



Л. М. Журавчак, В. В. Кашевка

Національний університет "Львівська політехніка", м. Львів, Україна

ІНТЕЛЕКТУАЛЬНА КОМП'ЮТЕРНА РОЛЬОВА ГРА НА ІГРОВОМУ РУШІІ UNREAL ENGINE 5

Розроблено новий підхід до застосування штучного інтелекту у комп'ютерних рольових іграх, що базується на використанні засобів рушія Unreal Engine 5 та модульної архітектури для адаптації поведінки неігрових персонажів до різних ігрових сценаріїв. Основною інновацією є поєднання дерев рішень з технологією візуального програмування Blueprint, що забезпечує динамічне управління неігровими персонажами, враховуючи їхні характеристики та тип зброї. Впроваджено GAS (англ. *Gameplay Ability System* – система ігрових можливостей) для гнучкого використання різноманітних здібностей NPC (англ. *Non-Player Character* – неігровий персонаж), що дає змогу їм адаптувати свою поведінку в реальному часі під час змін умов у грі. Це забезпечує значну інтерактивність та підвищує реалістичність ігрового процесу. Для підвищення продуктивності система налаштована під рушія Unreal Engine 5, що забезпечує стабільну роботу навіть в умовах великих ігрових світів та складних сценаріїв бою. Інтеграція системи вмінь персонажа з деревом рішень дає змогу неігровому персонажу використовувати різноманітні тактичні комбінації здібностей та дій залежно від позиції гравця, власного типу та інших чинників. Наприклад, неігрові персонажі можуть вибирати найкращі місця для атаки або захисту та адаптувати свої дії на підставі аналізу оточення. Проведено експериментальне тестування для перевірки ефективності роботи системи у різних ігрових сценаріях. Протестовано динамічну адаптацію неігрових персонажів до змінних умов гри, зокрема їхню реакцію на зміну зброї, позицій гравців та взаємодію між собою. Результати тестування продемонстрували високу ефективність системи, яка забезпечує плавну роботу неігрових персонажів без значних затримок навіть за умов складної ігрової динаміки. Запропонований підхід дає змогу істотно спростити процес розроблення складних ігрових систем завдяки модульній архітектурі та можливості легкої інтеграції штучного інтелекту в інші проекти. Це відкриває перспективи для подальших досліджень, зокрема, у сфері розширення моделей поведінки неігрового персонажа, інтеграції нових тактичних елементів та підтримки складніших сценаріїв, таких як командна взаємодія неігрових персонажів або їхнє спільне виконання завдань у кооперативному режимі ігри.

Ключові слова: ігровий рушія; дерева рішень; інструмент візуального програмування; система ігрових можливостей; інтегрована система штучного інтелекту; адаптація поведінки неігрового персонажу.

Вступ / Introduction

Проблему створення адаптивного та модульного штучного інтелекту (ШІ) у комп'ютерних рольових іграх активно досліджують у світовій науковій літературі. Значна частина досліджень зосереджена на розробленні алгоритмів штучного інтелекту, які забезпечують реалістичну поведінку ворожих неігрових персонажів та динамічний ігровий процес. Використання дерев рішень, системи Blackboard та інших підходів дає змогу створювати складніші ігрові сценарії. Однак більшість наявних рішень не враховують можливості модульної інтеграції ШІ та ефективного управління здібностями ворожих неігрових персонажів у різних контекстах гри.

Актуальність дослідження полягає у потребі розроблення такої системи ШІ, яка давала б змогу не тільки адаптувати поведінку ворожих неігрових персонажів до різних ситуацій у грі, але й інтегрувати ці рішення в інші проекти, цим самим економлячи час і ресурси розробників.

Основним напрямом дослідження є створення модульного ШІ, який можна легко інтегрувати в будь-яку іншу систему, забезпечуючи при цьому динамічну зміну поведінки ворожих неігрових персонажів.

Розроблення методів і засобів ШІ для комп'ютерних ігор є однією з найбільш досліджуваних тем у сучасній науці та технологіях. Велику увагу дослідників привертають питання створення адаптивних і реалістичних неігрових персонажів, які можуть динамічно реагувати на дії гравця. Однак, незважаючи на значні успіхи в цій галузі, низка питань залишається невирішеною, що вказує на актуальність цього дослідження. Запропонований у цій роботі підхід забезпечує створення більш гнучких й адаптивних ігрових систем, які можна буде легко інтегрувати в інші проекти, цим самим підвищуючи ефективність процесу розроблення ігор.

Об'єкт дослідження – розроблення інтелектуальних комп'ютерних рольових ігор.

Предмет дослідження – методи та засоби реалізації

Інформація про авторів:

Журавчак Любов Михайлівна, д-р техн. наук, професор, кафедра програмного забезпечення.

Email: liubov.m.zhuravchak@lpnu.ua; <https://doi.org/0000-0002-1444-5882>

Кашевка Владислав Васильович, магістр, кафедра програмного забезпечення. Email: regulus.cor.leonis.v@gmail.com;

<https://doi.org/0009-0009-7744-0462>

Цитування за ДСТУ: Журавчак Л. М., Кашевка В. В. Інтелектуальна комп'ютерна рольова гра на ігровому рушії Unreal Engine 5.

Науковий вісник НЛТУ України. 2024, т. 34, № 8. С. 120–128.

Citation APA: Zhuravchak, L. M., & Kashevka, V. V. (2024). Intelligent computer role-playing game on the Unreal Engine 5. *Scientific Bulletin of UNFU*, 34(8), 120–128. <https://doi.org/10.36930/40340814>

адаптивної та реалістичної поведінки неігрових персонажів в інтелектуальних комп'ютерних рольових іграх, що покращить ігровий досвід створення складних сценаріїв взаємодії з гравцями.

Мета роботи – розробити інтелектуальну систему для організації комп'ютерних рольових ігор із використанням рушія Unreal Engine 5, яка дасть змогу створювати гнучких й адаптивних ворожих неігрових персонажів.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

- розробити дерева рішень для ворожих неігрових персонажів з урахуванням їхньої зброї та типів, що дасть змогу створювати реалістичніші сценарії поведінки, які випливають з унікальних характеристик кожного ворожого неігрового персонажа, забезпечуючи гравцям більш захопливий ігровий досвід;
- інтегрувати систему ігрових можливостей із штучним інтелектом для динамічного управління здібностями ворожих неігрових персонажів, що забезпечить адаптивну зміну їхньої поведінки залежно від контексту гри;
- налаштувати роботу системи під рушієм Unreal Engine 5, яка уможливить підвищення її продуктивності навіть у складних ігрових сценаріях, забезпечуючи при цьому стабільність і плавність ігрового процесу;
- протестувати та впровадити розроблений модуль у реальні ігрові сценарії, що дасть змогу перевірити ефективність і надійність системи в різних типах ігор, отримати зворотний зв'язок від розробників і гравців.

Аналіз останніх досліджень та публікацій. Дослідження в галузі ШІ для комп'ютерних ігор активно зосереджуються на впровадженні різних алгоритмів, таких як дерева рішень, машини станів та нейронні мережі. Зокрема, дерева рішень набули значної уваги завдяки їхній здатності моделювати складні сценарії поведінки для неігрових персонажів, що підвищує реалістичність ігрового процесу. Наприклад, у роботі [8] розглянуто переваги дерев рішень над машинами станів, зокрема, щодо їхньої модульності та зручності використання в різних контекстах як в іграх, так і в робототехніці. Дослідження [7] порівнює ці два підходи і вказує на те, що дерева рішень легше підтримувати та адаптувати за зростання складності завдань, що є важливою перевагою для розробників ігор, які працюють з масштабними проектами. Проте в цих роботах не досліджено питання інтеграції таких систем у ширші модульні архітектури, що обмежує можливість їх використання в різних ігрових проектах.

Робота [13] є важливим ресурсом для розробників, які створюють ШІ у комп'ютерних іграх. У ній детально описано, як дерева рішень можуть бути використані для моделювання різних поведінкових сценаріїв неігрових персонажів. Автор вказує на те, що дерева рішень забезпечують гнучкість і модульність, даючи можливість розробникам легко адаптувати поведінку персонажів до різних ігрових ситуацій. Однією з ключових переваг дерев рішень, зазначених у роботі, є їхня здатність до інтеграції з іншими технологіями, що дає змогу створювати складніші ігрові сценарії. Зокрема, автор вказує на важливість використання дерев рішень у поєднанні з іншими системами для покращення управління здібностями персонажів. Це особливо актуально для ігор, де неігрові персонажі повинні адаптувати свої дії відповідно до змінних умов у грі, таких як бойові ситуації або взаємодія з іншими персонажами. Однак, незважаючи на загальний фокус на інтеграції дерев рішень з іншими системами, автор не надає детального розгля-

ду можливостей інтеграції цих технологій із системою ігрових можливостей. Це залишає відкритим питання про те, як саме можна поєднати можливості цієї системи з деревами рішень для створення ще більш гнучких і адаптивних ігрових моделей. Подальші дослідження в цьому напрямі могли б сприяти розширенню можливостей ШІ у відеоіграх, забезпечуючи більш інтерактивний і реалістичний ігровий досвід.

Робота [6] фокусується на розробленні колаборативного інструменту для створення ШІ в іграх з використанням дерев рішень. У ній автори акцентують увагу на тому, як інтеграція дерев рішень з іншими інструментами розроблення, такими як Blueprint, може значно підвищити ефективність створення та модифікації ШІ-компонент. Окрім цього, дослідження демонструє, що поєднання дерев рішень із Blueprint забезпечує гнучкіше налаштування поведінки персонажів у реальному часі, що є критично важливим у динамічних ігрових середовищах. Завдяки використанню цього підходу команди розробників можуть швидко адаптувати ігрові сценарії до змінних умов, забезпечуючи при цьому високий рівень реалістичності та занурення гравців у ігровий процес. Особливо варто відзначити, що така система спрощує координацію між членами команди, даючи можливість їм працювати над різними особливостями ШІ одночасно, що є важливим фактором для великих ігрових проєктів.

У дослідженні [11] аналізують застосування дерев рішень у поєднанні з технологією Blueprint для створення інтелектуальних поведінкових моделей у відеоіграх. Автори детально розглядають, як ці технології працюють разом для створення складних і гнучких сценаріїв поведінки персонажів. Дослідження показує, що поєднання дерев рішень з Blueprint дає змогу розробникам створювати реалістичніші ігрові сценарії, в яких ігрові персонажі здатні адаптуватися до різноманітних ігрових умов та подій, що значно підвищує якість ігрового процесу.

Проведений аналіз досліджень показує, що значна увага у сфері розроблення ШІ для ігор зосереджена на використанні дерев рішень і їх інтеграції з іншими технологіями, такими як Blueprint. Однак, попри досягнення в цій галузі, є недостатня кількість робіт, які розглядають поєднання цих підходів із системою ігрових можливостей. Це обмежує гнучкість та адаптивність створюваних моделей ШІ, що є важливим для покращення реалістичності ігрового процесу. Подальші дослідження в цьому напрямі могли б значно розширити можливості ШІ, забезпечуючи більш інтерактивний та насичений досвід для гравців.

Отже, на підставі проведеного аналізу літературних джерел можна зробити висновок, що незважаючи на значний прогрес у розробленні ШІ для ігор, є актуальною проблема створення модульних систем, які можуть бути легко інтегровані в різні проєкти. Окрім цього, потрібно досліджувати питання оптимізації таких систем під нові версії ігрових рушіїв, зокрема Unreal Engine 5, щоб забезпечити їх ефективність та продуктивність. Це і становить основу для представленого у цій роботі дослідження.

Матеріали та методи дослідження. Для досягнення поставлених завдань використано кілька ключових інструментів і методів. Основним середовищем для розроблення ігрового ШІ став ігровий рушіє Unreal Engine

5, який забезпечив необхідні ресурси для інтеграції дерев рішень із системою Blueprint. Документацію рушія Unreal Engine 5 [16] використовували для детального вивчення можливостей цього рушія та його інструментів. Окрім цього, застосовували методи тестування поведінкових сценаріїв неігрового персонажа [18], щоб оцінити ефективність ігрових механік в умовах реальних ігрових ситуацій. Для забезпечення інтеграції та модульності рішень використано кодові приклади ігрових сценаріїв та інструменти для налаштування поведінки персонажів у Blueprint.

Результати дослідження та їх обговорення / Research results and their discussion

Використання штучного інтелекту для неігрових персонажів. У комп'ютерних іграх є кілька підходів до створення ШІ для неігрових персонажів, кожен з яких має свої переваги та недоліки. Одним із найпоширеніших і найефективніших підходів є дерева рішень. Цей підхід дає змогу моделювати складну поведінку неігрового персонажа, що реагує на різні зміни в ігровому середовищі (рис. 1). Поведінка неігрового персонажа розбивається на окремі дії, кожна з яких може бути виконана за певних умов, що дає змогу створювати гнучкі та адаптивні сценарії.

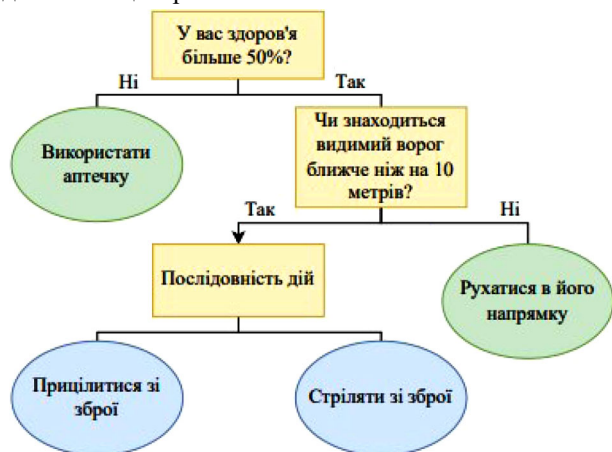


Рис. 1. Дерево рішень дій неігрового персонажа / Non-Player Character action decision tree

Дерева рішень, порівняно зі скриптовим ШІ, мають значні переваги завдяки своїй модульності та можливості легко розширювати функціонал. Це дає змогу легко змінювати або додавати нові поведінкові сценарії, зберігаючи прозорість логіки дій неігрового персонажа. Наприклад, дерево рішень може містити вузли, що відповідають за різні типи атак або ухилення, а залежно від ситуації ворожий неігровий персонаж буде обирати найоптимальніший варіант дії.

Також важливою складовою є система Blackboard, що забезпечує централізований механізм обміну інформацією між різними компонентами ШІ. Це дає змогу неігровому персонажу діяти відповідно до поточної ситуації, записуючи інформацію у цю систему чи зчитуючи її з неї. Наприклад, один компонент ШІ може відповідати за визначення ворожої цілі та записувати цю інформацію, тоді як інший використовуватиме ці дані для вибору конкретних дій неігрового персонажа. У рушії Unreal Engine 5 інтеграція системи Blackboard з деревами рішень є критичним компонентом для створення гнучкого і динамічного ШІ. Система Blackboard дає змогу різним компонентам ШІ, таким як вузли дерева

рішень, обмінюватися інформацією у реальному часі, що забезпечує адаптивність поведінки неігрового персонажа залежно від змін в ігровому середовищі.

На рис. 2 наведено приклад реалізованої інтеграції, де дерево рішень використовує систему Blackboard для пошуку найближчого гравця та переміщення до нього.

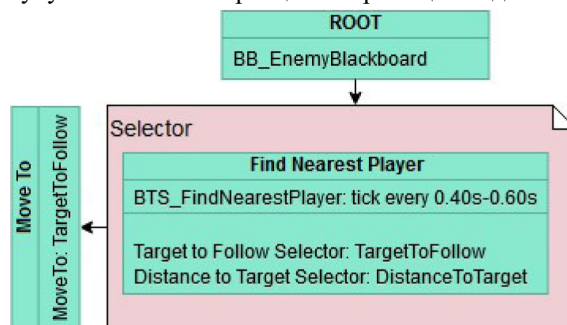


Рис. 2. Реалізація дерева рішень в ігровому рушії Unreal Engine 5 / Implementation of Behavior Trees in the Unreal Engine 5 game engine

У вузлі "Selector" (вибірка), який знаходиться під кореневим вузлом, відбувається звернення до системи Blackboard для отримання даних про найближчого гравця за допомогою спеціального вузла "Find Nearest Player". Цей вузол щосекунди перевіряє дистанцію до гравця та зберігає цю інформацію в систему Blackboard для подальшого використання іншими вузлами дерева. За допомогою вузла "Move To" неігровий персонаж виконує переміщення до найближчої цілі на підставі даних, збережених у систему Blackboard. Така інтеграція дає змогу відокремлювати логіку прийняття рішень від конкретних дій, надаючи гнучкіший контроль над поведінкою неігрового персонажа, що особливо корисно в ситуаціях, коли кілька різних компонентів ШІ повинні обмінюватися даними про поточний стан ігрового середовища.

Отже, комбінація дерева рішень із системою Blackboard у рушії Unreal Engine 5 дає змогу створювати більш інтелектуальні та адаптивні сценарії поведінки неігрового персонажа, що значно підвищує реалістичність ігрового процесу.

Використання системи атрибутів для персонажів. Система атрибутів у комп'ютерних рольових іграх дає змогу забезпечити складну взаємодію між базовими і похідними характеристиками персонажа, що безпосередньо впливає на ігровий процес. Як показано на рис. 3, первинні атрибути містять такі характеристики: сила, інтелект, витривалість і здоров'я. Від них залежать вторинні атрибути, що визначають специфічні бойові та тактичні можливості персонажа.

Первинні атрибути: Strength (сила) – збільшує фізичну шкоду; Intelligence (інтелект) – збільшує магічну шкоду; Resilience (витривалість) – збільшує броню та можливість пробивати броню; Vigor (здоров'я) – збільшує рівень здоров'я.

Вторинні атрибути: Armor (броня) – зменшує отримувану шкоду та підвищує шанс блокування атак, залежить від витривалості; Armor Penetration (проникнення броні) – ігнорує частину броні ворожого неігрового персонажа, підвищуючи шанс критичного удару, залежить від витривалості та показника проникнення броні; Block Chance (шанс блокування) – можливість блокувати вхідну шкоду наполовину, залежить від броні; Critical Hit Chance (шанс критичного удару) – шанс завдати подвійної шкоди із критичним бонусом, залежить від

показників проникнення броні та витривалості; Critical Hit Damage (шкода критичного удару) – додаткова шкода за критичного удару, залежить від показників проникнення броні та витривалості; Critical Hit Resistance (опір критичному удару) – зменшує шанс отримання критичного удару від ворожих неігрових персонажів, зале-

жить від показника витривалості; Health Regeneration (регенерація здоров'я) – визначає кількість здоров'я, що відновлюється кожену секунду, залежить від здоров'я; Mana Regeneration (регенерація магічної енергії) – визначає кількість магічної енергії, що відновлюється щосекунди, залежить від інтелекту.

Перелік атрибутів персонажа

Первинні атрибути	Вторинні атрибути
Str Збільшує фізичну шкоду	Resilience Armor Зменшує отримувану шкоду, покращує шанс блокування
Int Збільшує магічну шкоду	Resilience Armor Penetration Ігнорує відсоток броні ворога, збільшує шанс критичного удару
Res Збільшує броню та пробивання броні	Armor Block Chance Шанс зменшити вхідну шкоду наполовину
Vig Збільшує здоров'я	Armor Penetration Critical Hit Chance Шанс завдати подвійної шкоди з бонусом від критичного удару
	Armor Penetration Critical Hit Damage Додатковий бонус до шкоди при нанесенні критичного удару
	Armor Critical Hit Resistance Зменшує шанс критичного удару від атакуючих ворогів
	Vigor Health Regeneration Кількість здоров'я, що відновлюється щосекунди
	Intelligence Mana Regeneration Кількість магічної енергії, що відновлюється щосекунди

Рис. 3. Атрибути персонажа / Character attributes

Система атрибутів дає змогу розробникам створювати збалансовані і складні механіки налаштування персонажів. Використовуючи цю математичну модель, гравець може налаштовувати свого персонажа відповідно до викликів гри, адаптуючи стратегії залежно від ситуації, що сприяє глибшій взаємодії з ігровим процесом та стимулює багатократне проходження гри. Формула для обчислення значень атрибутів має такий вигляд:

$$CurAtr = C \cdot (Atr + PreV) + PostV, \quad (1)$$

де: Atr – необхідний залежний первинний або вторинний атрибут; $PreV$, $PostV$, C – коефіцієнти, що дають змогу контролювати баланс на початковій та кінцевій стадіях та для кожного окремого атрибута відповідно.

У системі розрахунку здоров'я персонажа використовується математична модель, що включає кілька ключових параметрів: показник витривалості і рівень гравця. Формула для обчислення базової величини здоров'я має такий вигляд:

$$H_{base} = C_{const} + C_{vigor} \cdot V + C_{level} \cdot L, \quad (2)$$

де: C_{const} – сталі значення здоров'я, що додається незалежно від характеристик персонажа; C_{vigor} , C_{level} – коефіцієнти, що впливають на збільшення здоров'я на підставі витривалості та рівня гравця відповідно; V – значення характеристики Vigor, яке може варіюватися залежно від характеристики персонажа; L – рівень персонажа.

Базову величину магічної енергії обчислюють за такою формулою:

$$M_{base} = C_{const} + C_{int} \cdot I + C_{level} \cdot L, \quad (3)$$

де: C_{const} – сталі значення магічної енергії, яке додається незалежно від характеристик персонажа; C_{int} , C_{level} – коефіцієнти, що впливають на збільшення магічної енергії на підставі характеристики інтелекту та рівня персонажа; I – значення характеристики інтелекту, яке залежить від персональних атрибутів персонажа; L – рівень персонажа.

Для розрахунку прогресу користувача використовується модель, яка базується на накопиченні досвіду та вартості його отримання для розблокування кожного рівня вміння. Формула враховує початкову кількість досвіду, необхідну для першого рівня, і коефіцієнт складності, який впливає на зростання вимог для кожного наступного рівня. Формула для обчислення досвіду така:

$$XP_{req}(level) = XP_{base} \cdot level^{XP_{mult}}, \quad (4)$$

де: $XP_{req}(level)$ – кількість досвіду, необхідна для досягнення певного рівня; XP_{base} – базова кількість досвіду для першого рівня; $level$ – поточний рівень; XP_{mult} – коефіцієнт, що визначає складність прогресу.

Ця формула дає змогу визначити кількість досвіду для кожного рівня. Коефіцієнт XP_{mult} регулює складність розвитку: значення більше 1 підвищує складність прокачування, тоді як значення менше 1 робить процес легшим з кожним новим рівнем. Такий підхід забезпечує баланс між викликами для гравців і доступністю розвитку персонажа, що підтримує інтерес до гри.

Реалізований підхід дає змогу адаптувати параметри персонажа до конкретних умов гри, забезпечуючи гнучкість у налаштуванні базових характеристик залежно від індивідуальних атрибутів та прогресу гравця.

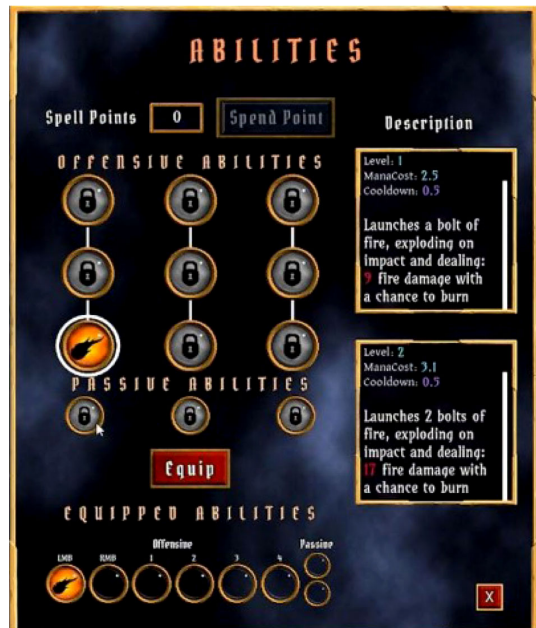


Рис. 4. Реалізація дерева навиків / Implementation of the skill tree

Використання системи навиків для персонажів. Використання системи навиків у комп'ютерних рольових іграх забезпечує логічний і структурований розвиток персонажа. Як показано на рис. 4, дерево навиків дає змогу гравцям поступово розвивати свого персона-

жа, освоюючи нові вміння, що сприяє глибшій кастомізації і стратегічному плануванню. Кожна гілка дерева навиків присвячена певній спеціалізації або стилю гри, що дає змогу гравцям обирати індивідуальний шлях розвитку.

Система також інтегрує механізм розблокування нових рівнів навиків на підставі попереднього прогресу. Це створює природний баланс ігрової кривої складності, оскільки гравці мають вивчити базові навички перед доступом до потужніших здібностей. Такий підхід полегшує розробникам процес балансування ігрових механік, даючи можливість точно визначити, які вміння будуть доступні на різних етапах гри.

Процес прогресу навиків базується на моделі накопичення досвіду, де кожен новий рівень вимагає певної

кількості досвіду. Формула для розрахунку кількості досвіду залежить від базової вартості досвіду для першого рівня і коефіцієнта складності, що визначає швидкість розвитку вмінь. Це дає змогу гнучко контролювати складність розвитку персонажа, підтримуючи баланс між складністю та доступністю нових можливостей для гравців.

На рис. 5 представлено схему поведінки ворожих неігрових персонажів, яка використовує дерево рішень для організації та адаптації їхніх дій у реальному часі, що забезпечує високий рівень динаміки та реалізму. Ключовим елементом є розподіл ворожих неігрових персонажів на дві основні категорії: далекобійні та ближні атакувальні персонажі, що дає змогу використовувати різні тактики відповідно до ігрових ситуацій.

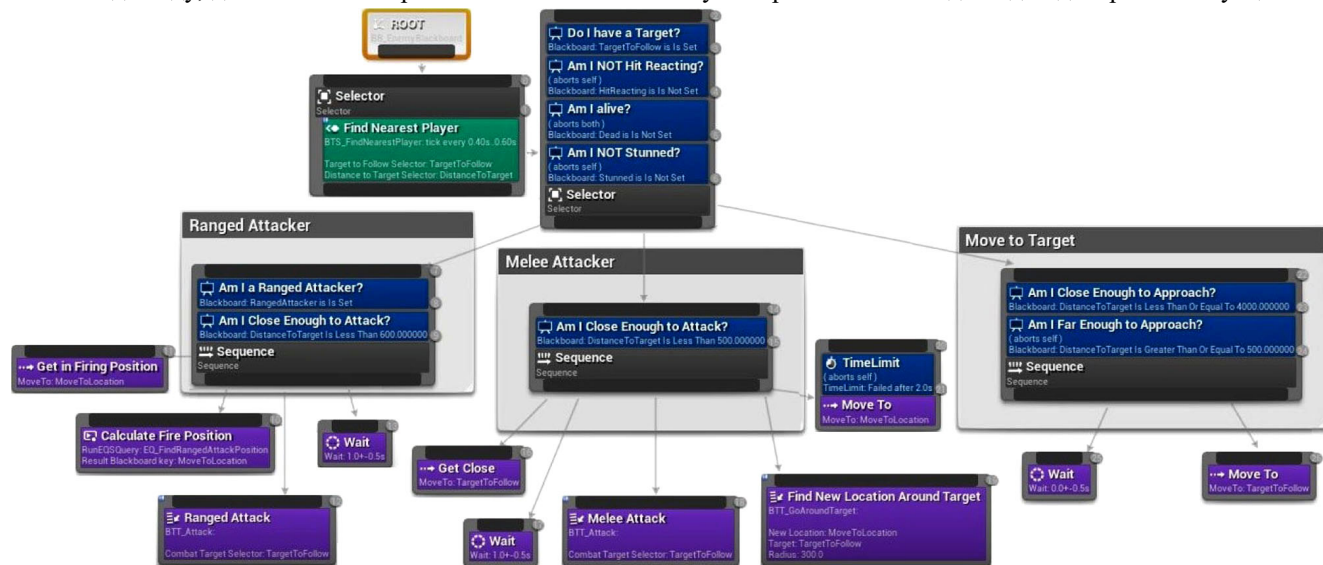


Рис. 5. Схема поведінки ворожих неігрових персонажів / Behavior scheme of enemy non-playable characters

Процес ухвалення рішень починається з вибору найближчого гравця через вузол "Find Nearest Player", який є фундаментальною частиною цієї системи. Далі для кожного типу ворожих неігрових персонажів релевантно виконуються відповідні перевірки умов, таких як наявність цілі та їхній статус (здоров'я, стан оглушення), за допомогою вузлів дерева рішень, що дає їм змогу адаптувати свої дії залежно від ігрової ситуації.

Для далекобійних атакувальних персонажів логіка побудована так, щоб перевірити, чи є персонаж далекобійним і чи знаходиться він на достатній відстані для виконання атаки. Якщо умови виконані, система ініціює обчислення позиції для атаки за допомогою спеціалізованого вузла "Calculate Fire Position". Цей процес дає змогу точно визначити оптимальну позицію для атаки, після чого виконується сама дія – стрільба по гравцеві. Ближні атакувальні персонажі діють за схожим принципом, але фокусуються на наблизенні до цілі, перш ніж виконати атаку. Вузол "Melee Attacker" контролює, чи знаходиться персонаж на достатній відстані для удару, і якщо так, ініціюється послідовність дій для наблизення й атаки.

Окрему увагу приділено блоку "Move to Target", який відповідає за переміщення ворожого неігрового персонажа до цілі в умовах, коли він перебуває на занадто великій відстані для атаки. Цей елемент дерева рішень релевантний для динамічного підлаштування позицій ворожих неігрових персонажів, даючи можливість їм ефективно наблизитися або змінювати позицію для найкращої атаки.

Інтеграція дерева рішень із системою ігрових можливостей. У реалізованій грі використання системи ігрових можливостей у поєднанні з деревом рішень забезпечує динамічний вибір та застосування здібностей неігрового персонажа у різних умовах ігрового процесу. Така інтеграція дає змогу неігровому персонажу адаптувати свою поведінку в реальному часі залежно від середовища, позицій гравця та інших чинників, що створює реалістичнішу ігрову взаємодію.

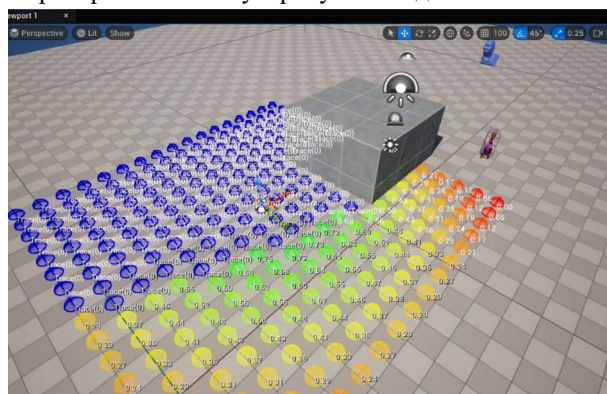


Рис. 6. Інтеграція дерева рішень із системою ігрових можливостей / Integration of Behavior Tree with the GAS system

На рис. 6 продемонстровано інтеграцію дерева рішень із системою ігрових можливостей, що дає змогу створити більш адаптивну та динамічну поведінку неігрового персонажа у процесі прийняття рішень. Цей підхід використовує можливості дерева рішень для аналізу

ігрового середовища та вибору найкращих дій залежно від контексту, а також здібності (атрибути і навички), надані системою ігрових можливостей.

Інтеграція дерева рішень із системою ігрових можливостей ґрунтується на аналізі навколишнього середовища, що реалізується через процес трасування (trace), зображений на рис. 6 у вигляді числових значень. Ці значення відображають вагомість певних точок на ігровому полі, що визначає найкращі позиції для виконання дій неігрового персонажа (атаки чи захисту). Дерево рішень обробляє ці дані, аналізуючи відстань до об'єктів, перешкоди та можливості, що виникають під час гри.

Водночас система ігрових можливостей надає неігровому персонажу можливість динамічно вибирати й активувати відповідні здібності, ґрунтуючись на результатах, які формує дерево рішень. Це забезпечує ефективну інтеграцію рішень щодо місця та часу використання певних здібностей у контексті конкретної ігрової ситуації. Наприклад, якщо дерево рішень визначає найбільш релевантне місце для атаки, система ігрових можливостей активує відповідну здібність неігрового персонажа, покращуючи її виконання на підставі аналізу оточення.

Тестування та впровадження розробленого модуля в ігрові сценарії. У тестуванні розробленого модуля інтеграції дерева рішень із системою ігрових можливостей використовували різні методи для забезпечення його ефективності у реальних ігрових сценаріях. Як показано на рис. 7, ігровий процес включав динамічну взаємодію ігрового персонажа з навколишнім середовищем і ворожими неігровими персонажами.



Рис. 7. Взаємодія персонажа з навколишнім середовищем і ворожими неігровими персонажами / Character interaction with the environment and enemy Non-Player Characters

Тестування було зосереджене на перевірці адаптивності поведінки неігрового персонажа до змін у геймплеї, а також на коректній роботі системи в умовах підвищеного навантаження на обчислювальні ресурси. Методологія тестування містила кілька підходів, серед яких:

- функціональне тестування, яке забезпечувало перевірку правильності виконання поведінкових сценаріїв неігрового персонажа у різних умовах бою. Досліджено, як неігрові персонажі реагують на дії гравця, вибирають відповідні атаки або відступають за потреби. Ці сценарії дали змогу виявити можливі недоліки у прийнятті рішень неігровими персонажами;
- стрес-тестування, яке оцінювало продуктивність системи за великої кількості одночасних об'єктів, що взаємодіють. Тут перевірялася стабільність системи за високих навантажень, що є важливим показником для збереження плавності ігрового процесу;
- аналіз на відповідність ігровим умовам, який полягав порівняно реальних ігрових дій з очікуваними сценарі-

ями, прописаними у дереві рішень. Це дало змогу переконатися, що всі можливі сценарії реалізовано коректно, і неігрові персонажі діють згідно з поведінковими шаблонами, визначеними системою.

Унаслідок тестування було отримано такі основні метрики, як швидкість відповіді на дії гравця – 0,1-0,5 с; швидкість реакції ШІ – 100-360 мс; тривалість інтеграції модулів в інші системи – 5-10 хв; кількість тактичних рішень ШІ – 40.

Отримані результати засвідчили, що модуль інтеграції дерева рішень із системою ігрових можливостей успішно адаптується до різних ігрових умов, забезпечуючи неігровому персонажу можливість ефективного використання здібностей гравця та прийняття відповідних рішень у реальному часі. Такий підхід підвищує реалістичність взаємодії неігрового персонажа із гравцем, знижуючи ризик шаблонних або передбачуваних дій, що, водночас, покращує загальний ігровий досвід.

Проведене тестування підтвердило надійність розробленого модуля, а його впровадження в реальні сценарії продемонструвало високий рівень адаптивності та відповідності вимогам сучасних ігрових систем.

Обговорення результатів дослідження. У роботі [9] досліджено інтеграцію дерев рішень зі MARL (англ. *Multi-Agent Reinforcement Learning* – метод багатоагентного навчання з підкріпленням). Автори пропонують інноваційний підхід для вирішення складних завдань в іграх, розбиваючи їх на підзадачі за допомогою дерев рішень, що потім вирішуються MARL-алгоритмами. Основною перевагою такого підходу є те, що він дає змогу уникнути потреби вирішувати складні проблеми безпосередньо за допомогою методу MARL, що зазвичай є надзвичайно ресурсомістким. В експериментах, проведених у рамках гри Google Research Football, застосування цього підходу дало змогу досягти покращення показників вигравів на 11,507 % у певних сценаріях. На відміну від цієї роботи, де значну увагу приділено інтеграції системи ігрових можливостей з деревом рішень, запропонований нами підхід використовує інші методи управління здібностями персонажів, що дає більше гнучкості у визначенні тактичних дій неігрового персонажа, а програмна система краще масштабується до багатокористувацьких ігрових середовищ, що дає змогу використовувати її в умовах більшого навантаження.

Робота [5] досліджує використання дерев рішень для моделювання людської поведінки, зокрема у контексті бойових дій для комп'ютерно згенерованих сил у системах військової симуляції. Основна мета дослідження полягає у розробленні бібліотеки моделей поведінки для механізованих піхотних підрозділів з використанням дерев поведінки в новому ШІ-фреймворку VBS Control. Автори наводять переваги дерев поведінки, зокрема їх модульність, реактивність та можливість комбонування, що робить їх зручними для створення складних моделей поведінки на рівні підрозділів і окремих солдатів. Дослідження демонструє потенціал дерев рішень для підвищення точності тактичного ШІ у симуляціях бойових дій, а також можливості для співпраці між країнами НАТО у створенні спільної бібліотеки бойових моделей. Запропонований нами підхід також використовує дерева рішень для моделювання поведінки неігрового персонажа, однак фокус більше спрямований на динамічне управління здібностями персонажів у контексті ігрового процесу з використанням системи

ігрових можливостей, він є гнучкішим у реалізації складних ігрових сценаріїв завдяки інтеграції дерев рішень із системою ігрових можливостей, що дає змогу неігровому персонажу не тільки реагувати на поточну ситуацію, але й активно використовувати різні комбінації тактичних здібностей.

У дослідженні [1] використано дерева рішення для симуляції поведінки велосипедистів у віртуальному середовищі. Автори поставили за мету створити автономного агента для управління моделлю велосипедиста у симуляторі, розробленому на підставі рушія Unreal Engine 4. Основну увагу приділяють моделюванню людської поведінки велосипедистів у різних дорожніх умовах та сценаріях. Тести продемонстрували успішну взаємодію агентів із середовищем, зокрема у складних трафікових ситуаціях. Наше дослідження також застосовує дерева рішень для моделювання поведінки персонажів, але основна відмінність полягає у використанні іншого контексту – відеогри, з акцентом на бойові сценарії та управління здібностями персонажів за допомогою системи ігрових можливостей.

У роботі [12] досліджують розроблення гнучкої системи ШІ для гри на підставі рушія Unity, з фокусом на створення ШІ для неігрових персонажів у грі Northbound. Основна мета дослідження – інтеграція інструментів для створення поведінки неігрового персонажа з мінімальними вимогами до програмування, що дає змогу не тільки розробникам з досвідом, але й дизайнерам без глибоких знань у програмуванні створювати складні поведінкові патерни. Одним із ключових елементів стала система ідентифікаторів, що давала можливість неігровим персонажам сприймати об'єкти та реагувати на них на підставі заданих умов. Результати дослідження демонструють, що Behavior Designer значно спростив процес розроблення ШІ, проте система виявилася менш стабільною у питаннях пересування та оптимізації під час роботи з великими сценаріями поведінки. Наш підхід дає змогу динамічно керувати здібностями неігрових персонажів у бойових ситуаціях та передбачає гнучке управління їхніми тактичними діями через систему ігрових можливостей, що забезпечує створення складніших сценаріїв поведінки.

Також варто згадати роботу [3], де досліджують використання дерев рішень в ігрових системах, зокрема, акцентується на проблемах, які виникають в разі їхнього неправильного застосування в сучасних ігрових рушіях. Автор вказує на те, що через свою уявну простоту та легкість використання дерева рішення часто неправильно інтегруються у процес прийняття рішень неігровим персонажем, адже вони мають відігравати значну роль саме в шарі виконання. Автор також пропонує інші підходи для покращення ефективності дерев рішень, наприклад, використання моніторингових вузлів для відстеження подій у середовищі, а також підхід до створення невеликих, ізольованих дій, що спрощує тестування та оптимізацію поведінки агентів. У нашому дослідженні використання деревоподібної структури зорієнтовано на гнучкішу інтеграцію з системою ігрових можливостей, що дає змогу ефективніше керувати поведінкою неігрового персонажа у складних бойових сценаріях, зокрема, не тільки приймати рішення, але й ефективно використовувати свої здібності на підставі контексту ігрового середовища.

У роботі [17] автори досліджують взаємодію ШІ та комп'ютерних ігор, особливо на прикладі серйозних

ігор, які використовують ШІ для поліпшення навчального процесу. Автори акцентують увагу на тому, що з моменту перших застосувань ШІ в шахових програмах, цей напрямок розвивався у багатьох сферах, враховуючи моделювання поведінки неігрового персонажа, створення реалістичних ігрових середовищ, профілювання гравців і балансування складності гри. У роботі представлено результати проєкту RAGE, що фінансувала Європейська Комісія, метою якого є створення платформи для обміну програмними компонентами ШІ, які можна використовувати у серйозних іграх, для сприяння ширшому використанню інноваційних компонент у різних проєктах. Наше дослідження також зосереджено на інтеграції нових технологій ШІ в ігрові середовища, але акцент зроблено на підвищення якості розважальних ігор через динамічне управління поведінкою неігрових персонажів та використання їх здібностей.

У роботі [2] досліджують розширення дерев рішень, які широко використовуються в індустрії відеоігор, для покращення координації неігрових персонажів у багатокористувацьких сценаріях. Основною проблемою, яку вирішує стаття, є створення більш природної та динамічної взаємодії між неігровими персонажами, які повинні співпрацювати між собою для виконання певних завдань. Для цього автори пропонують три нові типи вузлів координації, які дають можливість неігровим персонажам обмінюватися інформацією та узгоджувати свої дії через протокол запитів. Як приклад, автори демонструють реалізацію сценарію, де неігрові персонажі-пожежники координують свої дії для гасіння вогню, спричиненого гравцем, використовуючи запропоновані вузли координації. Вузли координації, запропоновані в цій роботі, могли б доповнити наш підхід, особливо в багатокористувацьких іграх, де неігрові персонажі можуть взаємодіяти із гравцями та один з одним для досягнення спільних цілей. Проте наш підхід забезпечує більше можливостей для індивідуальної оптимізації дій неігрових персонажів на підставі їхніх здібностей і контексту ігрового середовища, що дає їм змогу приймати рішення не тільки щодо взаємодії з іншими персонажами, але й щодо використання своїх бойових вмінь.

У роботі [4] досліджено особливості культивування етичних ігрових настроїв через матеріальність ігрового процесу в Ілюмінати. Автори досліджують, як правила та механіки діють як атрактори або стабілізуючі кінцеві точки в множині траєкторій у просторі можливостей, в іграх, які формують диспозиції через звички матеріальної гри, яким гравці можуть протистояти або прийняти через процес фронезис і метис. Вони записали на відео процес карткової гри "Ілюмінати" та опитали чотирьох гравців до та після гри. Було виявлено, що правило дозволу шахрайства та роль гравця з незбалансованою механікою діяли як матеріальні атрактори для ігрового процесу. Однак через процес метиса та фронезис гравці протистояли схильності до шахрайства та псування і грали в гру матеріально, щоб натомість втілити аристотелівську чесноту дружелюбності.

У дослідженні [10] автори розробили метод синтезу ігрових рівнів, який враховує рівень співпраці двох гравців для завершення певного рівня гри, та оптимізує цей рівень за допомогою алгоритму марковських ланцюгів Монте-Карло та моделювання загартування, що дає змогу швидко генерувати різні варіанти рівнів з визначеними параметрами. У цій роботі дерева рішень використовують для керування віртуальними агентами,

що проходять ігрові рівні з метою оцінювання рівня співпраці. У нашому дослідженні акцент робиться на індивідуальній поведінці неігрових персонажів та їх здатності адаптувати свої дії залежно від умов середовища та доступних вмінь без урахування кооперації між агентами.

У роботі [14] досліджено інтеграцію дерев рішень та багатоагентних систем для симуляції поведінки неігрових персонажів в ігрових середовищах з відкритим світом. Основна мета дослідження полягає у створенні моделей прийняття рішень, що дають можливість неігровим персонажам демонструвати правдоподібну поведінку в реальних умовах ігрового світу. Автори розробили систему на підставі циклу OODA (англ. *Observe-Orient-Decide-Act*) з додаванням методу Utility AI для оцінювання вигоди від кожної дії неігрового персонажа, що дає змогу приймати оптимальні рішення залежно від стану навколишнього середовища та особистих характеристик агента. Це дослідження фокусується на створенні неігрових персонажів з унікальною поведінкою, а наш підхід спрямований на динамічне управління тактичними діями неігрового персонажа в бойових сценаріях. Інтеграція з Utility AI могла б доповнити запропонований підхід, особливо у випадках, де важлива оцінка вигоди від дії у контексті використання здібностей неігрового персонажа.

Отже, внаслідок виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – розроблено методіку, яка дала змогу побудувати модульну систему ШІ для комп'ютерних рольових ігор, що забезпечує можливість інтеграції в інші ігрові системи з використанням дерева рішень у поєднанні з технологією Blueprint і системою ігрових можливостей для динамічної зміни поведінки ворожих неігрових персонажів залежно від їхніх характеристик і типу зброї.

Практична значущість результатів дослідження – результати дослідження можна використати для розроблення різних модульних систем штучного інтелекту для комп'ютерних рольових ігор на рушії Unreal Engine 5, що дасть змогу ефективно використовувати обчислювальні ресурси для збільшення кількості неігрових персонажів, забезпечення гнучкого управління їхніми здібностями, адаптації до ігрових умов за допомогою інтеграції із системою ігрових можливостей, що сприятиме підвищенню продуктивності ігрового процесу та відкриє нові можливості для реалізації складних ігрових механік, сценаріїв і тактичних моделей бою.

Висновки / Conclusions

Розроблено інтелектуальну систему для організації комп'ютерних рольових ігор, яка дала змогу створити гнучкий й адаптивний ворожих неігрових персонажів із використанням рушії Unreal Engine 5. За результатами проведеного дослідження можна зробити такі основні висновки:

1. Розроблено модульну систему штучного інтелекту для комп'ютерних рольових ігор, яка дає змогу легко інтегрувати його у будь-які інші проекти. Це значно скорочує час та ресурси, необхідні для розроблення простих і середніх ворожих неігрових персонажів.
2. Використання дерев рішень у поєднанні з технологією Blueprint забезпечило динамічне управління поведінкою неігрових персонажів, даючи можливість їм адап-

туватися до різних ситуацій у грі залежно від власного типу чи характеристик.

3. Інтеграція системи ігрових можливостей із деревом рішень дозволила неігровим персонажам використовувати різні тактичні комбінації здібностей, що підвищило динамічність бою та збагатило ігровий процес.
4. Налаштування модулів під рушій Unreal Engine 5 забезпечило покращену продуктивність та стабільність системи, що дає змогу ефективно використовувати її у сучасних ігрових проектах.
5. Тестування розробленої системи показало її здатність успішно працювати в реальних ігрових сценаріях, забезпечуючи гнучке управління поведінкою ворожих неігрових персонажів та покращення ігрового процесу.

References

1. Abrahamsen, R. N. (2019). Simulating Cyclists in a Simulator with the use of Behaviour Trees (Masters thesis, NTNU). URL: <http://hdl.handle.net/11250/2406931>
2. Agis, R. A., Gottifredi, S., & Garcia, A. J. (2020). An event-driven behavior trees extension to facilitate non-player multi-agent coordination in video games. *Expert Systems with Applications*, 155 p. <https://doi.org/10.1016/j.eswa.2020.113457>
3. Anguelov, B. (2020). Behavior Trees: Breaking the Cycle of Misuse. URL: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=O-RcxM4AAAAAJ&citation_for_view=O-RcxM4AAAAAJ:qjMakFHDy7sC
4. Colby, Rebekah Shultz, & Holmes, Steve. (2022, September). Cultivating ethical gameplay dispositions through the materiality of gameplay in *Illuminati*. *Computers and Composition*, Vol. 65, article ID 102724. <https://doi.org/10.1016/j.compcom.2022.102724>
5. Evensen, P. I., Stien, H., & Bentsen, D. H. (2018). Modelling human behaviour using behaviour trees. URL: <https://www.ffi.no/en/publications-archive/modelling-human-behaviour-using-behaviour-trees>
6. Hossain, M. Y., & Zaman, L. (2023). NCCollab: collaborative behavior tree authoring in game development. *Multimedia Tools and Applications*, 82(3), 4671–4708. <https://doi.org/10.1007/s11042-022-12307-2>
7. Iovino, M., Förster, J., Falco, P., Chung, J. J., Siegart, R., & Smith, C. (2024). Comparison between Behavior Trees and Finite State Machines. <https://doi.org/10.48550/arXiv.2405.16137>
8. Iovino, M., Scukins, E., Styru, J., Ögren, P., & Smith, C. (2022). A survey of behavior trees in robotics and ai. *Robotics and Autonomous Systems*, article ID 154. <https://doi.org/10.1016/j.robot.2022.104096>
9. Li, X., Li, Y., Zhang, J., Xu, X., & Liu, D. (2024). Embedding multi-agent reinforcement learning into behavior trees with unexpected interruptions. *Complex & Intelligent Systems*, 1–10. <https://doi.org/10.1007/s40747-023-01326-7>
10. Liu, H., Choi, M., Kao, D., & Mousas, C. (2023). Synthesizing game levels for collaborative gameplay in a shared virtual environment. *ACM Transactions on Interactive Intelligent Systems*, 13(1), 1–36. <https://doi.org/10.1145/3558773>
11. Marcotte, R. K. (2017). Modelling artificial intelligence in games using mindset behavior trees. The University of Regina (Canada). URL: <https://hdl.handle.net/10294/7884>
12. Rikkonen, J. (2017). Implementing a flexible artificial intelligence system for a video game: Case Northbound. URL: <https://core.ac.uk/outputs/161419157/?source=oai>
13. Sekhvat, Y. A. (2017). Behavior trees for computer games. *International Journal on Artificial Intelligence Tools*, 26(02). <https://doi.org/10.1142/S0218213017300010>
14. Simonov, A., Zagarskikh, A., & Fedorov, V. (2019). Applying Behavior characteristics to decision-making process to create believable game AI. *Procedia Computer Science*, 156, 404–413. <https://doi.org/10.1016/j.procs.2019.08.222>
15. Thurler, Leonardo, Melo, Sidney, Murta, Leonardo, Kohwalter, Troy, & Clua, Esteban. (2025, January). Using provenance and replay for qualitative analysis of gameplay sessions.

- Entertainment Computing*, Vol. 52, article ID 100778. <https://doi.org/10.1016/j.entcom.2024.100778>
16. Venter, H., & Ogterop, W. (2022). Unreal Engine 5 Character Creation, Animation, and Cinematics: Create custom 3D assets and bring them to life in Unreal Engine 5 using MetaHuman, Lumen, and Nanite. Packt Publishing Ltd. URL: https://books.google.com.ua/books/about/Unreal_Engine_5_Character_Creation_Anima.html?id=F954EAAAQBAJ&redir_esc=y
 17. Westera, W., Prada, R., Mascarenhas, S., Santos, P. A., Dias, J., Guimarães, M., & Ruseti, S. (2020). Artificial intelligence moving serious gaming: Presenting reusable game AI components. *Education and Information Technologies*, 25, 351–380. <https://doi.org/10.1007/s10639-019-09968-2>
 18. Zhong, Z., Tang, Y., Zhou, Y., Neves, V. D. O., Liu, Y., & Ray, B. (2021). A survey on scenario-based testing for automated driving systems in high-fidelity simulation. <https://doi.org/10.48550/arXiv.2112.00964>

L. M. Zhuravchak, V. V. Kashevska

Lviv Polytechnic National University, Lviv, Ukraine

INTELLIGENT COMPUTER ROLE-PLAYING GAME ON THE UNREAL ENGINE 5

Develops a new approach to the application of artificial intelligence for computer role-playing games, focusing on the utilization of Unreal Engine 5 tools combined with a modular architecture to adapt non-player characters to a broad spectrum of game scenarios. The core innovation lies in integrating decision trees with Blueprint technology, enabling dynamic and context-sensitive control of NPCs. This system considers factors such as NPC attributes and weapon types, allowing for flexible and responsive AI behavior. In particular, the integration of the Gameplay Ability System enables NPCs to use a variety of abilities in real-time, adjusting their strategies and actions dynamically as game conditions evolve. This significantly enhances both the interactivity and realism of the gameplay experience. The system has been configured for Unreal Engine 5, ensuring high performance and stability, even in large, complex game worlds with intensive battle scenarios. By combining GAS with decision trees, NPCs are able to execute tactical combinations of abilities and actions, depending on variables like player positioning, custom type, and environmental factors. For example, NPCs can strategically select the best locations for attack or defense and modify their behavior based on real-time environmental analysis. To assess the systems effectiveness, experimental testing was conducted across various game scenarios, evaluating how well NPCs adapted to changing game conditions. These tests included monitoring NPC reactions to changes in weapons, player movements, and interactions with other NPCs. The results demonstrated the systems high efficiency, as NPCs were able to perform smoothly without significant delays, even in dynamic and challenging gameplay situations. This approach simplifies the development of complex game systems by offering a modular AI structure that can be easily integrated into different projects. The flexibility of this architecture not only streamlines the development process but also opens up new avenues for further research. Future investigations could focus on expanding NPC behavior models, integrating more advanced tactical elements, and supporting more complex scenarios, such as cooperative gameplay with coordinated NPC teamwork. The systems adaptability positions it as a valuable tool for both current and future AI-driven game development projects.

Keywords: game engine; decision trees; blueprint; Gameplay Ability System; integrated system of artificial intelligence; adaptation of NPC behavior.



Й. З. Піскозуб^{1,2}, Б. В. Шамановський¹

¹ Національний університет "Львівська політехніка", м. Львів, Україна

² Краківська політехніка ім. Тадеуша Костюшка, м. Краків, Польща

ЕФЕКТИВНІ МЕТОДИ ВІДОКРЕМЛЕННЯ ГОЛОСУ ВІД ШУМУ НА ПІДСТАВІ ГЛИБОКОГО НАВЧАННЯ

Досліджено та проведено порівняльний аналіз двох сучасних методів відокремлення голосу від шуму на підставі глибокого навчання: моделей, що базуються на рекурентних нейронних мережах, та моделей на підставі трансформерів. Запропоновано гібридну модель, яка увібрала кращі властивості обох моделей. Дослідження охоплює актуальну проблему покращення якості мовлення в різних сферах, таких як слухові апарати, мобільний зв'язок та автоматичне розпізнавання мовлення, де важливим є відокремлення цільової мови від фонових шумів. Розглянуто архітектуру, особливості, алгоритми, переваги та недоліки моделей LSTM (англ. *Long Short-Term Memory*), SepFormer та гібридної архітектури. Також описано складнощі, що виникають під час навчання моделей нейронних мереж. Проаналізовано останні дослідження, які підтверджують ефективність підходів відокремлення голосу від шуму з використанням досліджуваних моделей. Експериментальні результати показали, за яких умов варто використовувати ту чи іншу з розглянутих моделей, та чому розвиток гібридних моделей може мати значну цінність у дослідженнях. Розглянуто практичні особливості використання моделей, зокрема підготовку та оброблення вхідних даних, навчання моделей та оцінювання результатів. Для оцінювання результатів дослідження застосовано метрики оцінювання якості звуку, які є важливими інструментами для покращення якості звуку в різних технологічних застосуваннях. Проведено оцінювання та порівняння якості відокремленого мовлення та його розбірливості у розглянутих моделях. Описано перспективи подальшого розвитку дослідження, інтеграцію нових джерел даних та покращення методів оброблення аудіосигналів. Проведене дослідження робить значний внесок у розуміння ефективності застосування різних методів відокремлення голосу від шуму, надає рекомендації щодо вибору найпридатнішої моделі для конкретних умов, а також пропонує гібридне рішення для завдань з потребою отримати найкращий результат. Отримані результати дослідження можуть бути корисними для науковців та інженерів, які працюють у галузі оброблення сигналів та розроблення аудіосистем, сприяючи підвищенню якості та ефективності технологій оброблення мовлення.

Ключові слова: відокремлення мовлення; рекурентні нейронні мережі; трансформери; гібридна архітектура.

Вступ / Introduction

Відокремлення голосу від шуму є однією з найбільш вивчених проблем у галузі оброблення аудіосигналів і знаходить своє застосування у численних сферах, таких як слухові апарати, мобільний зв'язок, автоматичне розпізнавання мови та інших. У світовій науковій літературі ґрунтовно досліджено різні підходи до цієї проблеми, зокрема на підставі традиційних методів оброблення сигналів, а також сучасних методів глибокого навчання.

Традиційні методи відокремлення голосу продемонстрували обмежену ефективність, особливо у разі складного фонового шуму. З появою глибокого навчання, підходи на підставі рекурентних нейронних мереж RNN (англ. *Recurrent Neural Networks*), зокрема моделей LSTM (англ. *Long Short-Term Memory* – довга короткочасна пам'ять) та трансформерів SepFormer значно підвищили продуктивність у завданнях відокремлення голосу від шуму. Проте досі існують виклики, зокрема обмеженість навчальних даних, розпізнавання різних

типів шуму та обчислювальна складність моделей, які потребують подальших досліджень.

Незважаючи на значні успіхи, досягнуті в цій галузі, досі не існує універсального рішення, яке могло б ефективно впоратися із широким спектром різних умов шумного середовища. Це підтверджує актуальність дослідження нових методів та підходів для покращення якості відокремлення голосу.

Постановка завдання: це дослідження спрямоване на порівняння ефективності двох сучасних методів відокремлення голосу від шуму, заснованих на глибокому навчанні, та їх комбінації у вигляді гібридної архітектури: моделей на підставі рекурентних нейронних мереж (LSTM-моделей) та моделей трансформерів (SepFormer).

Об'єкт дослідження – відокремлення голосу від шуму в аудіосигналах.

Предмет дослідження – встановлення ефективності застосування різних моделей глибокого навчання у ві-

Інформація про авторів:

Піскозуб Йосиф Збігневич, д-р фіз.-мат. наук, професор, кафедра прикладної математики. Email: yosyf.piskozub@pk.edu.pl;
<https://orcid.org/0000-0001-7978-4052>

Шамановський Богдан Валерійович, аспірант, кафедра прикладної математики. Email: b.shamanovskiy@gmail.com;
<https://orcid.org/0009-0002-0138-7414>

Цитування за ДСТУ: Піскозуб Й. З., Шамановський Б. В. Ефективні методи відокремлення голосу від шуму на підставі глибокого навчання. Науковий вісник НЛТУ України. 2024, т. 34, № 8. С. 129–135.

Citation APA: Piskozub, Y. Z., & Shamanovskiy, B. V. (2024). Effective methods of separation of voice from noise based on deep learning. *Scientific Bulletin of UNFU*, 34(8), 129–135. <https://doi.org/10.36930/40340815>