

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИГРОВОГО ИИ

В данной работе рассматриваются основные алгоритмы, используемые в процессе создания игр, в частности в Unity и Unreal Engine. Обсуждаются актуальные проблемы, возникающие при применении данных алгоритмов к системам игровых объектов с симуляцией физики.

Введение

Использование искусственного интеллекта в производстве игр охватывает создание контента, разработку логики и оптимизацию рабочих процессов. Среди наиболее распространённых инструментов можно выделить следующие: NavMesh-алгоритмы — система навигации, описывающая путь из точки А в точку Б в виде полигонов; RVO/ORCA-алгоритмы — локальное избегание коллизий; FSM, Behavior Trees, Utility AI, GOAP — методы принятия решений.

I. Класс NavMesh

Класс NavMesh представляет собой алгоритм построения навигационной сетки, использующий алгоритм A*. Окружающая среда представляется в виде ориентированного графа, где узлами являются состояния персонажа, то есть его положение. Каждый смежный узел исследуется до достижения конечной цели или тупика, при этом сохраняется информация об уже посещённых узлах. Если конечный узел совпадает с искомым, маршрут считается найденным. Алгоритм A* гарантирует минимальную длину пути, если эвристика (например, евклидово расстояние) не превышает реальную стоимость. При этом он обычно исследует меньше узлов, чем алгоритм Дейкстры.

II. Алгоритмы принятия решений

Выбор архитектуры принятия решений определяет правдоподобие поведения персонажей. Конечные автоматы (FSM) эффективны для простых сценариев, однако при масштабировании возникает проблема «взрыва состояний». Они представляют собой набор состояний (например, «Ожидание», «Работа», «Ошибка»), событий и переходов между ними.

Современным стандартом являются деревья поведения (Behavior Trees), обеспечивающие иерархическую структуру задач и гибкую приоритизацию действий. В отличие от FSM, они имеют древовидную структуру с отношениями родитель—потомок.

Петриняк Елизавета Павловна, магистрант кафедры интеллектуальных информационных технологий БГУИР, petrinakliza@gmail.com.

Шипова Дарья Руслановна, магистрант кафедры интеллектуальных информационных технологий БГУИР, shipova.darya2312@mail.ru.

Научный руководитель: Шатилова Ольга Олеговна, старший преподаватель кафедры вычислительных методов и программирования, магистр технических наук, o.shatilova@bsuir.by

III. Проблемы

Ключевой проблемой является «дрожание» решений (decision chattering) — частое переключение между равнозначными действиями. Для его устранения применяются механизмы гистерезиса и временные задержки. Другой важный аспект — рассинхронизация навигационного и физического уровней: алгоритм A* строит путь для точки, тогда как физический движок учитывает массу и инерцию, что может привести к застреванию объектов.

Динамическое обновление навигационной сетки (NavMesh Re-baking) в разрушаемой среде создаёт высокую нагрузку на процессор. Решением являются гибридные подходы, сочетающие глобальное планирование и локальное избегание столкновений (RVO, ORCA).

Например, алгоритм ORCA формирует линейные ограничения для каждого агента, определяя допустимые скорости, при которых столкновения не произойдёт. Затем выбирается скорость, максимально близкая к желаемой, но находящаяся в безопасной области.

IV. Заключение

Таким образом, интеграция алгоритмов поиска пути, принятия решений и избегания столкновений позволяет создавать адаптивное и реалистичное поведение NPC. Дальнейшее развитие связано с использованием методов машинного обучения и гибридных архитектур.

Список литературы

1. Epic Games. AI Behavior Trees Documentation / <https://dev.epicgames.com>
2. Unity Technologies. NavMesh and Navigation System Documentation / <https://docs.unity3d.com>
3. A Modern Approach, 4th Edition, ISBN 978-0-13-461099-3 by Stuart J. Russell and Peter Norvig, published by Pearson Education © 2021. - 1166 с.