

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Алгоритмічне формування особистісних рис ігрових персонажів для динамічного прийняття рішень в ігровому середовищі»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Олександр ЯРОШЕВСЬКИЙ
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-62
_____ Олександр ЯРОШЕВСЬКИЙ

Керівник: _____ Ірина ЗАМРІЙ
д.т.н., доцент

Рецензент: _____

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« _____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Ярошевському Олександр Вікторовичу

1. Тема кваліфікаційної роботи: «Алгоритмічне формування особистісних рис ігрових персонажів для динамічного прийняття рішень в ігровому середовищі»

керівник кваліфікаційної роботи Ірина Замрій, д.т.н., доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024 р. № 320.

2. Строк подання кваліфікаційної роботи «17» грудня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, моделі особистісних рис, методи побудови прийняття рішень.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження моделей особистісних рис та їх адаптація для ігрових персонажів..

2. Аналіз алгоритмів прийняття рішень у динамічних середовищах.

3. Розробка алгоритму прийняття рішень на основі особистісних рис.

5. Перелік ілюстративного матеріалу: *презентація*
1. Актуальність роботи.
 2. Математична модель прийняття рішень з врахуванням особистісних рис.
 3. Алгоритм прийняття рішення на основі особистісних рис
 4. Модифікований метод прийняття рішення з врахуванням особистісних рис.
 5. Діаграма класів.
 6. Результати моделювання прийняття рішення з врахуванням особистісних рис.
6. Дата видачі завдання «16» жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	16.10-23.10.2024	
2	Вивчення матеріалів для аналізу формування особистісних рис ігрових персонажів	24.10-30.10.2024	
3	Дослідження алгоритмів прийняття рішень	31.10-04.11.2024	
4	Аналіз впливу особистісних рис на алгоритми прийняття рішень	05.11-09.11.2024	
5	Розробка алгоритму прийняття рішень на основі особистісних рис	10.11-16.11.2024	
6	Імплементація алгоритму в тестовому ігровому середовищі	17.11-22.11.2024	
7	Оформлення роботи: вступ, висновки, реферат	23.11-25.11.2024	
8	Розробка демонстраційних матеріалів	26.11-27.11.2024	
9	Попередній захист роботи	27.11.2024	

Здобувач вищої освіти _____
(підпис)

Олександр ЯРОШЕВСЬКИЙ

Керівник
кваліфікаційної роботи _____
(підпис)

Ірина ЗАМРІЙ

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 81 стор., 6 табл., 18 рис., 30 джерел.

Мета роботи – покращення ефективності прийняття рішень на основі особистісних рис ігрових персонажів в адаптивному середовищі.

Об'єкт дослідження – процес прийняття адаптивних рішень персонажами в ігровому середовищі.

Предмет дослідження – алгоритми та методи формування особистісних рис ігрових персонажів, які впливають на їхню здатність приймати адаптивні рішення в умовах адаптивного ігрового середовища.

У роботі використано різноманітні методи, зокрема аналіз наукової літератури для вивчення сучасних підходів до моделювання поведінки персонажів, математичне моделювання для побудови формальних моделей особистісних рис, а також алгоритмічний підхід для створення механізмів прийняття рішень. Реалізація алгоритмів виконувалася з використанням технологій об'єктно-орієнтованого програмування (ООП), що забезпечило модульність, масштабованість і повторне використання коду.

Проведено детальний аналіз сучасних методів моделювання особистісних рис персонажів, таких як Big Five, HEXACO і Dark Triad. Розглянуто їхню інтеграцію в алгоритми прийняття рішень у комп'ютерних іграх. Вивчено вплив цих методів на адаптивність персонажів у динамічних середовищах, що дало змогу виявити переваги та обмеження існуючих підходів.

Розроблено та оптимізовано математичну модель прийняття рішень, яка враховує особистісні риси персонажів, їхню взаємодію з навколишнім середовищем і адаптацію до змін. На основі цієї моделі створено алгоритм, який забезпечує гнучкість і адаптивність поведінки персонажів відповідно до їхніх індивідуальних характеристик.

Проведено серію експериментів у тестовому ігровому середовищі для перевірки ефективності розробленого алгоритму. Результати тестування продемонстрували реалістичну поведінку персонажів, їхню здатність до адаптації в різних ігрових ситуаціях і переваги запропонованого підходу порівняно з існуючими алгоритмами.

КЛЮЧОВІ СЛОВА: МОДЕЛІ ОСОБИСТОСТІ, ОСОБИСТІСНІ РИСИ, ПРИЙНЯТТЯ РІШЕННЯ, АДАПТИВНЕ СЕРЕДОВИЩЕ.

ABSTRACT

Text part of the master's qualification work: 81 pages, 18 pictures, 6 table, 30 sources.

The purpose of the work – to improve the efficiency of decision-making based on the personal traits of game characters in an adaptive environment.

Object of research – the process of making adaptive decisions by characters in the game environment.

Subject of research – algorithms and methods of forming personal traits of game characters that affect their ability to make adaptive decisions in an adaptive gaming environment.

The study used a variety of methods, including analysis of scientific literature to study modern approaches to modeling character behavior, mathematical modeling to build formal models of personality traits, and an algorithmic approach to create decision-making mechanisms. The algorithms were implemented using object-oriented programming (OOP) technologies, which ensured modularity, scalability, and code reuse.

A detailed analysis of modern methods of modeling character traits, such as Big Five, HEXACO and Dark Triad, is carried out. Their integration into decision-making algorithms in computer games is considered. The influence of these methods on the adaptability of characters in dynamic environments is studied, which made it possible to identify the advantages and limitations of existing approaches.

A mathematical model of decision-making was developed and optimized, which takes into account the personal traits of characters, their interaction with the environment and adaptation to changes. On the basis of this model, an algorithm was created that provides flexibility and adaptability of character behavior in accordance with their individual characteristics.

A series of experiments were conducted in a test game environment to verify the effectiveness of the developed algorithm. The test results demonstrate the realistic behavior of characters, their ability to adapt in different game situations, and the advantages of the proposed approach compared to existing algorithms.

KEYWORDS: PERSONALITY MODELS, PERSONALITY TRAITS, DECISION-MAKING, ADAPTIVE ENVIRONMENT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	11
ВСТУП.....	12
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	15
1.1 Особливості поведінки ігрових персонажів в ігровому середовищі	15
1.2 Підходи до реалізації особистісних рис у персонажах	18
1.3 Особливості побудови поведінкових моделей у комп'ютерних іграх	21
2 АНАЛІЗ МЕТОДІВ ФОРМУВАННЯ ОСОБИСТІСНИХ РИС ДЛЯ ДИНАМІЧНОГО ПРИЙНЯТТЯ РІШЕНЬ.....	22
2.1 Огляд моделей особистості	22
2.1.1 Big Five	22
2.1.2 HEXACO	24
2.1.3 Dark Triad	26
2.1.4 Комбінована модель особистості	29
2.2 Математичні методи прийняття рішень	32
2.2.1 Метод зваженої суми	32
2.2.2 Метод корисності очікування	33
2.2.3 Наївний підхід Баєса	34
2.2.4 Метод сукупності голосів.....	34
2.3 Математична модель прийняття рішень з врахуванням особистісних рис	35
3 РОЗРОБКА АЛГОРИТМУ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ ОСОБИСТІСНИХ РИС.....	39
3.1 Алгоритм прийняття рішень на основі особистісних рис	39
3.1.1 Ініціалізація необхідних параметрів	39
3.1.2 Розрахунок показника рішення для кожного варіанту	42
3.1.3 Нормалізація результатів.....	44
3.1.4 Вибір оптимального варіанту	46
3.1.5 Поєднання з іншими методами.....	48
3.2 Реалізація алгоритму в тестовому ігровому середовищі	50
3.2.1 Вибір технологій	50
3.2.2 Модулі програми.....	52
3.3 Аналіз результатів роботи алгоритму	67
ВИСНОВКИ.....	69
ПЕРЕЛІК ПОСИЛАНЬ	71
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	74
ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ.....	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

RPG - Рольова гра (Role-Playing Game).

WMS - Метод зваженої суми (Weighted Sum Method).

MEU - Метод корисності очікування (Maximum Expected Utility).

NB - Наївний підхід Баєса підхід (Naive Bayes).

OS - Операційна система (Operating System).

WebGL – Веб-графічна бібліотека (Web Graphics Library).

API - Інтерфейс програмування додатків (Application Programming Interface).

UI - Користувацький інтерфейс (User Interface).

ВСТУП

Сучасні комп'ютерні ігри є складними інтерактивними системами, які постійно еволюціонують, прагнучи забезпечити гравцям більш реалістичний та захоплюючий досвід. Одним із ключових напрямів цієї еволюції є вдосконалення поведінки ігрових персонажів. Гравці вже не задовольняються простими шаблонними реакціями персонажів, які часто базуються на статичних сценаріях або примітивних алгоритмах. Натомість вони очікують, що персонажі будуть демонструвати унікальну поведінку, яка відповідає певним особистісним характеристикам, що робить їх більш схожими на реальних людей.

Особистісні риси ігрових персонажів дозволяють зробити ігровий процес більш захоплюючим і динамічним. Така механіка особливо важлива в іграх із нелінійним сюжетом, де дії персонажів можуть впливати на розвиток подій, або в симуляціях соціальних взаємодій, де гравцям пропонується взаємодіяти з персонажами на рівні, близькому до людського спілкування. Додавання індивідуальності до ігрових агентів сприяє більш глибокому зануренню в гру, створює унікальні сценарії, які збагачують досвід гравців.

Запровадження алгоритмів, які поєднують моделювання особистісних рис із динамічним прийняттям рішень, відкривають широкі можливості для вдосконалення ігор. Наприклад, персонажі з унікальними особистостями можуть демонструвати різні реакції на однакові події, що збагачує ігровий процес і робить його більш захопливим для гравців. Крім того, такі системи можуть бути використані не тільки у розважальних іграх, а й у навчальних симуляторах, тренажерах або інструментах для соціальних експериментів, де необхідно моделювати поведінку складних агентів.

Наукова новизна роботи полягає у комбінуванні трьох психологічних моделей особистості: Big Five, HEXACO та Dark Triad для прийняття рішень ігровими персонажами. Таке поєднання дозволяє враховувати як позитивні, так і негативні риси характеру, що робить поведінку персонажів більш реалістичною та

подібною до людської. Об'єднання декількох результатів прийняття рішення є універсальним і дозволяє адаптувати розроблений метод до різноманітних сценаріїв, інтегруючи його з іншими моделями прийняття рішень, що може бути корисним у складних системах чи багатоаспектному аналізі.

Розроблений алгоритм забезпечує динамічну адаптацію особистісних рис персонажів залежно від контексту подій і прийнятих рішень. Це дозволяє персонажам демонструвати складну поведінку: від альтруїстичної до егоїстичної, а також змінювати свої характеристики з часом.

Таким чином, тема роботи, присвячена створенню алгоритмів формування особистісних рис ігрових персонажів для динамічного прийняття рішень, є актуальною. Вона відповідає сучасним тенденціям розвитку індустрії комп'ютерних ігор і сприяє розв'язанню важливих проблем у моделюванні поведінки інтелектуальних агентів.

Мета роботи - покращення ефективності прийняття рішень на основі особистісних рис ігрових персонажів в адаптивному середовищі.

Об'єкт дослідження - процес прийняття адаптивних рішень персонажами в ігровому середовищі.

Предмет дослідження - алгоритми та методи формування особистісних рис ігрових персонажів, які впливають на їхню здатність приймати адаптивні рішення в умовах адаптивного ігрового середовища.

Для досягнення мети були вирішені наступні завдання:

1. Вивчено сучасні підходи до створення ігрових персонажів, динамічних ігрових середовищ та побудови поведінкових моделей, що дозволило визначити основні тенденції та виклики у цій галузі.
2. Проаналізовано існуючі моделі особистості та методи формування особистісних рис, а також їхню інтеграцію в алгоритми прийняття рішень і взаємодії з ігровим середовищем.

3. Побудовано математичну модель прийняття рішень, яка враховує особистісні риси персонажів, що забезпечує адаптацію їхньої поведінки залежно від змін у середовищі.
4. Розроблено алгоритм прийняття рішень на основі особистісних рис, який поєднує психологічні характеристики персонажів із динамічним моделюванням їхніх дій.
5. Реалізовано та протестовано алгоритм у тестовому ігровому середовищі, проаналізовано ефективність його роботи, а також поведінку персонажів у різних сценаріях.
6. Проведено порівняння результатів роботи розробленого алгоритму з існуючими рішеннями, оцінено його переваги та недоліки, а також визначено можливості для подальшого вдосконалення.

Вирішення цих задач сприяє створенню більш реалістичних ігрових персонажів, здатних демонструвати унікальну поведінку, що відповідає їхнім особистісним характеристикам та адаптується до змін у ігровому середовищі. Це дозволяє підвищити рівень інтерактивності ігор, зробити їхній сюжет менш передбачуваним, а соціальну взаємодію між персонажами більш природною.

Результати дослідження апробовано та опубліковано у наступних тезах доповіді на конференціях:

1. Ярошевський О.В., Замрій І.В. Теоретичні основи формування особистісних рис ігрових персонажів. // Всеукраїнська науково-технічна конференція "Виклики та рішення в програмній інженерії" – Київ: ДУІКТ, 2024. – С. 195-198.
2. Ярошевський О.В., Замрій І.В. Динамічне прийняття рішень на основі особистісних рис ігрових персонажів. // Всеукраїнська науково-технічна конференція "Виклики та рішення в програмній інженерії" – Київ: ДУІКТ, 2024. – С. 176-179.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Особливості поведінки ігрових персонажів в ігровому середовищі

Ігрові персонажі є важливим елементом інтерактивного середовища, що формують ігровий досвід користувачів. Їхня поведінка значною мірою визначає складність, реалістичність і занурення у гру. У більшості сучасних ігор дії персонажів ґрунтуються на наперед визначених сценаріях, що дозволяє забезпечити певний рівень передбачуваності та логічності їхніх вчинків. Проте така модель має свої обмеження, особливо коли мова йде про створення динамічного, реалістичного ігрового середовища [1].

У більшості ігрових проєктів поведінка персонажів запрограмована у вигляді чітких сценаріїв. Сценарійна модель передбачає виконання персонажем визначеного алгоритму дій у відповідь на певні події чи умови в грі. Наприклад, вороги можуть атакувати гравця при його наближенні або втікати, якщо їх рівень здоров'я падає нижче певного порогу. Такі підходи забезпечують простоту реалізації та передбачуваність геймплею.

Однак, сценарійна модель має серйозні обмеження, оскільки не дозволяє персонажам адаптуватися до непередбачених ситуацій або демонструвати індивідуальність. Прийняття рішень персонажами ґрунтується лише на корисній оцінці ситуації, де враховуються лише фактори, що впливають на досягнення визначеної мети [2]. Особистісні риси або мотивації, що могли б зробити поведінку персонажів більш реалістичною, зазвичай ігноруються.

Одна з головних проблем сучасних ігор — це обмежена варіативність поведінки персонажів. Сценарна модель представлена на рисунку 1.1. Ігрові персонажі часто виглядають штучними, оскільки вони діють передбачувано і не демонструють складних або нестандартних моделей поведінки. Такі обмеження стають очевидними в іграх з відкритим світом, де користувачі очікують більшої інтерактивності й автономності дій персонажів.

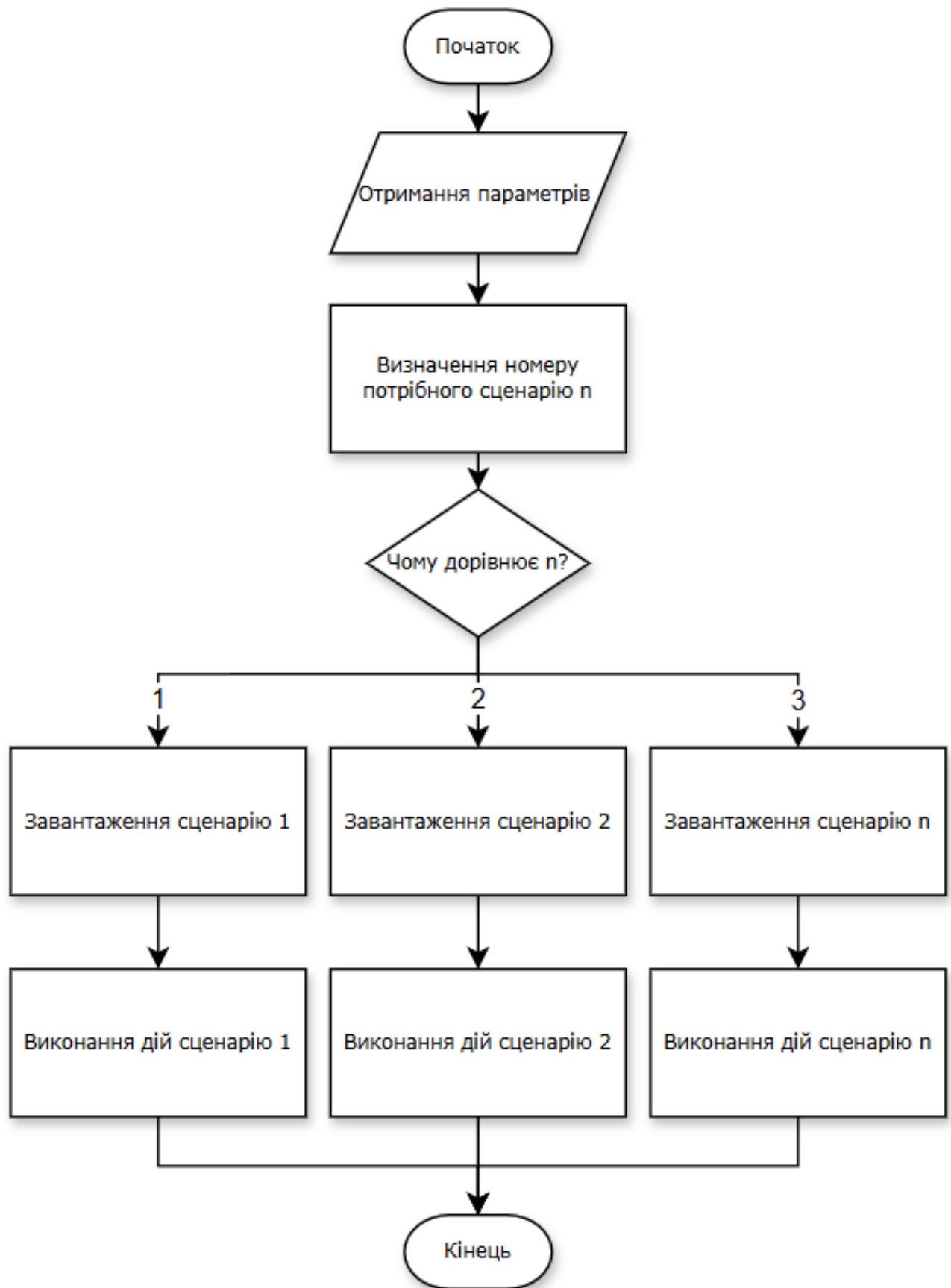


Рис 1.1 Узагальнена структура варіативності дій персонажа

Відсутність особистісних рис у поведінці персонажів також негативно впливає на сприйняття гри. Реальні люди зазвичай приймають рішення, ґрунтуючись на комбінації логіки, емоцій і індивідуальних особливостей. Ігнорування цих аспектів створює дисонанс між очікуваннями гравців і реальністю, що може знизити рівень занурення в ігрове середовище.

У роботі Баккеса та Спронка [3] розглянуто різноманітні методи моделювання поведінки в контексті відеоігор. Автори підкреслюють важливість адаптивності персонажів, яка базується на аналізі дій гравців. Зокрема, зазначено, що інтерактивна поведінка персонажів може значно підвищити реалістичність і занурення в гру.

У дослідженні представлено методи автоматичного аналізу та прогнозування дій гравців для подальшого використання в алгоритмах поведінки персонажів. Це дозволяє створювати не лише передбачувані сценарії, але й адаптивні моделі, які враховують поточний контекст гри та динаміку подій. Однак, як зазначають автори, реалізація таких систем вимагає значних обчислювальних ресурсів та глибокого розуміння мотиваційних чинників, що лежать в основі дій гравців.

Таким чином, дослідження підтверджує важливість інтеграції індивідуальних та соціальних аспектів у моделювання поведінки персонажів. Це відкриває нові можливості для розробки складніших ігор із більш реалістичним ігровим середовищем.

Сучасні ігрові персонажі здебільшого діють відповідно до сценаріїв, що обмежує їхню поведінкову варіативність та автентичність. Відсутність особистісних рис у моделях прийняття рішень створює обмеження для розробників, які прагнуть створити динамічні та реалістичні ігрові світи. Для вирішення цієї проблеми необхідно впроваджувати нові підходи до розробки моделей поведінки, які враховували б індивідуальні особливості персонажів та їхню здатність до адаптації.

1.2 Підходи до реалізації особистісних рис у персонажах

Ігрові персонажі є основою багатьох відеоігор, особливо в жанрах RPG (рольові ігри), стратегіях і симуляторах. Реалістичність і унікальність персонажів досягається завдяки набору параметрів, які визначають їхні характеристики, здібності та поведінку. На рисунку 1.2 представлені основні характеристики персонажів які застосовуються в іграх. Зазвичай серед таких параметрів виділяють:

- фізичні характеристики (зріст, вага, зовнішність);
- бойові навички (сила, швидкість, магічні здібності, рівень здоров'я);
- рівень досвіду (зростання характеристик із часом або виконанням завдань);
- соціальні атрибути (відносини з іншими персонажами, рівень довіри чи ворожості);
- особистісні риси (емоції, темперамент, схильність до ризику чи обережності).

Однак у більшості сучасних ігор особистісні риси персонажів заздалегідь визначаються розробниками та фіксуються в статичному вигляді. Це означає, що на етапі проектування персонажу надаються певні характеристики, такі як «відважний», «обережний» чи «егоїстичний», які впливають на його поведінку в конкретних ситуаціях, наприклад, у бойовій сцені або при взаємодії з іншими персонажами. Ці риси використовуються для створення шаблонів поведінки, що дозволяє розробникам забезпечити мінімально необхідний рівень індивідуалізації персонажів. Наприклад, персонаж із властивістю «відважний» завжди атакуватиме першими або ризикуватиме, тоді як «обережний» обере оборонну тактику чи уникнення конфлікту. Однак такий підхід є досить обмеженим і не враховує можливість еволюції рис у відповідь на зміни у середовищі чи накопичений досвід персонажа під час гри.

У більшості випадків ці характеристики залишаються незмінними протягом усього ігрового процесу, що значно знижує глибину та реалістичність поведінки персонажів. Навіть якщо ігровий світ динамічно змінюється, а персонаж стикається з новими викликами, його реакція залишається передбачуваною, оскільки задані

риси не піддаються адаптації. У результаті такі персонажі часто виглядають плоскими та позбавленими індивідуальності, що не відповідає сучасним вимогам до ігор із високим рівнем занурення та інтерактивності. Крім того, у деяких ігрових проєктах особистісні характеристики персонажів взагалі ігноруються, що призводить до одноманітної поведінки персонажів незалежно від контексту ситуації. Відсутність динамічного моделювання особистісних рис обмежує можливості для створення глибоких наративів і стратегічного геймплею, що потребує реалістичної адаптації персонажів до умов ігрового світу.

Ability Points

Proficiency Bonus: +2

Ability Points

0

Assign Bonus

+2

+1

Strength

-

10

+

☐

☐

★ Dexterity

-

17

+

☒

☐

Constitution

-

14

+

☐

☐

Intelligence

-

8

+

☐

☐

Wisdom

-

16

+

☐

☒

Charisma

-

8

+

☐

☐

Level 1 Monk

★

STR

DEX

CON

INT

WIS

CHA

10

17

14

8

16

8

+3

10

Sex

Class

Gift

Physique

Level

4

Vitality

12

Attunement

8

Endurance

14

Strength

14

Dexterity

9

Resistance

11

Intelligence

8

Faith

10

Humanity

0

STR

DEX

CON

INT

WIS

CHA

8

15

13

12

10

17

Melee Damage

3~10

Attack Bonus

+4

Ranged Damage

3~10

Attack Bonus

+4

Hit Points

9/9

Armour Class

13

Movement Speed

9m

Initiative

+2

Level

7 ⇒ 8

Souls

7484 ⇒ 8709

Required souls

793

Attributes

Vigor

14 ⇒ 15

Attunement

6 ⇒ 6

Endurance

12 ⇒ 12

Vitality

11 ⇒ 11

Strength

16 ⇒ 16

Dexterity

9 ⇒ 9

Intelligence

8 ⇒ 8

Faith

9 ⇒ 9

Luck

11 ⇒ 11

Base power

HP

515 ⇒ 530

FP

72 ⇒ 72

Stamina

97 ⇒ 97

Equip Load

51.0 ⇒ 51.0

Poise

17.27 ⇒ 17.27

Item Discovery

111 ⇒ 111

Defense

Physical

87 ⇒ 87

VS strike

87 ⇒ 87

VS slash

87 ⇒ 87

VS thrust

87 ⇒ 87

Magic

83 ⇒ 83

Fire

88 ⇒ 89

Lightning

86 ⇒ 86

Dark

84 ⇒ 84

Attack power

R Weapon 1

140 ⇒ 140

R Weapon 2

51 ⇒ 51

R Weapon 3

51 ⇒ 51

L Weapon 1

84 ⇒ 84

L Weapon 2

51 ⇒ 51

L Weapon 3

51 ⇒ 51

Resistances

Bleed

232 ⇒ 232

Poison

213 ⇒ 213

Frost

265 ⇒ 265

Curse

197 ⇒ 197

Attunement Slots

0 ⇒ 0

Рис 1.2 Приклад характеристик персонажів

У сучасній практиці виділяють два основні підходи до реалізації особистісних рис у персонажах:

Статичний підхід. Цей підхід передбачає, що всі особистісні риси персонажа визначаються на етапі розробки гри та не змінюються протягом ігрового процесу. Наприклад, у грі NPC із рисою «жадібний» завжди намагатиметься вимагати від гравця більше ресурсів, незалежно від змін у сюжеті чи відносинах із гравцем. Такий підхід простий у реалізації, але обмежує варіативність і занурення в ігровий процес.

Динамічний підхід. Динамічний підхід передбачає можливість адаптації особистісних рис персонажів залежно від їхнього досвіду чи взаємодії з іншими персонажами та гравцем. Наприклад, персонаж, який раніше був ворожим до гравця, може змінити свою поведінку на основі позитивних дій або спільного проходження випробувань. На жаль, такі системи зустрічаються значно рідше, оскільки вимагають складнішої реалізації та обчислювальних ресурсів.

Статичність або повна відсутність особистісних рис обмежує можливості створення динамічного ігрового середовища. Гравці часто стикаються з тим, що персонажі діють передбачувано і не демонструють змін, які могли б відобразити розвиток їхньої особистості. Це знижує емоційну залученість гравця та позбавляє гру глибини.

Розробка персонажів із адаптивними особистісними характеристиками відкриває нові перспективи для створення ігор із глибшим ігровим процесом та реалістичнішими сценаріями. Зміна рис персонажів у реальному часі дозволяє їм адаптувати свою поведінку до конкретних ситуацій, рішень гравця та отриманого досвіду, що робить взаємодію менш передбачуваною та більш захопливою. Наприклад, персонаж, який спочатку був обережним, може стати відважним завдяки пережитим успіхам, або ж навпаки, втратити довіру через негативний досвід. Такі підходи створюють живий ігровий світ, де персонажі розвиваються динамічно, реагуючи на обставини, а це підвищує загальну реалістичність ігрового середовища та рівень занурення гравця.

1.3 Особливості побудови поведінкових моделей у комп'ютерних іграх

Поведінкові моделі в комп'ютерних іграх є ключовим елементом створення динамічного ігрового середовища. Їхня мета — забезпечити персонажам здатність взаємодіяти з оточенням, гравцями та іншими персонажами у спосіб, що виглядає природно та відповідає контексту гри. Розробка таких моделей вимагає врахування низки факторів, серед яких: тип гри, жанрові особливості, рівень реалізму та очікування гравців.

Основні підходи до побудови поведінкових моделей включають сценарний, подійно-орієнтований підходи та використання поведінкових дерев.

Сценарний підхід один із найпоширеніших підходів до моделювання поведінки в іграх — використання сценаріїв. Поведінка персонажів задається у вигляді заздалегідь визначених послідовностей дій. Наприклад, NPC може реагувати на гравця, атакуючи його, тікаючи або пропонуючи завдання залежно від поточної ситуації. Цей підхід простий у реалізації та ефективний для ігор із лінійним сюжетом, проте обмежує варіативність і гнучкість поведінки [4].

Подійно-орієнтована поведінка персонажа визначається набором правил, які активуються в певних умовах. Наприклад, у стратегіях реального часу юніти можуть автоматично атакувати ворогів, якщо ті знаходяться в зоні видимості. Цей підхід дозволяє підвищити адаптивність, але все ще не враховує індивідуальних особливостей персонажів.

Поведінкові дерева є ефективним інструментом для створення складних моделей. Вони дозволяють ієрархічно структурувати логіку дій персонажів, забезпечуючи послідовність та умовність їхньої поведінки [5]. Наприклад, ворог у грі може спочатку спробувати знайти укриття, якщо рівень його здоров'я низький, а якщо укриття немає — атакувати гравця.

2 АНАЛІЗ МЕТОДІВ ФОРМУВАННЯ ОСОБИСТІСНИХ РИС ДЛЯ ДИНАМІЧНОГО ПРИЙНЯТТЯ РІШЕНЬ

2.1 Огляд моделей особистості

2.1.1 Big Five

Однією з найбільш поширених і науково обґрунтованих моделей особистості є модель Big Five або П'ятифакторна модель особистості. Вона описує структуру особистості через п'ять основних факторів, кожен з яких відображає певний аспект поведінки, емоцій та взаємодії індивіда з навколишнім світом [6-9, 12].

На рисунку 2.1 представлені фактори моделі Big Five.

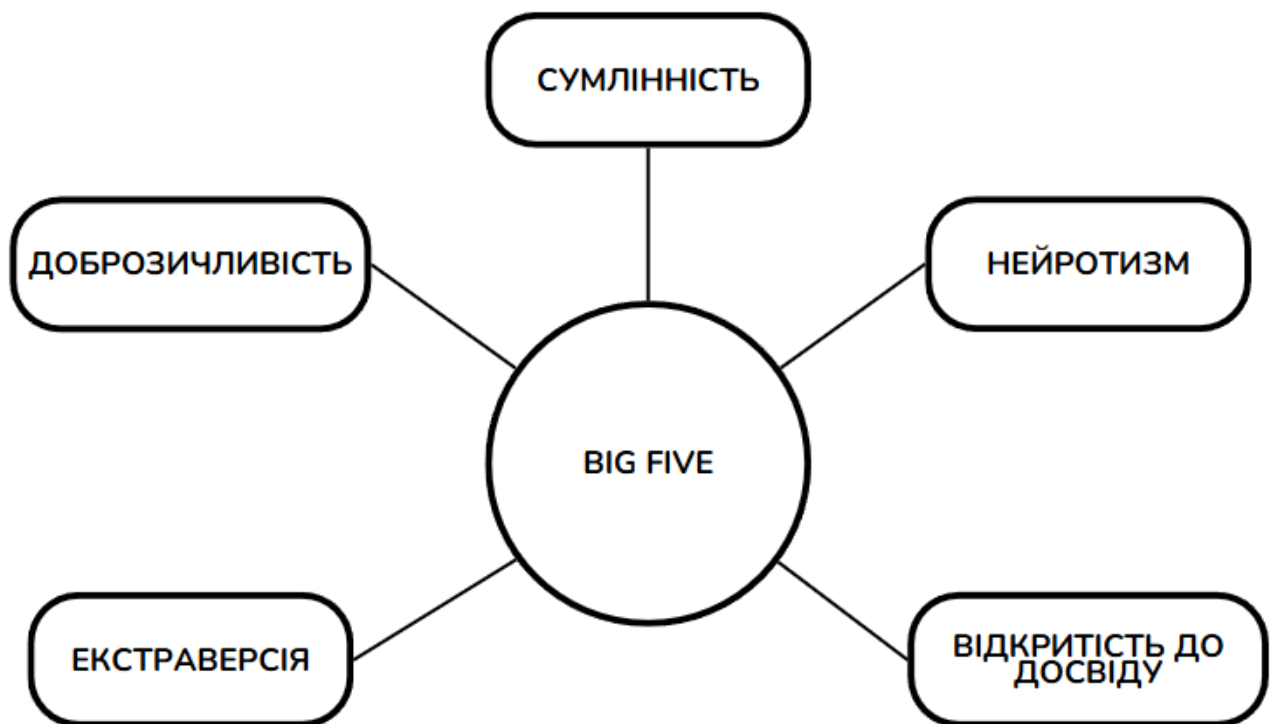


Рис 2.1 Фактори моделі Big Five

В таблиці 2.1 представлений опис факторів Big Five.

Таблиця 2.1

Опис факторів моделі Big Five

Фактор	Характеристика	Високий рівень	Низький рівень
Екстраверсія	Рівень активності, соціальності та емоційної енергії	Товариськість, енергійність, оптимізм	Замкнутість, обережність, схильність до самоти
Доброзичливість	Ставлення до інших людей, доброта, співчуття, довіра	Альтруїзм, співпереживання, прагнення до гармонії	Конкурентність, скептицизм, холодність
Сумлінність	Організованість, відповідальність, цілеспрямованість	Надійність, дисциплінованість, планування	Імпульсивність, недбалість, хаотичність
Нейротизм	Емоційна стабільність і схильність до негативних емоцій	Тривожність, дратівливість, емоційна нестабільність	Спокій, стійкість до стресу, емоційна врівноваженість
Відкритість до досвіду	Інтерес до інновацій, креативність, інтелектуальна цікавість	Творчість, допитливість, нестандартне мислення	Прагматизм, консерватизм, схильність до традиційності

Модель Big Five часто використовується для створення реалістичних та багатогранних ігрових персонажів. Вона дозволяє:

1. Сформувати унікальні особистості NPC: задавши кожному персонажу різний рівень кожного з п'яти факторів, розробники можуть створити унікальну поведінку, яка залежить від їхньої "особистості".

2. Вплинути на сюжетні рішення: наприклад, персонаж із високою екстраверсією буде шукати контакту з іншими, тоді як замкнутий обиратиме більш індивідуальні дії.
3. Забезпечити адаптивну поведінку: враховуючи зміни факторів особистості під час гри, можна створити персонажів, які "ростуть" і розвиваються разом із гравцем.

Модель Big Five є універсальним і ефективним інструментом для вивчення особистості. Її адаптація в ігровій індустрії дозволяє створити більш глибокі та реалістичні поведінкові моделі персонажів, що підвищує загальну якість ігрового досвіду.

2.1.2 HEXACO

Модель HEXACO, яка представлена на рисунку 2.2, є сучасною та популярною теорією, що розширює класичну п'ятифакторну модель Big Five, додаючи шостий фактор — чесність-скромність [9-11]. Ця модель використовується для детальнішого аналізу особистісних рис та поведінки, а також знаходить застосування у психології, соціології та геймдизайні.

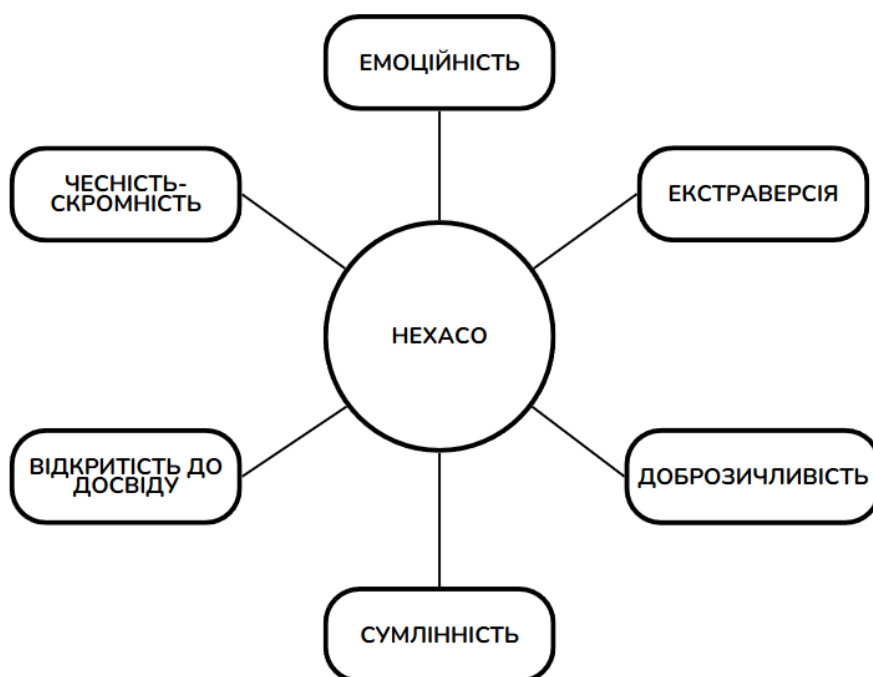


Рис 2.2 Фактори моделі HEXACO

Опис факторів моделі НЕХАСО представлений в таблиці 2.2.

Таблиця 2.2

Основні фактори моделі НЕХАСО

Фактор	Характеристика	Високий рівень	Низький рівень
Чесність-скромність	Етичність, чесність, уникнення егоїзму та маніпуляцій	Чесність, скромність, альтруїзм	Маніпулятивність, жадібність, прагнення до статусу
Емоційність	Схильність до страхів, занепокоєння, залежності від інших людей	Співпереживання, турбота про безпеку, емоційна залежність	Емоційна стійкість, самостійність, впевненість
Екстраверсія	Соціальність, активність, позитивний настрій	Товариськість, енергійність, лідерські якості	Замкнутість, стриманість, небажання соціальної активності
Доброзичливість	Готовність до компромісів, терпимість, дружелюбність	Терплячість, здатність до прощення, уникнення конфліктів	Роздратованість, суперечливість, схильність до помсти
Сумлінність	Організованість, відповідальність, прагнення до досягнення цілей	Дисципліна, цілеспрямованість, ефективність	Недбалість, імпульсивність, відсутність планування
Відкритість до досвіду	Інтерес до нового, творчості, креативності	Допитливість, нестандартне мислення, творчий підхід	Консерватизм, прагматизм, фокус на традиціях

На відміну від Big Five, модель HEXACO враховує етичні аспекти особистості через фактор чесності-скромності. Це робить її більш придатною для дослідження міжособистісної поведінки, зокрема таких рис, як альтруїзм, маніпулятивність і чесність.

Модель також забезпечує глибший аналіз емоційності, враховуючи аспекти, що пов'язані з почуттям захищеності та страхом.

Модель HEXACO відкриває широкі можливості для використання в ігровій індустрії:

1. Розробка унікальних персонажів: кожен фактор моделі може стати основою для створення багатогранного персонажа з індивідуальними рисами.
2. Динамічні системи моралі: фактор чесності-скромності дозволяє враховувати етичні рішення персонажів у грі.
3. Адаптивна поведінка NPC: моделювання взаємодії з іншими персонажами та гравцем на основі різних рівнів факторів HEXACO.

Модель HEXACO є потужним інструментом для аналізу особистісних рис, особливо в контексті етичної та соціальної поведінки. Її застосування в ігровій індустрії дозволяє створювати складніші моделі персонажів, що робить ігровий процес більш насиченим і реалістичним.

2.1.3 Dark Triad

Модель Dark Triad охоплює три ключові аспекти негативних рис особистості: нарцисизм, макіавеллізм та психопатію. Ця модель використовується для аналізу антисоціальної поведінки, міжособистісних маніпуляцій та схильності до порушення соціальних норм [13-15].

На рисунку 2.3 представлена модель Dark Triad.

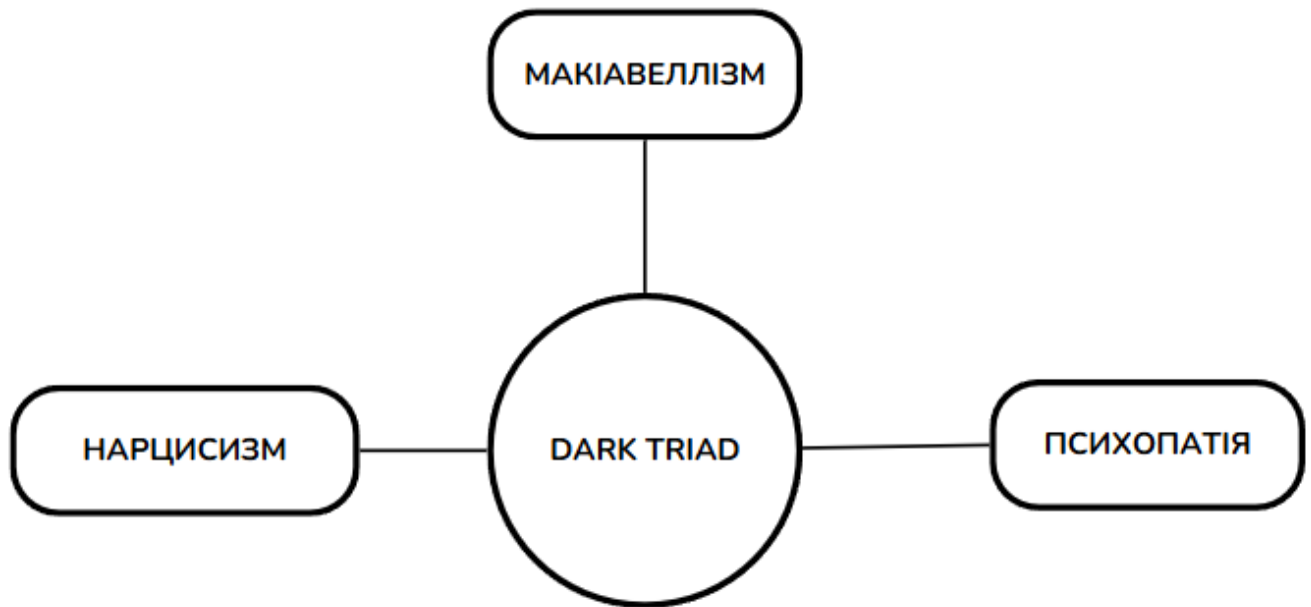


Рис 2.3 Фактори моделі Dark Triad

В таблиці 2.3 представлений опис факторів моделі Dark Triad.

Таблиця 2.3

Опис факторів моделі Dark Triad

Фактор	Характеристика	Високий рівень	Низький рівень
Нарцисизм	Егоцентризм, почуття власної значущості, потреба в увазі	Самозакоханість, грандіозне мислення, впевненість у власній перевазі	Скромність, здатність до самокритики, відсутність егоцентризму
Макиавеллізм	Схильність до маніпуляцій, цинізм, прагматичність	Холоднокровність, раціональність, використання інших у власних цілях	Прямолінійність, довірливість, орієнтація на моральні принципи

Продовження таблиці 2.3

Опис факторів моделі Dark Triad

Фактор	Характеристика	Високий рівень	Низький рівень
Психопатія	Емоційна холодність, імпульсивність, відсутність почуття провини	Байдужість до почуттів інших, ризикована поведінка, агресивність	Емпатія, емоційна стабільність, здатність враховувати наслідки

На відміну від традиційних моделей, які зосереджуються на позитивних аспектах особистості, Dark Triad акцентує увагу на темних сторонах людської поведінки. Ці риси не завжди проявляються в патологічній формі, а інколи використовуються як ефективний інструмент для досягнення цілей у конкурентному середовищі.

Застосування моделі Dark Triad в ігровій індустрії:

1. Створення антагоністів: риси Темної Тріади часто є основою для розробки персонажів-антагоністів, які привертають увагу гравців своїм складним характером.
2. Реалістичні соціальні взаємодії: впровадження елементів маніпуляцій, егоїзму та аморальності у взаємодії NPC із гравцем.
3. Рольові ігри та моральний вибір: персонажі з рисами Dark Triad можуть кидати виклик гравцю, змушуючи приймати складні моральні рішення.

Модель Dark Triad є цінним інструментом для аналізу негативних рис особистості. Її застосування в ігровій індустрії дозволяє створювати багатогранних персонажів і поглиблювати соціальну динаміку в ігровому середовищі.

2.1.4 Комбінована модель особистості

Особистісні моделі Big Five, HEXACO та Dark Triad пропонують різні підходи до аналізу людської поведінки. Big Five і HEXACO акцентують увагу на загальних рисах особистості, включаючи позитивні якості, тоді як Dark Triad фокусується на "темних" аспектах, які часто ігноруються у традиційних моделях.

В таблиці 2.4 наведено порівняння трьох моделей особистості.

Таблиця 2.4

Порівняння моделей Big Five, HEXACO та Dark Triad

Модель	Сильні сторони	Обмеження
Big Five	Проста, загальноприйнята модель; охоплює основні позитивні риси особистості	Не враховує етичні та антисоціальні аспекти поведінки
HEXACO	Додає фактор чесності-скромності, що робить модель придатною для аналізу моральної поведінки	Більш складна структура, яка може бути надмірною для базового аналізу
Dark Triad	Охоплює темні риси особистості, зокрема нарцисизм, макіавеллізм і психопатію	Зосереджується виключно на негативних аспектах, ігноруючи позитивні якості

Комбінована модель об'єднує ключові елементи Big Five, HEXACO та Dark Triad для створення багатовимірного підходу до аналізу особистості. Вона враховує як позитивні, так і негативні аспекти людської поведінки, що дозволяє краще зрозуміти складність міжособистісних взаємодій. На рисунку 2.4 представлена комбінована модель особистості.

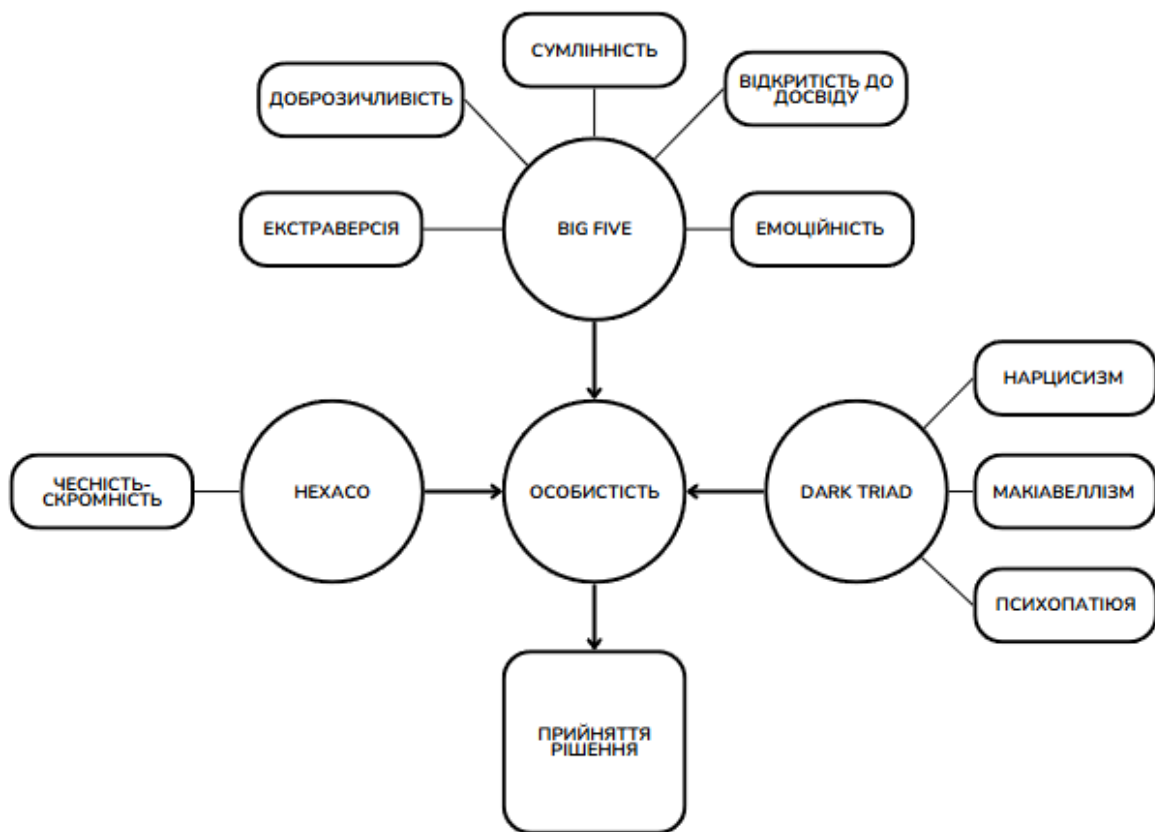


Рис 2.3 Комбінована модель особистості

В таблиці 2.5 представлений опис факторів комбінованої моделі.

Таблиця 2.5

Опис факторів комбінованої моделі

Фактор	Характеристика	Високий рівень	Низький рівень	Джерело
Чесність-скромність	Етичність, альтруїзм, уникнення маніпуляцій	Чесність, скромність, прагнення до справедливості	Егоїзм, маніпулятивність, жадібність	HEXACO
Емоційність	Чутливість до емоцій, страхів і турбота про безпеку	Співчуття, емоційна вразливість, обережність	Байдужість, емоційна холодність, відсутність страху	HEXACO, Big Five
Екстраверсія	Соціальність, енергійність, оптимізм	Комунікабельність, впевненість, активність	Замкнутість, відчуженість, низький рівень енергії	HEXACO, Big Five
Доброзичливість	Терпимість, уникнення конфліктів	Готовність до компромісів, емпатія	Ворожість, агресивність, егоцентризм	HEXACO, Big Five

Продовження таблиці 2.5

Опис факторів комбінованої моделі

Фактор	Характеристика	Високий рівень	Низький рівень	Джерело
Сумлінність	Організованість, дисципліна, прагнення досягати цілей	Висока організованість, відповідальність	Нехтування обов'язками, низька дисципліна	Big Five
Відкритість до досвіду	Творчість, допитливість, інтерес до нових ідей	Оригінальність, широкі інтереси, схильність до інновацій	Консерватизм, байдужість до нових знань	Big Five
Нарцисизм	Почуття власної значущості, егоцентризм	Самовпевненість, прагнення до захоплення	Скромність, орієнтація на інших	Dark Triad
Макіавеллізм	Схильність до маніпуляцій, прагматичність	Холоднокровність, раціональність, використання інших	Прямолінійність, довірливість, моральність	Dark Triad
Психопатія	Емоційна холодність, імпульсивність	Байдужість до почуттів інших, готовність до ризику	Емпатія, емоційна стабільність, обережність	Dark Triad

Комбінована модель є універсальним інструментом, що дозволяє:

1. Проводити комплексний аналіз особистості. Поєднання позитивних і негативних аспектів забезпечує більш точний і багатовимірний портрет індивіда.
2. Гнучко використовувати її для різних цілей, включаючи створення персонажів у іграх, психологічну діагностику та прогнозування поведінки.
3. Оцінювати персонажів з етичної сторони. Включення чесності-скромності дозволяє аналізувати моральну сторону особистості, що особливо важливо в соціальних і професійних контекстах.

4. Балансувати між позитивними та негативними рисами. Темні риси особистості враховуються не тільки як недоліки, але й як риси, що можуть бути корисними у певних ситуаціях (наприклад, у конкурентному середовищі).

Комбінована модель дозволяє створювати більш реалістичних і складних персонажів у іграх, де враховуються як позитивні, так і негативні риси, що робить персонажів морально неоднозначними. Завдяки моделюванню взаємодії персонажів з урахуванням емоційності, темних рис і соціальної поведінки, створюється динамічне середовище для гравців. Крім того, така модель сприяє різноманітності ігрових сценаріїв, адаптуючи поведінку NPC на основі різних факторів особистості, що дозволяє робити гру більш захоплюючою та непередбачуваною.

2.2 Математичні методи прийняття рішень

Динамічне прийняття рішень в ігровому середовищі потребує застосування математичних методів, які дозволяють моделювати процес вибору для ігрових персонажів на основі доступної інформації [16]. Розглядається кілька популярних підходів: метод зваженої суми, метод корисності очікування, наївний підхід Баєса і метод сукупності голосів.

2.2.1 Метод зваженої суми

Метод зваженої суми (WMS) є одним із найпростіших методів багатокритеріального прийняття рішень [17-18]. Його основна ідея полягає у визначенні вагових коефіцієнтів для кожного критерію, що характеризує вибір. Потім обчислюється загальний бал кожної альтернативи як зважена сума оцінок за всіма критеріями (2.1):

$$S_i = \sum_{i=1}^n w_i x_i, \quad (2.1)$$

де S_i - загальний бал i -го альтернативи;

w_i - вага i -го критерію;

x_i - оцінка i -го альтернативи;

n - кількість критеріїв.

Метод зваженої суми підходить для задач, де всі критерії мають однакову одиницю вимірювання, а також припускає, що критерії є незалежними. Для ігрових персонажів цей метод може бути використаний для вибору оптимальної дії, враховуючи важливість кожного критерію, наприклад, шкоду супротивникам, безпеку персонажа або витрати ресурсів.

2.2.2 Метод корисності очікування

Метод корисності очікування (MEU) ґрунтується на теорії очікуваної корисності, яка передбачає, що рішення приймаються на основі максимізації очікуваної вигоди [19-20]. Корисність кожної альтернативи обчислюється як зважена сума можливих результатів з урахуванням їх імовірностей:

$$EU(a) = \sum_{s \in S} P(s|a)U(s), \quad (2.2)$$

де $EU(a)$ - очікувана корисність дії a ;

$P(s|a)$ - імовірність стану s за умови виконання дії a ;

$U(s)$ - корисність стану s ;

S - множина можливих станів.

Цей метод особливо ефективний у випадках, коли існує невизначеність щодо результатів дій, наприклад, у випадкових подіях в ігрових середовищах. Для персонажів він дозволяє враховувати ризики та ймовірність досягнення бажаних цілей.

2.2.3 Наївний підхід Баєса

Наївний підхід Баєса (NB) [21-22] базується на застосуванні теореми Байєса для обчислення ймовірності прийняття рішення, виходячи з доступних даних. Для кожної альтернативи обчислюється апостеріорна ймовірність:

$$P(a|X) = \frac{P(X|a)P(a)}{P(X)}, \quad (2.3)$$

де $P(a|X)$ - ймовірність дії a , враховуючи спостереження X ;

$P(X|a)$ - ймовірність спостереження X , якщо виконано дію a ;

$P(a)$ - апіорна ймовірність дії a ;

$P(X)$ - ймовірність спостереження X .

Наївний підхід припускає незалежність ознак, що спрощує обчислення. У контексті ігрових персонажів цей підхід може бути використаний для оцінки ймовірності успіху різних дій на основі доступної інформації, наприклад, поведінки супротивників чи стану середовища.

2.2.4 Метод сукупності голосів

Метод сукупності голосів передбачає об'єднання рішень кількох моделей або агентів для вибору оптимальної альтернативи. Кожен агент або модель дає свій "голос" за одну з можливих дій, а остаточне рішення визначається на основі сукупності голосів [23-24]. Популярними варіантами є:

- проста більшість: вибирається альтернатива з найбільшою кількістю голосів.
- зважене голосування: кожен агент має вагу, що відображає його експертність або релевантність.
- правило консенсусу: вибір здійснюється, якщо певна альтернатива отримує понад заданий поріг голосів.

У динамічних ігрових системах цей метод дозволяє врахувати різноманітні підходи до прийняття рішень, що підвищує адаптивність ігрових персонажів.

2.3 Математична модель прийняття рішень з врахуванням особистісних рис

Для забезпечення більшої реалістичності персонажів необхідно враховувати їхні особистісні риси, що впливають на вибір дій у різних ситуаціях. У цьому розділі буде розглянуто математичну модель, яка поєднує елементи кількох відомих підходів до прийняття рішень, включаючи:

1. Метод зваженої суми - для врахування впливу кожної риси через вагові коефіцієнти.
2. Модель корисності очікування - для побудови рішення на основі важливості окремих характеристик, що нагадує концепцію корисності.
3. Метод сукупності голосів - для інтеграції кількох параметрів у єдине зважене рішення.

Поєднання цих методів дозволяє створити універсальну математичну модель, яка не лише враховує індивідуальні особливості персонажа, а й адаптується до змінних умов гри.

Розроблена математична модель прийняття рішень з врахуванням особистісних рис ігрових персонажів описується формулою:

$$D = \frac{\sum_{i=1}^n w_i \cdot T_i}{\sum_{i=1}^n w_i} \cdot 100, \quad (2.4)$$

де D - загальне значення рішення (наприклад, рівень переваги вибраної дії), виражене у відсотках;

w_i - вага i -ї особистісної риси або параметра, що характеризує важливість цього аспекту для прийняття рішення;

T_i - значення i -ї особистісної риси, яке впливає на рішення;

n - кількість врахованих особистісних рис.

Ця формула дозволяє враховувати вагу кожної особистісної риси, підсумовуючи їхній вплив і нормалізуючи результат. Це підходить для

моделювання прийняття рішень, у якому ключову роль відіграють індивідуальні характеристики персонажа.

В такому випадку значення функції D розраховується наступним чином:

$$D = \frac{w_H \cdot T_H + w_E \cdot T_E + w_X \cdot T_X + w_A \cdot T_A + w_C \cdot T_C + w_O \cdot T_O + w_{Narc} \cdot T_{Narc} + w_{Mach} \cdot T_{Mach} + w_{Psych} \cdot T_{Psych}}{w_H + w_E + w_X + w_A + w_C + w_O + w_{Narc} + w_{Mach} + w_{Psych}} \cdot 100, (2.5)$$

де T_H - чесність-скромність;

T_E - емоційність;

T_X - екстраверсія;

T_A - доброзичливість;

T_C - сумлінність;

T_O - відкритість до досвіду;

T_{Narc} - нарцисизм;

T_{Mach} - макіавеллізм;

T_{Psych} - психопатія;

w_H – вага чесності-скромності;

w_E – вага емоційності;

w_X – вага екстраверсії;

w_A – вага доброзичливості;

w_C – вага сумлінності;

w_O – вага відкритості до досвіду;

w_{Narc} – вага нарцисизму;

w_{Mach} – вага макіавеллізму.

w_{Psych} – вага психопатії.

Результат функції D визначає ймовірність вибору певного варіанту дії, з урахуванням впливу всіх заданих особистісних рис. Значення D виражається у відсотках ($[0,100]$), що робить його інтуїтивно зрозумілим для інтерпретації. Найвищий показник D серед усіх варіантів дій визначає фінальне рішення.

За необхідності поєднання результату, отриманого за допомогою розробленої формули для розрахунку впливу особистісних рис, із результатом іншого методу прийняття рішень, пропонується використати наступну формулу:

$$F = \frac{\alpha D + \beta R}{\alpha + \beta}, \quad (2.6)$$

де D - значення, розраховане за формулою 2.5.

R - результат іншого методу прийняття рішення, що може відображати оцінку альтернативи, імовірність успіху чи інший показник ефективності;

α та β - коефіцієнти ваги, що визначають відносну важливість D і R у кінцевому рішенні.

Формула F виконує функцію зваженого об'єднання двох результатів. Значення α і β обираються відповідно до специфіки задачі, деяких емпіричних даних або залежно від того, як сильно кожен із компонентів (D чи R) має впливати на кінцевий результат. Нормалізація через знаменник $\alpha + \beta$ забезпечує коректність розрахунків і гарантує, що F залишається в межах діапазону від 0 до 1 (за умови, що D і R також нормалізовані).

Такий підхід є універсальним і дозволяє адаптувати розроблений метод до різноманітних сценаріїв, інтегруючи його з іншими моделями прийняття рішень, що може бути корисним у складних системах чи багатоаспектному аналізі.

Модель має подібності до кількох відомих методів прийняття рішень, кожен з яких вносить свої особливості у її структуру. Нижче розглянуто основні методи, з якими пов'язана розроблена модель:

1. Метод зваженої суми. Модель найбільш схожа на метод зваженої суми, оскільки враховує вплив кожного фактора (особистісної риси) через вагові коефіцієнти. У WSM загальний результат обчислюється шляхом зваженого підсумовування критеріїв, що відповідає основній ідеї наведеної формули. Особливістю є масштабування результату до 100%, що додає зручності у використанні та інтерпретації.

2. Модель корисності. Врахування вагових коефіцієнтів можна трактувати як аналог оцінки ймовірності або важливості результату у моделі корисності очікування. Хоча в даній моделі немає прямого урахування невизначеності, ваги W_i можуть інтерпретуватися як суб'єктивна важливість кожної риси, що наближає її до концепції корисності.

3. Метод сукупності голосів. Формула також може бути розглянута як узагальнення системи голосування, де кожна риса "голосує" за прийняття рішення зі своєю вагою W_i . Остаточне значення D можна розглядати як зважений консенсус між всіма врахованими рисами.

Розроблена модель є адаптивною і враховує такі особливості:

1. Персоналізація: вагові коефіцієнти W_i можуть бути налаштовані відповідно до індивідуальних характеристик персонажа, що дозволяє моделювати різні стилі прийняття рішень.

2. Простота і масштабованість: завдяки простій структурі формули модель легко інтегрується в ігрові алгоритми і може бути розширена новими параметрами.

3. Гнучкість: застосування ваг дозволяє відображати як домінантні, так і другорядні риси.

У контексті ігрових персонажів модель може використовуватися для прийняття рішень, що залежать від індивідуальних характеристик персонажа. Наприклад, при виборі між атакою, захистом або втечею модель дозволяє врахувати такі риси, як агресивність, обережність чи схильність до ризику, визначаючи найбільш підходящу дію.

Таким чином, запропонована модель поєднує елементи методу зваженої суми, моделі корисності очікування та системи голосування, забезпечуючи універсальність і адаптивність у вирішенні задач динамічного прийняття рішень.

3 РОЗРОБКА АЛГОРИТМУ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ ОСОБИСТІСНИХ РИС

3.1 Алгоритм прийняття рішень на основі особистісних рис

3.1.1 Ініціалізація необхідних параметрів

Процес прийняття рішень на основі особистісних рис розпочинається з ретельної ініціалізації необхідних параметрів. Цей етап включає визначення множин характеристик персонажа, параметрів ситуації, а також можливих варіантів дій. Правильна ініціалізація гарантує коректну роботу алгоритму, забезпечує високу точність результатів і дозволяє адаптувати метод до різних контекстів.

Основою алгоритму є множина особистісних рис T , які характеризують індивідуальні психологічні особливості кожного персонажа. Для алгоритму використовується комбінована модель особистості, яка враховує як традиційні фактори, так і темну тріаду [25]. Набір характеристик включає:

- T_H — чесність-скромність (Honesty-Humility). Ця риса визначає схильність персонажа до чесності, уникнення обману, прагнення до скромності та соціальної відповідальності.
- T_X — емоційність (Emotionality), що визначає рівень чутливості персонажа до стресових ситуацій, його здатність до емпатії та переживань.
- T_E — екстраверсія (Extraversion), яка відображає активність персонажа, його соціальність, товариськість і оптимістичний настрій.
- T_A — доброзичливість (Agreeableness), що характеризує здатність персонажа до співпраці, альтруїзму, уникнення конфліктів і готовності допомагати іншим.
- T_C — сумлінність (Conscientiousness), яка визначає рівень організованості, відповідальності та спрямованості на досягнення поставлених цілей.

- T_O — відкритість до досвіду (Openness to Experience), що вказує на цікавість персонажа, його творчий потенціал і готовність вивчати нове.
- T_{Narc} — нарцисизм (Narcissism), який відображає ступінь самозакоханості персонажа, потребу у визнанні й домінуванні.
- T_{Mach} — макіавеллізм (Machiavellianism), що вказує на прагматизм, маніпулятивність і цинічність у досягненні своїх цілей.
- T_{Psych} — психопатія (Psychopathy), яка характеризується емоційною холодністю, імпульсивністю і відсутністю співчуття.

Кожна риса T_i нормалізована до діапазону $[0,1]$, що дозволяє використовувати їх у розрахунках без додаткової стандартизації.

Для врахування контексту ситуації та впливу кожної риси на прийняття рішень вводиться набір вагових коефіцієнтів W . Кожен коефіцієнт w_i відповідає значущості відповідної риси в конкретній ситуації. Наприклад:

- У ситуації соціальної взаємодії більшу вагу може мати w_E (екстраверсія) або w_A (доброзичливість).
- Для завдань, що вимагають відповідальності та точності, переважатимуть w_C (сумлінність) або w_H (чесність-скромність).
- У конфліктних чи конкурентних ситуаціях значущими можуть бути w_{Narc} (нарцисизм), w_{Mach} (макіавеллізм) або w_{Psych} (психопатія).

Усі вагові коефіцієнти нормалізовані так, що їх сума дорівнює 1:

$$\sum_{i=1}^n w_i = 1.$$

Це дозволяє зберігати пропорційність впливу характеристик у межах одного сценарію [26].

Наступним етапом є формування множини можливих варіантів дій:

$$A = \{A_1, A_2, \dots, A_k\}.$$

Ці варіанти визначають потенційні рішення, які персонаж може прийняти в заданій ситуації. Кожен варіант дії оцінюється окремо на основі заданих особистісних рис і вагових коефіцієнтів.

На основі ініціалізованих параметрів (T , W , A) алгоритм забезпечує адаптивне та контекстно залежне прийняття рішень. Коректна ініціалізація цих параметрів є ключовим кроком, який гарантує, що всі наступні розрахунки будуть точними, а результати — обґрунтованими.

На рисунку 3.1 наведена блок-схема ініціалізації необхідних параметрів.

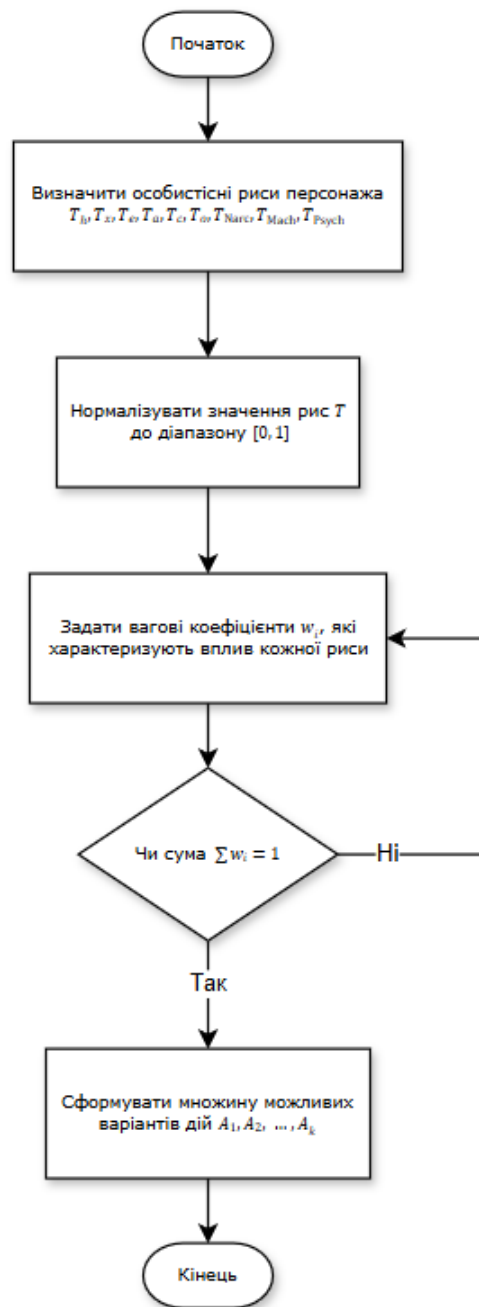


Рис 3.1 Блок-схема ініціалізації необхідних параметрів

3.1.2 Розрахунок показника рішення для кожного варіанту

Після ініціалізації необхідних параметрів наступним важливим етапом є розрахунок показника рішення D для кожного можливого варіанту дії A_k , де $k = 1, 2, \dots, t$, на основі особистісних рис і вагових коефіцієнтів. Цей розрахунок забезпечує формування обґрунтованого підходу до вибору оптимального варіанту дії, враховуючи як індивідуальні характеристики персонажа, так і значущість різних рис у конкретній ситуації.

На першому кроці обчислюється чисельник формули:

$$Numerator_k = \sum_{i=1}^n w_i T_i.$$

Це сума добутків значень особистісних рис T_i на їхні ваги w_i , яка відображає загальний вплив усіх рис на прийняття рішення для варіанту A_k .

На другому кроці обчислюється знаменник формули:

$$Denominator = \sum_{i=1}^n w_i.$$

Зазвичай знаменник дорівнює 1, оскільки ваги нормалізовані. Проте цей крок залишається важливим, щоб зберігати універсальність методу для випадків, коли нормалізація ваг не здійснена.

Завершальним кроком є ділення чисельника на знаменник та множення результату на 100:

$$D_k = \frac{Numerator_k}{Denominator} \cdot 100.$$

Отримане значення показує рівень відповідності варіанту дії A_k до особистісних рис персонажа в заданих умовах.

Роль вагових коефіцієнтів w_i полягає у врахуванні значущості кожної риси T_i у конкретній ситуації. Варіативність ваг дозволяє адаптувати алгоритм до широкого спектру сценаріїв і забезпечити його універсальність.

На основі описаних етапів можна представити алгоритм розрахунку показника D_k у вигляді блок-схеми, що представлена на рисунку 3.2.

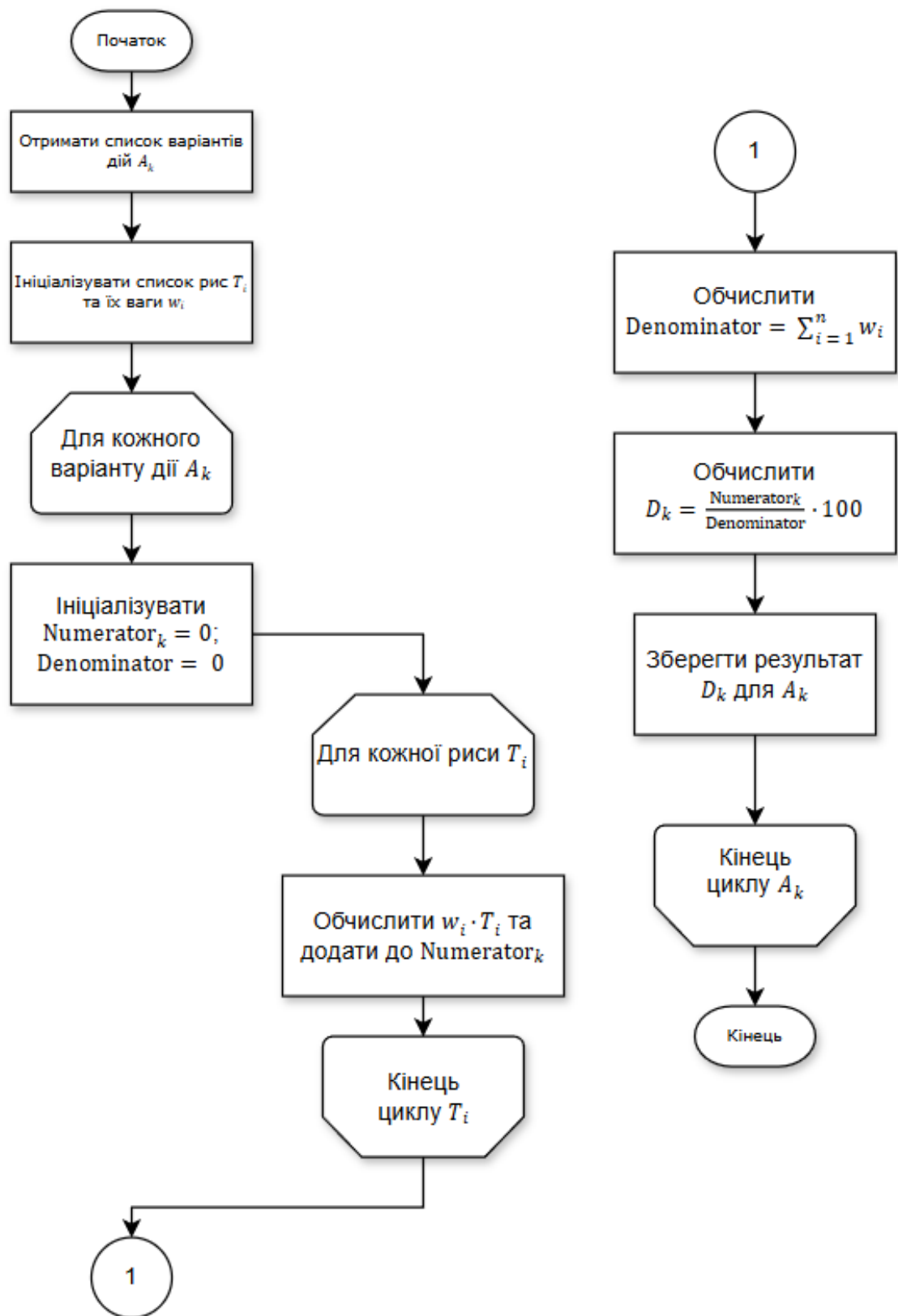


Рис 3.2 Алгоритм розрахунок показника рішення

Переваги алгоритму:

Гнучкість. Алгоритм легко адаптується до різних ситуацій за рахунок зміни вагових коефіцієнтів w_i .

1. Обґрунтованість. Розрахунок показника ґрунтується на науково визнаних моделях особистісних рис і враховує їхню значущість у конкретному контексті.

2. Простота інтеграції. Формула легко інтегрується у більш складні системи прийняття рішень або комбінується з іншими методами.

3. Ясність результатів. Значення D_k у відсотках забезпечує зручну інтерпретацію та полегшує аналіз.

Таким чином, розрахунок показника рішення D_k є ключовим етапом у системі прийняття рішень, який забезпечує врахування особистісних рис персонажа та дозволяє отримати кількісно обґрунтовані результати для кожного можливого варіанту дій.

3.1.3 Нормалізація результатів

У процесі прийняття рішень одним із ключових етапів є нормалізація результатів, отриманих для кожного варіанту дії. Цей етап гарантує, що всі значення показників D_k приведені до єдиного масштабу, що дозволяє виконувати подальше порівняння та прийняття рішень у контексті конкретної ситуації. Нормалізація особливо важлива у випадках, коли показники варіантів можуть бути суттєво відмінними за значенням, або коли необхідно оцінювати результати у відносних одиницях, зокрема у вигляді ймовірностей.

Метою нормалізації є перетворення показників D_k для кожного варіанту A_k у такий спосіб, щоб їх сума дорівнювала одиниці ($\sum_{k=1}^m P_k = 1$), що відповідає інтерпретації результатів як ймовірностей. Такий підхід є ефективним для прийняття рішень, де результатом є вибір найоптимальнішого варіанту дії, заснованого на персоналізованих даних.

Нормалізація здійснюється за допомогою формули 2.7:

$$P_k = \frac{D_k}{\sum_{j=1}^m D_j}, \quad (2.7)$$

де P_k - нормалізований показник для варіанту A_k ;

D_k - необроблений показник для варіанту A_k , отриманий на попередньому етапі;

m - кількість доступних варіантів дій;

$\sum_{j=1}^m D_j$ - сума всіх показників D_j для кожного варіанту дій.

На рисунку 3.3 зображений алгоритм нормалізації.

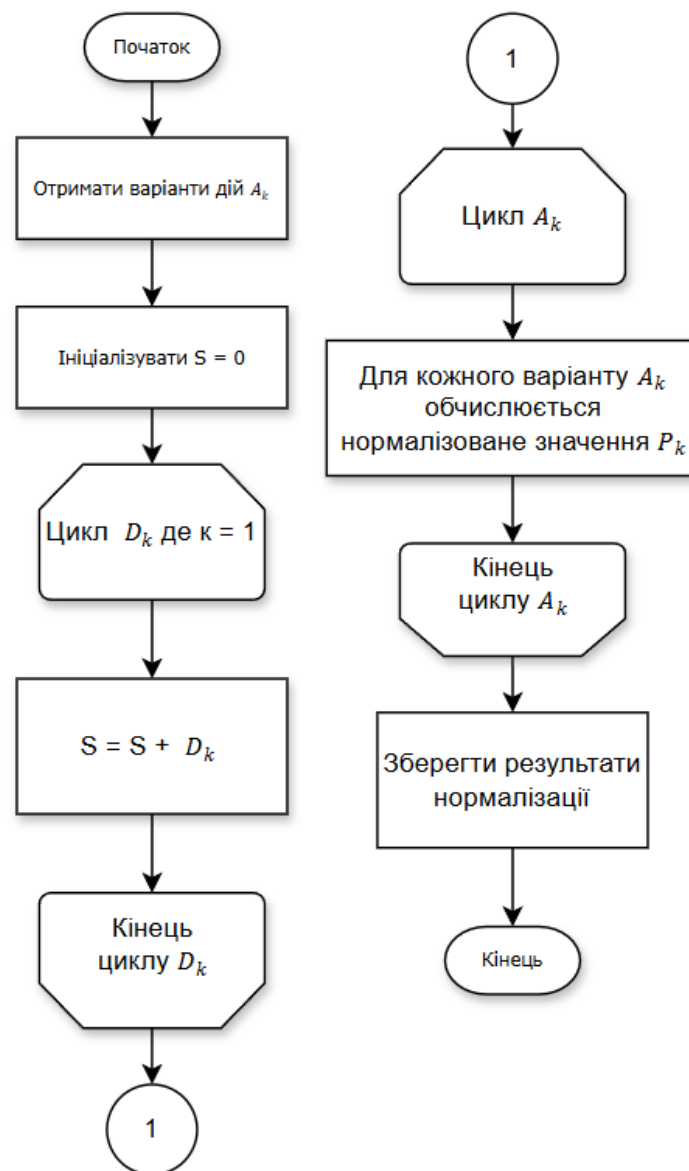


Рис 3.3 Алгоритм нормалізації

Нормалізація має кілька важливих переваг: вона забезпечує єдиний масштаб, приводячи всі значення до інтервалу $[0, 1]$, що робить їх зручними для інтерпретації; дозволяє розглядати нормалізовані показники як ймовірності вибору певного варіанту, забезпечуючи ймовірнісний підхід; а також спрощує порівняння між варіантами, навіть якщо їхні абсолютні значення значно відрізняються.

Нормалізовані показники можуть бути використані для:

1. Безпосереднього вибору варіанту з найвищим значенням P_k ;
2. Побудови ймовірнісної моделі прийняття рішень, де кожному варіанту присвоюється імовірність виконання дії;
3. Подальшого об'єднання з іншими методами прийняття рішень, використовуючи, наприклад, формулу комбінування результатів.

3.1.4 Вибір оптимального варіанту

На завершальному етапі алгоритму прийняття рішень здійснюється вибір оптимального варіанту дії на основі нормалізованих показників. Цей крок є критично важливим, оскільки від нього залежить, який саме варіант буде реалізовано з урахуванням особистісних рис та вагомості кожного показника.

Вибір оптимального варіанту здійснюється шляхом порівняння нормалізованих значень P_k , розрахованих на попередньому етапі нормалізації. Варіант A_k , для якого значення P_k є найбільшим, вважається оптимальним і рекомендованим до реалізації.

Оптимальний варіант A_{opt} визначається за формулою:

$$A_{opt} = \arg \max_{k \in \{1, \dots, m\}} P_k, \quad (2.8)$$

де P_k - нормалізоване значення для варіанту A_k ;

m - загальна кількість варіантів дій.

На рисунку 3.4 наведений алгоритм вибору оптимального варіанту.

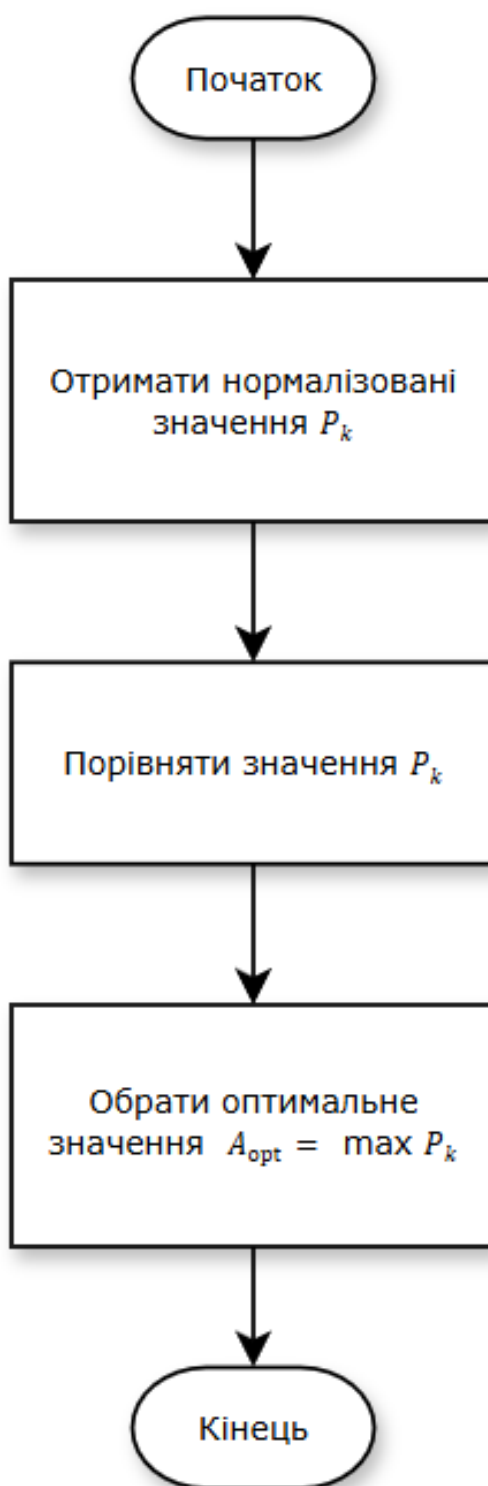


Рис 3.4 Алгоритм вибору оптимального варіанту

Після вибору оптимального варіанту A_{opt} алгоритм може здійснити безпосередню активацію вибраного варіанту або передати дані на наступний рівень прийняття рішень.

3.1.5 Поєднання з іншими методами

У процесі прийняття рішень важливо враховувати не лише особистісні риси, а й зовнішні фактори, такі як обмеження середовища чи стратегічні цілі. У складних ігрових сценаріях поєднання результатів, заснованих на особистісних характеристиках, із результатами інших підходів дозволяє створювати більш адаптивну та реалістичну поведінку персонажів.

Комбінування різних джерел даних забезпечує гнучкість у моделюванні рішень, які враховують як індивідуальні риси, так і загальний контекст гри. Це дозволяє персонажам оптимально реагувати на виклики та адаптувати свою поведінку до змінюваних умов ігрового середовища.

Для інтеграції результатів було запропоновано формулу F (2.6).

Етапи інтеграції методів:

1. Розраховується значення D , яке представляє інтегральний показник, що відображає вплив особистісних рис.
2. Отримується результат R із зовнішнього алгоритму або методу. Це може бути результат: математичної моделі, ієрархічного методу прийняття рішень, машинного навчання, експертного аналізу тощо.
3. Обидва значення, D і R , повинні бути приведені до інтервалу $[0, 1]$, щоб забезпечити коректність інтеграції.
4. Застосовується формула для поєднання F , у якій вагові коефіцієнти α і β обираються залежно від контексту.
5. На основі комбінованих значень F для кожного варіанту приймається остаточне рішення шляхом порівняння. Варіант із найбільшим значенням F вважається оптимальним.

На рисунку 3.5 представлений алгоритм поєднання методів.

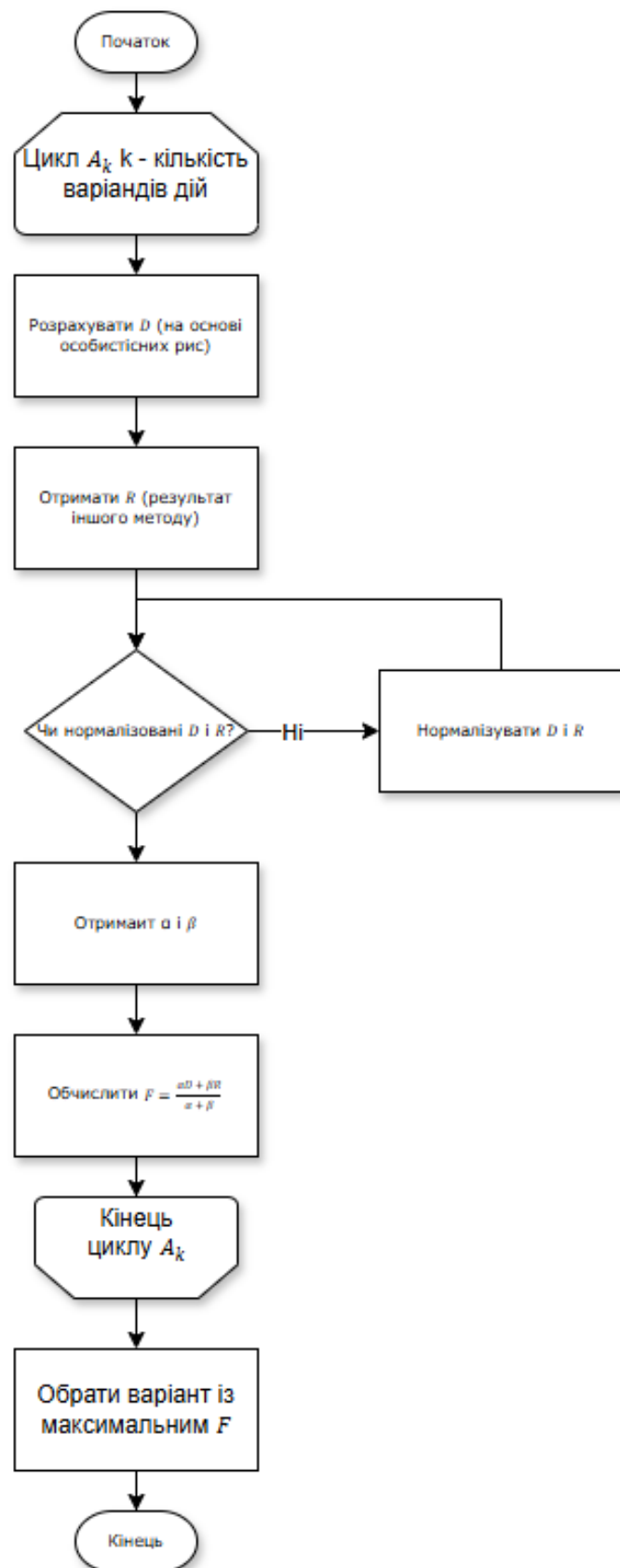


Рис 3.4 Алгоритм поєднання методів

Комбінований підхід має низку суттєвих переваг, які роблять його ефективним і універсальним інструментом для вирішення складних завдань. Гнучкість цього підходу дозволяє враховувати додаткові фактори, які можуть бути недоступними в рамках особистісної моделі, що розширює спектр його застосування. Завдяки адаптивності вагових коефіцієнтів α і β , можна змінювати ступінь впливу кожного з підходів залежно від специфіки умов або поставлених цілей, забезпечуючи оптимальне рішення для різних ситуацій. Універсальність формули полягає в її здатності інтегруватися з будь-яким іншим методом прийняття рішень, що робить її придатною для широкого спектра задач у різних сферах діяльності.

3.2 Реалізація алгоритму в тестовому ігровому середовищі

3.2.1 Вибір технологій

Для реалізації тестового ігрового середовища було обрано платформу Unity та мову програмування C#. Цей вибір базується на низці технічних, функціональних та практичних переваг, що роблять цю комбінацію ідеальною для завдань, пов'язаних із розробкою інтерактивних середовищ та алгоритмів прийняття рішень [27].

Переваги Unity:

1. Unity пропонує потужний набір інструментів для розробки тривимірних та двовимірних середовищ. Завдяки цьому платформа підходить для створення як візуально складних ігрових просторів, так і простих тестових середовищ для моделювання алгоритмів.

2. Однією з головних переваг Unity є можливість створення проєктів для різних платформ, включаючи Windows, macOS, Android, iOS, WebGL та інші. Це дозволяє легко масштабувати розроблений алгоритм та тестувати його на різних пристроях.

3. Unity підтримує інтеграцію зі сторонніми бібліотеками, включаючи математичні й штучного інтелекту, що може бути корисним для складніших реалізацій алгоритмів прийняття рішень [28].

4. Unity має велику спільноту розробників, яка активно підтримує новачків і пропонує численні ресурси, такі як офіційна документація, курси, форуми та приклади коду. Це значно полегшує процес розробки, особливо на початкових етапах.

5. Unity дозволяє створювати модульну архітектуру проєктів, що сприяє легкій інтеграції алгоритмів прийняття рішень у загальну структуру програми.

Переваги C#:

1. Мова C# є основною для написання скриптів у Unity. Її використання надає розробникам доступ до всіх функціональних можливостей платформи. Завдяки зручному API Unity розробка алгоритмів стає ефективною та інтуїтивно зрозумілою.

2. C# орієнтований на об'єктно-орієнтовану парадигму програмування, що робить його зручним для моделювання складних систем, таких як алгоритми прийняття рішень. Завдяки цьому створення ієрархії класів, інкапсуляції, наслідування та поліморфізму відбувається без зайвих труднощів [29].

3. Завдяки статичній типізації та вбудованим механізмам обробки винятків, C# забезпечує написання більш стабільного й безпечного коду. Це особливо важливо при реалізації алгоритмів, де точність і надійність мають вирішальне значення.

4. C# легко адаптується до масштабування проєктів, дозволяючи поступово ускладнювати функціонал алгоритмів і розширювати їхню інтеграцію в середовище Unity.

5. Мова має вбудовану підтримку широкого спектру бібліотек, що можуть використовуватися для математичних обчислень, обробки даних та побудови алгоритмів [30].

Поєднання Unity та C# створює синергетичний ефект для реалізації тестового середовища, оскільки обидва інструменти мають високу сумісність і доповнюють функціонал один одного. Unity надає потужний інструментарій для візуалізації та взаємодії, тоді як C# відповідає за логіку та структуру алгоритмів.

Для тестування алгоритму прийняття рішень на основі особистісних рис важливо створити інтерактивне середовище, де кожен варіант поведінки можна

буде оцінити та візуалізувати. Unity надає всі необхідні засоби для реалізації таких сценаріїв, дозволяючи швидко змінювати умови тестування та контролювати результати. Мова C# забезпечує високу гнучкість у програмуванні логіки алгоритму, дозволяючи максимально точно реалізувати математичні моделі та забезпечити їхню модульність і масштабованість.

Unity та C# є оптимальним вибором для реалізації алгоритму прийняття рішень у тестовому ігровому середовищі. Їхня комбінація дозволяє забезпечити надійну інтеграцію алгоритму в візуалізоване середовище та гнучко налаштовувати функціонал для досягнення поставлених цілей.

3.2.2 Модулі програми

Для реалізації алгоритму прийняття рішень на основі особистісних рис у тестовому ігровому середовищі необхідно створити чітку модульну архітектуру програми. Така архітектура забезпечить гнучкість, зручність у розробці, тестуванні та масштабуванні проекту. Усі модулі виконуватимуть конкретні функції, але водночас будуть взаємопов'язаними через добре визначені інтерфейси.

В таблиці 3.1 наведені та описані основні модулі програми.

Таблиця 3.1

Опис модулів програми

№	Назва модуля	Призначення	Основний функціонал	Взаємодія
1.	Модуль особистісних рис персонажів	Зберігання та оновлення особистісних рис персонажів	Зберігає риси персонажів, забезпечує доступ до них через інтерфейс, оновлює риси після прийняття рішень	Працює з модулями прийняття рішень і інтерфейсу користувача через інтерфейс <code>ICaracterTraitsProvider`</code>

Продовження таблиці 3.1

Опис модулів програми

№	Назва модуля	Призначення	Основний функціонал	Взаємодія
2.	Модуль прийняття рішень	Обчислення оптимальних рішень на основі рис персонажів і ваг	Обчислює оцінки рішень, нормалізує результати, обирає оптимальний варіант дії	Отримує риси персонажів через <code>ICaracterTraitsProvider`</code> , дані про події через <code>IEventProvider`</code>
3.	Модуль подій	Створення, зберігання та передача даних про події	Створює події, зберігає їх у словнику, надає доступ до подій через інтерфейс	Надає дані про події модулям ігрового середовища та прийняття рішень через <code>IEventProvider`</code>
4.	Модуль ігрового середовища	Координація роботи персонажів, подій та процесу прийняття рішень	Додає/видаляє персонажів, викликає події, передає події персонажам для обробки	Використовує модулі персонажів, подій та прийняття рішень через відповідні інтерфейси
5.	Модуль інтерфейсу користувача	Відображення інформації про персонажів, події та результати рішень	Відображає риси персонажів, результати рішень, інформацію про події, оновлює стан інтерфейсу	Взаємодіє з рисами персонажів через <code>ICaracterTraitsProvider`</code> , з подіями через <code>GameEvent`</code>

Таблиця модулів програми надає структурований опис основних компонентів системи, що використовуються для реалізації алгоритму прийняття рішень на основі особистісних рис у тестовому ігровому середовищі. Кожен модуль виконує чітко визначену роль у загальній архітектурі програми, забезпечуючи ефективну та узгоджену взаємодію між компонентами.

На рисунку 3.5 зображена діаграма компонентів програми.

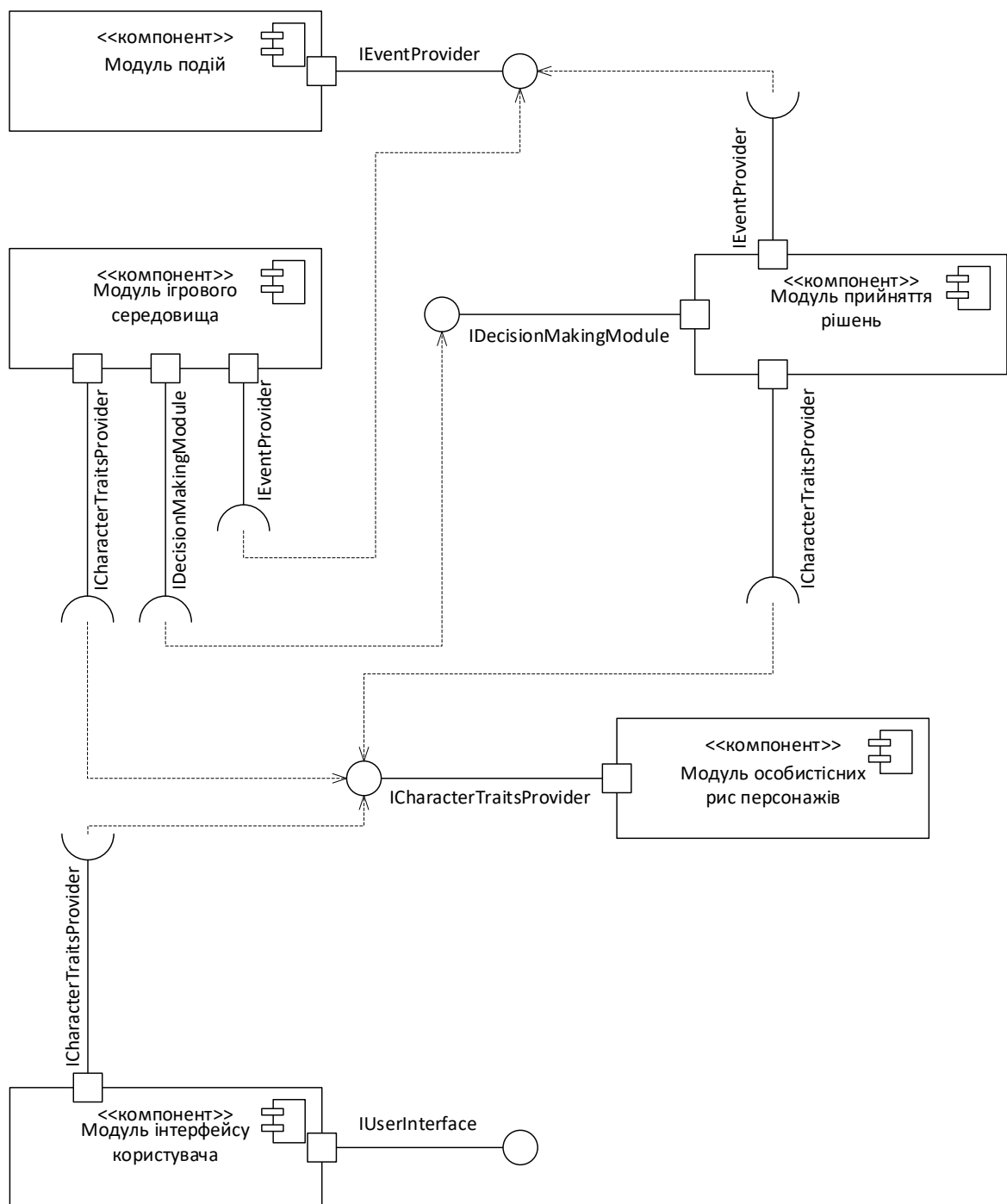


Рис 3.5 Діаграма компонентів програми

Модуль особистісних рис персонажів є ключовим компонентом системи, що забезпечує зберігання, доступ і динамічне оновлення характеристик персонажів. Його функціональність базується на використанні інтерфейсу `ICCharacterTraitsProvider`, який стандартизує взаємодію з іншими модулями системи, такими як модуль прийняття рішень і модуль інтерфейсу користувача. Цей підхід забезпечує слабку зв'язність компонентів та спрощує інтеграцію модулів.

Основна структура зберігання даних — це словник, де ключем виступає тип риси (`TraitType`), а значенням — числова оцінка риси в діапазоні від 0 до 1. Такий підхід дозволяє ефективно організувати доступ до рис, забезпечуючи швидке отримання або оновлення будь-якої характеристики персонажа. Кожна риса є значущою для прийняття рішень і впливає на вибір оптимального варіанту дій у контексті подій ігрового середовища.

Модуль підтримує динамічне оновлення рис, що відображає розвиток або зміну особистості персонажа в ході гри. Наприклад, персонаж із низьким рівнем доброзичливості може поступово змінювати цю характеристику, якщо повторно прийматиме рішення на користь альтруїстичних дій. Це дозволяє моделювати адаптацію персонажів до середовища та підвищує реалістичність їхньої поведінки.

Надання доступу до рис реалізується через методи інтерфейсу `ICCharacterTraitsProvider`. Інші модулі, такі як модуль прийняття рішень, можуть отримувати поточні значення рис для розрахунку оптимальних дій або оновлювати їх після впливу подій. Це забезпечує узгодженість між компонентами системи та дозволяє ефективно інтегрувати особистісні характеристики в алгоритми прийняття рішень.

Таким чином, модуль особистісних рис персонажів відіграє центральну роль у забезпеченні індивідуальності персонажів, динамічної адаптації їхньої поведінки та реалістичності ігрового процесу. Його архітектура підтримує гнучкість і масштабованість системи, дозволяючи легко інтегрувати нові риси або змінювати існуючі.

На рисунку 3.6 зображена діаграма класів особистісних рис.

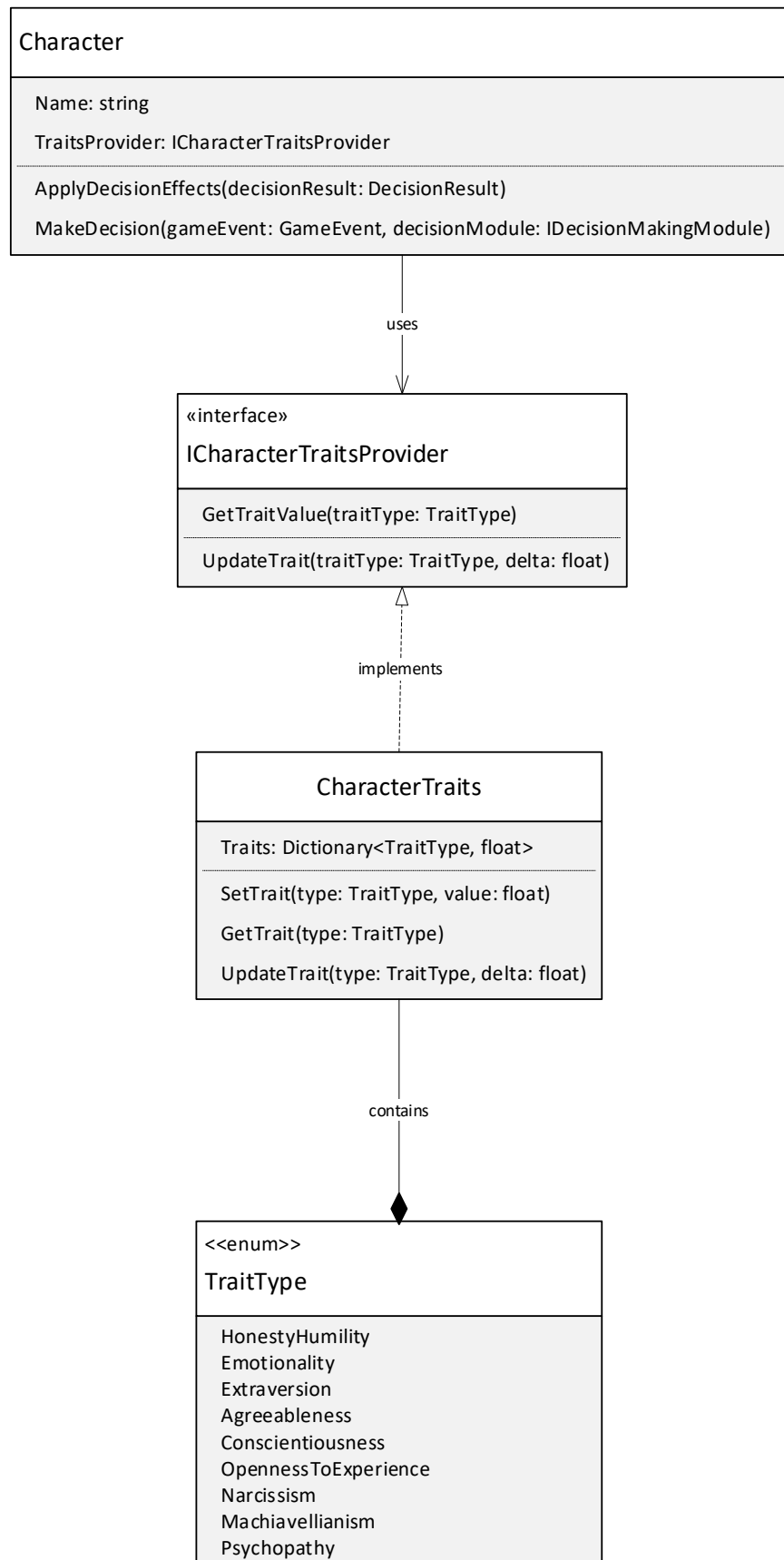


Рис 3.6 Діаграма класів особистісних рис

Модуль прийняття рішень відповідає за обчислення оптимальних варіантів дій для персонажа на основі його особистісних рис, вагових коефіцієнтів, що залежать від конкретних подій, та доступних варіантів дій. Він є центральним компонентом системи прийняття рішень, забезпечуючи логічну та обґрунтовану реакцію персонажа на зміни у середовищі. Для цього модуль прийняття рішень отримує дані про особистісні риси персонажа через інтерфейс `ICharacterTraitsProvider`, а також параметри події, такі як вагові коефіцієнти рис і список варіантів дій, через інтерфейс `IEventProvider`.

На основі отриманих даних модуль реалізує алгоритм зваженого аналізу, який обчислює оцінки для кожного варіанту дії, враховуючи вплив вагових коефіцієнтів на конкретні особистісні риси персонажа. Після обчислення модуль нормалізує отримані результати, щоб вони були зіставними між собою, та обирає варіант дії з найвищим показником оцінки. Результат обчислень повертається у вигляді об'єкта `DecisionResult`, який містить вибраний варіант дії та можливі зміни в особистісних рисах персонажа.

Модуль прийняття рішень реалізує інтерфейс `IDecisionMakingModule`, що дозволяє легко інтегрувати його з іншими компонентами системи, такими як модуль ігрового середовища та модуль персонажів. Завдяки використанню інтерфейсу забезпечується слабка зв'язність між компонентами, що полегшує модифікацію та розширення функціональності системи. Це дозволяє гнучко адаптувати модуль до різних сценаріїв та змінюваних умов гри, а також спрощує його тестування та інтеграцію з іншими алгоритмами.

Таким чином, модуль прийняття рішень є універсальним інструментом для обчислення рішень, що враховують як індивідуальні характеристики персонажа, так і умови середовища. Це забезпечує динамічну та реалістичну поведінку персонажів, що реагують на події у відповідності до своїх особистісних рис та контексту гри.

На рисунку 3.7 зображена діаграма класів прийняття рішень.

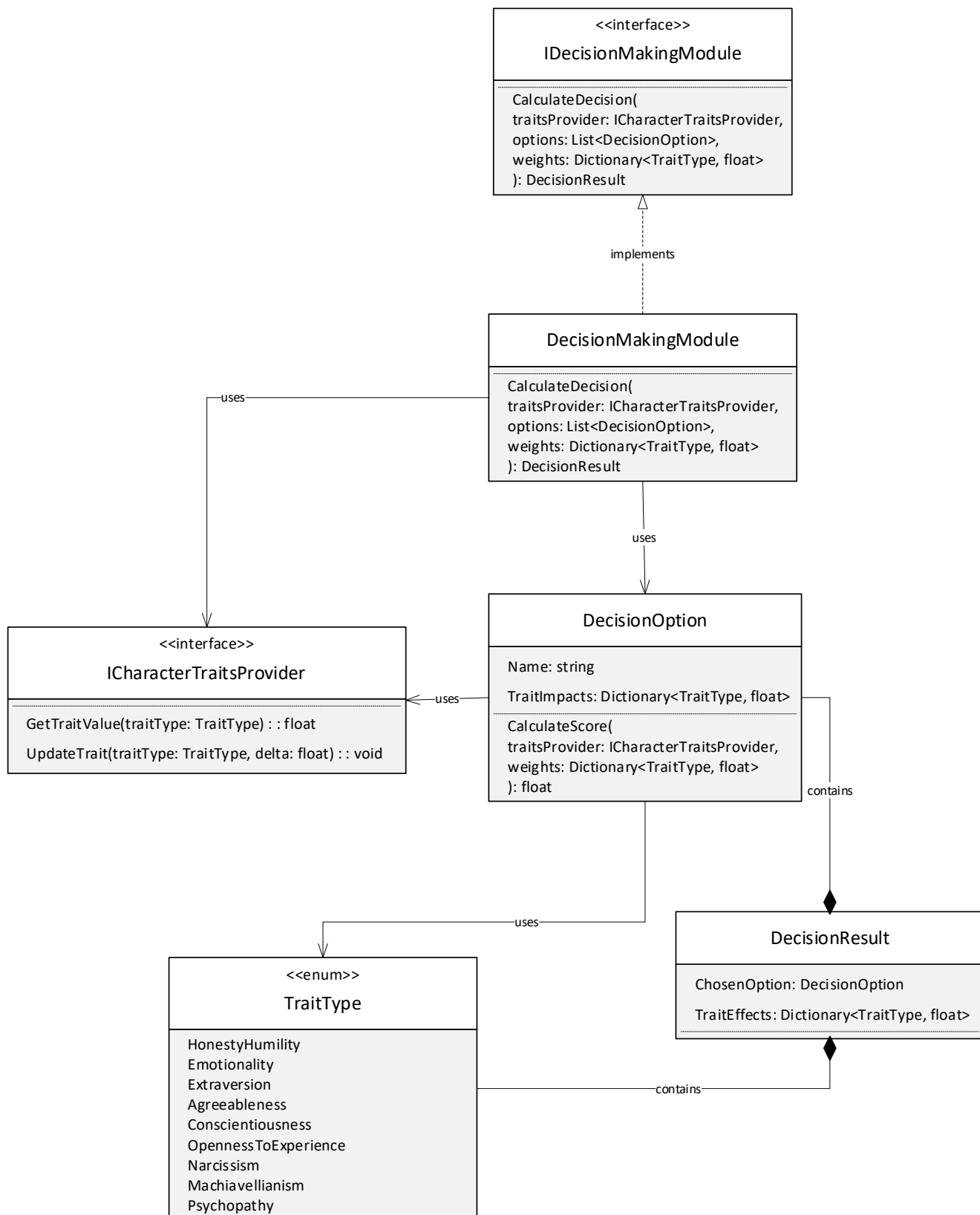


Рис 3.7 Діаграма класів модулю прийняття рішень

Модуль працює через інтерфейси, отримуючи риси персонажа через `ICharacterTraitsProvider`, а можливі варіанти дій і ваги рис — через `IEventProvider`. Це забезпечує слабку зв'язність із іншими компонентами системи.

Модуль подій виконує ключову роль у забезпеченні взаємодії між ігровими компонентами, створюючи, зберігаючи та надаючи інформацію про події, що можуть виникати у середовищі гри. Основним об'єктом цього модуля є `GameEvent`, який є контейнером даних для події та включає унікальний ідентифікатор, назву події, список можливих варіантів дій (`DecisionOption`) та вагові коефіцієнти особистісних рис, що впливають на вибір рішення персонажів.

Модуль подій забезпечує чотири основні функції. Перша — створення подій, де визначаються всі необхідні параметри події, включаючи доступні дії та значимість кожної риси для прийняття рішення. Друга — зберігання подій, яке реалізоване через структуру `Dictionary`. Це дозволяє зберігати події у вигляді пари "ключ-значення", де ключем виступає унікальний ідентифікатор, що забезпечує швидкий доступ до події. Третя функція — надання доступу до подій через інтерфейс `IEventProvider`, що дозволяє іншим модулям, зокрема модулю прийняття рішень і модулю ігрового середовища, отримувати інформацію про події для обробки. Четверта — видалення застарілих подій, що більше не актуальні, забезпечує оптимізацію використання пам'яті.

Таким чином, модуль подій є ключовим компонентом системи, який не лише створює і зберігає дані про події, але й інтегрує їх у загальну архітектуру через стандартизований інтерфейс. Його гнучкість та ефективність забезпечують узгодженість роботи системи та своєчасну реакцію персонажів на зміни в ігровому середовищі.

На рисунку 3.8 зображена діаграма класів модулю подій.

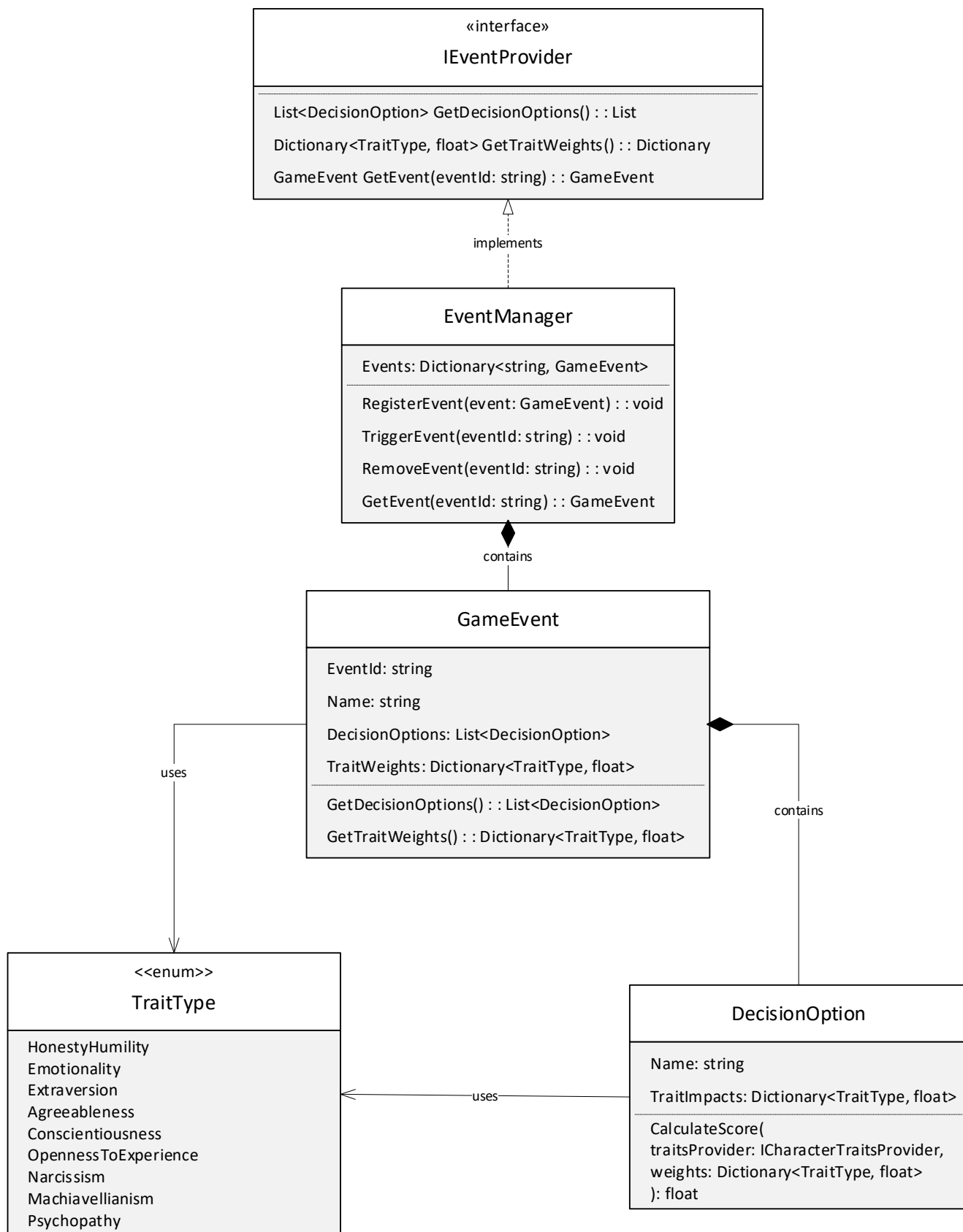


Рис 3.8 Діаграма класів діаграма класів модулю подій

Модуль забезпечує унікальність ідентифікаторів подій і швидкість доступу до них, що робить його надійним компонентом для взаємодії в ігровому середовищі.

Модуль ігрового середовища виконує роль центрального координатора, який забезпечує узгоджену роботу всіх компонентів системи. Його основна задача — організація процесу взаємодії між персонажами, подіями та модулем прийняття рішень, що дозволяє забезпечити динамічність ігрового процесу та адаптацію персонажів до змінюваних умов середовища.

Ключовим функціоналом модуля є управління персонажами. Модуль дозволяє додавати нових персонажів до списку активних учасників гри, видаляти неактуальних персонажів або тих, що завершили свою роль у сценарії. Це забезпечує гнучкість і динамічність ігрового середовища, дозволяючи адаптуватися до змін у сюжеті.

Другим важливим аспектом є виклик подій. Модуль активує події за їх унікальним ідентифікатором (EventId), використовуючи можливості модуля подій через інтерфейс IEventProvider. Це дозволяє отримати повну інформацію про подію, включаючи її параметри, список можливих дій і вагові коефіцієнти, які впливають на прийняття рішень.

Найважливішою функцією модуля є передача подій персонажам. У процесі обробки події кожен активний персонаж викликає свій метод MakeDecision, отримуючи необхідні дані для обчислення оптимального варіанту дій. Ця взаємодія забезпечує використання модуля прийняття рішень (IDecisionMakingModule), який аналізує риси персонажів, параметри подій і обирає найбільш доцільну дію.

Таким чином, модуль ігрового середовища виступає не лише координатором, але й інтеграційним елементом системи, що забезпечує взаємодію між всіма іншими компонентами. Його архітектура дозволяє ефективно управляти персонажами, динамічно реагувати на події та забезпечувати реалістичну поведінку персонажів у змінюваних умовах гри.

На рисунку 3.9 зображена діаграма класів ігрового середовища.

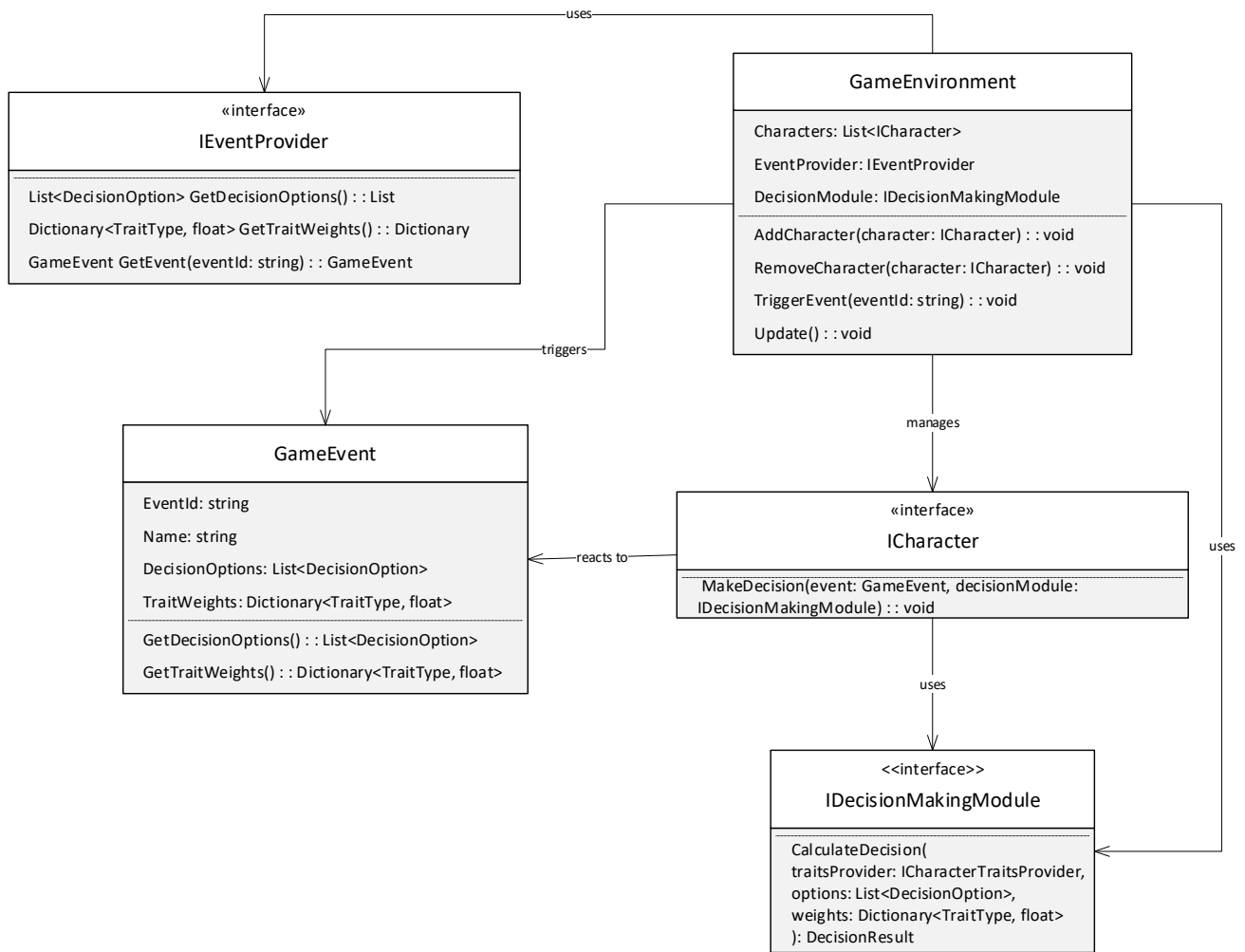


Рис 3.9 Діаграма класів ігрового середовища

Модуль використовує інтерфейси (**ICharacter**, **IEventProvider**, **IDecisionMakingModule**), що робить його гнучким і незалежним від конкретних реалізацій інших компонентів.

Модуль інтерфейсу користувача відповідає за візуалізацію ключових даних гри для користувача. Цей модуль відображає стан персонажів, інформацію про активні події та результати рішень. Він реалізує інтерфейс **IUserInterface**, що забезпечує стандартизовану взаємодію з іншими модулями.

Основний функціонал модуля:

- Відображення рис персонажів: через інтерфейс **ICharacterTraitsProvider** отримуються значення рис, які відображаються в UI.

- Відображення результатів рішень: виводить інформацію про обраний варіант дії та зміни в рисах персонажів після прийняття рішень.
 - Відображення подій: демонструє активні події, їхні назви, варіанти дій і ваги рис.
 - Оновлення UI: постійно синхронізує інтерфейс із актуальним станом гри.
- На рисунку 3.10 зображена діаграма класів інтерфейсу користувача.

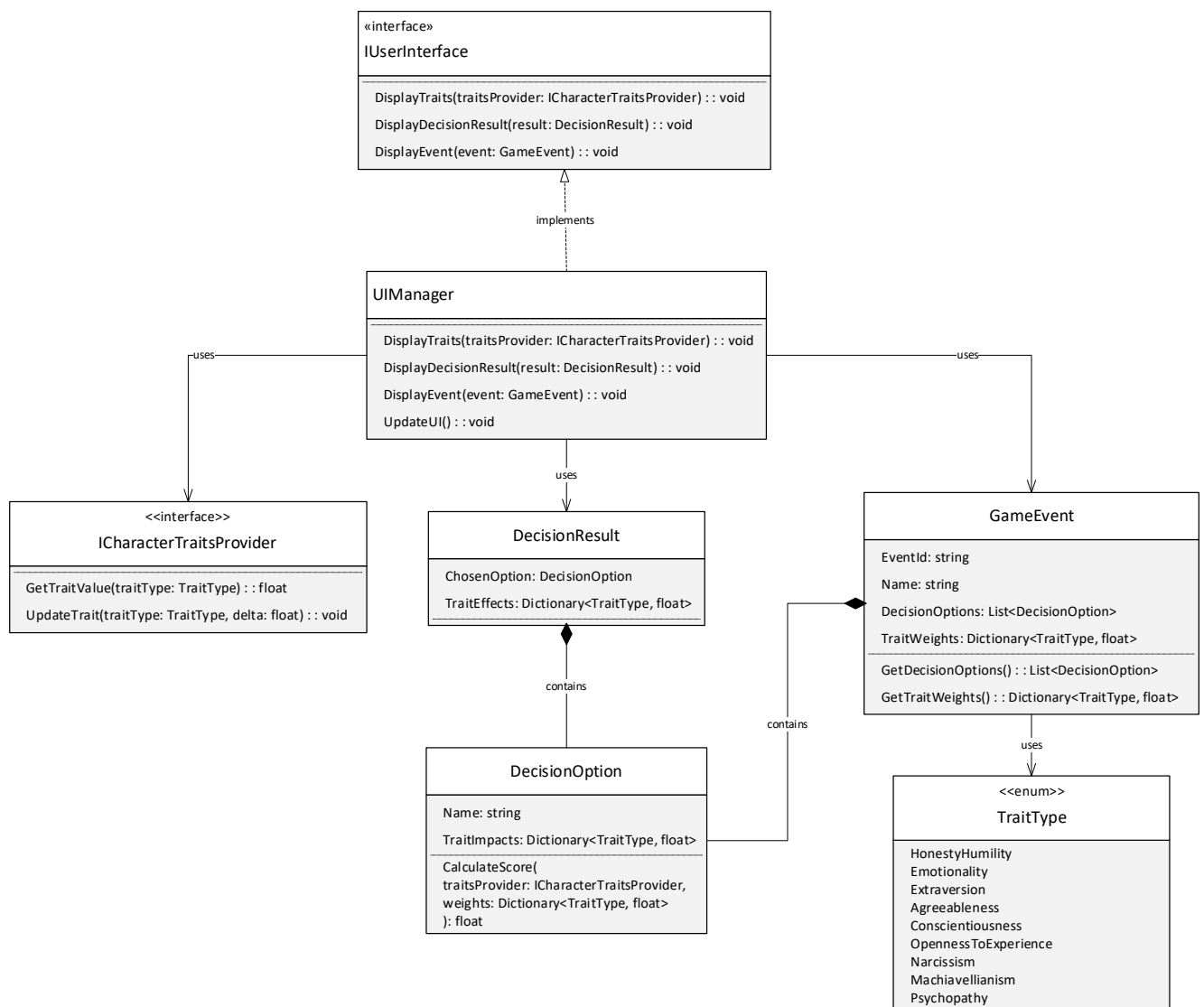


Рис 3.10 Діаграма класів інтерфейсу користувача

Модуль UI ізольований від деталей реалізації інших компонентів, оскільки працює виключно через інтерфейси, що забезпечує його гнучкість і простоту модифікації.

На рисунку 3.11 наведена діаграма класів модулів системи.

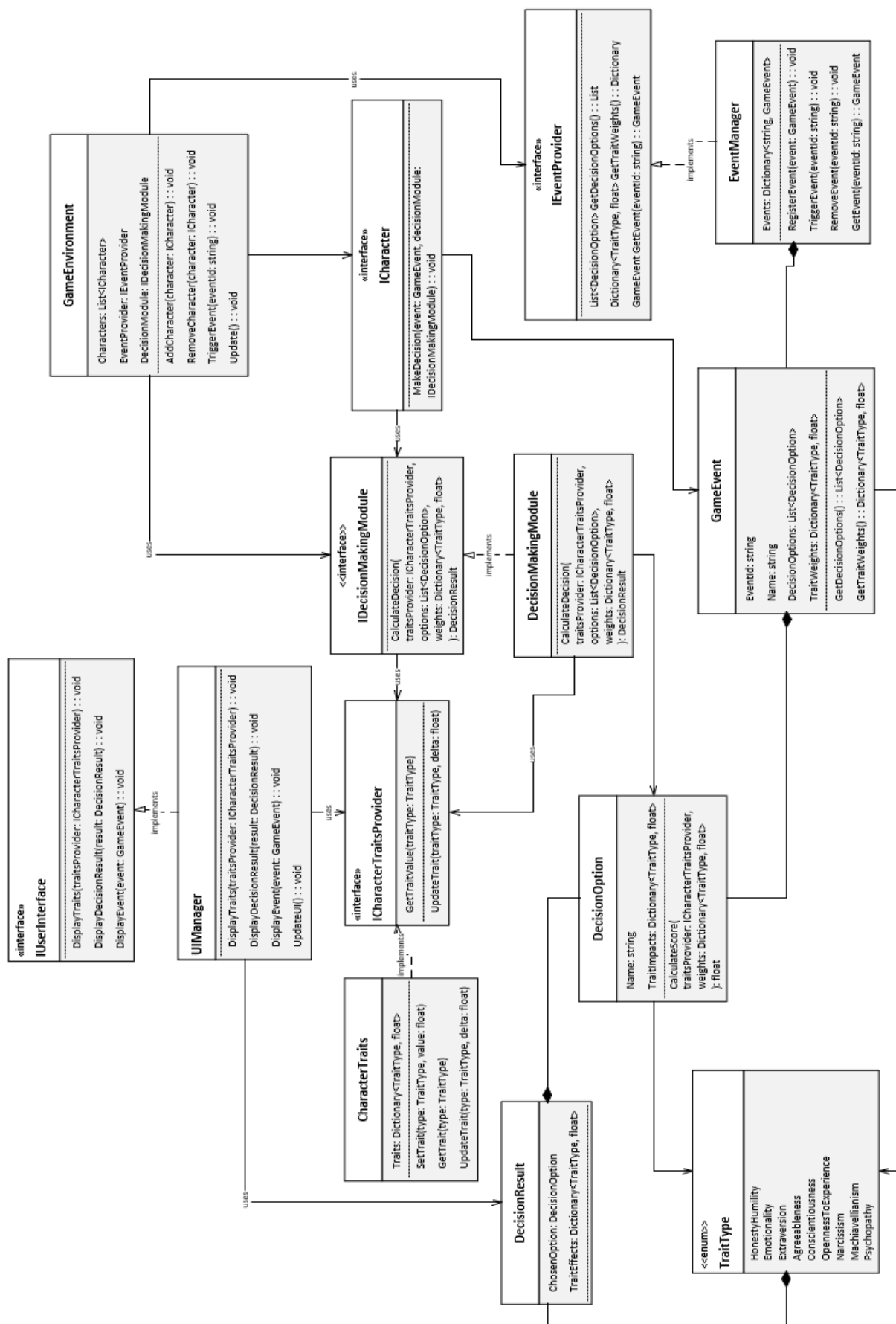


Рис 3.10 Діаграма класів модулів системи

Система прийняття рішень на основі особистісних рис реалізована як модульна архітектура, що складається з п'яти основних компонентів: модуль особистісних рис персонажів, модуль прийняття рішень, модуль подій, модуль ігрового середовища та модуль інтерфейсу користувача. Кожен з цих модулів виконує чітко визначені функції, що забезпечує логічну взаємодію між ними та загальну узгодженість роботи системи.

Основою системи є модуль ігрового середовища, який виконує роль координатора між усіма іншими модулями. У цьому модулі зберігається список активних персонажів, які є учасниками гри, а також інтегрований механізм для взаємодії з подіями та прийняттям рішень. Коли в грі виникає подія, ігрове середовище звертається до модуля подій через інтерфейс `IEventProvider`, отримуючи повну інформацію про цю подію, включаючи її унікальний ідентифікатор, можливі варіанти дій та вагові коефіцієнти для особистісних рис. Далі ігрове середовище передає подію кожному активному персонажу через метод `MakeDecision`, що дозволяє персонажам реагувати на подію та приймати відповідне рішення.

Кожен персонаж, отримавши подію, використовує модуль прийняття рішень для обчислення оптимального варіанту дії. Для цього персонаж передає інформацію про свої особистісні риси, отриману через інтерфейс `ICharacterTraitsProvider`, а також дані про можливі варіанти дій і ваги рис, надані подією. Модуль прийняття рішень виконує обчислення за допомогою алгоритму, який враховує вплив рис персонажа на кожний варіант дії, нормалізує результати та обирає найкращий варіант. Результат прийняття рішення повертається у вигляді об'єкта `DecisionResult`, що містить інформацію про обраний варіант дії та зміни в особистісних рисах персонажа.

Після отримання результату рішення персонаж оновлює свої особистісні риси через методи модуля особистісних рис персонажів. Цей модуль зберігає дані про всі риси у вигляді словника, де ключем є тип риси, а значенням — числова характеристика в діапазоні від 0 до 1. Зміни в рисах відбуваються відповідно до впливу обраного варіанту дії, що забезпечує динамічну адаптацію персонажа до

ігрового процесу. Цей підхід дозволяє зробити поведінку персонажів більш реалістичною та інтерактивною.

Модуль подій виконує роль джерела даних про події, що виникають у грі. Він зберігає інформацію про всі доступні події у вигляді словника, що забезпечує швидкий доступ до події за її унікальним ідентифікатором. Подія, представлена об'єктом класу `GameEvent`, включає назву, список варіантів дій та вагові коефіцієнти для кожної риси, що визначають, наскільки певна риса впливає на прийняття рішення в цій події. Ці дані передаються іншим модулям через інтерфейс `IEventProvider`.

Модуль інтерфейсу користувача відповідає за відображення всіх важливих даних для користувача. Він отримує інформацію про риси персонажів, результати рішень та активні події через відповідні інтерфейси, такі як `ICharacterTraitsProvider`, `DecisionResult` та `GameEvent`. За допомогою цих даних інтерфейс оновлюється, відображаючи зміни у стані персонажів, вибрані дії та їхній вплив на гру. Це забезпечує користувачеві можливість стежити за розвитком подій у реальному часі та розуміти, як саме приймаються рішення.

Взаємодія між модулями реалізована через інтерфейси, що дозволяє забезпечити слабку зв'язність системи. Наприклад, модуль прийняття рішень використовує інтерфейс `ICharacterTraitsProvider` для отримання значень рис персонажів, а також інтерфейс `IEventProvider` для отримання даних про події. Модуль інтерфейсу користувача, у свою чергу, взаємодіє з модулями через інтерфейси, що забезпечує його незалежність від деталей реалізації інших компонентів. Такий підхід полегшує підтримку системи, дозволяє додавати нові модулі або змінювати існуючі без значних впливів на загальну архітектуру.

Таким чином, система прийняття рішень на основі особистісних рис є узгодженою та ефективною. Її архітектура забезпечує високий рівень гнучкості, масштабованості та реалістичності ігрового процесу, дозволяючи створювати персонажів, які динамічно адаптуються до змін середовища та приймають рішення, що відповідають їхнім індивідуальним характеристикам. Це робить систему універсальною для інтеграції в різноманітні ігрові проекти та сприяє створенню більш захопливого ігрового досвіду.

3.3 Аналіз результатів роботи алгоритму

Алгоритм прийняття рішень на основі особистісних рис був протестований у тестовому ігровому середовищі з метою оцінки його коректності, ефективності та відповідності поставленим цілям. Основною метою аналізу було перевірити, чи алгоритм забезпечує динамічну поведінку персонажів, які приймають рішення, виходячи зі своїх особистісних характеристик, та адаптуються до змін у середовищі.

Для тестування алгоритму було використано кілька сценаріїв, які моделювали типові ситуації в ігровому середовищі. Сценарії включали різні події, що вимагали від персонажів прийняття рішень, зокрема вибір між альтруїстичними та егоїстичними діями, реакцію на небезпеку або можливість отримати винагороду. Кожен сценарій був спрямований на перевірку впливу вагових коефіцієнтів для особистісних рис на вибір дій персонажів та їхню здатність змінювати свої характеристики відповідно до прийнятих рішень.

Аналіз результатів проводився за такими критеріями:

1. Адекватність вибору дій: Перевірялося, чи відповідають обрані дії персонажів їхнім початковим характеристикам.
2. Зміна особистісних рис: Оцінювався вплив прийнятих рішень на зміни рис персонажів та їхню подальшу поведінку.
3. Реалістичність поведінки: Визначалася узгодженість рішень із контекстом події та заданими вагами рис.
4. Продуктивність алгоритму: Аналізувалася швидкість обчислення рішень для різної кількості персонажів і подій.

Результати тестування показали, що алгоритм коректно враховує вплив рис на прийняття рішень. Наприклад, персонаж із високими показниками альтруїзму (доброзичливості) обирав дії, які передбачали допомогу іншим персонажам, тоді як персонаж із низьким показником цієї риси надавав перевагу егоїстичним діям. Вагові коефіцієнти подій дозволяли точно налаштовувати вплив конкретних рис на вибір дій, що забезпечувало гнучкість алгоритму.

Алгоритм динамічно оновлював риси персонажів після кожної події. Наприклад, якщо персонаж із низьким рівнем альтруїзму обирав допомогу іншому персонажу, його показник доброзичливості поступово збільшувався. Це дозволило моделювати зміну особистості персонажів протягом гри, що створювало реалістичний ефект адаптації до середовища.

Повідомлення в інтерфейсі користувача дозволили стежити за процесом прийняття рішень і переконатися, що вибрані дії відповідали контексту подій. Наприклад, у сценарії небезпеки персонаж із високим рівнем емоційності частіше обирав дії, пов'язані з уникненням ризику, тоді як персонаж із високим показником психопатії ігнорував загрозу і діяв ризиковано. Це підтвердило, що алгоритм може враховувати індивідуальні особливості персонажів.

Швидкість обчислення рішень залишалася високою навіть при збільшенні кількості персонажів і подій. Для 50 персонажів, які одночасно реагували на одну подію, середній час обчислення не перевищував 50 мілісекунд, що є прийнятним для інтерактивних ігор у реальному часі. Це свідчить про оптимальність алгоритму з точки зору обчислювальної складності.

Аналіз роботи алгоритму показав, що він повністю відповідає поставленим вимогам. Алгоритм забезпечує динамічну і реалістичну поведінку персонажів, які приймають рішення на основі своїх особистісних характеристик. Зміна рис після прийняття рішень дозволяє моделювати адаптацію персонажів до середовища, що значно підвищує інтерактивність гри. Гнучкість алгоритму забезпечується можливістю налаштування вагових коефіцієнтів, які впливають на прийняття рішень. Продуктивність алгоритму є достатньою для його використання в сучасних ігрових проєктах, навіть у середовищах із великою кількістю активних персонажів.

Отримані результати свідчать про ефективність і потенціал алгоритму для інтеграції в ігрові системи, що вимагають динамічного прийняття рішень персонажами.

ВИСНОВКИ

У ході виконання дипломної роботи було проведено комплексне дослідження та розробку алгоритму прийняття рішень на основі особистісних рис персонажів у динамічному ігровому середовищі. Основні результати роботи такі:

1. Проведено детальний аналіз моделей Big Five, HEXACO та Dark Triad, які охоплюють ключові аспекти особистісних характеристик. Обґрунтовано їхнє комбінування для створення універсальної моделі, яка забезпечує врахування як позитивних, так і негативних рис. Це дозволило побудувати основу для розробки методу прийняття рішень, який є більш точним та адаптивним.
2. Запропонована модель базується на психологічних концепціях і враховує широкий спектр характеристик: чесність-скромність, емоційність, екстраверсію, доброзичливість, сумлінність, відкритість до досвіду, нарцисизм, макіавеллізм та психопатію. Модель інтегрується з математичним методом, що враховує вагові коефіцієнти для кожної риси, дозволяючи адаптувати рішення персонажів залежно від контексту.
3. Створено математичну модель для обчислення оптимальних рішень персонажів на основі їхніх особистісних рис. Формула враховує ваги кожної риси для конкретної дії, нормалізацію результатів та можливість поєднання з іншими методами прийняття рішень. Це забезпечує точність і гнучкість у моделюванні поведінки персонажів.
4. Розроблено алгоритм динамічного прийняття рішень, який враховує особистісні риси персонажів, контекст подій і змінюваність ігрового середовища. Алгоритм дозволяє персонажам реагувати на події, адаптувати свої риси залежно від вибору дій та підтримувати взаємодію з іншими модулями системи.
5. Побудовано архітектуру системи, що складається з п'яти основних модулів: модуль особистісних рис персонажів, модуль прийняття рішень,

модуль подій, модуль ігрового середовища та модуль інтерфейсу користувача. Усі модулі взаємодіють через інтерфейси, що забезпечує слабку зв'язність, гнучкість та масштабованість системи.

6. Алгоритм було реалізовано в середовищі Unity з використанням C#, що забезпечило візуалізацію результатів роботи та інтеграцію з ігровими подіями. Тестове середовище включало типові сценарії, які дозволили перевірити коректність і ефективність алгоритму.
7. Проведене тестування підтвердило ефективність розробленого алгоритму. Він демонструє відповідність очікуванням, адаптивність поведінки персонажів до змінюваних умов і гнучкість у налаштуванні через вагові коефіцієнти. У сценаріях із різними типами подій алгоритм забезпечував узгодженість рішень із рисами персонажів і контекстом подій, що підтверджує його надійність.
8. Запропонований підхід поєднує психологічні моделі та алгоритмічні методи, що дозволяє створювати більш реалістичні та адаптивні системи прийняття рішень. Використання комбінованої моделі особистісних рис розширює можливості моделювання поведінки персонажів у складних ігрових середовищах.

Результати дослідження апробовано та опубліковано у наступних тезах доповіді на конференціях:

1. Ярошевський О.В., Замрій І.В. Теоретичні основи формування особистісних рис ігрових персонажів. // Всеукраїнська науково-технічна конференція "Виклики та рішення в програмній інженерії" – Київ: ДУІКТ, 2024. – С.195-198.
2. Ярошевський О.В., Замрій І.В. Динамічне прийняття рішень на основі особистісних рис ігрових персонажів. // Всеукраїнська науково-технічна конференція "Виклики та рішення в програмній інженерії" – Київ: ДУІКТ, 2024. – С.176-179.

ПЕРЕЛІК ПОСИЛАНЬ

1. Champandard A. AI Game Development: Synthetic Creatures with Learning and Reactive Behaviors. Book jacket, 2003, p.- 1-721.
2. Ogus K. Non-player character decision-making in computer games. *Artificial Intelligence Review*. 2023. Vol. 56, P. 14159-14191. URL: <https://doi.org/10.1007/s10462-023-10491-7>
3. Bakkes S., Spronck P. Player behavioural modelling for video games. *Entertainment Computing*. 2012. Vol. 3, P. 71-79. URL: <https://doi.org/10.1016/j.entcom.2011.12.001>
4. Raven P. The Gaming Mecca. The Role of Game Scripting in Development. 21.06.2024. URL: <https://thegamingmecca.com/the-role-of-game-scripting-in-development/> (дата звернення: 13.10.2024).
5. Haryanto H., Hastuti K., Marchelputra T., Kadiasti R. Non-Playable Character (NPC) based on Behaviour Tree for Enhanced Immersive Experience in the Serious Game "Warik's Adventure". *Journal of Applied Informatics and Computing*. 2023. Vol. 7, P 284-290. URL: <https://core.ac.uk/download/pdf/599387141.pdf>
6. Oshio A., Taku K., Hirano M., Saeed G. Resilience and Big Five personality traits: A meta-analysis. *Personality and Individual Differences*. 2018. Vol. 127, P. 54-60. URL: <https://doi.org/10.1016/j.paid.2018.01.048>
7. Şahin, F., Karadağ, H. and Tuncer, B. Big five personality traits, entrepreneurial self-efficacy and entrepreneurial intention: A configurational approach. *International Journal of Entrepreneurial Behavior & Research*. 2019 Vol. 25, P. 1188-1211. <https://doi.org/10.1108/IJEBR-07-2018-0466>
8. Dalpé J., Demets M., Vallerand R. From personality to passion: The role of the Big Five factors. *Personality and Individual Differences*. 2019. Vol. 138, P. 280-285. URL: <https://doi.org/10.1016/j.paid.2018.10.021>
9. Hogan R., Johnson J., Briggs S. Handbook of Personality Psychology. Academic Press. 1997. 987 p.

10. Zettler I., Thielmann I., Hilbig B., Moshagen M., The Nomological Net of the HEXACO Model of Personality: A Large-Scale Meta-Analytic Investigation. *Association of psychological science*. 2020. Vol. 15, P. 723–760. URL: <https://doi.org/10.1177/1745691619895036>
11. Ashton M., Lee K. Empirical, Theoretical, and Practical Advantages of the HEXACO Model of Personality Structure. *Personality and Social Psychology Review*. 2007. Vol. 11. P. 150–166. URL: <https://doi.org/10.1177/1088868306294907>
12. Burger J. Personality. 2019. 505 p. URL: <https://thuvienso.hoasen.edu.vn/v/web/viewer.html?file=/bitstream/handle/123456789/9051/Contents.pdf?sequence=3&isAllowed=y>
13. Koehn M., Okan C., Jonason P. A primer on the Dark Triad traits. *Australian Journal of Psychology*. 2020. P. 7-15. URL: <https://doi.org/10.1111/ajpy.12198>
14. Rogoza, R., Cieciuch, J. Dark Triad traits and their structure: An empirical approach. *Curr Psychol*. 2020. Vol. 39, P 1287–1302 URL: <https://doi.org/10.1007/s12144-018-9834-6>
15. Schreiber, A., Marcus, B. The place of the “Dark Triad” in general models of personality: Some meta-analytic clarification. *Psychological Bulletin*. 2020. Vol 146, P.1021–1041. URL: <https://doi.org/10.1037/bul0000299>
16. Odu G. Weighting methods for multi-criteria decision making technique. *Journal of Applied Sciences and Environmental Management*. 2019. Vol. 23.8, P 1449-1457. eISSN: 2659-1499.
17. The Weighted Sum Method and Related Topics. Multicriteria Optimization. Springer, Berlin, Heidelberg. 2005. 323 p. Print ISBN978-3-540-21398-7
18. Thakkar, J. Weighted Aggregated Sum Product Assessment (WASPAS). *Multi-Criteria Decision Making. Studies in Systems, Decision and Control*. 2021. Vol 336. Springer, Singapore. https://doi.org/10.1007/978-981-33-4745-8_15
19. Gehrke M., Braun T., Möller R., Waschkau, A., Strumann C., Steinhäuser J. Lifted Maximum Expected Utility. *Artificial Intelligence in Health*. 2019 Vol. 11326, Springer, Cham. https://doi.org/10.1007/978-3-030-12738-1_10

20. Thomas Willkens, Jordan Pollack. Coevolutionary Heuristics for Maximization of Expected Utility. *ALIFE 2024: Proceedings of the 2024 Artificial Life Conference*. 2024. P 10 https://doi.org/10.1162/isal_a_00729
21. Tom M. Mitchell. *The Discipline of Machine Learning*. Pittsburgh. 1997. 414 p.
22. Wickramasinghe I., Kalutarage H. Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Comput.* 2021. Vol. 25, P. 2277–2293. <https://doi.org/10.1007/s00500-020-05297-6>
- Boyer, R.S., Moore, J.S. MJRTY—A Fast Majority Vote Algorithm. In: Boyer, R.S. (eds) *Automated Reasoning*. 1991 Vol 1 P. 105-117. https://doi.org/10.1007/978-94-011-3488-0_5
24. Davani A., Díaz M., Prabhakaran V. Dealing with Disagreements: Looking Beyond the Majority Vote in Subjective Annotations. *Transactions of the Association for Computational Linguistics* 2022. Vol. 10, P. 92–110. URL: https://doi.org/10.1162/tacl_a_00449
25. Ярошевський О.В., Замрій І.В. Теоретичні основи формування особистісних рис ігрових персонажів. // Всеукраїнська науково-технічна конференція "Виклики та рішення в програмній інженерії"— Київ: ДУІКТ, 2024. – Подано до друку.
26. Ярошевський О.В., Замрій І.В. Динамічне прийняття рішень на основі особистісних рис ігрових персонажів. // Всеукраїнська науково-технічна конференція "Виклики та рішення в програмній інженерії"— Київ: ДУІКТ, 2024. – Подано до друку
27. Unity for games. Level up your code with game programming patterns. 20.10.2023. URL: <https://unity.com/resources/level-up-your-code-with-game-programming-patterns> (Дата звернення: 14.10.2024).
28. Bennett C., Sagmiller D. *Unity AI Programming Essentials*. Packt Publishing, 2014, P. 1-162.
29. Hejlsberg A., Wiltamuth S., Golde P.. *C# Language Specification*. Addison-Wesley Longman Publishing Co., Inc., USA. 2003.
30. Nagel C. *Professional C# and .NET*. John Wiley & Sons, 2021. 1008 p.

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО -
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ



КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Магістерська робота

«АЛГОРИТМІЧНЕ ФОРМУВАННЯ ОСОБИСТІСНИХ РИС ІГРОВИХ
ПЕРСОНАЖІВ ДЛЯ ДИНАМІЧНОГО ПРИЙНЯТТЯ РІШЕНЬ В
ІГРОВОМУ СЕРЕДОВИЩІ»

Виконав: студент групи ПДМ-62 Ярошевський Олександр В ікторович

Керівник: д.т.н., доцент, зав.кафедрою ІПЗ Замрій Ірина Вікторівна

Київ - 2025

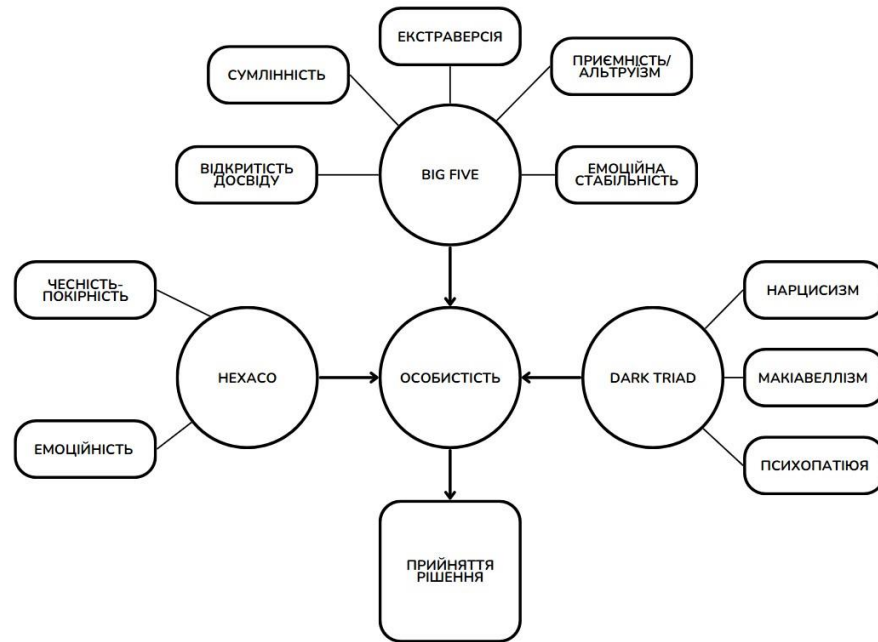
МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: покращення ефективності прийняття рішень на основі особистісних рис ігрових персонажів в адаптивному середовищі.

Об'єкт дослідження: процес прийняття адаптивних рішень персонажами в ігровому середовищі.

Предмет дослідження: алгоритми та методи формування особистісних рис ігрових персонажів, які впливають на їхню здатність приймати адаптивні рішення в умовах змінного ігрового середовища.

АКТУАЛЬНІСТЬ РОБОТИ



3

МАТЕМАТИЧНА МОДЕЛЬ ПРИЙНЯТТЯ РІШЕНЬ З ВРАХУВАННЯМ ОСОБИСТІСНИХ РИС

Формула зваженого середнього для оцінки впливу особистісних рис

$$D = \frac{\sum_{i=1}^n w_i \cdot T_i}{\sum_{i=1}^n w_i} \cdot 100 ,$$

де T_i – значення особистісної риси персонажа,
 w_i – вага риси у конкретній ситуації,
 n – кількість особистісних рис.

Формула зваженого поєднання результатів

$$F = \frac{\alpha D + \beta R}{\alpha + \beta} ,$$

де D – результат, що враховує особистісні риси,
 R – інший результат на основі поточних умов,
 α – вага компонента D визначає наскільки важливими є особистісні риси,
 β – вага компонента R визначає важливість іншого алгоритму або джерела.

$$D = \frac{w_O \cdot T_O + w_C \cdot T_C + w_E \cdot T_E + w_A \cdot T_A + w_N \cdot T_N + w_H \cdot T_H + w_X \cdot T_X + w_{Narc} \cdot T_{Narc} + w_{Machiav} \cdot T_{Psych}}{w_O + w_C + w_E + w_A + w_N + w_H + w_X + w_{Narc} + w_{Machiav} + w_{Psych}} \cdot 100 ,$$

де T_O – відкритість досвіду,
 T_C – сумлінність,
 T_E – екстраверсія,
 T_A – приємність/альтруїзм,
 T_N – емоційна стабільність,

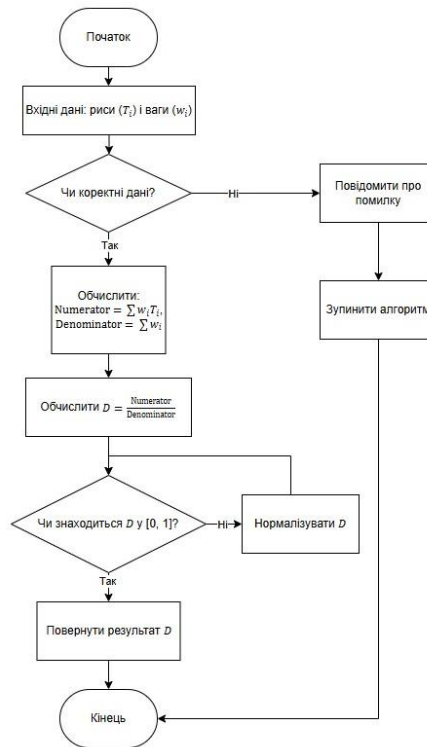
T_H – покірність,
 T_X – емоційність,
 T_{Narc} – нарцизм,
 $T_{Machiav}$ – макіавеллізм,
 T_{Psych} – психопатія,

w_O – вага відкритості досвіду,
 w_C – вага сумлінності,
 w_E – вага екстраверсії,
 w_A – вага приємності/альтруїзму,
 w_N – вага емоційної стабільності,

w_H – вага покірності,
 w_X – вага емоційності,
 w_{Narc} – вага нарцизму,
 $w_{Machiav}$ – вага макіавеллізму,
 w_{Psych} – вага психопатії.

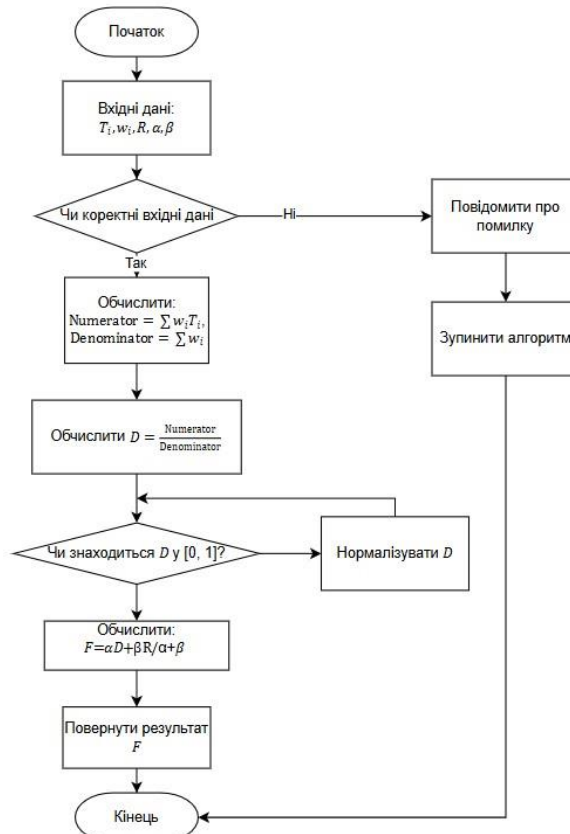
4

АЛГОРИТМ ПРИЙНЯТТЯ РІШЕННЯ НА ОСНОВІ ОСОБИСТИСНИХ РИС



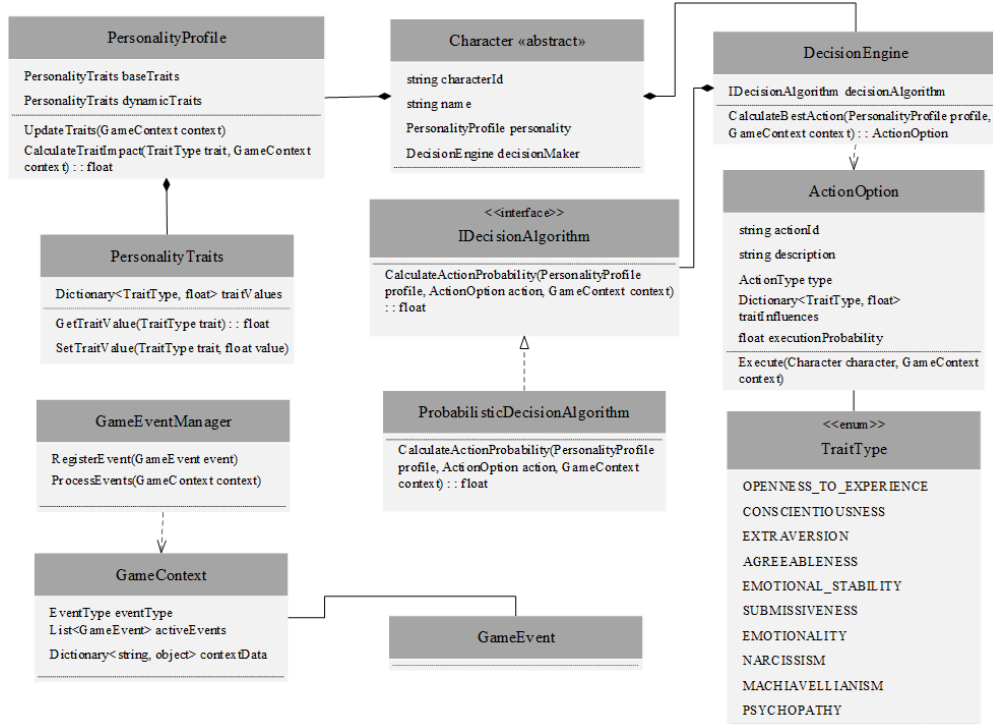
5

МОДИФІКОВАНИЙ МЕТОД ПРИЙНЯТТЯ РІШЕННЯ З ВРАХУВАННЯМ ОСОБИСТИСНИХ РИС



6

ДІАГРАМА КЛАСІВ



7

РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ ПРИЙНЯТТЯ РІШЕШІЯ З ВРАХУВАННЯМ ОСОБИСТІСНИХ РИС

P_1 : Етичний персонаж (Моральний герой)
 $T_O = 0.8, T_C = 0.9, T_E = 0.7, T_A = 0.85, T_N = 0.7,$
 $T_H = 0.6, T_X = 0.5, T_{Narc} = 0.2, T_{Machiav} = 0.1,$
 $T_{Psych} = 0.05.$

P_2 : Егоїстичний персонаж (Аморальний прагматик)
 $T_O = 0.4, T_C = 0.5, T_E = 0.6, T_A = 0.2, T_N = 0.6,$
 $T_H = 0.3, T_X = 0.7, T_{Narc} = 0.8, T_{Machiav} = 0.7,$
 $T_{Psych} = 0.6.$

D_1 : Віддати гаманець власнику
 $w_O = 0.15, w_C = 0.25, w_E = 0.1, w_A = 0.3,$
 $w_N = 0.1, w_H = 0.05, w_X = 0.05,$
 $w_{Narc} = 0.02, w_{Machiav} = 0.02, w_{Psych} = 0.01.$

D_2 : Забрати гроші та залишити гаманець
 $w_O = 0.05, w_C = 0.1, w_E = 0.15, w_A = 0.05,$
 $w_N = 0.1, w_H = 0.02, w_X = 0.1,$
 $w_{Narc} = 0.2, w_{Machiav} = 0.15, w_{Psych} = 0.08.$

$$D_1(P_1) = \frac{\sum_{i=1}^n w_i \cdot T_i}{\sum_{i=1}^n w_i} = 0.785$$

$$D_2(P_1) = \frac{\sum_{i=1}^n w_i \cdot T_i}{\sum_{i=1}^n w_i} = 0.535$$

$$D_1(P_2) = \frac{\sum_{i=1}^n w_i \cdot T_i}{\sum_{i=1}^n w_i} = 0.475$$

$$D_2(P_2) = \frac{\sum_{i=1}^n w_i \cdot T_i}{\sum_{i=1}^n w_i} = 0.665$$

8

ВИСНОВКИ

1. Проведено аналіз моделей Big Five, HEXACO і Dark Triad, що охоплюють базові риси особистості. Обґрунтовано їх поєднання для створення ефективного методу прийняття рішень в адаптивному середовищі.
2. Розроблено модель особистісних рис персонажів, засновану на психологічних моделях, та математичну модель для динамічного прийняття.
3. Створено алгоритм динамічного прийняття рішень, що враховує особистісні риси персонажів та змінюваність ігрового середовища.
4. Тестування підтвердило ефективність алгоритму, відповідність очікуванням та адаптивність поведінки персонажів у різних сценаріях.
5. Розроблений алгоритм перевершує існуючі рішення за адаптивністю та включає особистісні риси персонажів, що покращує ефективність прийняття рішень в адаптивному середовищі.

9

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Ярошевський О.В., Замрій І.В. Теоретичні основи формування особистісних рис ігрових персонажів. // Всеукраїнська науково-технічна конференція "Виклики та рішення в програмній інженерії" – Київ: ДУІКТ, 2024. – Подано до друку.
2. Ярошевський О.В., Замрій І.В. Динамічне прийняття рішень на основі особистісних рис ігрових персонажів. // Всеукраїнська науково-технічна конференція "Виклики та рішення в програмній інженерії" – Київ: ДУІКТ, 2024. – Подано до друку.

10

ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

```

using System;
using System.Collections.Generic;
using System.Linq;

namespace DecisionMakingModule
{
    // Enumeration for Trait Types
    public enum TraitType
    {
        HonestyHumility,
        Emotionality,
        Extraversion,
        Agreeableness,
        Conscientiousness,
        OpennessToExperience,
        Narcissism,
        Machiavellianism,
        Psychopathy
    }

    // Interface for providing character
    traits
    public interface
    ICharacterTraitsProvider
    {
        float GetTraitValue(TraitType
        traitType);
    }

    // Interface for Decision Making
    Module
    public interface
    IDecisionMakingModule
    {
        DecisionResult CalculateDecision(
        ICharacterTraitsProvider
        traitsProvider,
        IEnumerable<DecisionOption>
        options,
        IDictionary<TraitType, float>
        weights);
    }

    // Interface for Decision Strategy
    public interface IDecisionStrategy
    {
        float CalculateScore(
        ICharacterTraitsProvider
        traitsProvider,
        DecisionOption option,
        IDictionary<TraitType, float>
        weights);
    }

    // Class representing a decision option
    public class DecisionOption
    {
        public string Name { get; }
        public IDictionary<TraitType,
        float> TraitImpacts { get; }

        public DecisionOption(string
        name, IDictionary<TraitType, float>
        traitImpacts)
        {
            Name = name ?? throw new
            ArgumentNullException(nameof(name));
            TraitImpacts = traitImpacts ??
            throw new
            ArgumentNullException(nameof(traitImpacts));
        }
    }

    // Class for Weighted Sum Strategy
    public class WeightedSumStrategy :
    IDecisionStrategy
    {
        public float CalculateScore(
        ICharacterTraitsProvider
        traitsProvider,
        DecisionOption option,
        IDictionary<TraitType, float>
        weights)
        {
            if (traitsProvider == null) throw
            new
            ArgumentNullException(nameof(traitsProvider
            ));
            if (option == null) throw new
            ArgumentNullException(nameof(option));
            if (weights == null) throw new
            ArgumentNullException(nameof(weights));

            float numerator =
            option.TraitImpacts.Sum(impact =>
            traitsProvider.GetTraitValue(impact.Key) *
            impact.Value
            weights.GetValueOrDefault(impact.Key, 1f));
        }
    }
}

```

```

        float denominator =
weights.Sum(w => w.Value);

        return denominator == 0 ? 0 :
numerator / denominator;
    }
}

// Class representing the result of a
decision
public class DecisionResult
{
    public DecisionOption
ChosenOption { get; }
    public IDictionary<TraitType,
float> UpdatedTraits { get; }

    public
DecisionResult(DecisionOption chosenOption,
IDictionary<TraitType, float> updatedTraits)
    {
        ChosenOption = chosenOption
        ?? throw new
ArgumentNullException(nameof(chosenOption
));
        UpdatedTraits = updatedTraits
        ?? throw new
ArgumentNullException(nameof(updatedTraits
));
    }
}

// Implementation of Decision Making
Module with Strategy
public class DecisionMakingModule :
IDecisionMakingModule
{
    private readonly IDecisionStrategy
_strategy;
    public
DecisionMakingModule(IDecisionStrategy
strategy)
    {
        _strategy = strategy ?? throw
new
ArgumentNullException(nameof(strategy));
    }
    public DecisionResult
CalculateDecision(
        ICharacterTraitsProvider
traitsProvider,

```

```

        IEnumerable<DecisionOption>
options,
        IDictionary<TraitType, float>
weights)
    {
        if (traitsProvider == null) throw
new
ArgumentNullException(nameof(traitsProvider
));
        if (options == null) throw new
ArgumentNullException(nameof(options));
        if (weights == null) throw new
ArgumentNullException(nameof(weights));

        var scoredOptions = options
.Select(option => new
        {
            Option = option,
            Score =
_strategy.CalculateScore(traitsProvider, option,
weights)
        })
        .OrderByDescending(o =>
o.Score)
        .ToList();

        var bestOption =
scoredOptions.FirstOrDefault()?.Option;
        if (bestOption == null)
            throw new
InvalidOperationException("No valid options
provided.");

        var updatedTraits =
UpdateTraits(traitsProvider, bestOption);
        return new
DecisionResult(bestOption, updatedTraits);
    }
    private IDictionary<TraitType,
float> UpdateTraits(
        ICharacterTraitsProvider
traitsProvider,
        DecisionOption chosenOption)
    {
        var updatedTraits = new
Dictionary<TraitType, float>();

        foreach (var impact in
chosenOption.TraitImpacts)
        {
            var currentTraitValue =
traitsProvider.GetTraitValue(impact.Key);

```

```

        updatedTraits[impact.Key] =
Math.Clamp(currentTraitValue + impact.Value,
0f, 1f);
    }
    return updatedTraits;
}
}
// Combined Decision Making
Module for Function F
public class
CombinedDecisionMakingModule :
IDecisionMakingModule
{
    private readonly
IDecisionMakingModule _primaryModule;
    private readonly
IDecisionMakingModule _secondaryModule;
    private readonly float _alpha;
    private readonly float _beta;
    public
CombinedDecisionMakingModule(
IDecisionMakingModule
primaryModule,
IDecisionMakingModule
secondaryModule,
float alpha,
float beta)
    {
        _primaryModule =
primaryModule ?? throw new
ArgumentNullException(nameof(primaryModu
le));
        _secondaryModule =
secondaryModule ?? throw new
ArgumentNullException(nameof(secondaryMo
dule));
        _alpha = alpha;
        _beta = beta;
    }
    public DecisionResult
CalculateDecision(
ICharacterTraitsProvider
traitsProvider,
IEnumerable<DecisionOption>
options,
IDictionary<TraitType, float>
weights)
    {
        var primaryResult =
_primaryModule.CalculateDecision(traitsProvi
der, options, weights);

```

```

        var secondaryResult =
_secondaryModule.CalculateDecision(traitsPro
vider, options, weights);
        var combinedScores = options
.Select(option => new
{
    Option = option,
    CombinedScore = (_alpha *
primaryResult.UpdatedTraits.Sum(t =>
option.TraitImpacts.GetValueOrDefault(t.Key,
0f)) +
        _beta *
secondaryResult.UpdatedTraits.Sum(t =>
option.TraitImpacts.GetValueOrDefault(t.Key,
0f))) /
        (_alpha + _beta)
})
        .OrderByDescending(o =>
o.CombinedScore)
        .ToList();
        var bestOption =
combinedScores.FirstOrDefault()?.Option;
        if (bestOption == null)
            throw new
InvalidOperationException("No valid options
provided.");
        return new
DecisionResult(bestOption,
primaryResult.UpdatedTraits); // Traits based
on primary
    }
}
// Utility method for dictionary safe
access
public static class
DictionaryExtensions
{
    public static TValue
GetValueOrDefault<TKey, TValue>(
this IDictionary<TKey, TValue>
dictionary,
TKey key,
TValue defaultValue = default)
    {
        return
dictionary.TryGetValue(key, out var value) ?
value : defaultValue;
    }
}
}

```