

УДК 004.85:004.946

ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ У СУЧАСНИХ КОМП'ЮТЕРНИХ ІГРАХ

Литвин М.А.

e-mail: maksym.lytvyn@nure.ua

Науковий керівник – канд. техн. наук, ст. викл. Кобилін І.О.

Харківський національний університет радіоелектроніки, каф. ІНФ
м. Харків, Україна

This work investigates the application deep reinforcement learning (DRL) for creating adaptive behaviors in modern video game development. The paper analyzes the evolution from rigid scripted AI to autonomous agents capable of physically accurate locomotion. Using the development experience of ARC Raiders created by Embark Studio as a primary case study, an evaluation is performed on how agents develop emergent movement strategies based on mass, inertia, and sensory observations. Additionally, skill-balancing mechanisms, such as the Skilled Experience Catalogue, are briefly explored to demonstrate the adjustment of NPC proficiency in real-time. The study concludes that machine learning-driven approaches significantly optimize development pipelines by reducing the reliance on manual animation authoring, while also enhancing gameplay realism and fostering unpredictable interactive experiences.

Трансформація відеоігор у складні цифрові екосистеми зумовила пошук нових методів управління віртуальними об'єктами. Сучасна індустрія геймдеву поступово відмовляється від детермінованих алгоритмів на користь динамічних систем на основі глибинного навчання з підкріпленням (Deep Reinforcement Learning, DRL). Використання DRL дозволяє створювати персонажів, чия поведінка формується безпосередньо в процесі взаємодії з фізичним середовищем [2].

Центральним поняттям у цій парадигмі є програмний «агент» – автономна сутність, здатна сприймати стан оточення та виконувати дії для досягнення поставленої мети. На відміну від традиційного ШІ, агент у системі машинного навчання діє подібно до живого організму. Хоча математичним фундаментом тут є Марковський процес прийняття рішень (MDP), акцент зміщується на реалізацію фізично коректної локомоції, де агент самостійно визначає оптимальні патерни руху, аналізуючи наслідки своїх дій через систему зворотного зв'язку [4], [5].

Навчання агента базується на максимізації сумарної очікуваної винагороди r_t . У сучасних проектах ця величина часто є комбінованою для одночасного досягнення кількох цілей:

$$r_t = \omega_I r_t^I + \omega_G r_t^G$$

У цій моделі складова r_t^I відповідає за імітаційне навчання – наскільки рух агента подібний до еталонних даних (наприклад, професійно записаної анімації руху людини). Складова r_t^G характеризує ефективність виконання конкретного ігрового завдання (переміщення до цілі, утримання рівноваги). Вагові коефіцієнти ω дозволяють розробникам тонко балансувати між візуальною достовірністю та функціональністю. Політика дій агента $\pi(a|s)$ реалізується через глибоку нейронну мережу, яка зазвичай описується ймовірнісним розподілом:

$$\pi(a|s) = N(\mu(s), \Sigma)$$

де $\mu(s)$ визначає середнє значення дії, розраховане мережею, а Σ дозволяє системі зберігати певний рівень варіативності. Це гарантує, що персонаж не буде діяти завжди однаково, що критично важливо для запобігання передбачуваності NPC у взаємодії з гравцем [3].

Окрім фізичної локомоції (процесу руху), машинне навчання відкриває широкі можливості для адаптивного балансування складності (Dynamic Difficulty Adjustment). Одним із найбільш інноваційних механізмів у цій сфері є використання системи Skilled Experience Catalogue (SEC). Цей підхід базується на створенні «каталогу досвіду», що складається із серії збережених станів навчання нейронної мережі агента на різних етапах його підготовки. Якщо NPC виконує завдання занадто ефективно, система може автоматично змінити його політику на більш ранню, «менш досвідчену» версію. Це дозволяє підтримувати гравця у стані «потoku» (flow state), забезпечуючи ідеальний баланс між складністю гри та реальними навичками користувача [4].

Для оцінки перспективності поточного стану агента використовується функція цінності $V(s)$, яка прогнозує очікувану суму майбутніх винагород з урахуванням коефіцієнта дисконтування γ :

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s \right]$$

Саме цей механізм лежить в основі ефекту «інстинкту самозбереження», який спостерігається в проекті ARC Raiders від Embark Studios. Агент аналізує ризик падіння як стан з низькою майбутньою цінністю і тому динамічно змінює положення ніг або переносить вагу на вцілілі частини конструкції після отримання пошкоджень [1].

Особливого значення машинне навчання набуває у сфері генерації фізично правдоподібної анімації на складному рельєфі. У проекті ARC Raiders розробники повністю відмовилися від стандартних анімаційних циклів, надавши навченим агентам повний контроль над власною локомоцією.

Агенти сприймають віртуальний світ через систему «observations», використовуючи тисячі невидимих променів (Line Tracing) для сканування рельєфу навколо себе. Під час тренування параметри нейронної мережі θ постійно оновлюються за допомогою градієнта політики:

$$\nabla_{\theta} J(\theta) \approx \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) R_t$$

Це математично описує процес закріплення вдалих патернів руху. В результаті виникає самоорганізована адаптивна поведінка: агент самостійно вчиться балансувати після вибухів або втрати елементів конструкції, що створює враження справді живої, інтелектуальної машини [1, 2].

Підсумовуючи, можна стверджувати, що машинне навчання докорінно змінює архітектуру та філософію проектування відеоігор. Перехід від жорстких скриптів до автономних інтелектуальних агентів, здатних до навчання, фізичної адаптації та динамічного балансування навичок, дозволяє створювати живі інтерактивні світи з безпрецедентним рівнем занурення гравця [2].

Список використаних джерел:

1. ARC Raiders. The evolution of ARC raiders EP3 - building ARC machines, 2025. *YouTube*. URL: <https://www.youtube.com/watch?v=DRlhpc7ImA> (date of access: 11.03.2026).
2. A survey of deep reinforcement learning in video games / K. Shao et al. *arXiv.org*. URL: <https://arxiv.org/abs/1912.10944> (date of access: 11.03.2026).
3. DeepMimic: example-guided deep reinforcement learning of physics-based character skills / X. B. Peng et al. *arXiv.org*. URL: <https://arxiv.org/abs/1804.02717> (date of access: 11.03.2026).
4. Glavin F. G., Madden M. G. Skilled experience catalogue: a skill-balancing mechanism for non-player characters using reinforcement learning. *arXiv.org*. URL: <https://arxiv.org/abs/1806.07637> (date of access: 11.03.2026).
5. Sim-to-Real: learning agile locomotion for quadruped robots / V. Vanhoucke et al. *arXiv.org*. URL: <https://arxiv.org/abs/1804.10332> (date of access: 11.03.2026).