

ПРОЕКТУВАННЯ АДАПТИВНИХ ІГРОВИХ АЛГОРИТМІВ MINIMAX З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Вінницький національний технічний університет

Анотація

У статті представлено процес проектування адаптивного ігрового алгоритму, що використовує принцип MiniMax з використанням технології штучного інтелекту, що забезпечує стратегічну поведінку супротивника в логічних ігрових системах із динамічною складністю.

Ключові слова: геміфікація, штучний інтелект, логічне мислення, адаптивний алгоритм, MiniMax.

Abstract

The article presents the process of designing an adaptive game algorithm that uses the MiniMax principle with the use of artificial intelligence technology, which ensures strategic behavior of the enemy in logical game systems with dynamic complexity.

Keywords: gamification, artificial intelligence, logical thinking, adaptive algorithm, MiniMax.

Вступ

Сучасний розвиток інформаційних технологій, зокрема у сфері когнітивного моделювання, адаптивного навчання та гейміфікації, зумовлює потребу у створенні інтелектуальних алгоритмічних систем, здатних динамічно реагувати на дії користувача, моделювати складну логіку прийняття рішень та адаптуватися до змінних умов взаємодії. Ігрові алгоритми, що реалізують стратегічну поведінку супротивника, стали не лише популярним інструментом розваг, а й ефективним методом розвитку когнітивного мислення, концентрації, планування та аналітичних здібностей.

Одним з найбільш ефективних підходів до моделювання поведінки супротивника в логічних іграх є використання алгоритмів штучного інтелекту. Зокрема, алгоритм MiniMax продемонстрував високу ефективність у іграх з повною інформацією завдяки своїй здатності прогнозувати варіанти гри на декілька кроків уперед та формувати оптимальну стратегію дій. У поєднанні з адаптивними механізмами та евристичними функціями, MiniMax здатен забезпечити динамічну зміну рівня складності, підвищуючи залученість користувача.

Проектування таких ігрових алгоритмів є актуальним завданням дисципліни «Проектування інструментального програмного забезпечення», оскільки вимагає створення формалізованих моделей, структурної організації логіки, верифікації функціональності та інтеграції з обчислювальними сервісами. Особливу увагу приділяється правильній побудові логіки гри, модулю оцінки ходів, функціоналу прийняття рішень та зв'язку з інтерфейсом користувача [1].

Метою дослідження є реалізації стратегічної поведінки супротивника в логічних ігрових системах із динамічною складністю.

Аналіз методів побудови ігрових систем із використанням алгоритмів штучного інтелекту

Ігрові застосунки, що мають на меті розвиток мислення, поділяються на декілька категорій залежно від когнітивних функцій, які вони тренують, а також за логічною структурою та правилами. Найбільш поширеними є стратегічні ігри, головоломки, ігри на увагу та концентрацію, навчальні симулятори. Важливою характеристикою таких ігор є наявність виклику для інтелектуальних здібностей користувача, потреба в аналізі ситуації, формуванні стратегії та прийнятті рішень.

Ігри для розвитку мислення можуть бути класифіковані як:

- ігри з повною інформацією (наприклад, шахи, реверсі, хрестики-нулики), у яких усі елементи гри відомі учасникам, а процес прийняття рішень є строго логічним;
- ігри з неповною інформацією, які вимагають від гравця оцінки ризиків та формування гіпотез;
- покрокові логічні ігри, орієнтовані на вирішення послідовностей задач зі зростаючим рівнем складності;
- ігри з динамічною складністю, де правила або рівень супротиву змінюється відповідно до результатів користувача.

У контексті реалізації алгоритмів штучного інтелекту найбільшу практичну цінність мають саме ігри з повною інформацією, оскільки вони дозволяють формалізувати дерево можливих станів, визначити виграшні стратегії та реалізувати їх у вигляді програмного модуля.

Штучний інтелект в ігрових системах виконує функцію автоматизованого опонента, що моделює мислення людини та приймає рішення відповідно до визначених алгоритмів. Основними задачами ШІ в логічних іграх є: побудова дерева можливих ходів, аналіз ситуації на ігровому полі, прогнозування дій користувача, вибір найкращого ходу згідно з заданою метрикою ефективності.

У навчально-тренувальних ігрових середовищах ШІ повинен не лише перемагати гравця, а й підтримувати рівень виклику, поступово підвищуючи складність гри. Для цього застосовуються адаптивні механізми, що дозволяють коригувати поведінку ШІ відповідно до статистики перемог/поражок, часу на обдумування, кількості зроблених помилок тощо.

Алгоритм MiniMax є класичним прикладом детермінованого пошуку у дереві рішень і широко використовується в іграх із двома учасниками, де кожен гравець має протилежну мету — максимізувати або мінімізувати виграш [2]. Принцип дії алгоритму базується на рекурсивному аналізі всіх можливих ходів до певної глибини з подальшою оцінкою вигідності кожного з них за допомогою евристичної функції. У вузлах дерева, що відповідають ходам гравця, обирається максимум можливого виграшу, а у вузлах, що відповідають ходам супротивника, — мінімум.

Для зменшення обчислювальної складності використовується модифікація MiniMax — алгоритм з обрізанням α - β , який дозволяє відкидати гілки дерева, що не впливають на остаточне рішення. Це суттєво знижує кількість розрахунків, забезпечуючи прийнятну швидкість навіть у системах із великою кількістю можливих варіантів гри.

У контексті розвитку мислення алгоритм MiniMax має низку переваг:

- створює умови для стратегічного мислення користувача;
- стимулює навички аналізу ситуації на декілька кроків наперед;
- адаптується до дій користувача через зміну глибини дерева аналізу або вагових коефіцієнтів оцінки;
- може виступати як метод діагностики когнітивного рівня за часом і якістю рішень користувача.

Реалізація MiniMax у web-системах дозволяє досягнути балансу між складністю гри та комфортом взаємодії, забезпечуючи інтерактивність, повторюваність експериментів та масштабованість функціоналу. Це робить його одним з ключових інструментів для розробки систем, орієнтованих на інтелектуальний розвиток [3].

Розробка алгоритму MiniMax

Розробка адаптивного ігрового алгоритму MiniMax вимагає системного підходу до проєктування архітектури програмної системи, що передбачає декомпозицію задачі на незалежні компоненти, які можуть бути протестовані та вдосконалені автономно. Важливо забезпечити формалізацію логіки гри у вигляді чітких функціональних блоків – генерації можливих ходів, аналізу дерева рішень, застосування евристики та формування відповіді штучного інтелекту.

Особливу увагу необхідно приділити проєктуванню структури дерева пошуку. Для цього розробляються класи моделей гри, що інкапсулюють стан ігрового поля, допустимі дії та функцію переходу до нового стану. Реалізація MiniMax ґрунтується на рекурсивному обході цих моделей, для чого проєктується окрема функціональна одиниця, що забезпечує оцінювання кожного стану згідно з параметрами поточного гравця.

Адаптивність досягається через варіативну глибину рекурсії, яка визначається на основі динамічних метрик: результатів попередніх ігор, середнього часу прийняття рішень, кількості помилкових ходів. Це реалізується через спеціальний адаптивний модуль, який перед кожною сесією гри генерує конфігурацію ІІІ.

Оскільки алгоритм MiniMax виконує значну кількість обчислень, проектування повинно враховувати обмеження продуктивності – шляхом реалізації оптимізацій (обрізання α - β , кешування оцінок, використання кінцевих евристичних таблиць). Це знижує кількість рекурсивних викликів і забезпечує стабільну швидкодію навіть при розширенні глибини аналізу.

Для забезпечення коректної роботи всієї системи реалізується схема контролю переходів між станами гри, що синхронізується із зовнішніми подіями (хід користувача, кінець гри, оновлення рейтингу). Це вимагає інтеграції модуля MiniMax з іншими частинами web-додатку через API або подієву модель взаємодії, що забезпечує модульність і можливість масштабування. Реалізовані блок-схеми алгоритмів наведені на рисунку 1.

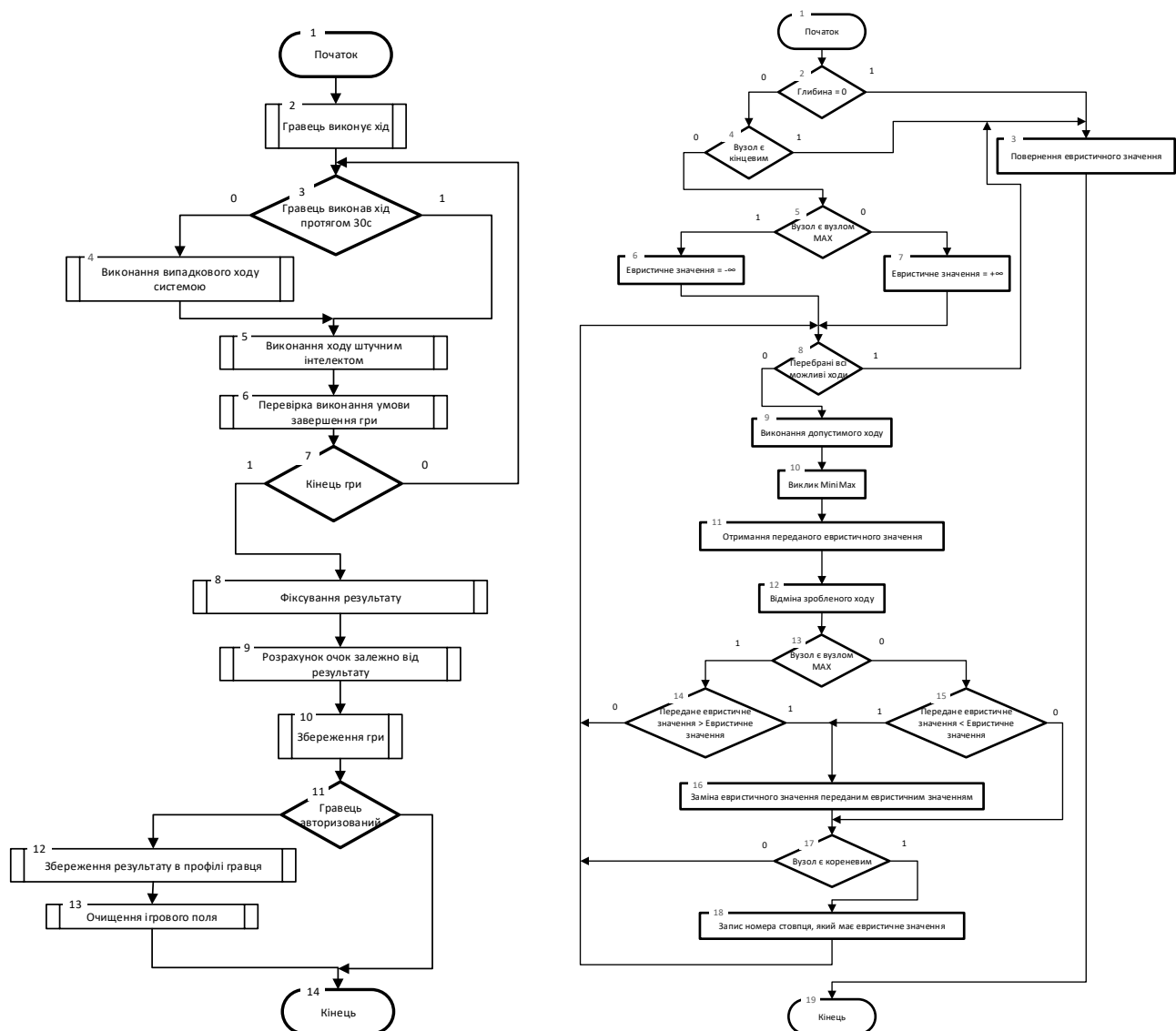


Рисунок 1 – Розроблений ігровий алгоритм MiniMax. а) Алгоритм гри проти штучного інтелекту б) Алгоритм MiniMax роботи штучного інтелекту

Отже, було виконано розробку ігрового алгоритму MiniMax з використанням штучного інтелекту. Розроблений алгоритм може бути використаний у гейміфікованих системах для розвитку логічного мислення. Реалізація адаптивного алгоритму MiniMax передбачає структурне планування, об'єктно-орієнтоване моделювання, оптимізацію алгоритмів, адаптацію до змінного навантаження та забезпечення взаємодії з іншими модулями в рамках інтегрованої програмної системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кучер М.В., Мацедонський С.О. Розробка ігрового web-додатку для розвитку логічного мислення/М.В. Кучер, С.О. Мацедонський, В.П. Майданюк// Матеріали конференції «LIII Науково-технічна конференція підрозділів Вінницького національного технічного університету (2024) », Вінниця, 2024. [Електронний ресурс] – Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2024/paper/view/20790/17230>
2. MiniMax algorithm in game theory [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/minimax-algorithm-in-gametheory-set-1-introduction/>
3. Development of students logical thinking [Електронний ресурс] – Режим доступу до ресурсу: <https://mental.ua/en/rozvytok-lohichnoho-myslennia-uchnia/>

Кучер Максим Володимирович – студент групи 2ПІ-24м, Вінницький національний технічний університет, м. Вінниця, Факультет інформаційних технологій та комп'ютерної інженерії, mag.kucher.2025@gmail.com.

Ліщинська Людмила Броніславівна – д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: llb@vntu.edu.ua

Kucher V. Maksym – student of the group 2PI-24m, Vinnytsia national technical university, Vinnytsia, faculty of information technology and computer engineering, mag.kucher.2025@gmail.com.

Lishchynska Lyudmyla Bronislavivna – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: llb@vntu.edu.ua