

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук
(повна назва)
Кафедра _____ Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Дослідження можливостей використання поведінкової архітектури, керованої
особистістю, для створення індивідуальних та адаптивних супротивників на
основі штучного інтелекту для стратегічних ігор піджанру 4X
(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-22-1

_____ Роман СПАЩЕНКО
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна
Освітня програма _____ Штучний інтелект

(повна назва освітньої програми)
Керівник _____ ст. викл. Ігор БІБІЧКОВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ

(підпис)

_____ Лариса ЧАЛА
(власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Спащенку Роману Тарасовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження можливостей використання поведінкової архітектури, керованої особистістю, для створення індивідуальних та адаптивних супротивників на основі штучного інтелекту для стратегічних ігор піджанру 4X

затверджена наказом університету від 5 червня 2026 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 9 липня 2026 р.

3. Вихідні дані до роботи Об'єкт модифікації, декомпілятор і редактор .NET-збірок dnSpyEx, система збирання MSBuild у складі Visual Studio (цільова платформа .NET Framework 3.5), мова програмування C#, засоби аналізу телеметрії (Python).

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі

2) Проектування та програмна реалізація системи динамічних особистостей

3) Експериментальне дослідження та аналіз результатів

4) Напрями подальшого розвитку

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз предметної галузі	09.06.2026–13.06.2026	виконано
3	Розроблення базових архетипів	14.06.2026–15.06.2026	виконано
4	Програмна реалізація та інтеграція з грою	16.06.2026–22.06.2026	виконано
5	Налагодження, багаторівневе тестування	23.06.2026–25.06.2026	виконано
6	Проведення експериментів	25.06.2026–26.06.2026	виконано
7	Оформлення пояснювальної записки	27.06.2026–29.06.2026	виконано
8	Нормоконтроль і підготовка до захисту	29.06.2026–08.07.2026	виконано
9	Захист перед ЕК	09.07.2026	

Дата видачі завдання 8 червня 20 26 р.

Здобувач Слащенко
(підпис)

Керівник роботи _____
(підпис)

ст. викл. Ігор БІБІЧКОВ
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 116 с., 13 рис., 11 табл., 6 дод., 23 джерела.

АРХЕТИП ХАРАКТЕРУ, ДЕКОМПІЛЯЦІЯ, ДИНАМІЧНА ОСОБИСТІСТЬ, КРИВА ВІДГУКУ, МОДИФІКАЦІЯ ГРИ, НАСТРІЙ, ПЕРЕКОНЛИВІ АГЕНТИ, ПЕРСИСТЕНТНА ПАМ'ЯТЬ, ПОВЕДІНКОВА РІЗНОМАНІТНІСТЬ, ШТУЧНИЙ ІНТЕЛЕКТ, 4X, ENDLESS LEGEND, QUALITY-DIVERSITY, UTILITY-BASED AI.

Об'єктом дослідження є системи штучного інтелекту для покрокових стратегічних ігор піджанру 4X.

Предметом дослідження є методи створення ІІІ-агентів з виразними індивідуальними особистостями, здатних до динамічної адаптації поведінки на основі ігрового досвіду.

Метою роботи є розробка системи динамічних особистостей для ІІІ-опонентів у 4X стратегіях, що дозволяє створювати агентів з виразними характерами, які зберігаються та еволюціонують протягом гри.

Методи дослідження: аналіз наукової літератури; математичне моделювання функцій корисності та кривих відгуку; реверс-інжиніринг через декомпіляцію .NET збірок мовою C#; статистичний аналіз експериментальних результатів.

У роботі проведено аналіз сучасних підходів до ІІІ-опонентів у 4X іграх та виявлено їх ключові обмеження. Запропоновано тришарову архітектуру системи динамічних особистостей, що складається з архетипу характеру, настрою та пам'яті. Спроековано модель прийняття рішень на основі кривих відгуку, параметри яких визначаються комбінацією рис архетипу та поточного стану. Систему реалізовано у вигляді модифікації до гри Endless Legend.

ABSTRACT

Bachelor's thesis contains: 116 pp., 13 fig., 11 tabl., 6 ann., 23 references.

The object of research is artificial intelligence systems for turn-based strategy games of the 4X subgenre.

The subject of research is methods for creating AI agents with distinct individual personalities capable of dynamic behavioral adaptation based on game experience.

The aim of the work is to develop a dynamic personality system for AI opponents in 4X strategy games, allowing the creation of agents with expressive characters that persist and evolve throughout a game session.

Research methods: analysis of scientific literature; mathematical modeling of utility functions and response curves; reverse engineering through decompilation of .NET assemblies in C#; statistical analysis of experimental results.

The work analyzes contemporary approaches to AI opponents in 4X games and identifies their key limitations. A three-layered architecture of the dynamic personality system is proposed, consisting of a character archetype, mood, and memory. A decision-making model based on response curves is designed, where parameters are determined by a combination of archetype traits and current state. The system is implemented as a modification for the game Endless Legend.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	9
Вступ.....	10
1 Аналіз предметної галузі та постановка задачі.....	12
1.1 Опис предметної галузі	12
1.1.1 Походження терміну 4X.....	12
1.1.2 Характерні особливості геймплею.....	12
1.1.3 Ключові представники жанру.....	13
1.1.4 Роль штучного інтелекту в 4X стратегіях	15
1.2 Класифікація сучасних підходів до ІІІ в стратегічних іграх.....	16
1.2.1 Системи на основі статичних коефіцієнтів	16
1.2.2 Скінченні автомати та дерева поведінки.....	18
1.2.3 Системи на основі корисності (Utility-based AI)	19
1.2.4 Підходи на основі навчання з підкріпленням	21
1.2.5 Підходи на основі великих мовних моделей	22
1.2.6 Дослідження переконливості ігрових агентів.....	24
1.2.7 Висновки до підрозділу.....	25
1.3 Виявлені обмеження та виклики	26
1.3.1 Передбачуваність систем зі статичними коефіцієнтами.	26
1.3.2 Збіжність до МЕТА-стратегій в оптимізаційних підходах.....	27
1.3.3 Відсутність персистентної пам'яті про події гри.	27
1.3.4 Відсутність динаміки емоційного стану.....	28
1.3.5 Обчислювальні обмеження сучасних методів.	29
1.3.6 Висновки до підрозділу.....	29
1.4 Концепція запропонованого рішення	30
1.4.1 Тришарова архітектура особистості	31
1.4.2 Механізм прийняття рішень	33
1.4.3 Підсумкова характеристика підходу.....	34
1.5 Постановка задачі.....	34

2	Проектування та програмна реалізація системи динамічних особистостей	37
2.1	Обґрунтування вибору напрямку реалізації та стратегії інтеграції.....	37
2.2	Правові аспекти модифікації гри	39
2.3	Інструментальні засоби декомпіляції та рекомпіляції	41
2.4	Загальна архітектура програмної реалізації	44
2.5	Модель особистості: реалізація трьох підсистем	46
2.5.1	Статичний архетип	47
2.5.2	Динамічний настрій.....	48
2.5.3	Персистентна пам'ять.....	49
2.6	Математична модель прийняття рішень.....	51
2.7	Інтеграція з ІІІ гри: точки вбудовування та забезпечення надійності	58
2.8	Конфігурація та підсистема інструментування	61
2.9	Висновки до розділу	62
3	Експериментальне дослідження та аналіз результатів.....	64
3.1	Умови та параметри проведення експерименту	64
3.2	Метрики оцінювання та очікувані напрями відмінностей	65
3.3	Перевірка коректності та надійності реалізації	66
3.4	Поведінкова різноманітність архетипів.....	68
3.5	Динаміка настрою та пам'яті: приклад партії.....	72
3.6	Життєздатність архетипів	74
3.7	Висновки до розділу	74
4	Напрями подальшого розвитку.....	76
4.1	Автоматизована генерація особистостей за допомогою алгоритму «якість – різноманіття».....	76
4.1.1	Передумови та мотивація.....	76
4.1.2	Сутність підходу «якість – різноманіття»	77
4.1.3	Запропонована схема пошуку.....	77
4.1.4	Очікуваний результат і виклики.....	79

4.2 Відображення настрою ШІ-гравців для гравця	80
4.3 Інші напрями розвитку	81
4.4 Висновки до розділу	82
Висновки	83
Перелік джерел посилання	85
Додаток А Вихідний код класу PersonalitySnapshot (поєднання трьох шарів моделі)	88
Додаток Б Вихідний код класу Mood (динамічний настрій)	95
Додаток В Вихідний код класу Memory (персистентна пам'ять подій)....	102
Додаток Г Вихідний код класу ResponseCurves (криві відгуку)	111
Додаток Д Вихідний код класу WarUtility (корисність оголошення війни)	113
Додаток Е Відомість кваліфікаційної роботи	116

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЄС – Європейський Союз;

AI – Artificial Intelligence – штучний інтелект;

API – Application Programming Interface – програмний інтерфейс;

CIL – Common Intermediate Language – спільна проміжна мова;

CLI – Common Language Infrastructure – спільна інфраструктура мови;
стандарт ECMA-335;

CSV – Comma-Separated Values – текстовий формат даних, розділених
комами;

EULA – End User License Agreement – ліцензійна угода з кінцевим
користувачем;

IAUS – Infinite Axis Utility System – нескінченно-осьова система
корисності;

LLM – Large Language Model – велика мовна модель;

MAP-Elites – Multi-dimensional Archive of Phenotypic Elites –
багатовимірний архів фенотипних еліт;

META – most effective tactic available – найефективніша доступна
тактика;

NPC – Non-Player Character – неігровий персонаж;

QD – Quality-Diversity – якість-різноманіття; клас еволюційних
алгоритмів;

RL – Reinforcement Learning – навчання з підкріпленням;

XML – eXtensible Markup Language – розширювана мова розмітки;

4X – eXplore, eXpand, eXploit, eXterminate – досліджуй, розширюйся,
експлуатуй, знищуй – піджанр покрокових стратегій;

.NET – програмна платформа Microsoft .NET.

ВСТУП

Сучасна індустрія відеоігор переживає період активного інтегрування методів штучного інтелекту в усі аспекти розробки. Хоча основну увагу академічних досліджень останнього десятиліття привертають проривні досягнення глибокого навчання – такі як AlphaStar для StarCraft II чи OpenAI Five для Dota 2 – комерційна реальність більшості ігор досі базується на класичних методах прийняття рішень. Це особливо помітно у жанрі 4X стратегій, де ШІ-опоненти зазвичай реалізуються через статичні таблиці коефіцієнтів та правила, що породжує проблему передбачуваності та одноманітності ігрового досвіду.

Гра проти ШІ є основним режимом гри в більшості 4X стратегій через значну тривалість партій, проте сучасний рівень якості штучних опонентів часто не відповідає очікуванням гравців. ШІ-опоненти у відомих представниках жанру, таких як серія Civilization, регулярно отримують критику за поведінку, що виглядає нелогічно, передбачувано та механічно. У відповідь розробники намагаються вдосконалити ШІ через ускладнення правил та додавання нових поведінкових модифікаторів, однак ці зусилля рідко долають фундаментальне обмеження статичних систем: однаковий стан гри породжує однакову реакцію опонента, що швидко стає очевидним для досвідченого гравця.

З іншого боку, найсучасніші підходи – навчання з підкріпленням та великі мовні моделі – теоретично здатні створювати більш складну поведінку, але мають свої власні обмеження. Оптимізаційна природа цих методів призводить до сходження агентів до домінуючих стратегій, що нівелює індивідуальні відмінності між опонентами. Окрім того, такі підходи потребують значних обчислювальних ресурсів, недоступних для одноосібних розробників та малих студій. Це створює потребу в альтернативному підході, який забезпечував би виразне поведінкове

розмаїття опонентів при збереженні обчислювальної ефективності та інтерпретованості.

Темою роботи є дослідження можливостей використання поведінкової архітектури для створення індивідуальних та адаптивних ШІ-опонентів для відеоігор у жанрі 4X стратегій. Метою роботи є розробка концепції системи динамічних особистостей, що поєднує статичний архетип характеру, динамічний настрій та персистентну пам'ять для створення опонентів з виразними індивідуальностями, які зберігають характер протягом гри та реагують на події емоційно. Концептуальною основою системи слугує поєднання підходу на основі корисності (utility-based AI), що забезпечує обчислювальну ефективність та інтерпретованість рішень, із принципами переконливих агентів – персистентною пам'яттю та емоційною динамікою, – запозиченими з відповідних досліджень. Запропонована архітектура спрямована на подолання ключових обмежень існуючих рішень – передбачуваності статичних систем, збіжності оптимізаційних підходів до домінуючих стратегій, відсутності персистентної пам'яті та емоційної динаміки, а також високих обчислювальних вимог методів на основі глибокого навчання та великих мовних моделей.

Запропоновану концепцію реалізовано у вигляді функціональної модифікації для гри Endless Legend та експериментально перевірено в ігрових симуляціях з ШІ-опонентами, що підтвердило поведінкову різноманітність і життєздатність розроблених архетипів. Практичне значення роботи полягає в тому, що запропонована архітектура не залежить від конкретних ігрових механік і може бути адаптована як для інших 4X стратегій, так і для неігрових персонажів в іграх інших жанрів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної галузі

Поняття «4X» (або «4X стратегії») використовується для позначення піджанру стратегічних відеоігор, у яких ігровий процес побудовано навколо чотирьох основних аспектів: дослідження (eXplore), розширення (eXpand), експлуатація ресурсів (eXploit) та знищення суперників (eXterminate).

1.1.1 Походження терміну 4X

Термін був запроваджений ігровим журналістом Аланом Емпіхом у вересні 1993 року в огляді гри Master of Orion для журналу Computer Gaming World. В оригінальному формулюванні Емпіх писав: «I give MOO a XXXX rating because it features the essential four X's of any good strategic conquest game: EXplore, EXpand, EXploit and EXterminate. » (переклад автора: «я даю Master of Orion рейтинг XXXX, тому що гра містить чотири основні X, притаманні будь-якій якісній стратегічній завойовницькій грі: eXplore, eXpand, eXploit та eXterminate») [1].

Хоча Master of Orion безпосередньо асоціюється з виникненням терміну, формувальним прикладом жанру є попередня гра – Sid Meier's Civilization (1991), розроблена Сідом Меєром та Брюсом Шеллі для компанії MicroProse.

1.1.2 Характерні особливості геймплею

Ігровий процес 4X стратегій характеризується кількома типовими елементами. Гравець керує цілою імперією або цивілізацією, починаючи з невеликого поселення та розвиваючи її протягом тривалого ігрового часу. На відміну від інших стратегічних піджанрів, які зосереджуються

переважно на військовому компоненті, 4X стратегії приділяють значну увагу економічному та технологічному розвитку, а також пропонують низку як військових, так і невійськових шляхів до перемоги.

Більшість 4X ігор є покроковими, проте існують і ігри з елементами реального часу, такі як Stellaris від Paradox Development Studio. Спільними для цих ігор є наступні елементи: процедурно згенеровані або обрані карти; кілька різних фракцій з унікальними особливостями та бонусами; багаторівнева система дипломатії; технологічне дерево або інша система наукових досліджень; кілька альтернативних шляхів до перемоги (наприклад, військова, наукова, культурна, дипломатична).

У академічних дослідженнях відеоігор 4X стратегії визнаються як один з усталених піджанрів стратегічних ігор, який належить до традиції, що походить від варгеймів [2].

1.1.3 Ключові представники жанру

Серія Civilization, розпочата в 1991 році, залишається одним з найбільш впливових представників жанру. До червня 2024 року сукупні продажі серії досягли 70 мільйонів копій. Найновішою на момент написання роботи є Civilization VII, випущена у 2025 році. Попередня гра серії, Civilization VI (2016), містить особливо релевантну для цієї роботи систему «агенд» – кожен III-лідер має історичну агенду, що визначає його стиль гри, та приховану агенду, яку гравець має дізнатися через шпигунство або дипломатію [3]. Приклад ігрового процесу гри, що показує карту та інтерфейс керування містом показано на рисунку 1.1.

Серед фентезійних представників жанру значним є Endless Legend (2014) від французької студії Amplitude Studios. Гра відрізняється високою асиметричністю фракцій: гра містить чотирнадцять ігравельних фракцій, кожна з яких має суттєво відмінні ігрові механіки, а не лише унікальні бонуси, як у багатьох інших представниках жанру. Вигляд ігрової

карти гри Endless Legend наведено на рисунку 1.2, на рисунку 1.3 показано вигляд екрану дипломатії.



Рисунок 1.1 – Скріншот ігрового процесу гри Civilization VI

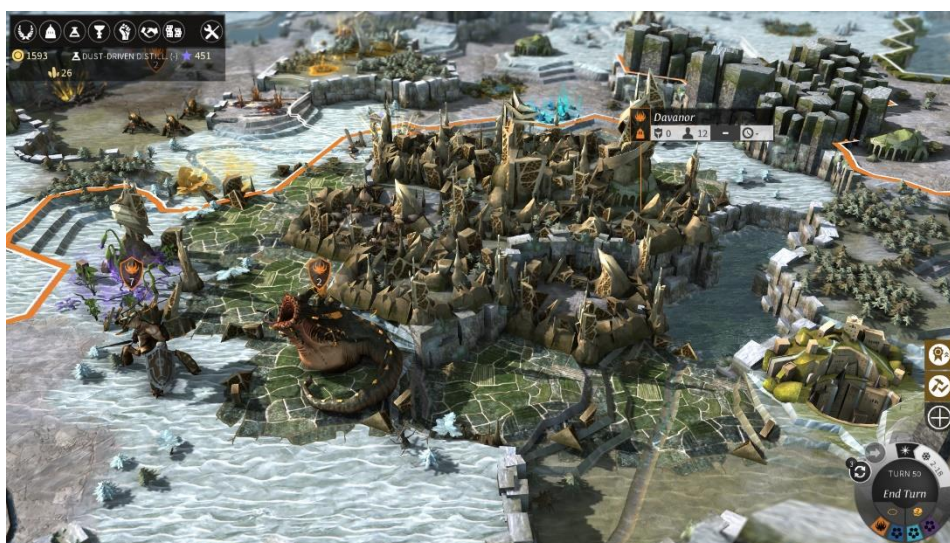


Рисунок 1.2 – Скріншот ігрового процесу гри Endless Legend

Серія Master of Orion, що дала назву жанру, заклала основу для космічних 4X стратегій. У 2010-х роках з'явилися нові впливові представники жанру, серед яких Stellaris (2016) від Paradox Development

Studio – космічна 4X стратегія з елементами Grand Strategy, що використовує реальний час з можливістю паузи.

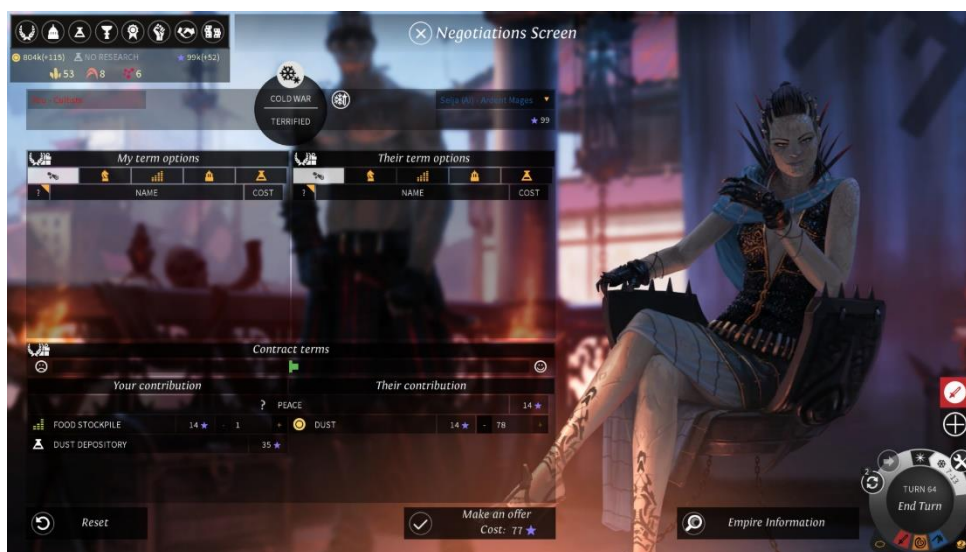


Рисунок 1.3 – Скріншот екрану дипломатії гри Endless Legend

1.1.4 Роль штучного інтелекту в 4X стратегіях

У більшості 4X ігор основним режимом гри є одиночна гра проти комп'ютерних опонентів. Civilization спочатку взагалі була випущена тільки з підтримкою одиночної гри, оскільки в 1991 році онлайн-ігри ще тільки зароджувались. Хоча сучасні представники жанру підтримують мультиплеєр, він зазвичай розглядається як додатковий, а не основний режим, через значну тривалість партій – типова гра може тривати до кількох десятків годин.

Ця обставина робить якість ШІ-опонентів критично важливим аспектом ігрового досвіду. Однак, як зазначається в індустрії, реалізація переконливого ШІ для 4X ігор є складною задачею. Так, ШІ в Civilization V критикували за поведінку, яка нагадувала «капризних, ірраціональних дітей – швидких до гніву, стійких до переговорів і з дуже малою прозорістю причин прийняття певних рішень» [4]. У відповідь на цю критику

розробники Firaxis Games під керівництвом Еда Бича запровадили систему «агенд» у Civilization VI, спробувавши надати ІІІ-лідерам більш зрозумілу та інтерпретовну мотивацію.

Слід зазначити, що, попри ці зусилля, користувацькі обговорення вказують на те, що навіть удосконалена система агенд сприймається багатьма гравцями як «штучна та поверхнева». Це свідчить про те, що проблема створення переконливих ІІІ-опонентів у 4Х стратегіях залишається відкритою, що і визначає актуальність даної роботи.

1.2 Класифікація сучасних підходів до ІІІ в стратегічних іграх

Створення штучного інтелекту для стратегічних ігор – це задача, яка протягом останніх трьох десятиліть розв'язується за допомогою принципово різних підходів: від простих систем правил зі статичними коефіцієнтами до сучасних методів глибокого навчання та використання великих мовних моделей. Кожна з цих парадигм має свої сильні та слабкі сторони, які визначають придатність методу для конкретного типу гри та конкретної мети розробника. У цьому підрозділі розглядаються основні класи методів, що історично домінували в індустрії та академічних дослідженнях, з аналізом їх застосовності до задачі створення особистісно-керованого ІІІ для 4Х стратегій.

1.2.1 Системи на основі статичних коефіцієнтів

Найпоширенішим у комерційних стратегічних іграх історично є підхід, побудований на статичних коефіцієнтах та скриптах. У цій парадигмі поведінка ІІІ-агента визначається набором заздалегідь заданих числових параметрів (ваг) та умов, які описують ставлення агента до різних аспектів гри: схильність до агресії, бажану швидкість розширення, інтерес до різних типів технологій, ставлення до інших фракцій тощо. Розробники індустрії

використовують скрипти як основну техніку реалізації ІІІ для складних ігор, оскільки вони зрозумілі, передбачувані, легкі в реалізації та можуть бути створені та змінені нерозробниками [5].

У стратегічних іграх, як-от серія Civilization, цей підхід реалізується через таблиці поведінкових ваг, прив'язаних до конкретних лідерів або цивілізацій. Кожен ІІІ-лідер має набір параметрів, які визначають його стиль гри. У Civilization VI цю систему було розширено через введення концепції «агенд» – кожен лідер отримує одну історичну агенду, відому з початку гри, та одну приховану, яку гравець може дізнатися лише через дипломатичну взаємодію або шпигунство [3]. Endless Legend використовує подібний підхід через XML-файли в директорії Public/AI/, де параметри особистостей фракцій задаються розробниками гри або модерами.

Сильні сторони цього підходу – детермінованість, передбачуваність для розробників та можливість швидкого налагодження. ІІІ працює за мілісекунди на хід, що задовольняє вимоги до продуктивності в реальному часі. Однак цей підхід має значні обмеження. По-перше, скрипти зазвичай довгі та складні, що робить їх схильними до помилок, які гравці можуть експлуатувати для легкої перемоги над сильними опонентами [5]. По-друге, через свою статичну природу скрипти не здатні справлятися з непередбачуваними тактиками гравця та не можуть масштабувати рівень складності відповідно до навичок гравця [5]. По-третє, поведінка ІІІ стає передбачуваною після кількох ігор, оскільки одні й ті самі параметри породжують одні й ті самі реакції в подібних ситуаціях.

Робота Spronck та інших запропонувала техніку «динамічного скриптингу» (dynamic scripting) як спосіб подолати ці обмеження. У динамічному скриптингу ваги правил оновлюються онлайн залежно від результатів попередніх зіткнень: правила, які призвели до успіху, отримують збільшення ваги, а ті, що призвели до невдачі, – зменшення. Експерименти в комерційній рольовій грі Neverwinter Nights показали, що ця техніка дозволяє ІІІ адаптуватись до тактики гравця, зберігаючи при

цьому інтерпретованість поведінки [5]. Проте динамічний скриптинг залишається фундаментально оптимізаційним підходом, який сходиться до домінуючих стратегій, а не зберігає поведінкове розмаїття.

1.2.2 Скінченні автомати та дерева поведінки

Скінченні автомати (Finite State Machines, FSM) – це класична техніка моделювання поведінки в іграх, де NPC представляється як набір станів і переходів між ними [6]. Кожен стан асоціюється з певною поведінкою, а переходи спрацьовують при виконанні умов. FSM є простими в реалізації та зрозумілими, але вони мають проблему масштабованості: для сучасних складних ігор кількість необхідних станів може бути великою, і навіть ієрархічні FSM (HFSM) лише частково розв'язують цю проблему [6].

Дерева поведінки (Behavior Trees, BT) виникли як альтернатива FSM з підвищеною масштабованістю та виразністю [6]. На відміну від явних переходів між станами, кожен вузол дерева має батьківські вузли, які визначають, як його виконувати. Якщо у FSM кожен стан повинен знати критерії переходу до кожного іншого стану, то BT є «безстановими» (stateless) і не потребують збереження інформації про попередньо активні вузли [6]. Дерева поведінки набули популярності в індустрії після виходу гри Halo 2 (Microsoft Game Studios, 2004), яка ввела BT для керування агентами та підштовхнула індустрію відмовитись від домінуючих ієрархічних скінченних автоматів [7].

Однак FSM і BT в чистому вигляді мають обмежену застосовність до стратегічного III 4X ігор. Ці методи добре підходять для тактичного рівня – поведінки окремих юнітів, реакції на конкретні події, ігрових циклів NPC. Стратегічні рішення в 4X іграх, такі як «який тип перемоги переслідувати?», «чи варто оголошувати війну?», «які технології досліджувати?», погано вкладаються в дискретні стани або деревовидну ієрархію умов. Через це BT

і FSM зазвичай використовуються як компоненти, що інтегруються з іншими методами прийняття рішень, такими як utility-based AI [6].

1.2.3 Системи на основі корисності (Utility-based AI)

Підхід на основі корисності (utility-based AI) розглядає прийняття рішень як математичну задачу: для кожного можливого рішення обчислюється числова «корисність» (utility), і обирається рішення з найвищим значенням. Найбільш розвинутим теоретичним викладом цього підходу є робота Дейва Марка «Behavioral Mathematics for Game AI» [8], яка лежить в основі архітектури, запропонованої в цій роботі.

Центральним поняттям підходу є крива відгуку (response curve) – математична функція, що перетворює вхідне ігрове значення (наприклад, рівень здоров'я, відстань до ворога, кількість ресурсів) на нормоване значення корисності в діапазоні від 0 до 1 [8]. Криві відгуку можуть мати різну форму: лінійну, квадратичну, логістичну (S-подібну) тощо. Форма кривої та її параметри (пороги, кутові коефіцієнти, точки перегину) кодують ставлення ШІ до конкретного ігрового фактора. Так, агент із логістичною кривою для оцінки загрози «не помічатиме» суперника, поки той не перетне певного порогу сили, після чого тривога швидко зросте до максимуму, тоді як агент із квадратичною кривою почне реагувати на загрозу раніше та сильніше [8]. Приклади різних кривих відгуку наведено на рисунку 1.4.

Для прийняття складних рішень, що залежать від кількох факторів, використовується метод зваженого підсумовування (factor weighting) [8]. Корисність рішення обчислюється як зважена сума окремих кривих відгуку:

$$U = \frac{\sum_{i=1}^n w_i \cdot v_i}{\sum_{i=1}^n w_i}, \quad (1.1)$$

де v_i – значення i -ї кривої відгуку,

w_i – її ваговий коефіцієнт.

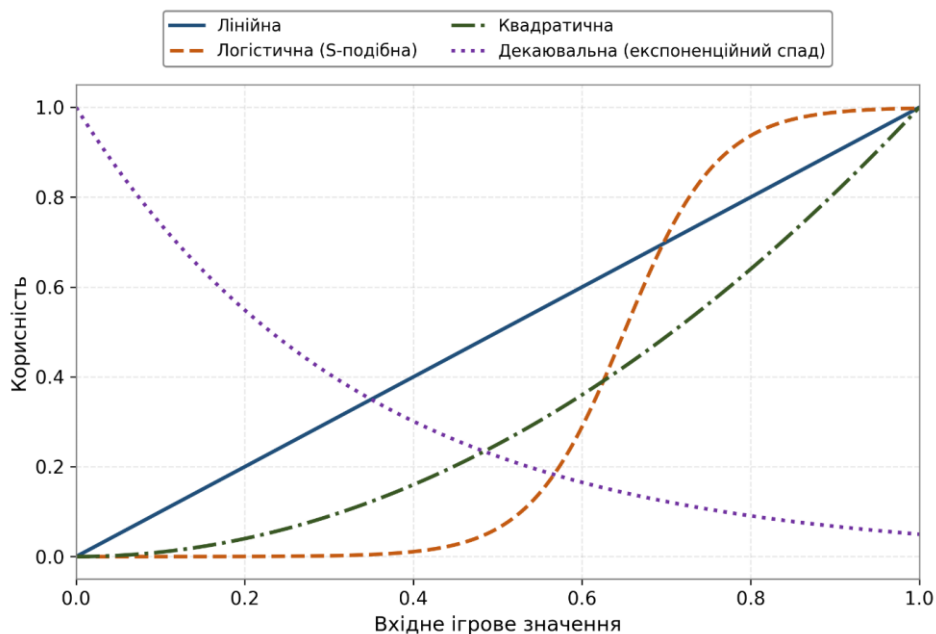


Рисунок 1.4 – Типові форми кривих відгуку у utility-based AI

Через нормалізацію за сумою ваг результат залишається в тому самому діапазоні, що й окремі значення, і піддається порівнянню.

Підхід Марка був формалізований як архітектура «Infinite Axis Utility System» (IAUS), де кожне рішення оцінюється за необмеженою кількістю осей (входів), і дозволяє створювати гнучкі, виразні системи прийняття рішень без необхідності явно описувати кожен можливий стан. Yannakakis та Togelius [7] зазначають, що системи на основі корисності, поряд із FSM і ВТ, є серед історично популярних методів, які широко використовуються в ігровій індустрії. Древа поведінки можуть інтегруватися з утилітарними оцінками: оцінка корисності визначає, коли переходити від одного вузла до іншого, та серед кількох можливих дій обирається та, що має найвищу корисність [6].

Перевагою utility-based AI є природна виразність: параметри функції корисності можуть бути інтерпретовані як «риса характеру» агента, а зміна параметрів напряду впливає на поведінку, не вимагаючи

перепрограмування логіки. Ця властивість робить підхід особливо привабливим як основу для системи особистостей, де параметри кривих відгуку визначаються індивідуальним архетипом агента.

1.2.4 Підходи на основі навчання з підкріпленням

Останнє десятиліття стало періодом значного прориву в застосуванні глибокого навчання та навчання з підкріпленням (Reinforcement Learning, RL) до стратегічних ігор. Ключовою віхою цього напрямку є проект AlphaStar від компанії DeepMind, який у 2019 році досяг рівня гросмейстера у грі StarCraft II (Blizzard Entertainment, 2010), перемігши професійних гравців [7]. AlphaStar успішно навчився грати в гру через глибоку нейронну мережу, що тренувалась шляхом комбінації навчання з вчителем та навчання з підкріпленням [7]. Серед головних викликів StarCraft, на відміну від таких ігор як Го, – відсутність точної та швидкої моделі прогнозування ходів (прямої моделі), а також ігрові характеристики недетермінованості та неповної інформації [7].

Інший значущий приклад – OpenAI Five, що став першим алгоритмом, який переміг чемпіонську команду світу у Dota 2 (Valve, 2013) у 2019 році [7]. Як і AlphaStar, цей алгоритм використовував переважно самонавчання через гру з самим собою (self-play) з застосуванням глибокого навчання з підкріпленням [7].

Проте, попри ці успіхи в окремих жанрах, для класичних 4X стратегій справа дещо інша. Yannakakis та Togelius прямо зазначають, що «грати в будь-яку гру серії Civilization добре за тих самих умов, що й людина, дуже складно, і, наскільки нам відомо, не існує жодного ШІ, здатного грати в ці ігри так само добре, як хороший гравець-людина» [7, переклад автора]. Тому, коли гра ведеться проти кількох комп'ютерних цивілізацій, гра зазвичай «обманює», надаючи цим цивілізаціям певні преференційні умови [7]. Це пов'язано з фундаментальною властивістю 4X ігор: коефіцієнт

розгалуження дерева пошуку є надзвичайно великим. Розглянемо ситуацію, у якій гравець має шість юнітів, і для кожного з них на поточному ході доступно десять різних дій (наприклад, рух у різних напрямках, атака, охорона, побудова покращення тощо). Оскільки на одному ході гравець вибирає по одній дії для кожного юніта незалежно, загальна кількість можливих комбінацій ходу становить $10^6 = 1000000$ – мільйон різних варіантів лише для одного ходу. У сучасних 4X іграх таких юнітів значно більше, кожен має більше можливих дій, а додатково на ході приймаються рішення щодо економіки, технологій та дипломатії. Це робить класичні методи пошуку в дереві гри практично нездійсненними.

Крім обчислювальних обмежень, RL-підхід має концептуальну проблему стосовно мети цієї роботи. Оскільки RL оптимізує єдину цільову функцію (зазвичай ймовірність перемоги), агенти, навчені таким способом, сходяться до однієї домінуючої стратегії – тієї, яка найчастіше призводить до перемоги. Це суперечить меті створення опонентів з виразними та відмінними особистостями: будь-який RL-агент, незалежно від його початкової «налаштованості», у процесі навчання нівелюватиме поведінкові відмінності, якщо вони знижують win-rate. До того ж проекти типу AlphaStar потребують значних обчислювальних ресурсів – навчання велось протягом місяців на сотнях прискорювачів, що робить підхід недоступним для індивідуальних розробників та малих студій.

1.2.5 Підходи на основі великих мовних моделей

Останнім часом активно досліджується застосування великих мовних моделей (Large Language Models, LLM) для прийняття стратегічних рішень в іграх. На відміну від RL-агентів, що навчаються з нуля, LLM мають доступ до накопичених у людському тексті знань про ігрові механіки, тактики та стратегії, і можуть генерувати обґрунтовані рішення через текстовий інтерфейс.

Найбільш релевантним для теми цієї роботи є проект Vox Deorum [9], представлений у грудні 2025 року на конференції FDG '26. Vox Deorum пропонує гібридну архітектуру «LLM+X» для гри Civilization V з модифікацією Vox Populi. Ключова ідея архітектури – багаторівневий поділ задач: великі мовні моделі обробляють макро-стратегічне планування, тоді як тактичне виконання делегується підсистемам (алгоритмічному ШІ або, потенційно, RL-агентам) [9]. Цей підхід було валідовано через 2 327 повних ігрових сесій, у яких порівнювались дві відкриті LLM зі простим промптом проти посиленого ШІ Vox Populi [9]. Результати показали, що LLM досягають конкурентоспроможної гри з кінцевими результатами, при цьому виявляючи стилі гри, які суттєво відрізняються від алгоритмічного ШІ та один від одного [9].

Цей результат особливо цікавий для теми цієї роботи: він демонструє, що поведінкове розмаїття між агентами можна досягти не лише через явне моделювання особистостей, а й як побічний ефект використання різних LLM. Однак цей побічний ефект є некерованим – розробник не контролює, якою саме буде особистість конкретної моделі. Vox Deorum також ілюструє один з фундаментальних викликів LLM-підходу: компонент квадратичного масштабування представлення стану в Markdown означає, що без інтеграції з ефективнішими методами зберігання контексту масштабованість обмежена.

Інше дослідження в цьому напрямку – CivRealm [10], платформа для агентів прийняття рішень на основі відкритої гри Freeciv. CivRealm надає інтерфейси як для тензорних RL-агентів, так і для мовних агентів на основі LLM, що дозволяє порівнювати ці підходи в єдиному середовищі.

Серед недоліків LLM-підходу – висока вартість запитів та значна затримка (до однієї хвилини на рішення), а також ефект сходження до консенсусних стратегій: LLM навчались на великих корпусах текстів, що описують ігрові тактики, форумні обговорення та посібники, тому при

запиті «гравцем-агресором» вони радше згенерують типову агресивну стратегію спільноти, ніж унікальний індивідуальний стиль.

1.2.6 Дослідження переконливості ігрових агентів

Паралельно з оптимізаційними підходами існує окремий напрямок досліджень, який сфокусований не на грі заради перемоги, а на створенні переконливих, людиноподібних агентів – з власною особистістю, характером та емоційним станом. Foundational роботою цього напрямку є стаття Joseph Bates «The Role of Emotion in Believable Agents» (1994) [11], яка стверджувала, що для штучних персонажів важливою є не раціональність чи ефективність виконання задач, а ілюзія життя. На думку Bates, ключовим є вчасно та виразно виражена емоція, особистість та послідовна поведінка – саме ці властивості перетворюють агента на персонажа, в якого віриш [11].

Цей напрямок отримав новий поштовх з появою великих мовних моделей. Робота Park та інших «Generative Agents: Interactive Simulacra of Human Behavior» (2023) [12] продемонструвала архітектуру LLM-агентів з потоком пам'яті (memory stream), рефлексією та плануванням, які виявляють емерджентну переконливу соціальну поведінку: агенти планували вечірки, поширювали чутки, формували та руйнували стосунки. Архітектура «потік пам'яті + рефлексія + планування» [12] суттєво вплинула на проектування системи пам'яті, запропонованої в цій роботі.

Важливо зазначити, що традиція дослідження переконливості ставить інший набір цілей перед розробником, ніж оптимізаційні підходи. Якщо RL та класичний LLM-підхід оптимізують «грати добре», то дослідження переконливості ставлять перед агентом мету «грати цікаво» – поводитися як персонаж з власним характером, навіть якщо це призводить до субоптимальних рішень з точки зору шансу або відсотку перемог. Таке свідоме відхилення від оптимальності є властивістю, а не недоліком,

оскільки саме воно робить агента запам'ятовуваним та надає кожній ігровій партії унікальної історії.

Сучасні дослідження інтегрують елементи переконливості з традиційними ігровими архітектурами. Sekhavat [6] описує спробу інтеграції емоційних факторів у дерева поведінки: дизайнер ІІІ вказує, які саме поведінки можуть зазнавати впливу емоційних факторів, а «емоційний селектор» – це вузол, який обирає, яке дочірнє правило виконувати, на основі поточних значень емоцій агента [6]. Такі гібридні підходи демонструють, що переконливість і функціональна поведінка не є взаємовиключними, але потребують свідомого об'єднання на рівні архітектури.

1.2.7 Висновки до підрозділу

Огляд сучасних підходів до ІІІ в стратегічних іграх виявляє кілька важливих закономірностей. По-перше, методи розташовані вздовж спектру від детермінованих, інтерпретованих та обчислювально дешевих (FSM, BT, скрипти на основі коефіцієнтів) до недетермінованих, важко інтерпретованих та обчислювально витратних (RL, LLM). Жоден з кінців цього спектру не є оптимальним для задачі цієї роботи: класичні методи створюють передбачувану, ригідну поведінку, а сучасні методи навчання та LLM сходяться до домінуючих стратегій або потребують ресурсів, недоступних для індивідуального розробника.

По-друге, підхід utility-based AI займає особливе місце як основа архітектури з керованою інтерпретованою поведінкою, де параметри функції корисності природно відображають індивідуальні характеристики агента. Цей підхід має достатню виразну силу для моделювання стратегічних рішень і одночасно зберігає прозорість, необхідну для відлагодження та аналізу поведінки. У поєднанні з принципами переконливості, обґрунтованими в роботах Bates [11] та Park [12], utility-

based AI становить технічну основу для запропонованої в цій роботі системи особистостей.

Запропонована архітектура заповнює нішу, недостатньо покриту існуючими підходами: вона дозволяє створити агентів з виразною індивідуальністю, які поведуться послідовно у відповідності зі своєю «особистістю», адаптуються до подій конкретної гри через систему настрою та пам'яті, і при цьому зберігають інтерпретованість та обчислювальну ефективність. Деталі архітектури буде розглянуто у наступних розділах.

1.3 Виявлені обмеження та виклики

Огляд сучасних підходів, проведений у попередньому підрозділі, демонструє, що жоден з існуючих методів повністю не задовольняє вимог, поставлених перед ШІ-опонентами у 4X стратегіях для забезпечення довготривалого ігрового досвіду. У цьому підрозділі систематизовано конкретні обмеження кожного класу підходів, які формують ту нішу, яку має заповнити запропонована в цій роботі система.

1.3.1 Передбачуваність систем зі статичними коефіцієнтами

Як показано у пункті 1.2.1, скриптові системи на основі статичних ваг є детермінованими за своєю природою: однаковий стан гри породжує однакову реакцію ШІ. Через це, після кількох ігрових сесій досвідчений гравець починає розпізнавати поведінкові шаблони опонента та експлуатувати їх. Лідер серії Civilization, налаштований на агресивну поведінку, оголошує війну за приблизно однакових умов у кожній грі; «миролюбний» лідер реагує на однакові дипломатичні пропозиції однаково. Spronck та інші [5] саме цю проблему намагалися подолати через динамічний скриптинг, однак запропоноване ними рішення оптимізує ваги

в напрямку перемоги, що породжує іншу проблему, розглянуту нижче. Введення прихованих агенд у Civilization VI [3] частково знижує передбачуваність, але самі агенти залишаються набором статичних правил, які гравець вивчає аналогічно до основних, лише потребуючи трохи більше часу.

1.3.2 Збіжність до META-стратегій в оптимізаційних підходах

Підходи на основі навчання з підкріпленням, обговорені в пункті 1.2.4, та LLM-агенти, розглянуті в пункті 1.2.5, мають спільне фундаментальне обмеження: їх оптимізаційна природа сприяє конвергенції до домінуючих стратегій. RL-агент, навчений на тисячах ігор, знаходить статистично найвигіднішу стратегію та поступово витісняє з своєї поведінки всі індивідуальні варіації, що знижують win-rate. LLM-агенти, у свою чергу, при запитах на гру в певному стилі (наприклад, «агресивно» або «дипломатично») генерують консенсусну стратегію спільноти, яка склалась у текстах, на яких модель навчалась. Емпіричні спостереження проекту Vox Deorum [9] показали, що різні LLM-моделі справді мають відмінні стилі гри, але ці відмінності є некерованими побічними ефектами, а не результатом авторського задуму розробника. Для задачі цієї роботи, метою якої є створення опонентів з виразними та запам'ятовуваними особистостями, оптимізаційні підходи не лише не допомагають, але й активно заважають – вони системно усувають саме ті «недоліки» (свідомі субоптимальні рішення, продиктовані характером), які роблять штучного опонента цікавим.

1.3.3 Відсутність персистентної пам'яті про події гри

Більшість сучасних реалізацій ШІ у 4X іграх обробляють кожний хід незалежно або тільки з мінімальним обліком найсвіжіших подій. ШІ-лідер у

Civilization VI може мати модифікатор дипломатичних відносин, що зменшується з часом, але цей модифікатор сам по собі не утворює структурованої пам'яті. Як результат, ІІІ виявляється вразливим до повторюваних маніпуляцій з боку гравця. Якщо гравець виявить, що певна послідовність дій породжує сприятливу реакцію опонента, він може повторювати її протягом гри без ризику, що ІІІ розпізнає шаблон і змінить ставлення. Реальна людина в подібній ситуації запам'ятала б, що з цим конкретним суперником вже відбувались подібні інциденти, і ставилась би до нього з відповідною обережністю. У концепції переконливих агентів, запропонованій Bates [11], та в архітектурі агентів з потоком пам'яті Park та інших [12], персистентна пам'ять про значущі події є основою для формування характеру, що розвивається протягом тривалої взаємодії. Існуючі рішення для 4X ігор переважно не реалізують цей принцип.

1.3.4 Відсутність динаміки емоційного стану

Тісно пов'язаним з попереднім обмеженням є відсутність системи настрою чи емоційної динаміки в більшості існуючих ІІІ. Хоча параметри особистості можуть змінюватись у відповідь на конкретні події (наприклад, дипломатичні модифікатори у Civilization VI), не існує загального стратегічного стану, який би відображав, чи ІІІ відчуває себе впевнено, чи у відчаї. Людський гравець у складній ситуації змінює стратегію: програючи, починає робити ризиковані ходи; маючи перевагу, стає консервативнішим; будучи зрадженим довіреним союзником, поступово втрачає здатність довіряти. Bates [11] прямо стверджував, що своєчасно та виразно виявлена емоція є центральною вимогою до переконливих ігрових агентів. Сучасні ІІІ-опоненти у 4X іграх не моделюють цей вимір повноцінно, що знижує їх правдоподібність та робить поведінку механічною.

1.3.5 Обчислювальні обмеження сучасних методів

Найсильніші сучасні методи з точки зору продуктивності – глибокі RL-агенти та великі мовні моделі – потребують значних обчислювальних ресурсів. Проект AlphaStar [7] тренувався протягом місяців на сотнях прискорювачів, що робить такий підхід фактично недоступним для одноосібних розробників, малих студій та академічних проектів. LLM-підходи, як показано у Vox Deorum [9], мають менший бар'єр входу, але все одно потребують або платних API-запитів (з затримкою до однієї хвилини на рішення та помітною вартістю на повну гру), або достатнього локального GPU для запуску квантизованих моделей. Тобто, використання таких підходів призводить до значних витрат ресурсів розробників (необхідність потужної інфраструктури для тренування або платних API для запитів) та підвищення системних вимог для гравців. У контексті ігор ця проблема має додатковий аспект: GPU вже використовується для рендерингу графіки, тому виділення його обчислювальних ресурсів під роботу мовної моделі може негативно вплинути на продуктивність та якість зображення, або вимагатиме від гравця значно потужнішого обладнання, ніж потребує сама гра. Це значно ускладнює впровадження подібних технологій у комерційних іграх.

1.3.6 Висновки до підрозділу

Описані вище обмеження не є ознакою непрофесійності існуючих розробок – кожен з розглянутих методів був створений для розв'язання конкретної задачі та добре справляється з нею в межах свого призначення. Проблема полягає в тому, що жоден з них не був спеціально спроектований для задачі, поставленої в цій роботі: створення набору ШІ-опонентів з виразними індивідуальностями, які зберігають характер протягом гри, реагують на події емоційно та запам'ятовують свій досвід, при цьому

залишаючись обчислювально доступними для реалізації одним розробником. Запропонована система особистостей закриває саме цю прогалину, поєднуючи інтерпретованість utility-based AI з принципами переконливості з традиції дослідження believable agents, додаючи компоненти настрою та пам'яті, відсутні у класичних ігрових архітектурах. Деталі цієї системи розглядаються у наступному підрозділі.

1.4 Концепція запропонованого рішення

З урахуванням обмежень існуючих підходів, виявлених у попередньому підрозділі, пропонується система динамічних особистостей для ШІ-агентів у 4X стратегіях [13]. Ключова ідея полягає в тому, що поведінка опонента визначається не єдиним фіксованим набором параметрів і не результатом оптимізації за показником win-rate, а взаємодією трьох концептуально розрізнених підсистем: статичного архетипу характеру, динамічного настрою та персистентної пам'яті. Кожен з цих компонентів усуває одне з обмежень, описаних у підрозділі 1.3.

Перш ніж розглянути структуру системи, важливо зазначити фундаментальну зміну в постановці мети для ШІ-опонентів [13]. Як зауважує Joseph Bates [11], «зовнішні бажання персонажа і те, як персонаж відчуває себе у відношенні до подій світу через призму цих бажань, є тим, що змушує нас піклуватись про персонажа. Якщо персонаж не реагує емоційно на події, якщо він не переймається ними – то і ми не будемо. Беземоційний персонаж є неживим, як машина» (переклад автора). Тобто, для досягнення мети – створення цікавих та запам'ятовуваних опонентів – недостатньо зробити ШІ «сильнішим». Метою має бути не перемога у грі, а створення найцікавішого ігрового досвіду для гравця-людини. Це означає, що ШІ-опоненти повинні приймати рішення, які не обов'язково є оптимальними, але мають сенс з емоційної точки зору та відповідають характеру конкретного агента.

1.4.1 Тришарова архітектура особистості

Архетип характеру (завойовник, дипломат, економіст, тощо) представляє собою сукупність рис, тенденцій та пріоритетів – таких як рівень агресивності, довіри, зацікавленості в дипломатії, схильності до прийняття ризиків та інших характеристик, що впливають на прийняття рішень протягом усієї гри [13]. Архетип кожного опонента обирається перед початком гри гравцем або призначається випадково та залишається незмінним протягом партії. Саме архетип є джерелом базової індивідуальності, відсутньої в оптимізаційних підходах: оскільки параметри архетипу задаються розробником, а не виводяться з оптимізації за win-rate, вони не сходяться до єдиної домінуючої МЕТА-стратегії.

Настрій відповідає за короткострокові цілі та стратегічні пріоритети агента і динамічно змінюється кожного ходу в залежності від подій, що відбуваються в грі [13]. Так, якщо інший гравець відхиляє пропозицію дипломатичного пакту або атакує юніт агента, це підвищує пріоритет для посилення армії та створення захисних альянсів з іншими гравцями. Якщо інший гравець засновує нове місто поряд із територією агента, це сприймається як експансія в його зону впливу й посилює настороженість та ворожість агента саме до цього конкретного суперника, що згодом враховується при прийнятті дипломатичних і воєнних рішень стосовно нього. Настрій також включає емоційний стан – рівень страху, впевненості, гніву, довіри щодо кожного конкретного суперника окремо. Саме цей шар забезпечує динаміку реакції на ігрові події, відсутню в скриптових системах зі статичними коефіцієнтами.

Пам'ять дозволяє агенту зберігати інформацію про значущі події гри та враховувати її при подальших рішеннях [13]. Наприклад, якщо певний суперник неодноразово оголошує війну, через кілька ходів пропонує мир за викуп, передачу ресурсів або території, а потім розпочинає війну знову, з часом це робить агента менш схильним приймати такі пропозиції від цього

конкретного суперника. Це дозволяє агенту «не наступати на ті самі граблі» та вчитися з власного досвіду в межах окремої гри. Кожен запис у пам'яті має параметр значущості та емоційну вагу, що зменшується з часом зі швидкістю, залежною від характеру агента (зокрема, рівня його злопам'ятності) та важливості події: один архетип може швидко прощати всі, крім найсерйозніших образ, а інший – пам'ятати навіть невеликі інциденти до кінця гри. Схему роботи наведено на рисунку 1.5.

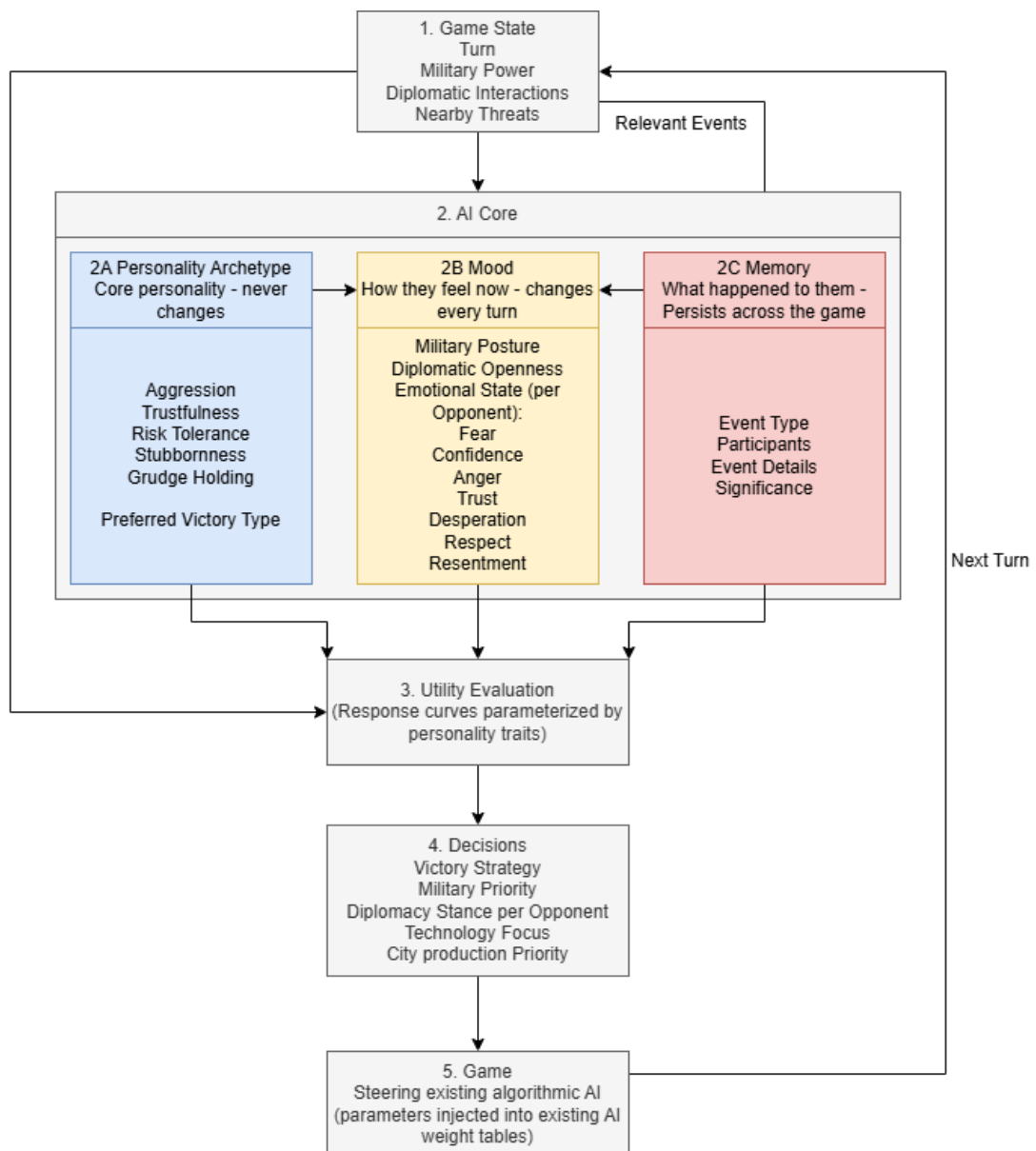


Рисунок 1.5 – Архітектура системи динамічних особистостей

1.4.2 Механізм прийняття рішень

Кожного ходу агент отримує дані про поточний стан гри, поточні значення параметрів усіх трьох шарів моделі особистості та приймає рішення щодо подальших дій за допомогою кривих відгуку (response curves) [8], форма та параметри яких визначаються обраним архетипом [13]. Цей механізм базується на класичному підході utility-based AI, описаному в підрозділі 1.2.3, однак ключова відмінність полягає в тому, що параметри кривих відгуку не є статичними константами, а формуються з композиції рис архетипу та поточного емоційного стану. Таким чином, та сама ігрова подія в одній грі може бути обробленою агентом інакше, ніж в іншій грі – або навіть інакше, ніж 50 ходів тому в межах тієї самої партії – залежно від того, що сталося з агентом за цей час.

Реалізацію системи пропонується виконати у вигляді модифікації (моду) до гри Endless Legend [13]. Запропонована система не замінює існуючий алгоритмічний ШІ гри повністю, а доповнює його, динамічно змінюючи таблиці ваг та пріоритетів, які цей ШІ використовує для прийняття рішень. Цей шаруватий підхід інспірований архітектурою Vox Deorum [9], де макро-стратегічні рішення приймаються одним шаром системи, а тактичне виконання делегується нижчому шару. Для практичного впровадження буде використано інструменти декомпіляції та рекомпіляції C# збірок гри. Параметри архетипів не виводяться шляхом оптимізації, а свідомо задаються розробником – виходячи з бажаного характеру кожного архетипу та потреби забезпечити виразну поведінкову різноманітність ансамблю при збереженні його життєздатності. Саме тому, на відміну від оптимізаційних методів (RL, динамічний скриптинг, LLM), поведінкові відмінності між архетипами не нівелюються заради максимізації відсотка перемог (win-rate), а навпаки – є основною проектною метою.

1.4.3 Підсумкова характеристика підходу

Запропонована система поєднує переваги декількох парадигм, розглянутих у підрозділі 1.2, при цьому уникаючи їх характерних обмежень. Від класичних utility-based систем вона успадковує детермінованість, інтерпретованість та обчислювальну ефективність – кожне рішення може бути простежене назад до конкретних параметрів характеру, а вся система працює за мілісекунди на хід без потреби в GPU чи API-запитах. Від традиції дослідження переконливих агентів [11], [12] вона запозичує принципи персистентної пам'яті та емоційної динаміки, що дозволяють агенту розвивати характер протягом гри. Від концепції шаруватої архітектури [9] вона переймає ідею делегування тактичного виконання існуючому алгоритмічному ШІ, що значно знижує складність впровадження. На відміну від оптимізаційних підходів (RL, LLM), запропонована система не сходиться до єдиної домінуючої стратегії, оскільки її параметри не оптимізуються за win-rate, а свідомо проектуються розробником для забезпечення поведінкової різноманітності. Це створює ШІ-опонентів з унікальними особистостями, що залежать як від «вроджених рис характеру», так і від подій конкретної гри, та можуть поступово змінюватись і еволюціонувати з часом [13]. Це підвищує ефект занурення гравця та перетворює кожну ігрову партію на власну історію зі своїм сюжетом і персонажами.

1.5 Постановка задачі

На основі аналізу предметної галузі, виявлених обмежень сучасних підходів та запропонованої концепції розв'язання, було формалізовано задачу дослідження.

Об'єктом дослідження є системи штучного інтелекту для покрокових стратегічних ігор піджанру 4X.

Предметом дослідження є методи створення ІІІ-агентів з виразними індивідуальними особистостями, здатних до динамічної адаптації поведінки на основі ігрового досвіду, що поєднують інтерпретованість класичних підходів utility-based AI з персистентною пам'яттю та емоційною динамікою, характерними для досліджень переконливих агентів.

Метою роботи є розробка та реалізація системи динамічних особистостей для ІІІ-опонентів у 4Х стратегіях, що дозволяє створювати агентів з виразними характерами, які зберігаються та еволюціонують протягом гри, з метою підвищення поведінкового розмаїття опонентів, ефекту занурення гравця та зацікавленості в ігровому процесі, при збереженні обчислювальної ефективності, достатньої для реалізації у вигляді модифікації комерційної гри.

Для досягнення поставленої мети необхідно розв'язати наступні задачі дослідження:

- провести аналіз предметної галузі 4Х стратегічних ігор, дослідити сучасні підходи до реалізації ІІІ-опонентів у цьому жанрі та виявити їх обмеження;
- розробити архітектуру системи динамічних особистостей, що складається з трьох взаємодіючих підсистем: статичного архетипу характеру, динамічного настрою та персистентної пам'яті;
- спроектувати математичну модель прийняття рішень на основі кривих відгуку, параметри яких визначаються поточним станом усіх трьох шарів моделі особистості;
- розробити набір базових архетипів характеру з визначеними значеннями параметрів, які забезпечують виразну поведінкову різноманітність;
- реалізувати запропоновану систему у вигляді модифікації гри Endless Legend через декомпіляцію, модифікацію та рекомпіляцію .NET збірок гри;

- провести експериментальну валідацію розробленої системи через ігрові симуляції з ШІ-опонентами, виміряти показники поведінкової різноманітності та конкурентоспроможності архетипів;

- проаналізувати отримані результати, сформулювати висновки та визначити напрямки подальшого розвитку.

Методами дослідження є: аналіз наукової літератури та технічної документації для дослідження сучасних підходів до ШІ в стратегічних іграх; методи об'єктно-орієнтованого проектування для розробки архітектури системи особистостей; математичне моделювання для опису функцій корисності та кривих відгуку; методи реверс-інжинірингу (декомпіляція та модифікація .NET збірок мовою C#) для практичної інтеграції з існуючим ШІ гри; методи статистичного аналізу для оцінки експериментальних результатів та порівняння поведінки різних архетипів.

Очікуваними результатами роботи є функціональна модифікація (мод) для гри Endless Legend, що реалізує запропоновану систему динамічних особистостей та інтегрується з існуючим алгоритмічним ШІ гри; набір базових архетипів характеру з налаштованими параметрами, які демонструють виразно різну поведінку у грі при збереженні конкурентоспроможності; експериментальні дані, що підтверджують поведінкове розмаїття розроблених архетипів та їх життєздатність відносно стандартного ШІ гри; текст кваліфікаційної роботи з описом теоретичних основ, архітектури системи, методології реалізації та результатів експериментів.

Практичне значення роботи полягає у тому, що запропонована архітектура може бути впроваджена в інші 4X стратегічні ігри для покращення якості ШІ-опонентів та підвищення задоволеності гравців. На відміну від підходів на основі великих мовних моделей або глибокого навчання з підкріпленням, система не потребує спеціалізованого обладнання (GPU) або платних API-сервісів, що робить її доступною для індивідуальних розробників, малих ігрових студій та модерських спільнот.

2 ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДИНАМІЧНИХ ОСОБИСТОСТЕЙ

У цьому розділі обґрунтовано вибір напрямку та стратегії реалізації запропонованої у підрозділі 1.4 системи динамічних особистостей, наведено інструментальні засоби, що використовувалися для практичної інтеграції з грою, та їх порівняльну оцінку, окремо розглянуто правові підстави модифікації комерційної гри. Далі описано загальну архітектуру програмної реалізації, модель особистості та математичну модель прийняття рішень, точки вбудовування у штучний інтелект гри й засоби забезпечення надійності, а також підсистеми конфігурації та інструментування. Завершується розділ методикою проведення експериментального дослідження. Безпосередні результати експериментів наведено у розділі 3, а напрями подальшого розвитку – у розділі 4.

2.1 Обґрунтування вибору напрямку реалізації та стратегії інтеграції

Як показано у підрозділі 1.4, концептуальною основою роботи є поєднання utility-based підходу [8] з принципами переконливих агентів [11], [12], реалізоване у вигляді модифікації гри Endless Legend. Ключовим інженерним рішенням, від якого залежить уся подальша реалізація, є спосіб інтеграції запропонованої системи особистостей у закриту комерційну гру, вихідний код якої недоступний.

Принципово можливі дві стратегії. Перша – повна заміна наявного алгоритмічного штучного інтелекту гри власною системою. Друга – доповнення наявного ШІ, при якому власна система не перебирає на себе всі функції, а лише корегує стратегічні та дипломатичні рішення, делегуючи тактичне виконання (переміщення юнітів, ведення бою, управління містами) штатному штучному інтелекту гри. У роботі обрано другу стратегію. Її перевагою є суттєво нижча складність реалізації: штатний ШІ

Endless Legend цілком задовільно виконує тактичні завдання, і відтворення цієї функціональності з нуля виходить за межі мети роботи й не є необхідним. Така багатошарова організація, за якої макростратегічні рішення приймає один шар системи, а тактичне виконання делегується нижчому шару, узгоджується з архітектурним підходом проєкту Vox Deorum [9].

Вибір гри Endless Legend як платформи зумовлений сукупністю чинників. Гру побудовано на Unity game engine з виконавчим середовищем Mono, а її логіку реалізовано у керованих .NET-збірках, що піддаються декомпіляції, унаслідок чого обраний механізм інтеграції є технічно здійсненним. Штатний штучний інтелект гри вже побудовано за принципом корисності й оснащено реєстром особистісних параметрів, до якого можна під'єднатися, що природно узгоджується із запропонованою системою. Крім того, гра вирізняється високою асиметричністю фракцій та має активну спільноту модифікаторів і офіційну підтримку моддингу, що додатково обґрунтовує її придатність як об'єкта дослідження.

Обрана стратегія доповнення потребує визначення механізму, через який власна система впливатиме на рішення штатного ШІ. Було розглянуто чотири можливі механізми інтеграції, порівняльну оцінку яких наведено у таблиці 2.1. Перший механізм – офіційні засоби моддингу гри, що ґрунтуються на редагуванні XML-файлів у каталозі гри [14]. Він є найбезпечнішим, проте обмежений лише тими параметрами, які розробники свідомо винесли у конфігурацію; він не дозволяє вбудовувати нову логіку прийняття рішень або динамічно змінювати значення під час гри. Другий механізм – зовнішня програма, що модифікує пам'ять процесу гри під час її роботи. Він є вкрай нестійким до оновлень, складним у супроводі й не надає чистого програмного інтерфейсу. Третій механізм – пряма модифікація збірок гри: декомпіляція бібліотеки Assembly-CSharp.dll, вставлення у точки прийняття рішень викликів до окремої супровідної бібліотеки, що містить логіку особистостей, та зворотне збереження модифікованої збірки.

Четвертий механізм – модифікація на рівні вихідного коду – є неможливим, оскільки гра постачається у закритому вигляді.

Таблиця 2.1 – Порівняльна оцінка механізмів інтеграції

Механізм	Доступ до логіки ШІ	Складність	Стійкість	Тестованість
XML-моддинг	Лише відкриті параметри	Низька	Висока	Низька
Ін'єкція в пам'ять	Повний, але нестабільний	Висока	Низька	Низька
Модифікація збірок	Повний	Середня	Середня	Висока
Рівень коду	Недоступно			

За результатами порівняння обрано третій механізм – пряму модифікацію збірок гри з винесенням логіки особистостей в окрему супровідну бібліотеку. Цей варіант надає повний доступ до конкретних точок прийняття рішень штатного ШІ (обчислення корисностей та дипломатичних оцінок), виконується в одному процесі з грою, а завдяки винесенню логіки в окрему бібліотеку забезпечує її незалежне модульне тестування. Недоліком підходу є залежність від стабільності бінарного файлу гри, що компенсується дисципліною розгортання, описаною у підрозділі 2.3.

2.2 Правові аспекти модифікації гри

Обраний механізм передбачає декомпіляцію та модифікацію збірок комерційної гри, що ставить питання про правомірність таких дій. У роботі встановлено, що така модифікація має щонайменше три незалежні правові підстави: ліцензійну угоду видавця, національне законодавство України про авторське право та відповідне законодавство Європейського Союзу.

По-перше, ліцензійна угода кінцевого користувача (EULA) компанії Amplitude Studios, що поширюється на Endless Legend, у секції 9 прямо дозволяє відповідні дії за умови, що видавець допускає створення модифікацій [15]. Угода встановлює: «When the LICENSOR allows the creation of modifications for use within the SOFTWARE PRODUCT (...) the LICENSEE is allowed to reverse engineer, decompile, disassemble, translate, adapt, arrange or modify the SOFTWARE PRODUCT to create or to perform a function not originally conceived or intended by the LICENSOR and/or to create new or altered content».

Endless Legend офіційно підтримує модифікації: гра постачається з інструментами моддингу, інтегрована з майстернею Steam і супроводжується офіційним посібником зі створення модифікацій [14]. Отже, умова «коли ліцензіар дозволяє створення модифікацій» виконується, а передбачений секцією 9 дозвіл на reverse engineering, декомпіляцію та модифікацію програмного продукту для створення модів є чинним для цієї гри.

По-друге, правомірність декомпіляції з метою забезпечення взаємодії незалежно встановлюється законодавством України. Стаття 25 («Вільне використання комп'ютерних програм») Закону України «Про авторське право і суміжні права» № 2811-IX від 01.12.2022 у частині 1, пункті 2, надає законному користувачеві право без дозволу суб'єкта авторського права декомпілювати комп'ютерну програму «з метою одержання інформації, необхідної для досягнення її взаємодії з іншою незалежно розробленою комп'ютерною програмою» за дотримання трьох умов [16]: інформація раніше не була доступною з інших джерел; декомпіляція здійснюється виключно щодо тих частин програми, які необхідні для досягнення взаємодії; одержана інформація використовується лише для досягнення взаємодії, не передається іншим особам і не застосовується для створення суттєво подібної програми чи іншого порушення авторського права.

Запропонована модифікація точно відповідає цьому правовому складу. Супровідна бібліотека є самостійною, незалежно розробленою комп'ютерною програмою, яка взаємодіє з грою; декомпіляція обмежена лише тими точками логіки ІІІ, що необхідні для організації цієї взаємодії; одержана інформація використовується винятково для досягнення взаємодії та не застосовується для створення конкурентного продукту. Аналогічну норму містить і законодавство Європейського Союзу – стаття 6 Директиви 2009/24/ЄС про правову охорону комп'ютерних програм, яка дозволяє декомпіляцію без дозволу правовласника, коли вона необхідна для досягнення сумісності незалежно створеної програми з іншими програмами, за подібних умов [17]; вітчизняна норма гармонізована з цією директивою.

По-третє, реалізація передбачає додаткові запобіжники, які виключають порушення авторського права. Модифікація не поширює жодного коду гри: кінцевому користувачеві передаються лише власна супровідна бібліотека, файл конфігурації та текстова інструкція щодо точок вставлення викликів, які застосовуються користувачем до власної легально придбаної копії гри. Не відбувається обходу технічних засобів захисту, а використання має некомерційний, навчально-дослідницький характер. З огляду на сукупність зазначених підстав, виконувана у роботі модифікація є правомірною.

2.3 Інструментальні засоби декомпіляції та рекомпіляції

Цільовим об'єктом модифікації є керована .NET-збірка гри. Endless Legend побудовано на ігровому рушії Unity версії 5.3 з виконавчим середовищем Mono; логіку штучного інтелекту зосереджено у керованій бібліотеці Assembly-CSharp.dll, скомпільованій у спільну проміжну мову CIL відповідно до стандарту ECMA-335 [18] і націленій на .NET Framework 3.5. Саме керована природа цієї збірки уможливорює її

декомпіляцію у читабельний код мовою C#, точкове редагування та зворотне збереження.

Керована збірка містить не лише виконуваний код у вигляді команд проміжної мови, а й метадані, що описують типи, методи та поля; саме наявність метаданих робить декомпіляцію у близький до оригіналу код C# принципово можливою, на відміну від суто машинного коду. Натомість зворотне збереження зміненої збірки є нетривіальним, оскільки потребує узгодженого перезапису таблиць метаданих і потоку команд зі збереженням цілісності модуля. Тому вирішальною вимогою до інструмента була не якість декомпіляції сама по собі, а здатність коректно зберегти змінений модуль.

Виявлення точок прийняття рішень виконувалося безпосередньо у декомпільованому коді: за назвами класів дипломатичних агентів-доданків, реєстру особистісних параметрів та планувальника штучного інтелекту було встановлено конкретні методи, у які слід вставити виклики фасаду. Оскільки точкове втручання у чужий код є джерелом ризику, кількість і обсяг правок зведено до мінімуму, а самі правки задокументовано для відтворюваності.

До інструментального засобу декомпіляції висувалися такі вимоги: декомпіляція CIL у читабельний код C#; можливість редагування на рівні C# та проміжної мови; та, що найважливіше, можливість перекомпіляції і збереження зміненого модуля назад у файл збірки. Порівняльну оцінку поширених засобів за цими критеріями наведено у таблиці 2.2.

Таблиця 2.2 – Порівняльна оцінка засобів декомпіляції та редагування .NET-збірок

Засіб	Декомпіляція	Редагування та збереження модуля	Рівень CIL	Підтримка
ILSpy	Так	Обмежено	Перегляд	Активна

Продовження таблиці 2.2

Засіб	Декомпіляція	Редагування та збереження модуля	Рівень CIL	Підтримка
dotPeek	Так	Ні	Перегляд	Активна
dnSpy	Так	Так	Так	Припинена
dnSpyEx	Так	Так	Так	Активна

За результатами порівняння обрано dnSpyEx – підтримувану спільнотою спадкоємицю широко відомого редактора й налагоджувача .NET-збірок dnSpy [19]. Серед розглянутих засобів лише dnSpy та dnSpyEx підтримують безпосереднє редагування коду й команду збереження зміненого модуля («Save Module»), а також редагування на рівні проміжної мови, необхідне для точкового вставлення викликів. Оскільки розробку оригінального dnSpy припинено, у роботі використано саме підтримувану версію dnSpyEx.

Супровідну бібліотеку реалізовано як бібліотеку класів, націлену на .NET Framework 3.5, і зібрано за допомогою системи збирання MSBuild у складі Visual Studio [20]. Націлювання саме на версію 3.5 є обов’язковим: виконавче середовище Mono гри завантажує лише сумісні з цією версією збірки, а спроба використати новіші версії платформи призводить до помилок завантаження типів під час виконання. З цієї ж причини для збирання застосовано класичний MSBuild, а не сучасний інтерфейс командного рядка dotnet, у складі якого відсутні довідкові збірки для застарілої версії 3.5.

Повний цикл інтеграції складається з таких кроків: декомпіляція бібліотеки Assembly-CSharp.dll та виявлення точок прийняття рішень штатного III; вставлення у ці точки викликів до фасаду супровідної бібліотеки на рівні C# або проміжної мови; збереження зміненого модуля; розгортання супровідної бібліотеки разом зі зміненою збіркою гри поряд із виконуваним файлом гри; перевірка коректності. Окремої уваги потребує

дисципліна розгортання: перед модифікацією зберігається еталонна резервна копія збірки, а для зміненого файлу фіксується контрольний відбиток (розмір і хеш-сума). Це необхідно тому, що перевірка цілісності файлів у Steam або оновлення гри відновлюють оригінальний бінарний файл і знищують усі внесені зміни; контроль відбитка дозволяє своєчасно виявити таке відновлення.

2.4 Загальна архітектура програмної реалізації

Програмна реалізація складається з двох збірок: супровідної бібліотеки модифікації (бібліотека класів .NET 3.5) та зміненої збірки гри. Фундаментальним принципом архітектури є те, що модифікація ніколи не змінює власний стан гри: вона лише зчитує значення, обчислює корегування та повертає результат, який штатний ШІ використовує далі. Це гарантує, що збій модифікації не може пошкодити стан партії.

Єдиною відкритою точкою взаємодії між грою та модифікацією є фасад – статичний клас, що є єдиним публічним типом супровідної бібліотеки. Усі вставлені у збірку гри виклики звертаються саме до методів фасаду. Після вставлення викликів сигнатури фасаду стають незмінним інтерфейсом: зміна назв методів, параметрів чи типів повернення призвела б до помилки під час першого ж спрацювання вставленого виклику. Решта типів бібліотеки є внутрішніми.

Незмінність інтерфейсу фасаду має практичний наслідок: її порушення компілюється без помилок, але виявляється лише під час першого спрацювання вставленого виклику у грі – у вигляді помилки відсутнього методу. Тому фасад спроектовано як такий, що лише доповнюється: до нього дозволено додавати нові методи, але заборонено змінювати наявні. Збірку ніколи не підписують строгим іменем і не змінюють номер її версії, щоб уникнути неузгодженості з уже вставленими викликами.

Внутрішня організація бібліотеки є багатошаровою, причому межі шарів збігаються з межами просторів імен. Шари фасаду та інтеграції є єдиними місцями, де дозволено використання типів рушія та гри; дані перетинають цю межу у вигляді простих значень (індекс імперії, ідентифікатор параметра, числові значення). Шари логіки особистості, конфігурації та інструментування не містять жодних типів гри й використовують лише базову бібліотеку класів. Завдяки цьому ядро логіки особистості піддається повноцінному модульному тестуванню незалежно від гри.

Потік даних під час прийняття одного рішення схематично зображено на рисунку 2.1. Кожного ходу штатний ШІ гри під час обчислення певного параметра звертається до вставленого виклику, який передає фасаду індекс імперії, ідентифікатор параметра та базове значення. Фасад визначає актуальний знімок особистості відповідної імперії, застосовує до базового значення корегування, що впливає з архетипу, настрою та пам'яті, записує рядок телеметрії та повертає скореговане значення. Гра продовжує обчислення вже зі скоригованим значенням. Важливою властивістю фасаду є те, що його методи є тотальними функціями: за будь-якої внутрішньої помилки вони повертають вхідне значення без змін, унаслідок чого збій модифікації не здатний ані аварійно завершити, ані «підвісити» гру.

Реалізацію описаного принципу тотальної функції на прикладі хука H1 (коригування числового параметра) наведено у лістингу 2.1.

Лістинг 2.1 – Фасадний метод коригування значення із запобіжником

```
public static float AdjustFloat(int empireIndex, string path,
float baseValue)
{
    try
    {
        PersonalitySnapshot snap = BeginCall(empireIndex);
        if (snap == null) return baseValue;           // не
персоналізована імперія
        float adjusted = baseValue;
        if (! latched)                               // якщо мод не вимкнено
```

Продовження лістингу 2.1

```

    {
        float product = snap.Apply(_pathMap, path,
baseValue);
        if (float.IsNaN(product) ||
float.IsInfinity(product))
            FaultGuard.Record(new ArithmeticException(
                "non-finite AdjustFloat product"));
        else
            adjusted = product;
    }
    bool latched = RefreshAndRecord(empireIndex, -1, path,
snap,
        baseValue.ToString(CultureInfo.InvariantCulture),
        adjusted.ToString(CultureInfo.InvariantCulture));
    return latched ? baseValue : adjusted;
}
catch (Exception ex)
{
    FaultGuard.Record(ex);
    return baseValue;        // запобіжник: повертаємо вхід
}
}

```

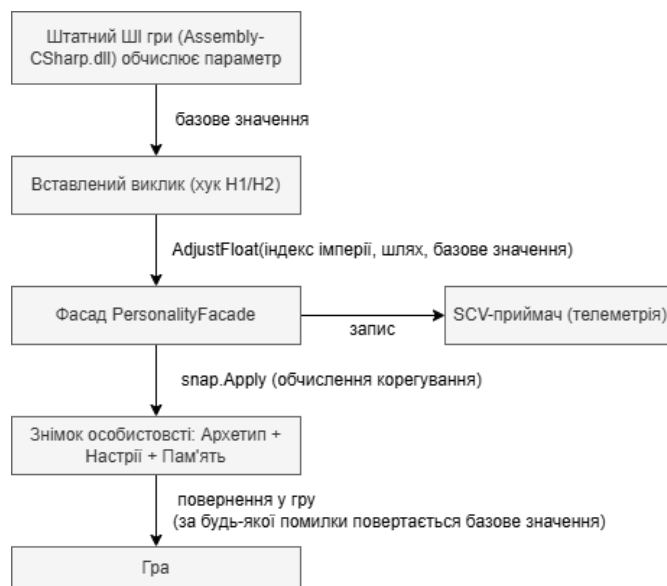


Рисунок 2.1 – Потік даних під час прийняття одного рішення

2.5 Модель особистості: реалізація трьох підсистем

Відповідно до концепції, викладеної у підрозділі 1.4, модель особистості складається з трьох взаємодіючих підсистем – статичного

архетипу, динамічного настрою та персистентної пам'яті, – реалізованих у шарі логіки особистості. Центральним класом цього шару є PersonalitySnapshot, що поєднує всі три підсистеми в єдиний коригувальний множник; його повний вихідний код наведено у додатку А.

2.5.1 Статичний архетип

Архетип є незмінним протягом партії набором даних, що описує характер агента. Він містить базові схильності у діапазоні від 0 до 1 – агресивність, довірливість, злопам'ятність, схильність до ризику та впертість; параметри кривих відгуку – гостроту реакції на загрозу, поріг використання можливостей та швидкість згасання дипломатичного терпіння; вектор бажаного типу перемоги – ваги військового, наукового, дипломатичного, економічного та дивовижного (особливий тип перемоги у грі, що потребує завершення фракційного квесту та побудову спеціальної споруди – дива) шляхів, сума яких дорівнює одиниці; а також шість поведінкових рис-особливостей, заданих булевими значеннями. Значення архетипу завантажуються із зовнішнього файлу конфігурації або з вбудованих типових значень. Параметри трьох базових архетипів, розроблених у роботі, наведено у таблиці 2.3.

Таблиця 2.3 – Базові схильності розроблених архетипів

Схильність	Завойовник	Дипломат	Економіст	Діапазон
Агресивність	0.90	0.45	0.25	[0, 1]
Довірливість	0.20	0.80	0.65	[0, 1]
Злопам'ятність	0.20	0.50	0.80	[0, 1]
Схильність до ризику	0.80	0.45	0.20	[0, 1]
Впертість	0.60	0.40	0.35	[0, 1]

Окрім схильностей, кожен архетип задає вектор бажаного типу перемоги (наприклад, у завойовника переважає військова складова, в

економіста – економічна) та набір поведінкових особливостей, які активують додаткові зсуви у відповідних точках прийняття рішень. Призначення архетипів імперіям може обмежуватися сумісністю з фракціями: для окремих фракцій певні архетипи дозволено або заборонено, що відображає їхню ігрову природу. Саме архетип є джерелом базової індивідуальності: оскільки його параметри задаються розробником, а не виводяться з оптимізації за відсотком перемог, різні архетипи не сходяться до єдиної домінуючої стратегії.

Параметри кривих відгуку уточнюють реакцію архетипу на конкретні чинники: гострота реакції на загрозу визначає, наскільки різко зростає тривога у відповідь на посилення суперника; поріг використання можливостей задає, наскільки вигідною має бути ситуація, щоб архетип нею скористався; швидкість згасання дипломатичного терпіння визначає, як швидко повертається до норми негативна реакція на дипломатичні подразники. Вектор бажаного типу перемоги задає стратегічний нахил архетипу – співвідношення військового, наукового, дипломатичного, економічного та дивовижного шляхів, – що згодом проявляється у розподілі стратегічного фокуса.

Поведінкові особливості, такі як сувора кара за зраду та вірність союзам розширюють базовий характер архетипу дискретними рисами, що активують додаткові зсуви у відповідних точках прийняття рішень. У роботі реалізовано шість таких особливостей. На відміну від неперервних схильностей, особливості є булевими: відповідний зсув або застосовується, або ні, залежно від того, чи притаманна особливість конкретному архетипу.

2.5.2 Динамічний настрій

Настрій є динамічним станом, що змінюється кожного ходу й відображає поточне ставлення агента до подій гри. Він складається з двох осей загальної поведінкової пози – військової пози, що задає поточну

схильність до військових дій, та дипломатичної відкритості, – а також з набору емоцій, які зберігаються окремо щодо кожного конкретного суперника: страху, гніву, впевненості, відчаю, довіри, поваги та образи. На початку партії осі пози ініціалізуються відповідними рисами архетипу (військова поза – агресивністю, дипломатична відкритість – довірливістю), а емоції – нейтральними чи похідними від рис значеннями.

Кожного ходу настрої оновлюються за трьома механізмами. По-перше, ігрові події спричиняють зсуви емоцій: наприклад, напад на юніта агента підвищує страх і гнів щодо нападника, причому величина зсуву масштабується гостротою реакції на загрозу, властивою архетипу. По-друге, емоції згасають у напрямку до нейтрального рівня з періодом напіврозпаду, що залежить від злопам'ятності архетипу: злопам'ятний агент зберігає гнів та образу довше. По-третє, осі пози поступово повертаються до базового рівня, заданого архетипом, що моделює заспокоєння за відсутності нових подразників. Саме цей шар забезпечує динаміку реакції на ігрові події, відсутню у скриптових системах зі статичними коефіцієнтами.

Важливою властивістю настрою є його несиметричність та індивідуальність щодо кожного суперника. Так, довіра зростає повільно, але спадає різко, що моделює природну обережність після зради. Окремі емоції згасають з різною швидкістю: страх має власний період напіврозпаду, тоді як гнів та образа згасають зі швидкістю, що залежить від злопам'ятності. Завдяки роздільному зберіганню емоцій щодо кожного суперника агент може одночасно довіряти одному гравцеві й остерігатися іншого, що неможливо у системах з єдиним глобальним станом. Повний вихідний код підсистеми динамічного настрою наведено у додатку Б.

2.5.3 Персистентна пам'ять

Пам'ять зберігає інформацію про значущі події гри щодо кожного суперника окремо й враховує її у подальших рішеннях. Кожен запис пам'яті

має параметр значущості та емоційну вагу, що зменшується з часом зі швидкістю, яка залежить від злопам'ятності архетипу та важливості події. Таким чином, один архетип швидко прощає всі образи, крім найсерйозніших, а інший пам'ятає навіть незначні інциденти до кінця партії.

Підсистема пам'яті відстежує широкий перелік типів подій – оголошення війни, захоплення міста, напад на армію, заснування міста поблизу території агента, отримання подарунка чи вигідної угоди, зраду союзу або договору, ворожі вимоги, відхилення пропозиції, укладення союзу чи миру, відгук на заклик до зброї, появу спільного ворога тощо. Приклади подій та їх характер наведено у таблиці 2.4. Накопичена пам'ять підсилює ставлення агента до конкретного суперника: так, суперник, який неодноразово оголошує війну, накопичує негативну вагу, унаслідок чого агент стає дедалі менш схильним приймати його пропозиції. Цей механізм реалізує принцип «потoku пам'яті» з архітектури переконливих агентів [12] та усуває вразливість штучного інтелекту до повторюваних маніпуляцій з боку гравця.

Таблиця 2.4 – Приклади відстежуваних подій пам'яті

Подія	Характер
Оголошення війни	негативний
Захоплення міста	негативний
Напад на армію	негативний
Зрада союзу або договору	негативний
Ворожа вимога	негативний
Заснування міста поряд із територією	негативний
Отримання подарунка	позитивний
Вигідна угода	позитивний
Укладення миру	позитивний
Відгук на заклик до зброї	позитивний

Кожна подія має знак (позитивний або негативний) та значущість, що визначає її початкову емоційну вагу. Негативні події щодо певного суперника накопичуються й поступово формують стійке упереджене ставлення до нього, тоді як позитивні події здатні частково його пом'якшити. Швидкість, з якою ця вага згасає, є індивідуальною для кожного архетипу й визначається його злопам'ятністю.

Пам'ять та настрої тісно пов'язані: значуща подія не лише фіксується у пам'яті, а й одночасно спричиняє відповідний зсув настрою. У подальшому збережений у пам'яті досвід масштабує ставлення агента до конкретного суперника у точках прийняття дипломатичних рішень – зокрема, накопичена негативна вага підвищує внутрішню «ціну» згоди на пропозиції такого суперника. Таким чином, два розрізнені, але взаємодоповнювані механізми – швидкий (настрої) і тривалий (пам'ять) – разом формують поведінку, що враховує як безпосередню реакцію на подію, так і весь попередній досвід взаємодії. Повний вихідний код підсистеми персистентної пам'яті наведено у додатку В.

2.6 Математична модель прийняття рішень

Математичною основою прийняття рішень є підхід на основі корисності [8]. Для кожного рішення обчислюється числова корисність як зважена сума значень кривих відгуку, і обирається варіант з найвищою корисністю; у випадку корегування окремого числового параметра вихідне значення кривої безпосередньо визначає величину зсуву. Центральним поняттям є крива відгуку – функція, що перетворює вхідне ігрове значення у нормоване значення корисності в діапазоні від 0 до 1.

У реалізації використано бібліотеку кривих відгуку чотирьох форм – лінійної, квадратичної, логістичної та експоненційного згасання, які описуються відповідно формулами (2.1)–(2.4):

$$f(x) = \max(0, \min(1, x)), \quad (2.1)$$

де $f(x)$ – вихідне значення (корисність) лінійної кривої відгуку;

x – вхідне ігрове значення (оцінюваний чинник).

$$f(x) = \max(0, \min(1, x^p)), \quad p = 2, \quad (2.2)$$

де $f(x)$ – вихідне значення квадратичної кривої відгуку;

x – вхідне ігрове значення;

p – показник степеня, що задає крутість кривої (типове значення $p = 2$).

$$f(x, m, k) = \frac{1}{1 + e^{-k \cdot (x - m)}}, \quad (2.3)$$

де $f(x, m, k)$ – вихідне значення логістичної кривої відгуку, $f \in (0, 1)$;

x – вхідне ігрове значення;

m – точка перегину (поріг): значення x , за якого $f = 0,5$;

k – коефіцієнт крутості переходу в околі порогу;

e – основа натурального логарифма ($e \approx 2,718$).

$$f(x, \lambda) = e^{-\lambda \cdot x}, \quad (2.4)$$

де $f(x, \lambda)$ – вихідне значення кривої експоненційного згасання;

x – вхідне значення (наприклад, кількість ходів, що минули від події);

λ – коефіцієнт швидкості згасання ($\lambda > 0$);

e – основа натурального логарифма.

Розглянемо чотири форми кривих детальніше. Спільним для всіх є те, що вони перетворюють довільне вхідне ігрове значення на нормовану корисність у діапазоні $[0, 1]$, причому вибір форми кодує якісний характер реакції агента на відповідний чинник, тоді як параметри форми задають кількісні особливості цієї реакції.

Лінійна крива (2.1) є найпростішою: вихідне значення дорівнює вхідному, обмеженому до відрізка $[0, 1]$ функціями $\min$ та $\max$. Вона описує пряму пропорційну реакцію без порогів і прискорень – корисність зростає рівномірно зі зростанням чинника, а її нахил є сталим. У роботі лінійна крива застосовується, зокрема, для оцінки можливості скористатися послабленням суперника, де відгук має бути прямо пропорційним ступеню цього послаблення.

Квадратична крива (2.2) підносить вхідне значення до степеня p (типове значення $p = 2$) з подальшим обмеженням до $[0, 1]$. На відрізку $[0, 1]$ вона є опуклою: малі значення чинника дають ще менший відгук (придушуються), тоді як зі зростанням входу нахил кривої збільшується й корисність зростає дедалі швидше, наближаючись до одиниці. Така форма моделює реакцію агента, який майже не зважає на слабкі подразники й відчутно реагує лише тоді, коли чинник стає значним; показник степеня p керує крутістю цього ефекту.

Логістична (S-подібна) крива (2.3) є центральною для моделювання порогових реакцій. Її параметр m задає точку перегину – значення входу, за якого вихід дорівнює 0,5, тобто поріг спрацювання реакції; параметр k визначає крутість переходу в околі цього порогу. За вхідних значень, значно менших за m , вихід близький до нуля, і чинник фактично ігнорується; поблизу порогу корисність різко зростає; за значень, значно більших за m , вихід насичується, наближаючись до одиниці. Завдяки цьому агент «не помічає» чинник, доки той не перетне поріг, після чого реагує різко, а далі стабілізується. Оскільки точку перегину m обчислюють з рис особистості агента, різні архетипи мають різні пороги спрацювання тієї самої реакції, що детальніше розглянуто нижче. Важливою технічною властивістю логістичної функції є те, що для будь-якого скінченного входу її значення лежить у відкритому інтервалі $(0, 1)$, тому вона не потребує додаткового обмеження і не може дати нескінченний чи невизначений результат.

Крива експоненційного згасання (2.4) є спадною: за нульового входу її значення дорівнює одиниці й монотонно зменшується до нуля зі зростанням входу, а параметр λ задає швидкість згасання – що більший λ , то швидше спадає крива (вихід зменшується вдвічі за входу $\ln 2 / \lambda$). Ця форма застосовується там, де вагомість чинника має зменшуватися з його зростанням, – зокрема, для згасання емоційної ваги записів пам'яті з плином ходів, описаного у підрозділі 2.5.3. Повну реалізацію бібліотеки кривих відгуку наведено у додатку Г.

Ключовою відмінністю від класичного utility-based підходу є те, що форма кривих є спільною, а їхні параметри визначаються особистістю агента. Наприклад, для оцінки доцільності оголошення війни використовується логістична крива від співвідношення сил, точка перегину якої обчислюється за формулою (2.5):

$$m = 1.5 - 0.5 \cdot \text{RiskTolerance}, \quad (2.5)$$

де m – точка перегину логістичної криви оцінки оголошення війни;

RiskTolerance – схильність архетипу до ризику, $\text{RiskTolerance} \in [0, 1]$.

Унаслідок цього сміливіший архетип (з вищою схильністю до ризику) оголошує війну за меншої переваги у силі, тоді як обережніший потребує більшої переваги. Підсумкова корисність рішення, що залежить від кількох чинників, обчислюється як нормована за сумою ваг зважена сума значень кривих відгуку за формулою (2.6):

$$U = \frac{\sum w_i \cdot v_i}{\sum w_i}, \quad (2.6)$$

де v_i – значення i -ї кривої відгуку;

w_i – її ваговий коефіцієнт.

Нормування за сумою ваг утримує результат у тому самому діапазоні, що й окремі значення.

Для прикладу розглянемо обчислення корисності оголошення війни. Воно поєднує три часткові оцінки – логістичну оцінку переваги у силі, військову схильність агента та оцінку можливості скористатися послабленням суперника, – кожна з яких має власну вагу, що залежить від характеру. Здобуту зважену корисність додатково помножують на множник стримування, який послаблює готовність нападати на союзника пропорційно до міцності союзу. Завдяки цьому обережний архетип з низькою схильністю до ризику потребує суттєвої переваги у силі, тоді як агресивний архетип здатний оголосити війну навіть за приблизно рівних сил.

Обчислення корисності оголошення війни як нормованої зваженої суми наведено у лістингу 2.2; повний вихідний код відповідного класу наведено у додатку Д.

Лістинг 2.2 – Корисність оголошення війни

```
internal static float DeclareWarUtility(Archetype a,
    float powerRatio, float opportunity, float bond, float
    logisticK)
{
    float m =
ResponseCurves.LogisticMidpoint(a.RiskTolerance);
    float v1 = ResponseCurves.Logistic(powerRatio, m,
logisticK);
    float v2 = a.Agression; // військова
схильність
    float v3 = ResponseCurves.Linear(opportunity);
    float w1 = PowerWeight(a);
    float w2 = MilitaryWeight();
    float w3 = OpportunityWeight(a);

    float sumWv = w1 * v1 + w2 * v2 + w3 * v3;
    float sumW = w1 + w2 + w3;
    float u = sumW <= 0f ? 0f : sumWv / sumW;

    u *= AllyRestraintFactor(a, bond); // стримування
щодо союзника
    return u;
}
```

У роботі змодельовано чотири основні точки прийняття рішень. Перша – корисність оголошення війни, що враховує перевагу у силі, військову схильність та можливість скористатися послабленням суперника, із застосуванням множника стримування щодо союзників. Друга – корегування оцінок дипломатичних угод (війна, мир, союз, перемир'я), у яке вноситься поточний настрій агента. Третя – акцент стратегічного фокуса, у якому військовий пріоритет динамічно модулюється військовою позою настрою. Четверта – оцінка прийняття контрактної пропозиції, яка за своєю природою є знаковою сумою; тому особистісний зсув для неї має бути адитивним, а не мультиплікативним, оскільки лише додавання здатне змінити знак суми, тобто рішення про прийняття чи відхилення пропозиції. Короткий опис всіх точок наведено в таблиці 2.5.

Таблиця 2.5 – Змодельовані точки прийняття рішень

Рішення	Основні чинники	Тип зсуву
Оголошення війни	Перевага у силі, військова схильність, можливість	Мультиплікативний
Оцінка угоди	Поточний настрій щодо суперника	Мультиплікативний
Стратегічний фокус	Військова поза настрою	Мультиплікативний
Прийняття контракту	Знакова сума часткових оцінок	Адитивний

Три шари моделі поєднуються таким чином: архетип задає базові параметри й ваги кривих, настроїв щоходово зсуває їх, а пам'ять масштабує ставлення до конкретного суперника. Додатково реалізовано два спеціальні механізми. Поведінкові особливості діють як булеві перемикачі, що вмикають додаткові числові зсуви у відповідних точках (наприклад, особливість «суворо карає зраду» підсилює вагу відповідного чинника ставлення). Механізм воєнної втоми реалізовано як накопичувач, що зростає кожного воєнного ходу зі швидкістю, обернено пропорційною опору, який поєднує агресивність та впертість архетипу; за досягнення високого рівня

воєнна втома підвищує готовність агента укласти перемир'я, виконуючи роль запобіжника проти нескінченних воєн.

Воєнну втому формалізовано таким чином. Опір згоранню обчислюється як опукле поєднання агресивності та впертості за формулою (2.7):

$$\text{resistance} = w \cdot \text{Aggression} + (1 - w) \cdot \text{Stubbornness}, \quad (2.7)$$

де w – ваговий коефіцієнт.

Накопичувач воєнної втоми кожного воєнного ходу зростає на величину, обернено пропорційну опорі, згідно з формулою (2.8), а кожного мирного ходу спадає на сталу величину:

$$\Delta \text{exhaustion} = \text{rate} \cdot (1 - \text{resistance}). \quad (2.8)$$

Унаслідок цього агресивні та вперті архетипи, що мають вищий опір, втомлюються від війни повільніше, що відповідає інтуїтивним уявленням про їхній характер. За досягнення накопичувачем порогового рівня формується адитивний внесок, який підвищує оцінку прийняття перемир'я й тим самим завершує затяжну війну.

Реалізацію накопичення та спаду воєнної втоми наведено у лістингу 2.3.

Лістинг 2.3 – Накопичення та спад воєнної втоми

```
// приріст за один воєнний хід до стелі: rate * (1 -
resistance)
internal static float Accrue(float current, float resistance,
                             float rate, float ceiling)
{
    float delta = rate * (1f - resistance);
    float result = current + delta;
    if (result < 0f) result = 0f;
    if (result > ceiling) result = ceiling;
    return result;
}
```

Продовження лістингу 2.3

```

}

// спад за один мирний хід до нуля
internal static float Decay(float current, float decayRate)
{
    float result = current - decayRate;
    if (result < 0f) result = 0f;
    return result;
}

```

Приклади кривих відгуку для різних архетипів наведено на рисунку 2.2.

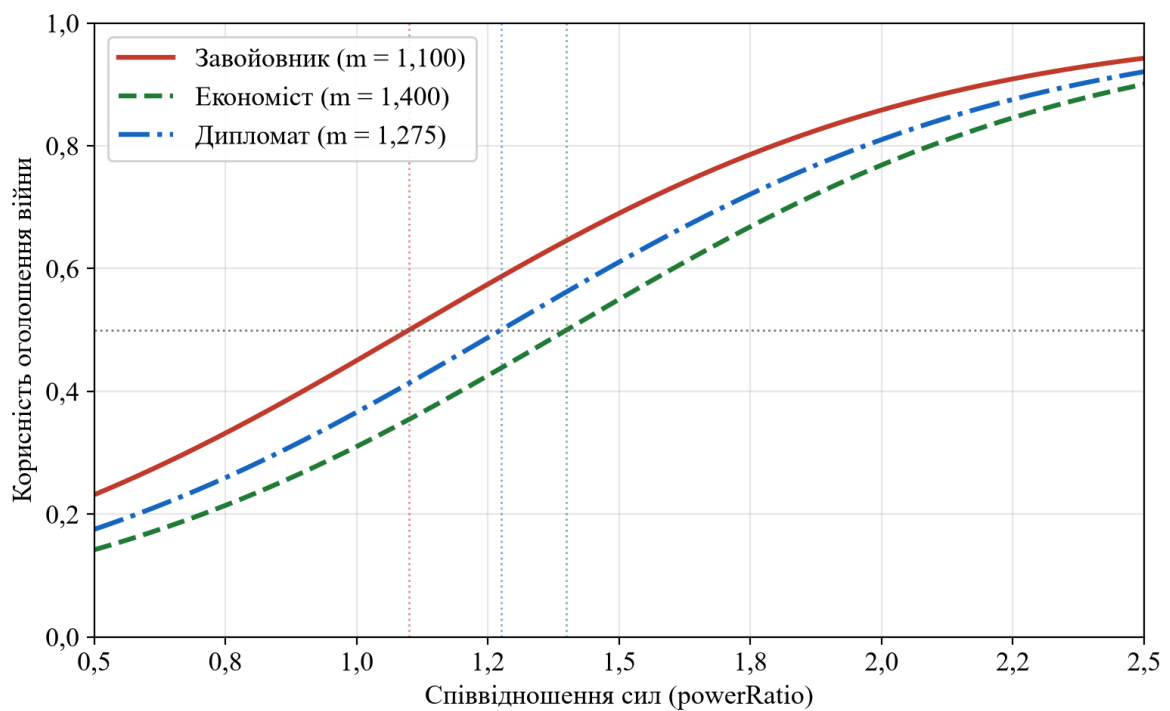


Рисунок 2.2 – Приклади кривих відгуку для різних архетипів

2.7 Інтеграція з ІІІ гри: точки вбудовування та забезпечення надійності

Інтеграцію реалізовано через вбудовування викликів фасаду у заздалегідь визначені точки логіки штатного ІІІ – так звані точки

вбудовування (hooks). Стратегія полягала у виявленні «вузьких місць» у коді ШІ, де особистісний вплив можна внести з мінімальною кількістю правок. Виокремлено три групи точок вбудовування.

Перша група (Н1) – узагальнений механізм зчитування параметрів особистості. Штатний ШІ зчитує численні налаштовувані параметри через єдиний узагальнений метод; одне вбудовування у цей метод дозволяє охопити одразу багато параметрів. Друга група (Н2) – «гарячий шлях», що виконується кожного проходу оцінювання для кожного суперника: обчислення оцінок дипломатичних агентів-доданків (війна, мир, союз, перемир'я), реєстрація стратегічного фокуса та оцінка контракту. Третя група (Н3) – приймання подій до пам'яті: на такті планувальника ШІ модифікація підписується на ігрові події та передає їх підсистемам пам'яті й настрою. Перелік точок вбудовування наведено у таблиці 2.6.

Таблиця 2.6 – Точки вбудовування у штучний інтелект гри

Група	Призначення	Що корегується	Тип впливу
Н1	Зчитування параметрів особистості	Числові параметри стратегії та ставлення	Мультиплікативний
Н2	Гарячий шлях оцінювання	Оцінки угод, фокус, контракт	Мульти. / адитивний
Н3	Приймання подій	Наповнення пам'яті та настрою	—

В усіх точках дотримується принцип «зчитати–скоригувати–повернути»: модифікація зчитує базове значення та повертає скориговане, не змінюючи внутрішній стан гри. Це гарантує неушкодженість стану партії.

Стратегію «вузьких місць» обрано тому, що вона мінімізує кількість правок у збірці гри: одне вбудовування в узагальнений механізм зчитування охоплює багато параметрів, що зменшує ризик помилки та полегшує

супровід після оновлень гри. Точки гарячого шляху обрано як такі, що виконуються для кожного суперника кожного проходу оцінювання, завдяки чому особистісний вплив проявляється послідовно протягом усієї партії, а не епізодично.

Приклад вставленого у збірку гри виклику фасаду (точка Н2 – реєстрація стратегічного фокуса) наведено у лістингу 2.4.

Лістинг 2.4 – Вставлений у збірку гри виклик фасаду

```
// оригінальний рядок у збірці гри:
this.FocusAIData.RegisterValue("PrepareWar",
    globalWarNeed * this.prepareWarScoreFactor);
// після вставлення виклику фасаду:
this.FocusAIData.RegisterValue("PrepareWar",
    PersonalityFacade.AdjustStrategicFocus(
        base.AIEntity.Empire.Index, "PrepareWar",
        globalWarNeed * this.prepareWarScoreFactor));
```

Окрему увагу приділено забезпеченню надійності, що впливає з нефункціональних обмежень середовища виконання. Методи фасаду є тотальними функціями із вбудованим запобіжником: за будь-якої помилки повертається базове значення, і жоден виняток не виходить за межі фасаду. Передбачено механізм самовимкнення: жорстка помилка ініціалізації переводить модифікацію у режим прозорого пропускання значень до кінця партії. Нескінченні та невизначені числові результати трактуються як помилки, оскільки запобіжник, орієнтований на винятки, не здатний «побачити» псування числа з рухомою комою. На шляху обчислення одного виклику виключено будь-яке виділення пам'яті, оскільки збирач сміття рушія Unity 5.3 призупиняє виконання повністю, а виділення пам'яті у кожному виклику спричиняло б ривки під час ходу. Дотримано обмеження цільової платформи 3.5; уникнуто статичних конструкторів, які у середовищі Mono здатні «отруїти» тип; усе зчитування та форматування виконується з інваріантною культурою. Нарешті, оскільки оцінювання ШІ може виконуватися у робочому потоці, тоді як події

надходять у головному потоці, увесь спільний стан модифікації захищено єдиним блокуванням, причому код гри ніколи не викликається з утримуванням цього блокування, що унеможливорює взаємне блокування (зависання гри наприкінці ходу).

Сукупність цих заходів забезпечує важливу властивість: модифікація є «прозорою за замовчуванням». За будь-якого збою – помилки конфігурації, несподіваного значення чи внутрішнього винятку – вона повертає гру до її штатної поведінки, а не порушує її. Це робить модифікацію безпечною для застосування навіть за неповної відповідності версій гри, що є істотним для практичного розгортання поза контрольованим середовищем розробки.

2.8 Конфігурація та підсистема інструментування

Поведінку модифікації визначає зовнішній файл конфігурації, що задає налаштування коефіцієнти, опис архетипів (схильності, параметри кривих, вектор перемоги та особливості), призначення архетипів імперіям і правила сумісності з фракціями. Винесення цих значень у конфігурацію дозволяє переналаштовувати поведінку без повторного збирання бібліотеки, що суттєво пришвидшує ітеративне налаштування.

Фрагмент файлу конфігурації з описом одного архетипу наведено у лістингу 2.5.

Лістинг 2.5 – Фрагмент файлу конфігурації архетипів

```
<Archetype id="Warlord">
  <Dispositions Aggression="0.90" Trustfulness="0.20"
  Grudge="0.20"
    RiskTolerance="0.80" Stubbornness="0.60" />
  <ResponseCurves ThreatOverreaction="0.30"
  OpportunismThreshold="0.35"
    DiplomaticPatienceDecay="0.70" />
  <VictoryPath military="0.55" science="0.10" diplomacy="0.05"
    economy="0.20" wonder="0.10" />
  <Quirks CourtsTheStrong="false" BalancesAgainstLeader="true"
```

Продовження лістингу 2.5

```
PunishesBetrayalHarshly="false"
ExpandsBeforeFighting="false"
ExploitsWeakness="true" HonorsAlliances="false" />
</Archetype>
```

Невіддільною частиною реалізації є підсистема інструментування. Кожен виклик фасаду записує один рядок у файл-приймач формату CSV, що містить індекс імперії, індекс суперника, ідентифікатор параметра, базове та скориговане значення, лічильник помилок і номер ходу. Окремий файл поведінкової телеметрії фіксує стратегічний стан кожної імперії на кожному ході. Це інструментування є засобом верифікації, що робить інакше непрозорі рішення ШІ спостережуваними, та становить основу для подальшого експериментального аналізу.

Завдяки винесенню поведінкових коефіцієнтів у конфігурацію переналаштування зводиться до редагування текстового файлу й перезапуску гри, без повторного збирання бібліотеки та повторного вбудовування викликів. Це утворює швидкий ітеративний цикл налаштування. Зібрані дані інструментування дозволяють не лише пересвідчитися у відсутності помилок, а й кількісно простежити, як саме особистість впливає на кожне окреме рішення.

2.9 Висновки до розділу

У розділі обґрунтовано стратегію реалізації запропонованої системи як доповнення штатного ШІ гри через пряму модифікацію її збірок із винесенням логіки особистостей в окрему супровідну бібліотеку, та проведено порівняльну оцінку механізмів інтеграції й інструментальних засобів. Окремо встановлено правомірність такої модифікації на підставі ліцензійної угоди видавця, статті 25 Закону України «Про авторське право і суміжні права» та статті 6 Директиви 2009/24/ЄС.

Описано загальну архітектуру програмної реалізації з єдиною межею фасаду та багатошаровою організацією, реалізацію тришарової моделі особистості (архетип, настрої, пам'ять) і математичну модель прийняття рішень на основі кривих відгуку, параметри яких визначаються особистістю агента. Центральною рисою цієї моделі є те, що форма кривих відгуку лишається спільною для всіх агентів, а їхні параметри й ваги задаються особистістю; саме це дає змогу досягти поведінкової різноманітності, уникнувши збіжності до єдиної домінуючої стратегії, властивій оптимізаційним підходам. Водночас статичний архетип формує базову індивідуальність, а взаємодія швидкого (настрої) і тривалого (пам'ять) шарів забезпечує динаміку поведінки, відсутню у скриптових системах зі сталими коефіцієнтами. Наведено точки вбудовування у штучний інтелект гри та засоби забезпечення надійності, а також підсистеми конфігурації й інструментування. Ключовою властивістю реалізації є прозорість за замовчуванням: усі коригувальні методи фасаду побудовано як тотальні функції, що за будь-якої помилки повертають незмінене вихідне значення гри, а в разі критичного збою ініціалізації модифікація самовимикається в режим прямого передавання, – отже, доповнення не змінює власного стану гри й не може порушити її роботу. Нарешті, розроблено методику експериментального дослідження, що ґрунтується на ігрових симуляціях з ШІ-опонентами та оцінює поведінкову різноманітність і життєздатність архетипів. Викладене у розділі відповідає задачам 2–5 постановки задачі; експериментальна валідація розробленої системи становить зміст розділу 3.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

Метою дослідження є експериментальна перевірка двох ключових властивостей системи – поведінкової різноманітності архетипів (чи демонструють вони відмінну, упізнавану поведінку) та їхньої життєздатності (чи залишається агент конкурентоспроможним). Спершу описано умови проведення експерименту й набір метрик оцінювання, далі перевірено коректність і надійність самої реалізації у середовищі гри, після чого проаналізовано поведінкову різноманітність архетипів, динаміку їхнього настрою та пам'яті й, нарешті, життєздатність. Усі кількісні показники обчислено зі зібраних даних телеметрії, приймача інструментування та підсумкової статистики збережених ігор за допомогою власних сценаріїв аналізу.

3.1 Умови та параметри проведення експерименту

Експеримент проведено у формі ігрових симуляцій з ШІ-опонентами без участі гравця-людини в ухваленні рішень. У кожній партії брали участь сім імперій: шість персоналізованих імперій під керуванням модифікованого ШІ (по дві на кожен архетип – Завойовник, Дипломат, Економіст) та одна людська імперія-наглядач, що не приймала активної участі у грі. Призначення архетипів імперіям було однаковим в усіх партіях.

Наявність однієї імперії-людини зумовлена технічним обмеженням: роздільна версія Endless Legend не дозволяє розпочати партію зовсім без гравця-людини, примусово закріплюючи за гравцем імперію з індексом 0. Цю імперію використано як пасивного погонича ходу: гравець лише завершує хід, не розвиваючи імперію, не оголошуючи воєн і не нападаючи; на дипломатичні пропозиції суперників застосовано сталу нейтральну політику, щоб ця імперія залишалася нейтральним, незмінним за

поведінкою учасником. Імперію 0 не персоналізовано й повністю виключено з оцінювання – її власний ШІ не працює, і вона не породжує телеметрії; об'єктами аналізу є виключно шість персоналізованих імперій.

Проведено три повні партії на різних картах (з різними зернами генерації світу). Для контролю впливу ігрових фракцій усі шість персоналізованих імперій в усіх партіях грали за одну й ту саму фракцію – Wild Walkers.

Під час кожної партії збиралися три типи даних: файл поведінкової телеметрії (стратегічний стан кожної персоналізованої імперії на кожному ході – зокрема військова позиція, дипломатична відкритість та злопам'ятність щодо головного суперника), файл-приймач інструментування (один рядок на кожен виклик фасаду зі скоригованими значеннями) та підсумковий файл збереження гри з повною статистикою імперій. На основі цих джерел обчислено набір метрик, описаний у наступному підрозділі.

3.2 Метрики оцінювання та очікувані напрями відмінностей

Для кількісної оцінки поведінкової різноманітності використано шість метрик, що охоплюють усі три шари моделі особистості. Важливо, що ці метрики та очікувані напрями відмінностей між архетипами було зафіксовано заздалегідь, до проведення партій, що надає аналізу характеру перевірки гіпотез, а не підбору пояснень під отримані дані. Перелік метрик, спосіб їх обчислення та очікуваний напрям наведено у таблиці 3.1.

Таблиця 3.1 – Метрики оцінювання та очікувані напрями відмінностей

Метрика	Спосіб обчислення	Очікуваний напрям
Кількість суперників із воєнними намірами	К-сть суперників, для яких оцінка угоди «Війна» перетнула поріг залученості	Завойовник >> Економіст

Продовження таблиці 3.1

Метрика	Спосіб обчислення	Очікуваний напрям
Хід першого воєнного наміру	Найраніший хід, коли цей поріг уперше перетнуто	Завойовник найраніше
Військ./екон. рахунок	Військовий рахунок, поділений на загальний, на останньому ході	Завойовник найвищий
Пікова злопам'ятність	Максимум ваги злопам'ятності щодо головного суперника за партію	Економіст > Дипломат > Завойовник
Середня дипломатична відкритість	Середнє значення осі відкритості за всю партію	Дипломат > Економіст > Завойовник
Фінальний рахунок і результат	Загальний рахунок на останньому ході та підсумок (вибула / дожила)	без напрямку (описово)

Окремо наголосимо, що останню метрику – фінальний рахунок – внесено як описову, без зафіксованого напрямку. Підсумковий рахунок є ненадійним показником поведінкових відмінностей, оскільки різні за характером архетипи можуть досягати схожого результату різними шляхами; цю метрику наведено лише для оцінки життєздатності, а не різноманітності.

3.3 Перевірка коректності та надійності реалізації

Перш ніж аналізувати поведінку, перевірено коректність і надійність самої реалізації безпосередньо у середовищі гри. Ключовою вимогою до модифікації є її прозорість і безвідмовність: за весь час партії жоден виклик фасаду не повинен породжувати помилки, а механізм самовимкнення не повинен спрацьовувати. Зведені показники надійності за трьома партіями наведено у таблиці 3.2.

За трьома партіями модифікація обробила понад 20 мільйонів викликів фасаду (близько 6,6–7,0 мільйонів за партію) з нульовим

лічильником помилок ($\text{faultCount} = 0$ у кожному рядку) та без жодного хибно сформованого рядка телеметрії. Перевірка «серцебиття» (heartbeat) підтвердила, що особистісне коригування стратегічного фокуса спрацювало щонайменше раз на кожному ході для всіх шести персоналізованих імперій, а перевірка повноти – що в усіх партіях задіяно всі чотири дипломатичні угоди (Війна, Мир, Союз, Перемир'я). Механізм самовимкнення жодного разу не спрацював. Це підтверджує властивість «прозорості за замовчуванням», закладену в архітектуру (підрозділ 2.7): у реальному середовищі Mono модифікація працювала безвідмовно протягом усіх партій.

Таблиця 3.2 – Показники надійності виконання за партіями

Партія	Ходів	Викликів фасаду	Хибних рядків	faultCount	Перевірки
1	200	6 686 183	0	0	серцебиття 6/6, угоди 4/4, без самовимкнення
2	200	6 972 378	0	0	серцебиття 6/6, угоди 4/4, без самовимкнення
3	196	6 580 831	0	0	серцебиття 6/6, угоди 4/4, без самовимкнення

Дані приймача також підтверджують, що особистісні коригування застосовувалися саме за архетипами. Наприклад, коефіцієнт підготовки до війни ($\text{PrepareWarScoreFactor}$) множився на 1,7 у Завойовників, на 0,63 у Дипломатів і на 0,49 в Економістів, що відповідає спадній агресивності; коефіцієнт економічного рахунку підвищувався в 1,5 рази саме в Економістів; а швидкість згасання емоційної ваги подій пам'яті була різною (множник близько 0,84 у Завойовника, 1,02 у Дипломата та 1,1 в Економіста), що відповідає зростанню злопам'ятності від Завойовника до Економіста. Отже, особистісний вплив не лише є безвідмовним, а й застосовується диференційовано відповідно до архетипу.

3.4 Поведінкова різноманітність архетипів

Зведені значення шести метрик за архетипами (середнє по шести спостереженнях на архетип, у дужках – діапазон від мінімуму до максимуму) наведено у таблиці 3.3.

Таблиця 3.3 – Зведені поведінкові метрики за архетипами

Метрика (середнє [мін–макс])	Завойовник	Дипломат	Економіст
Суперників із воєнними намірами	5,3 [3–6]	5,0 [4–6]	4,0 [2–6]
Хід першого воєнного наміру	47,5 [31–73]	62,0 [41–79]	52,7 [43–68]
Військ./екон. рахунок	17,0 [0–27]	28,9 [10–89]	18,5 [0–39]
Пікова злопам'ятність	0,80 [0,73–0,81]	0,98 [0,90–1,00]	0,99 [0,98–1,00]
Середня дипл. відкритість	0,08 [0,05–0,09]	0,63 [0,59–0,70]	0,38 [0,30–0,49]
Фінальний рахунок	1110 [205–1801]	841 [393–1668]	1150 [135–2509]

Найвиразніше архетипи розділяє військова поза – вісь настрою, що задає поточну схильність до військових дій. Її базовий рівень визначається агресивністю архетипу й становить 0,90 у Завойовника, 0,45 у Дипломата та 0,25 в Економіста; під тиском війни всі архетипи здатні підняти позу до близького до 1,0, проте їхній «спокій» – рівень, до якого поза повертається за відсутності загроз, – лишається чітко розмежованим. Кількість суперників, проти яких архетип сформував воєнні наміри (рисунок 3.1), узгоджується з цим: Економіст послідовно найменш войовничий (у середньому 4,0 суперника проти 5,3 у Завойовника), тоді як Завойовник і Дипломат близькі між собою – Дипломат, попри помірну агресивність, активно вступає у війни через риси-особливості (балансування проти лідера, сувора кара за зраду).

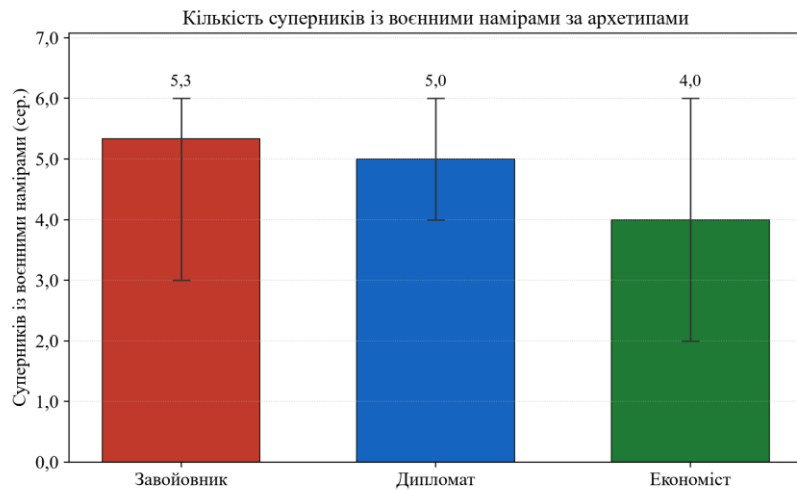


Рисунок 3.1 – Кількість суперників із воєнними намірами за архетипами

Найчистіше розмежування дає дипломатична відкритість (рисунок 3.2): Дипломат демонструє найвищу відкритість, Економіст – проміжну, Завойовник – найнижчу, причому цей порядок виконується в усіх трьох партіях без винятку (у першій – на рівні статичних базових значень 0,80 / 0,65 / 0,20, у другій і третій – на рівні динамічних середніх 0,63 / 0,38 / 0,08). Це безпосередньо відображає довірливість архетипів.

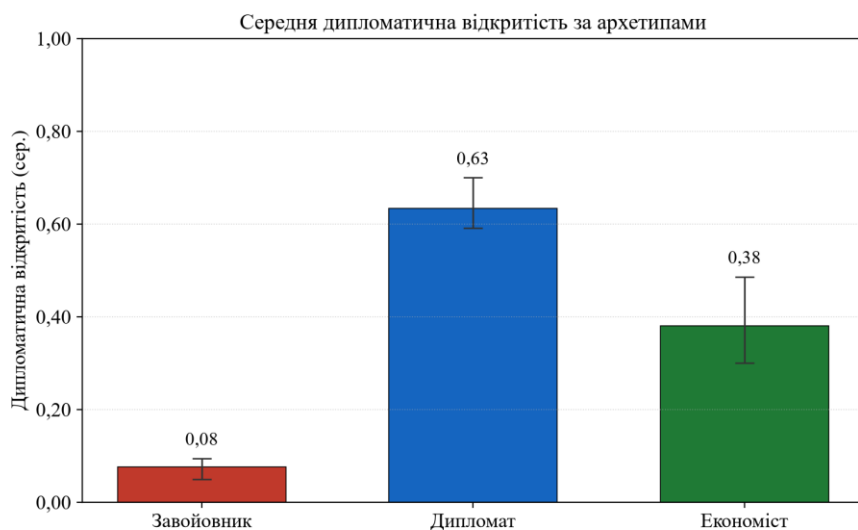


Рисунок 3.2 – Середня дипломатична відкритість за архетипами

Так само виразним є розмежування за піковою злопам'ятністю (рисунок 3.3): Завойовник має найнижчу злопам'ятність (у середньому 0,80 і жодного разу не сягає максимуму), тоді як Дипломат та Економіст накопичують значно вищі піки (0,98 і 0,99 відповідно). Це прямо відповідає рисі злопам'ятності архетипів (0,20 / 0,50 / 0,80) і механізму згасання пам'яті: Завойовник швидко «прощає» образи, тоді як Економіст пам'ятає їх найдовше.

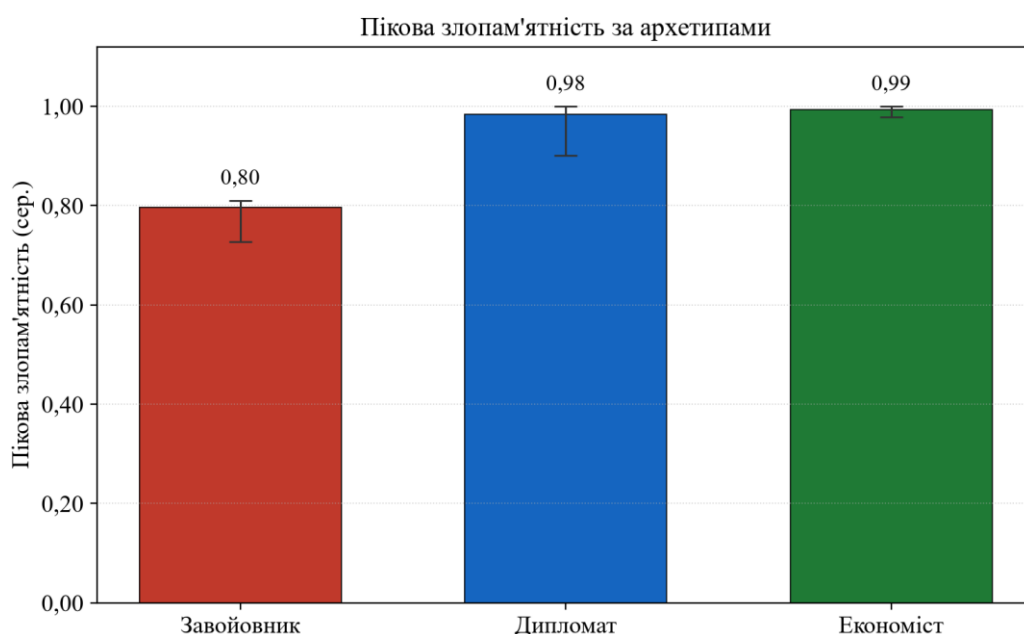


Рисунок 3.3 – Пікова злопам'ятність за архетипами

Натомість метрика співвідношення військового та економічного рахунку (рисунок 3.4) очікуваного розмежування не дала: середні значення архетипів близькі, а їхні діапазони значно перекриваються (зокрема, в окремих партіях найвищий показник мав Економіст або Дипломат, а не Завойовник). Це не суперечить гіпотезі, а підтверджує застереження методики: показники, похідні від підсумкового рахунку, є ненадійними індикаторами поведінки, оскільки економічно сильний Економіст здатний наприкінці партії наростити велику армію, а знищений Завойовник –

втратити її. Саме тому основою оцінювання різноманітності слугують динамічні поведінкові ознаки (поза, відкритість, злопам'ятність, момент і кількість воєнних намірів), а не агреговані підсумкові величини.

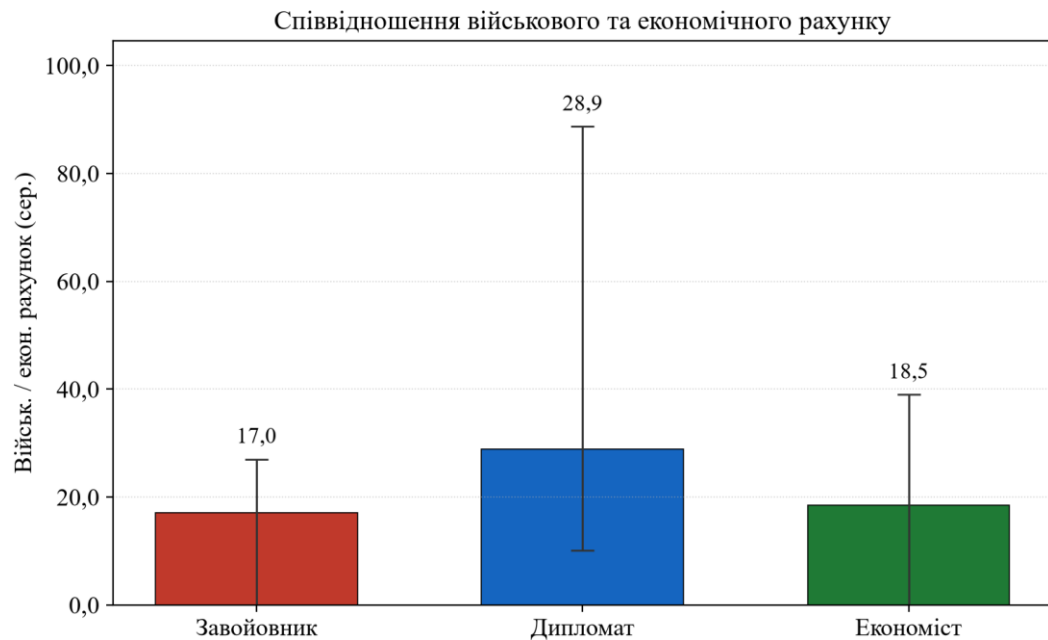


Рисунок 3.4 – Співвідношення військового та економічного рахунку за архетипами

Перевірку відповідності спостережуваних відмінностей очікуваним напрямам за окремими партіями зведено у таблиці 3.4. Чотири з п'яти напрямних метрик виконуються (з них дипломатична відкритість і пікова злопам'ятність – в усіх партіях), а єдиний виняток – похідний від рахунку коефіцієнт – пояснено вище.

Таблиця 3.4 – Перевірка очікуваних напрямів за партіями

Очікуваний напрям	Партія 1	Партія 2	Партія 3	Підсумок
Воєнні наміри: $З \geq Д > Е$	$З=Д>Е$	$З>Д>Е$	$З=Д>Е$	виконується (Е завжди найменше)

Продовження таблиці 3.4

Очікуваний напрям	Партія 1	Партія 2	Партія 3	Підсумок
Перший хід війни: 3 найраніше	так	так	$\approx$	переважно виконується
Злопам'ятність: $E \geq D$ > 3	$E=D>3$	$E=D>3$	$E>D>3$	виконується (3 завжди найнижче)
Відкритість: $D > E > 3$	так	так	так	виконується в усіх партіях
Військ./екон.: 3 найвищий	так	ні	ні	не виконується (шумна метрика)

Найсуворішим критерієм різноманітності методика визначає сліпе зіставлення міток – здатність впізнати архетип лише за його поведінковим слідом. Стійкість напрямів ранжування за незалежними поведінковими осями (відкритість, злопам'ятність, войовничість) фактично і є таким зіставленням: знаючи лише профіль агента за цими осями, його архетип можна впевнено визначити (висока відкритість і помірна злопам'ятність – Дипломат; низька відкритість, висока войовничість і швидке прощення – Завойовник; найвища злопам'ятність за помірної войовничості – Економіст). Отже, відмінності між архетипами є не лише наявними, а й змістовно впізнаваними.

3.5 Динаміка настрою та пам'яті: приклад партії

Окрім статичних відмінностей між архетипами, реалізована система забезпечує динаміку поведінки впродовж партії – саме цього бракує скриптовим системам зі сталими коефіцієнтами. Для ілюстрації на рисунку 3.5 наведено повну траєкторію трьох ключових величин одного агента-Дипломата (імперія 3 у другій партії): військової пози, дипломатичної відкритості та злопам'ятності щодо головного суперника.

Траєкторія наочно демонструє взаємодію швидкого (настрій) і тривалого (пам'ять) шарів. На початку партії агент мирний: військова поза

тримається на базовому рівні 0,45, а відкритість – висока (близько 0,8). Із накопиченням образ відкритість поступово знижується, а злопам'ятність зростає, причому на графіку чітко видно пилюкоподібний характер цих змін – різкі зсуви у відповідь на події та плавне згасання між ними. Близько 90-го ходу, з початком війни, військова поза різко зростає до максимуму, а відкритість падає. Показовим є проміжок близько 140-го ходу: після тимчасової деескалації відкритість частково відновлюється, тоді як накопичена злопам'ятність спадає лише незначно – це ілюструє асиметрію шарів, за якої швидкий настрій повертається до норми, а тривала пам'ять утримує досвід. У завершальній фазі затяжної війни відкритість обвалюється до близько 0,05, а злопам'ятність сягає максимуму.

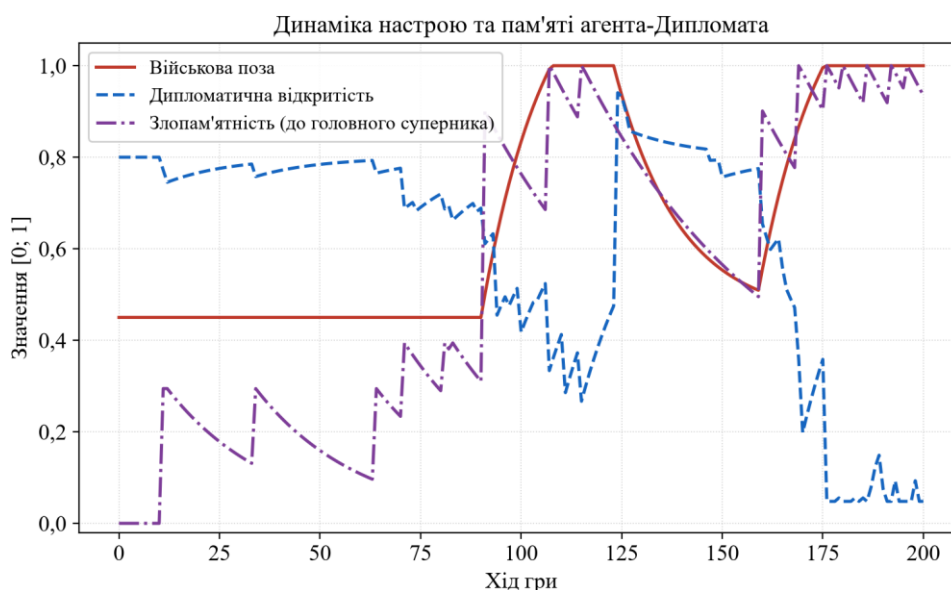


Рисунок 3.5 – Динаміка настрою та пам'яті агента-Дипломата

Ця траєкторія також спростовує проблему, що спостерігалася на ранніх етапах розробки, – насичення злопам'ятності на максимумі в усіх агентів. У підсумкових партіях злопам'ятність є градуйованою за архетипом: Завойовники взагалі не досягають максимуму (пік близько 0,81), а Дипломати й Економісти, навіть досягнувши 1,0 під час затяжної війни,

згодом частково знижують її в періоди миру. Отже, механізм пам'яті працює коректно, відображаючи як накопичення, так і згасання досвіду.

3.6 Життєздатність архетипів

Життєздатність оцінювалася за здатністю архетипів зберігати конкурентоспроможність і доживати до пізніх стадій партії, не вибуваючи систематично першими. Підсумки трьох партій наведено у таблиці 3.5.

Таблиця 3.5 – Підсумки партій: вибуття та лідер за фінальним рахунком

Партія	Вибула імперія (архетип)	Лідер за фінальним рахунком (архетип)	Рахунок лідера
1	імперія 5 – Економіст	імперія 2 – Завойовник	1801
2	імперія 1 – Завойовник	імперія 6 – Економіст	2509
3	немає	імперія 4 – Дипломат	1668

За трьома партіями жоден архетип не вибуває систематично першим. Дипломати не вибули жодного разу; Економіст і Завойовник втратили по одній імперії (відповідно у першій і другій партії), причому в третій партії всі шість персоналізованих імперій дожили до кінця. Прикметно, що лідером за фінальним рахунком у трьох партіях ставали представники трьох різних архетипів – Завойовник, Економіст і Дипломат відповідно, – і жоден архетип не домінував за рахунком. Це узгоджується з висновком підрозділу 3.4 про ненадійність рахунку як поведінкового показника й водночас підтверджує життєздатність усіх трьох архетипів.

3.7 Висновки до розділу

У розділі наведено результати експериментальної валідації розробленої системи динамічних особистостей. Перевірку проведено на

трьох повних партіях з ШІ-опонентами за контрольованих умов (однакова фракція суб'єктів, заздалегідь зафіксовані метрики та очікувані напрями).

Підтверджено надійність реалізації: за понад 20 мільйонів викликів фасаду в трьох партіях не зафіксовано жодної помилки чи спрацювання самовимкнення, а особистісні коригування застосовувалися безвідмовно й диференційовано за архетипами, що підтверджує закладену властивість прозорості за замовчуванням у реальному середовищі гри.

Підтверджено поведінкову різноманітність архетипів: чотири з п'яти напрямних метрик відповідають заздалегідь зафіксованим очікуванням, причому дипломатична відкритість та пікова злопам'ятність розмежовують архетипи в усіх партіях, а військова поза – за самою своєю побудовою. Єдиний виняток – похідний від підсумкового рахунку коефіцієнт – лише підтверджує застереження методики щодо ненадійності агрегованих показників. Архетипи виявилися впізнаваними за їхніми поведінковими слідами, а приклад окремої партії продемонстрував коректну динамічну взаємодію швидкого (настрій) і тривалого (пам'ять) шарів.

Підтверджено життєздатність архетипів: жоден з них не вибуває систематично першим, а лідерами партій ставали представники всіх трьох архетипів, у тому числі найменш войовничий Економіст. Отже, особистісно зумовлена, подеколи субоптимальна гра залишається конкурентоспроможною. Викладене відповідає задачам 6–7 постановки задачі.

4 НАПРЯМИ ПОДАЛЬШОГО РОЗВИТКУ

Розроблену в роботі систему динамічних особистостей реалізовано й експериментально підтверджено у вигляді вручну налаштованого базового варіанта. У цьому розділі окреслено напрями її подальшого розвитку. Головним з них є автоматизація створення архетипів за допомогою еволюційного алгоритму типу «якість – різноманіття», що дозволив би відкривати ширший набір характеристик замість ручного підбору. Другим напрямом є відображення внутрішнього настрою ІІІ-гравців безпосередньо гравцеві-людині через динамічні повідомлення. Розділ завершується переліком менших напрямів розширення.

4.1 Автоматизована генерація особистостей за допомогою алгоритму «якість – різноманіття»

4.1.1 Передумови та мотивація

Кожен архетип задається кількома десятками числових параметрів – базовими схильностями, параметрами кривих відгуку, вектором бажаного типу перемоги та набором поведінкових особливостей. Підбір таких значень, які забезпечують водночас виразно відмінну й життєздатну поведінку, у роботі виконувався вручну. Цей спосіб є повільним, значною мірою суб'єктивним і погано масштабується на більший набір архетипів.

Природним способом автоматизації видається еволюційний алгоритм – генетичний пошук значень параметрів за певною функцією придатності. Проте прямолінійне застосування звичайного генетичного алгоритму з оптимізацією за відсотком перемог (win-rate) відтворило б саме те обмеження, яке описано у підрозділі 1.3.2: оптимізаційний пошук сходиться до єдиної домінуючої МЕТА-стратегії й поступово витісняє поведінкові відмінності, що знижують win-rate. Інакше кажучи, оптимізація

за перемогами знищила б саме те різноманіття, заради якого систему й створено. Тому потрібен принципово інший клас алгоритмів.

4.1.2 Сутність підходу «якість – різноманіття»

Таким класом є алгоритми «якість – різноманіття» (Quality – Diversity, QD) – родина еволюційних методів, що шукають не єдиний оптимум, а різноманітний архів достатньо добрих рішень, які покривають увесь простір можливих варіантів поведінки [21]. Замість максимізації однієї цільової функції QD-алгоритми винагороджують рішення за відмінність від уже знайдених за умови збереження прийнятної якості.

Концептуальною основою QD є пошук новизни (novelty search) – підхід, що повністю відмовляється від градієнта цільової функції й винагороджує особин лише за поведінкову новизну відносно архіву раніше знайдених рішень, завдяки чому уникає передчасної збіжності до локальних оптимумів [22]. Доповнення пошуку новизни складником локальної конкуренції за якістю й утворює власне підхід «якість – різноманіття». Споріднений метод MAP-Elites дискретизує простір поведінок на ніші й зберігає в кожній найкраще («елітне») знайдене рішення, тим самим «висвітлюючи» розподіл якості за всім простором поведінок [23].

Ключовим є перевизначення мети пошуку: різноманіття стає основною метою, а якість (життєздатність) – обмеженням-порогом, а не максимізованою величиною. Це безпосередньо відповідає проектній меті роботи (підрозділ 1.4), у якій поведінкові відмінності між архетипами є метою, а не побічним наслідком.

4.1.3 Запропонована схема пошуку

На основі підходу QD пропонується така схема автоматичної генерації архетипів. Одиницею еволюції є не окремий архетип, а цілий

ансамбль (ростер) із чотирьох–шести архетипів, оскільки різноманіття є властивістю набору, а не окремого його члена. Популяцію утворюють 30 – 50 таких ростерів.

Придатність ростера обчислюється за двома осями. Перша – поведінкове різноманіття: середня попарна відстань між векторами поведінкових метрик архетипів усередині ростера (а в межах усієї популяції – розкид кожної метрики). Як вектор поведінкових ознак використовуються самі метрики, запроваджені у розділі 3 (кількість суперників із воєнними намірами, співвідношення військового та економічного рахунку, пікова злопам'ятність, середня дипломатична відкритість тощо). Друга вісь – життєздатність, що є саме порогом, а не ціллю: кожен архетип має лишатися достатньо конкурентоспроможним, щоб вигравати щонайменше близько 15–20% ігор (за випадкового паритету близько 25%), а архетип, що не сягає цього порога, дістає істотний штраф. Дві осі поєднуються мультиплікативно (різноманіття $\times$ множник життєздатності), тож відсоток перемог фігурує лише як поріг і ніколи не є величиною, яку максимізують.

Оцінювання придатності спирається на вже наявну в роботі інфраструктуру: кожен ростер проганяється через пакет автоматичних ігор з ШІ-опонентами, а його поведінкові метрики та життєздатність зчитуються з тієї самої телеметрії й підсумкової статистики, що використовувалися для валідації у розділі 3. Еволюційний цикл має такий вигляд: змішане засівання початкової популяції (частина – збурені копії прототипних архетипів, частина – випадкові, частина – результати схрещення), прогін пакетів ігор, обчислення двоосової придатності, відбір (турнірний або пропорційний), схрещення на рівні ростерів та гаусова мутація параметрів – і так упродовж 10–20 поколінь. Підсумково з фінальної популяції витягуються кластерні представники, які й стають іменованими архетипами, відкритими пошуком, а не заданими вручну. Концептуальну відмінність звичайної

оптимізації від підходу «якість – різноманіття» проілюстровано на рисунку 4.1.

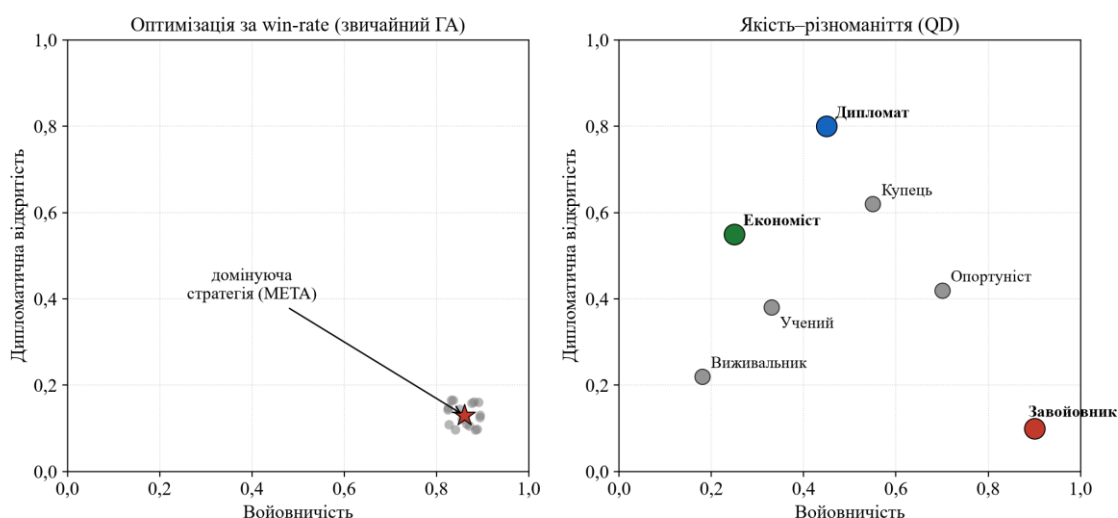


Рисунок 4.1 – Збіжність звичайної оптимізації проти заповнення простору поведінок алгоритмом «якість – різноманіття»

На лівій панелі рисунка звичайний генетичний алгоритм, оптимізуючи відсоток перемог, стягує всіх кандидатів до однієї домінуючої точки простору поведінок (META-стратегії), знищуючи різноманіття. На правій панелі підхід «якість – різноманіття» натомість заповнює простір поведінок набором життєздатних, але виразно відмінних архетипів, що займають різні поведінкові ніші.

4.1.4 Очікуваний результат і виклики

Очікуваним результатом такого пошуку є ширший за трійку базового варіанта набір архетипів (наприклад, окрім Завойовника, Дипломата та Економіста – Учений, Купець, Опортуніст, Виживальник тощо), параметри яких відкрито автоматично й підтверджено життєздатними. Центральною методологічною тезою підходу є те, що оптимізація на META-рівні – пошук

найкращих дизайнів характерів – здатна збільшувати поведінкове різноманіття, тоді як оптимізація на рівні самої гри його знищила б.

Головним викликом є обчислювальна вартість: оцінювання придатності одного ростера потребує цілого пакета повних ігор, а пошук загалом – тисяч симуляцій. Це, по-перше, вимагає істинного режиму гри без гравця-людини (наявне обмеження, описане у підрозділі 3.1, де доводилося використовувати пасивного гравця-погонича ходу); по-друге – пришвидщеної або «безголової» (headless) симуляції, а можливо, і сурогатних моделей придатності для попереднього відсіювання невдалих ростерів. Додатковим напрямом є фракційно залежні архіви – окремий пошук для різних ігрових фракцій. Важливо, що передумови для такого пошуку вже закладено в реалізації: винесення всіх поведінкових параметрів у зовнішній файл конфігурації (підрозділ 2.8) дозволяє змінювати їх без перезбирання бібліотеки, що є необхідною умовою автоматичного перебору параметрів у тисячах симуляцій.

4.2 Відображення настрою ІІІ-гравців для гравця

Другим напрямом є підвищення «видимості» особистості для гравця-людини. У поточній реалізації система динамічних особистостей є, по суті, невидимою підсистемою: багатий внутрішній стан – настрої і пам'ять – впливає на рішення ІІІ, проте безпосередньо гравцеві не відображається. У цій роботі цей стан зроблено спостережуваним лише для дослідника – через телеметрію та рисунки, але не для гравця під час самої гри.

Перспективним розширенням є динамічні внутрішньоігрові повідомлення ІІІ-гравців, що відображають їхній поточний настрій. Дипломатичні репліки та сповіщення можна формувати на основі осей настрою (військова поза, дипломатична відкритість) і поемоційного стану щодо конкретного суперника (гнів, страх, довіра, образа) разом із накопиченою злопам'ятністю: ворожий, злопам'ятний агент надсилав би

загрозливі повідомлення, наляканий – прохальні чи відчайдушні, а Дипломат із високою відкритістю – доброзичливі пропозиції. Це зробило б особистість агента більш виразною безпосередньо у грі та посилило б ефект «читабельної дуги», коли гравець може простежити, як ставлення суперника змінюється протягом партії.

Таке розширення безпосередньо відповідає меті створення переконливих, емоційно реактивних агентів, що спирається на дослідження переконливих агентів [11], [12]. Окрім того, воно відкрило б шлях до прямого оцінювання переконливості – властивості, яку в цій роботі, на відміну від різноманітності та життєздатності, не вимірювали безпосередньо. З погляду реалізації цей напрям є низькоризиковим: формування повідомлень лише зчитує наявний знімок особистості й не змінює стану гри, тобто не порушує закладеного принципу «зчитати–повернути».

4.3 Інші напрями розвитку

Окрім двох головних напрямів, можливими є такі менші розширення. По-перше, побудова поведінкового шару: задіяння вже каталогізованих, але не під'єднаних рис-особливостей (зокрема остраху війни на два фронти та захисту слабших), додавання контр-інтуїтивного «показового» архетипу, а також механізму глобального узагальнення недовіри, за якого серйозна зрада знижувала б довіру до всіх суперників, а не лише до її безпосереднього винуватця.

По-друге, розширення моделі стану: збереження й відновлення настрою та пам'яті між сесіями гри (наразі цей стан існує лише в межах однієї партії). По-третє, строге профілювання продуктивності – вимірювання витрат часу на хід у повній багатоімперній грі пізньої стадії, яке в цій роботі виконано лише якісно. По-четверте, глибші еволюційні розширення – еволюція не лише параметрів, а й структури самих кривих

відгуку, а також додавання шару рефлексії, за якого агент періодично переглядав би власну стратегію.

Нарешті, окремим напрямом є перенесення запропонованої архітектури на інші ігри. Оскільки систему спроектовано як надбудову над утилітарним ШІ й вона не залежить від конкретних ігрових механік, її принципи можуть бути адаптовані як для інших 4X-стратегій, так і для NPC-агентів в іграх інших жанрів – рольових іграх, тактичних стратегіях та симуляторах.

4.4 Висновки до розділу

У розділі окреслено напрями подальшого розвитку запропонованої системи. Головним визначено автоматизовану генерацію особистостей за допомогою еволюційного алгоритму «якість – різноманіття»: на відміну від звичайної оптимізації за відсотком перемог, яка сходиться до єдиної МЕТА-стратегії, QD-пошук – реалізований як пошук новизни з локальною конкуренцією над ансамблями архетипів із двоосьовою придатністю «різноманіття $\times$ поріг життєздатності» – здатний автоматично відкривати ширший набір виразно відмінних і водночас життєздатних характеристик, повторно використовуючи поведінкові метрики та інфраструктуру самогри з розділу 3.

Другим напрямом визначено відображення настрою ШІ-гравців гравцеві через динамічні повідомлення, що зробило б особистість більш виразною у грі та уможливило б оцінювання переконливості. Окреслено також низку менших розширень – від добудови поведінкового шару й збереження стану до перенесення архітектури на інші ігри. Реалізоване у роботі рішення є придатним фундаментом для всіх цих напрямів: винесення параметрів у конфігурацію, набір поведінкових метрик і безвідмовна, прозора інтеграція з грою безпосередньо уможливлюють подальший автоматизований пошук та розширення.

ВИСНОВКИ

Результатом роботи є вирішення задачі підвищення поведінкової виразності та різноманітності штучного інтелекту опонентів у 4X стратегічних іграх – проблему, зумовлену передбачуваністю наявних рішень і відсутністю в них персистентної пам'яті та емоційної динаміки. Запропоновано, спроектовано й реалізовано систему динамічних особистостей для гри Endless Legend, яка не замінює штатний ШІ гри, а доповнює його, зміщуючи стратегічні та дипломатичні рішення відповідно до заданого характеру агента.

Проаналізовано предметну галузь 4X стратегій і сучасні підходи до реалізації ШІ-опонентів – системи на основі статичних коефіцієнтів, скінченні автомати, підхід на основі корисності, навчання з підкріпленням, великі мовні моделі та дослідження переконливих агентів. Виявлено їх ключові обмеження: передбачуваність скриптових систем, збіжність оптимізаційних підходів до єдиної домінуючої МЕТА-стратегії, відсутність пам'яті про події гри, відсутність динаміки емоційного стану та значні обчислювальні витрати. Це обґрунтувало вибір власного підходу, у якому поведінкові відмінності закладаються як проектна мета, а не виводяться з оптимізації за відсотком перемог.

Розроблено тришарову архітектуру особистості, що поєднує статичний архетип характеру, динамічний настрій і персистентну пам'ять, та математичну модель прийняття рішень на основі кривих відгуку, параметри яких визначаються поточним станом усіх трьох шарів. На цій основі створено три базові архетипи – Завойовник, Дипломат та Економіст – з визначеними значеннями параметрів, що забезпечують виразно відмінну поведінку.

Запропоновану систему реалізовано як модифікацію гри Endless Legend через декомпіляцію, точкове редагування та рекомпіляцію керованих .NET-збірок засобом dnSpyEx, із винесенням логіки

особистостей в окрему супровідну бібліотеку (.NET 3.5) з єдиним незмінним інтерфейсом-фасадом. Особливу увагу приділено надійності: методи фасаду є тотальними функціями із запобіжником, передбачено самовимкнення за критичної помилки, а вся взаємодія підпорядкована принципу «зчитати–скоригувати–повернути», що гарантує неушкодженість стану партії.

Проведено експериментальну валідацію на трьох повних партіях у режимі гри між ІІІ-опонентами. Підтверджено надійність реалізації: за понад 20 мільйонів викликів фасаду не зафіксовано жодної помилки (`faultCount = 0`) чи спрацювання самовимкнення, а особистісні коригування застосовувалися безвідмовно й диференційовано за архетипами. Підтверджено поведінкову різноманітність: чотири з п'яти заздалегідь зафіксованих напрямів відмінностей виконалися. Підтверджено й життєздатність: жоден архетип не вибував систематично першим, а лідерами партій за фінальним рахунком ставали представники всіх трьох архетипів.

Отримані результати свідчать, що легка, прозора надбудова над утилітарним ІІІ здатна породжувати різноманітних, життєздатних і впізнаваних за поведінкою опонентів. Тим самим підтверджено центральну тезу роботи: поведінкові відмінності можна закласти як мету проєктування, не вдаючись до оптимізації за перемогами, яка нівелювала б ці відмінності. Практичне значення роботи полягає в тому, що запропонована архітектура, незалежна від конкретних ігрових механік, може бути адаптована як для інших 4X стратегій, так і для неігрових персонажів в іграх інших жанрів.

Визначено напрями подальшого розвитку: автоматизована генерація архетипів за допомогою еволюційного алгоритму «якість – різноманіття», що шукав би різноманітний набір життєздатних характеристик замість ручного підбору, та відображення внутрішнього настрою ІІІ-гравців безпосередньо гравцеві через динамічні внутрішньоігрові повідомлення. Усі поставлені задачі виконано, а мету роботи – досягнуто.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Emrich A. MicroProse' strategic space opera is rated XXXX. *Computer gaming world*. 1993. No. 110. P. 92–93.
2. Dor S. Strategy in games or strategy games: dictionary and encyclopaedic definitions for game studies. *Game studies*. 2018. Vol. 18, no. 1. URL: https://gamestudies.org/1801/articles/simon_dor (date of access: 14.05.2026).
3. Agenda (civ6). *Civilization Wiki*. URL: [https://civilization.fandom.com/wiki/Agenda_\(Civ6\)](https://civilization.fandom.com/wiki/Agenda_(Civ6)) (date of access: 17.05.2026).
4. Fulton W. Civilization VI brings sprawling cities, complex combat, cunning enemies. *Digital Trends*. URL: <https://www.digitaltrends.com/gaming/civilization-vi-first-details/> (date of access: 14.05.2026).
5. Adaptive game AI with dynamic scripting / P. Spronck et al. *Machine learning*. 2006. Vol. 63, no. 3. P. 217–248. URL: <https://doi.org/10.1007/s10994-006-6205-6> (date of access: 14.05.2026).
6. Sekhavat Y. A. Behavior trees for computer games. *International journal on artificial intelligence tools*. 2017. Vol. 26, no. 02. P. 1730001. URL: <https://doi.org/10.1142/s0218213017300010> (date of access: 14.05.2026).
7. Yannakakis G. N., Togelius J. Artificial intelligence and games. Cham : Springer Nature Switzerland, 2025. URL: <https://doi.org/10.1007/978-3-031-83347-2> (date of access: 17.05.2026).
8. Behavioral mathematics for game AI. Boston, MA : Charles River Media, Course Technology, Cengage Learning, 2009. 459 p.
9. Vox deorum: a hybrid LLM architecture for 4X / grand strategy game AI – lessons from civilization V / J. Chen et al. *arXiv*. 2025. URL: <https://doi.org/10.48550/arXiv.2512.18564> (date of access: 17.05.2026).

10. CivRealm: a learning and reasoning odyssey in civilization for decision-making agents / S. Qi et al. *The twelfth international conference on learning representations* : materials of the International Scientific Conference, Vienna, 7–11 May 2024. URL: <https://doi.org/10.48550/arXiv.2401.10568> (date of access: 17.05.2026).

11. Bates J. The role of emotion in believable agents. *Communications of the ACM*. 1994. Vol. 37, no. 7. P. 122–125. URL: <https://doi.org/10.1145/176789.176803> (date of access: 14.05.2026).

12. Generative agents: interactive simulacra of human behavior / J. S. Park et al. *UIST '23: the 36th annual ACM symposium on user interface software and technology*, San Francisco CA USA. New York, NY, USA, 2023. URL: <https://doi.org/10.1145/3586183.3606763> (date of access: 14.05.2026).

13. Спащенко Р.Т., Бібічков І.Є. Створення ШІ-агента для 4X стратегій на основі системи динамічних особистостей. *Радіоелектроніка та молодь у XXI столітті* : матеріали 30-го Міжнар. молодіж. форуму, 22–24 квітня 2026 р. Харків : ХНУРЕ, 2026. Т. 6. С. 116 – 118.

14. Endless Legend modding tutorial. revision 5. *Games2Gether*. URL: <https://medias.games2gether.com/forums/Pikou/tutorial-modding.pdf> (date of access: 22.06.2026).

15. Amplitude Studios. Endless Legend EULA. *Steam*. URL: http://store.steampowered.com/eula/289130_eula_0 (date of access: 22.06.2026).

16. Про авторське право і суміжні права : Закон України від 01.12.2022 № 2811-IX : станом на 12 груд. 2025 р. URL: <https://zakon.rada.gov.ua/laws/show/2811-20#Text> (дата звернення: 22.06.2026).

17. Директива Європейського Парламенту і Ради 2009/24/ЄС від 23 квітня 2009 року про правову охорону комп'ютерних програм : Директива Європ. Парламенту і Ради від 23.04.2009 № 2009/24/ЄС.

URL: https://zakon.rada.gov.ua/laws/show/9a3_001-09#Text (дата звернення: 22.06.2026).

18. ECMA-335. *Common language infrastructure (CLI). Official edition. 2012.*

URL: <https://ecma-international.org/publications-and-standards/standards/ecma-335/> (date of access: 22.06.2026).

19. GitHub - dnSpyEx/dnSpy: Unofficial revival of the well known .NET debugger and assembly editor, dnSpy. *GitHub*.

URL: <https://github.com/dnSpyEx/dnSpy> (date of access: 22.06.2026).

20. Use the MSBuild XML schema to control builds - MSBuild. *Microsoft Learn: Build with answers in reach*. URL: <https://learn.microsoft.com/en-us/visualstudio/msbuild/msbuild?view=visualstudio> (date of access: 22.06.2026).

21. Pugh J. K., Soros L. B., Stanley K. O. Quality diversity: a new frontier for evolutionary computation. *Frontiers in robotics and AI*. 2016. Vol. 3. URL: <https://doi.org/10.3389/frobt.2016.00040> (date of access: 22.06.2026).

22. Lehman J., Stanley K. O. Abandoning objectives: evolution through the search for novelty alone. *Evolutionary computation*. 2011. Vol. 19, no. 2. P. 189–223. URL: https://doi.org/10.1162/evco_a_00025 (date of access: 22.06.2026).

23. Mouret J.-B., Clune J. Illuminating search spaces by mapping elites. *arXiv*. 2015. URL: <https://doi.org/10.48550/arXiv.1504.04909> (date of access: 22.06.2026).