

Розроблено систему автоматичного управління температури купола повітрянагрівача з можливістю регулювання вмісту кисню в повітрі горіння. Використання запропонованої системи дасть можливість більш гнучко регулювати температуру купола за рахунок зміни вмісту кисню в повітрі горіння і витрати доменного газу.

Ключові слова: *повітрянагрівач, доменний газ, кисень, збагачення киснем, температура купола, температура дуття, управління*

Koifman O.O., Korol M.O., Simkin O.I.

AUTOMATED CONTROL SYSTEM FOR HEATING THE HOT BLAST STOVE CHECKERWORK WITH THE OPTION OF ADJUSTING THE OXYGEN CONTENT IN THE COMBUSTION AIR

The temperature of the blast is largely determined by the temperature under the hot blast stove dome. The effect of the oxygen content in the enriched air, that goes for burning in the burner of a hot blast stove of a blast furnace, on the increase in the temperature of the dome, and, consequently, the increase in the temperature of the hot blast, was studied. The basic calculation formulas of combustion during enrichment in general form were given. The combustion temperature of the gas increases with an increase in the oxygen content in the enriched air and the amount of combustion products decreases, which will directly affect the heating rate of the dome and checkerwork. To compensate for the decrease in heat transfer in the nozzle, it is necessary to increase the flow rate of blast furnace gas. With an increase in the oxygen content in the combustion air from 21% to 50%, an increase in the calorimetric temperature of the combustion of the blast furnace gas from 1451 °C to 1821 °C was observed, and the combustion temperature - from 1306 to 1639 °C.

Using the archive database for 7 months of the automatic control system of the hot blast stoves block of the iron and steel plant, the calculation of the basic operational indicators of the block with a higher oxygen content in the combustion air was carried out. An increase in oxygen content from 21% to 30% can increase the blast temperature by 100 °C, which will reduce the coke consumption and increase the productivity of the blast furnace.

A system for automatic control of the temperature of the hot blast stove dome with the option to control the oxygen content in the combustion air has been developed. Using the proposed system will make it possible to more flexibly control the temperature of the dome by changing the oxygen content of the combustion air and the flow of blast furnace gas.

Keywords: *hot blast stove, blast furnace gas, oxygen, oxygen enrichment, dome temperature, blast temperature, control*

Рецензент доц., к.т.н. Добровольська Л.А.

Стаття поступила 14.11.2019

УДК № 004.4:004.8

Кривенко О. В., Загірний М. М.

ДОСЛІДЖЕННЯ ТА МОДЕЛЮВАННЯ ІГРОВОГО ПРОЦЕСУ НА БАЗІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Методи машинного інтелекту з навчанням привносять свою специфіку до створення і налагодження ігрової системи.

Одна з головних проблем - це необхідність передбачення усього набору вхідних ситуацій і можливих відповідей на моменті проектування і неможливість розширення їх списку без перенавчання. Це призводить до звуження можливості їх використання у реальних ігрових системах.

Об'єктом дослідження є процеси створення і навчання ігрових агентів на основі еволюційних підходів штучного інтелекту.

Мета роботи - розробка і обґрунтування формальної моделі ігрового агента, заснованої на методах машинного навчання, програмна реалізація ігрового процесу з застосуванням нейронних мереж та еволюційних методів оптимізації при множинних генераціях популяції ігрових агентів.

Для здійснення поставленої мети досліджена теоретична база, існуюча дослідження та розробки в галузі; спроектовано ігровий сценарій, визначені основні агенти, їх основні можливості та очікувана поведінка; реалізоване навчання ігрових агентів різними типами нейронних мереж; виконане тестування розробок, оцінка якості поведінки й прийняття рішень штучним інтелектом з використанням нейронних мереж; проведений порівняльний аналіз різних типів нейронних мереж, запропонування рекомендацій щодо їх використання для заданих умов.

Для відображення штучної нейронної мережі використані шаблонні компоненти слою, нейрону і мосту.

Створений програмний модуль дозволить ігровим агентам, побудованим з використанням різних нейронних мереж, суперничати один з іншим (а не з людиною, як у звичайному режимі гри), дозволить виявити більш підготовленого з них.

Наукова новизна одержаних результатів полягає у тому, що формалізовано модель ігрового агента, в основі якої методи машинного навчання.

Отримані в роботі результати можуть бути використані при розробці відеоігор, побудованих на базі штучного інтелекту ігрових агентів, що гуртуються на методах машинного навчання, а також в рамках інших наукових досліджень.

Ключові слова: *Unity, машинне навчання, машинний інтелект, нейронні мережі, генетичні алгоритми, ігровий агент, відеогра, програмний модуль*

Постановка проблеми. Методи машинного інтелекту з навчанням привносять свою специфіку до створення і налагодження ігрової системи.

Одна з головних проблем - це необхідність передбачення усього набору вхідних ситуацій і можливих відповідей на моменті проектування і неможливість розширення їх списку без перенавчання. Це призводить до звуження можливості їх використання у реальних ігрових системах.

До категорій машинного навчання підпадає достатньо велика кількість концепцій, підходів, методів, алгоритмів, що застосовуються для автоматичного отримання будь-яких нетривіальних висновків на основі будь-яких емпіричних даних [1].

Одним з основних критеріїв класифікації задач машинного навчання є спосіб машинного навчання. На даний момент не існує сталої класифікації методів машинного навчання, але можна виділити за характерними рисами: навчання з вчителем, навчання без вчителя та змішані підходи навчання.

З іншої сторони, системи машинного навчання можна охарактеризувати з точки зору вирішуваних ними задач і бажаного виходу навченої системи. З такої точки зору можна виділити декілька наступних категорій: задача класифікації, задача регресії, задача кластеризації, задача оцінки щільності, задача зниження розмірності.

Окрім цих класичних задач машинного навчання існує велика кількість інших задач, які пов'язані з машинним навчанням: задачі керування, задачі побудови рангової залежності, тощо.

Методи машинного навчання мають достатньо широкий спектр застосування і відеоігри не виключення. Відповідні методи можуть застосовуватися, як для побудови самого ігрового процесу, так і для вирішення завдань пов'язаних з ігровим процесом. З точки зору ігрового процесу відповідні алгоритми можуть бути застосовані у реалізації штучного інтелекту ігрових опонентів і механік, а з точки зору завдань пов'язаних з ігровим процесом – анімації руху, постобробки, тощо.

Найбільш широкою областю для застосування методів машинного навчання виступає саме створення реалістичних ігрових супротивників. В цьому випадку відповідні методи можуть, як в прямому вигляді контролювати поведінку ігрових агентів, так і в більш складному випадку відповідати за вибір стратегії поведінки відповідно до ігрових обставин.

Можливість використання певних методів машинного навчання прямо залежить від ігрової концепції, які реалізує відеогра. Здебільшого основними характерними рисами, які обумовлюють можливість використання певних методів це наявність шуму у даних і виникнення апріорних ситуацій. Одним з найбільш універсальних засобів, який може використовуватися в даних ситуаціях, є нейронні мережі.

Аналіз останніх досліджень і публікацій. За тематикою роботи було досліджене питання історичного ракурсу та базових концепцій застосування [2 - 3]. Більш детально концепції використання розглядаються у роботах [4 - 6]. Хоча в даних матеріалах розглядаються концепції, принципи, підходи до створення штучного інтелекту на базі машинного навчання, але не проводиться фактична оцінка складності розробки і налагодження агентів.

З проведеного аналізу отримано, що питання використання методів машинного навчання, а саме, питання реалізації штучного інтелекту ігрового агента, розглядається не тільки науковою спільнотою, але й ще спільнотою програмістів в цілому.

Проаналізувавши роботи за тематикою можна виявити, що найбільш часто вживаною технологією реалізації штучного інтелекту ігрового агента є нейронні мережі, а найчастішим типом виступають глибинні нейронні мережі.

Навчання нейронної мережі здебільшого здійснюється за допомогою навчання з підкріпленням. Основною причиною поширення методу навчання з підкріпленням пояснюється відсутністю необхідності створення навчальних вибірок, що приводить до пришвидшення та зниження вартості розробки.

Серед методів навчання, які часто використовуються, найбільше представлено навчання за допомогою генетичного алгоритму. На другому місці за поширеністю Q-навчання.

В жодній з розглянутих робіт не порівнювались особливості реалізації ігрового штучного інтелекту на базі класичних методів і методів на базі машинного навчання і інтелекту. Також в роботах не проводився аналіз доцільності використання методів машинного навчання та особливості реалізації ігрових систем на базі даних методів.

Завдання дослідження. Метою роботи є розробка і обґрунтування формальної моделі ігрового агента, заснованої на методах машинного навчання, програмна реалізація ігрового процесу з застосуванням нейронних мереж та еволюційних методів оптимізації при множинних генераціях популяцій ігрових агентів.

Для здійснення поставленої мети вирішувались ряд задач:

- а) дослідження теоретичної бази, існуючих досліджень й розробок в галузі;
- б) проектування ігрового сценарію, визначення основних агентів, їх основних можливостей та очікуваної поведінки;

- в) реалізація навчання ігрових агентів різними типами нейронних мереж;
- г) тестування розробок, оцінка якості поведінки й прийняття рішень штучним інтелектом з використанням нейронних мереж;
- д) порівняльний аналіз різних типів нейронних мереж, запропонування рекомендацій щодо їх використання для заданих умов.

Основний матеріал дослідження.

Існує велика кількість реалізацій систем програмної логіки нейронних мереж. У випадку ігрового рушія Unity вони, як правило, зовнішні. Зовнішня логіка отримує від ігрового рушія певні сигнали, а усі обрахунки нейронної мережі в середині ігрового рушія є чорною скринькою з тієї точки зору, що отримання результатів проміжних обрахунків не є можливим. Оскільки необхідна наглядна демонстрація процесу навчання ігрової системи, то відповідна така логіка повинна бути реалізована в середині ігрового рушія.

Логіка обчислення нейронної мережі, питання генерації ваги мостів, питання відображення нейронної мережі реалізуються трьома окремими скриптами NNAgent, NNMath і NNView, відповідно. Скрипти NNAgent і NNMath повністю забезпечують логіку функціонування нейронної мережі. Під час виконання обчислень скрипт NNAgent зберігає результуючі значення нейронів для того, щоби запобігти виконанню зайвих обчислень під час відображення за допомогою скрипту NNView.

Скрипт NNView виконаний в тенденціях і підходах характерних до організації елементів користувацького інтерфейсу Unity. Базова ієрархія компоненту інтерфейсу наведена на рис. 1, а приклад відображення нейронної мережі наведено на рис. 2. Відображення складається з шаблонних компонентів, які лежать в відповідному місці в ієрархії елемента.

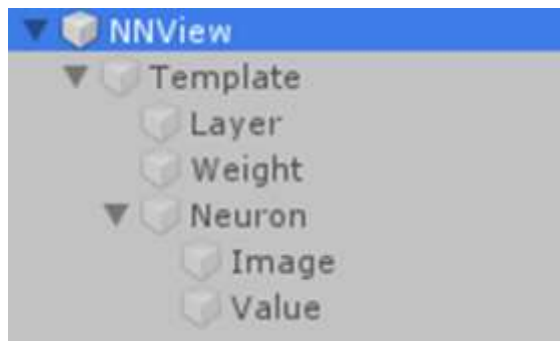


Рисунок 1 – Ієрархія компоненту NNView

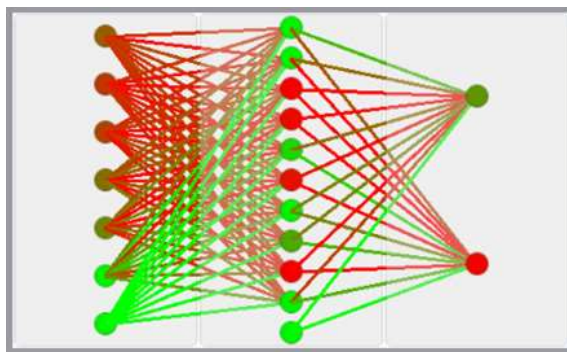


Рисунок 2 – Відображення штучної мережі на базі скрипту NNView

Головним каталогом проекту Unity є каталог Assets і він повинний включати всі активи, які використовуються проектом. Вирішальний аспект формування файлової

структури (рис. 3) полягає в тому, що матеріали проекту було умовно поділено на дві глобальні категорії, які було вирішено розмістити в окремі підкаталоги у каталозі Assets.

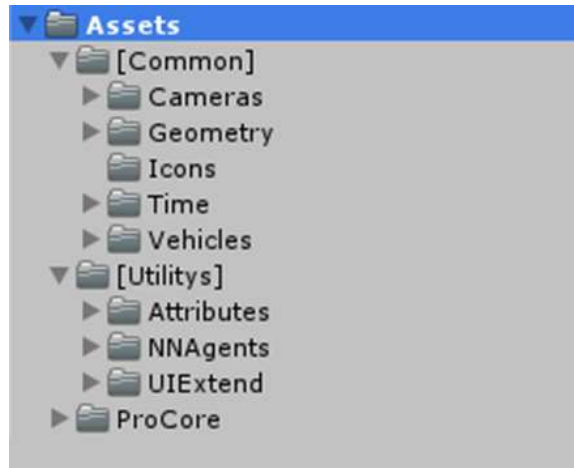


Рисунок 3 – Файлова структура проекту

Перша категорія представляє собою код реалізації ігрової логіки самого проекту, відповідний код розміщується у підкаталог [Common], а друга категорія представляє зовнішні для логіки проекту скрипти і вони розміщується у підкаталог [Utilities]. Також у каталозі Asset присутній підкаталог ProCore, у якому знаходяться матеріали деяких пакетів системи і використовуваних активів, що розширюють можливості редактору Unity.

Структура розглянутих підкаталогів формувалась в подальшому процесі розробки за необхідністю розмежування певних компонентів системи за їх призначенням чи іншими ознаками.

Для відображення штучної нейронної мережі використані шаблонні компоненти слою, нейрону і мосту. Основний компонент представляє собою горизонтальну групу з автоматичним розміщенням елементів, компонент слою – вертикальну групу також з автоматичним розміщенням компонентів, нейрон представляє собою компонент слою, а мости розміщуються до нейронів, з яких вони виходять.

З точки зору роботи з побудови ігрової системи і здійснення її налагодження ігровий персонаж, заснований на базі агента нейронної мережі, представляє собою чорну скриньку. Але це не чорна скринька в сенсі розуміння внутрішніх процесів роботи, оскільки внутрішній алгоритм в даному випадку цілком відомий. Це чорна скринька в сенсі розуміння залежності, яка була визначена під час навчання.

Відповідно до цього внесення невеликих поправок до визначеної залежності не є можливим. Хоча і існує фізична можливість ручного внесення виправлення ваги певних мостів нейронних мереж. Але передбачити заздалегідь до яких наслідків приведене відповідне виправлення зазвичай представляється не можливим. В випадку не повного розуміння результатів, до яких приведе внесення такого виправлення, необхідно проведення тестування виправленого агента у всіх можливих сценаріях.

Фактично ми приходимо до того, що у випадку необхідності внесення правок до агентів на базі нейронних мереж необхідно здійснювати повторно навчальний процес з деякими корективами, які повинні привести систему до бажаного результату. Фактично в багатьох випадках це робить процес корегування інтелекту достатньо трудомістким і займає багато часу.

Тут можна навести у порівняння ситуацію, яка відбувається при необхідності зміни поведінки при використанні класичних підходів. На відміну від агентів, заснованих на

нейронних мережах, класичні підходи дозволяють дуже легко коректувати логіку взаємодії і власні параметри, а розширення логіки зазвичай не представляє нічого важкого і здійснюється доданням нових методів.

Фактично кожен з підходів до створення ігрових агентів має свої плюси і мінуси, але в цілому обидва підходи мають право на існування. Підхід, заснований на базі агентів нейронних мереж, має своє місце і навіть в деяких випадках може спростувати процес розробки і дозволяти виходити на новий якісний рівень ігрового інтелекту. Але, прибігаючи до використання, обов'язково необхідно враховувати специфіку.

Даний підхід дуже гарно може бути застосований у ігрових персонажах, взаємодії і параметри яких точно не будуть змінюватися як під час самої гри, так і бажано під час створення ігрового продукту. Тоді один раз правильно навчений ігровий агент буде реалізовувати логіку певного персонажу, що в багатьох випадках може скоротити час на написання великої кількості коду, який реалізує часто доволі просту логіку персонажу. І у багатьох випадках набагато більш якісніше, ніж в вигляді реалізованого класичним методом.

Формування нового навчального покоління представлено на рис. 4. Мутація відбувається на базі модифікованої функції безпечної мутації. Кросовер здійснюється на базі рівномірного принципу.

Тестування та аналіз отриманих проміжних результатів проводиться протягом всього процесу розробки з метою коригування підходів до побудови алгоритмів та простеження ступеня навчання інтелекту.

Створений програмний модуль дозволить ігровим агентам, побудованим з використанням різних нейронних мереж, суперничати один з іншим (а не з людиною, як у звичайному режимі гри), дозволить виявити більш підготовленого з них.



Рисунок 4 – Файлова структура проекту

Нейронні мережі, залежно від типу задачі, що вирішується, мають певні чисельні параметри, за якими можна проводити їх порівняння. Так, швидкість збіжності (швидкість навчання) є однією з головних характеристик мережі. Вона особливо критична, коли мова йде про автономне навчання ігрового інтелекту в процесі використання, тобто використовуючи знання, отримані динамічно в процесі гри. Коли кількість агентів зростає, алгоритм пошуку рішення може потребувати досить часу, що теж треба мати на увазі під час проектування мережі. У покроковій грі користувач не погодиться довго чекати на хід комп'ютера, а в режимі реального часу повільний алгоритм просто не дозволить штучному інтелекту суперничати з людиною.

Іншим параметром, який легко оцінити, є отримувана похибка навчання (найчастіше це середньоквадратична помилка, але можуть бути й інші варіанти). Можна встановити потрібну похибку, до якої слід продовжувати навчання – для певних типів мереж й алгоритмів доведено їх збіжність, але для нестандартних реалізацій це не гарантовано. Програма може «застрягти» на локальному мінімумі й не забезпечити бажаного результату. Для порівняння якості навчання різних нейронних мереж за певний час (або кількість ітерацій/епох навчання) можна обмежитись заданим періодом й порівняти похибку після його завершення.

Ще один критерій порівняння – обсяг тренувальних даних, необхідних для навчання мережі до певного рівня похибки. Насправді не для кожної задачі нам може бути доступний

великий обсяг даних для навчання. Наприклад, якщо в шаховій програмі можна аналізувати безліч проведених іншими програмами та людьми партій (існує досить відкритих баз даних з такими даними), то в менш розповсюдженій або новій грі таке неможливо. У таких випадках також можна використати гру «сама проти себе», коли штучний інтелект буде робити ходи як за себе, так і за людину, аналізуючи результат та змінюючи поведінку. Але скільки таких наборів даних йому треба проаналізувати, щоб досягти майстерності, також слід визначити.

Точно можна оцінити необхідні для навчання й функціонування різних програмних розробок ресурси комп'ютера (зокрема, оперативної пам'яті).

Окрім наведених, можуть бути використані і інші критерії порівняння. Зокрема, кількість переміг над суперниками – аналогічними ігровими агентами з іншими нейронними мережами. Менш об'єктивний, але також потребує дослідження параметр – кількість перемог над людиною-суперником.

ВИСНОВКИ

В роботі було проведено порівняльний аналіз сучасних методів та алгоритмів машинного навчання. На базі проведеного аналізу відбулось обґрунтування і розробка формальної моделі ігрового агента на базі методів машинного навчання, визначено підхід до навчання розроблюваного агента. В роботі отримані рекомендації відповідно роботоспроможності запропонованої концепції, оцінки складності розробки і налагодження агентів.

Результатом виконання даної роботи стала формалізована модель агента ігрового процесу на базі методів машинного навчання і запропонований підхід до здійснення навчання даної моделі. Також результат роботи представлено середовищем моделювання ігрового процесу з фактичною реалізацією запропонованої моделі та підходу до його навчання. Відповідне середовище моделювання дозволяє проводити експерименти з навчання моделі агента.

Список використаних джерел

1. Outline of machine learning [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Outline_of_machine_learning#Machine_learning_algorithms. – Заголовок з екрану.
2. Charles D. The Past, Present and Future of Artificial Neural / D. Charles, S. Mcglinchey // in Proceedings of the Fifth International Conference on Computer Games: Artificial Intelligence, Design and Education. - 2004. - pp. 163–169.
3. Miikkulainen R. Creating Intelligent Agents in Games / R. Miikkulainen, R. Miikkulainen // Proc. Natl. Acad. Eng. 20006 Conf. Front. Eng. - vol. 2004.- no. MI. - 2006.
4. Buckland M. Programming Game AI by Example / M. Buckland // Wordware Game Developers Library. - Jones & Bartlett Learning. -2005.- 495 p.
5. Schwab B. AI game engine programming / B. Schwab // Second edn. Cengage Learning.- 2009.- 736 p.
6. Wender S. Data Mining and Machine Learning with Computer Game Logs / S. Wender, I. Watson.- no. October, 2007.

Кривенко О. В., Загирний М. М.

ИССЛЕДОВАНИЕ И МОДЕЛИРОВАНИЕ ИГРОВОГО ПРОЦЕССА НА БАЗЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

Методы машинного интеллекта с обучением вносят свою специфику к созданию и наладке игровой системы.

Одна из главных проблем — это необходимость предвидения всего набора входных ситуаций и возможных ответов на моменте проектирования и невозможность расширения их списка без переобучения. Это приводит к сужению возможности их использования в реальных игровых системах.

Объектом исследования являются процессы создания и обучения игровых агентов на основе эволюционных подходов искусственного интеллекта.

Цель работы - разработка и обоснование формальной модели игрового агента, основанного на методах машинного обучения, программная реализация игрового процесса с применением нейронных сетей и эволюционных методов оптимизации при множественных поколениях популяций игровых агентов.

Для осуществления поставленной цели исследована теоретическая база, существующие исследования и разработки в отрасли; спроектирован игровой сценарий, определены основные агенты, их основные возможности и ожидаемое поведение; реализовано обучение игровых агентов различными типами нейронных сетей; выполнено тестирование разработок, оценка качества поведения и принятия решений искусственным интеллектом с использованием нейронных сетей; проведен сравнительный анализ различных типов нейронных сетей, предлагаемых рекомендаций по их использованию для заданных условий.

Для отображения искусственной нейронной сети использованы шаблонные компоненты слоя, нейрона и моста.

Созданный программный модуль позволит игровым агентам, построенным с использованием различных нейронных сетей, соперничать друг с другом (а не с человеком, как в обычном режиме игры), позволит выявить более подготовленного из них.

Научная новизна исследования заключается в том, что формализована модель игрового агента, в основе которой методы машинного обучения.

Полученные в работе результаты могут быть использованы при разработке видеоигр, построенных на базе искусственного интеллекта игровых агентов, в основе которых методы машинного обучения, а также в рамках других научных исследований.

Ключевые слова: *Unity, машинное обучение, машинный интеллект, нейронные сети, генетические алгоритмы, игровой агент, видеоигра, программный модуль*

Kryvenko Olha, Zahirnyi Mykhailo

RESEARCH AND MODELING OF GAME PROCESS ON THE BASIS OF MACHINE TRAINING METHODS

Methods of machine intelligence with training contribute their specifics to the creation and commissioning of a gaming system.

One of the main problems is the need to anticipate the entire set of input situations and possible answers at the time of design and the impossibility of expanding their list without retraining. This leads to a narrowing of the possibility of their use in real gaming systems.

The object of research is the processes of creating and training game agents based on the evolutionary approaches of artificial intelligence.

The purpose of the work is the development and justification of a formal model of a game agent based on machine learning methods, software implementation of the game process using neural networks and evolutionary optimization methods for multiple generations of game agent populations.

To achieve this goal, the theoretical base, the existing research and development in the industry; a game scenario was designed, the main agents were identified, their main capabilities and expected behavior; training of game agents by various types of neural networks; development testing, quality assessment of behavior and decision-making by artificial intelligence using neural networks were performed; a comparative analysis of various types of neural networks, the proposed recommendations for their use for given conditions.

To display the artificial neural network, the template components of the layer, neuron, and bridge were used.

The created software module will allow game agents built using various neural networks to compete with each other (and not with a person, as in the normal game mode), and will reveal more prepared ones.

The scientific novelty of the study lies in the fact that a model of a game agent is formalized, based on machine learning methods.

The results obtained in this work can be used in the development of video games built on the basis of artificial intelligence of game agents based on machine learning methods, as well as in other scientific studies.

Keywords: *Unity, machine learning, machine intelligence, neural networks, genetic algorithms, game agent, video game, software module*

Рецензент: доц., канд. техн. наук Левицька Т. О.

Стаття надійшла 15.11.2019 р.

УДК № 004.413 + 303.09

Кривенко О. В., Пилипенко О. О.

АНАЛІЗ І МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ КОМП'ЮТЕРНОЇ КОМПАНІЇ

Сучасні підприємства змушені постійно займатися вдосконаленням своєї діяльності. Це вимагає розробки нових технологій і прийомів ведення бізнесу, підвищення якості кінцевих результатів діяльності і впровадження нових, більш ефективних методів управління і організації діяльності підприємств.

Проведено аналіз існуючих методологій моделювання бізнес-процесів і моделей розробки програмного забезпечення, на підставі якого проводиться реінжиніринг бізнес-процесів компанії.

Виконана реорганізація організаційної структури комп'ютерної компанії. З урахуванням недоліків функціональної структури організації пропонується інша форма підпорядкування співробітників – матрична.

Досліджено вживання нейронних мереж для вирішення багатьох задач ігрової розробки, моделювання та аналіз бізнес-процесів «як є», діагностика проблем підприємства, моделювання бізнес-процесів «як повинно бути», виконаний порівняльний аналіз проектів в MS Project.