

# Нечітка логіка в комп'ютерних іграх

Ю. І. Деніс

*Розглядаються можливості і особливості реалізації штучного інтелекту в комп'ютерних іграх. Пропонується використання нечітких множин з переліком специфічних ситуацій їх використання. Основний штучний інтелект комп'ютерної гри постійно потребує поліпшення для задоволення все зростаючих вимог гравців. У цій статті обговорюється, як інтелектуальні агенти і нечітка логіка може допомогти підвищити якість і кількість найважливіших елементів комп'ютерної гри: взаємодії. Застосування нечіткої логіки в дизайні поведінки докладно ілюструється на прикладах простої гри.*

---

## Вступ

Перерахуємо чотири основні елементи комп'ютерних ігор: представлення, взаємодія, конфлікт і безпека. З чотирьох елементів взаємодія вважається найважливішим.

Інтелектуальні агенти і нечітка логіка - два методи, які можуть допомогти підвищити якість взаємодії в грі. Агенти забезпечують нову архітектуру ігрового дизайну, що призводить до більш гнучкої взаємодії, а нечітке управління дає практичний метод для генерування вправної поведінки.

В області ігор, нечітке управління є не просто ще одним методом управління: це якісний генератор поведінки. Істоти із вправною поведінкою підвищують комплексність сприйняття, можливість приносити насолоду і довіру до віртуального середовища. У порівнянні з деякими іншими інтелектуальними методами засновані на нечітких правилах системи прості в розробці та реалізації, а також вони дуже схожі на засновані на чітких правилах системи. Як природне продовження класичної логіки, нечітка логіка грає все більш важливу роль в іграх.

---

## Передмова

Теорія агентів визначає якості агентів. Серед багатьох теорій виділимо одну з найпопулярніших і найвпливовіших: модель знання-ціль-вибір (ЗЦВ).

Модель ЗЦВ вивчає намір і його зв'язок з іншими інтелектуальними відношеннями. Як випливає з назви, ЗЦВ має три основні розумові позиції в якості будівельних блоків - знання, ціль, вибір (намір):

- Знання - інформація агента про навколишнє середовище та про себе самого.
- Ціль описує мету агента: стан системи, що агент хоче досягти.
- Вибір - це ряд дій, які агент вибрав для досягнення цієї мети.

Агент ЗЦВ діє через ітерації циклу його виконання. На початку кожного циклу, агент відчуває своє навколишнє середовище і оновлює свої знання. Як наслідок, цілі можуть пробуджуватися у відповідь на нещодавно отримані подразнення. Потім агент протиставляє своїм наявним планам цілі знайти найкращий придатний до застосування план для кожної цілі (на основі планового показника відповідності) і ініціює нові наміри виконання. Після закінчення циклу, вибір, який служить найбільш невідкладній цілі, вибирається з бібліотеки намірів для справжнього здійснення.

Комунікація має першочергове значення в будь-якій серйозній програмі з агентами: без зв'язку майже неможливо здійснювати контроль над мультиагентною системою, дозволити саме по собі об'єднане рішення проблем.

Обчислення на базі нечітких правил є найбільш широко використовуваною формою нечіткої логіки. Системи на базі нечітких правил, звичайно, складаються з чотирьох компонентів: 1) "фазифікатор" перетворює чіткі вхідні дані у нечіткі множини; 2) "база правил" зберігає відношення між входами і бажаними виходами, виражена набором правил вигляду ЯКЩО-ТОДИ; 3) "елемент міркування" застосовує входи до правил; 4) "дефазифікатор" перетворює результат міркування назад у чіткі вихідні значення.

---

## Гра

---

Для перевірки теоретичних аспектів та тестування ефективності системи на базі нечітких правил у вирішенні задач поведінки програмованого агента нам необхідно створити гру. Гра-імітація буде являти собою модель різних територій. Агенти керуватимуть кожен своїм танком, який матиме на меті проїхати крізь територію до встановленої мети, об'їжджаючи перешкоди, і знищити всі ворожі танки. Всі вороги мають одну й ту ж саму стохастичну поведінку. Під час гри, агенти повинні розробити стратегію на основі властивостей місцевості, щоб ефективно запобігти ворожій загрозі, залишаючись при цьому у безпеці. Таким чином при виборі цілі агент має вирішувати багатокритеріальну задачу. Імітація закінчиться, коли танк досягне мети - певного місця на карті або знищивши всіх ворогів.

У агентно-орієнтованому підході цільова система розкладається на автономні істоти і неживі об'єкти. Першим кроком необхідно розкласти систему належним чином і виявити агентів. Типи сутностей складають елементи місцевості, різні види танків супротивника, намічені точки перебування - бази (мета, якщо вона необхідна), і танки гравців. У теорії, всі сутності можуть бути агентами, але це призведе до розтрати обчислювальних ресурсів.

Сутності танків-гравців і сутності баз проектуються в якості агентів, залишивши решту в якості неживих об'єктів. Для створення більш реалістичної поведінки, агенти танків-гравців мають обмежену видимість, що, у свою чергу, вимагає, щоб база (також з обмеженою видимістю) контролювала свої околиці і оповіщала агентів танків-гравців, коли вороги наближаються. Крім того, база постійно оцінює своє власне положення та обмежує відстань до танків-гравців до рівня, що відповідає передбачуваній загрозі в даний момент часу. Ця стратегія гарантує уникнення ситуації, де танки гравця опиняться занадто далеко, щоб виключити загрозу бази.

З точки зору проектування, агентно-орієнтоване рішення - це природне розширення об'єктно-орієнтованого програмування. Окрема програмна частина - тривимірний двигун - надає рендеринг, звук, виявлення зіткнень, збереження геометрії. Агенти типу ЗЦВ імплементовані в об'єкти гри для керування їх поведінкою, але фактичні дії здійснюються за допомогою самих об'єктів.

Структура ігри забезпечує цикл управління обумовлений зовнішнім таймером для обробки анімації і зіткнень, а також, щоб дати іншим ігровим сутностям можливість виконувати свою власну обробку ситуації. Ворожі танки використовують цей час, щоб зробити випадкові рухи, а іноді і відкрити вогонь, в той час як танки гравця і база передають контроль до агентів.

Хоча агентна структура гри включає розумові операції агента, плани повинні бути встановлені, щоб інформувати агента, що може бути зроблено і як це зробити. Є шість основних завдань для агента гравця: уникати снарядів ворога, захищати базу, знищити ворогів, в яких можна потрапити, відстежувати ворогів, уникати зіткнення з товаришами по команді, і при відсутності інших завдань, досліджувати поле бою. Кожна з цих ситуацій має бути повністю владнана принаймні одним планом. Аналогічно, для агента бази, тільки два плани необхідні: клікати на допомогу і обмежувати діапазон вільного переміщення танків гравця.

---

## Нечітка логіка у ігрових агентах

---

Як зазначалося раніше, база накладає динамічні обмеження на радіус вільного пересування танка-гравця у відповідності з її сприйнятим рівнем загрози. Система на базі нечітких правил використовується тут, щоб підвищити рівень деталізації процесу прийняття рішення. База розглядає два фактори при оцінці загрози: 1) рівень захоплення (для остаточного захоплення бази необхідно пробути в певному радіусі від неї деякий час, нехай 100 секунд); 2) сили ворогів у полі зору. Система на базі нечітких правил приймає ці два входи і отримує бажаний радіус для танків гравця.

Нижче наводиться серія фотографій гри, що показують ефект від систем на базі нечітких правил в різних ситуаціях. Позначки 1, 2 і 0 вказують на два танки-гравці і базу, з їх межами видимості позначеними як 1в, 2в і 0в. Межа радіуса вільного переміщення танків-гравців також показана на фотографії (позначається як 00) для візуалізації цілей.

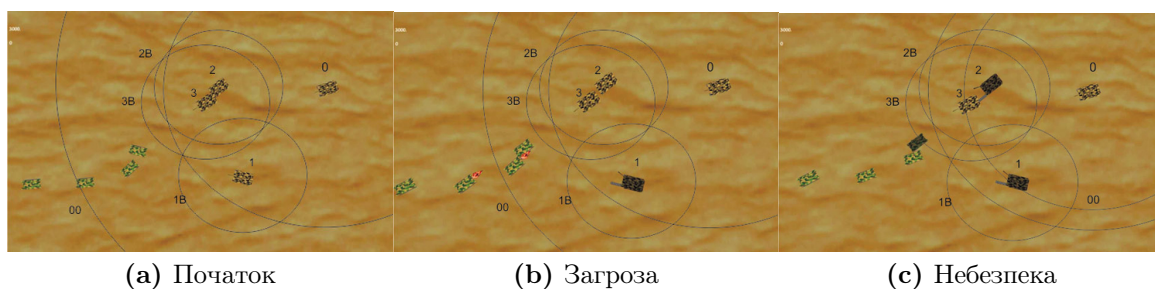


Рис. 1. Ситуації

На рис. 1а гра тільки почалася і всі умови сприятливі: захоплення не відбувається, ворогів в зоні видимості бази немає. Як результат, танкам-гравцям дозволено пересуватися майже вільно. Коли ситуація погіршується на рис. 1b, танки-гравці інформуються залишатися ближче до бази. Нарешті, на рис. 1с база знаходиться перед обличчям неминучої загрози - ворог наближається дуже сильно, змушуючи направити танків-гравців в безпосередню близькість від бази.

---

## Висновки

Агентно-орієнтований підхід до розробки ігор пропонує багато переваг в усьому циклі розробки. Він забезпечує природний спосіб моделювання ігрових істот і програмну архітектуру високої гнучкості і низької стохастичності, що дозволяє варіюватися поведінці. Крім того, розробники можуть легко інтегрувати свої старі середовища розробки гри у новий підхід в проектуванні.

Нечітка логіка - це ще одна інтелектуальна методика, яка може бути використана для підвищення продуктивності в іграх. Вона оперує проблемами комплексного управління при низьких обчислювальних витратах, без відмови від деталізації поведінки. Логічним продовженням піднятої теми є застосування нечіткої логіки вищих порядків. Хоча стаття розглядає її застосування в рамках агентно-орієнтованого ігрового проектування, нечітке управління, безумовно, може застосовуватися в іграх будь-якої архітектури.

---

## Список літератури

- [1] С. Рассел, П. Норвіг, Штучний інтелект: сучасний підхід, 2-е вид., видавничий дім "Вільямс Москва, 2006
- [2] С. Мацієвський, Нечіткі множини: навчальний посібник, видавництво Калінінградського державного університету, Калінінград, 2004.
- [3] Л. Заде, Поняття лінгвістичної змінної та його застосування до прийняття наближених рішень, видавництво "Світ Москва, 1976.
- [4] М. Міцумото, К. Танака, Нечіткі множини другого типу у нормах алгебраїчного множення та суми, Нечіткі множини та системи №05, видавничий дім "Північна Голландія Амстердам, 1981.

---

## Автори

**Юрій Ігорович Деніс** — аспірант 3-го року навчання, кафедра інформаційних систем, факультет кібернетики, Київський національний університет імені Тараса Шевченка, Київ, Україна; E-mail: [kukuskay@ukr.net](mailto:kukuskay@ukr.net)