



DOI 10.28925/2663-4023.2025.27.701

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  
[trofymenko@onua.edu.ua](mailto:trofymenko@onua.edu.ua)

**Дика Анастасія Іванівна**

асистент кафедри інформаційних технологій  
Національний університет «Одеська юридична академія», Одеса, Україна  
ORCID ID: 0000-0002-4196-8734  
[dyka.anastasiia@gmail.com](mailto:dyka.anastasiia@gmail.com)

**Лобода Юлія Геннадіївна**

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій  
Національний університет «Одеська юридична академія», Одеса, Україна  
ORCID ID: 0000-0001-7083-552X  
[jul.loboda@gmail.com](mailto:jul.loboda@gmail.com)

**Толокнов Анатолій Арнольдович**

викладач кафедри інформаційних технологій  
Національний університет «Одеська юридична академія», Одеса, Україна  
ORCID ID: 0000-0002-2614-2109  
[arnoldovich1@gmail.com](mailto:arnoldovich1@gmail.com)

**Бондаренко Максим Євгенович**

здобувач вищої освіти  
Національний університет «Одеська юридична академія», Одеса, Україна  
ORCID ID: 0009-0002-7325-1727  
[maximbondarenko20@gmail.com](mailto:maximbondarenko20@gmail.com)

## АЛГОРИТМИ ШТУЧНОГО ІНТЕЛЕКТУ В РОЗРОБЦІ ІГОР

**Анотація.** Статтю присвячено висвітленню питань можливого використання алгоритмів штучного інтелекту (ШІ) в сферу розробки ігор. Через зростання складності відеоігор (ігрових механік, різнопланових сюжетів, ігрового оточення) стає все більш актуальним ефективне застосування новітніх технологій ШІ для розробки відеоігор. Метою роботи є дослідження алгоритмів ШІ щодо можливого застосування у розробці відеоігор та ігровому процесі. Гіпотезою дослідження є те, що використання ШІ у розробці ігор (GameDev) може призвести до скорочення витрат на розробку. На ШІ можна перекласти задачі генерування текстур ландшафту ігрового середовища, адаптивно змінюваного відповідно дій реальних персонажів. Діалоги фонових неігрових персонажів (Non-Player Character, NPC) та звукові ефекти можна згенерувати за допомогою ШІ. Ба-більше, ШІ може генерувати і самі неігрові персонажі, при цьому поведінка таких персонажів може змінюватися, залежно від дій реальних гравців, що створює захоплюючий ігровий досвід гравців. Такий підхід здатен не лише зменшити вартість та тривалість виробництва, а й позитивно вплинути на різноманітність ігрового процесу. У статті розглянуто та систематизовано алгоритми ШІ, які можуть бути використані у розробці ігор, наведено приклади, розглянуто специфіку та переваги їх використання. Для цього використано методологію дослідження, яка поєднала аналіз літературних джерел та проведення дослідження на основі відомостей про впровадження ШІ у GameDev. З'ясовано, що алгоритми ШІ мають великий потенціал ефективного впровадження у GameDev, адже їх впровадження допомагає урізноманітнити ігровий процес, надає унікальний досвід гри кожному гравцю. Гравець отримує значну свободу дій, різні можливі динамічно змінювані сценарії і діалоги з неігровими персонажами. ШІ може не лише реагувати на дії гравця, а й змінювати оточення для того, щоб ігровий процес був цікавим і захоплюючим для гравця, не залежно від рівня його прогресу в грі. Впровадження технологій ШІ в GameDev здебільшого



показує гарну ефективність, але вони потребують високої кваліфікації розробників та врахування ризиків у процесі його впровадження.

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

## ВСТУП

**Постановка проблеми.** Індустрія відеоігор є однією з важливих культурних й економічних явищ сучасності. Прогнозується, що до 2027 року доходи ігрової індустрії складатимуть 213.3 мільярдів доларів США [1]. Але в цій сфері постає проблема у генеруванні нових ідей для ігрових сюжетів. Зі збільшенням кількості ігор створити унікальну гру, яка матиме нові ігрові механіки та по-новому даватиме гравцям відчуття ігрового процесу, дедалі важче.

Штучний інтелект (ШІ) на сьогодні є одним з провідних напрямків в сфері інформаційних технологій. Стрімкий розвиток ШІ та його активна імплементація в усі галузі людської діяльності вказує на його корисність, не залежно від сфери використання. Не в останню чергу бум ШІ (AI Boom [2]) зумовлений популяризацією генеративного ШІ. Його призначення полягає у створенні цікавих зображень, аудіо, відео, медіа з використанням породжувальних моделей (generative model) [3].

**Аналіз останніх досліджень і публікацій.** Можливі сфери і перспективи використання алгоритмів ШІ в розробці відеоігор та ігровому процесі є недостатньо дослідженими. Так, у статті [4] розглянуто етапи формування українського сегменту індустрії GameDev, починаючи з 1990-х років. Автор цієї роботи проаналізував економічні та соціальні аспекти розвитку, вплив міжнародної співпраці та інвестицій, але питання та можливості використання ШІ в розробці ігор не досліджені. У статті [5] розглянуто різні способи використання платформ метавсесвіту (Metaverse) в ігровій індустрії та можливий вплив генеративного ШІ на розробку ігор у метавсесвіті. Зазначено, що ШІ може покращити створення персонажів, ігровий дизайн, дизайн рівнів та візуальні ефекти у виробництві ігор. Це дослідження спрямовано на історичний огляд інтеграції Metaverse з іграми на основі ШІ протягом 2020–2024 років. Автори роботи [6] розробили інструмент Sketchar на базі генеративного ШІ, який допомагає дизайнерам створювати прототипи ігрових персонажів і генерувати зображення на основі концептуальних вхідних даних. У статті [7] представлено систему розпізнавання емоцій на основі нечіткої логіки з використанням нейронних мереж, яку можна використовувати для динамічної генерації ігрового контенту, оскільки емоційна складова для гравця є важливою. В роботі [8] йдеться про можливі проблеми використання ШІ для підтримки дизайну у розробленні навчальних ігор.

Проведений аналіз наявних публікацій свідчить про важливість дослідження можливих сфер застосування ШІ в індустрії GameDev, а повсюдне впровадження технологій ШІ потребує підвищеної уваги до цього напрямку та систематизації підходів щодо можливого впровадження алгоритмів ШІ у відеоігрову індустрію.

**Мета статті:** дослідити та систематизувати алгоритми ШІ щодо їх можливого застосування у розробці відеоігор та ігровому процесі.

## РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У розробці відеоігор ШІ може використовуватися для створення складних механік гри, керування поведінкою персонажів, розробки ігрових сценаріїв, а також для аналізу даних про різні аспекти процесу створення та тестування ігор, а також покращення ігрового досвіду. На різних етапах розробки ігор зі ШІ задля створення інтерактивних, адаптивних, реалістичних ігрових світів можуть використовуватись різні алгоритми та методи ШІ (рис. 1).



Рис. 1. Алгоритми ШІ в GameDev

– **Нейронні мережі** (Neural networks, NN) та **глибоке навчання** (Deep Learning, DL) — це алгоритми машинного навчання, які можна навчати та на льоту перенавчати за різними наборами даних, тим самим забезпечуючи різні сценарії та адаптацію поведінки неігрових персонажів (Non-Player Character, NPC) при взаємодії з реальними гравцями, створюючи складний і реалістичний досвід для геймерів. Нейронні мережі використовуються для розпізнавання патернів, генерації контенту та адаптивної поведінки в іграх. Так, **глибокі нейронні мережі** (Deep Neural Networks, DNN) можуть навчатися на великих наборах даних для аналізу поведінки гравців, прогнозування стратегій ворогів, розпізнавання облич, автоматичного генерування рівнів і сценаріїв гри, адаптивних та реалістичних реакцій NPC на дії гравця [9]. **Генеративні змагальні мережі** (Generative Adversarial Networks, GANs) можуть використовуватися для створення реалістичних текстур та моделей для ігрових персонажів [10]. Загалом нейронні мережі та глибоке навчання є одною з найпотужніших технологій для розв'язання складних задач розпізнавання образів, прогнозування результатів і для створення поведінки NPC, адаптованої до взаємодії з гравцем.

– **Генетичні алгоритми** (Genetic Algorithm, GA) є еволюційними алгоритмами, які використовують принципи природного відбору для знаходження оптимальних рішень у складних середовищах. Вони є потужним інструментом для складних завдань оптимізації, навчання та еволюції ігрових систем. Ці алгоритми здатні покращити стратегії ворогів, оптимізувати ігровий баланс [11]. Замість того щоб вручну створювати великі обсяги контенту, можна застосовувати генетичні алгоритми для генерування нових елементів гри (рівнів, різноманітних карт, ландшафтів або навіть унікальних сюжетних ліній), що відповідають заданим вимогам або бажаним характеристикам і водночас оптимізують баланс між складністю і цікавістю. Генетичний алгоритм може



використовувати зворотний зв'язок від гравців (наприклад, оцінки або результати ігор), щоб знаходити комбінацію параметрів, яка зробить гру більш цікавою та збалансованою. Перевагами використання генетичних алгоритмів у відеоіграх є динамічне адаптування до змін у поведінці гравців, оптимізація ігрових систем для покращення досвіду користувачів, автоматичне створення контенту, що дозволяє зекономити час розробників, та інноваційний підхід до створення нестандартних, складних ігрових механік. Генетичний алгоритм використовує процес еволюції, щоб знаходити найкращі стратегії за допомогою селекції, кросоверних схрещувань та мутацій, щоб оптимізувати поведінку NPC для більш реалістичного й складного досвіду гри. У складних ігрових світах, де NPCs або гравці мають знаходити оптимальні маршрути у складних ландшафтах або лабіринтах, генетичні алгоритми можуть допомогти в оптимізації процесу пошуку шляхів. В цілому генетичні алгоритми дають можливість розробникам створювати ігри з високим рівнем адаптивності, креативності і складності, що значно покращує ігровий досвід.

– **Навчання з підкріпленням** (Reinforcement Learning, RL) — це одна з основних технологій, що використовується для тренування агентів в іграх. RL є формою глибокого навчання, яка використовує підхід врахування помилок і наслідків під час пробних дій. Завдяки цьому, з часом NPC здатні навчатися на своїх попередніх помилках, максимізуючи позитивні наслідки своїх рішень і мінімізуючи кількість програвів [12]. Базовими алгоритмами навчання з підкріпленням є *Q-learning* та *Deep Q-Network (DQN)*. Перший з них може бути застосований для створення адаптивних агентів, наприклад, ворогів чи то монстрів, які змінюють свою стратегію, залежно від дій гравця. DQN є вдосконаленою версією алгоритму Q-learning і використовує глибокі нейронні мережі для оцінки дій у складних, багатовимірних просторах. Він може бути використаний для складних ігрових середовищ, де агент має аналізувати та продукувати стратегії в реальному часі в складних ігрових світах.

– **Нейронні мережі для обробки природної мови** (Natural Language Processing, NLP) можна використовувати в розробці відеоігор для покращення інтерактивності, реалістичності та залучення гравців, наприклад, для створення систем діалогів в іграх. Зокрема, *моделі GPT* або *BERT* можуть використовуватись для генерації тексту, створення інтерактивних діалогів з неігровими персонажами та розпізнавання намірів гравця. На основі тренуваних даних, замість фіксованих відповідей, персонажі можуть адаптувати свої відповіді до контексту розмови, враховуючи стиль спілкування гравця і навіть емоційний настрій. *Аналіз настроїв* (sentiment analysis) визначатиме емоційний стан гравця (радість, гнів, сум чи то інші емоції), що впливатиме на реакцію персонажа або розвиток сюжету. *Системи на основі NLP* можуть розпізнавати голосові інтерфейси і команди, що дозволяє гравцям взаємодіяти з грою не лише через традиційні контролери, а й через природну мову. Все це дозволяє створювати гнучку, реалістичну взаємодію з персонажами і не обмежуватись заздалегідь написаними діалогами. При цьому створення великих обсягів тексту для ігор не вимагає написання діалогів вручну, що полегшує роботу розробника. У відкритих світах з безліччю можливих дій, як-от: *Minecraft* чи то *Skyrim*, система NLP може генерувати нові квести, описувати нові локації або навіть адаптувати вже наявні завдання, залежно від вибору гравця та його стилю гри [13]. *Нейронні мережі «Google Translate»* та *«Deer»* можуть використовуватися для автоматичного перекладу тексту розмов у реальному часі [14]. Так, у глобальних багатокористувацьких іграх *«World of Warcraft»* та *«Fortnite»* гравці з різних країн можуть спілкуватися через автоматичний переклад їхніх повідомлень. Крім того, NLP може бути використано для персоналізації взаємодії з гравцем на основі його уподобань,



стилю гри або попередніх дій. Це дозволяє створювати інтуїтивний досвід для кожного гравця з адаптивними діалогами та змінюваними реакціями NPC.

– **Алгоритми класифікації та регресії** (Classification and Regression) є алгоритмами машинного навчання, які можуть бути застосовані для класифікації дій гравців, виявлення аномалій у грі, адаптації до стилю гри гравця тощо. Наприклад, *алгоритми класифікації K-Nearest Neighbors, Decision Trees* є корисними для аналізу поведінки гравців, визначення типів гравців або їх стратегій [15]. В онлайн-шутерах або стратегіях вони можуть допомогти класифікувати гравців на основі їх стилю гри (наприклад, агресивний, стратегічний, оборонний) і надавати персоналізовані рекомендації або контент. За допомогою класифікаційних алгоритмів можна адаптувати контент гри до інтересів і вподобань гравця. Наприклад, на основі історії гравця класифікація може визначити, які жанри, рівень складності чи елементи гри йому подобаються, і формувати відповідні завдання, рівні та нагороди. Використання *регресії* є корисним для прогнозування того, як зміни в ігровому середовищі або поведінці гравця можуть вплинути на його подальші дії, і який час витратить гравець на виконання конкретного завдання, залежно від його попередніх дій, складності рівня або типу гравця. Також регресія може бути використана для прогнозування шансів на успіх у битві на основі рівня гравця, вибору зброї, рівня навичок та інших параметрів. Крім того, в реальному часі можна прогнозувати й оптимізувати складність, залежно від попереднього прогресу гравця (наприклад, якщо гравець занадто швидко проходить рівні, складність збільшується). Загалом алгоритми класифікації та регресії можуть бути ефективно використані для розробки відеоігор у різних аспектах: від покращення взаємодії з гравцем до оптимізації ігрових процесів і поведінки персонажів.

– **Алгоритми для планування** (Planning Algorithms) можуть бути корисними для створення реалістичних і складних взаємодій NPC, наприклад, для виконання місій або досягнення стратегічних цілей у грі. III-алгоритми планування *HTN (Hierarchical Task Networks)* корисні для розробки складних послідовностей дій NPC [16]. Вони дозволяють агентам планувати виконання складних завдань шляхом розбиття їх на підзадачі в ієрархічному порядку, щоб досягти своїх цілей, враховуючи обмеження середовища. Цей підхід дозволяє ігровим агентам адаптувати свої стратегії та виконувати складні завдання, що є важливим для створення реалістичних і адаптивних NPC у відеоіграх. Використання HTN дозволяє створювати більш розумних і багатозадачних ігрових персонажів, що діють на основі планів та змінюють свою стратегію, залежно від нових умов.

– **Алгоритми комп'ютерного зору** (Computer Vision, CV) використовуються для розпізнавання об'єктів у грі, що дозволяє NPC реагувати на зміну середовища. Ці алгоритми комп'ютерного зору дозволяють III виявляти об'єкти в ігровому середовищі (наприклад, стежити за персонажем, визначати перешкоди або взаємодіяти з елементами середовища). Комп'ютерний зір може використовуватися для того, щоб NPCs могли «бачити» та реагувати на навколишнє середовище, подібно до того, як це робить гравець. Алгоритми CV можуть використовуватися для визначення, де розташовані перешкоди або інші персонажі, що дозволяє NPC уникати зіткнень або шукати шляхи до своїх цілей. Так, *конволюційні нейронні мережі (convolutional neural network, CNN)* за допомогою навчання на великих наборах даних можна використовувати для розпізнавання характерних об'єктів гри, таких як зброя, гравці або навіть спецефекти [17]. Алгоритми комп'ютерного зору можуть використовуватися для аналізу просторових даних або для процедурної генерації контенту, такого як рівні або об'єкти на основі наданих зображень і відео. Тим самим алгоритми комп'ютерного зору застосовуються для значного



полегшення розробки ігор, покращення геймплейного досвіду. Крім того, ці алгоритми є ефективними для автоматизації багатьох аспектів тестування відеоігор.

Щодо тестування ігор, розроблених за допомогою ШІ, то це є доволі складною задачею, адже ШІ в іграх складно тестувати через величезну кількість можливих сценаріїв і ситуацій, які генеруються вже під час самої гри. Це може призвести до помилок (багів) і неочікуваних результатів, що ускладнює процес налагодження і знижує якість гри. Саме тому використання ШІ для тестування ігор, розроблених за допомогою ШІ, є інноваційним й ефективним підходом для покращення процесу тестування і забезпечення якості продукту [18]. Оскільки сучасні ігри, зокрема з відкритими світами та складною механікою, мають величезну кількість змінних, застосування ШІ для тестування може значно прискорити і покращити виявлення помилок та проблем із балансом гри. При цьому для автоматизації тестування відеоігор можуть застосовуватись різні алгоритми. Так, алгоритми комп'ютерного зору можуть використовуватись для аналізу сцен на наявність неправильних анімацій, спотворених текстур чи інших дефектів, що можуть бути важко помітними при звичайному тестуванні. Методи сегментації зображень (Image segmentation) дозволяють ділити зображення на різні сегменти для виявлення аномалій в окремих частинах сцени. Нейронні мережі та глибоке навчання для виявлення артефактів можуть бути навчені для виявлення помилок у графічних елементах або для перевірки коректності анімацій, текстур, освітлення. Алгоритми машинного навчання (Machine Learning, ML) можуть скріншотити ігрові екрани та порівнювати їх з еталонними значеннями, що дозволяє виявляти неочевидні баги і проблеми з графікою. Це є корисним для регресійного тестування, коли потрібно перевірити, чи не з'явилися нові проблеми після змін у коді. Отже, ШІ може допомогти автоматизувати тестування ігор на виявлення помилок в ігровому процесі, перевіряючи геймплей на різних етапах розвитку.

Розглянутий список алгоритмів ШІ не є вичерпним, але інші алгоритми або не є поширеними в застосуванні для розробки ігор, або не є безпосередньо алгоритмами ШІ, хоча поширено використовуються в поєднанні з такими алгоритмами в GameDev. Так, еволюційні алгоритми, до складу яких, крім розглянутих вже генетичних алгоритмів, входять *диференціальна еволюція* (Differential Evolution, DE) та *еволюційні стратегії* (Evolution Strategy, ES), можуть застосовуватись для генерації ігрового контенту або для покращення стратегії NPC через багатократні ітерації та еволюційні процеси. Хоча застосування DE та ES не таке поширене, як генетичних алгоритмів, але й вони можуть допомагати створювати складні світи та персонажів, адаптивно змінювані при динамічних змінах у грі. DE є еволюційним методом оптимізації, який використовує різниці між членами популяції для генерації нових кандидатів (рішень). Це дозволяє досягати більшої стабільності та ефективності в процесі оптимізації і може використовуватись для оптимізації стратегії або поведінки NPCs в іграх, особливо коли треба працювати з великими просторами рішень. Еволюційний алгоритм ES орієнтований на оптимізацію неперервних задач і фокусується на використанні спеціальних адаптивних мутацій для покращення пошуку рішень для задач з великими або складними просторами рішень. Тому ES можна використовувати для налаштування параметрів і покращення стратегій та адаптивної поведінки агентів, де важливим є поступовий розвиток і змінюваність поведінки залежно від середовища. Крім того, *еволюційні програми* можуть використовуватись для автоматичного генерування скриптів або програм для віртуальних агентів в іграх. Подекуди для симуляції поведінки та адаптації можна застосовувати *агентно-орієнтовані еволюційні моделі*, в яких агенти віртуального світу самі еволюціонують, покращуючи свої стратегії чи взаємодію з



середовищем. Все це доводить, що DE, ES та інші еволюційні алгоритми є потужними інструментами ІІІ для складних завдань у GameDev.

В GameDev у тісному поєднанні з алгоритмами ІІІ використовують різні математичні моделі, алгоритми та структури даних, які є важливими інструментами для управління поведінкою агентів (NPC) в ігровій розробці і можуть бути частиною більш складної системи ІІІ в грі. Так, абстрактна математична модель *FSM (Finite State Machine, машини кінцевих станів)*, яка описує поведінку системи як набір станів з можливістю переходу між ними на основі певних вхідних сигналів або умов, є класичним методом для управління поведінкою NPC. Вони активно використовуються в ігровій розробці для визначення станів персонажа та управління його переходами з одного стану в інший (наприклад, зі «спокійного» до «агресивного»). Для ворогів у шутерах FSM може використовуватися для визначення того, коли NPC має атакувати, коли тікати, а коли шукати укриття.

Також для розробки поведінки агентів широко використовують «поведінкові дерева» (*Behavior Trees, BT*), оскільки ця структура даних надає навіть більш гнучкий підхід до моделювання та управління поведінкою NPC, порівняно з FSM. BT дозволяють створювати складні, багаторівневі структури для організації дій і рішень NPC [19]. Варіанти адаптивної, реалістичної поведінки NPC формують у вигляді дерева, де кожен вузол відповідає за конкретну дію або умову. Тим самим можна легко створювати складні поведінкові патерни. Керовані алгоритми машинного навчання формують набір правил для NPC на основі конкретних факторів. Наприклад, ворожий NPC у грі може визначати статус персонажа (ворог, союзник, тварина тощо) залежно від того, носить він зброю чи ні. Якщо у персонажа є зброя, NPC може вирішити, що він є ворогом, і зайняти оборонну позицію, зокрема в бойових чи стратегічних іграх. Саме тому поведінкові дерева часто використовуються для створення інтелектуальних ворогів або союзників, де кожна поведінка є результатом виконання низки умов і перевірок.

Окремої уваги заслуговують *алгоритми пошуку  $A^*$  та Дейкстри*, оскільки вони є незамінними для управління рухом ігрових персонажів, забезпечуючи їх здатність рухатися в реальному часі та реагувати на змінення у грі. Алгоритм  $A^*$  поширено використовують в розробці ігор для визначення найкоротшого шляху між двома точками в грі, наприклад, для навігації персонажа. Алгоритм Дейкстри (Dijkstra) призначений для пошуку найкоротших шляхів у графах, а тому є корисним для навігації на великих картах або в складних локаціях.

Техніка *процедурної генерації контенту (Procedural Content Generation, PCG)* використовується в розробці ігор для створення ігрових рівнів, ландшафту і навіть цілих ігор за допомогою різних алгоритмів та математичних моделей замість ручного створення. Такий підхід забезпечує розмаїття й унікальність ігрового контенту. Для PGD в GameDev використовують доволі широкий спектр алгоритмів, серед яких: L-системи, стільникові автомати (Cellular Automata), еволюційні алгоритми, шум Перліна (Perlin Noise), алгоритм ромбовидного квадрата тощо. Так, алгоритм *Perlin Noise* генерує псевдовипадкові значення з певними природними властивостями і тому часто використовується для створення текстур, ландшафтів, хмар тощо. По суті, це тип градієнтного шуму, розроблений Кеном Перліном у 1985 році [20]. Цей алгоритм зазвичай використовується для генерування рельєфу природного вигляду, наприклад гір, печер і річок. *L-система* (або Лінденмейєр-система) є формальною граматичною системою для генерування фракталів та інших самоподібних шаблонів, які можна використовувати для створення складних, деталізованих середовищ, як-от: міста, ліси, підземелля тощо. Математичні методи L-системи через повторювані операції



переписування здатні створювати складні структури рослин, дерев чи то візерунків. *Стільникові автомати* є обчислювальною моделлю для генерування шаблонів і форм, а тому їх можна використовувати для створення природних рельєфів, води та рослинності. *Еволюційні алгоритми* використовують принципи природного відбору та генетики для створення різноманітного, складного ігрового контенту (рівні, вороги, предмети тощо). Отже, методи генерації процедурного контенту активно використовуються в ігровій розробці для автоматичного створення ландшафтів, текстур та рослинності, хоча вони не є алгоритмами ШІ, оскільки не використовують процеси навчання, адаптації чи оптимізації, характерні для ШІ. Завдяки потужним технікам PCG, розробники можуть робити ігри більш привабливими, інноваційними та креативними.

## ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У розробці відеоігор ШІ використовується для широкого кола завдань: від створення адаптивної поведінки персонажів і ворогів до генерації контенту та покращення досвіду гравців. Відповідно до поставлених задач можуть використовуватися різні алгоритми ШІ. Вибір конкретного алгоритму чи методу залежить від типу гри, її складності та бажаних результатів. Використання ШІ-технологій дозволяє створювати динамічні, інтерактивні, гнучкі ігрові світи, що значно підвищують якість ігрового процесу та досвід гравця. У платформах та шутерах алгоритми ШІ можуть допомогти персонажам визначати місцезнаходження ворогів або предметів, а також допомагати гравцям у розпізнаванні важливих об'єктів в ігровому середовищі.

Алгоритми ШІ дозволяють автоматизувати й удосконалювати процес створення контенту, розробляти динамічні, персоналізовані ігри, а також тестувати їх. Використання ШІ не лише зменшує час і витрати на розробку, а й покращує досвід гравців, роблячи ігри більш захоплюючими і реалістичними. Створення адаптивних світів із живим оточенням за простим запитом може стати майбутнім для відеоігрової індустрії. Сучасні дослідження та проєкти показують перспективність цього напрямку розвитку технологій ШІ. Але, попри численні переваги, при використанні ШІ потрібно враховувати етичні проблеми, спричинені їх використанням. Подальші дослідження в цій галузі допоможуть уникнути подібних проблем, при цьому не зменшуючи потенціал ШІ.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Global Games Market Report 2024. (2024). Newzoo. [https://best-of-gaming.be/wp-content/uploads/2024/09/2024\\_Newzoo\\_Global\\_Games\\_Market\\_Report.pdf](https://best-of-gaming.be/wp-content/uploads/2024/09/2024_Newzoo_Global_Games_Market_Report.pdf)
2. Knight, W. (2023). *Google's Gemini Is the Real Start of the Generative AI Boom*. <https://www.wired.com/story/google-gemini-generative-ai-boom/>
3. Shabinskiy, A. (2013). Ontologies, probabilistic topic models, and topic maps in ontology-driven information systems. *NaUKMA Research Papers. Computer Science*, 151, 60–66.
4. Skidan, V. (2024). Domestic segment of gamedev: history of development and current state. *Digital Economy and Economic Security*, 3(12), 76–80. <https://doi.org/10.32782/dees.12-13>
5. Alotaibi, O. (2024). New Metaverse Games Based on Artificial Intelligence: A Review. *International Journal of Metaverse*, 2, 20–32. <https://doi.org/10.54536/ijm.v2i1.3230>
6. Ling, L., Chen, X., Wen, R., Li, T., & Li, 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>
7. 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>





8. Humble, N. (2024). AI Supported Game Development: Exploring Opportunities and Challenges. *2<sup>nd</sup> Symposium on AI Opportunities and Challenges (SAIOC)*. <https://doi.org/10.13140/RG.2.2.30170.76480>
9. Vesseron, N., Redko, I., & Laclau, Ch. (2020). Deep Neural Networks are Congestion Games: from Loss Landscape to Wardrop Equilibrium and Beyond. *arXiv*. <https://doi.org/10.48550/arXiv.2010.11024>
10. Aydin, A. & Surer, E. (2020). Using Generative Adversarial Nets on Atari Games for Feature Extraction in Deep Reinforcement Learning. *arXiv*. <https://doi.org/10.48550/arXiv.2004.02762>
11. Trofymenko, O. & Bondarenko, M. (2024). Artificial intelligence in video game development. *Modern technologies in power engineering, electromechanics, control systems and mechanical engineering: 7 All-Ukrainian scientific and practical conference*.
12. Krepchenko, A. (2023). *AI in Gamedev*. Medium <https://medium.com/@anna.krepchenko/ai-in-gamedev-bc4be34ae55e>
13. Gupta, P. (2024). *AI in Gaming: Creating Adaptive and Intelligent Game Environments*. <https://fpgainsights.com/artificial-intelligence/ai-in-gaming-creating-adaptive-and-intelligent/>
14. Reed V. (2024). *Real-Time Translation in Gaming: Cross-Cultural Multiplayer Experiences*. AI Proficiency Hub #AICompetence.org. <https://aicompetence.org/real-time-translation-in-gaming/>
15. Lung, R. I. & Suciu, M. A. (2023). A Game Theoretic Decision Tree for Binary Classification. *Lecture Notes in Computer Science, 14091*, 29–41. [https://doi.org/10.1007/978-3-031-42616-2\\_3](https://doi.org/10.1007/978-3-031-42616-2_3)
16. GeeksforGeeks. (2024). *Hierarchical Task Networks (HTNs): Structure, Algorithms, and Applications in AI - GeeksforGeeks*. <https://www.geeksforgeeks.org/hierarchical-task-networks-htns-structure-algorithms-and-applications-in-ai/>
17. *Enhancing Game Immersion: The Role of Computer Vision in Gaming in 2023*. (2023). <https://www.augmentedstartups.com/blog/enhancing-game-immersion-the-role-of-computer-vision-in-gaming-in-2023>
18. Trofymenko, O. G., Sokolov, A. V., Chygunov, P. O., Akhmametiyeva, H. V., & Atanasevych, A. O. (2024). Analysis of the role of testing and quality assurance professionals. *Collection of Scientific Papers of Admiral Makarov National University of Shipbuilding, 3(496)*, 106–113. [https://doi.org/10.15589/znp2024.3\(496\).16](https://doi.org/10.15589/znp2024.3(496).16)
19. Urwin, M. (2023). *Artificial Intelligence in Gaming (and 11 AI Games to Know)*. <https://builtin.com/artificial-intelligence/ai-games>
20. *Revolutionizing Game Development: The Power of Procedural Content Generation*. (n.d.) <https://30dayscoding.com/blog/procedural-content-generation-in-games>

**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](mailto: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](mailto:dyka.anastasiia@gmail.com)

**Yuliia Loboda**

PhD, Associate Professor,

Associate Professor at the Department of Information Technologies

National University "Odessa Law Academy", Odessa, Ukraine

ORCID ID: 0000-0001-7083-552X

[jul.loboda@gmail.com](mailto:jul.loboda@gmail.com)

**Anatoly Tolocknov**

Lecturer of the Department of Information Technologies

National University "Odessa Law Academy", Odessa, Ukraine

ORCID ID: 0000-0002-2614-2109

[arnoldovich1@gmail.com](mailto:arnoldovich1@gmail.com)

**Maksym Bondarenko**

Student

National University "Odessa Law Academy", Odessa, Ukraine

ORCID ID: 0009-0002-7325-1727

[maximbondarenko20@gmail.com](mailto:maximbondarenko20@gmail.com)

## ARTIFICIAL INTELLIGENCE IN THE GAMEDEV

**Abstract.** The article explores the issue of the possible use of artificial intelligence (AI) algorithms in the field of game development. The increasing complexity of video games (game mechanics, diverse plots, game environments) necessitates the effective use of the latest AI technologies for game development. The aim of the work is to investigate the possibility of using algorithms in game development and gameplay. The hypothesis of the study is that the use of AI in game development (GameDev) can lead to a reduction in development costs. AI can be used to generate game environment landscape textures that adapt to the actions of real characters. Dialogues of background non-player characters (NPCs) and sound effects can be generated using AI. AI can generate NPCs whose behavior can change depending on the actions of real players. This creates an exciting gaming experience for players. This approach can reduce the cost and duration of production and positively affect the diversity of the gameplay. The article reviews and systematizes AI algorithms that can be used in game development, provides examples, and discusses the specifics and advantages of their use. For this purpose, a research methodology was used that combined the analysis of literary sources and conducting research based on information about the implementation of AI in GameDev. AI algorithms have great potential for effective implementation in GameDev, because their implementation helps to diversify the gameplay, provides a unique gaming experience for each player. The player receives significant freedom of action, various possible dynamically changing scenarios and dialogues with NPCs. AI can not only respond to the player's actions, but also change the environment to make the gameplay interesting and exciting for the player, regardless of the level of his progress in the game. The implementation of AI technologies in GameDev mostly shows good efficiency, but they require high qualifications of developers and consideration of risks in the process of its implementation.

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



## REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Global Games Market Report 2024. (2024). Newzoo. [https://best-of-gaming.be/wp-content/uploads/2024/09/2024\\_Newzoo\\_Global\\_Games\\_Market\\_Report.pdf](https://best-of-gaming.be/wp-content/uploads/2024/09/2024_Newzoo_Global_Games_Market_Report.pdf)
2. Knight, W. (2023). *Google's Gemini Is the Real Start of the Generative AI Boom*. <https://www.wired.com/story/google-gemini-generative-ai-boom/>
3. Shabinskiy, A. (2013). Ontologies, probabilistic topic models, and topic maps in ontology-driven information systems. *NaUKMA Research Papers. Computer Science*, 151, 60–66.
4. Skidan, V. (2024). Domestic segment of gamedev: history of development and current state. *Digital Economy and Economic Security*, 3(12), 76–80. <https://doi.org/10.32782/dees.12-13>
5. Alotaibi, O. (2024). New Metaverse Games Based on Artificial Intelligence: A Review. *International Journal of Metaverse*, 2, 20–32. <https://doi.org/10.54536/ijm.v2i1.3230>
6. 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>
7. 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>
8. Humble, N. (2024). AI Supported Game Development: Exploring Opportunities and Challenges. *2<sup>nd</sup> Symposium on AI Opportunities and Challenges (SAIOC)*. <https://doi.org/10.13140/RG.2.2.30170.76480>
9. Vesseron, N., Redko, I., & Laclau, Ch. (2020). Deep Neural Networks are Congestion Games: from Loss Landscape to Wardrop Equilibrium and Beyond. *arXiv*. <https://doi.org/10.48550/arXiv.2010.11024>
10. Aydın, A. & Surer, E. (2020). Using Generative Adversarial Nets on Atari Games for Feature Extraction in Deep Reinforcement Learning. *arXiv*. <https://doi.org/10.48550/arXiv.2004.02762>
11. Trofymenko, O. & Bondarenko, M. (2024). Artificial intelligence in video game development. *Modern technologies in power engineering, electromechanics, control systems and mechanical engineering: 7 All-Ukrainian scientific and practical conference*.
12. Krepchenko, A. (2023). *AI in Gamedev*. Medium <https://medium.com/@anna.krepchenko/ai-in-gamedev-bc4be34ae55e>
13. Gupta, P. (2024). *AI in Gaming: Creating Adaptive and Intelligent Game Environments*. <https://fpgainsights.com/artificial-intelligence/ai-in-gaming-creating-adaptive-and-intelligent/>
14. Reed V. (2024). *Real-Time Translation in Gaming: Cross-Cultural Multiplayer Experiences*. AI Proficiency Hub #AICompetence.org. <https://aicompetence.org/real-time-translation-in-gaming/>
15. Lung, R. I. & Suciu, M. A. (2023). A Game Theoretic Decision Tree for Binary Classification. *Lecture Notes in Computer Science*, 14091, 29–41. [https://doi.org/10.1007/978-3-031-42616-2\\_3](https://doi.org/10.1007/978-3-031-42616-2_3)
16. GeeksforGeeks. (2024). *Hierarchical Task Networks (HTNs): Structure, Algorithms, and Applications in AI* - GeeksforGeeks. <https://www.geeksforgeeks.org/hierarchical-task-networks-htns-structure-algorithms-and-applications-in-ai/>
17. *Enhancing Game Immersion: The Role of Computer Vision in Gaming in 2023*. (2023). <https://www.augmentedstartups.com/blog/enhancing-game-immersion-the-role-of-computer-vision-in-gaming-in-2023>
18. Trofymenko, O. G., Sokolov, A. V., Chygunov, P. O., Akhmametieva, H. V., & Atanasevych, A. O. (2024). Analysis of the role of testing and quality assurance professionals. *Collection of Scientific Papers of Admiral Makarov National University of Shipbuilding*, 3(496), 106–113. [https://doi.org/10.15589/zn2024.3\(496\).16](https://doi.org/10.15589/zn2024.3(496).16)
19. Urwin, M. (2023). *Artificial Intelligence in Gaming (and 11 AI Games to Know)*. <https://builtin.com/artificial-intelligence/ai-games>
20. *Revolutionizing Game Development: The Power of Procedural Content Generation*. (n.d.) <https://30dayscoding.com/blog/procedural-content-generation-in-games>

