

УДК 004.89:004.96

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РОЗРОБКИ КОМП'ЮТЕРНИХ ІГОР В UNREAL ENGINE

Кулішов Є. А.

Науковий керівник – асист. Солодов В. Д.

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

e-mail: yehor.kulishov@nure.ua

For the past decades, videogame industry has gained a massive influence over the globe, and it grows larger each passing year. One of the key elements in games, that captivates eyes of many gamers around the world, is artificial intelligence, the ever-evolving technology. It allows developers to create behavioral patterns for in-game characters, that dictates them how to act when meeting certain situations, markers, etc. And the more industry grows, more advancements on videogame AI will be made. This text's purpose is the analysis and comparison of pros and cons of game engines, methods of creation of projects and its influence on AI, and difference in usage of in-game mechanics between AI and player.

За останні 25 років відеоігри остаточно укорінилися у суспільстві одночасно як розвага, вид мистецтва, та спосіб заробітку. В грудні 2023 року, вже майже 3,2 мільярда людей на постійній основі грали у комп'ютерні ігри, а річний дохід індустрії за 2023 рік склав за різними оцінками від 217,6 до 233,5 мільярдів доларів лише за продажу самих ігор. Та, за прогнозами, за 2024 рік дохід від індустрії може збільшитися на 8,76 відсотків відносно 2023 р. [1, 2].

І невід'ємною частиною процесу розробки ігор є використання методів штучного інтелекту (AI – Artificial Intelligence) для створення неігрових персонажів. Не завжди ці методи вдало використовуються під час розробки ігор: багато ігор не досягли бажаного рівня якості та втратили у продажах (як, наприклад, гра «Redfall» [2]) через недостатньо продуману роботу над AI компонентами.

Дана робота ставить за ціль аналіз процесу розробки неігрових персонажів з використанням методів штучного інтелекту під час створення комп'ютерних ігор.

Перед початком розробки потрібно обрати відповідне ігрове програмне забезпечення, або ігровий «рушій», відповідно до проекту. Серед існуючих на ринку найчастіше використовуються такі, як Unreal Engine, Unity та Godot через їх доступність [3]. Unreal Engine найчастіше використовується у великих проектах, де більш важливими є візуальне сприйняття гри користувачами та правдоподібність фізичних анімацій. Програмування AI в ньому хоч і досить гнучке, але створення складного сценарію поведінки для неігрового персонажа на основі штучного

інтелекту буде вибагливим щодо обчислювальних ресурсів, що знижує загальну продуктивність. Unity має просте та інтуїтивне управління, через що його використовують розробники без значного досвіду, але на цьому рушії існує проблема взаємодії великої кількості неігрових персонажів із штучним інтелектом через недосконалість технології прокладання шляху для AI. Godot гнучкий у створенні комплексного штучного інтелекту і дозволяє самому розробнику модифікувати «рушій» згідно особливостям завдання. Але він не розрахований на розробників без значного досвіду та на великі проекти, оскільки орієнтований на створення 2D-ігор.

Після обрання ігрового «рушія» потрібно обрати спосіб створення проектів. Існує два методи створення проектів: перший - за допомогою системи «Blueprint» (або «Креслення»), де зі спеціальних блоків з записаними функціями створюють послідовності дій персонажів. Другий - кодувати всі дії через мову програмування, яку підтримує «рушій». Ці методи можна використовувати одночасно в одному проекті, але це ускладнюватиме архітектуру створеного прототипу, додаючи ще один шар коду, який треба відгладжувати, щоб він працював коректним чином.

Як приклад, у проектах на «рушії» Unreal Engine мову програмування C++ використовують тоді, коли проекту в першу чергу потрібна велика точність дій для складних задач та обчислювальна ефективність. Проекти, створені цим методом, краще оптимізовані, їх легше перенести з однієї платформи на іншу. При такому методі можливо зробити комплексний AI, який буде легко налагодити. Але такий метод не розрахований на розробників без значного досвіду та малі команди розробників через складність мови, тож такі проекти потребують дуже великої тривалості розробки. Також, C++ має менш потужний функціонал для роботи з візуальною частиною гри, ніж «Blueprint», і погано узгоджується з такими функціями, як «line trace» («трасування лінії»), що може погіршити взаємодію AI з навколишнім середовищем та призвести до того, що функції ігрового персонажу не будуть працювати.

«Blueprint», в свою чергу, дуже добре підходить для розробників без значного досвіду, надає більш доступне візуальне подання даних та функцій для тих, хто не має досвіду програмування. При такому методі легко працювати із дизайном гри та її кодуванням одночасно, а присутня система налагодження в режимі реального часу дозволяє розробникам виявляти та виправляти проблеми під час роботи програми. Недоліки цього методу у тому, що його складно організувати та оптимізувати у великих проектах та складніше налагодити у порівнянні C++. На додаток, «Blueprint» не має повного доступу до інтерфейсу програмування додатку, що блокує використання передових засобів розробки. AI при «Кресленні» складніше редагувати, але він матиме кращу взаємодію з ігровим середовищем.

Жоден з методів не є кращим за інший, вони лише розраховані на

різні рівні досвіду та вміння команди розробників. У професійному середовищі найчастіше використовують гібридний підхід, оскільки він дає свободу в роботі з різними аспектами гри та можливістю оптимізувати процес, як за часом, так і за ресурсами.

Створюючи AI, слід розуміти, що при будь-якому методі розробки створений для гравця функціонал, починаючи від переміщення до нестандартних здібностей, не може бути просто перетягнутий на штучний інтелект. Наприклад, у створеному на Unreal проекті, механізм пострілу гравця викликається через затискання клавіш, але для AI ця функція має бути автоматизована та викликатися за певними критеріями. Для створення більш реалістичної поведінки персонажів в Unreal можна використовувати конфігурації почуттів. Якщо персонаж гравця загалом базується на почуттях зору та слуху самого гравця, то в AI потрібно вручну прописувати це, як окремі системи, відповідно до поставлених задач. Таким чином, штучний інтелект буде виконувати дії, базуючись на різних почуттях, наче людина. Але може статися так, що через отримання великої кількості сигналів з різних систем водночас, модель прийняття рішень неігрового персонажу буде нездатною згенерувати відповідь, і персонаж зупиниться. Через це, потрібно чітко розуміти, які механіки буде вжито та які відповідні конфігурації почуттів буде додано до AI.

Таким чином, можна зрозуміти чому штучний інтелект є такою важкою частиною розробки ігор. Починаючи від вибору «рушія» під задуманий проект до створення простих рухів персонажів - усе має бути зважено, оцінено та реалізовано в проекті під час розробки гри, яка буде дивувати гравців у всьому світі.

Список використаних джерел:

1. 59+ Video Gaming Statistics For 2024 (Users & Popular Games). URL: <https://www.answeriq.com/video-gaming-statistics/> (дата звернення: 10.02.2024).

2. What's the most popular gaming genre in 2023? URL: <https://pinglestudio.com/blog/industry-news/whats-the-most-popular-gaming-genre-in-2023> (дата звернення: 10.02.2024).

3. Unity or Unreal? Find the Perfect Engine for Beginners. URL: <https://www.toolify.ai/ai-news/unity-or-unreal-find-the-perfect-engine-for-beginners-38382> (дата звернення: 10.02.2024).

4. Олещенко В. Б. Використання штучного інтелекту для створення ігрового контенту // Радіоелектроніка та молодь у XXI столітті: тези доповідей 27-го Міжнародного молодіжного форуму, 10–12 травня 2023 р. – Харків : ХНУРЕ, 2023. – Т. 3. – С. 153–154.