УДК 004.93:004.032.6

РОЗРОБКА АДАПТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ NPC ДЛЯ ТАКТИЧНИХ СИМУЛЯТОРІВ З ВИКОРИСТАННЯМ МЕТОДУ GOAP

Веліканов О.Р.
e-mail: velikanov.exe@gmail.com
Науковий керівник – д-р техн. наук, доц., с.н.с. Шафроненко А.Ю.
Харківський національний університет радіоелектроніки, каф. ІНФ
м. Харків, Україна

This work presents the development of an adaptive artificial intelligence system for non-player characters (NPC) in tactical simulators using the Goal-Oriented Action Planning (GOAP) method. Unlike traditional approaches based on finite-state machines (FSMs) or behavior trees, the proposed system enables agents to dynamically build action sequences in response to changing environmental conditions. The architecture is implemented in the Unity game engine using C# and includes a perception module, a GOAP planner based on A* search in the state-action space, and a library of atomic actions. The system demonstrates emergent tactical behavior, including flanking maneuvers, suppression fire, cover usage, and coordinated group actions in close-quarters battle (CQB) scenarios without hard-coded scripting.

Розробка систем штучного інтелекту для неігрових персонажів (NPC) є актуальною задачею сучасної ігрової індустрії та суміжних галузей, наприклад: навчальних тренажерів або тактичних симуляторів. Традиційні підходи, засновані на кінцевих автоматах (FSM) або деревах поведінки (Behavior Trees), гарантують передбачувану та легку у реалізації логіку, але мають суттєві недоліки у вигляді жорстко заданих структур переходів, які не дозволяють агенту гнучко адаптуватися до непередбачуваних та нестандартних ситуацій. Ціль даної роботи – розробити інтелектуальну систему поведінки NPC на основі архітектури GOAP (Goal-Oriented Action Planning), яка сприяє автономному плануванню дій в умовах тактичного ближнього бою.

Метод GOAP є технологією планування на основі станів і дій. На відміну від FSM, де переходи між станами задаються явно, GOAP-агент самостійно будує оптимальну послідовність атомарних дій, що веде від поточного стану світу до визначеної цілі агента. Кожна дія характеризується передумовами (preconditions), за яких вона може бути виконана та ефектами (effects – змінами ігрового світу після виконання окремих дій). Планувальник формує граф станів, де вузли відповідають станам системи, а ребра – можливим діям із відповідною вартістю. Для пошуку оптимального плану застосовується алгоритм A* у просторі станів.

Архітектура складається з трьох підсистем: сенсорної системи, планувальника і бібліотеки атомарних дій. Сенсорна система (Perception Module)

забезпечує сприйняття агентом стану оточення, виявлення противників, аналіз геометрії рівня, ідентифікацію потенційних укриттів та оцінку загроз. Це формує поточний стан світового простору (World State), який насамперед є вхідними даними для планувальника.

Планувальник (GOAP Planner) – центральний обчислювальний модуль системи. Він отримує поточний стан світу та активну ціль агента і будує оптимальну послідовність дій. Планування відбувається не у фізичному просторі рівня, а у просторі станів, де кожен вузол відображає набір булевих або чисельних характеристик ситуації (наприклад: isEnemyVisible, hasAmmo, isInCover). Завдяки цьому знижується обчислювальна складність порівняно з методами планування у фізичному просторі.

Бібліотека атомарних дій може включати такі примітиви: MoveTo – переміщення до заданої точки, TakeCover – зайняття укриття, Attack – відкриття вогню по противнику, Reload – перезарядка зброї, Flank – фланговий маневр, Retreat – тактичний відступ. Кожна дія реалізується за допомогою інтерфейса з методами CheckPreconditions(), ApplyEffects() та GetCost(). Завдяки цій архітектурі розширення системи новими діями не потребує модифікації планувальника, це робить можливим масштабування системи.

Система цілей агента включає дві основні цілі: виживання (Survive) та нейтралізація противника (EliminateEnemy). Пріоритет між цілями визначається динамічно на основі поточного стану – рівня здоров'я, наявності боєприпасів, дистанції до противника та інших ігрових умов. Таким чином, забезпечується реалістична реакція агента на різноманітні ситуації у симуляції.

Практичним результатом роботи є діючий прототип тактичного шутера з елементами CQB (Close Quarters Battle). Ключовою перевагою обраної архітектури є можливість виникнення емерджентної поведінки агентів. Замість жорсткого сценарного програмування складних тактичних маневрів, система дозволяє агентам динамічно формувати такі сценарії, як наслідок взаємодії індивідуальних планів. Завдяки цьому симуляції мають високу варіативність ігрового процесу. Хоча порівняно з реалізацією на основі FSM, GOAP-агенти демонструють значно вищу адаптивність в нестандартних ситуаціях, проте вимагають більших обчислювальних ресурсів через необхідність побудови плану в реальному часі.

Список використаних джерел:

1. Orkin J. Three States and a Plan: The A.I. of F.E.A.R. // Game Developers Conference Proceedings. San Francisco, 2006. URL: https://www.gamedevs.org/uploads/three-states-plan-ai-of-fear.pdf.

2. Game AI Planning Analytics: The Case of Three First Person Shooters. Eric Jacopin, 2014. URL: https://aaai.org/papers/00119-12728-game-ai-planning-analytics-the-case-of-three-first-person-shooters.