

[DOI 10.28925/2663-4023.2025.29.832](https://doi.org/10.28925/2663-4023.2025.29.832)

УДК 004.89

Трофименко Олена Григорівна

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій,
Національний університет "Одеська юридична академія", Одеса, Україна

ORCID ID: [0000-0001-7626-0886](https://orcid.org/0000-0001-7626-0886)trofymenko@onua.edu.ua**Дика Анастасія Іванівна**

асистент кафедри інформаційних технологій

Національний університет «Одеська юридична академія», Одеса, Україна

ORCID ID: [0000-0002-4196-8734](https://orcid.org/0000-0002-4196-8734)dyka.anastasiia@gmail.com**Задерейко Олександр Владиславович**

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій

Національний університет «Одеська юридична академія», Одеса, Україна

ORCID ID: [0000-0003-0497-9861](https://orcid.org/0000-0003-0497-9861)zadereyko@onua.edu.ua**Шевченко Олександр В'ячеславович**

здобувач вищої освіти

Національний університет «Одеська юридична академія», Одеса, Україна

ORCID ID: 0009-0002-1873-1963

draqnslsnatsu@gmail.com

СФЕРИ ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В РОЗРОБЦІ ТА ТЕСТУВАННІ ІГОР

Анотація. Інтеграція штучного інтелекту (ШІ) у процес розробки програмного забезпечення стає справжньою революцією в індустрії комп'ютерних ігор. У статті досліджено та систематизовано можливі сфери застосування ШІ в розробці відеоігор та ігровому процесі. Загалом описано вісім сфер можливого впровадження ШІ в GameDev, а саме: створення графічного інтерфейсу; створення 3D-об'єктів у відеоіграх; написання діалогів та текстового наповнення у відеоіграх; створення музики; озвучення персонажів; створення анімації; написання та перевірка програмного коду; тестування відеоігор. Різні сфери можливого застосування ШІ в розробці відеоігор були експериментально апробовані та підтверджені під час розробки авторами гри у жанрі пригод з елементами жахів. Автори деталізували прикладні аспекти використання різних сучасних інструментів ШІ на прикладі розробки відповідних складових ігрового 3D-середовища гри «Labyrinth of Darkness». Використання ШІ допомагає оптимізувати трудомісткі процеси, як-от створення контенту та тестування, що сприяє пришвидшенню виробничих циклів та підвищенню якості кінцевого продукту. Для досягнення бажаного ігрового стилю інструменти ШІ дозволяють згенерувати декілька варіантів зображень елементів ігрового середовища, щоб вибрати найбільш відповідні з них для подальшої адаптації у графічний інтерфейс гри. Для цього алгоритми глибокого навчання аналізують тисячі зразків текстур, створюючи нові варіації, які виглядають природно та відповідають сучасним вимогам графіки. Тим самим активна інтеграція технології ШІ в сучасний процес розробки відеоігор значно змінює підходи до тестування та вдосконалення ігрових продуктів. З іншого боку, в роботі розглянуто складнощі та недоліки використання ШІ в розробленні та тестуванні ігор. Наразі генеративний ШІ у розробці та тестуванні ігор потребує ретельного контролю, адаптації та тестування. Найкращим рішенням є поєднання можливостей ШІ та творчого підходу розробників, що гарантує якісний контент й унікальний ігровий досвід.

Ключові слова: штучний інтелект; ШІ; алгоритми ШІ; GameDev; розробка ігор; тестування.



ВСТУП

Постановка проблеми. Ігрова індустрія сьогодні посідає провідне місце серед розважальних секторів, генеруючи сотні мільярдів доларів доходу щорічно. За оцінками експертів світовий ринок відеоігор до 2033 року зросте вдвічі й становитиме 503,14 млрд доларів США [1]. Зростання зумовлене не лише випуском проєктів із великими бюджетами та появою консолей нового покоління, а й впровадженням штучного інтелекту (ШІ) у розробку відеоігор, що дозволяє індустрії залишатись конкурентоспроможною навіть за умов насиченого ринку.

Аналіз останніх досліджень і публікацій. Стаття [2] досліджує переваги та обмеження технології ШІ, подолання технічних проблем та етичні проблеми, які можуть виникнути під час впровадження ШІ в розробку ігор. У дослідженні [3] систематизовано алгоритми ШІ щодо їх можливого застосування у розробці відеоігор та ігровому процесі. У роботі [4] проаналізовано вплив використання ШІ на покращення реакції в грі неігрових персонажів (non-playable character, NPC). У статті [5] розглядається різні ролі великих мовних моделей (Large Language Model, LLM) для ігор. Автори статті [6] дослідили використання LLM для генерації рівнів гри Sokoban. Вони дійшли висновку, що продуктивність LLM залежить від розміру набору даних. У роботі [7] досліджується, як LLM та GPT можуть виступати ефективними, високорівневими творчими партнерами у геймдизайні. Автори [8] дослідили використання LLM для створення прототипів карткових ігор і розробили систему на основі LLM для узгодженої генерації ігрового коду задля прискорення прототипування карткових ігор та полегшення праці розробників ігор. Дослідження [9] розглядає потенціал застосування GPT для ігор. Для цього його автори оглянули 131 статтю, 76 з яких були опубліковані 2024 року, і визначили п'ять можливих сфер застосування GPT у розробці ігор: процедурна генерація контенту, дизайн ігор зі змішаною ініціативою, ігровий процес зі змішаною ініціативою, гра в ігри та дослідження поведінки гравців. Автори роботи [10] створили інструмент генеративного штучного інтелекту (GenAI), який дозволяє дизайнерам створювати прототипи ігрових персонажів. У статті [11] розроблено систему розпізнавання емоцій на основі нечіткої логіки за допомогою штучних нейронних мереж, яка може використовуватись для динамічної процедурної генерації контенту з ігровою механікою взаємодії. В роботі [12] досліджені можливості та проблеми впровадження ШІ для дизайну та розробки ігор.

Численні публікації, спрямовані на висвітлення проблемних аспектів та можливостей використання ШІ для розробки ігор, свідчить про високий рівень зацікавленості та актуальність дослідження можливих сфер застосування ШІ в індустрії GameDev. При цьому у всіх досліджених публікаціях не приділяється увага прикладним аспектам можливим сферам використання ШІ в GameDev.

Мета статті: дослідити та систематизувати можливі прикладні сфери застосування ШІ у розробці відеоігор та ігровому процесі.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Створення комп'ютерної гри — це багатогранний процес за участі фахівців різних напрямів, кожен із яких відповідає за певні складові проєкту. Геймдизайнери та сценаристи формують концепцію, розробляють механіку та створюють сюжет, забезпечуючи інтеграцію ігрових елементів з історією. 3D-художники та модельєри

готують детальні моделі персонажів, об'єктів і локацій, які визначають візуальний стиль продукту, тоді як аніматори відповідають за надання руху цим моделям, забезпечуючи природну плавність та реалістичність анімацій. Програмісти та інженери пишуть програмний код, що реалізує ігрові механіки, фізичні взаємодії й алгоритми штучного інтелекту, необхідні для створення інтерактивного середовища. Саунддизайнери та композитори створюють музичний супровід і звукові ефекти, що занурюють гравця у світ гри, а актори озвучування, завдяки своїм голосам, надають персонажам емоційної виразності. Також важливу роль відіграють тестувальники, які ретельно виконують перевірку продукту на наявність помилок, контролюють баланс ігрового процесу та забезпечують відповідну якість готової версії. Попри все це, стрімкий розвиток ШІ відкриває нові можливості впровадження в кожен етап розробки гри та руйнує традиційні методи її створення, автоматизуючи процеси та змінюючи роль розробників. За результатами опитування, проведеного на початку 2025 року, 45% розробників ігор використовували генеративні ШІ-інструменти для розробки ігрового процесу [13]. Найпоширенішими сферами застосування інструментів ШІ в розробці ігор є написання або вдосконалення коду, модерація текстових та голосових чатів у грі, покращення анімації персонажів та тестування ігор. Сторонні постачальники ігрових сервісів також використовують інструменти ШІ для підтримки гравців та автоматичного перекладу на інші мови. Використання ШІ допомагає оптимізувати трудомісткі процеси, як-от створення контенту та тестування, що сприяє пришвидшенню виробничих циклів та підвищенню якості кінцевого програмного продукту. Загалом на різних етапах розробки комп'ютерних ігор інструменти ШІ можуть використовуватись у різних сферах (рис. 1).



Рис. 1. Сфери застосування ШІ в GameDev

Різні сфери можливого застосування ШІ в розробці відеоігор були експериментально апробовані та підтверджені під час розробки авторами гри «Labyrinth of Darkness», яка є грою у жанрі пригод з елементами жахів (horror). Вона занурює гравця у темний лабіринт, сповнений небезпек, ворогів та пасток. Гравець керує солдатом, озброєним гвинтівкою, а його завдання полягає у проходженні низки рівнів, виконуючи різноманітні завдання, борючись із ворогами та долаючи численні небезпеки, що підстерігають на кожному кроці. Однією з особливостей гри є вбудований конструктор рівнів, який надає можливість не лише проходити попередньо створені рівні, а й самостійно проєктувати власні.

1 Використання ШІ для створення графічного інтерфейсу

Графічний інтерфейс є обличчям будь-якого програмного продукту, а в комп'ютерних іграх — це перший контакт користувача з творчою ідеєю розробників. Сучасні інструменти на базі штучного інтелекту, як-от Leonardo.Ai (<https://leonardo.ai/>), кардинально змінюють підхід до розробки UI/UX, даючи можливість оперативно генерувати візуальні компоненти без залучення великої команди дизайнерів. Це особливо актуально для невеликих команд, які прагнуть оптимізувати ресурси та зосередитися на функціональності. Платформа Leonardo.Ai пропонує широкий спектр інструментів для створення узгоджених ресурсів стилю та надає можливість створення ілюстрацій та дизайн персонажів на основі простих текстових описів або начерків.

На рис. 2 зображено процес генерації елементів графічного інтерфейсу для розробленої авторами гри «Labyrinth of Darkness» за допомогою Leonardo.Ai.

Для досягнення бажаного ігрового стилю у Leonardo.Ai було вибрано пресет «Game Concept» з високим контрастом, що забезпечує насиченість і чіткість зображень. Далі формується текстовий запит, в якому зазначаються необхідні об'єкти: зброя, серце, зілля, пастки, вогонь, зомбі, скелети та боси. Система генерує чотири варіанти зображень у форматі 16:9 середнього розміру (Medium), що дає змогу обрати найвідповідніші для подальшої адаптації в інтерфейсі.

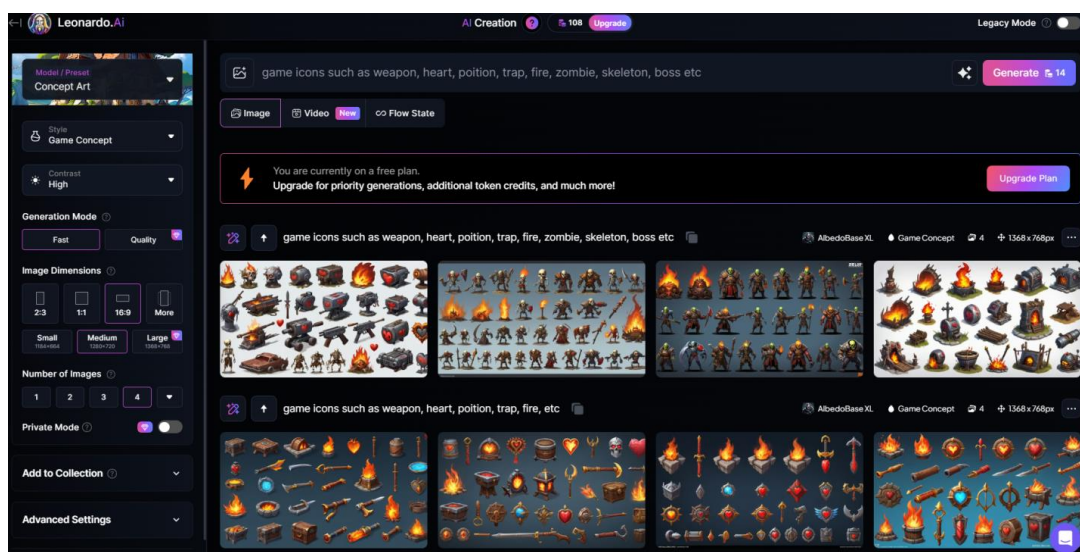


Рис. 2. Створення графічних елементів інтерфейсу для панелі інструментів у гри «Labyrinth of Darkness»

Після вибору оптимального зображення його можна доопрацювати і внести додаткові деталі у Canvas із застосуванням ШІ Leonardo.Ai. Наприклад, для гри «Labyrinth of Darkness» було створено сувій із переліком завдань, якому не вистачало оригінального елемента (штампа у формі черепа). Цей елемент було додано авторами за допомогою окремого текстового запиту (промта). Саме завдяки Leonardo.Ai вдалось досягти високої якості візуальних ресурсів (карти, персонажі, предмети та обладнання) зокрема та інтерфейсу конструктора лабіринтів (рис. 3) загалом, при цьому значно прискоривши процес розробки гри «Labyrinth of Darkness». Платформа дозволяє перетворювати ескізи в деталізовані якісні зображення практично в режимі реального часу за допомогою Realtime Canvas. Це дає змогу миттєво тестувати ітерації дизайнерських ідей перед їх впровадженням у масштабні ігрові проєкти. Крім того,

можливості Elements дозволяють точно контролювати стиль зображень, а функції Sketch to image та Pose to image допомагають формувати композицію та відтворювати пози персонажів відповідно.

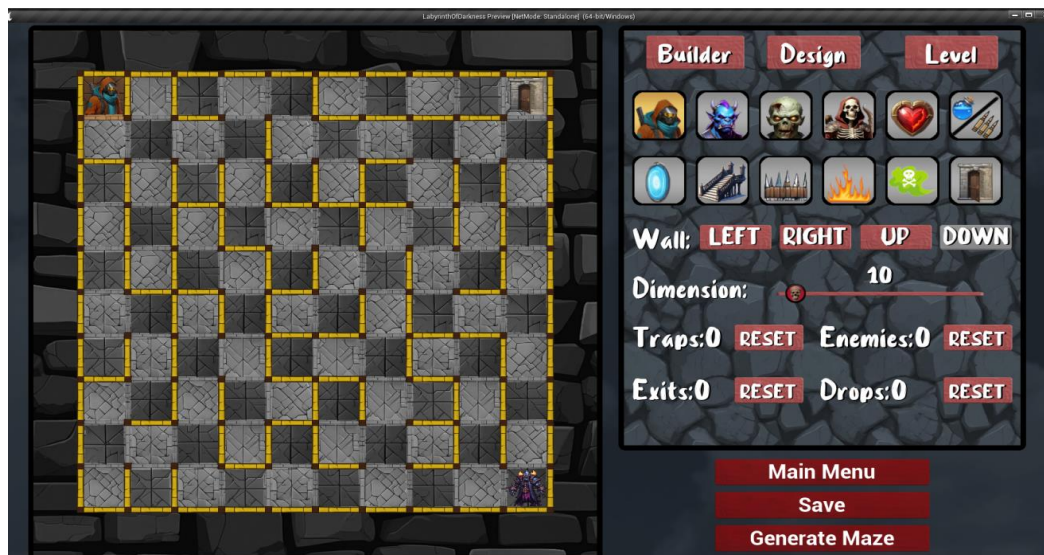


Рис. 3. Розроблені графічні елементи для конструктора рівнів у грі «Labyrinth of Darkness»

Крім того, за допомогою інструмента Leonardo.Ai було розроблено зручний ефективний HUD (Heads-Up Display) — графічний інтерфейс користувача, який відображає важливу інформацію під час гри, не відволікаючи увагу гравця від основного ігрового процесу. Приклад такого інтерфейсу показано на рис. 4, де на екрані є кілька ключових елементів HUD: таймер у верхньому лівому куті (показує обмежений час), червона іконка (у лівому нижньому куті) для відстеження стану здоров'я персонажа гри, список завдань у правому верхньому куті та приціл посередині екрана.

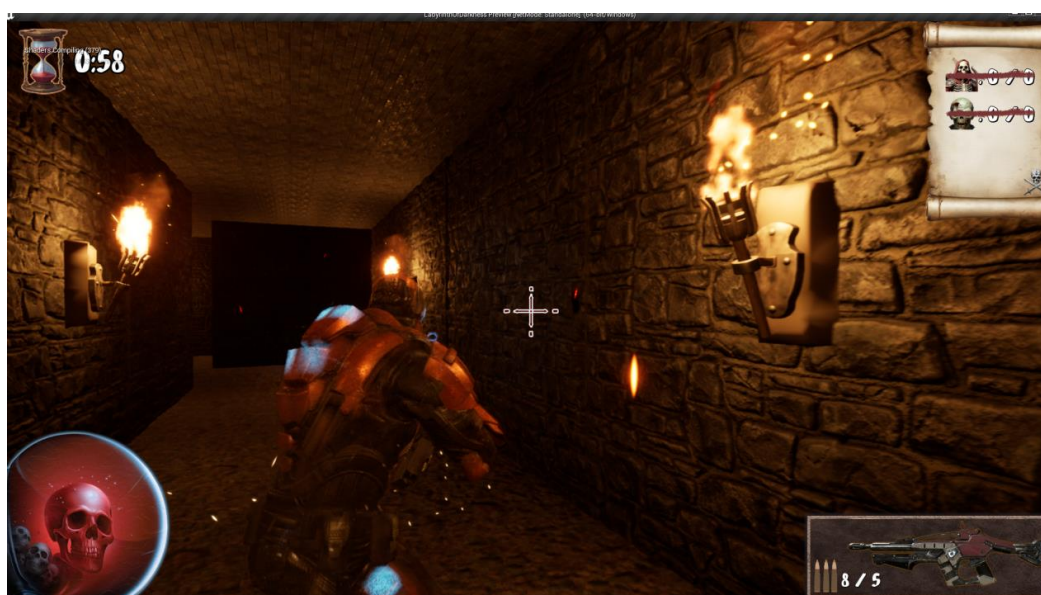


Рис. 4. Результат створення HUD для гри «Labyrinth of Darkness»



Тим самим, експериментальне використання III-інструментів Leonardo.Ai для створення елементів графічного інтерфейсу підтвердило можливість та зручність формування унікального образу гри, який безпосередньо впливає на її атмосферу та загальне сприйняття користувачем.

2. Використання III для створення 3D-об'єктів для ігор

У сучасному ігровому дизайні об'єкти та персонажі становлять основу візуального світу, а їх деталізація є ключовою для занурення гравця у гру. Для професійних 3D-художників розробка детальних моделей може тривати кілька тижнів, тоді як для звичайних творців контенту опанування складних програм, таких як Maya чи Blender, є доволі складним завданням. Розрив між високим попитом на якісний контент та можливістю його ефективного створення ускладнюють також висока вартість ліцензійного програмного забезпечення та значні часові витрати на навчання [12].

Одним із помітних рішень цієї проблеми є платформа Meshy.ai, яка демонструє, як штучний інтелект може значно полегшити та пришвидшити процес генерації 3D-моделей і текстур. Платформа Meshy.ai (<https://www.meshy.ai>) надає інтуїтивно зрозумілий інтерфейс, що дозволяє навіть новачкам у сфері 3D-моделювання створювати базові моделі, завантаживши простий ескіз або надавши текстовий опис бажаного об'єкта. Технологія аналізує введені дані, розуміючи як геометрію, так і матеріали об'єкта, що дозволяє створювати реалістичні моделі з правильною глибиною, текстурами та освітленням. Процес роботи Meshy.ai відбувається у три основних етапи: спочатку відбувається аналіз і розуміння текстового запиту за допомогою великих мовних моделей, потім генерується базова 3D-структура, а на завершальному етапі відбувається деталізація та текстурування моделі [14].

Ще однією важливою перевагою платформи Meshy.ai є її здатність працювати з комплексною геометрією та генерувати надзвичайно реалістичні текстури. Алгоритми глибокого навчання аналізують тисячі зразків текстур, створюючи нові варіації, які виглядають природно та відповідають сучасним вимогам графіки. Після завершення генерації отримані 3D-моделі з вбудованими картами текстур експортуються у формати FBX, OBJ та GLB, готові для застосування в ігровому середовищі.

Типова модель, створена за допомогою Meshy.ai, характеризується оптимальним співвідношенням між кількістю полігонів і ступенем деталізації, що дозволяє працювати з моделями як для інтер'єрного дизайну гри (наприклад, меблі та декоративні елементи), так і для створення персонажів, транспортних засобів, архітектурних елементів та інших елементів ігрового середовища. Крім того, система дозволяє користувачам експериментувати з різними базовими матеріалами (метал, тканина, дерево, скло, синтетичні матеріали) і навіть змішувати їх для досягнення бажаного ефекту. Meshy AI пропонує можливості як текст-до-текстури, так і зображення-до-текстури, що дає змогу генерувати текстури на основі описових підказок або концептуального мистецтва [15].

Створення ігрових 3D-моделей починається з пошуку ідей та референсів, тобто з визначення загальної концепції персонажів. Під час розробки гри «Labyrinth of Darkness» для створення концептуального зображення 3D-моделі «скелетний лицар» у Leonardo.Ai, ключовими словами, що описують бажаний вигляд, були: «скелетний лицар», «темна броня», «готичний стиль». Отримане зображення (рис. 5, а) стало основою для подальшого деталізованого моделювання. Далі зображення імпортовано до Meshy.ai, де на його основі була автоматично згенерована початкова 3D-модель з базовою формою та текстурами. Після цього модель перевірили як у Meshy.ai, так і в 3D-редакторах (це може бути Blender, ZBrush, 3ds Max чи то Maya), де провели корекцію геометрії, усунули

дефекти сітки та виконали ретопологію для оптимізації структури полігонів, що важливо для подальшої анімації. Модель доопрацьовувалась за допомогою скульптингу в ZBrush, де додали дрібні деталі — орнаменти на броні, текстури кісток, тріщини та рельєф. Після цього виконано етап текстуровання, під час якого сформовано і налаштовано UV-розгортку, а також виконано випікання нормалей, рельєфу та карти оточуючого освітлення, доповнюючись базовою кольоровою текстурою. Для оновлення текстур персонажа було застосовано текстовий опис (Prompt), що спрямовував генерацію стародавнього зношеного лицарського обладунку з металевими декоративними елементами. Згенерована лицарська броня має вишукані декоративні елементи, а до текстури застосовано ефекти металевого покриття, що разом додають реалістичності (рис. 5, б). Для подальшої анімації до моделі створено ріг (Rig), що дозволяє зв'язати скелет і геометрію та перевірити деформації під час руху (рис. 5, в). Після успішного тестування 3D-модель експортували у формат, сумісний із рушієм Unreal Engine 4 (UE4) — тим самим Meshy.ai допоміг отримати готову до використання в ігровому рушії модель «скелетного лицаря» (рис. 5, г). Модель демонструє високий рівень деталізації, з ретельно опрацьованою текстурою та анатомічно коректною структурою, що забезпечує зручність анімації в ігровому рушії Unity або Unreal Engine.

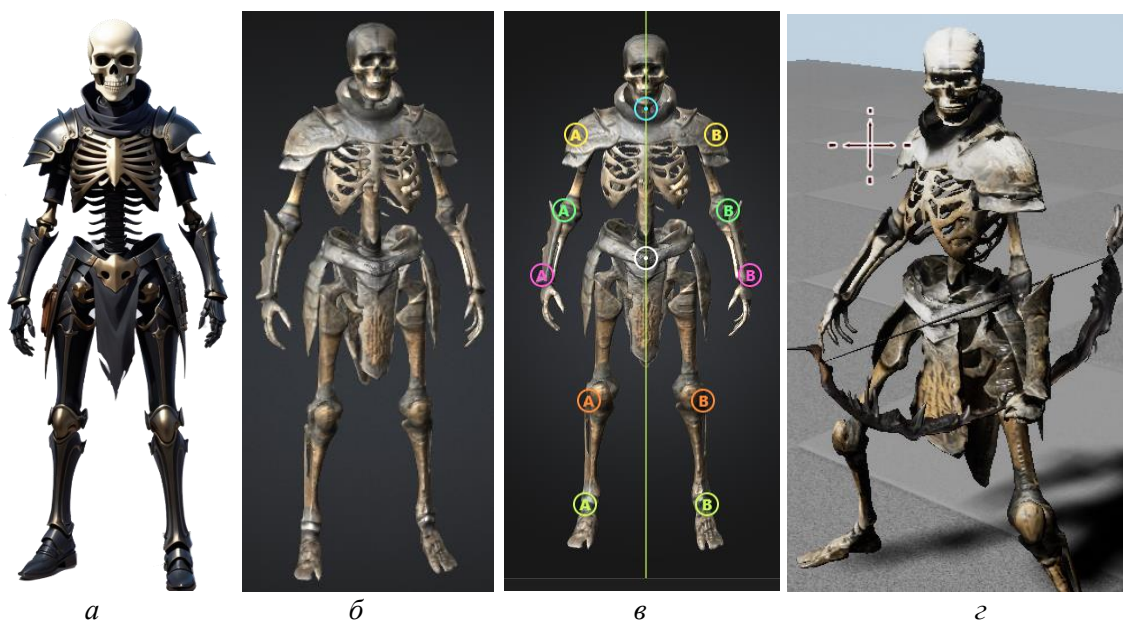


Рис. 5. Поетапне створення 3D-моделі лицаря-скелета для гри «Labyrinth of Darkness» за допомогою ШІ-інструментів

- а) концептуальне зображення створене в Leonardo.Ai;
- б) 3D-модель в Meshy.ai з оновленою текстурою й додаванням ефекту металевого покриття;
- в) додавання до моделі опорних точок для подальшої анімації в Meshy.ai;
- г) застосування розробленої 3D-моделі в ігровому рушії UE4

За аналогічною схемою було створено не лише персонажів, а й зброю, спорядження та інші елементи ігрового середовища. Запропонований підхід дозволяє швидко отримувати базову модель, допрацьовувати її на власний розсуд та інтегрувати у рушій, що було успішно реалізовано у «Labyrinth of Darkness».



3. Використання ШІ для створення музики для відеоігор

Музика та звуковий супровід є важливими складовими ігрового досвіду, адже вони формують емоційний фон і сприяють зануренню користувача в ігровий світ. У традиційній парадигмі створення музичного контенту залучаються композитори, студії звукозапису та професійне обладнання, що зумовлює високі витрати часу й ресурсів. Сучасні технології на базі ШІ, як-от платформа AIVA, змінюють цей процес, роблячи його більш доступним та гнучким.

AIVA (<https://www.aiva.ai>) застосовує методи глибокого навчання для аналізу великої кількості музичних творів, створених відомими композиторами, з метою виявлення стилістичних закономірностей у нотних партитурах. Як стверджує компанія, «AIVA практикує своє розуміння музики, пророкуючи, що буде далі в треку. Як тільки вона досягає успіху в цих прогнозах, вона створює набір математичних правил для даного стилю музики» [16]. Для запобігання монотонності композицій система використовує глибокі нейронні мережі, які формують варіанти нових творів. За допомогою AIVA було створено саундтрек Genesis для відеогри «Pixelfield», що вважається першим комерційним проєктом із музичним супроводом, згенерованим ШІ.

Для гри «Labyrinth of Darkness» автори статті використовували можливості AIVA для розробки музичного супроводу. Це дозволило створити унікальне звукове оформлення, яке гармонійно доповнює похмуру, напружену атмосферу гри. ШІ згенерував динамічні, емоційно насичені композиції, які підсилюють напругу під час дослідження лабіринтів, зіткнень із ворогами та проходження пасток. Музика плавно змінюється відповідно до подій у грі, тим самим сприяє глибшому зануренню гравця в ігровий світ і робить рівень більш захопливим та атмосферним.

Інтеграція можливостей AIVA допомагає згенерувати якісні музичні треки для відеогри з мінімальними часовими та фінансовими витратами. Крім автоматичної генерації, платформа надає інструменти для редагування та персоналізації композицій, що забезпечує можливість адаптації звучання до конкретної атмосфери й емоційного навантаження проєкту.

4. Використання ШІ для озвучування персонажів

Озвучування персонажів у відеоіграх традиційно вимагало залучення професійних акторів, оренди студій звукозапису та тривалих періодів постпродакшну. Сучасні ШІ-технології, зокрема сервіс Replica Studios та подібні платформи, оптимізують ці процеси, зменшуючи витрати часу й ресурсів. Озвучування персонажів гри за допомогою інструментів ШІ відрізняє реалістичність та широкий спектр ролей, що дає творчим особам повний контроль над своєю розповіддю. Цей сервіс використовує технології синтезу промови на основі ШІ, дозволяючи розробникам вибирати з безлічі голосів та налаштовувати інтонацію, темп та емоції, щоб створити унікальні озвучки для своїх персонажів [17].

Застосування ШІ-технологій в озвученні є особливо корисним для незалежних розробників, котрі можуть використовувати готові платформи з інструментаріями розробника (SDK), а також відкриті бібліотеки й API, що сприяє зниженню витрат і прискоренню інтеграції голосових рішень [18]. Наприклад, студія Neon Giant у співпраці зі Sweet Justice Sound та компанією Altered (розробник інструмента Altered Studio) інтегрували ШІ-технологію для генерації людських голосів у грі «The Ascent» (виданої Curve Games). Використовуючи запис голосу актора Сема Х'юза, було додано 800 нових діалогових реплік для 40 NPC, які раніше йшли лише з субтитрами; це дозволило суттєво зменшити обсяг необхідних ресурсів для виробництва озвучення [19Error! Reference source not found.].

Наразі інструменти ШІ спрощують процес локалізації голосового контенту в іграх. Завдяки ШІ розробники можуть автоматично перекладати та озвучувати діалоги персонажів, зберігаючи інтонацію, емоційність та культурні особливості мови. Це дозволяє охопити ширшу аудиторію та зробити гру доступною для гравців з різних країн.

5. Використання ШІ для створення анімації у відеоіграх

Традиційно створення реалістичних анімацій вимагало покадрову ручну роботу. Сучасні технології ШІ дозволяють автоматизувати цей процес, використовуючи алгоритми глибокого навчання для аналізу рухів реальних рухів і перенесення їх на цифрові моделі. Одним із головних напрямків використання ШІ в анімації є автоматичне синтезування рухів персонажів. Відповідні моделі аналізують відеозаписи рухів виконавців і генерують плавні анімаційні переходи, інколи навіть прогножуючи наступні кадри на основі наявних даних.

Яскравим прикладом використання ШІ для автоматичного створення рухів персонажів є платформа DeepMotion (<https://www.deepmotion.com>). Вона дозволяє завантажити відео з рухами реальної людини, після чого ШІ аналізує ці рухи та генерує відповідну 3D-анімацію персонажа. Для створення анімації DeepMotion використовує хмарний конвеєр машинного навчання «Motion Intelligence», який навчений відтворювати складні рухові навички, зокрема паркур, танцювальні комбінації, легку атлетику та елементи бойових мистецтв. DeepMotion надає SDK, сумісний з рушіями Unity та Unreal Engine, що забезпечує сумісність конвеєра з типовими робочими процесами розробки ігор [20].

Процес створення анімації розпочинається з того, що користувач завантажує відео з рухами реальної людини на платформу DeepMotion. У цьому випадку було використано відео, яке демонструє рух натягування лука, з метою створення відповідної анімації для скелета лицаря в грі «Labyrinth of Darkness». Після завантаження система автоматично визначає ключові точки тіла, необхідні для побудови скелетної анімації. Користувач має змогу вибрати додаткові налаштування, зокрема активувати трекінг голови, рук чи записувати тільки верхню частину тіла. На основі цих параметрів ШІ виконує аналіз рухів, після чого створює 3D-модель скелета, яка відображає точну кінематику людини з відео. Користувач може переглянути результат у вбудованому плеєрі та, за потреби, скоригувати окремі пози перед повторним обробленням.

Після отримання прийнятної якості анімації результат експортують у форматі, сумісному з ігровим рушієм. Для гри «Labyrinth of Darkness» такого роду анімація була створена для 3D-моделі персонажа лицаря-скелета та експортована в Unreal Engine 4 (рис. 6), де вона була адаптована до реального ігрового середовища.

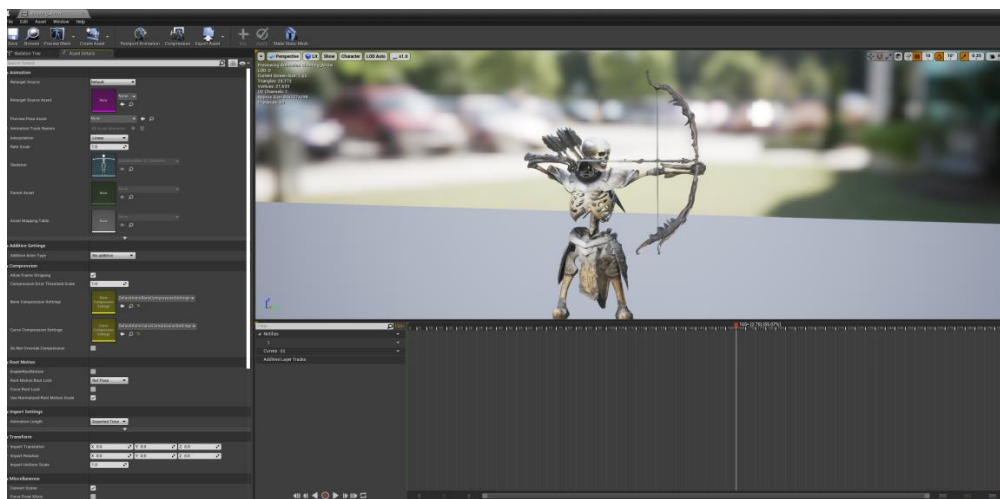


Рис. 6. Результат застосування анімації створеної ШІ від DeepMotion до лицаря-скелета для гри «Labyrinth of Darkness»

Застосування DeepMotion дозволило швидко й ефективно перетворити живий рух у повноцінну ігрову анімацію без потреби у використанні дорогого обладнання для захоплення рухів. Також цей інструмент використовувався для створення поз персонажів.

6. Використання ШІ для написання та перевірки коду в іграх

Інтеграція ШІ у процес розробки програмного забезпечення суттєво змінює індустрію відеоігор. Традиційно написання програмного коду є тривалим і складним процесом, яке вимагає від розробників високої концентрації та точності, а також постійного аналізу можливих помилок і вразливостей. Сучасні інструменти на базі ШІ дозволяють автоматизувати цей процес: вони пропонують розробникам рекомендації з оптимізації коду, генерують шаблони та виконують перевірку на наявність помилок у реальному часі.

Одним з прикладів таких інструментів є GitHub Copilot, який використовує моделі машинного навчання для аналізу контексту коду та надання варіантів вирішення конкретних задач. Ця технологія дозволяє прискорити розробку, зменшити кількість людських помилок і підвищити ефективність робочого процесу. Крім того, інтелектуальні системи здатні оцінювати код на відповідність до кращих практик програмування, виявляти потенційні вразливості в аспектах безпеки та пропонувати шляхи їх усунення ще на ранніх стадіях розробки [21]. Такий підхід зменшує час, витрачений на ручне тестування і налагодження, а також дозволяє розробникам зосередитись на творчих завданнях, розробляти нові функції та покращувати взаємодію користувачів з грою. Завдяки застосуванню інтелектуальних інструментів розробка коду стає більш ефективною, а процес контролю якості програмного забезпечення — точнішим і надійнішим, що сприяє створенню високоякісних та стабільних відеоігор.

Наразі GitHub Copilot використовують понад мільйон розробників та впровадили понад 20 000 організацій. Використання можливостей GitHub Copilot допомогло розробникам виконувати завдання на 55% швидше. Ба більше, за розрахунками фахівців [22] цей інструмент, завдяки підвищенню продуктивності розробників програмного забезпечення на 30%, до 2030 року може збільшити світовий ВВП на понад 1,5 трильйона доларів, що сприятиме економічній активності та прискоренню цифрової трансформації в усьому світі.

7. Використання ШІ для написання діалогів і текстового наповнення для ігор

Розробка сценаріїв та написання діалогів традиційно потребує значних зусиль від сценаристів, оскільки текст має бути граматично коректним, узгодженим зі стилістикою та відповідати сюжетним вимогам відеогри. Сучасні моделі обробки природної мови, наприклад GPT, дозволяють частково автоматизувати цей процес, генеруючи варіативні версії діалогів, які адаптуються до контексту конкретних ігрових ситуацій. За допомогою ШІ сценаристи отримують кілька варіантів тексту, що допомагає краще підібрати тон, емоційну насиченість та стилістику голосу персонажів.

Крім того, інтелектуальні системи здатні аналізувати контекст і структуру сценарію, виявляти логічні невідповідності та пропонувати корективи, що зменшує обсяг вручну виконуваних правок і підвищує якість текстового наповнення. У практиці провідних ігрових студій подібні технології вже застосовуються для розробки квестів зі складною розгалуженою логікою, а також для автоматичного перекладу діалогів з урахуванням мовних і культурних особливостей цільової аудиторії. Останні досягнення в сфері ШІ значно поліпшили якість і ефективність процесу локалізації ігор. Зокрема, нейронно-машинний переклад (NMT) продемонстрував покращення точності та швидкості перекладу



[23]. На відміну від попередніх систем, NMT здатен краще розуміти контекст, що дозволяє



створювати більш природні й адекватні переклади. Це особливо важливо для збереження цілісності сюжету та діалогів гри.

Прикладом інтерактивної системи на основі ШІ, що використовується для динамічного створення історій і діалогів, є проєкт AI Dungeon (<https://aidungeon.com/>), розроблений компанією Latitude. У цій текстовій пригодницькій грі мовна модель генерує описи подій і відповіді персонажів у режимі реального часу залежно від виборів користувача, забезпечуючи варіативність сюжету. Завдяки застосуванню GPT-3 і здатності моделі до глибокого аналізу контексту розробники отримують інструменти для створення адаптивного оповідання, яке зберігає внутрішню узгодженість і підтримує відповідну стилістику при локалізації на різні мови.

8. Використання ШІ для тестування комп'ютерних ігор

Сучасні технології ШІ відкривають нові можливості для тестування відеоігор, що є важливим етапом у забезпеченні якості кінцевого продукту. У традиційних методах тестування основний обсяг робіт виконується вручну, що вимагає великих часових та людських ресурсів і може бути схильним до людських помилок. Завдяки застосуванню ШІ розробники можуть автоматизувати виявлення багів та недоліків. Системи, що базуються на глибокому навчанні, здатні аналізувати великі обсяги у режимі реального часу, що дозволяє швидко знаходити проблеми, навіть ті, які раніше могли залишатися непоміченими. Такі системи можуть імітувати різні стилі поведінки користувачів та тестувати альтернативні шляхи проходження гри, що допомагає виявити помилки в логіці геймплею або збої в роботі штучного інтелекту ігрових персонажів.

Прикладом інтеграції ШІ для автоматизованого тестування є рішення компанії Razer: Razer Game Assistant та Razer QA Companion, які входять до складу ігрової екосистеми WYVRN і спрямовані на оптимізацію процесу розробки ігор. Інтелектуальний помічник Razer QA Companion допомагає автоматизувати процес тестування, оптимізувати ідентифікацію та відтворення помилок шляхом автоматичного виявлення аномалій під час сеансів тестування, фіксуючи точну послідовність дій користувача, реакцій системи та умов навколишнього ігрового середовища, що призводять до помилки [24]. Іншим прикладом є хмарний плагін AI QA Copilot для ігрових рушіїв Unreal Engine та Unity, який може спостерігати за процесом гри, виявляти баги та проблеми з продуктивністю, формувати звіти про тестування з виявленими помилками. За даними досліджень [25] AI QA Copilot здатен знаходити на 20–25% більше помилок порівняно з традиційним тестуванням, знизити час на тестування на 50% та зменшити витрати до 40%.

Ще одним ШІ-інструментом використання для тестування гри є застосування Behavior Tree (поведінкового дерева) — одного з ключових інструментів, який надає Unreal Engine 4 для створення, візуалізації та налагодження логіки поведінки персонажів. За допомогою цього механізму розробники можуть конструювати складну поведінку ботів або NPC без потреби прописувати її на рівні коду, використовуючи візуальний редактор з ієрархічною структурою. Завдяки цьому UE4 надає можливості для створення та тестування гнучкої, адаптивної поведінки ворогів. У процесі тестування гри «Labyrinth of Darkness» інструмент Behavior Tree застосовано для автоматизованої перевірки реакцій NPC у різних ігрових умовах, перевірки коректності анімацій та вибору дій, а також виявлення «зависань» логіки, що суттєво підвищило швидкість і якість виявлення помилок.

Тим самим активна інтеграція технології ШІ в сучасний процес розробки відеоігор змінює підходи до тестування та вдосконалення ігрових продуктів. Разом з тим використання ШІ у GameDev не є ідеальним інструментом, що зумовлено як технічними,



так і концептуальними обмеженнями сучасних генеративних моделей, а також природою самого процесу розробки ігор.

Складнощі та недоліки використання ШІ в розробленні та тестуванні ігор

Генеративні ШІ-моделі часто працюють на основі текстових або візуальних підказок, але не завжди розуміють повну картину світу гри, її механіки або сюжет. Це може призвести до створення нелогічного або неузгодженого контенту. Наприклад, у фентезі-грі з суворою внутрішньою логікою може з'явитися NPC, поведінка якого суперечить встановленим правилам вигаданого світу через те, що ШІ просто не врахував всіх деталей і логіки вигаданого світу гри — того, що називається лором (Game Lore). Генеративний ШІ може створювати унікальний ігровий контент, але його якість не завжди відповідає стандартам. Наприклад, алгоритми можуть генерувати нерівномірний рівень складності або недопрацьовані сюжетні лінії. Це може привести до ситуацій, коли гра здається незбалансованою чи нецікавою.

У тестуванні ігор генеративний ШІ теж має обмеження. Хоча він може автоматизувати створення тест-кейсів і виявлення помилок, його складно навчити розуміти складні ситуації зі значною кількістю взаємопов'язаних змінних, які виникають у грі з багатьма змінними. Автоматизований тест може пропустити баги, які для людини є очевидними, наприклад, злам балансу через певну комбінацію навичок або неприродна поведінка персонажа в конкретному сценарії [26]. Це пов'язано з тим, що генеративний ШІ не має інтуїції чи когнітивного досвіду гравця, який мають тестувальники. Людина, яка тестує гру, орієнтується на емоції, очікування, звички гравців і здатна виявити «тонкі» баги, які ШІ не вважає аномаліями. Крім того, моделі не мають постійного стану пам'яті чи внутрішньої моделі ігрового світу, тому не можуть виявити логічні порушення на великих часових відрізках чи в складних ланцюгах подій.

Ще одна важлива причина — етичні та правові обмеження даних. Моделі навчаються на відкритих джерелах, які далеко не завжди стосуються професійних геймдизайнерських практик, технічних вимог до ігор чи то комерційних стандартів. У підсумку генеративний контент часто виявляється поверхневим, шаблонним або навіть надмірно повторюваним. Це також пояснює проблему авторського права: якщо модель тренується на творах без ліцензії або дозволів, вона може згенерувати результат, подібний до наявного твору, що створює юридичні ризики порушення прав третіх осіб. Крім того, є ризик створення небажаного або токсичного контенту. Це особливо критично в комерційній розробці, де будь-який юридичний спір може обернутись серйозними наслідками.

З іншого боку, надмірне використання генеративного ШІ може призвести до витіснення творчих спеціалістів — художників, сценаристів, дизайнерів ігрових рівнів. Водночас результати ШІ часто потребують доопрацювання, що означає додаткові витрати часу і ресурсів. Наприклад, згенерований сценарій може мати правильну структуру, але бути емоційно порожнім або не відповідати загальному тону гри. Генеративний ШІ не здатен повноцінно замінити цю командну динаміку, бо не володіє емоційною, креативною чи стратегічною гнучкістю людини. Навіть найкращі ШІ-моделі залишаються лише зручними інструментами, а не повноцінними учасниками процесу.

Також варто врахувати високі вимоги до обчислювальних ресурсів, оскільки генеративний ШІ потребує значних обчислювальних потужностей для створення складного ігрового контенту. Технологія нейронного рендерингу, яку застосовують у сучасних іграх, вимагає потужних серверних інфраструктур, що значно підвищує бюджет на розробку [27]. Це може бути фінансово обтяжливо для інді-студій або малих розробників.



Сучасна розробка ігор є складним міждисциплінарним процесом. У ньому взаємодіють програмісти, сценаристи, художники, дизайнери, продюсери, тестувальники, психологи. Якщо студії надто покладаються на автоматичне створення контенту засобами ШІ, унікальність та оригінальність ігрового досвіду можуть постраждати, гра може відчуватися менш «живою», а її рівні й механіки надто одноманітними.

Генеративний ШІ використовується для створення рівнів у багатьох rogue-like іграх, де кожен рівень та сценарій важливий для створення непередбачуваного і захоплюючого досвіду для гравця, але часто вони відчуються шаблонними та позбавленими «людської» творчості. ШІ, хоч і здатен створювати нескінченну кількість варіацій, все одно залишається в межах визначених алгоритмами патернів, що може призвести до відчуття шаблонної повторюваності. Так, у більшості rogue-like ігор згенеровані рівні часто будуються за допомогою певних стандартних елементів: приміщення, коридори, вороги та предмети, і ці елементи часто розташовуються за принципом випадковості, без урахування загального балансу та атмосфери. Хоча такі рівні можуть бути різними по своїй структурі, вони все одно можуть виглядати схожими через відсутність глибокої людської креативності в їх дизайні. Подекуди це може призвести до відчуття, що, хоча рівні і змінюються, сама структура гри залишається насправді одноманітною. Особливо це відчувається, коли ті самі типи ворогів та пасток зустрічаються знову і знову. Це негативно позначається на оригінальності ігрового досвіду, бо гравці починають помічати схожі патерни при проходженні рівнів. При цьому проблема не стільки у використанні ШІ, а в тому, що він не здатен створити того рівня розмаїття, який можуть запропонувати справжні дизайнери. Саме дизайнери можуть адаптувати ігрові рівні, додавати елементи сюжету разом з відчуттям прогресії та глибини, а також враховувати емоційні реакції гравця на певні ситуації, що є складним для ШІ.

Всі зазначені складнощі й недоліки переважно зумовлені глибинними технічними обмеженнями моделей, відсутністю глибокого контекстного розуміння, обмеженістю навчальних даних і невідповідністю ШІ до багатогранності людської творчої діяльності в ігровій індустрії. Тому з розвитком технологій і можливостей генеративного ШІ, стає важливою роль людського контролю та креативного втручання в процес розроблення та тестування гри. Хоча генеративний ШІ відкриває можливості для швидкого створення великого обсягу ігрового контенту, важливо збалансувати його використання з людським дизайном, щоб зберегти унікальність і глибину геймплею. Людська творчість дозволяє внести несподівані елементи, емоційні моменти та складну взаємодію, чого поки що не може досягти ШІ в контексті геймдизайну. Наразі генеративний ШІ у розробці та тестуванні ігор потребує ретельного контролю, адаптації та тестування. Найкращим рішенням є поєднання можливостей ШІ та творчого підходу розробників, що гарантує якісний контент й унікальний ігровий досвід.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Сучасні технології ШІ радикально трансформують всі етапи розробки комп'ютерних ігор, відкриваючи нові можливості для оптимізації робочих процесів та підвищення якості кінцевого продукту. Використання інструментів на базі ШІ, таких як Meshy.ai, Leonardo.Ai, AIVA, DeepMotion, GitHub Copilot, Replica Studios та AudioCraft, дозволяє знизити трудомісткість розробки, скоротити часові витрати і, що не менш важливо, покращити кінцевий результат. Завдяки впровадженню цих інноваційних рішень розробники отримують можливість зосередитись на творчості, а рутинні



операції, такі як генерація 3D-моделей, створення анімацій, написання коду та локалізація текстів, стають більш автоматизованими та точними.

Впровадження ІІІ-технологій не лише підвищує ефективність виробничих процесів, а й допомагає вирішувати соціальні проблеми ігрової галузі, зокрема надмірне навантаження на співробітників («crunch»). Застосування генеративних інструментів може значно знизити кількість годин роботи, що позитивно впливає на психологічний стан розробників та сприяє більш збалансованій організації робочого процесу створення ігрових проєктів.

З точки зору перспектив подальших досліджень, актуальним залишається аналіз впливу ІІІ на всі аспекти розробки комп'ютерних ігор. Необхідно дослідити, як нові технології можуть сприяти більшій інтеграції творчих процесів у виробничі цикли, забезпечуючи при цьому високий рівень інновацій та унікальності кінцевих продуктів. Особлива увага має бути приділена дослідженню можливостей автоматизації тестування і перевірки коду за допомогою ІІІ, що може забезпечити вищу стабільність й безпеки ігрового програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Global video game market value from 2022 to 2032 (in billion U.S. dollars). Statista. <https://www.statista.com/statistics/292056/video-game-market-value-worldwide/>
2. Трофименко, О.Г., Дика, А.І., Лобода, Ю.В., Толокнов, А.А., Бондаренко, М.С. (2025). Штучний інтелект у розробці ігор. *Кібербезпека: освіта, наука, техніка*, 3(27), 109-119. <https://doi.org/10.28925/2663-4023.2025.27.701>
3. Cui, Y. (2025). The Exploring of AI Applications in Game Development. *Applied and Computational Engineering*, 158, 59-64. <https://doi.org/10.54254/2755-2721/2025.TJ23324>
4. Filipović, A. (2023). The role of artificial intelligence in video game development. *Kultura Polisa*, 20(3), 50-67. <https://doi.org/10.51738/Kpolisa2023.20.3r.50f>
5. Gallotta, R., Todd, G., Zammit, M., Earle, S., Liapis, A., Togelius, J. (2024). Large Language Models and Games: A Survey and Roadmap. *IEEE Transactions on Games*, 13 September 2024, 1-18. <https://doi.org/10.1109/TG.2024.3461510>
6. Todd, G., Earle, S., Nasir, M., Green, M., Togelius, J. (2023). Level Generation Through Large Language Models. *Proceedings of the 18th International Conference on the Foundations of Digital Games (FDG '23)*, 70, 1-8. <https://doi.org/10.1145/3582437.3587211>
7. Anjum, A., Li, Y., Law, N., Charity, M., & Togelius, J. (2024). The Ink Splotch Effect: A Case Study on ChatGPT as a Co-Creative Game Designer. *Proceedings of the 19th International Conference on the Foundations of Digital Games (FDG '24)*, 18, 1-15. <https://doi.org/10.1145/3649921.3650010>
8. Li, D., Zhang, S., Sohn, S., Hu, K., Usman, M., & Kapadia, M. (2025). Cardiverse: Harnessing LLMs for Novel Card Game Prototyping. *arXiv*, 2502.07128. <https://doi.org/10.48550/arXiv.2502.07128>
9. Yang, D., Kleinman, E., Hartevel, C. (2024). GPT for Games: An Updated Scoping Review (2020-2024). *arXiv*, 2411.00308. <https://doi.org/10.48550/arXiv.2411.00308>
10. Ling, L., Chen, X., Wen, R., Li, T., Lc, R. (2024). Sketchar: Supporting Character Design and Illustration Prototyping Using Generative AI. *Proceedings of the ACM on Human-Computer Interaction*, 8, 1-28. <https://doi.org/10.1145/3677102>
11. Saersenk, K., Chinibayeva, T. (2024). Affective computing methods for simulation of action scenarios in video games. *Procedia Computer Science*, 231, 341-346. <https://doi.org/10.1016/j.procs.2023.12.214>
12. Humble, N. (2024). AI Supported Game Development: Exploring Opportunities and Challenges. *2nd Symposium on AI Opportunities and Challenges (SAIOC)*. <https://doi.org/10.13140/RG.2.2.30170.76480>
13. AI use in video game development - statistics & facts. <https://www.statista.com/topics/13437/artificial-intelligence-ai-use-in-video-game-development/>
14. Meshy AI Review: How I Generated 3D Models in One Minute. <https://www.unite.ai/meshy-ai-review/>
15. Meshy AI: Transform Text into 3D Textures. <https://www.cutout.pro/learn/meshy-ai/>
16. AIVA Technology – Composing Music using AI. https://d3.harvard.edu/platform-digit/submission/aiva-technology-composing-music-using-ai/?utm_source=chatgpt.com



17. Replicastudios: AI Voice Actors for Games, Films, and Animation I. <https://hunts-ai.onrender.com/ai/replicastudios/>
18. Furkan, Ö. How AI Voice Technology is Transforming the Gaming Industry. <https://speaktor.com/ai-voice-in-gaming/>
19. Neon Giant teams up with Altered for voice in action RPG game The Ascent. <https://www.altered.ai/blog/neon-giant-teams-up-with-altered-for-voice-in-action-rpg-game-the-ascent/>
20. Tokarev, K. DeepMotion: AI Driven Motion for Games. <https://80.lv/articles/deepmotion-ai-driven-motion-for-games>.
21. What is GitHub Copilot? <https://docs.github.com/en/copilot/about-github-copilot/what-is-github-copilot>.
22. The economic impact of the AI-powered developer lifecycle and lessons from GitHub Copilot. <https://github.blog/news-insights/research/the-economic-impact-of-the-ai-powered-developer-lifecycle-and-lessons-from-github-copilot/>
23. Game On: How AI is Revolutionizing Game Localization! <https://www.linkedin.com/pulse/game-how-ai-revolutionizing-localization-midlocalize-wq6of>
24. Razer Reveals WYVRN: AI-Powered Gaming Ecosystem Set to Transform Play and Development. <https://press.razer.com/company-news/razer-reveals-wyvrn-ai-powered-gaming-ecosystem-for-game-developers/>
25. Razer launches new Wyvrn game dev platform with automated AI bug tester. <https://www.theverge.com/news/632121/razer-wyvrn-developer-ai-qa-gamer-copilot-sensa-haptics-thx>.
26. Трофименко, О.Г., Дика, А.І., Лобода, Ю.Г. (2023). Аналіз інструментів тестування вебзастосунків. *Кібербезпека: освіта, наука, техніка*, 4(20), 62-71. <https://doi.org/10.28925/2663-4023.2023.20.6271>.
27. Inzoi's 'Smart Zoi' AI system sounds great on paper but seeing it in a live demo didn't exactly wow me. <https://www.pcgamer.com/games/life-sim/inzois-smart-zoi-ai-system-sounds-great-on-paper-but-seeing-it-in-a-live-demo-didnt-exactly-wow-me/>

**Olena Trofymenko**

PhD, Associate Professor, Associate Professor at the Department of Information Technologies
National University "Odessa Law Academy", Odessa, Ukraine

ORCID ID: 0000-0001-7626-0886

trofymenko@onua.edu.ua

Anastasiia Dyka

Assistant of Department of Information Technologies
National University "Odessa Academy of Law", Odessa, Ukraine

ORCID ID: 0000-0002-4196-8734

dyka.anastasiia@gmail.com

Olexander Zadereyko

PhD, Associate Professor, Associate Professor at the Department of Information Technologies
National University "Odessa Law Academy", Odessa, Ukraine

ORCID ID: 0000-0003-0497-9861

zadereyko@onua.edu.ua

Olexander Shevchenko

Student

National University "Odessa Law Academy", Odessa, Ukraine

ORCID ID: 0009-0002-1873-1963

dragnilsnatsu@gmail.com

AREAS OF APPLICATION OF ARTIFICIAL INTELLIGENCE IN GAME DEVELOPMENT AND TESTING

Abstract. The integration of artificial intelligence (AI) into the software development process is becoming a real revolution in the video game industry. The article explores and systematizes possible areas of application of AI in video game development and gameplay. In general, eight areas of possible implementation of AI in GameDev are described, namely: creating a graphical interface; creating 3D objects in video games; writing dialogues and text content in video games; creating music; voicing characters; creating animation; writing and testing program code; testing video games. Various areas of possible application of AI in video game development were experimentally tested and confirmed during the development by the authors of a game in the adventure genre with horror elements. The authors detailed the applied aspects of using various modern AI tools using the example of developing the corresponding components of the 3D game environment of the game "Labyrinth of Darkness". The use of AI helps to optimize labor-intensive processes, such as content creation and testing, which helps to speed up production cycles and improve the quality of the final product. To achieve the desired gaming style, AI tools allow you to generate several variants of images of game environment elements in order to select the most suitable ones for further adaptation into the game's graphical interface. To do this, deep learning algorithms analyze thousands of texture samples, creating new variations that look natural and meet modern graphics requirements. Thus, the active integration of AI technology into the modern video game development process significantly changes the approaches to testing and improving game products. On the other hand, the paper considers the difficulties and disadvantages of using AI in game development and testing. Currently, generative AI in game development and testing requires careful monitoring, adaptation, and testing. The best solution is to combine the capabilities of AI and the creative approach of developers, which guarantees high-quality content and a unique gaming experience.

Keywords: artificial intelligence; AI; AI algorithms; GameDev; game development; testing.



REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Global video game market value from 2022 to 2032 (in billion U.S. dollars). Statista. <https://www.statista.com/statistics/292056/video-game-market-value-worldwide/>
2. Trofymenko, O., Dyka, A., Loboda, Y., Tolocknov, A., & Bondarenko, M. (2025). Artificial Intelligence in the GameDev. *Electronic Professional Scientific Journal «Cybersecurity: Education, Science, Technique»*, 3(27), 109-119. <https://doi.org/10.28925/2663-4023.2025.27.701>
3. Cui, Y. (2025). The Exploring of AI Applications in Game Development. *Applied and Computational Engineering*, 158, 59-64. <https://doi.org/10.54254/2755-2721/2025.TJ23324>
4. Filipović, A. (2023). The role of artificial intelligence in video game development. *Kultura Polisa*, 20(3), 50-67. <https://doi.org/10.51738/Kpolisa2023.20.3r.50f>
5. Gallota, R., Todd, G., Zammit, M., Earle, S., Liapis, A., Togelius, J. (2024). Large Language Models and Games: A Survey and Roadmap. *IEEE Transactions on Games*, 13 September 2024, 1-18. <https://doi.org/10.1109/TG.2024.3461510>
6. Todd, G., Earle, S., Nasir, M., Green, M., Togelius, J. (2023). Level Generation Through Large Language Models. *Proceedings of the 18th International Conference on the Foundations of Digital Games (FDG '23)*, 70, 1-8. <https://doi.org/10.1145/3582437.3587211>
7. Anjum, A., Li, Y., Law, N., Charity, M., & Togelius, J. (2024). The Ink Splotch Effect: A Case Study on ChatGPT as a Co-Creative Game Designer. *Proceedings of the 19th International Conference on the Foundations of Digital Games (FDG '24)*, 18, 1-15. <https://doi.org/10.1145/3649921.3650010>
8. Li, D., Zhang, S., Sohn, S., Hu, K., Usman, M., & Kapadia, M. (2025). Cardiverse: Harnessing LLMs for Novel Card Game Prototyping. *arXiv*, 2502.07128. <https://doi.org/10.48550/arXiv.2502.07128>
9. Yang, D., Kleinman, E., & Hartevelde, C. (2024). GPT for Games: An Updated Scoping Review (2020-2024). *arXiv*, 2411.00308. <https://doi.org/10.48550/arXiv.2411.00308>
10. Ling, L., Chen, X., Wen, R., Li, T., & Le, R. (2024). Sketchar: Supporting Character Design and Illustration Prototyping Using Generative AI. *Proceedings of the ACM on Human-Computer Interaction*, 8, 1-28. <https://doi.org/10.1145/3677102>
11. Saerssenbek, K. & Chinibayeva, T. (2024). Affective computing methods for simulation of action scenarios in video games. *Procedia Computer Science*, 231, 341-346. <https://doi.org/10.1016/j.procs.2023.12.214>
12. Humble, N. (2024). AI Supported Game Development: Exploring Opportunities and Challenges. *2nd Symposium on AI Opportunities and Challenges (SAIOC)*. <https://doi.org/10.13140/RG.2.2.30170.76480>
13. AI use in video game development - statistics & facts. <https://www.statista.com/topics/13437/artificial-intelligence-ai-use-in-video-game-development/>
14. Meshy AI Review: How I Generated 3D Models in One Minute. <https://www.unite.ai/meshy-ai-review/>
15. Meshy AI: Transform Text into 3D Textures. <https://www.cutout.pro/learn/meshy-ai/>
16. AIVA Technology – Composing Music using AI. https://d3.harvard.edu/platform-digit/submission/aiva-technology-composing-music-using-ai/?utm_source=chatgpt.com
17. Replicastudios: AI Voice Actors for Games, Films, and Animation I. <https://hunts-ai.onrender.com/ai/replicastudios/>
18. Furkan, Ö. How AI Voice Technology is Transforming the Gaming Industry. <https://speaktor.com/ai-voice-in-gaming/>
19. Neon Giant teams up with Altered for voice in action RPG game The Ascent. <https://www.altered.ai/blog/neon-giant-teams-up-with-altered-for-voice-in-action-rpg-game-the-ascent/>
20. Tokarev, K. DeepMotion: AI Driven Motion for Games. <https://80.lv/articles/deepmotion-ai-driven-motion-for-games>
21. What is GitHub Copilot? <https://docs.github.com/en/copilot/about-github-copilot/what-is-github-copilot>
22. The economic impact of the AI-powered developer lifecycle and lessons from GitHub Copilot. <https://github.blog/news-insights/research/the-economic-impact-of-the-ai-powered-developer-lifecycle-and-lessons-from-github-copilot/>
23. Game On: How AI is Revolutionizing Game Localization! <https://www.linkedin.com/pulse/game-how-ai-revolutionizing-localization-midlocalize-wq6of>
24. Razer Reveals WYVRN: AI-Powered Gaming Ecosystem Set to Transform Play and Development. <https://press.razer.com/company-news/razer-reveals-wyvrn-ai-powered-gaming-ecosystem-for-game-developers/>



25. Razer launches new Wyvrn game dev platform with automated AI bug tester. <https://www.theverge.com/news/632121/razer-wyvrn-developer-ai-qa-gamer-copilot-sensa-haptics-thx>.
26. Trofymenko, O., Dyka, A., & Loboda, Ju. (2023). Analysis of web application testing tools. *Electronic Professional Scientific Journal «Cybersecurity: Education, Science, Technique»*, 4(20), 62-71. <https://doi.org/10.28925/2663-4023.2023.20.6271>
27. Inzoi's 'Smart Zoi' AI system sounds great on paper but seeing it in a live demo didn't exactly wow me. <https://www.pcgamer.com/games/life-sim/inzois-smart-zoi-ai-system-sounds-great-on-paper-but-seeing-it-in-a-live-demo-didnt-exactly-wow-me/>



This work is licensed under Creative Commons Attribution-noncommercial-sharelike 4.0 International License.