

УДК 004.89

Фурсова Н.А., Кривицький Д.М.

ТЕХНОЛОГІЇ МАШИННОГО НАВЧАННЯ В РОЗРОБЦІ КОМП'ЮТЕРНИХ ІГОР

У статті досліджено використання сучасних технологій машинного навчання в розробці комп'ютерних ігор на прикладі двовимірної навчально-пізнавальної гри в жанрі космічного симулятора. Визначено можливості інструментарію Unity ML-Agent та його об'єктів. Розкрито основні етапи розробки комп'ютерної гри, проведена декомпозиція задач використання інтелектуальних агентів. Обрано ігрові механіки, розроблено архітектуру конфігурації навчального середовища. Представлено послідовність взаємодії гравця з грою та інтелектуальними агентами під час ігрової сесії.

Ключові слова: машинне навчання, комп'ютерна гра, інтелектуальний агент, ігровий двигун, ML-Agents.

Постановка проблеми. Ігрова індустрія невпинно розвивається, з'являються нові підходи та інструментальні засоби розробки комп'ютерних ігор. Це обумовило появу різноманітності напрямів, жанрів, видів та технологій проектування ігор. Разом з тим, значний інтерес становить застосування технологій машинного навчання у розробці штучного інтелекту ігрових світів. Виходячи з цього, вважаємо за доцільне дослідити сучасні технології машинного навчання в розробці комп'ютерних ігор.

Аналіз останніх досліджень і публікацій. Дослідженню штучного інтелекту в розробці комп'ютерних ігор та використанню технологій машинного навчання присвячені праці науковців та практиків А. Самуеля, Я. Міллінгтона, С. Рассела, П. Норвіга, А. Джуліані, В.-П. Берже, Е. Вкай, Ю. Гао, Х. Генрі, М. Маттар та Д. Ланжа [1–4, 8]. Проблема використання різних технологій, зокрема, машинному навчанню, в розробці комп'ютерних ігор присвячена значна кількість праць, але деякі аспекти практичного застосування сучасних технологій потребують окремої уваги.

Мета роботи полягає у дослідженні використання сучасних технологій машинного навчання в розробці комп'ютерних ігор.

Основний матеріал дослідження. Комп'ютерні ігри наділені спільними рисами з прикладним програмним забезпеченням, водночас, комп'ютерні ігри мають певні особливості і відрізняються призначенням, цільовою аудиторією, процесом розробки, ризиками [5]. Сьогодні без застосування штучного інтелекту та технологій машинного навчання не здійснюється сучасна розробка комп'ютерних ігор. Оскільки машинне навчання використовує алгоритми, які можуть визначити значимість необроблених даних, без необхідності залучення експерта для інтерпретації даних, також ці алгоритми застосовуються до інтелектуальних агентів. У комп'ютерних іграх інтелектуальні агенти використовуються як самостійна програма, що може виконувати завдання користувача протягом відведеного часового проміжку. Існує можливість використати навчання з підкріпленням для підготовки інтелектуальних агентів щодо оцінки вартості дій у середовищі гри. Після навчання інтелектуальні агенти виконують дії в грі, які мають найбільшу цінність, без необхідності їх чіткого програмування.

Дослідження технологій машинного навчання проведено на двовимірній грі. Комп'ютерна гра з використанням технологій машинного навчання реалізована як навчально-пізнавальна в жанрі космічного симулятора та наділена елементами жанру текстового квесту з розгалуженою сюжетною лінією. Жанр космічного симулятора найкраще підійшов для того, щоб наявно розкрити застосування деяких наукових принципів,

законів, познайомитися у ігровому форматі з космічними тілами та їх структурою. Окрім того, космічна тематика має значну популярність і розрахована на масового користувача.

Для програмної реалізації навчально-пізнавальної комп'ютерної гри в жанрі космічного симулятора використано ігровий двигун Unity, мову програмування C#, набір засобів розробки (Software Development Kit, SDK) Unity Machine Learning Agent toolkit (ML-Agents), інтегроване середовище розробки (Integrated Development Environment, IDE) Visual Studio 2019, графічний редактор Gimp, редактор звукових файлів Audacity.

Наразі ігровий двигун Unity є популярним та потужним середовищем для розробки комп'ютерних ігор, що містить інструментарій машинного навчання – ML-Agents.

ML-Agents – це набір засобів розробки, який дозволяє дослідникам та розробникам трансформувати ігри та імітації, створені за допомогою редактора Unity, в середовища, в яких інтелектуальні агенти можуть навчатись за допомогою глибинного навчання та навчання з підкріпленням, еволюційних стратегій або інших методів машинного навчання за допомогою використання програмного інтерфейсу Python [6].

Навчальне середовище Unity ML-Agents має три види об'єктів: агент (agent); мозок (brain); академія (academy).

Агент наділений унікальним набором станів та спостережень, має можливість здійснювати унікальні дії в оточенні та отримувати унікальні винагороди за події всередині середовища. Дії агента визначаються мозком, з яким він пов'язаний.

Мозок визначає конкретний простір стану та дій і несе відповідальність за вирішення того, які дії здійснює кожен з його пов'язаних агентів. Інструментарій ML-Agents надає можливість налаштування мозку на один з чотирьох режимів: зовнішній; внутрішній (експериментальний); гравця; евристичний.

Академія в сцені містить усі мізки в середині середовища, яке містить єдину академію, що визначає сферу навколишнього середовища з точки зору конфігурації двигуна – швидкість і якість візуалізації ігрового двигуна в режимах тренувань і виводу; зміни кадрів – кількість кроків, які потрібно пропустити між кожним агентом, щоб прийняти нове рішення; тривалість епізоду, який визначає агентів налаштованих на виконання.

Стани та спостереження за всіма агентами з мізками, які встановлені як зовнішні, збираються зовнішнім комунікатором та передаються програмним інтерфейсом Python для обробки за допомогою обраної бібліотеки ML-Agents. При встановленні декількох агентів в один мозок, є можливість вирішувати дії послідовно, відкриваючи можливість отримання переваг паралельних обчислень при їх підтримці [7].

У деяких випадках легше демонструвати поведінку, яку необхідно виконати ігровому агенту, а не намагатися навчити. Unity ML-Agents надає можливість записувати всю інформацію про стан/дію/винагороду для використання у контрольованих сценаріях навчання, таких як імітаційне навчання. При використанні імітаційного навчання, гравець має можливість наводити демонстрації того, як агент повинен поводитись у навколишньому середовищі, а потім використовувати ці демонстрації для тренування агента самостійним способом, або як перший крок у процесі навчання з підкріпленням [4].

Процес розробки комп'ютерної навчально-пізнавальної гри в жанрі космічного симулятора складається з наступних етапів:

- проектування архітектури підсистем гри: діалогова, квестова, звукова, підсистема користувацького інтерфейсу;
- проектування підсистеми штучного інтелекту;
- проектування ігрового процесу, а саме ігрових механік (gameplay), які навчають та розважають гравця;
- розробка анімацій та графічних елементів, за допомогою яких гра концентрує увагу гравця на потрібних елементах ігрового процесу;

- створення механізму збереження досягнень користувача.

Визначимо декомпозицію задач створення підсистем комп'ютерної гри, які відповідають за використання інтелектуальних агентів:

- виділення механік гри, які найбільше підходять для використання штучного інтелекту;
- опрацювання механіки таким чином, щоб машинне навчання додало логічне представлення ігровим механікам;
- створення навчального середовища для обраних механік, тобто виділення агентів, створення так званого «мозку», а саме поведінки для кожної групи агентів та академію, яка містить поведінки усіх агентів;
- навчання інтелектуальних агентів;
- збереження моделі та додання до гри;
- використання моделі, яка навчена до ігрових агентів.
- тестування ігрових механік, які використовують інтелектуальні агенти.

Виділено ігрові механіки, які мають можливість використовувати інтелектуальні агенти: «живий світ», «дрони/кораблі-розвідники», «аркадні бої».

1. «Живий світ». Ігрові персонажі реагують на різні події в ігровому світі та можуть виконувати непередбачувані дії, що здійснюють імітацію реальності подій.

2. Механіка «дрони/кораблі-розвідники». Гравець може відправити кораблі, які самостійно шукають тіла в планетній системі (планети, зірки, астероїди, станції та інші кораблі), сканувати/аналізувати їх та відправляти інформацію гравцю. Знайдена інформація зберігається у «книзі відкриттів» користувача. У гравця з'являється більше часу на взаємодію з ігровими механіками, а не їх пошук.

3. «Аркадні бої». Класична механіка двовимірного шутеру, в якому супротивники під управлінням штучного інтелекту на основі дій гравця пристосовуються для більш складної гри.

Наступним етапом розроблення конфігурації навчального середовища для обраних механік є:

- створення так званого «мозку», тобто це поведінка для кожної групи агентів (з урахуванням обраних механік створено три типи мізків);
- створення групи агентів, де одна група відповідає одному типу поведінки – мозку;
- створення академії, яка містить поведінки усіх агентів та є сполучною ланкою між навчальним середовищем та зовнішньою бібліотекою машинного навчання.

З урахуванням вищеописаних механік розроблено конфігурацію навчального середовища. Архітектура має структуру, яка представлена на рис. 1.

За замовчуванням ML-Agents SDK має скрипти з базовими функціями для сутностей агент, мозок, академія. Щоб зв'язати скрипти з ігровими об'єктами виконано наступні дії:

1. Створено новий ігровий об'єкт (game object) за допомогою редактора Unity. Додано до об'єкта скрипт, що відповідає академії.

2. Створено новий game object, додано до нього два скрипти: Behavior Parameters та Basic Agent, які виконують роль сутностей (мозок та агент відповідно). Далі збережено створений об'єкт як об'єкт-шаблон для створення групи таких об'єктів.

3. Здійснено програмування доданих скриптів додатковим функціоналом, необхідним для обраних ігрових механік.

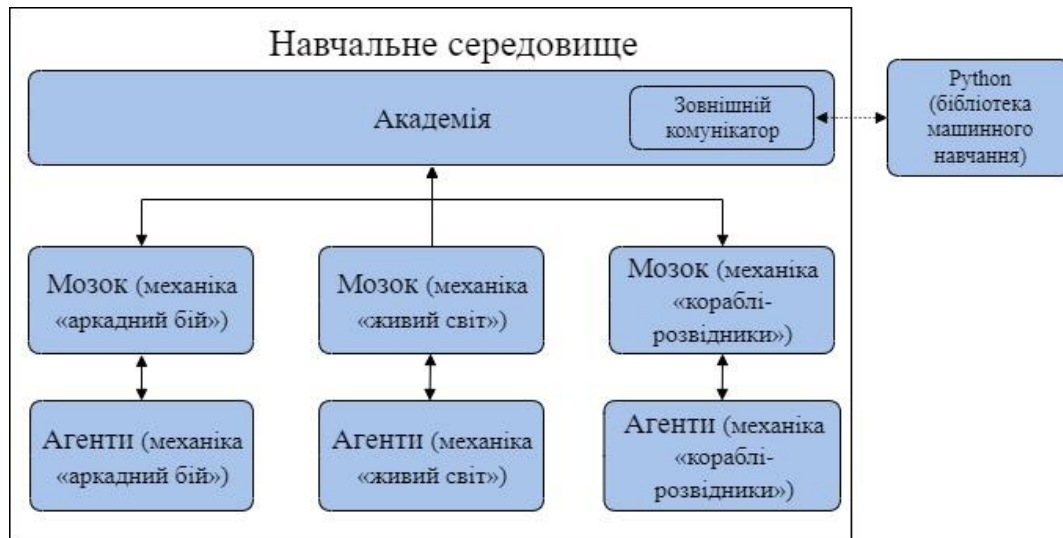


Рисунок 1 – Архітектура конфігурації навчального середовища комп'ютерної гри в жанрі космічного симулятора

Після створення навчального середовища проведено навчання інтелектуальних агентів комп'ютерної гри. Існує декілька способів навчання за допомогою ML-Agents, нами обрано метод імітаційного навчання.

На наступному етапі використано моделі, які додано до проекту та навчені і приєднані до поведінки кожного агента (використано скрипт Behavior Parameters). Після встановлення моделей агентами запускаємо гру та проводимо тестування механік, в яких присутні зазначені агенти. Навчання агентів може здійснюватися велику кількість разів з різними конфігураціями середовищ, поки не будуть отримані моделі, що підходять до обраних ігрових механік.

З огляду на навчально-пізнавальну мету комп'ютерної гри гравець має можливість отримувати нові знання та перевіряти здобуті. За цю частину гри відповідає ігрова механіка внутрішньої енциклопедії «книга відкриттів», механіка квестів та система діалогів. Розглянемо їх механіки:

1. Механіка внутрішньої енциклопедії «книга відкриттів». Книга розділена на розділи, де кожен розділ відповідає окремій планетній системі. У випадку коли гравець вперше зустрічає космічне тіло (зірку, астероїд, планету та інші) або отримує інформацію про планетну систему та космічне явище, в книзі з'являється інформація. Окремо міститься розділ з відомими астрономічними термінами, законами, описом різного роду космічних тіл, а також формулами, які необхідні для виконання квестів.

2. Механіка квестів та система діалогів. Гра містить різного типу квести: текстові, логічні, місії. За навчальний елемент відповідають текстові та логічні. Для виконання завдання гравець має відповісти на питання з різних тем астрономії, використати правильно формулу або відгадати загадку про космічні тіла. Інформація щодо поточних квестів та підказки (паролі, формули) зберігаються у журналі завдань.

Враховуючи те, що вищезазначені механіки схожі за суттю та структурою прийнято рішення розробити загальну архітектуру для них. Користувач взаємодіє з механікам через інтерфейс користувача (user interface, UI). Для реалізації використано стандартну вбудовану підсистему Unity UI. За обидві механіки відповідає одне ігрове вікно, але з двома вкладками: журнал завдань та «книга відкриттів». Програмна реалізація спільна та має визначену архітектуру, яка викладена нижче.

1. Створено користувацькі редактори Unity для введення діалогів, розділів внутрішньої енциклопедії та журналу завдань. Це необхідно для спрощення процесу наповнення гри контентом.

2. Після введення в редакторі діалогу, розділу або завдання та натиснення відповідної кнопки створений елемент зберігається в xml файлі, який має спеціально визначену структуру (різну для різних механік).

3. Модуль, за допомогою якого здійснюється інтеграція інформації, що міститься в xml файлах в текстові елементи системи UI.

4. Окремо реалізовані модулі, які відповідають за взаємодію гравця з ними.

Виконання та отримання нових знань повинно приносити гравцю задоволення, з цією метою реалізована система, яка пов'язує ігрові системи разом та використовує єдиний сюжет, а гравець постійно взаємодіє з грою через різні ігрові події та механіки. На рис. 2 схематично зображено взаємодію гравця з механіками квестів, «книги відкриттів» та діалогової системи.

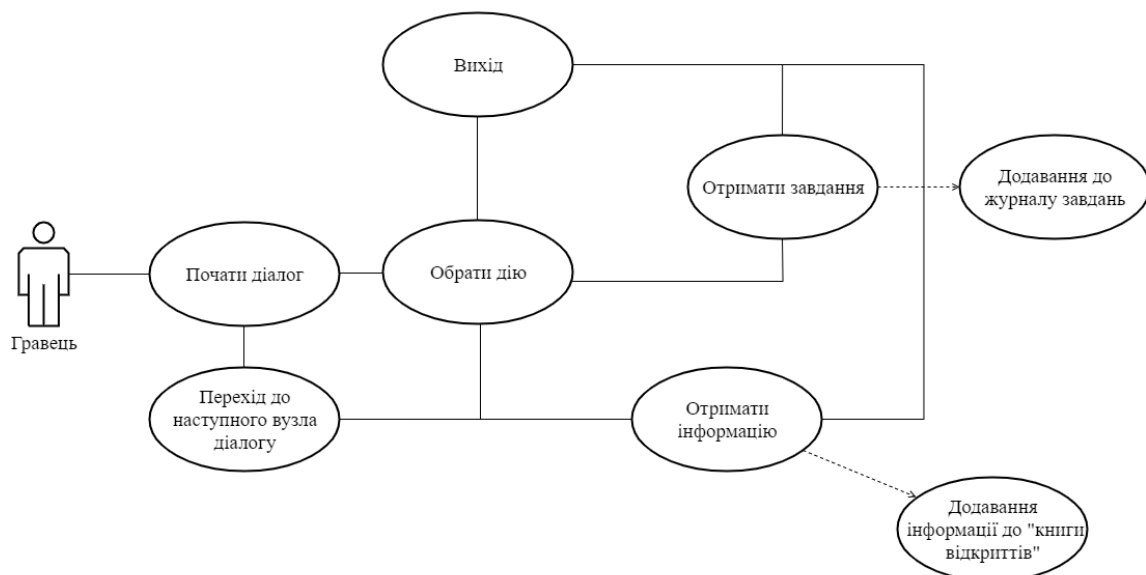


Рисунок 2 – Схематичне представлення взаємодії гравця із системою діалогів, механіками «книги відкриттів» та журналу завдань

Елементи аркадного симулятора додають різноманітність у поступовість викладення текстової та пізнавальної частини та відповідають за розважальний аспект, який є невід'ємною частиною кожної комп'ютерної гри. А елементи відкритого світу використовують моделі, що створені на основі штучного інтелекту з використанням машинного навчання, це дозволяє більш глибоко поринути в ігровий світ.

Розроблена комп'ютерна гра виконує як навчально-пізнавальну, так і розважальну функції, а механіки поєднують гру в єдине ціле і спонукають гравця пізнавати ігровий світ. Окрім того, передбачено розгортання на різних платформах, зокрема комп'ютерах з операційними системами Windows, MacOS, а також розміщення в мережі Інтернет. Зважаючи на специфіку розробленої гри, як типу програмного забезпечення та жанр комп'ютерної гри, продукт має доступні вимоги до апаратної складової персонального комп'ютера.

Проект реалізовано за допомогою методології Agile, оскільки серія ітерацій дозволила виявити особливості використання машинного навчання вже у процесі розробки та виконати синтез технологій машинного навчання та ігрового двигуна для реалізації цілісного продукту. В результаті отримано навчені та протестовані моделі інтелектуальних агентів, які впроваджені в ігрові механіки.

ВИСНОВКИ

Технологія Unity ML-Agents надає зручний інструментарій для роботи з інтелектуальними агентами, а безпосередня інтеграція в проект дозволила якісно та швидко провести тестування навчених моделей та впровадити в ігрові механіки.

Отже, розроблена комп'ютерна гра з використанням сучасних технологій машинного навчання поєднує як наукові, так і розважальні механіки, виконує навчально-пізнавальну функцію та є цілісним продуктом. Основними перевагами використання технологій машинного навчання в розробці комп'ютерних ігор є гнучка реалізація ігрових механік для досягнення поставлених завдань.

Список використаних джерел:

1. *Samuel, A.* Some Studies in Machine Learning Using the Game of Checkers / *A. Samuel* - IBM Journal of Research and Development 3 (3): p. 210–229
2. *Millington, Ian.* Artificial intelligence for games / *Ian Millington, John Funge.* – 2nd ed. – 2009. – p. 895
3. *Russell, S.* Artificial Intelligence: A Modern Approach (3rd Edition) / *Russell S., Norvig P.* – Pearson Education Limited, 2009 – 1152 p.
4. *Juliani A.* Unity: A General Platform for Intelligent Agents / *A. Juliani, V.-P. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, D. Lange.* – arXiv preprint arXiv:1809.02627 – 2009 – 18 p.
5. *Фурсова Н.А.* Особливості розробки мережевої комп'ютерної гри в жанрі Multiplayer First-Person Shooter / *Н.А. Фурсова, О.С. Козак* // Наука і виробництво: ДВНЗ «ПДТУ». – Маріуполь – 2019. – № 1 (20) – С. 200 – 206
6. Unity. Unity ML-Agents Toolkit Documentation [Електронний ресурс] – Режим доступу: <https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Readme.md>
7. *Фурсова Н.А.* Навчання ігрових агентів з використанням Unity Machine Learning Agents/ *Н.А. Фурсова, Д.М. Кривицький* // Концептуальні напрямки розвитку наукових знань (частина II): матер. II Міжнар. наук.-практ. конф. м. Київ, 30-31 грудня 2019 року. – Київ : МЦНІД, 2019. – С. 40-41
8. *Juliani A.* Introducing: Unity Machine Learning Agents Toolkit/ *Arthur Juliani* [Електронний ресурс]: – Режим доступу: <https://blogs.unity3d.com/2017/09/19/introducing-unity-machine-learning-agents/>

Фурсова Н.А., Кривицький Д.М.

ТЕХНОЛОГИИ МАШИННОГО ОБУЧЕНИЯ В РАЗРАБОТКЕ КОМПЬЮТЕРНЫХ ИГР

В статье исследовано использование современных технологий машинного обучения в разработке компьютерных игр на примере двумерной учебно-познавательной игры в жанре космического симулятора. Определены возможности инструментария Unity ML-Agent и его

объектов. Раскрыты основные этапы разработки компьютерной игры, проведена декомпозиция задач использования интеллектуальных агентов. Выбраны игровые механики, разработана архитектура конфигурации учебной среды. Представлена последовательность взаимодействия игрока с игрой и интеллектуальными агентами во время игровой сессии.

Ключевые слова: машинное обучение, компьютерная игра, интеллектуальный агент, игровой движок, ML-Agents.

Fursova N.A., Krivitsky D.M.

MACHINE LEARNING TECHNIQUES IN THE DEVELOPMENT OF COMPUTER GAMES

In the article, usage of modern machine learning technologies in the development of computer games was investigated on the example of two-dimensional cognitive and training space flight simulation game. The capabilities of the Unity ML-Agent were identified. The basic stages of computer game development are revealed: design of game subsystem architecture (dialog, quest, sound, user interface subsystem); design of artificial intelligence subsystem; designing a gameplay, namely gameplay mechanics that teach and entertain a player; development of animations and graphic elements by which the game concentrates the player's attention on the necessary elements of the gameplay; creating a mechanism for storing user achievements. The use of intelligent agents consist of the following phases: selection of game mechanics, which are most suitable for the use of artificial intelligence; elaboration of mechanics in such a way that machine learning gives a logical representation to game mechanics; setting training environment for selected mechanics, selection of agents, creation of a Brain, namely behavior for each group of agents and an Academy that contains the behavior of all agents; training of intelligent agents; saving the models and adding them to the game; using models that are trained in game agents. There are most suitable game mechanics to use intelligent agents: "living world", "drones/reconnaissance ships", "arcade battles". Game characters from "Living world" mechanic respond to different game events and perform unpredictable actions that simulate reality of events. Player can send ships to research planetary systems (planets, stars, asteroids, stations, and other ships), to scan/analyze them and send information to the player. It is all about "Drones" mechanic. Opponents of "Arcade battles" mechanic are under the control of artificial intelligence, based on the actions of the player and adapt to a more difficult game. The architecture of the training environment configuration was developed. There are sequence of interaction of the player with the game and intelligent agents during game session.

Keywords: machine learning, computer game, intelligent agent, game engine, ML-Agents.

Рекомендовано до публікації: д-р техн. наук, проф. О.Л. Ляхов

Стаття надійшла _____.