

АНАЛИЗ И СРАВНИТЕЛЬНОЕ ИССЛЕДОВАНИЕ МЕТОДОВ МОДЕЛИРОВАНИЯ ПОВЕДЕНИЯ НЕИГРОВЫХ ПЕРСОНАЖЕЙ В ВИДЕОИГРАХ

Яхья-заде С.С.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Столер В. А. – к.т.н., доцент, профессор кафедры ИКТ

Аннотация. В статье рассматриваются основные методы моделирования поведения неигровых персонажей (NPC) в видеоиграх: конечные автоматы, деревья поведения, планировщики и подходы на основе машинного обучения. Проводится их сравнительный анализ по критериям производительности, гибкости и реализуемости. Предложены рекомендации по выбору метода в зависимости от типа игрового проекта.

Ключевые слова: неигровые персонажи, искусственный интеллект в играх, конечные автоматы, деревья поведения, машинное обучение, поведенческое моделирование

Введение. Неигровые персонажи NPC (Non-Player Characters) являются неотъемлемой частью современных видеоигр. Их поведение во многом определяет игровой опыт: достаточно реалистичное и непредсказуемое поведение персонажей создаёт ощущение живого мира, тогда как примитивные реакции снижают погружение. С ростом вычислительных мощностей и усложнением игровых сценариев требования к системам искусственного интеллекта NPC существенно возросли. Разработчики вынуждены искать баланс между вычислительной эффективностью, реалистичностью поведения и сложностью реализации. Кроме того, развитие игровых технологий и рост ожиданий игроков требуют создания всё более сложных моделей поведения, способных имитировать элементы человеческого принятия решений.

Задача моделирования поведения NPC сводится к формальному описанию правил, по которым персонаж воспринимает игровую среду и выбирает дальнейшие действия. Выбор подходящего метода влияет не только на качество поведения, но и на общую производительность игры, трудоёмкость разработки и поддержки игровой логики. Существующие подходы значительно варьируются по уровню сложности – от детерминированных конечных автоматов до адаптивных нейронных сетей.

Цель данной работы – провести систематический сравнительный анализ наиболее распространённых методов моделирования поведения NPC, выявить их достоинства и недостатки, а также определить области наиболее эффективного применения каждого подхода в зависимости от масштаба и жанра игрового проекта.

Основная часть. Конечные автоматы FSM (Finite State Machines) – один из старейших и наиболее широко применяемых методов моделирования поведения NPC. Принцип работы FSM основан на разбиении всего поведения персонажа на конечное число состояний (например: «патрулирование», «преследование», «атака», «отступление»), между которыми осуществляются переходы при выполнении определённых условий [1].

Главным достоинством FSM является простота реализации и высокая производительность: вычислительная стоимость обновления состояния агента минимальна. Кроме того, поведение NPC легко предсказуемо и прозрачно с точки зрения разработчика, что существенно упрощает отладку. Тем не менее, по мере усложнения логики FSM превращается в труднопрогнозируемое «спагетти состояний»: количество переходов растёт квадратично относительно числа состояний.

Для смягчения этой проблемы используются иерархические конечные автоматы (HFSM), где состояния могут содержать подсостояния, что позволяет структурировать поведение по уровням абстракции. HFSM широко применяются в играх жанра RTS и шутерах от первого лица, где требуется поддержка как высокоуровневых стратегических, так и низкоуровневых тактических состояний [2].

Деревья поведения (Behavior Trees, BT) получили широкое распространение в игровой индустрии благодаря их модульности и читаемости. Структура BT представляет собой дерево, узлы которого делятся на управляющие (Sequence, Selector, Parallel) и листовые (Action, Condition). Обход дерева осуществляется сверху вниз, и каждый узел возвращает одно из трёх значений: Success, Failure или Running [3].

По сравнению с FSM деревья поведения значительно проще масштабировать: новые поведения добавляются как новые ветки без необходимости пересматривать существующую логику. Ключевым архитектурным дополнением к BT является «чёрная доска» (Blackboard) – общее хранилище данных, к которому обращаются отдельные узлы дерева. Это позволяет реализовывать сложную разделяемую логику без избыточного дублирования данных. BT активно применяются в AAA-проектах: Halo 2, Spore, серии Uncharted и многих других.

Недостатком является несколько более высокая вычислительная стоимость по сравнению с FSM, а также риск возникновения логических ошибок в глубоко вложенных деревьях. Для оптимизации нередко применяется кэширование результатов выполнения узлов между кадрами, что позволяет избежать повторного обхода неизменившихся ветвей.

Планировщики задач GOAP (Goal-Oriented Action Planning) – подход, при котором агент самостоятельно планирует последовательность действий для достижения поставленной цели, используя алгоритм A* (эвристический алгоритм поиска) в пространстве состояний мира. Каждое действие описывается условиями и эффектами, что позволяет планировщику динамически строить цепочки действий, ранее не предусмотренные разработчиком явно [4]. GOAP обеспечивает высокую гибкость и адаптивность: NPC способен реагировать на неожиданные изменения ситуации, формируя нестандартные решения из доступного набора примитивных действий. Примером успешной реализации служит игра F.E.A.R. (Monolith Productions, 2005), где GOAP обеспечил впечатляющее тактическое поведение противников, которое воспринималось игроками как высокоинтеллектуальное даже при относительно небольшом наборе доступных действий.

Вычислительная стоимость планирования может быть значительной при большом числе одновременно активных агентов и длинных цепочках действий. На практике применяются ограничения глубины поиска и предварительное разбиение целей на подзадачи для снижения нагрузки на планировщик.

С развитием нейронных сетей и методов обучения с подкреплением (Reinforcement Learning, RL) в игровой индустрии всё активнее исследуются подходы к созданию NPC на основе машинного обучения. Агент обучается посредством взаимодействия с игровой средой, получая скалярное вознаграждение или штраф за свои действия, и постепенно вырабатывает оптимальную политику поведения [5].

Наиболее известными демонстрациями возможностей RL в играх являются AlphaGo (DeepMind), AlphaStar (StarCraft II) и проект OpenAI Five (Dota 2), где агенты достигли сверхчеловеческого уровня мастерства. В коммерческих видеоиграх применение RL ограничено непредсказуемостью обученного поведения, сложностью интерпретации и отладки, а также высокими требованиями к вычислительным ресурсам в процессе обучения. Тем не менее, активное развитие специализированных библиотек и игровых движков постепенно упрощает интеграцию методов машинного обучения в реальные игровые проекты.

Перспективным направлением является использование имитационного обучения (Imitation Learning), при котором агент обучается на записанных данных поведения человека-игрока. Данный подход позволяет получить более предсказуемое поведение NPC, сохраняя при этом часть адаптивности, присущей методам RL [6].

На основании рассмотренных подходов проведём их сравнение по ключевым критериям: производительности, гибкости, масштабируемости и сложности реализации.

Конечные автоматы демонстрируют наивысшую производительность при наименьшей гибкости; деревья поведения обеспечивают хороший баланс между производительностью и масштабируемостью. GOAP обеспечивает высокую адаптивность ценой дополнительных вычислительных затрат. Методы на основе машинного обучения превосходят остальные по потенциальной гибкости, однако остаются наименее предсказуемыми и наиболее ресурсоёмкими при обучении.

Выбор метода моделирования поведения NPC должен определяться исходя из особенностей конкретного игрового проекта. Для простых игр с ограниченными ресурсами (мобильные платформы, инди-игры) рекомендуется применение FSM или иерархических FSM ввиду их низкой вычислительной стоимости и простоты реализации. Для крупных проектов с богатым игровым миром оптимальным выбором являются деревья поведения, возможно дополненные элементами GOAP.

На практике наиболее распространённым решением является гибридный подход: деревья поведения используются как основная управляющая структура, тогда как отдельные листовые узлы могут задействовать GOAP для локального планирования или нейронные сети для специализированных задач – навигации в сложном пространстве, оценки уровня угрозы либо имитации человекоподобного стиля игры. Кроме того, современные игровые проекты всё чаще используют комбинированные архитектуры искусственного интеллекта, в которых различные методы моделирования поведения NPC дополняют друг друга. Такая архитектура сочетает предсказуемость и производительность с гибкостью более сложных методов [3, 4].

Методы машинного обучения целесообразно рассматривать в качестве экспериментальных дополнений, особенно в проектах с процедурной генерацией контента и адаптивным балансом сложности. С развитием инструментальной базы и ростом производительности аппаратного обеспечения ожидается более широкое внедрение RL и имитационного обучения в коммерческих играх в ближайшее десятилетие.

Заключение. В работе проведён сравнительный анализ четырёх основных методов моделирования поведения неигровых персонажей в видеоиграх: конечных автоматов, деревьев поведения, GOAP-планировщиков и методов на основе машинного обучения. Показано, что каждый из подходов обладает специфическими достоинствами и ограничениями, определяющими сферу его применения. Наиболее универсальным и практически применимым в современных проектах является подход на основе деревьев поведения, который может быть расширен элементами планирования для повышения адаптивности NPC. Интеграция методов машинного обучения остаётся актуальной задачей дальнейших исследований в области игрового искусственного интеллекта.

Список литературы

1. Millington, I. *Artificial Intelligence for Games* / I. Millington, J. Funge. – 2nd ed. – Burlington : Morgan Kaufmann Publishers, 2009. – 896 p.
2. Buckland, M. *Programming Game AI by Example* / M. Buckland. – Plano : Wordware Publishing, 2005. – 520 p.
3. Colledanchise, M. *Behavior Trees in Robotics and AI: An Introduction* / M. Colledanchise, P. Ögren. – Boca Raton : CRC Press, 2018. – 222 p.
4. Orkin, J. *Three States and a Plan: The A.I. of F.E.A.R.* / J. Orkin // *Game Developers Conference Proceedings*. – 2006. – P. 1–9.
5. Sutton, R.S. *Reinforcement Learning: An Introduction* / R. S. Sutton, A. G. Barto. – 2nd ed. – Cambridge : MIT Press, 2018. – 552 p.
6. Шишков, А. В. *Методы обучения с подкреплением в системах игрового ИИ* / А. В. Шишков // *Информационные технологии и системы*. – 2022. – № 4. – С. 45–53.

UDC 004.8:004.94

ANALYSIS AND COMPARATIVE STUDY OF METHODS FOR MODELING NON-PLAYER CHARACTER BEHAVIOR IN VIDEO GAMES

Yahya-zada S. S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Stoler V. A. – Cand. of Sci., associate professor, professor of the department of ECG.

Annotation. The article examines the main methods for modeling non-player character (NPC) behavior in video games: finite state machines, behavior trees, planners, and machine learning-based approaches. A comparative analysis is conducted based on the criteria of performance, flexibility, and implementability. Recommendations are provided for choosing a method depending on the type of game project.

Keywords: non-player characters, artificial intelligence in games, finite state machines, behavior trees, machine learning, behavioral modeling