

# АРХІТЕКТУРА АДАПТИВНОГО МОДУЛЯ АІ НА БАЗІ GOAL-ORIENTED ACTION PLANNING (GOAP) ДЛЯ ДИНАМІЧНИХ СЕРЕДОВИЩ

Вінницький національний технічний університет

## **Анотація**

*У роботі розглянуто архітектурні особливості та переваги використання алгоритму планування дій на основі цілей (GOAP) для створення інтелектуальних агентів у сучасних ігрових системах. Розглянуто механізми функціонування планувальника, структуру станів світу та опис дій. Проаналізовано методи адаптації поведінки NPC до динамічних змін ігрового середовища через маніпуляцію ваговими коефіцієнтами дій. Визначено ефективність застосування GOAP порівняно з деревами поведінки (Behavior Trees) для створення емерджентної поведінки персонажів.*

**Ключові слова:** Goal-Oriented Action Planning, GOAP, штучний інтелект, ігрові системи, динамічне середовище, Unity, C#.

## **Abstract:**

*The report explores the architectural features and advantages of using the Goal-Oriented Action Planning (GOAP) algorithm for creating intelligent agents in modern gaming systems. The mechanisms of the planner's operation, the structure of world states, and the description of actions are considered. Methods for adapting NPC behavior to dynamic changes in the game environment through the manipulation of action weight coefficients are analyzed. The effectiveness of using GOAP compared to Behavior Trees for creating emergent character behavior is determined.*

**Keywords:** Goal-Oriented Action Planning, GOAP, Artificial Intelligence, gaming systems, dynamic environment, Unity, C#.

## **Вступ**

Сучасна індустрія відеоігор вимагає від не ігрових персонажів (NPC) дедалі вищого рівня автономності та здатності адекватно реагувати на непередбачувані дії гравця. Класичні підходи, такі як скінченні автомати (FSM), ускладнюються в геометричній прогресії при додаванні нових станів, що робить їх важкими для підтримки в масштабних проектах. [1] У цьому контексті архітектура Goal-Oriented Action Planning (GOAP) стає оптимальним рішенням, оскільки вона переносить логіку прийняття рішень з жорстко прописаних переходів на динамічне планування. Це дозволяє агентам самостійно обирати найефективніший шлях до мети, виходячи з поточного стану ігрового світу. Проте впровадження GOAP у динамічних середовищах, де умови змінюються щомиті, потребує розробки спеціальних механізмів адаптивності для забезпечення продуктивності.

## **Актуальність дослідження**

Актуальність даного дослідження зумовлена передусім критичною потребою в забезпеченні емерджентної поведінки інтелектуальних агентів, оскільки сучасні гравці очікують від NPC здатності до прийняття нестандартних рішень у відповідь на непередбачувані дії користувача. [2] Технологія Goal-Oriented Action Planning (GOAP) вирішує це завдання, дозволяючи агентам автономно будувати логічні ланцюжки дій, що забезпечує високий рівень автономності та варіативності ігрового процесу.

Додатковим чинником виступає зростаюча динамічність сучасних ігрових середовищ, де руйнування об'єктів, зміна ландшафту та складу команд вимагають від архітектури АІ здатності до миттєвого перепланування стратегії без втрати продуктивності. Водночас розділення цілей та конкретних реалізацій дій у системі GOAP гарантує високу масштабованість розробки, дозволяючи інтегрувати нові механіки без ризику порушити існуючу логіку, що суттєво скорочує час виходу продукту на ринок.

Зрештою, питання створення інтелектуального, але продуктивного АІ є критичним для оптимізації обчислювальних ресурсів та підтримки стабільної частоти кадрів (FPS). Розробка адаптивного модуля, що використовує ефективні евристичні алгоритми пошуку в просторі станів, є

актуальним науково-практичним завданням, яке дозволяє збалансувати складність поведінки персонажів та навантаження на центральний процесор.

**Метою розробки** є проектування та реалізація архітектури адаптивного модуля штучного інтелекту на базі алгоритму Goal-Oriented Action Planning (GOAP), який здатен автономно функціонувати в умовах високої динамічності ігрового середовища. Розробка спрямована на створення системи, що поєднує гнучкість прийняття рішень із високою обчислювальною ефективністю.

### Основні відомості про технологію та її призначення

Архітектура GOAP базується на відокремленні цілей агента від конкретних дій, які він може виконувати. Це дозволяє системі бути надзвичайно гнучкою: програміст лише описує набір доступних дій (прекурсори та ефекти), а планувальник самостійно будує ланцюжок для досягнення цілі. [3]

Однією з головних переваг GOAP є підтримка емерджентної поведінки. Оскільки агент щоразу аналізує стан світу World State, він може знайти нестандартні шляхи вирішення задачі, які не були заздалегідь передбачені розробником. Наприклад, якщо у ворога закінчилися набої, він не просто переходить у стан "паніки", а шукає найближчу дію, результатом якої є наявність зброї, що може включати як пошук складу, так і перехід до ближнього бою.

Однак використання GOAP у динамічних середовищах має свої виклики. Основним недоліком є обчислювальна складність алгоритму пошуку шляху в просторі станів (зазвичай  $A^*$ ). Для вирішення цієї проблеми в адаптивних модулях застосовується система динамічної вартості дій Action Cost.

Вартість дії може змінюватися в реальному часі залежно від контексту: якщо персонаж поранений, вартість дій, пов'язаних з агресією, зростає, а вартість пошуку аптечки — падає. Це забезпечує природну адаптацію AI без необхідності створювати сотні додаткових правил.

Механізм роботи модуля GOAP включає три основні рівні:

- Датчики (Sensors): Збирають дані про оточення та оновлюють World State.
- Планувальник (Planner): Формує послідовність дій, використовуючи евристичні алгоритми.
- Виконавець (Executor): Послідовно запускає обрані дії та відстежує їх успішність.

Важливою частиною адаптивності є система перепланування Re-planning. У динамічних системах план може стати неактуальним ще до завершення першої дії (наприклад, ціль зникла або шлях заблоковано). Ефективний модуль AI повинен підтримувати переривання Interrupts, які змушують планувальника миттєво переглянути стратегію без суттєвої втрати FPS. [4] Стратегія AI модуля визначається за загальним коефіцієнтом ваги  $C_{total}$ , розрахунок якого показаний у формулі:

де:

$$C_{total} = C_{base} + \sum_{i=1}^n w_i \cdot v_i(S)$$

- $w_i$  - ваговий коефіцієнт пріоритету (наприклад, рівень страху або агресії).
- $C_{base}$  - базова складність дії.
- $v_i(S)$  - значення параметрів стану (здоров'я, кількість патронів тощо).

Приклад роботи алгоритму на прикладі дії Enemy Attack:

- Action Name: AttackEnemy
- Preconditions: { hasWeapon: true, enemyInSight: true, distance: < 5m }
- Effects: { enemyIsDead: true, ammoCount: -1 }
- Base Cost: \$10\$ (може змінюватись динамічно)

### Порівняння методологій

Вибір архітектури керування поведінкою є фундаментальним етапом проектування ігрової системи. Необхідність порівняння FSM, BT та GOAP зумовлена пошуком оптимального балансу між трьома критичними показниками: обчислювальною складністю, швидкістю розробки та рівнем автономності агентів. Оскільки сучасні ігри вимагають від NPC здатності адаптуватися до змін середовища без втручання програміста, аналіз цих технологій дозволяє визначити межі застосовності кожної з них залежно від поставлених геймдизайнерських завдань. Порівняння цих підходів наведено у таблиці 1.

Порівняно з деревами поведінки (BT), GOAP є більш "розумним" у виборі ресурсів. У той час як BT вимагає чіткої структури пріоритетів, GOAP дозволяє агенту бути більш ситуативним. Проте для простих завдань GOAP може бути надлишковим, тому в реальних проєктах часто використовують гібридний підхід: високорівневе планування (GOAP) для стратегічних рішень та низькорівневі дерева поведінки для безпосереднього керування анімаціями та пересуванням. [5]

Таблиця 1 – Порівняння підходів керування поведінкою в ігрових системах

Критерій	Скінченні автомати (FSM)	Дерева поведінки (BT)	GOAP
Маштабованість	Низька	Середня	Висока (модульність)
Гнучкість	Жорстко визначена	Напівдинамічна	Повністю динамічна
Обчислення	Мінімальні	Низькі	Середні (A*)
Передбачуваність	Повна	Висока	Емерджентна

Аналіз ігрових проєктів, таких як F.E.A.R., S.T.A.L.K.E.R. та Deus Ex, виявив системні недоліки класичного GOAP: «роботизованість» через статичну вартість дій, виникнення «паралічу аналізу» в динамічних сценах та відсутність координації між агентами при розподілі ресурсів. Для усунення цих вразливостей у роботі запропоновано перехід до адаптивної моделі, що поєднує гнучке стратегічне планування з механізмами миттєвої реакції.

Ключові вдосконалення системи:

- Динамічні ваги: коригування вартості дій залежно від стану агента (здоров'я, стрес) для реалістичної поведінки.
- Переривчасте планування: миттєва перевірка валідності плану, що усуває затримки реакції на зміни середовища.
- Гібридна модель (GOAP + BT): використання дерев поведінки для мікроконтролю анімацій та руху, що розвантажує основний планувальник.
- Кооперативне планування.

### Висновки

Проведений аналіз архітектури GOAP дозволяє стверджувати, що цей підхід є одним із найбільш раціональних рішень для розробки ігрових систем із динамічним оточенням. Завдяки розмежуванню цілей та механізмів їх реалізації, архітектура забезпечує високу маштабованість, дозволяючи інтегрувати нові типи поведінки без ризику дестабілізації загальної логіки. Ключовим аспектом при побудові адаптивного модуля AI є вдосконалення алгоритмів планування та використання системи динамічних ваг для дій. Це не лише наближає поведінку NPC до антропоморфної, але й гарантує оптимальний розподіл обчислювальних ресурсів, що є критично важливим для ігрових систем.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Duarte A., Santos R., Pereira J. Goal-Oriented Action Planning for Intelligent Agents in Modern Game Engines, DOI:10.1109/GAME. 2023. 10214, 2023.URL: [https://www.researchgate.net/publication/368291456\\_Goal-Oriented\\_Action\\_Planning\\_for\\_Intelligent\\_Agents\\_in\\_Modern\\_Game\\_Engines](https://www.researchgate.net/publication/368291456_Goal-Oriented_Action_Planning_for_Intelligent_Agents_in_Modern_Game_Engines)
2. Vaneck J., Galambos P. Adaptive Goal-Oriented Action Planning for Autonomous Systems in Non-Static Environments, DOI: 10.1016/j.procs.2022.01.123, 2022. URL: [https://www.researchgate.net/publication/358129410\\_Adaptive\\_Goal-Oriented\\_Action\\_Planning\\_for\\_Autonomous\\_Systems\\_in\\_Non-Static\\_Environments](https://www.researchgate.net/publication/358129410_Adaptive_Goal-Oriented_Action_Planning_for_Autonomous_Systems_in_Non-Static_Environments)
3. Sarkar A., Cooper S. An Analysis of Decision-Making Algorithms for Non-Player Characters in Dynamic Environments», 2021. URL: [https://www.researchgate.net/publication/354721834\\_An\\_Analysis\\_of\\_Decision-Making\\_Algorithms\\_for\\_Non-Player\\_Characters\\_in\\_Dynamic\\_Environments](https://www.researchgate.net/publication/354721834_An_Analysis_of_Decision-Making_Algorithms_for_Non-Player_Characters_in_Dynamic_Environments)
4. Wang L., Zhang Y. Efficient A\* Search Optimization for Real-Time Pathfinding and Planning in Video Games», DOI: 10.3390/app13158742, 2023. URL: [https://www.researchgate.net/publication/372718392\\_Efficient\\_A\\_Search\\_Optimization\\_for\\_Real-Time\\_Pathfinding\\_and\\_Planning\\_in\\_Video\\_Games](https://www.researchgate.net/publication/372718392_Efficient_A_Search_Optimization_for_Real-Time_Pathfinding_and_Planning_in_Video_Games)
5. Kozak M., Fedorov S. Hybrid Architectures for NPC Decision Making: Integrating Behavior Trees and GOAP, 2024. URL: [https://www.researchgate.net/publication/379124562\\_Hybrid\\_Architectures\\_for\\_NPC\\_Decision\\_Making\\_Integrating\\_Behavior\\_Trees\\_and\\_GOAP](https://www.researchgate.net/publication/379124562_Hybrid_Architectures_for_NPC_Decision_Making_Integrating_Behavior_Trees_and_GOAP)

**Сингаївський Владислав Ігорович** - студент групи ІПІ-25м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: [vladyslav.synhaiivskyi@gmail.com](mailto:vladyslav.synhaiivskyi@gmail.com)

**Ліщинська Людмила Броніславівна** - д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: [llb@vntu.edu.ua](mailto:llb@vntu.edu.ua)

**Synhaiivskyi Vladyslav Igorovych** - student of group IPI-25m, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: [vladyslav.synhaiivskyi@gmail.com](mailto:vladyslav.synhaiivskyi@gmail.com)

**Lishchynska Lyudmyla Bronislavivna** - Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: [llb@vntu.edu.ua](mailto:llb@vntu.edu.ua)