



- [1] Цмоць І. Г., Демида Б.А., Подольський М.Р. Методи проектування спеціалізованих комп'ютерних систем управління та обробки сигналів у реальному часі мон украины, нан украины, украинская ассоциация по автоматическому управлению. Научно-технический журнал «Автоматика. Автоматизация. Электротехнические комплексы и системы», 2009, №2, сс.137-145.
- [2] Параллельная обработка информации: в 5 т. / АН УССР. Физ-мех. ин-т. – Киев: Наукова думка, 1984. – том 5.
- [3] Проблемно-ориентированные и специализированные средства обработки информации / А.И. Аксенов, В.В. Аристов, Е.Ю. Барзилович и др.; Под ред. Б.Н. Малиновского и Грицика В. В. – 1990. – 504с.
- [4] Методи розробки алгоритмів : Тексти лекцій / О. В. Костів, С. А. Ярошко; Львів. нац. ун-т ім. І.Франка. - Л., 2002. - 99 с.
- [5] Форсайт Дж., Малькольм., Мулер Р. Машинные методы математических вычислений. – М.: Мир, 1980. – 279с.
- [6] Д. Мэтьюз, Г. Цинк, Д. Куртис. Численные методы. Использование Matlab, –М. Издательский дом “Вильямс”, 2001. – 720 с.
- [7] <http://www.klaster-plus.ua/ua/about/dejatelnost/kompjuternye-sistemy/>
- [8] Хобин В.А., Титлова О.А. К вопросу измерения парожидкостного фронта в дефлегматоре абсорбционно-диффузионной холодильной машины (АДХМ) // Автоматика. Автоматизация. Электротехнические комплексы и системы. – 2007. – №2 (20). – С. 94 – 102.
- [9] Антонова, А. Р. Теоретичне та експериментальне дослідження процесів у газорідних сонячних колекторах-регенераторах альтернативних холодильних систем [Текст] : дис. канд. техн. наук : 05.14.06 : захищена 29.09.2014 : затвердж. 25.11.2014 / Альфія Раїсівна Антонова. – Одеса, 2014. – 205 с. – Бібліогр.: с. 186–205.

УДК 004.8

ИССЛЕДОВАНИЕ ИСПОЛЬЗОВАНИЯ НЕЧЕТКОЙ ЛОГИКИ ДЛЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ИГРАХ

Бутов Б.В.¹, Тишин П.М.², Шапорин В.О.³

^{1,2,3} – Одесский национальный политехнический университет, Одесса, Украина

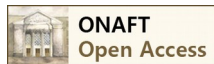
ORCID: ¹0000-0003-2506-5348, ²0000-0002-9508-9700, ³0000-0001-6494-7648

E-mail: ¹bogdan17but@gmail.com, ²petrmettal@gmail.com, ³shaporin_v@ukr.net

Copyright © 2018 by author and the journal “Automation technologies and business - processes.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: [10.15673/atbp.v10i4.1236](https://doi.org/10.15673/atbp.v10i4.1236)

Аннотация: В статье описывается использование нечеткой логики для решения задач искусственного интеллекта в играх. Представлены основные проблемы в создании искусственного интеллекта, а также основные методы реализации искусственного интеллекта в играх. Основное внимание уделено методу нечеткой логики, преимуществам и недостаткам его в практическом применении. Нечеткая логика представляет собой надмножество традиционной логики, которое было расширено для обработки понятия значений частичной правды между булевой функции истины и ложности. Нечеткая логика обычно принимает форму нечеткой системы рассуждений, а ее компоненты - это нечеткие переменные, нечеткие правила и механизм нечеткого вывода. Этот метод позволяет лингвистически сформулировать цели несобственного персонажа (NPC) и предоставляет возможность оценки ситуаций для принятия решения.

Основной целью работы представлена разработка интеллектуального агента, который реализует поведенческие реакции на основе правил с использованием нечетких множеств. В качестве практической задачи представлена задача описания поведенческой реакции NPC на интерактивные объекты, при достижении определенной цели. Для решения этой задачи введено понятие постортгонального пространства, которое позволяет описывать множество нечетких ситуаций и сравнить их между собой.

В результате была построена таблица правил поведения NPC, на основе которой можно спроектировать поведенческую реакцию NPC. Основным преимуществом данной модели является задание нечетких ситуаций с



помощью лингвистических переменных. Это обеспечивает простоту понимания условий на этапе разработки и создания модели более сложных ситуаций.

Abstract: The article describes the use of fuzzy logic to solve artificial intelligence problems in games. The main problems in the creation of artificial intelligence, as well as the basic methods for the implementation of artificial intelligence in games are presented. The focus is on the method of fuzzy logic. advantages and disadvantages of it in practical application. The fuzzy logic represents the superstition of traditional logic, which was expanded to handle the concept of partial truths between the Boolean function of truth and falsehood. Fuzzy logic usually takes the form of an obscure system of reasoning, and its components are fuzzy variables, fuzzy rules, and the mechanism of fuzzy output. This method allows the linguistic formulation of the objectives of the improper character (NPC) and provides an opportunity to assess situations for decision making.

The main purpose of the work is the development of an intellectual agent that implements behavioral responses based on rules using fuzzy sets. As a practical task, the task of describing the behavioral response of an NPC to interactive objects, when achieving a specific goal, is presented. In order to solve this problem, the concept of a post-thoroganal space is introduced, which allows describing a set of fuzzy situations and comparing them with each other.

As a result, a table of NPP behavior rules was constructed, on the basis of which it is possible to design the behavioral response of NPPs. . The main advantage of this model is the task of fuzzy situations with the help of linguistic variables. This ensures easy understanding of the conditions at the design stage and the creation of a model of more complex situations.

Ключевые слова: Искусственный интеллект, NPC, нечеткое множество, нечеткая логика, компьютерные игры, принятие решения, постортогональное пространство.

Keywords: Artificial Intelligence, NPC, fuzzy set, fuzzy logic, computer games, decision making, post-orthogonal space.

Введение. На сегодняшний день разработка игр является трудоемким и сложным процессом, который включает в себя множество отраслей от графического оформления, сценарного мастерства до программирования. Широкое применение в игровой индустрии отведено разработке искусственного интеллекта (ИИ). К ИИ относятся следующие аспекты видеоигры: управление анимацией, рулевое управление, процесс поиска на местности, планирование, тактическое и стратегическое мышление и обучение [1, 2]. Развитие новых методов и технологий ИИ является основной проблемой в разработке NPC, так как ИИ это сложно-программируемый комплекс команд, который основан на системе принятия решений в определенной ситуации.

Но игровой ИИ является лишь одной из ветвей более широкого применения искусственного интеллекта. Согласно Алану Тьюрингу, которого считают отцом искусственного интеллекта, агент является интеллектуальным, если его поведение невозможно отличить от человека [3]. Игровой ИИ добавляет еще одну часть определения ИИ. Для того, чтобы игра была успешной, не нужны высокоумные, человекоподобные, непобедимые экспертные оппоненты, требуется внедрить убедительных противников, агентов ИИ, с которыми не скучно играть [4].

Поэтому большинство игр используют очень простые методы ИИ такие как конечные автоматы (FSM) и деревья принятия решений [1, 2].

Байесовские сети используются в стратегии реального времени игры (RTS) для планирования цели [1]. Появляются нейронные сети в серии игр Creatures, а также в нескольких режимах реального времени стратегические игры и в отмеченной наградами игре Black & White 2. В последнее время использование эволюционных алгоритмов был популяризирован в видеоигре Галактическое Оружие [5].

Эти методы трудно реализовать в сравнении для FSM или деревьев решений. Некоторые из более продвинутых методов, например, генетические алгоритмы, обычно приносят с собой либо высокую вычислительную стоимости или высокой стоимости памяти.

Среди этих методов нечеткая логика является одним из важнейших инструментов, которые должен использовать разработчик игрового ИИ из-за простоты её формулировки в сочетании с её выразительности.

Нечеткая логика представляет собой надмножество традиционной логики, которое было расширено для обработки понятия значений частичной правды между булевой функции истины и ложности. Нечеткая логика обычно принимает форму нечеткой системы рассуждений, а ее компоненты - это нечеткие переменные, нечеткие правила и механизм нечеткого вывода.

Теория нечетких множеств была введена в 1965 году Лотти А. Заде в области академического искусственного интеллекта [6], но реализовать его идеи удалось японским исследователям, которые продемонстрировали практическое использование нечетких систем управления. Нечеткая логика также используется в сочетании с другими методами ИИ, такими как эволюционные алгоритмы или нейронные сети, в обучении и классификации [7].

Как и многие другие академические методы ИИ, нечеткая логика была протестирована в видеоиграх [7]. Нечеткая логика была официально представлена в разработке игр в 1996 Ларри О'Брайеном [8] и с тех пор изучена и усовершенствована другими авторами [1,2].

Преимущества нечеткой логики

Нечеткая логика может быть полезна для игрового ИИ в нескольких аспектах. Она может использоваться для принятия решений NPC, таких как выбор предметов или оружия, для управления движением юнитов.



Из-за лингвистического характера нечеткой логики формулировка правил может быть выполнена экспертами в области ИИ, а затем нечеткая система может использоваться для подражания рассуждениям эксперта [9]. Это большое преимущество по отношению к другим методам, которые требуют знания как самого метода, так и правил его настройки.

Нечеткие методы позволяют строить отношения ввода-вывода на основе знаний эксперта без необходимости создания сложной математической модели, которая может оказаться утомительной или невозможной для получения [7].

Нечеткая логика может использоваться для моделирования сложного поведения с низкой вычислительной стоимостью [1].

Недостатки применения нечеткой логики

Нечеткая логика требует правильного определения входных и выходных переменных, а также их отношений. Если нет эксперта в области ИИ, сложно будет разработать правила и, возможно, потребуется много настроек, что приведет к увеличению времени разработки.

Конструирование нечеткой системы для игры со многими агентами, в случае если они не будут тщательно разработаны, может привести к проверке сотен правил на каждом временном шаге, полностью исключая преимущества низкой вычислительной стоимости единичных проверок. Хотя самый простой вариант нечеткой системы появляется в большинстве игр, из-за этого недостатка были предложены улучшения, связанные с природой игрового ИИ. [2].

Один конкретный пример недостатка многих правил в нечеткой логике называется комбинаторным взрывом [10]. Разработчики игрового ИИ предлагают принять метод Комбса [11], который позволяет линейный рост правил в отношении числа переменных и множеств.

Литературный обзор. Пересматривая литературу, мы можем найти несколько упоминаний об использовании нечеткой логики в видеоиграх.

BattleCruiser: 3000AD6 - это космическая стратегическая игра с противоречивой историей развития, которая использует нечеткую логику наряду с нейронными сетями для управления неигровыми персонажами в игре [12].

S.W.A.T. 27, представляет собой тактическую игру в реальном времени, которая была посвящена широкому использованию нечеткой логики, позволяющей неигровым персонажам вести себя спонтанно на основе их определенных личностей и способностей [12,3].

Civilization: Call to Power8, пошаговая стратегическая игра, являющаяся побочным эффектом очень известной франшизы, использует FuSMs для определения приоритетов для AI стратегического уровня, позволяя определить индивидуальные черты для разных лидеров цивилизации [3].

Из приведенных примеров мы видим, что нечеткая логика успешно используется в играх, хотя разработчики, как правило, не идут дальше, чем простые логические системы вывода.

Цель и задачи исследования. Основной целью работы представлена разработка интеллектуального агента, который реализует поведенческие реакции на основе правил с использованием нечетких множеств. В качестве практической задачи будем рассматривать конкретную ситуацию в которой участвуют объекты интерактивного окружения, такие как дверь и лестница. Задача состоит в том, чтобы реализовать поведенческую реакцию на открытие двери и поднятие по лестнице для достижения определенной цели в игровой сцене [13].

Разработка правил с использованием нечетких множеств. При общем подходе NPC будет принимать эти объекты как препятствия и не станет взаимодействовать с ними. В целях устранения этой проблемы можно использовать нечеткую логику для выработки форм поведения, необходимых для взаимодействия с этими объектами. Чтобы избежать многозначности трактования семантических значений одного и того же параметра в различных ситуациях, строятся полные ортогональные семантические (постортогональные) пространства [14], которые служат областями нечетких значений каждого из параметров вне зависимости от рассматриваемой системы.

Для построения полного ортогонального семантического пространства (ПОСП) некоторого нечеткого параметра \tilde{p}_i определяется множество нечетких значений $\tilde{D}_i = \{\tilde{p}_i^k\} k = 1 \dots K_i$, где K_i количество нечетких значений, принимаемых i -м параметром, в виде нечетких чисел с функцией принадлежности μ_i^k , которая определена на интервале (p_{ib}^k, p_{ie}^k) , де $(p_{ib}^k, p_{ie}^k) \in D_i$ – значения начала и конца интервала соответственно, а D_i - базовое множество нечетких значений параметра \tilde{p}_i .

Для того, чтобы построенные \tilde{D}_i множества были ПОСП, необходимо, чтобы они удовлетворяли следующим аксиомам :

Аксиома 1– нормальность: каждая функция принадлежности μ_i^k нечетких значений \tilde{p}_i^k достигает единицы на



некотором ненулевом отрезке значений $[p_{ib_1}^k, p_{ie_1}^k]$ базового множества D_i :

$$\forall k \in [1; K_i] \quad \exists \tilde{p}_i^k \in \tilde{D}_i : \mu_i^k(p) = 1 \quad p \in [p_{ib_1}^k, p_{ie_1}^k]$$

Аксиома 2 – функция μ_i^k не убывает слева от $p_{ib_1}^k$ и не возрастает справа от $p_{ie_1}^k$:

$$\begin{aligned} \mu_i^k(p) &\geq \mu_i^k(p_{ib_1}^k), & p < p_{ib_1}^k \\ \mu_i^k(p) &\leq \mu_i^k(p_{ie_1}^k), & p > p_{ie_1}^k \end{aligned}$$

Аксиома 3 - функции μ_i^k не могут иметь более двух точек разрыва первого рода.

Аксиома 4 - полнота: для любого значения P из множества D_i найдется нечеткое значение $\tilde{p}_i^k \in \tilde{D}_i$ с ненулевым значением функции принадлежности $\mu_i^k(p)$ в данной точке:

$$\forall p \in \tilde{D}_i \quad \forall k \in [1; K_i] : \mu_i^k(p) \neq 0$$

Аксиома 5 - ортогональность: сумма всех значений функций принадлежности $\mu_i^k(p)$ в некоторой точке P базового множеств D_i должна равняться единице:

$$\sum_{k=1}^{K_i} \mu_i^k(p) = 1, \quad p \in D_i$$

Каждое нечеткое число $\tilde{p}_i^k \in \tilde{D}_i$ определяется через функцию принадлежности следующего вида:

$$p_i^k \Rightarrow \mu_i^k(p_i^k) = \begin{cases} 0, & p_i^k \leq p_{kb_1}^i, p_i^k \geq p_{ke_1}^i \\ \frac{p_i^k - p_{kb_1}^i}{p_{kb_1}^i - p_{kb_1}^i}, & p_{kb_1}^i < p_i^k < p_{kb_1}^i \\ 1, & p_{kb_1}^i < p_i^k < p_{ke_1}^i \\ \frac{p_i^k - p_{ke_1}^i}{p_{ke_1}^i - p_{ke_1}^i}, & p_{kb_1}^i < p_i^k < p_{ke_1}^i \end{cases} \quad i = 1..N_p, k = 1..K_i \quad (1)$$

С учетом $\mu_1^k < D_i, \mu_2^k < D_i, \mu_n^k < D_i$ где $< D_i$ отношение строгого порядка на множестве нечетких значений i -го параметра, должны выполняться следующие условия:

$$\begin{cases} p_{bk_1}^i = p_{bk_1}^i = \frac{\min(p_i)}{D_i} \\ p_{ke_1}^i = p_{ke_1}^i = \frac{\max(p_i)}{D_i} \end{cases} \quad i = 1..N_p$$

где p_i^k - некоторое четкое значение i -го нечеткого параметра $p_{ke_1}^i, p_{kb_1}^i$ соответственно начальное и конечное значения интервала значений базового множества D_i , на котором функция принадлежности k -го нечеткого значения i параметра положительно определена; $p_{ke_1}^i, p_{kb_1}^i$ - начальное и конечное значения соответственно интервала значений базового множества D_i , на котором функция принадлежности k -го нечеткого значения i -го параметра равна единице.

Для определения состояния объекта необходимо сравнивать некоторую входную нечеткую ситуацию S_i с каждой нечеткой ситуацией S_j . В качестве меры для определения степени близости нечеткой ситуации S_i нечеткой ситуации S_j используются: степень нечеткого включения нечеткой ситуации S_i в нечеткую ситуацию S_j , степень нечеткого равенства нечеткой ситуации S_i и нечеткой ситуации S_j . Выбор меры близости определяется особенностями нечеткой ситуации и организацией блока принятия решений.

Так, для следующих нечетких ситуаций:

$$S_i = \left\{ \frac{\mu_{S_i}(p_m)}{p_m} \right\}_{m=1}^M, \quad S_j = \left\{ \frac{\mu_{S_j}(p_m)}{p_m} \right\}_{m=1}^M,$$



где $\{p_m\}_{m=1}^M$ - набор признаков, по которым определяются нечеткие ситуации, а

$$\mu_{S_i}(p_m) = \left\{ \frac{\mu_k^i(T_{mk})}{T_{mk}} \right\}_{k=1}^{P(m)}$$

где $\mu_{S_i}(p_m)$ - степень принадлежности признака p_m к терму T_{mk} , а $P(m)$ - количество термов в терм-множестве признака p_m .

Определение: Степенью включения ситуации S_i в ситуацию S_j является величина $\vee(S_i, S_j)$, определяемая выражением:

$$\vee(S_i, S_j) = \wedge_{m=1}^M \vee \mu_{S_i}(p_m), \mu_{S_j}(p_m)$$

где $\mu_{S_i}(p_m), \mu_{S_j}(p_m)$ вычисляется следующим образом:

$$\vee \mu_{S_i}(p_m), \mu_{S_j}(p_m) = \wedge_{m=1}^{P(m)} (\mu_k^i(T_{mk}) \rightarrow (\mu_k^j(T_{mk})))$$

где $(\mu_k^i(T_{mk}) \rightarrow (\mu_k^j(T_{mk})))$ определяется отношением:

$$(\mu_k^i(T_{mk}) \rightarrow (\mu_k^j(T_{mk}))) = \max(1 - \mu_k^i(T_{mk}), \mu_k^j(T_{mk}))$$

При этом считается, что ситуация S_i нечетко включается в ситуацию S_j , $S_i \subseteq S_j$ если $\vee(S_i, S_j) \geq \lambda$, где λ - некоторое пороговое значение. Существование двух взаимных включений ситуаций S_i и S_j означает, что при пороговом включении λ ситуации S_i и S_j примерно одинаковы. Такое сходство ситуаций называется нечетким равенством.

Прежде всего необходимо обеспечить управление движением. Для этого используется лингвистическая переменная move (движение), которая определяется значением скорости движения. Она представлена тремя различными термами forwards (вперед), stop (останова), backwards(назад).

Основываясь на предыдущих формулах (1) необходимо построить пост ортогональные пространства. Терм forwards (вперед) определяется как нечеткое число с помощью функции принадлежности $\mu_{for\ var\ d}(v)$, которая задается соотношением:

$$\mu_{for\ var\ d}(v) = \begin{cases} \frac{v}{V1}, & 0 \leq v \leq V1 \\ 1, & V0 \leq v \leq V_{max} \end{cases}$$

Терм backwards (назад) определяется на нечеткое число с помощью функции принадлежности $\mu_{backwards}(v)$, которая задается соотношением:

$$\mu_{backwards}(v) = \begin{cases} \frac{v}{V1}, & 0 \leq v \leq -V1 \\ 1, & -V1 \leq v \leq -V_{max} \end{cases}$$

Терм stop (стоп) определяется на нечеткое число с помощью функции принадлежности $\mu_{stop}(v)$, которая задается соотношением (4).

$$\mu_{stop}(v) = \begin{cases} \frac{v+V1}{V1}, & -V1 \leq v \leq 0 \\ 1, & V=0 \\ \frac{V1-v}{V1}, & 0 \leq v \leq V1 \end{cases}$$

Постортогональное пространство переменной move представлено на рисунке 1.

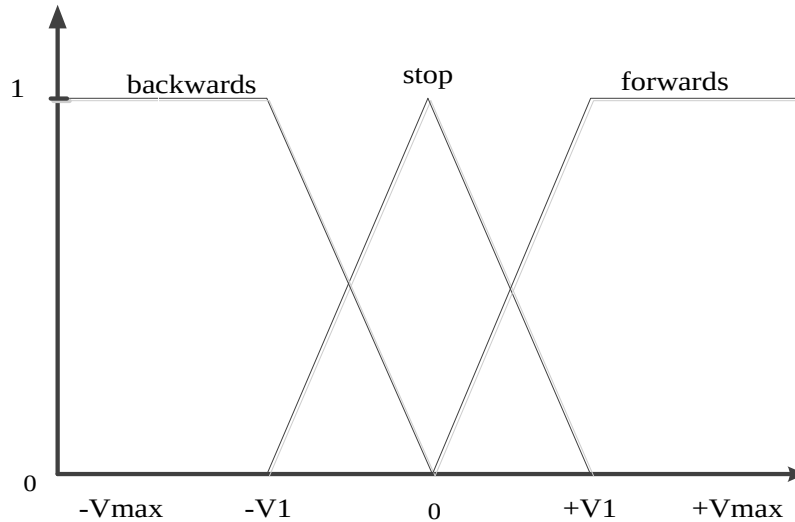


Рис. 1 – Постортогональное пространство множества переменной move

К процессу передвижения необходимо применить ещё одну лингвистическую переменную turn, которая отвечает за поворот тела. Описанная базовая переменная определяется относительным углом поворота в градусах. Переменная turn указывается 4 нечеткими ситуациями для переменных, каждая из которых указывает на объект, к которому должен повернуться NPC: button(кнопка), door(дверь), ladder (лестница), exit(выход). Функции этих переменных изменяются динамически, так как позиция этих объектов изменяется со временем.

Также предусмотрена лингвистическая переменная look, которая позволяет управлять углом наклона головы. В переменную look входят 2 терм-множества up (вверх), down (вниз), no (не изменять).

Вышеперечисленные лингвистические переменные door и ladder являются переменными описывающими восприятия объекта.

Переменная door представлена термами open(открыта) close(закрыта) и fully open (полностью открыта). Переменная button представляет собой блок управляющих решений, значение которого имеет несколько состояний B_1 (закрыть полностью), B_2 (закрыть наполовину), B_3 (оставить), B_4 (открыть наполовину), B_5 (открыть полностью). Для каждого определённого состояния строится нечеткая ситуационная сеть (НСС). Нечеткая ситуационная сеть [15] представляет собой нечеткий ориентированный взвешенный граф переходов по нечетким эталонным ситуациям.

Лингвистическая переменная ladder, определена с помощью двух термов held(занятый) и top(вверх). Терм held определен во времени и имеет максимальное значение 1, через секунду как NPC коснется лестницы и возвращается в 0 с тот момент, когда через секунду NPC сойдет с лестницы. Терм top представлен переменной height (высота). Переменная принимает значение 1 в момент, когда NPC готов сойти с лестницы и значение 0, после того, как будет сделан шаг от верхней ступени лестницы.

Результаты исследования. Далее необходимо разработать набор правил для последовательного выполнения действий. Для того, чтобы NPC прошел через дверной проем вначале необходимо нажать на кнопку, а затем пройти через дверной проем. Правила подобной задачи описываются в следующей таблице.

При организации этой формы поведения необходимо обеспечить приоритет с обращением к нажатию кнопки, а после проходить через дверной проем.

Для того, чтобы NPC взобрался по лестнице необходимо сначала подойти к лестнице, схватиться за неё, затем поднять взгляд вверх и осуществлять движение в прямом направлении до тех пор, пока не будет достигнута вершина лестницы, после NPC необходимо направить взгляд вперед и сойти с лестницы на пол. Правила подобной задачи описываются следующим образом.

Таблица 1 – Правила задачи взаимодействие с дверьми

Условия	Действие
If not button_pressed	Повернуться в сторону кнопки, двинуться вперед
If button_pressed	Повернуться к двери
If button_pressed and door not fully open	Остановиться
If door fully open	Двинуться вперед, повернуться к двери

**Таблица 2 – Правила задачи взаимодействие с лестницей**

Условия	Действие
True	Двинуться вперед
If ladder is not held	Повернуться к лестнице
If ladder is held	Посмотреть вверх
If ladder is top	Посмотреть вверх, двигаться, повернуться к выходу

На основе представленных правил поведения можно описать ряд подобных ситуаций взаимодействия NPC с объектами окружающей среды. Основным преимуществом данной модели является задание нечетких ситуаций с помощью лингвистических переменных. Это обеспечивает простоту понимания условий на этапе разработки и создания модели более сложных ситуаций.

Список использованных источников

- [1] Mat Buckland. Programming Game AI by Example. Jones & Bartlett Publishers, 1 edition, 2004. P. 36-70.
- [2] Ian Millington. Artificial Intelligence for Games (The Morgan Kaufmann Series in Interactive 3D Technology). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006. P. 371-395.
- [3] Alan Turing. Computing machinery and intelligence. Mind LIX, 1950. P. 2.
- [4] Daniel Johnson and Janet Wiles. Computer games with intelligence. In In Procs. 10th IEEE Intl Conf. on Fuzzy Systems, IEEE, 2001. P. 61–68.
- [5] E. Hastings, R. Guha, and K. Stanley. Automatic content generation in the galactic arms race video game. In Proceedings of the 5th international conference on Computational Intelligence and Games, CIG'09, 2009. P. 5.
- [6] Lofti A. Zadeh. Fuzzy sets and systems. System Theory, 1965.
- [7] Jiljang Wang Gabriyel Wong. A fuzzy-control approach to managing scene complexity. Game Programming Gems 6, 2006. P. 5-7.
- [8] Larry O'Brien. Fuzzy logic in games. Game Developer Magazine, 1996. P. 6-7.
- [9] Mason McCuskey. Fuzzy logic for video games. Game Programming Gems 1, 2000. P. 7-8.
- [10] Michael Zarozinski. Imploding combinatorial explosion in a fuzzy system. Game Programming Gems 2, 2001 P 342-350.
- [11] William E. Combs. The Fuzzy Systems Handbook 2nd Ed, Academic. 1999. P 48-50.
- [12] Penelope Sweetser and Janet Wiles. Current ai in games: a review. Australian Journal of Intelligent Information Processing Systems, 2002 P. 24-42.
- [13] Алекс Дж. Шампандар. Искусственный интеллект в компьютерных играх. Москва. «Вильямс», 2007 стр. 450-478.
- [14] Копытчук И. Н., Тишин П. М. Модели, методы и информационная технология повышения точности регистрации массы движущихся грузов в тензометрических системах. Одеса — 2016 стр.35-46.
- [15] (Борисов В.В., Круглов В.В., Федулов А.С. Нечеткие модели и сети. //Телеком – 2007г. стр.218-230.

References

- [1] Mat Buckland. Programming Game AI by Example. Jones & Bartlett Publishers, 1 edition. pp. 36-70, 2004.
- [2] Ian Millington. Artificial Intelligence for Games (The Morgan Kaufmann Series in Interactive 3D Technology). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. pp. 371-395, 2006.
- [3] Alan Turing. Computing machinery and intelligence. Mind LIX. pp. 2, 1950.
- [4] Daniel Johnson and Janet Wiles. Computer games with intelligence. In In Procs. 10th IEEE Intl Conf. on Fuzzy Systems, IEEE. pp. 61–68, 2001.
- [5] E. Hastings, R. Guha, and K. Stanley. Automatic content generation in the galactic arms race video game. In Proceedings of the 5th international conference on Computational Intelligence and Games, CIG'09. pp. 5, 2009.
- [6] Lofti A. Zadeh. Fuzzy sets and systems. System Theory. 1965.
- [7] Jiljang Wang Gabriyel Wong. A fuzzy-control approach to managing scene complexity. Game Programming Gems 6. pp. 5-7, 2006.
- [8] Larry O'Brien. Fuzzy logic in games. Game Developer Magazine. pp. 6-7, 1996.
- [9] Mason McCuskey. Fuzzy logic for video games. Game Programming Gems. pp. 7-8, 2000.
- [10] Michael Zarozinski. Imploding combinatorial explosion in a fuzzy system. Game Programming Gems 2. pp. 342-350, 2001.
- [11] William E. Combs. The Fuzzy Systems Handbook 2nd Ed, Academic. pp. 48-50, 1999.
- [12] Penelope Sweetser and Janet Wiles. Current ai in games: a review. Australian Journal of Intelligent Information Processing Systems. pp. 24-42, 2002.
- [13] Alex J. Champandar. Iskysstvennsy intellect v computernih igrakh. Moscwa. "Williams". pp. 450-478, 2007.



- [14] Kopytchuk I.N., Tishin P.M. Modeli, metody i informacionnaia tehnologia povishenia tochnosti registracii massi dvishuchihsia grysov v tenzometricheskikh sistemah. Odessa. pp. 35-46, 2016.
[15] Borisov V.V., Kruglov V.V., Fedulov A.S. Nechetkie modeli I seti. Telecom. pp. 218-230, 2007.

УДК 621.867.2

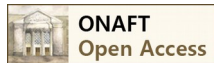
РОЛИКОВИЙ ВУЗОЛ СТРІЧКОВОГО ТРАНСПОРТЕРА

Сандлер А.К.¹, Дрозд О.В.²

^{1,2} – Національний університет "Одеська морська академія", м. Одеса, Україна

E-mail: ¹albertsand4@gmail.com, ²ELENADROZD912@gmail.com

Copyright © 2018 by author and the journal "Automation technologies and business - processes.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0>



DOI: [10.15673/atbp.v10i4.1237](https://doi.org/10.15673/atbp.v10i4.1237)

Анотація: Розглянуті відомі конструкції роликів опор стрічкових транспортерів, як елементу вантажної системи судна. Визначені недоліки та шляхи вдосконалення. Запропоновано нове схематичне рішення роликів опор.

Summary: Considered known designs of roller bearings of belt conveyors, as an element of the ship's cargo system. Deficiencies and ways to improve. A new circuit design of roller support is proposed.

Ключові слова: судновий стрічковий конвеєр, стрічка, ролик, експлуатація, ремонт, обслуговування.

Keywords: ship conveyor belt, belt, roller, operation, repair, service.

1. Вступ

Багаторічний досвід експлуатації транспортерів підтверджує головні до-стоїнства конвеєрного транспорту - високий рівень продуктивності праці, що досягається шляхом автоматизації роботи устаткування, можливість транспортування вантажу на великі відстані і низькі виробничі витрати. Вони є одним з основних засобів безперервного дільничного і магістрального транспорту в морських портах, на вугільних шахтах, розрізах і рудниках. Застосування стрічкових транспортерів постійно зростає в зв'язку з ростом вантажопотоків і відстаней транспортування, що зумовило необхідність створення високопродуктивних стрічкових транспортерів більшої довжини і потужності (рис. 1).

Досить високі капітальні витрати на придбання транспортерів компенсуються низькими експлуатаційними витратами. Собівартість одиниці вантажу, що перевозиться стрічковими транспортерами нижче, ніж у інших транспортних засобів, що застосовуються для перевезень вантажу на ту ж відстань.