

**Автор:**

Лихошапко Ярослав Сергійович,  
студент 41 КН групи

**Науковий керівник:**

Франчук Василь Михайлович,  
доктор педагогічних наук,  
доцент, завідувач кафедри комп'ютерної  
та програмної інженерії

## ІНТЕГРАЦІЯ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ В ІГРОВИЙ РУШІЙ UNITY ДЛЯ МОДЕЛЮВАННЯ ПОВЕДІНКИ ПЕРСОНАЖІВ

**Анотація.** У роботі розглянуто підхід до моделювання поведінки неігрових персонажів у 2D-середовищі Unity. Об'єктом дослідження є процес моделювання поведінки неігрових персонажів у 2D-середовищі Unity. Предметом дослідження є методи інтеграції FSM, правил логіки та компонентів Unity для реалізації поведінки NPC. Метою дослідження є розроблення та обґрунтування 2D-прототипу в Unity, у якому поведінка NPC реалізується за допомогою простих алгоритмів штучного інтелекту, скінченного автомата станів і компонентів реагування на положення гравця. Практична частина орієнтована на використання компонентної моделі Unity, зони реагування Area Awareness, скриптів керування рухом і механізмів відстеження поточного стану персонажа. Запропонований підхід дає змогу створити зрозумілу та керовану модель поведінки NPC без застосування складних методів машинного навчання.

**Ключові слова:** штучний інтелект, Unity, NPC, 2D-прототип, скінченний автомат станів, Area Awareness, поведінка персонажів.

**Вступ.** У сучасному ігровому розробленні штучний інтелект є важливим засобом формування динамічного ігрового середовища [7; 3]. Саме поведінка неігрових персонажів значною мірою визначає, наскільки переконливою та інтерактивною сприймається гра. Окрім значення має візуальна й інтерактивна складова таких прототипів, оскільки комп'ютерна графіка та інтерактивні технології підсилюють наочність роботи цифрової системи. Навіть у невеликому 2D-проекті NPC (неігровий персонаж), який не лише перебуває на сцені, а й реагує на положення гравця, робить ігровий процес більш змістовним. Для реалізації такої поведінки доцільно використовувати прості алгоритмічні підходи, зокрема скінченні автомати станів, умовну логіку та параметри зони виявлення [1; 2]. Unity є зручним середовищем для такої реалізації, оскільки підтримує компонентну структуру об'єктів, 2D-колізії та програмування мовою C# [5; 6].

**Постановка задачі.** Основне завдання дослідження полягає у створенні 2D-прототипу, в якому використання NPC можна змінювати поведінку відповідно до дій гравця. Для цього необхідно визначити структуру сцени, налаштувати об'єкти Player (Гравець), Enemy (Ворог) та Ground (Земля, платформа), реалізувати компоненти руху й колізії, а також побудувати логіку переходів між станами. Особливу увагу потрібно приділити тому, щоб поведінка NPC була зрозумілою для користувача та водночас достатньо гнучкою для подальшого розширення. Важливо, щоб алгоритм не перевантажував систему і міг бути пояснений через просту модель FSM (Скінчений автомат станів) [1].

**Мета роботи.** Метою роботи є розроблення та обґрунтування 2D-прототипу в Unity, у якому поведінка NPC реалізується за допомогою простих алгоритмів штучного інтелекту, скінченного автомата станів і компонентів реагування на положення гравця.

**Основна частина.** У практичній реалізації поведінка NPC подана як система станів. Базовим станом є патрулювання, під час якого персонаж виконує задану поведінку на сцені. Після появи гравця в зоні реагування Area Awareness (Зона реагування) NPC переходить до стану переслідування. Така організація відповідає моделі скінченного автомата станів, оскільки кожен стан має окреме призначення, а перехід між ними відбувається лише за визначених умов [1].

Для організації ігрової сцени використано компонентний підхід Unity. Об'єкти сцени мають власні компоненти: візуалізацію, колайдери, скрипти керування та поведінкову логіку. Об'єкт Enemy містить компоненти Character Controller (Компонент керування персонажем), Custom Collision (Власна логіка колізій), Enemy Fsm (Скінченний автомат станів ворога) і Area Awareness. Box Collider 2D (Прямокутний 2D-колайдер) використовується для визначення меж зіткнення, а компонент Enemy Fsm відповідає за зміну станів NPC [1; 6; 5].

Особливістю реалізації є відокремлення логіки руху від логіки прийняття рішень. За допомогою компонента Character Controller задаються параметри переміщення персонажа, тоді як Enemy Fsm керує поведінковими переходами. У межах автомата використовуються умови наявності цілі та віддалення гравця. Це дозволяє NPC переходити від патрулювання до переслідування, а після втрати цілі повертатися до базової поведінки [1].

Для перевірки роботи алгоритму застосовується візуальне відображення поточного стану NPC. Компонент AI State Logger (Компонент відображення стану ШІ) передає дані до текстового елемента Current State (Поточний стан), що спрощує тестування у режимі Play Mode (Режим тестування). Завдяки цьому можна безпосередньо спостерігати, чи коректно відбувається зміна станів у відповідь на появу або віддалення гравця [5].

**Висновки.** У результаті дослідження показано, що прості алгоритми штучного інтелекту можуть бути ефективно використані для моделювання поведінки NPC у 2D-проекті Unity. Скінченний автомат станів забезпечує логічну структуру поведінки, Area Awareness дозволяє визначати момент реагування на гравця, а компонентний підхід Unity спрощує налаштування та тестування системи. Запропонована модель може бути використана як основа для подальшого розвитку, зокрема додавання нових станів, складніших умов виявлення, анімацій або взаємодії з кількома NPC. Використання візуальної демонстрації та інтерактивної перевірки відповідає загальній логіці застосування інтерактивних технологій у роботі з комп'ютерною графікою [4].

## Список використаних джерел

1. Nugraha, C., Purnamasari, A. I., Bahtiar, A., & Tohidi, E. (2025). Implementation of finite state machine on NPCs to improve game productivity. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, 4(3), 1673-1677.
2. Larsson, M. Viability of Machine Learning for enemies in Video Games. 2022.
3. Chen, X. (2025). An Overview of Different Artificial Intelligence Applications in Games.
4. Мальований В. А., Франчук Н. П. Використання технологій інтерактивного навчання під час навчання комп'ютерної графіки. 2024. URL: [https://lib.iitta.gov.ua/id/eprint/743063/1/80-82\\_Malovanyi\\_Franchuk.pdf](https://lib.iitta.gov.ua/id/eprint/743063/1/80-82_Malovanyi_Franchuk.pdf)

5. Unity Technologies. Unity User Manual 2022.3 (LTS). URL: <https://docs.unity3d.com/2022.3/Documentation/Manual/UnityManual.html>
6. Unity Technologies. GameObject. Unity Scripting API 2022.3. URL: <https://docs.unity3d.com/2022.3/Documentation/ScriptReference/GameObject.html>
7. Wan, W. (2025, August). Application of Artificial Intelligence Technology in Gaming NPC and Existing Problems. In *Proceedings of the 2025 3rd International Conference on Image, Algorithms, and Artificial Intelligence (ICIAAI 2025)* (p. 104). Springer Nature.