

ОПТИМІЗАЦІЯ ПРОЦЕСІВ ЖИТТЄВОГО ЦИКЛУ ІГРОВОГО АІ ШЛЯХОМ ВПРОВАДЖЕННЯ АДАПТИВНОЇ АРХІТЕКТУРИ GOAP

Вінницький національний технічний університет

Анотація

У роботі досліджується вплив архітектури Goal-Oriented Action Planning (GOAP) на ефективність життєвого циклу розробки інтелектуальних агентів. Розглянуто, як відокремлення цілей від реалізації дій спрощує етапи підтримки та масштабування програмного забезпечення. Проаналізовано механізми адаптації поведінки через динамічні ваги, що дозволяє скоротити витрати на ітеративне тестування та налагодження логіки NPC.

Ключові слова: життєвий цикл ПЗ, GOAP, адаптивна архітектура, інтелектуальні агенти, Unity, ігрові системи.

Abstract:

The paper investigates the impact of Goal-Oriented Action Planning (GOAP) architecture on the efficiency of the software development life cycle for intelligent agents. It considers how decoupling goals from action implementation simplifies the maintenance and scaling stages. Methods of behavior adaptation through dynamic weights are analyzed, allowing for reduced costs in iterative testing and logic debugging of NPCs.

Keywords: software life cycle, GOAP, adaptive architecture, intelligent agents, Unity, gaming systems.

Вступ

Еволюція сучасних ігрових систем вимагає перегляду традиційних моделей життєвого циклу розробки АІ. Статичні підходи, зокрема скінченні автомати (FSM), на етапах супроводу призводять до надмірної складності коду («спагеті-код»), що значно здорожує вартість підтримки проєкту. Використання архітектури GOAP дозволяє перенести акцент із жорсткого програмування переходів на декларативний опис дій, що спрощує процеси оновлення та модифікації логіки персонажів у динамічних середовищах. [1]

Актуальність дослідження

Актуальність теми зумовлена необхідністю оптимізації процесів супроводу складних програмних систем. Сучасний життєвий цикл ігрового ПЗ передбачає постійну адаптацію під непередбачувані дії користувачів та динамічні зміни оточення. Технологія GOAP забезпечує високу масштабованість, дозволяючи додавати нові модулі поведінки без ризику дестабілізації існуючої архітектури, що суттєво скорочує показник Time-to-Market та полегшує еволюційне супроводження системи. [2] Крім того, розробка адаптивних модулів з використанням евристичних алгоритмів дозволяє збалансувати навантаження на ресурси, що є критичним для забезпечення стабільності продукту на етапі експлуатації. [3]

Метою розробки є дослідження та розробка адаптивного модуля АІ на базі GOAP у контексті покращення процесів життєвого циклу ПЗ, що забезпечує гнучкість прийняття рішень та зниження витрат на етапі підтримки системи. [4]

Моделювання процесів та механізми адаптивності

У життєвому циклі розробки GOAP-систем ключовим є розділення цілей та інструментів їх досягнення. Це дає розробникам можливість ітеративно додавати нові дії (Actions), описуючи лише їх прекурсори та ефекти, не змінюючи при цьому ядро планувальника. [5]

Для забезпечення стабільної поведінки в умовах невизначеності застосовується система динамічної оцінки вартості дій (Action Cost). Це дозволяє модулю АІ самостійно адаптуватися до контексту (наприклад, стан здоров'я агента) без впровадження додаткових статичних правил, що спрощує верифікацію логіки на етапі тестування.

Структура модуля включає три рівні, що відповідають етапам обробки даних у реальному часі:

- **Sensors (Датчики):** Актуалізація стану світу (World State).

- **Planner (Планувальник):** Формування стратегії через евристичний пошук.
- **Executor (Виконавець):** Контроль успішності реалізації плану.

де:

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

- w_i - ваговий коефіцієнт пріоритету.
- C_{base} - базова вартість.
- $v_i(S)$ - параметри поточного стану.

Порівняльний аналіз 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.

При виборі моделі життєвого циклу AI важливо враховувати баланс між складністю та автономністю (Табл. 1).

Таблиця 1 – Порівняння архітектур AI за критеріями життєвого циклу

Критерій	Скінченні автомати (FSM)	Дерева поведінки (BT)	GOAP
Маштабованість	Низька (високі витрати на оновлення)	Середня	Висока (модульність)
Гнучкість розробки	Жорстко обмежена	Напівдинамічна	Повністю адаптивна
Складність супроводу	Дуже висока (ризик регресії)	Низькі	Низька (ізолювані дії)
Поведінка NPC	Передбачувана	Структурована	Емерджентна

Аналіз реальних проєктів (F.E.A.R., S.T.A.L.K.E.R.) вказав на необхідність усунення «паралічу аналізу» та покращення координації. Для оптимізації процесів життєвого циклу в даній роботі запропоновано наступні вдосконалення:

Ключові вдосконалення системи:

- **Динамічні ваги:** автоматизація вибору стратегії залежно від контексту.
- **Переривчасте планування:** механізм швидкої валідації плану для миттєвої реакції на зміни.
- **Гібридна модель (GOAP + BT):** розподіл рівнів абстракції для зниження навантаження на планувальник.

Висновки

Впровадження адаптивної архітектури GOAP є раціональним рішенням для оптимізації життєвого циклу складних ігрових систем. Розмежування логіки та цілей забезпечує високу маштабованість та спрощує етап супроводу ПЗ, дозволяючи інтегрувати новий функціонал без порушення цілісності системи. Використання динамічних ваг не тільки наближає поведінку NPC до антропоморфної, але й дозволяє ефективно керувати обчислювальними ресурсами на всіх етапах експлуатації продукту.

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

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

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

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

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

Сингаївський Владислав Ігорович - студент групи ІПІ-25м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: vladyslav.synhaivskyi@gmail.com

Науковий керівник: Ракитянська Ганна Борисівна - к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: rakit@vntu.edu.ua

Synhaivskyi Vladyslav Igorovych - student of group ІPI-25m, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: vladyslav.synhaivskyi@gmail.com

Academic supervisor: Rakytianska Hanna Borysivna - in Engineering, Associate Professor of the Department of, Vinnytsia National Technical University, Vinnytsia, e-mail: rakit@vntu.edu.ua