

УДК 004.81

В.Ю. ТІТОВА, В.В. БОЧУЛЯК
Хмельницький національний університет**АЛГОРИТМ ТА МАТЕМАТИЧНА МОДЕЛЬ ПОШУКУ
ШЛЯХУ ІГРОВИМ БОТОМ НА БАЗІ РОЙОВОГО ІНТЕЛЕКТУ**

У статті розглянуто методи пошуку шляхів ігровими ботами в комп'ютерних відеоіграх. Визначено їх переваги та недоліки та зроблено висновок, що для більш реалістичної поведінки ботів доцільно застосовувати методи, які використовують спільну пам'ять вже пройдених шляхів, а саме методи на базі ройового інтелекту. Автором запропоновано математичну модель методу пошуку шляху ігровим ботом та проведено моделювання його роботи.

Ключові слова: комп'ютерні відеоігри, ігровий бот, штучний інтелект, ройовий інтелект.

V.YU. TITOVA, V.V. BOCHULYAK
Khmelnytsky National University**GAMING BOT PATH SEARCH ALGORITHM AND MATHEMATICAL
MODEL BASED ON SWARM INTELLIGENCE**

This article describes gaming bots path search algorithms such as "face and return" and systematic search. Was identified the shortcomings of these algorithms and proposed for use the algorithm based on swarm intelligence. The author has presented the mathematical model of this algorithm, which allows formalizing of the search problem. Work of all three algorithms was simulated and was conducted comparative analysis of simulation results at the article. The simulation results allow concluding, that algorithm based on swarm intelligence can reduce time for shortest path search by memorizing the shortest path of one of the bots and transmitting information about it to other bots. The algorithm and the mathematical model presented in the article are the basis for the creation of gaming bot intelligent behaviour in computer games.

Keywords: computer videogames, gaming bot, artificial intelligence, swarm intelligence.

Вступ. У традиційних дослідженнях в галузі штучного інтелекту метою є створення справжнього інтелекту, здатного до навчання, соціальної взаємодії та прояву емоцій. Але у комп'ютерних іграх вимоги до штучного інтелекту значно менші [1].

Такому інтелекту не потрібно бути наділеним почуттями і самосвідомістю, не потрібно самонавчатися поза межами ігрового процесу. Справжня мета штучного інтелекту в іграх полягає в імітації розумної поведінки і в наданні гравцеві переконливих, правдоподібних супротивників – так званих ігрових ботів [1].

Основним принципом, що лежить в основі роботи ігрового штучного інтелекту, є прийняття рішень. І однією з важливих задач, що постають перед ним у процесі прийняття рішень, є сприйняття оточуючого середовища та пошук шляхів у ньому для переміщення від однієї точки до іншої.

Характеристика предметної області. Найпростішим алгоритмом переміщення ігрового бота з точки А в точку Б є алгоритм, умовно званий «зіткнутися та повернути» [1]. Його суть полягає у наступному.

1. Бот рухається у вибраному напрямку.
2. У випадку виникнення перешкоди рух змінюється у напрямку, при якому шлях до точки Б буде найкоротшим. Якщо визначити найкоротший шлях неможливо, вибір напрямку здійснюється випадковим чином.

Такий алгоритм непогано працює для нескладних ігор, але при його використанні ігрові боти часто опиняються у кутах або тупиках та не можуть з них вибратися, раз за разом тикаючись у стіни, що робить їх поведінку «безтолковою та неправдоподібною». Крім того, даний алгоритм не передбачає координації груп ботів та запам'ятовування ними найкращих шляхів від однієї точки до іншої, що змушує ботів раз за разом ходити одними і тими самими шляхами, застрягаючи в одних і тих самих кутах, що є суттєвим недоліком для будь-якої гри.

Тому, у сучасних комп'ютерних іграх, можливості ботів розширюють за рахунок надання їм пам'яті [1]. Це надає ботам можливість запам'ятовувати місця, в яких вони вже бували, та приймати грамотні рішення проте, куди їм рухатися далі. Якщо ж рухатися далі неможливо, алгоритм надає ботам можливість повертатися назад та шукати інший шлях. Таким чином, боти виконують систематичний пошук шляху від точки до точки.

Ось як це виглядає:

1. Бот рухається у вибраному напрямку.
2. Якщо шлях розгалужується, обирається один з можливих напрямків.
3. Якщо обраний шлях приводить в глухий кут, бот повертається до останнього розгалуження та обирає інший напрямок.
4. Якщо усі можливі шляхи пройдені безрезультатно, пошук шляхів припиняється.

Перевага цього алгоритму полягає в невисокому навантаженні на обчислювальні ресурси. Це означає, що можна підтримувати велику кількість ботів без уповільнення гри. Цей алгоритм також може використовувати переваги багатопоточної архітектури [1].

Єдиним недоліком є використання величезного обсягу пам'яті, оскільки кожен бот може відслідковувати цілу карту можливих шляхів, відшукуючи заново і ті, які вже були пройдені іншими ботами.

Щоб уникнути нераціонального використання пам'яті, доцільно використовувати для ботів спільну пам'ять, в якій буде зберігатися інформація про вже відомі шляхи. І зазначена інформація буде використовуватися ботами при прийнятті рішення щодо вибору напрямку свого руху. Таким чином, поведінка ботів буде відповідати поведінці «ройових агентів», прийняття рішень у яких базується на алгоритмах та методах ройового інтелекту.

Ройовий інтелект (англ. Swarm intelligence) описує колективну поведінку децентралізованої самоорганізації. Розглядається в теорії штучного інтелекту, як метод оптимізації. Термін був введений Херардо Бені і Ван Цзінем в 1989 році, в контексті системи клітинних роботів [2].

Системи ройового інтелекту, як правило, складаються з безлічі агентів (боїдов), локально взаємодіючих між собою і з навколишнім середовищем. Ідеї поведінки, як правило, виходять з природи, а особливо, з біологічних систем. Кожен боїд слідує дуже простим правилам і, незважаючи на те, що немає якоїсь централізованої системи управління поведінки, яка б вказувала кожному з них на те, що йому слід робити, локальні і, в деякій мірі, випадкові взаємодії призводять до виникнення інтелектуальної глобальної поведінки, неконтрольованої окремими боїдами. В цілому, ройовий інтелект повинен являти собою багатоагентну систему, яка б володіла самоорганізуючою поведінкою [2].

Алгоритм взаємодії ігрових ботів на основі ройового інтелекту. Для його реалізації введемо наступні поняття з використанням теорії множин [3]. Нехай існує скінчена множина ігрових ботів B , $B = \{b_1, b_2, \dots, b_n\}$.

Кожен з ботів в один момент часу може перебувати в одному з станів зі скінченої множини S , $S = \{s_1, s_2, \dots, s_m\}$ та виконувати одну з дій зі скінченої множини D , $D = \{d_1, d_2, \dots, d_o\}$.

Кожен бот через спільну пам'ять може обмінюватися інформацією з іншими ботами, повідомляючи про свій поточний стан та визнаючи про їх поточні стани. В залежності від отриманої інформації ним приймається рішення про вибір наступної дії d_{p+1}^k , тобто $d_{p+1}^k = f(s_p^1, s_p^2, \dots, s_p^n)$, де k – номер боту, який приймає рішення, p – поточний номер дії та стану усіх ботів, n – кількість усіх ігрових ботів, які взаємодіють у системі.

Перед кожним з ботів може поставати задача зі скінченої множини Z , $Z = \{z_1, z_2, \dots, z_q\}$. Під вирішенням поточної задачі z_p будемо розуміти досягнення ботом b_k такого стану $s_p^k = s_b$, при якому досягається мінімум деякого функціоналу F_b , $F_b = f(d_1^k, s_1^k, \dots, d_{p-1}^k, s_{p-1}^k)$, де $d_1^k, s_1^k, \dots, d_{p-1}^k, s_{p-1}^k$ – множина дій та станів, які необхідно виконати або пройти, щоб досягнути стану s_p^k .

Розглянемо принцип дії алгоритму на прикладі вирішення задачі пошуку ботом шляху з точки А до точки Б. При цьому під мінімумом функціоналу F_b будемо розуміти пошук найкоротшої відстані між цими двома точками.

При цьому можливі три варіанти розвитку подій.

Варіант 1. У спільній пам'яті наявна інформація про шляхи з точки А до точки Б, тобто множина можливих шляхів $L \neq \{\}$, $L = \{l_1, l_2, \dots, l_a\}$, де a – кількість можливих шляхів з точки А до точки Б. Тоді дія бота полягає у виборі найкоротшого шляху, $d_{p+1}^k = \min(l_1, l_2, \dots, l_a)$.

Варіант 2. У спільній пам'яті відсутня інформація про шляхи з точки А до точки Б, $L = \{\}$, але один або кілька ботів мають інформацію про точку Б (тобто знаходяться у цій точці). Тоді дії боту будуть наступними:

1. Після отримання задачі бот визначає свій поточний стан (в даному випадку місцезнаходження) та повідомляє його іншим ботам, а від них отримує інформацію про їх поточні стани (місцезнаходження).

2. Якщо між ботом b_k та ботами, які мають інформацію про точку Б немає перешкод, тоді $d_{p+1}^k = \min((s_p^k - s_p^1), (s_p^k - s_p^2), \dots, (s_p^k - s_p^a))$, де a – кількість ботів, які мають інформацію про точку Б. Якщо ж перешкоди наявні, тоді виконується перехід до п.3.

3. Бот визначає проміжкову точку Б', яка відповідає поточному стану одного з ботів, які не знаходяться у точці Б, але є близькими до неї. Якщо між точкою Б' та точкою А немає перешкод, тоді $d_{p+1}^k = \min((s_p^c - s_p^1), (s_p^c - s_p^2), \dots, (s_p^c - s_p^a)) + \min((s_p^k - s_p^1), (s_p^k - s_p^2), \dots, (s_p^k - s_p^c))$, де a – кількість ботів, які мають інформацію про точку Б, а c – кількість ботів, які не знаходяться у точці Б, але є близькими до неї. Якщо ж між точкою А та точкою Б' є перешкоди, то дії п. 3 повторюються доти, доки не буде знайдено точку, до якої перешкоди відсутні. А загальний шлях, який прокладається ботом, буде визначатися за

формулою $d_{p+1}^k = \sum_{i=1}^{j-1} \min((s_p^k - s_p^1), (s_p^k - s_p^2), \dots, (s_p^k - s_p^e))$, де j – кількість проміжкових точок, які необхідно пройти на шляху до точки Б.

Варіант 3. У спільній пам'яті відсутня інформація про шляхи з точки А до точки Б, $L = \{\}$, і жоден з ботів не має інформації про точку Б. В цьому випадку дії боту будуть відповідати діям у алгоритмі систематичного пошуку з вдосконаленням, яке полягає у тому, що при досягненні кожного нового розгалуження бот опитує інших ботів про зміну їх станів та при надходженні інформації про точку Б, починає діяти за варіантом 2.

Графічно алгоритм дій боту можна зобразити наступним чином:

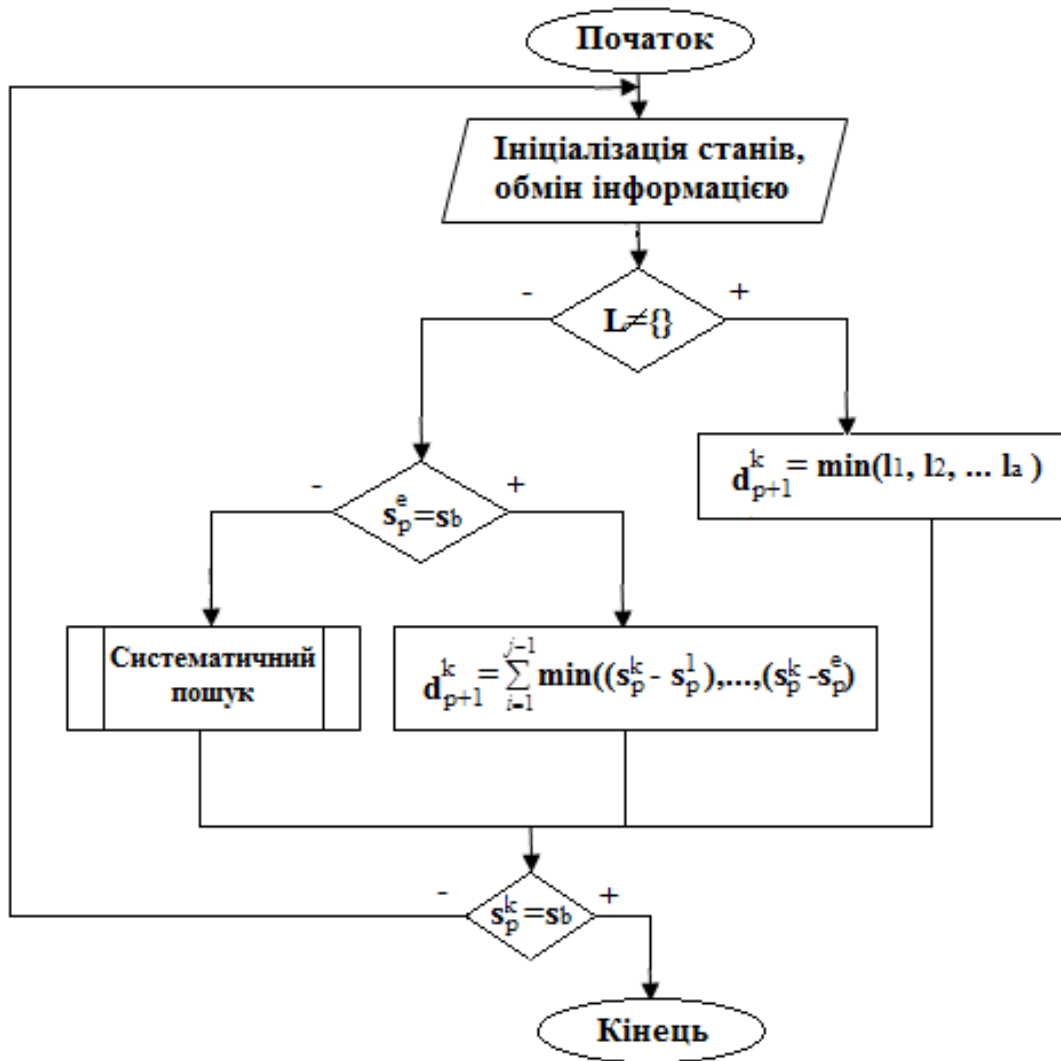


Рис. 1. Алгоритм пошуку ботом шляху з точки А до точки Б

Результати роботи трьох алгоритмів було змодельовано за допомогою відеогри, яка містить лабіринт. Перед п'ятьма ігровими ботами стояла задача знайти найкоротший шлях по ньому.

Результати моделювання представлені у таблиці 1. Для кожного з алгоритмів кількість експериментів становила десять. В якості результатів обирався час, який знадобився кожному конкретному боту для того, щоб пройти лабіринт. Бот, перед яким ставилася задача пошуку шляху, обирався випадковим чином.

Мінімальний час пошуку становив 11 с. При цьому, боти, які шукали шлях за допомогою алгоритму «зіткнутися та повернути» так і не змогли досягнути цього показника, через часті потрапляння в кути та простоювання у них. Один з ботів, які шукали шлях за допомогою алгоритму систематичного пошуку, зміг досягнути цього показника один раз, проте через відсутність обміну результатами, він не зміг передати цю інформацію іншим ботам, і кожен з них був змушений шукати свій власний шлях.

Боти, які шукали шлях за допомогою алгоритму на базі ройового інтелекту, після знаходження найкоротшого шляху, мали змогу його запам'ятовувати та передавати інформацію про нього іншим ботам, а ті, в свою чергу, використовували цю інформацію при власному пошуку.

Результати моделювання роботи алгоритмів пошуку шляху ботом

№ експерименту	Алгоритм «зіткнутися і повернути»	Алгоритм систематичного пошуку	Алгоритм на базі ройового інтелекту
1	16 с	17 с	20 с
2	22 с	14 с	16 с.
3	25 с	13 с	14 с
4	16 с	13 с	13 с
5	20 с	13 с	12 с
6	20 с	11 с	11 с
7	24 с	14 с	11 с
8	35 с	16 с	11 с
9	23 с	20 с	11 с
10	23 с.	14 с	11 с
Сер. зн.	22,4 с	14,5 с	13 с

Висновки. У даній статті розглянуто алгоритми пошуку шляху ігровими ботами такі, як «зіткнутися та повернути» та систематичний пошук. Визначено недоліки цих алгоритмів та запропоновано для використання ігровими ботами алгоритм на базі ройового інтелекту. Автором представлено математичну модель зазначеного алгоритму, яка дозволяє провести формалізацію задачі пошуку.

Також у статті було змодельовано роботу усіх трьох алгоритмів та проведено їх порівняльний аналіз. Результати моделювання дозволяють зробити висновки, що алгоритм на базі ройового інтелекту дозволяє зменшити час пошуку найкоротшого шляху ботами за рахунок запам'ятовування найкоротшого шляху одним з ботів та передачі інформації про нього іншим ботам.

Наведені у статті алгоритм та математична модель є основою для створення інтелектуального методу поведінки ігрового боту у комп'ютерних відеоіграх.

Література

1. Algorithms and Networking for Computer Games / Jouni Smed, Harri Nakonen. – John Wiley & Sons, 2006. – 226 p.
2. Субботін С. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей : монографія / С.О. Субботін, А.О. Олійник, О.О. Олійник ; під заг. ред. С. О. Субботіна. — Запоріжжя : ЗНТУ, 2009. — 375 с.
3. Верещагин Н.К. Лекции по математической логике и теории алгоритмов. Часть 1. Начала теории множеств / Н. К. Верещагин, А. Шень. – 3-е изд., стереотип. – М. : МЦНМО, 2008. – 128 с.

Рецензія/Peer review : 2.3.2017 р. Надрукована/Printed : 19.4.2017 р.
Рецензент: д.т.н., проф. Боровик О.В.