

# АЛГОРИТМ ОПИСАНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ НЕЗАВИСИМОГО ИГРОВОГО МИРА

Дмитриев А. С.

Кафедра информационных технологий автоматизированных систем,  
Белорусский государственный университет информатики и радиоэлектроники  
Минск, Республика Беларусь  
E-mail: lex150498@mail.ru

*Рассматривается алгоритм и принципы описания искусственного интеллекта для использования в игровых мирах компьютерных игр независимого типа для усиления эффекта погружения игрока в игровой процесс.*

## ВВЕДЕНИЕ

Игровой мир компьютерной игры может быть зависимым и независимым от действий игрока. В случае независимого мира большая часть процессов в нём протекает без внешнего вмешательства. Задачей такого мира является демонстрация игроку своей полноценности путём имитации процессов реального мира. Поведение игровых персонажей в таких мирах является одним из важнейших элементов, так как обычно с ними часто взаимодействует игрок.

## I. АКТУАЛЬНОСТЬ

Персонажи под управлением искусственного интеллекта в играх могут вести себя абсолютно по разному, но в мирах с имитацией реальных процессов их основная задача - это создать у игрока ощущение взаимодействия с чем-то живым и самобытным. Данный алгоритм даёт возможность сбалансировать между собой затраты по производительности и интересность искусственного интеллекта для игрока.

## II. ОСНОВНЫЕ ФАКТОРЫ УЧИТЫВАЕМЫЕ В АЛГОРИТМЕ

При описании алгоритма основными факторами стали:

1. Самодостаточность – у игрового персонажа под управлением искусственного интеллекта всегда должно быть занятие. Даже если он ожидает некоего события, то это должно быть похоже на ожидание живого существа, например, проигрывать анимация ожидания, с, к примеру, покачиваниями тела, изменением направления взгляда и так далее. Персонаж ни в коем случае не должен выглядеть как бездушная статуя.
2. Производительность – так как персонажи всегда должны быть заняты неким делом, то их действия будут нагружать вычислениями центральный процессор, поэтому вычисления должны быть минимизированы, особенно если в игровом мире действует одновременно большое количество персонажей. Иногда для оптимизации стоит пожертвовать функционалом, так как ни

один игрок не будет рад какой-нибудь игровой детали, если при этом игра работает нестабильно.

3. Изменчивость – шаблонное поведение персонажей очень быстро приедается игроку. Это не значит, что искусственный интеллект должен постоянно удивлять игрока чем-то новым, но небольшие изменения маршрутов, действий или реакций на события хорошо разбивают шаблонность и делают персонажей менее предсказуемыми, а значит более интересными для игрока.

## III. ОСНОВНЫЕ ПРИНЦИПЫ АЛГОРИТМА

При описании алгоритма были выработаны следующие принципы его построения:

1. Основная логика персонажей должна быть представлена в виде деревьев поведения. Такой подход требует описания большого количества ситуаций, но он будет куда более гибким и производительным, чем построение логики на основе нейронных сетей, так как они хорошо работают с узкоспециализированными задачами, но хуже с широкопрофильными, как, например, с имитацией поведения. При этом отдельные сегменты побочной узконаправленной логики, к примеру поиск оптимального укрытия или пути, можно сделать с использованием нейронной сети, если при этом не страдает производительность.
2. У персонажей должны быть свои области влияния: своё расписание действий, свои территории передвижения и активности, свои социальные группы, и своё положение в иерархии. Например, персонаж с более высоким приоритетом в иерархии может передать выполнение одной из задач из своего расписания в расписание персонажа с более низким приоритетом, но с учётом их нахождения в одной социальной группе.
3. Персонажи из взаимодействующих групп должны без ведома игрока обмениваться вычислительной информацией в целях оптимизации и быстрого действия. К примеру если один из персонажей атакующий, а второй должен от него искать укрытие, то ата-

кующий должен посчитать с учётом своих параметров эффективность атаки ближайших к обороняющемуся укрытий и передать эту информацию ему. При получении такой информации от нескольких противников обороняющийся сможет узнать суммарную эффективность каждого из укрытий и уже рассчитывать её с остальными параметрами, например дистанцией до укрытия. При этом когда атакующий обнаруживает обороняющегося, то он сразу же проводит данные расчёты и передаёт их обороняющемуся одновременно и передавая информацию, и уведомляя его, что он обнаружен. Это избавляет от лишних проверок для обороняющегося был ли он замечен или нет и даёт ему оперативную информацию по факту обнаружения.

#### IV. АЛГОРИТМ ОПИСАНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

В ходе работы над алгоритмом были выработаны следующие шаги разработки искусственного интеллекта:

1. Описать логику пространств в которых может перемещаться искусственный интеллект.
2. Задать формальные правила создания уровней игрового мира, основываясь на логике пространств.
3. Описать логику перемещения и поиска путей искусственного интеллекта в подготовленных для него пространствах.
4. Задать ключи для изменения состояний искусственного интеллекта.
5. Создать счётчик времени на уровне и основываясь на нём описать функционал для создания расписаний активностей искусственного интеллекта.
6. Создать зоны активности персонажей.
7. Наполнить зоны активности предметами для взаимодействия.
8. Описать типовое поведение искусственного интеллекта по расписанию в зонах активности.
9. Присвоить персонажам группы и описать взаимодействие в них.
10. Создать иерархию в группах и между группами.
11. Добавить возможность искусственному интеллекту передавать задачи вышестоящим в иерархии единиц нижестоящим.
12. Присвоить зонам активности статусы и условия доступности для различных групп.
13. Описать генератор расписаний с учётом параметров персонажей и их групп.
14. Описать логику взаимодействия групп между друг другом.
15. Описать поведение искусственного интеллекта в нестандартных для единиц и групп ситуациях.

Далее описаны некоторые решения для шагов алгоритма. В качестве границ зон действий было решено выбрать все замкнутые или частично замкнутые контуры в игровом мире, состоящие из объектов, через которые не могут проходить игровые персонажи, а в качестве переходов установить триггеры на местах границ где персонажи могут проходить, чтобы уведомлять их о смене зоны. Зонам были присвоены идентификаторы доступности для разных групп персонажей, периоды времени данной доступности и выполняемые внутри активности. Персонажам были присвоены социальные группы, расписание возможных активностей и черты характера, представленные числовыми значениями, и действующие как ключи при выборе последующих действий в дереве поведения. При создании логики передачи задач в расписании между персонажами учитывалась не только позиция в иерархии, но также может ли получатель задачи её выполнять и есть ли у него доступ в необходимую зону. Если доступа нет, то он выдаётся, а в расписании передавшего задачу на освобождённом месте создаётся новая активность. После описания типовых ситуаций поведения для персонажей было описано нетиповое поведение с реакциями на игрока и других персонажей. К примеру, если персонаж при неких обстоятельствах, например, нанесении ему урона, получает статус испуга и принимает решение бежать, то он начинает игнорировать правила доступа в зоны, и может забежать туда, где при типовых условиях находиться не должен.

#### V. ВЫВОДЫ

В ходе тестов выяснилось что алгоритм работает стабильно с большим количеством персонажей, но с условием равномерного переключения их между активностями, чтобы не создавать большое количество одновременных изменений состояний персонажей и обчёта путей передвижения.

#### VI. ЗАКЛЮЧЕНИЕ

Данный алгоритм поведения имеет высокий коэффициент масштабируемости, но при его использовании стоит установить границы возможного взаимодействия игрока с ним, так как лучше сделать некое законченное поведение с необходимыми вариациями, чем описывать множество ситуаций с которыми игрок возможно даже ни разу не столкнётся.

#### VII. СПИСОК ЛИТЕРАТУРЫ

1. Artificial Intelligence: A Modern Approach / J. R. Stuart, P. Norvig // Pearson, 1994.
2. AI Game Programming Wisdom / S. Rabin // Cengage Learning, 2002.
3. AI for Game Developers / D. M. Bourg, G. Seemann // O'Reilly Associates, 2004.