

УДК 004.8:004.946.005.336

АНАЛИЗ АРХИТЕКТУР ИГРОВОГО ИИ ПРИМЕНЯЕМЫХ ПРИ СОЗДАНИИ ПЕРСОНАЖЕЙ В ВИДЕОИГРАХ

Денис Владимирович Топильский

студент

denis.topiskiy@mail.ru

Лариса Ивановна Никонорова

кандидат сельскохозяйственных наук, доцент

Lenaniknrva@rambler.ru

Сергей Олегович Калинин

старший преподаватель

kalinin-sergei@yandex.ru

Мичуринский государственный аграрный университет

г. Мичуринск, Россия

Аннотация. В представленной статье проведен анализ моделей игрового ИИ. Перечислены недостатки и преимущества.

Ключевые слова: информатика, программирование, видеоигры, анализ.

При разработке современных видеоигр создание производительных и достоверных систем искусственного интеллекта для неигровых персонажей представляет собой одну из наиболее сложных проектных задач. Игровой ИИ в основном занимается выбором действий сущности в зависимости от текущих условий. Эффективный игровой ИИ должен сочетать: хорошую производительность, правдоподобность демонстрируемого поведения, способность функционировать в условиях жестких ограничений реального времени. В литературе по ИИ используется термин "агент". Агентом обычно является персонаж игры.

В настоящее время в игровой индустрии сформировался ряд устоявшихся архитектурных моделей ИИ, каждая из которых обладает уникальным набором характеристик, преимуществ и ограничений. К числу наиболее распространенных моделей относят: Finite State Machine, Behavior Tree, Goal-Oriented Action Planning, Utility AI.

Finite State Machine – это детерминируемая модель представляет собой формальную модель, в которой агент (неигровой персонаж) характеризуется набором состояний. Логика поведения агента строго зависит от условий. В любой момент времени агент может находиться только в одном активном состоянии. Поведенческая логика агента детерминируется predefined условиями. В любой момент времени агент находится в одном из нескольких возможных состояний. Благодаря своей концептуальной простоте, данная архитектура является одной из наиболее широко распространенных в области игрового искусственного интеллекта.

Behavior Tree – это модель, в которой решения выстроены в форме дерева, состоящую из узлов. Принцип работы заключается в последовательном обходе этого дерева от корневого узла к дочерним элементам. Специальные узлы управляют порядком и логикой выполнения, определяя, какое действие будет выполняться следующим. Другие узлы непосредственно отвечают за

проверку условий или выполнение конкретных действий, воздействующих на игровой мир.

Goal-Oriented Action Planning – это модель, которая реализует подход, основанный на автоматическом планировании последовательности действий. Агент в ней обладает конкретной целью и набором доступных действий. Каждое действие имеет четкие требования для выполнения и определенные последствия, изменяющие состояние игрового мира. Специальный модуль-планировщик осуществляет поиск в пространстве возможных состояний, автоматически выстраивая последовательность действий (план), которая позволяет выполнить цель. Оптимизация процесса планирования возможна через минимизацию совокупной "стоимости" плана, где алгоритмом расчета является A^* является самым популярным, эффективно находящий оптимальный план.

Модель Utility AI работает по принципу оценки полезности возможных действий. Каждое доступное агенту действие получает числовую оценку — показатель полезности в момент выполнения расчета. На основе анализа различных параметров игрового мира показатель вычисляется с использованием специальных математических формул—для полезности каждого действия они будут разными. Агент постоянно производит сравнительную оценку всех доступных действий, выбирая для выполнения вариант с максимальным значением полезности. Такой подход обеспечивает принятие решений, где выбор действия динамически адаптируется к изменяющимся условиям виртуальной среды.

Выбор конкретной архитектуры определяется множеством факторов:

- Требуемый уровень предсказуемости поведения
- Сложность решаемых агентом задач
- Доступные вычислительные ресурсы
- Требования к масштабируемости системы

Преимущества и недостатки различных архитектур игрового ИИ рассмотрены в таблице 1.

Таблица 1

Преимущества и недостатки и лучший вариант применения различных архитектур игрового ИИ.

Архитектура	Преимущества	Недостатки	Лучше всего подходят для
FiniteStateMachine	Простота, предсказуемость, не потребляют много ресурсов игрового устройства	Низкая масштабируемость, жесткая архитектура	Простые персонажи с предсказуемым поведением
BehaviorTree	Модульность, повторная использование, хорошая масштабируемость	Временное связывание решений, сложность изменения структуры	Сложные NPC, требующие модульной логики
Goal-OrientedActionPlanning	Гибкость, адаптивность, органичное поведение	Вычислительная сложность, выбор цели не всегда оптимален	Агенты, для которых важна способность к импровизации
Utility AI	Отсутствие временного связывания, учет контекста, плавность поведения	Сложность настройки и отладки, не интуитивное поведение при ошибках	Реалистичные симуляции в динамичных мирах с множеством факторов

Проведенный сравнительный анализ архитектур игрового искусственного интеллекта позволяет сделать следующий вывод: выбор оптимальной архитектуры должен основываться на ответах на следующие вопросы:

- Какой требуется уровень предсказуемости поведения? Детерминированные модели (FiniteStateMachine, BehaviorTree) обеспечивают полный контроль, в то время как плановые и оценочные системы (GOAP, Utility AI) демонстрируют более высокую адаптивность ценой снижения предсказуемости.
- Какие задачи должен выполнять агент? Для простых сценариев достаточно FiniteStateMachine. Реализация сложных, структурированных поведений требует масштабируемости Behavior Tree. Задачи, требующие адаптивного планирования в динамической среде, являются областью применения GOAP и Utility AI.
- Как сильно ИИ может влиять на производительность? Необходимо учитывать высокую стоимость планирования в GOAP для сложных пространств

состояний и потенциальную ресурсоемкость постоянного пересчета полезности в Utility AI.

- Насколько ИИ должен быть масштабируемым и поддерживаемым?

Модульность BehaviorTree и гибкость Utility AI имеют преимущество перед FiniteStateMachine, архитектура которой подвержена комбинаторному росту сложности.

Критически важным принципом является соответствие сложности ИИ поставленным задачам. Избыточно сложная модель для простого персонажа приводит к нерациональному использованию вычислительных ресурсов и времени разработки.

Также более предсказуемый ИИ позволяет облегчить процесс отладки багов в ИИ, но, также в некоторых играх ИИ должен быть менее предсказуемым, но это уже зависит от конкретной видеоигры.

На практике доказала свою эффективность стратегия гибридизации архитектур. Комбинирование моделей, например, использование Utility AI для высокоуровневого выбора стратегии с последующей ее реализацией через FiniteStateMachine или BehaviorTree, позволяет нивелировать недостатки отдельных подходов и создавать сбалансированные, эффективные системы, отвечающие специфическим требованиям игрового проекта.

Список литературы:

1. Капанов А. А., Лямин А. С. Создание игровой логики. Практикум / Санкт-Петербург: Лань, 2023. 48 с. ISBN 978-5-507-45629-1.
2. Обзор техник реализации игрового ИИ // Хабр – URL: <https://habr.com/ru/articles/420219/>
3. Разработка 3D-игр в Unity: руководство / Э. Дэвис, Т. Батист, Р. Крейг, Р. Станкел; перевод с английского П. М. Бомбаковой. Москва: ДМК Пресс, 2023. 298 с. ISBN 978-5-93700-254-9.

4. Романова М. В., Романов Е. П. Информатика: учебное пособие / 3-е изд., стер. Москва: ФЛИНТА. 2023. 190 с. ISBN 978-5-9765-3791-0.

UDC 004.8:004.946.005.336

ANALYSIS OF GAME AI ARCHITECTURES USED IN VIDEO GAME CHARACTER CREATION

Denis V. Topilsky

student

denis.topiskiy@mail.ru

Larisa Iv. Nikonorova

candidate of agricultural sciences, associate professor

lenaniknrva@rambler.ru

Sergey Ol. Kalinin

senior lecturer

kalinin-sergei@yandex.ru

Michurinsk State Agrarian University

Michurinsk, Russia

Abstract. The presented article analyzes the models of gaming AI. The disadvantages and advantages are listed.

Keywords: computer science, programming, video games, analysis.

Статья поступила в редакцию 24.10.2025; одобрена после рецензирования 20.12.2025; принята к публикации 29.12.2025.

The article was submitted 24.10.2025; approved after reviewing 20.12.2025; accepted for publication 29.12.2025.