

УДК 004.942

DOI: 10.20998/2411-0558.2021.02.05

В. В. ВІНОГРАДОВ, магістр, СХУ ім. В. Даля, Сєверодонецьк,
Л. О. ШУМОВА, канд. техн. наук, доц., СХУ ім. В. Даля,
Сєверодонецьк,
Т. О. БІЛОБОРОДОВА, канд. техн. наук, доц., ІПМЕ, Київ

УДОСКОНАЛЕННЯ МОДЕЛІ ПОВЕДІНКИ ІНТЕЛЕКТУАЛЬНОГО АГЕНТУ НА ОСНОВІ ГЕНЕТИЧНОГО АЛГОРИТМУ

Представлено вирішення завдання удосконалення моделі поведінки неігрового персонажу, що виступає в якості інтелектуального агента, за рахунок оптимізації вхідних параметрів на основі генетичного алгоритму. Запропонований підхід включає розробку моделі неігрового персонажу: скелету, твердих тіл; реалізацію динамічної моделі персонажу, заснованої на алгоритмі Фізерстоуна, та моделювання поведінки персонажу на основі генетичного алгоритму. Запропоновано удосконалення моделі поведінки моделі неігрового персонажу із використанням генетичного алгоритму, що моделює фізичні властивості персонажу з урахуванням його дій. Етапи роботи генетичного алгоритму включають: створення початкової популяції, оцінка пристосовуваності, відбір, схрещування та мутації. За результатами проведених експериментів визначено вхідні параметри моделі поведінки моделі неігрового персонажу, що максимізують сукупну оцінку пристосовності, яка виступає в якості оцінки винагороди. Визначені параметри можуть бути використані як початкові вхідні значення при проведенні подальших експериментів. Іл.: 4. Табл.: 2. Бібліогр.: 16 назв.

Ключові слова: неігровий персонаж; інтелектуальний агент; моделювання; генетичний алгоритм.

Вступ. Ключовим елементом реалізму у відеоіграх є поведінка персонажів, що не керуються користувачем. Програмне забезпечення найчастіше ґрунтується на простій ілюзії інтелекту ігрового персонажу, що створюється умілим використанням методів ігрового дизайну та використовує прості алгоритми управління, реалістичну графіку, переконливу анімацію персонажів, голоси, запозичені у відомих кіноакторів. Часто персонажі комп'ютерних ігор керуються глобальним алгоритмом, який має доступ до всіх змінних програм.

У сучасній ігровій індустрії дедалі більшої популярності набуває використання методів штучного інтелекту для реалізації ігрового персонажу. Методи штучного інтелекту, що використовуються для реалізації подібних ігрових персонажів, дозволяють моделювати послідовність дій персонажів, необхідних для досягнення поставленої мети. Така послідовність залежить не тільки від мети, а й від фактичного стану персонажу та навколишнього середовища. Це означає, що коли два

персонажі з різними станами переслідують одну й ту саму мету, вони можуть генерувати дві зовсім різні послідовності дій. Такий підхід робить ігровий процес динамічнішим і реалістичнішим.

Неігровий персонаж або негральний персонаж (англ. Non-Player Character, NPC) – це будь-який персонаж у грі, який не контролюється гравцем [1] і, зазвичай, має зумовлений набір поведінки, яка потенційно може вплинути на ігровий процес. Використання інтелектуальних NPC у комп'ютерних іграх потенційно дозволить зробити ігри більш складними та захоплюючими. Таким чином, у сучасній ігровій індустрії моделювання поведінки NPC є важливим компонентом комп'ютерних ігор [2, 3]. У сучасних шутерах від першої особи та інших іграх-шутерах NPC запрограмовані імітувати людей [4] для підвищення реалізму ігрового процесу.

Ранні реалізації NPC не демонстрували якоїсь складної поведінки, крім нападу чи пересування. Їх реалізації являли собою статичні моделі із задалегідь створеними дорожніми точками для кожної карти, за якою вони прямували. Пізніші реалізації об'єднали динамічний підхід, який допускає, так званий, стан полювання, в якому NPC активно шукає сліди ворога поряд із пошуком шляху та вивченням карти в режимі реального часу. Складна поведінка NPC, що ґрунтується на політиці прийняття рішень, зазвичай реалізується за допомогою евристичних методів штучного інтелекту [5], якому, проте, не завжди вдається досить реалістично симулювати поведінку інтелектуального об'єкту. Таким чином, завдання удосконалення поведінки NPC як інтелектуального агенту ігрового процесу залишається актуальним.

Аналіз останніх досліджень і публікацій. Недоліком ігрових персонажів на основі штучного інтелекту є те, що вони використовують методи, які є чітко запрограмованими, і, отже, є статичними і вимагають ручного налаштування параметрів [6]. До цих методів відносять кінцеві автомати, експертні системи та системи, засновані на правилах. Спроби покращити ситуацію із застосуванням різних методів обчислювального інтелекту, такі як еволюційні алгоритми, почали застосовуватися відносно недавно [7], і загалом використовують два підходи: удосконалення існуючих методів штучного інтелекту, включених в ядро гри, або запровадження альтернативних способів управління NPC. Наприклад, у [8] використовуються обидва підходи. Генетичний алгоритм використовується для удосконалення параметрів нечіткого кінцевого автомату, включеного в ядро гри, що керує поведінкою персонажу. Інші підходи, створені задля реалізації штучного інтелекту ігрових персонажів, використовують нейронні мережі. Наприклад, у роботі [9] представлена модель інтелектуального ігрового агенту на основі

технології глибокого навчання з підкріпленням, що демонструє симуляцію людського інтелекту, перемагаючи гравців-людей в тестових змаганнях. В іншому дослідженні [10] представлений гібридний агент, реалізований з використанням технологій імітації та навчання з підкріпленням на основі нейронної мережі.

Підходи на основі генетичних алгоритмів та навчання з підкріпленням мають свої переваги та недоліки. Навчання з підкріпленням показує хороші результати при вирішенні проблем, що вимагають послідовного прийняття рішень, проте вимагає великої кількості даних і, також, використання навчання з підкріпленням може призвести до перевантаження станів, що може знизити результати. Враховуючи недоліки навчання з підкріпленням, використання генетичного алгоритму для моделювання поведінки NPC є доцільною парадигмою для інтелектуального агента NPC з метою автономного засвоєння своєї політики дій за допомогою взаємодії зі своїм середовищем у динамічному процесі.

Постановка завдання. Інтелектуальний NPC – це, по суті, інтелектуальний агент, що прагне дістати певної моделі поведінки через взаємодію з навколишнім середовищем з урахуванням своїх цілей. Розглянемо наступні визначення, що використовуються для постановки задачі.

Визначення 1. Простір станів середовища: простір станів S агента – це набір станів $\{s_1, s_2, \dots, s_n\}$, де кожен стан s_i є моментальним знімком середовища, в якому знаходиться агент. Стан являє собою поточну позу агента, включаючи положення та швидкості відповідних суглобів, а також стан цільового об'єкту або середовища, якщо вони є (наприклад, положення, швидкість, геометрична інформація тощо).

Визначення 2. Простір дій: простір дій C агента – це набір дій $\{c_1, c_2, \dots, c_m\}$, де кожна дія отримує унікальну винагороду довкілля.

Визначення 3. Винагорода: винагорода R – це оціночний зворотній зв'язок, отриманий агентом зі свого оточення.

Визначення 4. Модель поведінки: модель поведінки B агента – це внутрішня модель або відображення функцій простору станів S в простір дій C агента.

Формально, модель поведінки визначається наступним чином (1):

$$B: S \rightarrow C, \quad (1)$$

де кожен стан $s_i \in S$ відображається у дію $c_i \in C$.

Модель поведінки диктує, як NPC реагує на ситуації у своєму середовищі. Одним із підходів до отримання моделі поведінки є вивчення

функції поведінки B безпосередньо із заданого набору пар вибірок (s_i, c_i) , де c_i вказує дію, яка має бути виконана у стані s_i .

Поведінка агента у віртуальному середовищі визначається як функція стохастичної поведінки $b(c|s)$, що пов'язує кожен можливий стан s , який належить безперервному простору станів S , з розподілом ймовірностей для вибірки дії c безперервної дії простору C . Після спостереження поточного стану s_t на кожному часовому кроці t агент визначає відповідну дію c_t на основі стохастичної поведінки $b(a|s)$ шляхом вибірки із розподілу ймовірностей. Для створення реального руху NPC агента після вибору дії c_t слідує регулювання бажаних параметрів NPC агента. Змодельоване віртуальне середовище призначає винагорода r_t агенту, оцінюючи продуктивність агента по відношенню до заданої мети на основі попередньо визначеної функції винагорода $R(s, a)$. Крім того, винагорода переводить NPC агента в новий стан s_{t+1} , обчислюючи рух агента протягом інтервалу часу між t и $(t + 1)$ на основі динаміки параметрів тіла агента.

Ця парадигма навчання на основі спостережень відома як імітаційне навчання. Імітаційне навчання – це процес навчання, в якому агент виводить свою функцію моделі поведінки B з набору спостережень, кожне з яких містить вхідний стан $s_i \in S$ та дію $a_i \in A$.

Основне припущення імітаційного навчання полягає в тому, що кожна поведінка, що спостерігається, відповідає конкретному середовищу, в якому знаходиться агент.

Шляхом імітаційного навчання знання, що формують функцію моделі поведінки, мають форму серії пар стан-дія $f_i(s_i, a_i)$ з логікою, згідно якої, коли відбувається певний стан s_i , виконується відповідна дія a_i .

Очікувана сукупна винагорода $E(b)$ для певної поведінки агента у змодельованому віртуальному середовищі може бути розрахована наступним чином (2).

$$E(b) = \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^{T_i} R_b(s_t, c_t \sim b(c|s_t)) \right), \quad (2)$$

де s_i – стан, представлений структурою твердих тіл і суглобів агента, c_i – дія агента представлена значеннями функцій сили (імпульс, швидкість, прискорення) за напрямом вектора руху, R_b – набір дійсних чисел отриманих винагород, а N та T_i – кількість епізодів та кількість часових кроків для i -го епізоду, відповідно.

Таким чином, завдання удосконалення моделі поведінки можна представити як знаходження оптимальної поведінки, тобто, оптимальної

пари стан-дія $b'(s',c')$, яка максимізує очікувану сукупну винагороду $E(b)$ у змодельованому віртуальному середовищі

$$b' = \operatorname{argmax}_{b \in B} E(b),$$

де B – модель поведінки, представлена набором значень параметрів моделі, що характеризує можливу поведінку агенту.

Мета статті. Метою дослідження є удосконалення моделі поведінки NPC за рахунок оптимізації вхідних параметрів на основі генетичного алгоритму.

Запропонований підхід включає реалізацію наступних складових елементів: розробку моделі NPC, реалізацію моделі поведінки NPC з використанням генетичного алгоритму.

Модель NPC. Модель NPC складається з ієрархії твердих тіл і суглобів, що приводяться у дію з використанням динамічної моделі [11].

Скелет представлений ієрархічною структурою абсолютно твердих тіл, (англ. rigid body) кожне з яких відповідає кістці в системі скелетної анімації графічного движку, пов'язані разом за допомогою системи обмежень, що визначаються суглобами. Ієрархія визначає всі об'єкти, що йдуть за батьківським об'єктом, дочірніми об'єктами попереднього, тобто такими, що слідує рухам та поворотам з урахуванням ієрархії об'єктів у скелеті.

На основі моделі стану NPC формується простір дій NPC шляхом безперервного застосування функцій сили за напрямом вектора руху. Простір дій характеризують такі функції дій (англ. force): імпульс, швидкість, прискорення.

Тверде тіло представлене набором вершин, розділених фіксованими відстанями [12]. Таким чином, тверде тіло займає об'єм у просторі і як таке може орієнтуватися навколо точки в просторі. Наслідком наявності просторової протяжності є те, що тіло має матеріальну точку, яка збігається з геометричним центром тіла. Лінійна динаміка твердого тіла тепер зводиться до динаміки матеріальної точки з масою m і дозволяє розділити лінійну динаміку і динаміку обертання, при цьому тіло зазнає зміни лінійного імпульсу тільки тоді, коли сила діє через матеріальну точку.

Добуток передбачуваної постійної маси m твердого тіла та прискорення a [13] описує чисту силу F , яку відчуває тверде тіло $F = ma$.

Розташування твердого тіла в 3D просторі описується за допомогою вектору положення \vec{p} (x, y, z) та кутів θ, φ, ψ . Швидкість твердого тіла у

просторі описується за допомогою вектору швидкості v та вектору кутової швидкості ω .

Тверде тіло може бути описане простіром станів, що описує зміни тіла у власній локальній системі координат, і простором дій, що описує зміни тіла у світовій глобальній системі координат.

Суглоби (з'єднання твердих тіл) створюються в системі скелетної анімації (спосіб анімації тривимірних моделей у мультиплікації та комп'ютерних іграх) ігрового двигуна. Дія суглобів провадиться з використанням алгоритму Фізерстоуна [14, 15], що заснований на використанні скороченого координатного представлення. Алгоритм Фізерстоуна реалізує розрахунок рухів суглобів розглядаючи тверді тіла та суглоби починаючи з батьківського суглоба та додаючи дочірні суглоби один за одним.

Динаміка моделі NPC. Реалізація алгоритму Фізерстоуна для трьох твердих тіл представлена в такий спосіб. Враховуючи поточний суглоб q , поточну похідну швидкості суглоба за часом \dot{q} , зовнішні сили F^E крутного моменту в суглобі G , може бути обчислено прискорення суглоба за часом, представлене як друга похідна

$$\ddot{q} = f(q, \dot{q}, F^E, G). \quad (4)$$

Просторове представлення моделі NPC поєднує лінійні та кутові величини. Два звичайних 3-мірних вектори замінюються одним 6-мірним просторовим вектором, описуючи швидкість NPC v і прискорення a , використовуючи кутову швидкість тіла і її похідну за часом, та похідну переміщення тіла \dot{x} , що визначає лінійну швидкість тіла, та похідну похідної переміщення тіла за часом як це представлено в (5)

$$v = \begin{bmatrix} \omega \\ \dot{x} \end{bmatrix}, \quad a = \begin{bmatrix} \dot{\omega} \\ \ddot{x} \end{bmatrix}. \quad (5)$$

Швидкість суглоба v_i^J між твердим тілом i та його дочірнім твердим тілом $\lambda(i)$ визначається на основі швидкості v_i наступним чином

$$v_i^J = v_i - v_{\lambda(i)}. \quad (6)$$

Спільна швидкість також може бути описана наступним чином (7)

$$v_i^J = h_i \dot{q}_i, \quad (7)$$

де h_i – матриця розміром $6 \times d_i$, \dot{q} – вектор розміром d_i на 1, а d_i – ступінь свободи суглоба i .

Швидкість i -го твердого тіла визначається як

$$v_i = v_{\lambda(i)} + h_i \dot{q}_i. \quad (8)$$

Просторове прискорення кожного твердого тіла a_i на основі прискорення його дочірнього твердого тіла $a_{\lambda(i)}$ визначається наступним чином

$$a_i = a_{\lambda(i)} + \dot{h}_i \dot{q}_i + h_i \ddot{q}_i, \quad (9)$$

де \dot{h}_i – похідна за часом матриці h_i , \dot{q} – вектор розміром d_i на 1, d_i – ступінь свободи суглоба i , \ddot{q} – похідна за часом вектору \dot{q} .

Таким чином, використовуючи рівняння Ньютона-Ейлера, дії i -го твердого тіла представлено в такий спосіб

$$f_i + f_i^E = I_i a_i + v_i \times I_i v_i, \quad (10)$$

де f_i – чиста сила, застосована до твердого тіла i через суглоби, а f_i^E – сума всіх інших сил, які діють на тверде тіло i , I_i – інерція твердого тіла i .

Чиста сила f_i , направлена на тверде тіло i через суглоби, розраховується з використанням (10).

$$f_i = f_i^J - \sum_{j \in \mu(i)} f_i^J, \quad (11)$$

де f_i^J – сила, що передається по твердому тілу $\lambda(i)$, що розраховується як це представлено (11), а $\mu(i)$ – множина дочірніх елементів твердого тіла i

$$f_i^J = I_i a_i + v_i \times I_i v_i - f_i^E - \sum_{j \in \mu(i)} f_i^J. \quad (12)$$

Реалізація алгоритму Фізерстоуна в моделі NPC забезпечує створення реалістичних та плавних моделей обертання.

Простір дій NPC. Простір дій NPC визначає зміни твердого тіла у світовій глобальній системі координат. Простір дій характеризують такі функції дій (англ. force): імпульс, швидкість, прискорення [13]. Дії розраховуються через зміни положення тіла у просторі середовища. Положення тіла x може бути визначено як функція часу $x(t)$, що дозволяє

визначити швидкість \mathbf{v} та прискорення \mathbf{a} тіла у глобальній системі координат наступним чином

$$\mathbf{v}(t) = \frac{\dot{\mathbf{x}}(t)}{t}, \quad (13)$$

$$\mathbf{a}(t) = \frac{\dot{\mathbf{v}}(t)}{t} = \frac{\ddot{\mathbf{x}}(t)}{t^2}. \quad (14)$$

Таким чином, сила \mathbf{F} , що діє на тіло, викликаючи це прискорення, дорівнює

$$\mathbf{F} = m\mathbf{a} = m \frac{\dot{\mathbf{v}}(t)}{t}. \quad (15)$$

Сили, що діють на тверде тіло за принципом Д'Аламбера [13], визначають наступним виразом

$$\mathbf{F}_{net} = \sum_i \mathbf{F}_i. \quad (16)$$

Застосування набору сил до твердого тіла дозволяє оновити положення, швидкість та прискорення тіла. Час дії сил визначається імпульсом \mathbf{I} , що розраховується наступним чином

$$\mathbf{I} = \mathbf{F}\Delta t = m\Delta\mathbf{v}. \quad (17)$$

Це рівняння повністю описує рух твердого тіла під дією кількох сил і дозволяє проводити обчислення зі збільшенням часу.

Модель поведінки NPC. Модель поведінки NPC є відображенням функцій моделі NPC у простір його дій. Запропоновано формування моделі поведінки із використанням генетичного алгоритму.

Для реалізації дії визначається сила на вектор положення $\vec{p}(x, y, z)$, тобто сила застосовується до кожного твердого тіла у всіх просторових координатах, внаслідок чого отримуємо вектор значень сил \vec{f} . Для застосування нової сили до твердого тіла для кожної координати вектора x, y, z задається випадкова величина сили в межах $[-100, 100]$. Визначення сили для кожного твердого тіла об'єкта NPC дозволяє отримати однозв'язний список сил, які є геном – елементом моделювання генетичного алгоритму.

Генетичний алгоритм моделює фізичні властивості об'єкта NPC з урахуванням дій NPC. Етапи роботи генетичного алгоритму включають (1) створення початкової популяції, (2) оцінка пристосовуваності, (3) відбір, (4) схрещування та (5) мутації [17].

Процедура роботи генетичного алгоритму відбувається наступним чином (рис. 1).

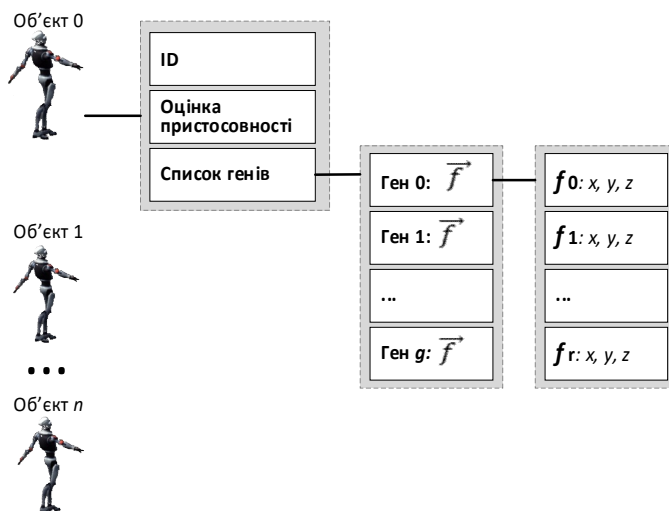


Рис. 1. Параметри моделювання генетичного алгоритму

Процедура починається з набору об'єктів. Кожен окремий об'єкт характеризується набором параметрів: ID об'єкта, оцінка пристосованості та список генів. У запропонованому підході список генів має динамічний діапазон, єдиним обмеженням якого є можливості оперативної пам'яті. Визначення сили для кожного твердого тіла об'єкту NPC дозволяє отримати однозв'язний список сил, які є одним геном.

Оцінка пристосованості визначає можливість того, що об'єкт буде обраний на наступному етапі для відтворення. Оцінки, що присвоюються об'єктам, обчислюються таким чином

$$R = L * T, \quad (18)$$

де L – пройдена відстань до падіння, T – кількість часових кроків, тобто, час утримання рівноваги об'єкта під час ходьби до падіння.

На етапі відбору відбираються об'єкти передачі своїх генів наступному поколінню. Реалізація відбору проводиться з використанням методу рулетки, при якому ймовірність відбору Pr i -го об'єкту розраховується як це представлено в (19)

$$Pr_i = \frac{R_i}{\sum_{i=1}^n R_i}, \quad (19)$$

де R_i оцінка пристосованості для i -го об'єкту, n – кількість об'єктів у популяції.

Схрещування є найважливішим етапом генетичного алгоритму. Для відібраних об'єктів точка схрещування вибирається випадковим чином.

Мутація використовується для підтримки різноманітності у популяції та запобігання передчасній конвергенції.

Етапи оцінки, відбору, схрещування та мутації повторюються ітеративно, доки не буде досягнуто критерію зупинення алгоритму або визначеної максимальної кількості ітерацій. В якості критерію припинення використовується метод знаходження оптимального або субоптимального рішення, що задовольняє функцію пристосованості.

В результаті застосування генетичного алгоритму реалізується оптимізація параметрів об'єкта популяції і вдосконалення моделі поведінки об'єкта, що дозволяє NPC триваліше утримувати рівновагу під час ходьби. Удосконалена модель може бути використана як початкова конфігурація об'єкту популяції.

Експеримент. Параметри моделі NPC. У першому етапі реалізації запропонованого підходу створено об'єкт NPC. Реалізація проведена з використанням середовища Unity. Об'єкт NPC є людиноподібною двоногою істотою (рис. 2).

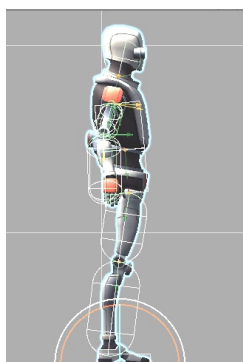


Рис. 2. Модель об'єкту NPC

Використано крок часу інтеграції 0.02 секунди. Навчання відбувалося у симуляції з прискореним часом, у якому швидкість перебігу часу становила від 1.0 до 3.0. У системі масштабування часу 1.0 є нормальною течією часу, а 0.0 є повною зупинкою часу. Використано нормальні параметри гравітації. Коефіцієнт повітряного опору, що впливає на об'єкт, доки він обертається під впливом сили обертання, становить 0.05. При цьому повітряний опір, який впливає на об'єкт, доки

він переміщується під впливом цих сил, відсутній. Загальна маса об'єкта NPC становить 60 кг з використанням коефіцієнту тертя 10.

Параметри роботи генетичного алгоритму. Обсяг початкової популяції становить 100 об'єктів. При значеннях оцінки пристосованості 60 робота генетичного алгоритму припиняється. На підставі результату оцінки пристосованості на етапі відбору відбирається 40 % об'єктів з найбільш високою оцінкою і 10 % вибирається випадково з інших об'єктів. Схрещування реалізується для відібраних об'єктів шляхом знаходження параметрів, що найбільш часто зустрічаються, визначаючи таким чином базову хромосому, яка схрещується з іншими за списком, тим самим формуючи нове покоління. Коефіцієнт схрещування у різних експериментах варіюється від 10 % до 40 % в серіях експериментів, а коефіцієнт мутації від 4 % до 20 %. Умовою припинення роботи з об'єктом є падіння. Падіння фіксується дією як дотику тулуба об'єкта до поверхні. Також падіння фіксується за допомогою певних параметрів осі z об'єкта.

Результати експерименту. Проведено серію експериментів, що мали на меті визначення оптимальних параметрів моделі за умови максимальної оцінки пристосованості, співвідношення часу симуляції популяції необхідного для отримання максимальної оцінки пристосованості та кількості поколінь.

Перша серія експериментів спрямована на дослідження впливу масштабування часу на результати симуляції з використанням генетичного алгоритму. Параметр швидкості перебігу часу становить 2.0, отже, за 1 секунду загалом проходить одне покоління. Загальна кількість поколінь становила 11458, максимальна одержана оцінка пристосованості 29,77, кількість ключових поколінь – 30. Час навчання 1000 поколінь становить близько 17 хвилин. Отримані оцінки пристосованості для популяцій цієї серії експериментів представлені на графіку (рис. 3) та таблиці (табл. 1).

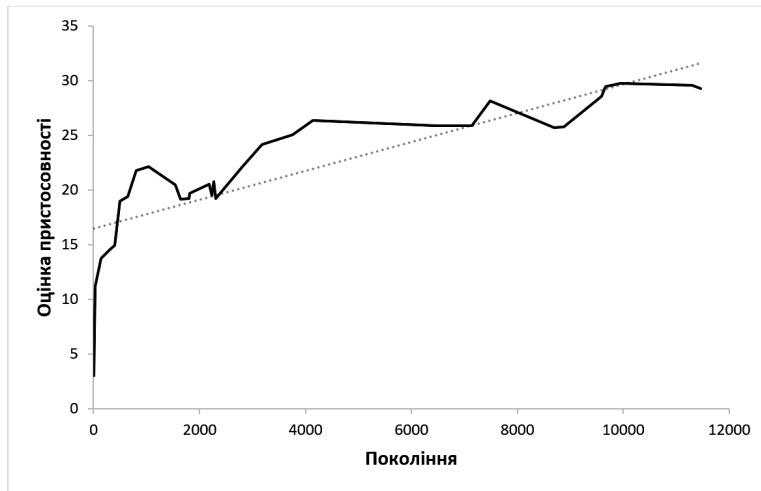


Рис. 3. Еволюційний розвиток поколінь з масштабуванням часу

Таблиця 1

Результати оцінки пристосованості ключових поколінь

Ключові покоління	Оцінка пристосованості	Ключові покоління	Оцінка пристосованості
1	3,05	2311	19,22
34	11,21	2811	22,15
137	13,78	3180	24,16
311	14,58	3756	25,05
405	14,97	4139	26,41
495	19,01	6438	25,91
648	19,43	7144	25,89
810	21,78	7487	28,19
1045	22,17	8700	25,74
1546	20,48	8874	25,78
1640	19,21	9589	28,59
1807	19,24	9660	29,49
1821	19,71	9944	29,77
2179	20,54	11299	29,62
2236	19,52	11458	29,31
2276	20,79		

З табл. 1 можна побачити, що на початок симуляції спостерігалось швидке зростання пристосовуваності об'єктів у перших поколіннях. Вже на 34-му поколінні об'єкти NPC набирають оцінку 11,21. Початкова випадкова комбінація дій виявилася ефективною. Єдиний і відповідно

домінуючий спосіб переміщення швидко закріпився і розвинувся до свого локального максимуму. Після цього з 648 по 2236 покоління можна спостерігати низку невдалих мутацій, які не прижилися, оскільки початковий метод переміщення показував вищу оцінку пристосовності. Початковий метод, як домінуючий, швидко витісняє мутації з нижчою оцінкою пристосовності. Починаючи з 2311 покоління та до 4139 покоління можна спостерігати планомірне зростання пристосовності об'єктів. Після досягнення свого локального максимуму наступного періоду, аж до 6438 покоління, не відбувається значних змін. З період 6438 і до 11458 покоління спостерігаються дві довготривалі мутації, перша з яких не приживається, і її оцінка пристосовності падає до своїх колишніх показників, а друга виявляється дуже ефективною і стає домінуючою.

Дана серія показує високу оцінку пристосованості ключових поколінь з використанням параметрів зміни швидкості часу та підтверджує, що даний метод є ефективним для навчання у системах із генетичним алгоритмом. Однак, описаний підхід має такі недоліки: відсутність можливості спостереження стимуляції в режимі реального часу, наявність певного обсягу обчислювальних ресурсів апаратного забезпечення.

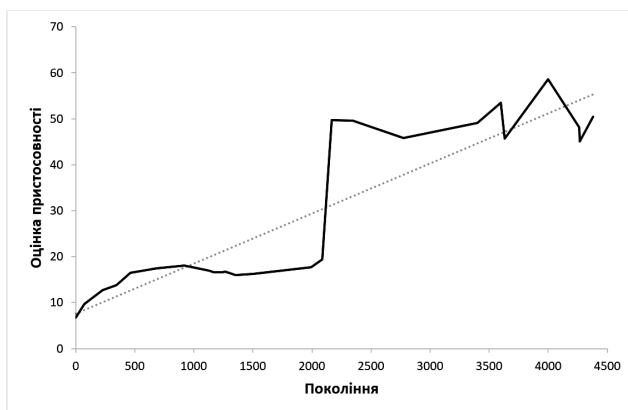


Рис. 4. Еволюційний розвиток поколінь

Друга серія експериментів проведена в умовах симуляції з використанням генетичного алгоритму у режимі реального часу. Параметр швидкості перебігу часу становить 1.0. Друга серія експериментів зайняла більше години на навчання поколінь. Навчання 1000 поколінь у реальному часі займає середньому від 1 до 2 годин. Загальна кількість поколінь становила 4377, максимальна одержана

оцінка пристосованості 58,65, кількість ключових поколінь – 24. Отримані оцінки пристосованості для популяції цієї серії експериментів представлені на рис. 4 та в табл. 2.

Таблиця 2

Ключові результати прогресії поколінь

Ключові покоління	Оцінка пристосовності	Ключові покоління	Оцінка пристосовності
1	6,72	1992	17,68
70	9,72	2087	19,46
226	12,72	2169	49,67
344	13,87	2346	49,53
463	16,43	2773	45,85
684	17,45	3398	49,12
921	18,01	3599	53,48
1125	16,96	3633	45,65
1168	16,64	3999	58,65
1235	16,64	4260	48,07
1270	16,76	4268	45,07
1353	16,06	4377	50,45
1499	16,26		

За даними рис. 4 та табл. 2 спостерігається значний ріст оцінки пристосовності у період з 2087 покоління до 2169. До 2087 покоління пристосовність зростала досить повільно і моделювання могло застрягти в локальному оптимумі, оскільки не одне покоління не могло дати результату 921 покоління, яке набрало 18,01 очок винагороди. У 2087 році поколінні відбулася успішна мутація, яка змогла закріпитися. Ця мутація є новим способом переміщення, який є більш ефективним у цьому середовищі, що й призвело до такого значного стрибка. На графіку (рис. 3) зростання оцінки пристосовності у 3599 поколінні до 53,48 балів та у 3999 до 58,6 балів. Дане зростання обумовлене мутацією.

Висновки. За результатами проведених експериментів визначено, що масштабування часу дозволяє прискорити навчання, що витрачається на симуляцію еволюції популяції, але потребує більшої кількості поколінь для отримання високої оцінки пристосовності об'єктів. Проведення симуляції в режимі реальної години потребує більше години на симуляцію, але дозволяє отримати більш високу оцінку пристосовності і таким чином, досягти поставленої мети – визначити вхідні параметри моделі поведінки об'єкту NPC, що максимізують сукупну оцінку пристосовності, яка виступає в якості оцінки винагороди.

Отримані параметри включають дані 197 генів об'єкта популяції, кожен з яких містить параметри координат x , y , z для 12 параметрів сил,

що застосовуються до твердих тіл об'єкту. Отримані параметри можуть бути використані як початкові вхідні значення при проведенні подальших експериментів з навчання моделі NPC.

Таким чином, застосування генетичного алгоритму дозволило удосконалити модель NPC за рахунок оптимізації параметрів моделі її поведінки з використанням генетичного алгоритму.

Список літератури:

1. The Next Generation 1996 Lexicon A to Z: NPC (Nonplayer Character). Next Generation. No. 15. Imagine Media. March 1996. – P. 38.
2. Wang, D. Creating human-like autonomous players in real-time first person shooter computer games / D. Wang , B. Subagdja, A.-H. Tan, G. W. Ng // In Proceedings of Twenty-First Annual Conference on Innovative Applications of Artificial Intelligence, 2009. – P. 14-16.
3. Wang, D. Creating autonomous adaptive agents in a real-time first-person shooter computer game / D. Wang, A.-H. Tan // Computational Intelligence and AI in Games, IEEE Transactions on, 7, 2015. – P.123-138.
4. Soni, B. Bots trained to play like a human are more fun / B. Soni, P. Hingston // In: IEEE International Joint Conference on Neural Networks, IJCNN 2008. – IEEE World Congress on Computational Intelligence, 2008. – P. 363-369.
5. Simon, H.A. Artificial intelligence: an empirical science / H.A. Simon // Artif. Intell. 77 (1), 1995. – P. 95-127.
6. McPartland M. Creating a multi-purpose shooter bot with reinforcement learning / M. McPartland, M. Gallagher // In Computational Intelligence and Games, 2008. CIG '08. IEEE Symposium On, Dec.2008. – P. 143-150.
7. Hong J.-H. Evolving reactive NPCs for real-time simulation game / J.-H. Hong, S.-B. Cho // In Proceedings of the 2005 IEEE Symposium on Computational Intelligence and Games (CIG05), 4-6 April 2005.
8. Mora A. M. et al. Evolving bot AI in unreal tm. European Conference on the Applications of Evolutionary Computation / A. M. Mora et al. – Springer, Berlin, Heidelberg, 2010. – P. 171-180.
9. Wang X. et al. SCC: efficient deep reinforcement learning agent mastering the game of StarCraft II / X. Wang et al. // International Conference on Machine Learning. – PMLR, 2021. – P. 10905-10915.
10. Pezzotti N. MimicBot: Combining Imitation and Reinforcement Learning to win in Bot Bowl / N. Pezzotti. – arXiv preprint arXiv:2108.09478. – 2021.
11. Geyer H. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities / H. Geyer, H. Herr // IEEE Trans Neural Syst Rehabil Eng., 18 (3). – 2010. – P. 263-73. doi: 10.1109/TNSRE.2010.2047592.
12. Wittenburg J. Dynamics of systems of rigid bodies / J. Wittenburg. – Springer-Verlag, 2013. – T. 33.
13. Falconer R. et al. Large Polygon Collider A Modular Physics Engine / R. Falconer et al. – 2009.
14. Featherstone R. Rigid body dynamics algorithms / R. Falconer. – Springer, 2014.
15. Mirtich B. V. Impulse-based dynamic simulation of rigid body systems / B. V. Mirtich. – University of California, Berkeley, 1996.
16. Goldberg D. E. Genetic algorithms and machine learning / D. E. Goldberg, J. H. Holland. – 1988.

References:

1. The Next Generation 1996 Lexicon A to Z: NPC (Nonplayer Character). *Next Generation. No. 15. Imagine Media*. March 1996. – P. 38.
2. Wang, D., Subagdja, B., Tan, A.-H., Ng, G. W. (2009), Creating human-like autonomous players in real-time first person shooter computer games. *In Proceedings of Twenty-First Annual Conference on Innovative Applications of Artificial Intelligence*. P. 14-16.
3. Wang, D., Tan, A.-H. (2015), Creating autonomous adaptive agents in a real-time first-person shooter computer game. *Computational Intelligence and AI in Games, IEEE Transactions on*, 7, P. 123–138.
4. Soni, B., Hingston, P. (2008), Bots trained to play like a human are more fun. *In: IEEE International Joint Conference on Neural Networks, IJCNN 2008. – IEEE World Congress on Computational Intelligence*, P. 363-369.
5. Simon, H.A. (1995), Artificial intelligence: an empirical science. *Artif. Intell.* 77 (1), P. 95-127.
6. McPartland M., Gallagher M. (2008), Creating a multi-purpose перша людина shooter bot with reinforcement learning. *In Computational Intelligence and Games, 2008. CIG '08. IEEE Symposium On*, Dec. P. 143-150.
7. Hong J.-H., Cho S.-B. (2005), Evolving reactive NPCs for real-time simulation game. *In Proceedings of the 2005 IEEE Symposium on Computational Intelligence and Games (CIG05)*, 4-6 April 2005.
8. Mora A. M. et al. (2010), Evolving bot AI in unreal tm. *European Conference on the Applications of Evolutionary Computation*. Springer, Berlin, Heidelberg. P. 171-180.
9. Wang X. et al. (2021), SCC: efficient deep reinforcement learning agent mastering the game of StarCraft II. *International Conference on Machine Learning. PMLR*. P. 10905-10915.
10. Pezzotti N. (2021), MimicBot: Combining Imitation and Reinforcement Learning to win in Bot Bowl. arXiv preprint arXiv:2108.09478.
11. Geyer H, Herr H. (2010), A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Trans Neural Syst Rehabil Eng.*, 18(3). P. 263-73. doi: 10.1109/TNSRE.2010.2047592.
12. Wittenburg J. (2013), *Dynamics of systems of rigid bodies*. Springer-Verlag. T. 33.
13. Falconer R. et al. (2009), *Large Polygon Collider A Modular Physics Engine*.
14. Featherstone R. (2014), *Rigid body dynamics algorithms*. Springer.
15. Mirtich B. V. (1996), *Impulse-based dynamic simulation of rigid body systems*. University of California, Berkeley.
16. Goldberg D. E., Holland J. H. (1988), *Genetic algorithms and machine learning*.

Статтю представив д.т.н., проф. Національного технічного університету "Харківський політехнічний інститут" О.А. Серков

Надійшла (received) 29.11.2021

Vynogradov Volodymyr, Master student
Volodymyr Dahl East Ukrainian National University
59-a Central Avenue, Severodonetsk, Luhansk region, Ukraine, 93400
tel./phone: (064) 522-89-97, e-mail: vova.vinog@gmail.com

Shumova Larysa, Cand.Sci.Tech, Assotiate Professor
Volodymyr Dahl East Ukrainian National University
59-a Central Avenue, Severodonetsk, Luhansk region, Ukraine, 93400
tel./phone: (064) 522-89-97, e-mail: shumova@ukr.net

Biloborodova Tetiana, Cand.Sci.Tech, Assotiate Professor
G.E. Pukhov Institute for Modelling in Energy Engineering
15 General Naumov Street, Kyiv, 03164, Ukraine
e-mail: beloborodova.t@gmail.com
ORCID ID: 0000-0001-7561-7484

УДК 004.942

Удосконалення моделі поведінки інтелектуального агента на основі генетичного алгоритму / Виноградов В.В., Шумова Л.О., Білобородова Т.О. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2021. – № 2 (6). – С. 70 – 88.

Представлено вирішення завдання удосконалення моделі поведінки неігрового персонажу, що виступає в якості інтелектуального агента, за рахунок оптимізації вхідних параметрів на основі генетичного алгоритму. Запропонований підхід включає розробку моделі неігрового персонажу: скелету, твердих тіл та реалізація динамічної моделі персонажу, заснованої на алгоритмі Фізерстоуна, та моделювання поведінки персонажу на основі генетичного алгоритму. Запропоновано формування моделі поведінки із використанням генетичного алгоритму, що моделює фізичні властивості персонажу з урахуванням його дій. Етапи роботи генетичного алгоритму включають: створення початкової популяції, оцінка пристосовуваності, відбір, схрещування та мутації. За результатами проведених експериментів визначено вхідні параметри моделі поведінки неігрового персонажу, що максимізують сукупну оцінку пристосованості, яка виступає в якості оцінки винагороди та які можуть бути використані як початкові вхідні значення при проведенні подальших експериментів. Іл.: 4. Табл.: 2. Бібліогр.: 16 назв.

Ключові слова: неігровий персонаж; інтелектуальний агент; моделювання; генетичний алгоритм.

УДК 004.942

Усовершенствование модели поведения интеллектуального агента на основе генетического алгоритма / Виноградов В.В., Шумова Л.А., Белобородова Т.А. // Вестник НТУ "ХПИ". Вестник НТУ "ХПИ". Серія: Інформатика и моделирование. – Харьков: НТУ "ХПИ". – 2021. – № 2 (6). – С. 70 – 88.

Представлено решение задачи усовершенствования модели поведения неигрового персонажа, выступающего в качестве интеллектуального агента, за счет оптимизации входных параметров на основе генетического алгоритма. Предлагаемый подход включает в себя разработку модели неигрового персонажа: скелета, твердых тел и реализация динамической модели персонажа, основанной на алгоритме Физерстоуна, и моделирование поведения персонажа на основе генетического алгоритма. Предложено формирование модели поведения с использованием генетического алгоритма, моделирующего физические свойства персонажа с учетом его действий. Этапы работы генетического алгоритма включают создание начальной популяции, оценка приспособляемости, отбор, скрещивание и мутации. По результатам проведенных экспериментов определены входные параметры модели поведения неигрового персонажа, максимизирующие совокупную оценку приспособляемости, которая выступает в качестве оценки вознаграждения, которые могут быть использованы как исходные значения при проведении дальнейших экспериментов. Ил.: 4. Табл.: 2. Библиогр.: 16 назв.

Ключевые слова: неигровой персонаж; интеллектуальный агент; моделирование; генетический алгоритм.

UDC 004.942

Behavior model of an intelligent agent based on a genetic algorithm / Vynogradov V.V., Shumova L.O., Biloborodova T.O. // Herald of the National Technical University "KhPI". Series of "Informatics and Modeling". – Kharkov: NTU "KhPI". – 2021. – № 2 (6). – P. 70 – 88.

A solution of improving the behavior model of a non-player character as an intelligent agent by optimizing input parameters based on a genetic algorithm is presented. The proposed approach includes the development of a non-player character model: a skeleton, rigid bodies, the implementation of a dynamic model based on the Featherstone algorithm, and modeling of the character's behavior based on a genetic algorithm. The formation of a behavior model using a genetic algorithm that simulates the physical properties of a character, taking into account his actions, is proposed. The stages of the genetic algorithm include creating an initial population, fitness score, selection, crossing and mutation. Based on the results of the experiments, the input parameters of the non-player character behavior model were determined, maximizing the cumulative fitness score, which acts as an estimate of the reward, which can be used as initial values for further experiments. Figs.: 4. Tabl.: 2. Refs.: 16 titles.

Keywords: non-player character; intelligent agent; simulation; genetic algorithm.