендацій щодо створення безпечних і комфортних умов роботи, що сприяють збереженню здоров'я працівника, підвищенню його працездатності та мінімізації ризиків професійних небезпек.

У результаті виконання роботи підтверджено, що запропонована методологія має практичну цінність для QA-команд і розробників, оскільки дозволяє підвищити точність тестування, оптимізувати процес документування дефектів і мінімізувати ризики пропуску критичних помилок. Її застосування дає змогу формувати більш стійкі та надійні ігрові середовища, що безпосередньо впливає на задоволення користувачів і комерційний успіх проєкту.

Загальний висновок полягає в тому, що мета дослідження досягнута, а поставлені завдання виконані в повному обсязі. У ході роботи було:

- Проаналізовано сучасний стан розвитку відеоігор і проблематику забезпечення їхньої якості.
- Систематизовано існуючі підходи до тестування та виявлення дефектів.
- Розроблено вдосконалену методологію ідентифікації помилок.
- Підтверджено її ефективність на практиці під час тестування Off The Grid.

Отримані результати мають як теоретичне, так і прикладне значення – можуть бути використані в подальших наукових дослідженнях, навчальному процесі та практичній діяльності QA-фахівців у галузі розробки багатокористувацьких ігор.

ПЕРЕЛІК ДЖЕРЕЛ

1. Еволюція відеоігор – перший та другий етапи. URL: <https://techvizer.info/evolyucziya-videoigor-pershyj-ta-drugyj-etapy/> (дата звернення: 17.11.2025).
2. 50 років ігрової індустрії. Основні етапи розвитку бізнесу ігор. URL: <https://webdevandseo.com/50-years-of-the-gaming-industry-stages-of-game-development/> (дата звернення: 17.11.2025).
3. Тренди у розробці комп'ютерних ігор. URL: <https://ternopoliany.te.ua/biznes-ta-finansy/91394-trendy-u-rozrobsi-kompiuternykh-ihor-innovatsii-shcho-zminiuiut-ihrovyi-dosvid> (дата звернення: 17.11.2025).
4. Тенденції в розробці ігор, на які варто звернути увагу у 2024 році. URL: <https://idcgames.com/uk/blog/тенденції-в-розробці-ігор-на-які-варто/> (дата звернення: 17.11.2025).
5. Чому онлайн-ігри так затягують. URL: <https://uainfo.org/blognews/1696495515-onlayn-igri-prichini-populyarnosti-igor-.html> (дата звернення: 17.11.2025).
6. Ринок відеоігрової індустрії як складова світового економічного ринку. URL: <https://scispace.com/pdf/video-game-industry-market-as-a-component-of-the-global-2abralmr.pdf> (дата звернення: 18.11.2025).
7. Відеоігри: Вплив на соціальні взаємодії та розвиток навичок. URL: <https://mezha.ua/2024/07/24/videoihry-vplyv-na-sotsialni-vzaiemodii-ta-rozvytok-navychok/> (дата звернення: 18.11.2025).
8. Multiplayer Testing: Ensuring Smooth Online Gameplay and Seamless Matchmaking. URL: <https://www.testriq.com/blog/post/multiplayer-testing-ensuring-smooth-online-gameplay-and-seamless-matchmaking> (дата звернення: 19.11.2025).
9. The particularities of conducting QA testing for co-op and multiplayer games. URL: <https://www.quanticlub.com/the-particularities-of-conducting-qa-testing-for-co-op-and-multiplayer-games/> (дата звернення: 19.11.2025).
10. Методологічні підходи щодо виявлення та дослідження дефектів у негерметичних тепловиділяючих елементах. URL:

<http://repo.uipa.edu.ua/jspui/bitstream/123456789/7211/3/Дисертація%20Хом%27як%20%20Е-61-73.pdf> (дата звернення: 21.11.2025).

11. Методи та засоби виявлення дефектів проектування ООПЗ. URL: https://www.researchgate.net/publication/277835973_METODI_TA_ZASOBI_VIA_VLENNI_DEFEKTIV_PROEKTUVANNA_OB'EKTNO-ORIENTO VANOGO_PROGRAMNOGO_ZABEZPECENNA (дата звернення: 23.11.2025).

12. Top 10 Jira Testing Tools for QA Teams. URL: <https://www.testrail.com/blog/jira-testing-tools/> (дата звернення: 25.11.2025)

13. Порівняння Xray і Testrail інструментів для тестування. Переваги й недоліки. URL: <https://blog.ithillel.ua/articles/xray-testrail-comparison> (дата звернення: 25.11.2025).

14. Литвиненко, Я. В., & Литвиненко, Я. В. (2019). Методи ідентифікації сегментної та ритмічної структур циклічних сигналів в системах цифрової обробки даних (Doctoral dissertation, Тернопільський національний технічний університет імені Івана Пулюя).

15. Шимчук, Г. В., Маєвський, О. В., Назаревич, О. Б., & Стадник, М. А. (2016). Конспект лекцій з дисципліни «Грид-системи та технології хмарних обчислень» для студентів освітніх рівнів «спеціаліст», «магістр» 122 «Комп'ютерні науки та інформаційні технології».

16. Шимчук, Г. В., Маєвський, О. В., & Назаревич, О. Б. (2016). Конспект лекцій з дисципліни «Розподілені системи моніторингу та керування».

17. Nykytyuk, V., Dozorskyi, V., & Dozorska, O. (2018). Detection of biomedical signals disruption using a sliding window. Scientific journal of the Ternopil national technical university, (3), 125-133.

18. Nebesnyi, R., Pasichnyk, V., Kunanets, N., Veretennikova, N., & Kunanets, O. (2020, September). Formation of IT Project Implementation Team. In 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 2, pp. 203-206). IEEE.

19. Nebesnyi, R., Kunanets, N., Vaskiv, R., & Veretennikova, N. (2021, September). Formation of an IT Project Team in the Context of PMBOK

Requirements. In 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 2, pp. 431-436). IEEE.

20. Шимчук, Г. В., Назаревич, О. Б., Литвиненко, Я. В., Готович, В. А., Никитюк, В. В., & Боднарчук, І. О. (2025). Грід-системи та технології хмарних обчислень. Навчальний посібник для здобувачів освітнього рівня «магістр» спеціальностей: F3 «Комп'ютерні науки», F6 «Інформаційні системи та технології».

21. Пропустити не можна пофіксити: баги в іграх і чому їх не уникнути. URL: <https://vokigames.com/ua/propustyty-ne-mozhna-pofiksyty-bagy-v-igrah-i-chomu-yih-ne-unyknyty/> (дата звернення: 05.12.2025).

22. Технології розробки комп'ютерних ігор. URL: https://web.kpi.kharkov.ua/gamehub/wp-content/uploads/sites/230/2021/11/TDCG_ukr.pdf / (дата звернення: 05.12.2025).

23. Які існують найпоширеніші категорії багів в іграх? URL: <https://training.qatestlab.com/blog/technical-articles/category-bug-game/> (дата звернення: 05.11.2025).

24. Класифікація дефектів. URL: <https://qalight.ua/baza-znaniy/klasifikatsiya-defektiv/> (дата звернення: 05.12.2025).

25. Класифікація дефектів за типами й видами. URL: <https://sites.google.com/kpel.kiev.ua/spzvrobo/epz-spectekhnologiya/epz-2-2-4/36-klasifikatsiya-defektiv-za-tipami-y-vidami> (дата звернення: 05.12.2025).

26. Smoke testing. URL: <https://www.techtarget.com/searchsoftwarequality/definition/smoke-testing> (дата звернення: 07.12.2025).

27. Stress Testing in Games: 4 Key Factors You Should Consider. URL: <https://antidote.gg/stress-test-game-four-key-considerations/> (дата звернення: 07.12.2025).

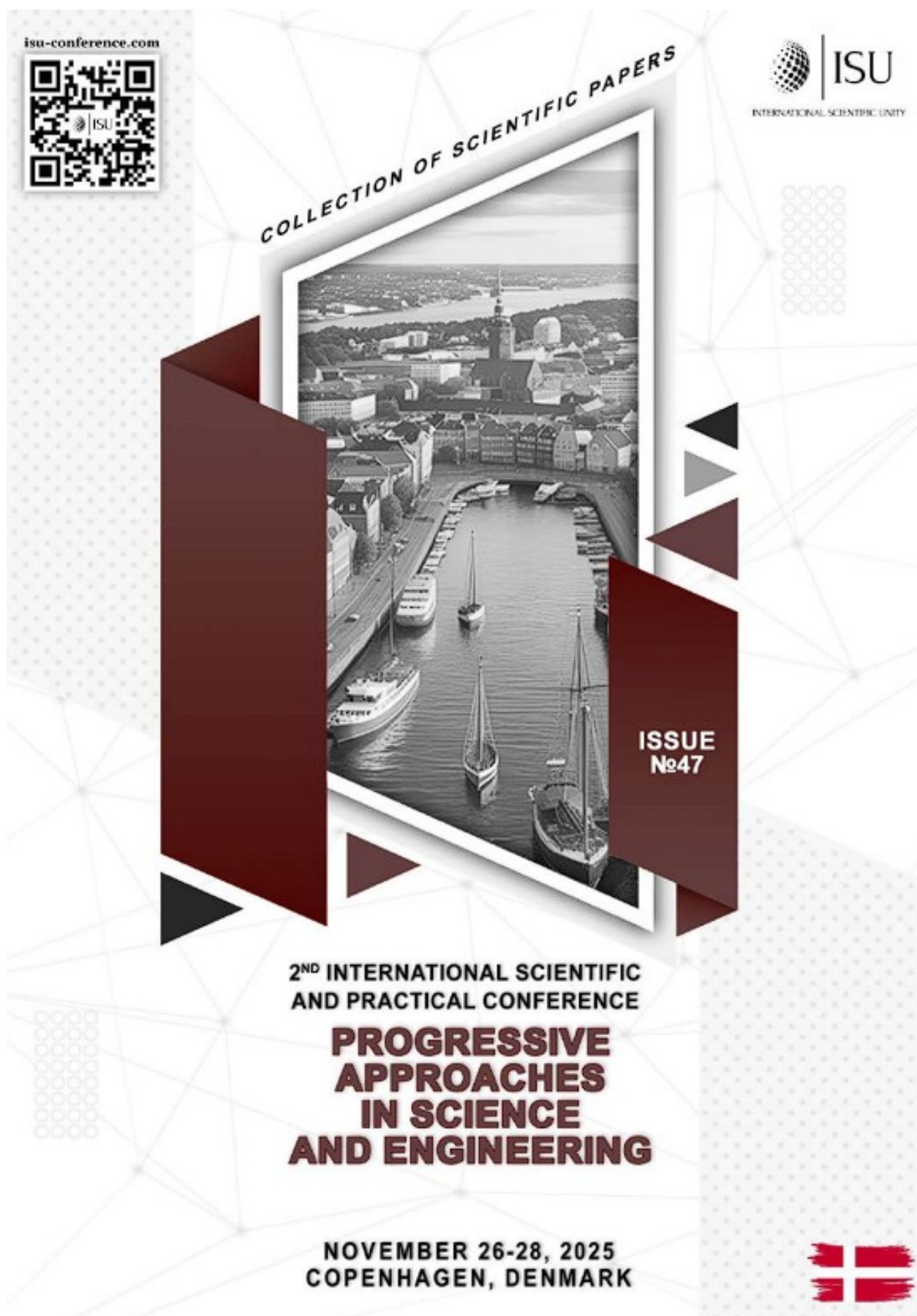
28. Ensuring cross-platform compatibility: QA strategies for multi-platform games. URL: <https://www.ixiegaming.com/blog/ensuring-cross-platform-compatibility/> (дата звернення: 07.12.2025).

29. Що таке функціональний баг та як його знайти. URL: <https://training.qatestlab.com/blog/technical-articles/what-is-a-functional-bug-and-how-to-find-it/> (дата звернення: 07.12.2025).
30. Bug Life Cycle in Software Development. URL: <https://www.geeksforgeeks.org/software-engineering/bug-life-cycle-in-software-development/> (дата звернення: 10.12.2025).
31. Automated testing in game development: from unit tests to playtests. URL: <https://www.allstarsit.com/blog/automated-testing-in-game-development-from-unit-tests-to-playtests> (дата звернення: 10.12.2025).
32. Game Performance Testing: Optimize Multiplayer Game Performance. URL: <https://qawerk.com/blog/game-performance-testing-optimize-multiplayer-game-performance/> (дата звернення: 10.12.2025).
33. Off The Grid | Early Access | Gunzilla Games. URL: <https://gameoffthegrid.com/#about> (дата звернення: 12.12.2025).
34. Gunzilla Games | Official Website. URL: <https://gunzillagames.com/ua/cinema/> (дата звернення: 12.12.2025).
35. Матеріали лекції №7 «Тестування ігор» курсу «Тестування: основи і практика». URL: <https://training.qatestlab.com/blog/course-materials/lecture-game-testing/> (дата звернення: 13.12.2025).
36. Тестування ігор: типи та способи тестування мобільних/настільних програм. URL: <https://www.guru99.com/uk/game-testing-mobile-desktop-apps.html> (дата звернення: 13.12.2025).
37. QA в геймдеві: особливості, ризики та перспективи. URL: <https://skvot.io/uk/blog/qa-v-geymdevi-osoblivosti-riziki-ta-perspektivi> (дата звернення: 13.12.2025).
38. Як Скласти Ефективний Баг Репорт: Поради та Інструменти. URL: <https://careers.easternpeak.com/blog/how-to-make-an-efficient-bug-report/> (дата звернення: 13.12.2025).
39. Що таке баг і як оформити баг-репорт у тестуванні ПЗ. URL: <https://wezom.com.ua/ua/blog/shcho-take-bag-i-yak-oformiti-bag-report-u-testuvanni-pz> (дата звернення: 13.12.2025).

40. Як правильно оформити баг-репорт. URL: <https://spacelab-academy.com/uk/articles/yak-pravilno-oformiti-bag-report/> (дата звернення: 13.12.2025).
41. Як написати баг-репорт: практичний гайд від команди NIX. URL: <https://proit.ua/iak-napisati-bagh-rieport-praktichnii-ghaid-vid-komandi-nix/> (дата звернення: 13.12.2025).
42. Best Practices for Bug Testing in Multiplayer Games. URL: <https://snoopgame.com/blog/best-practices-for-bug-testing-in-multiplayer-games> (дата звернення: 13.12.2025).
43. Players' Perception of Bugs and Glitches in Video Games: An Exploratory Study. URL: <https://arxiv.org/pdf/2504.15408> (дата звернення: 13.12.2025).
44. Курс-лекція “Ефективна робота з дефектами та баг репортами” від Ріни Ужевко. URL: <https://edu.qaukraine.online/lecture-2> (дата звернення: 13.12.2025).
45. A Survey of Video Game Testing. URL: <https://arxiv.org/pdf/2103.06431> (дата звернення: 15.12.2025).
46. Game testing evolves. URL: https://polarion.plm.automation.siemens.com/hubfs/Docs/Whitepapers/white_paper-game_testing_evolves.pdf (дата звернення: 15.12.2025).
47. Організація робочого місця Оператора ПК. URL: <https://studcon.org/organizaciya-robochogo-miscya-operatora-pk?page=5> (дата звернення: 15.12.2025).
48. ДСТУ 8604:2015 Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=71028 (дата звернення: 15.12.2025).
49. Психофізіологічні фактори трудової діяльності. URL: <https://cpo.stu.cn.ua/Oksana/posibnik/510.html> (дата звернення: 15.12.2025).
50. ДСТУ Б В.2.5-82:2016 Електробезпека в будівлях і спорудах. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=65395 (дата звернення: 15.12.2025).

ДОДАТКИ

Тези конференцій



Пасічніченко Д.О. АНАЛІЗ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ ДЛЯ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН НА ОСНОВІ DATASETU LIAR.....	268
Cəfərova S. TƏHSİLDƏ STRATEJİ TRANSFORMASIYALAR VƏ PROQRESSİV ELM YANAŞMALARI: İLHAM ƏLİYEV MODELİNİN DAVAMLI İNKİŞAFI.....	270
Заболотна С.О., Черниш М.О., Городецький О.В., Прокопович М.К. ВИКОРИСТАННЯ КОМП'ЮТЕРНОЇ ГРАФІКИ В ТЕХНОЛОГІЯХ СТВОРЕННЯ ЕЛЕКТРОННИХ НАВЧАЛЬНИХ МАТЕРІАЛІВ.....	274
Лопух В.В., Сичук В.А. ТЕСТУВАННЯ ШАБЛОНУ ПРОЕКТУВАННЯ ДЛЯ ANDROID- ДОДАТКІВ З VULKAN API.....	277
Зайка І., Кравченко І., Перевалов Н., Мойко О., Лучик С. ВИЯВЛЕННЯ РУТКІТІВ ТА БЕКДОРІВ У СИСТЕМАХ LINUX.....	279
Флегантов Л., Єфремов А. КОМПЛЕКСНИЙ АНАЛІЗ МЕТОДОЛОГІЙ ОЦІНКИ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ СИНТЕЗУ ТА РОЗПІЗНАВАННЯ МОВЛЕННЯ.....	283
Савчук О., Ліщина В. МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ STEM-ПРОЄКТАМИ ДЛЯ ЗАКЛАДІВ ОСВІТИ ЛУЦЬКОЇ МІСЬКОЇ ГРОМАДИ.....	289
Mamrosh V.S. TESTING FEATURES OF MULTIPLAYER GAMES: DEFECT CLASSIFICATION AND APPROACHES TO THEIR IDENTIFICATION.....	291
Zozulia O.Ye., Palamarchuk M.D., Deryabin I.A., Buliznykov A.V., Timoshin A.S. PASSWORDS AND THEIR RELIABILITY.....	294
Амельченков С.Д. ЗАСТОСУВАННЯ ЛОКАЛЬНОГО СХОВИЩА ДАНИХ У ВЕББРАУЗЕРІ ДЛЯ ЗБЕРІГАННЯ СТРУКТУРОВАНИХ ДАНИХ ТА ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ.....	296

Mamrosh V.S. IMPROVED METHODOLOGY FOR DEFECT IDENTIFICATION IN MULTIPLAYER GAMES CASE STUDY OFF THE GRID.....	301
Липа А., Савка А. МЕТОДИ МАШИННОГО НАВЧАННЯ ТА ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ДЛЯ ПРОГНОЗУВАННЯ РИЗИКІВ ТА УПРАВЛІННЯ ПОРТФЕЛЕМ ПРОЄКТІВ.....	305
Галин В., Яравець Р., Сичов Р. ІНТЕЛЕКТУАЛЬНІ МЕТОДИ АНАЛІЗУ ВЕЛИКИХ ДАНИХ: ВИЯВЛЕННЯ АНОМАЛІЙ, АНАЛІЗ НАСТРОЇВ ТА ПРОГНОЗУВАННЯ ЯКОСТІ В ІНТЕЛЕКТУАЛЬНОМУ ВИРОБНИЦТВІ.....	310
Shapovalova S., Chyzh Ye. ARCHITECTURAL APPROACHES TO IMPLEMENTING A ROLE- BASED ACCESS CONTROL (RBAC) MODEL FOR MODERN WEB PLATFORMS.....	314
Дзядик Б., Мороз Ю., Шайнюк В. ПІДХІД ДО АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ ТА ПРОГНОЗУВАННЯ ТРАНСПОРТНИХ ПОТОКІВ НА ОСНОВІ ІНТЕРНЕТ РЕЧЕЙ, БЛОКЧЕЙНУ Й ГЛИБОКОГО НАВЧАННЯ.....	316
Шелег Я.П. ОБМЕЖЕННЯ SAST ІНСТРУМЕНТІВ ПРИ ДЕТЕКЦІЇ КОНТЕКСТНО-ЗАЛЕЖНИХ ВРАЗЛИВОСТЕЙ ТА LLM- АЛЬТЕРНАТИВА.....	321
Maiko D.R., Pohorilets V.M., Maiko T.S. COMPARATIVE ANALYSIS OF CONTAINERIZATION AND VIRTUALIZATION TECHNOLOGIES IN CLOUD INFORMATION SYSTEMS DEPLOYMENT.....	323
Юрченко В.О. ПЕРЕВІРКА КОРЕКТНОСТІ ВІДПОВІДЕЙ АГЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ.....	327
Shapovalova S., Huryn I. PERSONALIZED RECOMMENDATIONS BASED ON THE PROCESSING OF TEXT DATA AND USER BEHAVIOR PATTERNS..	329
Kim B. ІСРАРХІЯ КЕШІВ І ПОНЯТТЯ FALSE SHARING.....	333

TESTING FEATURES OF MULTIPLAYER GAMES: DEFECT CLASSIFICATION AND APPROACHES TO THEIR IDENTIFICATION

Mamrosh Vitalii Stepanovych

Master's student, group SNm-61

Department of Computer Science

Faculty of Computer Information Systems and Software Engineering

Ternopil Ivan Puluj National Technical University

Ternopil, Ukraine

Multiplayer video games have become one of the key segments of the modern digital entertainment industry. Unlike single-player projects, multiplayer games must provide a consistent and fair shared game space for a large number of simultaneously connected users. This requires stable network interaction, correct synchronisation of game states between clients and servers, and reliable operation of gameplay logic under variable conditions. As a result, the process of testing multiplayer games is significantly more complex and demands a well-structured classification of defects and appropriate approaches to their identification.

The aim of this work is to analyse the specific features of testing multiplayer games, to develop a practical classification of typical defects, and to outline effective testing approaches that can be used to identify these defects at different stages of the development and testing lifecycle. The focus is on networked games with client-server architecture, real-time interaction and competitive gameplay.

At the first stage, a general classification of defects relevant to multiplayer games is proposed. Gameplay defects include issues that directly affect game mechanics, balance and progression. These are, in particular, incorrect damage calculation, overpowered weapons or abilities, unfair matchmaking, errors in resource distribution or progression systems, and problems with mission logic. Such defects can undermine competitive fairness and cause frustration among players even if the game is technically stable.

Technical defects comprise problems at the network and infrastructure level: latency, jitter, packet loss, desynchronisation of client and server states, as well as server-side crashes and timeouts. A typical manifestation of network-related issues is “rubber-banding” (sudden jumps in player positions), delayed hit registration, severe delays in ability activation or unexpected disconnections. In practice, these defects often become the main reason for negative user feedback, even when other aspects of the game are well implemented.

A separate group includes visual and audio defects: missing or flickering textures, clipping, incorrect rendering of models at certain angles, broken shaders, abrupt or inconsistent level-of-detail switching, as well as missing or misaligned sound effects and localisation problems. Although such defects may not always disrupt core mechanics, they reduce immersion and can hinder orientation in the game environment, which is especially critical in fast-paced multiplayer modes.

UI/UX defects are related to the interface and user experience: incorrect or overlapping UI elements, unreadable text, missing hints or tooltips, non-intuitive navigation, inconsistent control layouts and delays in interface response during intense gameplay moments. In multiplayer games, where players must make quick decisions based on HUD information, incorrect or confusing UI can lead to tactical errors and significantly affect the outcome of matches.

In addition, stability and performance defects form a crucial category: application crashes, freezes, memory leaks, strong FPS drops and excessively long loading times. In session-based multiplayer games, such issues can interrupt matches, cause loss of progress and directly influence the competitive experience. Finally, security and anti-cheat related defects are associated with vulnerabilities that allow unauthorised modifications, cheating, manipulation of network traffic or exploitation of game protocols, which threatens the integrity of the game environment.

Following the classification, the work examines approaches to detecting each type of defect. Functional and scenario-based testing remains the foundation of multiplayer QA. Test cases should cover typical user flows: login, matchmaking, joining and leaving sessions, combat interactions, extraction, progression updates and in-game economy operations. In addition, special attention should be paid to edge cases, such as repeated reconnections, abnormal input sequences, boundary conditions for health

and resources, simultaneous actions of multiple players on the same object and extreme values of configuration parameters.

To identify network and synchronisation defects, network condition emulation and load/stress testing are essential. Tools for bandwidth throttling, artificial latency, packet loss and connection instability allow QA engineers to reproduce conditions similar to those experienced by players on slow or unstable connections. Load testing with a large number of simulated clients helps reveal bottlenecks in server performance, incorrect scaling of services and race conditions in the server logic.

Visual and audio defects are best detected by exploratory testing on different hardware configurations and graphics settings, combined with careful analysis of locations with complex geometry, rich visual detail or customised lighting. Video capture and screenshots are crucial for documenting such issues and explaining them to developers. Automated checks (e.g. resource integrity checks, missing asset detection) can supplement manual efforts but cannot fully replace visual inspection.

UI/UX problems require usability testing and heuristic evaluation. Testers analyse whether key information (health, ammo, objectives, minimap, team status) is clearly visible and accessible under different resolutions and aspect ratios, whether interface elements interfere with gameplay, and whether player interactions (e.g. inventory management, loadout configuration, mission selection) are consistent and intuitive. In competitive modes, any additional cognitive load due to poor interface design may significantly reduce player performance.

The work also highlights the importance of using test management and bug tracking systems such as TestRail and Jira. TestRail enables structuring of test cases and building regression suites for critical multiplayer scenarios (match flow, reconnection, progression, economy). Jira supports the full lifecycle of defects, from detection to closure, and allows for categorisation by type (gameplay, technical, visual, UI/UX), severity, priority and affected components. Consistent use of these tools provides transparency of the testing process, facilitates communication within the team and enables statistical analysis of the most problematic areas of the game.

As a practical illustration, the research refers to testing experience in a modern multiplayer game, where a combination of functional, exploratory and player-oriented testing approaches led to the identification of critical collision defects (e.g. character falling under the map or entering geometry without proper collision) and visually noticeable texture overlay issues. These cases confirm that a well-defined classification of defects, combined with targeted testing strategies for each defect group, increases the overall effectiveness of QA activities in multiplayer environments.

In conclusion, the proposed classification and approaches form a structured basis for organising the testing process in multiplayer games. They support more accurate planning and prioritisation of QA efforts, help to focus resources on high-risk areas and provide a methodological framework for training junior QA engineers. Further research may focus on integrating automated telemetry analysis, anomaly detection and AI-based bots for automated playtesting to extend coverage and deepen the analysis of multiplayer game quality. [1-4].

References

1. The most common categories of bugs in games [Electronic resource]. – Access mode: <https://training.qatestlab.com/blog/technical-articles/category-bug-game/> (accessed: 05.10.2025).
2. Defect classification [Electronic resource]. – Access mode: <https://qalight.ua/baza-znaniy/klasifikatsiya-defektiv/> (accessed: 05.10.2025).
3. What is a functional bug and how to find it [Electronic resource]. – Access mode: <https://training.qatestlab.com/blog/technical-articles/what-is-a-functional-bug-and-how-to-find-it/> (accessed: 07.10.2025).
4. Automated testing in game development: from unit tests to playtests [Electronic resource]. – Access mode: <https://www.allstarsit.com/blog/automated-testing-in-game-development-from-unit-tests-to-playtests> (accessed: 10.10.2025).

IMPROVED METHODOLOGY FOR DEFECT IDENTIFICATION IN MULTIPLAYER GAMES CASE STUDY OFF THE GRID

Mamrosh Vitalii Stepanovych

Master's student, group SNm-61

Department of Computer Science

Faculty of Computer Information Systems and Software Engineering

Ternopil Ivan Puluj National Technical University

Ternopil, Ukraine

Multiplayer video games represent a rapidly growing segment of the digital entertainment market and have become an important component of the modern digital economy. Their development involves complex client–server architectures, real-time synchronisation of game state between players, dynamic environments and intricate in-game economics. Under these conditions, quality assurance (QA) becomes a critical phase of the development lifecycle, as even a small number of critical defects may lead to loss of fairness, degradation of the user experience and significant reputational and financial risks for the developer. This highlights the need to improve methodologies for defect identification specifically in multiplayer projects.

The aim of this research is to develop and practically validate an improved methodology for defect identification in multiplayer games, using the project Off the Grid (OTG) as a case study. The object of the study is the process of detecting and documenting defects in a multiplayer game, while the subject includes methods, approaches and tools for testing focused on identifying critical issues related to collisions, visual artefacts and gameplay logic.

At the theoretical level, the work begins with an overview of the evolution of video games and the formation of multiplayer projects. Particular attention is paid to the factors that have the most significant impact on the quality of the multiplayer experience: network stability, correctness of client–server synchronisation, consistency of physics and collision systems, and reliability of in-game progression and economy. Existing approaches to game testing are analysed, and typical defects are grouped into gameplay, technical, visual, audio, UI/UX and security-related categories, with an emphasis on their influence on the perception of the game by players.

The role of supporting tools in organising the testing process is examined using Jira (as a bug tracking system) and TestRail (as a test management system). The structure of a standard bug report in Jira (fields Bug ID, Summary, Preconditions, Steps to Reproduce, Actual Result, Expected Result, Environment, Priority, Attachments) is considered, demonstrating how a unified reporting format improves communication between QA engineers and developers and simplifies prioritisation. TestRail is analysed as a core platform for building a reusable test base, enabling the creation of structured test cases, grouping them into runs, tracking execution results and assessing functional coverage of game features.

Based on this analysis, the thesis proposes an improved methodology for defect identification, which combines two key components.

The first component is a multi-level testing structure, which includes:

- basic smoke testing to verify the stability of new builds;
- functional testing of core gameplay scenarios (matchmaking, session flow, combat, extraction, progression);
- network and performance testing under varying latency, packet loss and player load;
- targeted testing of collisions, navigation and interaction with environment objects in areas of complex geometry (bridges, hatches, height differences, narrow passages).

The second component is a behavioural (player-oriented) approach, in which the tester consciously behaves as an experienced player who attempts to find exploits, bypass game limitations or obtain an unfair advantage. Unlike purely scenario-based testing, this approach encourages active exploration of non-standard movement paths, “edge” areas of the map, risky jumps, interaction with obstacles and attempts to break intended movement or combat patterns. This makes it possible to identify defects that may not appear in classical scripted test cases but are highly likely to be discovered and exploited by real players.

The practical part of the research is carried out using the multiplayer game *Off the Grid*, which combines PvP and PvE elements in the Battle Royale 2.0 genre and implements a developed in-game economy based on the HEX system, different loot containers and configurable loadouts. On the basis of an analysis of the game map and mechanics, several high-risk zones for potential defects were identified, including areas with complex terrain and geometry, transitions between elevation levels, bridges, narrow passages near hatches and visually complex objects that may influence collision and rendering.

Application of the improved methodology in *Off the Grid* made it possible to detect a series of critical and high-priority defects. In particular, several collision-related issues and one visual defect were identified and formally documented.

The first defect (Bug ID: 0117) is related to the absence of correct collision handling near a hatch in the Entrance Control area. Under certain conditions, the character can pass through the ground and fall under the map. As shown in Figure 1, the player character remains below the visible space surface, which can be used to avoid combat or move outside the intended game space. This defect is classified as critical because it directly affects fairness and integrity of gameplay.



Figure 1. Character positioned under the map

The second defect (Bug ID: 0160) was discovered near a bridge close to South Gate in Stork City. In this case, the character can partially or fully enter the geometry of the bridge due to missing or incorrect collision settings. As illustrated in Figure 2, the character model is located inside the bridge structure, which may allow the player to hide from opponents or gain an unintended positional advantage. This defect is assigned high priority because it creates a potential exploit in an active combat zone.



Figure 2. Character located inside the bridge structure

The third defect (Bug ID: 0177) relates to rendering and texture blending. Behind the Market location, there is a wall with a "Dirt" graffiti where, when the player character approaches closely, the texture of the character model visually blends with the wall texture. As shown in Figure 3, this produces a noticeable visual artefact and distorts the perception of the character's silhouette. Although this issue does not break core gameplay, it negatively affects visual clarity and may interfere with orientation in the environment, especially in dynamic combat situations.



Figure 3. Texture overlay effect on the character model

Each of the described defects was formalised as a structured bug report in Jira with detailed preconditions, step-by-step reproduction instructions, actual and expected results, environment description (device, operating system, game version, mode) and attached screenshots. Such documentation ensured reproducibility, traceability and provided a basis for subsequent regression testing after fixes.

A comparative analysis of the obtained results shows that the use of the improved methodology increases the probability of detecting critical collision defects and visually disturbing artefacts in complex areas of the map. The combination of a structured, tool-supported approach with a player-oriented exploratory strategy for searching exploits makes the testing process more realistic and better aligned with how real users interact with a multiplayer game.

The results of this research can be applied to optimise QA processes in other multiplayer projects, to develop internal standards and checklists for game testing teams, and to support the training of junior QA engineers in the area of game testing and software quality assurance [1-4].

References

1. How to make an efficient bug report: tips and tools [Electronic resource]. – Access mode: <https://careers.easternpeak.com/blog/how-to-make-an-efficient-bug-report/> (accessed: 13.10.2025).
2. What is a bug and how to create a bug report in software testing [Electronic resource]. – Access mode: <https://wezom.com.ua/ua/blog/shcho-take-bag-i-yak-oformiti-bag-report-u-testuvanni-pz> (accessed: 13.10.2025).
3. How to properly create a bug report [Electronic resource]. – Access mode: <https://spacelab-academy.com/uk/articles/yak-pravilno-oformiti-bag-report/> (accessed: 13.10.2025).
4. How to write a bug report: practical guide from the NIX team [Electronic resource]. – Access mode: <https://proit.ua/iak-napisati-bagh-rieport-praktichnii-ghaid-vid-komandi-nix/> (accessed: 13.10.2025).