

УДК: 004.896

СПОСОБИ КЕРУВАННЯ ПОВЕДІНКОЮ НЕІГРОВОГО ПЕРСОНАЖУ В RPG-ГРІ

Сеніва К. Р.

e-mail: katelyna.seniva@gmail.com

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Україна, Київ.

Сучасні комп'ютерні ігри – яскравий приклад бурхливого розвитку інформаційних аудіовізуальних технологій XXI століття. Неймовірно популярні зараз, комп'ютерні ігри в своєму розвитку пройшли довгий шлях від примітивних аркад до повноцінних віртуальних світів, для повноцінного освоєння яких потрібно не один місяць реального часу. Для більшої зацікавленості гравців розробники все частіше застосовують технології штучного інтелекту. Тому в даній статті розглянута задача керування поведінкою неігрового персонажу в комп'ютерній рольовій грі. Визначено, які способи можуть бути застосовані для вирішення цієї задачі. Розглянуто відомі способи реалізації ігрового штучного інтелекту.

Ключові слова: ігровий штучний інтелект, машинне навчання, неігровий персонаж, корисність, прийняття рішень.

Сенива К. Р. Способы управления поведением неигрового персонажа в RPG-игре / Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», Украина, Киев

Современные компьютерные игры - яркий пример бурного развития информационных аудиовизуальных технологий XXI века. Невероятно популярные сейчас, компьютерные игры в своем

развитии прошли долгий путь от примитивных аркад до полноценных виртуальных миров, для освоения которых потребуется не один месяц реального времени. Для большей заинтересованности игроков разработчики все чаще применяют технологии искусственного интеллекта. Поэтому в данной статье рассмотрена задача управления поведением неигрового персонажа в компьютерной ролевой игре. Определено, какие способы могут быть применены для решения этой задачи. Рассмотрены известные способы реализации игрового искусственного интеллекта.

Ключевые слова: игровой искусственный интеллект, машинное обучение, неигровой персонаж, полезность, принятие решений.

K. R. Seniva Ways to Control NPC Behavior in an RPG Game / National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, Kiev

Modern computer games are a clear example of the rapid development of information and audiovisual technologies of the XXI century. Incredibly popular now, computer games in their development have come a long way from primitive arcades to full-fledged virtual worlds, for the full development of which requires more than one month of real time. To increase the interest of players, developers are increasingly using artificial intelligence technologies. Therefore, this article discusses the problem of controlling the behavior of a non-player character in a computer role-playing game. It is determined which methods can be used to solve this problem. Known methods of realization of game artificial intelligence are considered.

Key words: game artificial intelligence, machine learning, non - player character, usefulness, decision making.

Вступ. Комп'ютерні ігри вже давно ввійшли у життя користувачів персональних комп'ютерів та є одним із способів проведення вільного часу. Вони є не тільки розвагами для дітей та підлітків, а ще й приносять користь для дорослих, які можуть зняти стрес після важкого робочого дня, відволіктися від буденності та розвиватися, придумуючи нові тактики та стратегії. Кожному гравцю щоразу хочеться нових вражень, тому з року в рік комп'ютерні ігри стають все складнішими, цікавішими та захоплюючими. Для того, щоб задовольнити потреби користувачів, розробникам необхідно використовувати технології ігрового штучного інтелекту.

Мета статті. Метою цієї роботи є обґрунтування доцільності використання штучного інтелекту в комп'ютерних іграх та огляд способів реалізації ігрового штучного інтелекту для керування поведінкою неігрового персонажа в RPG-грі (Role-Playing Game).

Виклад основного матеріалу. Ігровий штучний інтелект – набір програмних методик, які використовуються у відеоіграх для створення ілюзії інтелекту в поведінці персонажів, керованих комп'ютером[1]. Ігровий ШІ дещо відрізняється від традиційних уявлень про штучний інтелект, оскільки в іграх перед ШІ не стоїть задача створення розумної машини, максимально наближеної до людини, яка буде взаємодіяти з оточуючими об'єктами та проявляти емоції. Для ігрового ШІ немає необхідності навчатися за рамками ігрового середовища та процесу, достатньо лише зімітувати правдоподібну поведінку персонажа, щоб зробити гру цікавішою.

Для створення ігрового ШІ використовуються різні методи, наприклад:

- Дерево прийняття рішень;
- Скінченний автомат;
- Utility-based system (система, заснована на корисності);

- Машинне навчання.

Дерево прийняття рішень – засіб підтримки прийняття рішень. Структура дерева містить такі елементи як: «листя» та «гілки». Кожна частина дерева називається вузол, ШІ використовує теорію графів для опису таких структур. Є два типи вузлів:

- Вузли прийняття рішень: на основі перевірки певної умови виконується вибір між двома варіантами, що представлені у вигляді окремого вузла.
- Кінцеві вузли: дії, які потрібно виконати; кінцеве рішення[2].

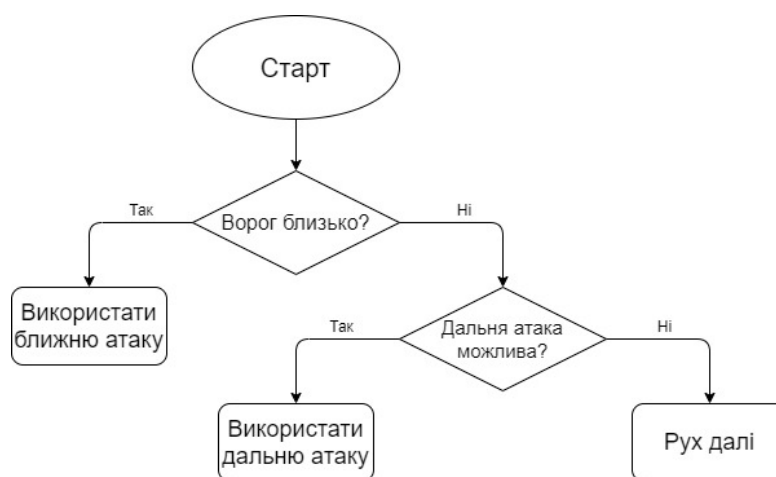


Рисунок 1. Приклад дерева прийняття рішень

Алгоритм починається з першого вузла, він або приймає рішення до якого дочірнього вузла необхідно перейти, або виконує дію, що зберігається у вузлі і завершується.

Такі дерева корисні, коли вони будуються автоматично на основі великого набору прикладів, це робить їх ефективним та високопродуктивним інструментом для класифікації ситуацій на основі отриманих даних.

Скінченний автомат – це особливий різновид автомата-абстракції, що використовується для опису шляху зміни стану об'єкта в залежності від поточного стану та інформації отриманої ззовні. Тобто це спосіб сказати, що наш неігровий персонаж знаходиться в одному із можливих станів та може переходити з одного стану в

інший. Особливість такого автомату є скінченність множини можливих станів[3].

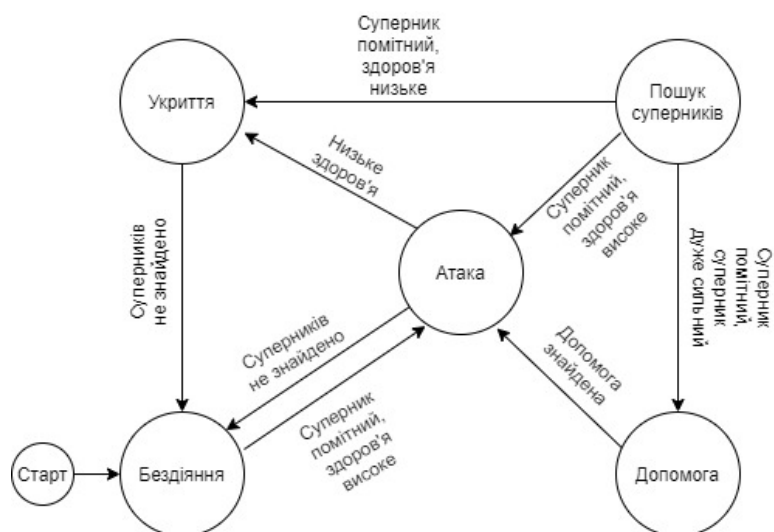


Рисунок 2. Приклад скінченного автомату

Діаграма наведена на рисунку 2 відображає суть прийняття рішень для неігрового персонажа на основі поточної ситуації. Стрілка показує перехід між станами, якщо умова біля неї істина. Щоразу під час оновлення стану проходить перевірка поточного стану персонажа, списку станів переходів та, якщо умова виконана, виконується перехід у новий стан. Постійна неперервна перевірка умов переходу є доволі «дороговартісною», наприклад, якщо для перевірки потрібно виконувати багато обчислень, це займе багато часу процесора.

Замість того, щоб мати набір виборів чи дерево можливих рішень, краще визначити всі дії та вибрати найбільш потрібну в даний момент. У цьому може допомогти Utility-based system, це система, в якій у персонажа є певний набір дій і він сам обирає яку потрібно виконати в залежності від відносної корисності кожної дії. Корисність – це міра того на скільки та чи інша дія важлива для персонажа та чи бажано її використовувати в певний момент[4]. Основна увага приділяється визначенню причин чому певна дія (поведінка) буде корисною. Розраховану корисність дії на основі поточного

середовища персонаж може перевірити та обрати найбільш підходящий інший стан в будь-який момент часу.

Спочатку система призначає довільний діапазон корисності. У кожній дії є набір параметрів, які впливають на значення корисності. Переходи між діями неоднозначні, будь-яка дія може слідувати за будь-якою іншою. Пріоритети дій залежать від значення корисності. Використовуючи таку систему, слід дбати про стандартизацію: необхідно, щоб усі показники оцінки корисності використовували однакові або подібні умови, а визначення, обробка десятків, сотень форм поведінки має вивантажитися з дизайнера та вести виконання у самій системі.

Основою штучного інтелекту може бути нейронна мережа – математична модель та її програмне або апаратне втілення, побудована за принципом організації та функціонування біологічних нейронних мереж – мереж нервових клітин живого організму. Це мережа простих елементів – нейронів, які отримують вхід, змінюють свій внутрішній стан відповідно до цього входу і формують вихід. Можливість навчання – це одне із головних переваг нейронних мереж перед традиційними алгоритмами.

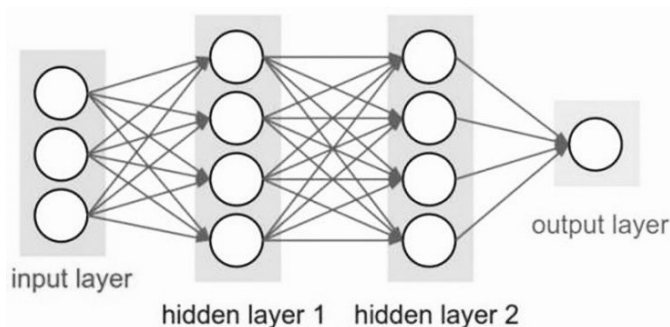


Рисунок 3. Проста схема нейронної мережі

Машинне навчання – це клас методів штучного інтелекту, характерною ознакою якого є не прямий розв’язок задачі, а навчання в процесі застосування рішень множини подібних задач. Для побудови таких методів використовуються засоби математичної

статистики, чисельних методів, математичного аналізу, методів оптимізації, теорії ймовірностей, теорії графів, різні техніки роботи з даними в цифровій формі[5]. Існує багато способів навчання: навчання з вчителем, навчання без вчителя, навчання з підкріпленням, активне навчання та інші. Для того, щоб керувати поведінкою неігрового персонажа найбільш підходить навчання з підкріпленням, в ході якого він навчається взаємодіючи з певним середовищем. Персонаж впливає на середовище, а середовище на персонажа, про таку систему можна сказати, що вона має зворотній зв'язок. При такому підході можна використати метод Q -навчання. На основі отриманого від середовища винагороди персонаж формує функцію корисності Q , що дає йому можливість не випадково обирати стратегію, а враховувати досвід попередньої взаємодії з середовищем.

Алгоритм Q -навчання:

1. Ініціалізація.

1.1. *for each s and d do $Q[s, d] = RND$* – ініціалізуємо функцію корисності Q від дії d в ситуації s як випадкову для будь-яких вхідних даних

2. Спостерігаємо.

2.1. $s_1 = s$ – запам'ятати попередній стан

2.2. $d_1 = d$ – запам'ятати попередню дію

2.3. $s = CURRENT_STATE$ – отримати поточний стан

2.4. $r = GET_REWARD$ – отримати винагороду за попередню дію

3. Оновлення корисності.

3.1. $Q[s_1, d_1] = Q[s_1, d_1] + LF * (r + DF * MAX(Q, s) - Q[s_1, d_1])$

, де LF – фактор навчання, чим він більше, тим сильніше персонаж довіряє новій інформації, а DF – фактор

дисконтування, чим він менший, тим менше персонаж замислюється над вигодою від майбутніх дій.

4. Вибір дії.

4.1. $d = \text{MAX}Q$, де $\text{MAX}Q$ – порівняння корисності всіх дій

5. Повернутись до пункту 2.

Таким чином можна навчити модель в умовах оточуючого середовища.

Висновки. У даній роботі розглянуто застосування різних способів реалізації ігрового штучного інтелекту для керування поведінкою неігрового персонажа в рольовій комп'ютерній грі. На мою думку, найбільш перспективним є розвиток машинного навчання в якості штучного інтелекту, оскільки цей спосіб хоч і важкий в реалізації та потребує багато ресурсів, проте такий після навчання може приймати стратегічно правильні рішення у будь-яких ігрових ситуаціях, тоді коли в таких методах, як дерево прийняття рішень, скінченний автомат велику роль грає людський фактор: людина не може передбачити все, тому може скластись ситуація, коли розробник пропустив якусь умову чи ситуацію та ШІ не знатиме як діяти в такому середовищі. Методи машинного навчання дозволяють поліпшити «поведінку» керованих комп'ютером неігрових персонажів, полегшуючи автоматизовану генерацію і відбір моделей поведінки, тим самим розширюючи можливості цифрового ігрового штучного інтелекту і надаючи можливість створювати більш захоплюючі і цікаві ігрові враження.

Література:

1. Game artificial intelligence.

<https://en.wikipedia.org/wiki/Artificial_intelligence_in_video_games>

(2020, квітень, 01).

2. Как создать игровой ИИ
<<https://habr.com/ru/company/pixononic/blog/428892/>> (2018, листопад, 06).
3. Finite-state machine < https://en.wikipedia.org/wiki/Finite-state_machine> (2019, листопад, 09).
4. Graham, David. An Introduction to Utility Theory
<http://www.gameapro.com/GameAIPro/GameAIPro_Chapter09_An_Introduction_to_Utility_Theory.pdf> (2013, вересень).
5. Машинное обучение в видеоиграх.
<https://ru.qaz.wiki/wiki/Machine_learning_in_video_games> (2021, січень, 08).

References:

1. Game artificial intelligence. Retrieved from https://en.wikipedia.org/wiki/Artificial_intelligence_in_video_games [in English]. (2020, April, 01).
2. Kak sozdat yhrovoi YY [How to create game AI]. Retrieved from <https://habr.com/ru/company/pixononic/blog/428892/> [in Russian]. (2018, November, 06).
3. Finite-state machine. Retrieved from https://en.wikipedia.org/wiki/Finite-state_machine [in English]. (2019, November, 09).
4. Graham, David. An Introduction to Utility Theory. Retrieved from http://www.gameapro.com/GameAIPro/GameAIPro_Chapter09_An_Introduction_to_Utility_Theory.pdf [in English] (2013, September).
5. Mashynnoe obuchenye v vydeoyhrakh [Machine learning in video games]. Retrieved from https://ru.qaz.wiki/wiki/Machine_learning_in_video_games [in Russian] (2021, January, 08).