

Сравнение моделей Игрового Искусственного Интеллекта

Алексей Линявский¹

¹Иркутский государственный университет путей сообщения, ул. Чернышевского, д. 15, Иркутск, 664074, Россия

Аннотация

Рассматриваются существующие модели игрового искусственного интеллекта. Приведено краткое описание базовой реализации и работы моделей игрового искусственного интеллекта Finite State Machine (FSM), Behavior Tree, Utility AI, Goal-Oriented Action Planner (GOAP), Hierarchical Task Network (HTN). Для каждой модели выделены положительные и отрицательные стороны разработки, дальнейшего поддержания и использования. Произведён анализ и сравнение со стороны рационального агента на основе описания базовой реализации каждой модели.

Ключевые слова

Игровой Искусственный Интеллект, искусственный интеллект, сравнение, GOAP, Behavior Tree, HTN, Utility AI, FSM.

1. Введение

При выборе искусственного интеллекта для управления игровым персонажем или автономными роботами существует проблема выбора искусственного интеллекта, который будет принимать решения о дальнейших действиях агента.

В данной статье представлено сравнение большинства популярных моделей игрового искусственного интеллекта.

2. Модели

Сравнение будет основано на следующих моделях игрового искусственного интеллекта:

1. FSM - Аппарат конечных состояний;
2. Behavior Tree – Дерево поведения;
3. Utility AI – Утилитарный искусственный интеллект;
4. GOAP – Ориентированный на задачу планировщик действий;
5. HTN – Сеть иерархических задач.

Каждая модель, описанная ниже, представлена в своей "минимально комплектации", поскольку для большинства реализаций эти модели немного модифицируются, и сравнение такого количества модификаций не представляется возможным.

2.1. FSM

Finite State Machine (FSM) [1, 3] считается одним из самых простых реализаций игрового искусственного интеллекта.

6th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2024), July 01–05, 2024, Irkutsk, Russia

EMAIL: inetlsec@mail.ru;

ORCID: 0009-0008-0412-9158



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2024 Workshop Proceedings

DOI: 10.47350/ICCS-DE.2024.32

Каждое состояние FSM описывает действие искусственного интеллекта и имеет возможность перехода в другое состояние (или состояния).

Для перехода между состояниями должно быть соблюдено условие перехода.

FSM обеспечивает достаточную гибкость разработки и использования при малых затратах ресурсов. Однако расширение подобной модели искусственного интеллекта невозможно без прямого взаимодействия с кодом FSM. Это, в свою очередь, влечёт за собой увеличение затрат на разработку подобной модели, а расширение большого FSM может ещё и значительно замедлить работу над моделью.

2.2. Behavior Tree

BT (Behavior Tree) [1, 3, 8] представляет собой дерево, которое состоит из root (корня), node (технических вершин) и leaf (вершин-листьев).

Root выполняет роль бесконечного цикла в дереве, постоянно перебирая его детей-вершин в зависимости от ответа, который они вернут. node содержат в себе описание возвращаемого ответа и порядок перебора детей-вершин. Обязательно содержит хотя бы 1 вершину в детях.

Leaf описывают непосредственно действие, которое необходимо выполнить, а также decorator - описание выполнения действия.

Leaf может вернуть 3 состояния: Success (Успех), Failed (неудача) и Running (в процессе). Эти состояния обрабатываются decorator, после чего он их передаёт в node. node, в зависимости от полученного ответа, переходит либо к следующему своему ребёнку, либо возвращает ответ своему родителю. И так, пока ответ не достигнет root.

Дерево поведения можно представить как "памятку при пожаре", которой следует агент искусственного интеллекта. Это позволяет достаточно точно описать все необходимые действия, которые искусственный интеллект должен предпринять.

Деревья поведения требуют от агента искусственного интеллекта знания об окружающей среде, что также требует и ориентирование искусственного интеллекта. Также необходим объектно-ориентированный подход при создании вершин, а значит для Деревьев поведения необходим высокоуровневый язык программирования.

2.3. Utility AI

Концепция Utility AI (UAI) [5, 6] достаточно проста:

Агент искусственного интеллекта знает о каких-то возможных ему действиях, сравнивает релевантность (вес) этих действий, находит наиболее ему подходящее и выполняет это действие.

Релевантность может быть представлена нелинейной зависимостью от нескольких переменных, что позволяет гибко настраивать "ценность" действий в глазах искусственного интеллекта.

UAI обеспечивает гибкое поведение искусственного интеллекта в зависимости от окружающей среды, а расширение возможностей модели UAI не займёт больших затрат ресурсов.

Однако UAI необходимо знать все доступные ему действия и сравнивать их вес между собой. Иначе говоря, оптимизация поиска оптимального действия при их большом количестве может быть затруднительна. А поскольку вес изменяется каждую условную единицу времени, то UAI может показать неэффективное использование ресурсов на достаточно большом количестве действий. Также, UAI должен знать об окружающей среде, в которой он находится, иначе говоря, он должен как-то в ней ориентироваться.

2.4. GOAP

Goal-Oriented Action Planner (GOAP) [1, 2] строит цепочку выполняемых действий исходя из текущей задачи агента искусственного интеллекта.

Задача – некоторое конечное состояние, которого желает достичь агент искусственного интеллекта. Каждая задача имеет несколько параметров, на основе которых и она и выбирается планировщиком задач: Начальное состояние агента, Конечное состояние агента, Вес.

Начальное состояние – состояние, в котором необходимо находиться агенту искусственного интеллекта.

Конечное состояние – состояние, в которое перейдёт агент искусственного интеллекта в случае успеха выполнения задачи.

Действие – непосредственно действие, которое выполнит агент искусственного интеллекта.

Каждое действие состоит из нескольких параметров: Непосредственно действие, Конечное состояние, Начальное состояние, Условие выполнения и Вес.

GOAP хоть и прост в своей концепции, требует работу с объектно-ориентированной парадигмой, поскольку реализация действий и задач без ООП представляется затруднительной. Иначе говоря, GOAP, в большинстве случаев, может работать только на высокоуровневых языках.

GOAP требует от агента искусственного интеллекта знание о мире, а потому ему необходимо какое-то ориентирование в окружающей среде.

Однако, взамен на вышеперечисленные требования, GOAP обеспечивает достаточно высокую интеллектуальность искусственного интеллекта, а также очень просто экстенсивное расширение возможностей искусственного интеллекта.

И, что тоже важно, GOAP может быть применён не к одному агенту искусственного интеллекта, а сразу к нескольким. Это позволяет построить на GOAP роевой интеллект.

2.5. HTN

Hierarchical Task Network (HTN) [4, 7] состоит из плана, который выполняет агент искусственного интеллекта. План в себе содержит 2 типа задач: Примитивные и Составные. Примитивные задачи, как правило, олицетворяют собой конкретное действие, которое выполняет агент искусственного интеллекта.

Составные задачи включает в себя как примитивные, так и другие составные задачи.

Образованный план выглядит как иерархическое представление папок в операционных системах, за что и получил своё название.

HTN позволяет искусственному интеллекту "мыслить" более абстрактно, так, как это делает человек. Это позволяет искусственному интеллекту создавать достаточно сложные и длинные цепочки действий. А гибкость добавления задач может стать сильным подспорьем в выборе именно данной модели.

В то же время, HTN, также как и GOAP, требует ориентацию в окружающем пространстве и ООП. А разработка HTN потребует хорошего специалиста, который понимает, как работает эта модель.

3. Выводы

Каждая модель Игрового искусственного интеллекта уникальна в своём исполнении. Иначе говоря, представленные выше модели представляют собой некоторую идею (концепцию), которую эти модели отражают. Как правило, перед реальным использованием этих моделей их немного модифицируют под конкретную задачу, по аналогии с Очередью (Queue) и Кучей (Stack). Так что сравнение моделей ограничено их базовыми вариациями, поскольку учесть все модификации моделей не представляется возможным.

Все сравнения и возможности применения построены с точки зрения рационального агента, который знает обо всех остальных возможных моделях.

FSM, несмотря на свою гибкость, применим только для простых систем, поскольку обеспечить его поддержку в дальнейшем может быть трудозатратно из-за огромной диаграммы состояний.

ВТ обеспечивает точность действий агента искусственного интеллекта, однако недостаточно гибок, в сравнении с другими моделями. Остаётся проблема расширения искусственного интеллекта. Требует ориентирования в пространстве для своей работы.

Гибкость UAI позволяет применить его в совершенно разной среде, однако на большом количестве необходимых действий может значительно замедлить свою работу. Также требует ориентирования в пространстве.

GOAP позволяет создать гибкий искусственный интеллект, который позволит агенту исполнять сложные действия. Однако потребует ориентирования в пространстве для своей работы, а также высокоуровневых языков (напр. C++, Java, C# и др.).

HTN обеспечивает весьма гибкую и легко расширяемую систему задач искусственного интеллекта, но требует хорошего специалиста для создания такой модели. Также, как и с GOAP, требует высокоуровневый язык и ориентирование в пространстве.

4. Список литературы

- [1] DigDigger, How does videogame AI make its decisions? (FSM, Behaviour Trees, BDI, GOAP) | Bitwise. URL: https://www.youtube.com/watch?v=ValJk15l_y8
- [2] sgkryvenko, Проектирование непредсказуемого интеллекта в играх. Часть 2 интеллект толпы. URL: <https://habr.com/ru/articles/654601/>
- [3] Kylotan, The Total Beginner's Guide to Game AI. URL: <https://www.gamedev.net/tutorials/programming/artificial-intelligence/the-total-beginners-guide-to-game-ai-r4942/>
- [4] T. Humphreys, Exploring HTN Planners through Example. URL: https://www.gameapro.com/GameAIPro/GameAIPro_Chapter12_Exploring_HTN_Planners_through_Example.pdf
- [5] This is Vini, Utility AI with examples. URL: <https://www.youtube.com/watch?v=d63hbJYYqM8>
- [6] AI and Games, How Utility AI Helps NPCs Decide What To Do Next | AI 101. URL: <https://www.youtube.com/watch?v=p3Jbp2cZg3Q>
- [7] AI and Games, HTN Planning in Transformers: Fall of Cybertron | AI and Games #14. URL: <https://www.youtube.com/watch?v=kXm467TFTcY>
- [8] Cyberstars Entertainment, Урок 6 | Unreal Engine 4 Искусственный интеллект - Дерево поведения ИИ | AI Behavior tree Blackboard. URL: https://www.youtube.com/watch?v=hWfIRS_DSuE