

Технологии искусственного интеллекта в компьютерных играх

В настоящее время, во всех сферах деятельности человека, все более широкое применение находят различные интеллектуальные системы. Они позволяют не просто автоматизировать рутинные операции, но и выполнять при этом функции, свойственные творческой работе человека. Как правило, интеллектуальная система представляет собой техническую или программную систему, способную решать задачи традиционно считающиеся творческими, принадлежащими конкретной предметной области, знания о которой хранятся в памяти такой системы [1]. Важной особенностью интеллектуальных систем является их возможность обучаться решению той или иной конкретной задачи.

Интеллектуальные системы развиваются в рамках научного направления, работающего над задачей создания искусственного интеллекта. Это направление содержит технологии, позволяющие моделировать интеллектуальное поведение систем. Технологии искусственного интеллекта призваны решать проблемы принятия решения, распознавания образа, оптимизации, прогнозирования и ряда других проблем, с которыми приходится сталкиваться при разработке интеллектуальных систем.

Наряду с использованием технологий искусственного интеллекта в традиционных областях, в последнее время, они получили широкое распространение в игровой индустрии. Развитие вычислительной техники и компьютерных технологий привело к повышению степени сложности компьютерных игр, а возможность программно моделировать в играх сложные системы, позволяет рассматривать игру как удобный полигон для создания новых методов и алгоритмов искусственного интеллекта. Целый ряд традиционных настольных игр, таких например, как шахматы, служил и до сих пор служит площадкой для испытаний новых методов и алгоритмических решений.

С развитием компьютерной игровой индустрии появилась необходимость решать задачи, связанные с моделированием поведения персонажей игры в различных игровых ситуациях. Использование технологий искусственного интеллекта позволяет справиться с этими задачами достаточно эффективно. Для того, чтобы рассматривать применяемые технологии, необходимо описать условия в которых существуют объекты игрового мира. Компьютерная игра представляет собой непрерывный цикл, который осуществляет работу логики программы и вывод графической информации на экран. Упрощенно, можно представить игровую программу следующими блоками [2]:

1. Инициализация.
2. Главный цикл.
3. Выход из программы.

В блоке инициализации осуществляются стандартные операции по распределению памяти, загрузке ресурсов и созданию игровых объектов. Основная работа осуществляется в главном цикле. Формально, главный цикл программы можно разделить на следующие части:

1. Получение пользовательского ввода.
2. Искусственный интеллект и логика игры.
3. Подготовка очередного кадра.

Очевидно, что нас интересует вторая часть. Она содержит основной код игры. Именно здесь выполняется работа искусственного интеллекта, логики игры и физической модели, результаты которой используются для вывода на экран очередного кадра [2].

Объекты игрового мира взаимодействуют как с внешней средой, так и друг с другом. Таким образом, одной из задач, является задача получения объектом информации об окружающем мире. Игровой объект должен уметь распознавать своих и врагов, пищу и укрытия, оружие и орудия труда.

В простых системах такое восприятие может ограничиваться простой проверкой положения объекта игрока. В более сложных системах требуется определять основные характеристики и свойства игрового мира, например возможные маршруты для передвижения, наличие естественных укрытий на местности, области конфликтов. При этом

разработчикам необходимо придумывать способ выявления и определения основных свойств игрового мира, важных для системы искусственного интеллекта. Например, укрытия на местности могут быть заранее определены дизайнерами уровней или заранее вычислены при загрузке или компиляции карты уровня [3].

Игровой объект получив информацию об окружающем мире принимает решение в отношении своих дальнейших действий. Процесс принятия решения является важной частью работы искусственного интеллекта. При этом, в играх используются как простые формы искусственного интеллекта, так и более сложные.

Простейшей формой искусственного интеллекта является система на основе правил. Такая система дальше всего стоит от настоящего искусственного интеллекта. Набор заранее заданных алгоритмов определяет поведение игровых объектов. С учетом разнообразия действий конечный результат может быть неявной поведенческой системой, хотя такая система на самом деле вовсе не будет «интеллектуальной» [3]. К таким системам относятся детерминированные алгоритмы, алгоритмы на основе шаблонов, алгоритмы, использующие конечные автоматы.

Детерминированные алгоритмы означают предопределенное и заранее запрограммированное поведение объектов [2]. К ним относятся такие алгоритмы, как:

1. случайное движение;
2. алгоритм следования за объектом;
3. алгоритм уклонения от объекта.

Довольно часто требуется создать объект, который действует по некоторому сценарию, то есть, выполнить целый ряд последовательных действий, повторяющихся каждый раз при возникновении определенных условий. В этом случае, для объекта создается шаблон поведения. Каждый шаблон хранится в виде последовательностей инструкций. Вместе с каждой инструкцией могут храниться дополнительные данные, уточняющие ее, например указывающие, на какое расстояние следует переместиться [2].

При добавлении к логике обработки шаблонов условных операторов, появляется возможность учитывать состояние игры и действия игрока. Такая технология позволяет составлять шаблоны «на лету», в процессе игры. Таким образом может быть реализовано подражание игроку. Эти шаблоны, обеспечивая предсказуемость персонажа игры, позволяют ему при необходимости полностью сменить тактику и стратегию игры [2].

Шаблоны представляют собой простейшие конечные автоматы. Конечный автомат (машина с конечным числом состояний) является способом моделирования и реализации объекта, обладающего различными состояниями в течение своей жизни. Каждое «состояние» может представлять физические условия, в которых находится объект, или, например, набор эмоций, выражаемых объектом [3]. Для создания устойчивого конечного автомата требуется следующее [2]:

1. Разумное количество состояний, каждое из которых представляет различные цели или мотивы.
2. Входная информация для конечного автомата.

Сделать игровой объект более интеллектуальным поможет добавление ему свойств, придающих ему индивидуальность. Это возможно при использовании вероятностной модели выбора перехода из одного состояния в другое. Различные объекты, имеющие в качестве модели поведения один конечный автомат, будут вести себя по-разному, если вероятность состояний для каждого из них будет различной. Для того, чтобы еще больше приблизить поведение объекта к «разумному» часто используется зависимость распределения вероятностей персонажа от некоторого фактора (например, расстояния до какого-то другого объекта). В этом случае, учитывая этот фактор, в зависимости от игровой ситуации будет использоваться своя таблица распределения вероятностей.

Следующей, по степени сложности реализацией игрового искусственного интеллекта, является реализация модели, позволяющей объекту реализовать запоминание и обучение. Используя технологию с запоминанием и обучением можно адаптировать действия объекта к

действиям игрока, более целенаправленно осуществлять поиск пищи, боеприпасов и снаряжения, обмениваться информацией с другими объектами. Технологии адаптивного искусственного интеллекта часто используются в боевых и стратегических играх со сложной механикой и огромным количеством разнообразных возможностей в игровом процессе.

Другой распространенной технологией, используемой для реализации игрового искусственного интеллекта, является поиск пути. При этом часто используются такие методы как:

1. Метод проб и ошибок.
2. Обход по контуру.
3. Поиск путей с использованием промежуточных пунктов.
4. Поиск в ширину.
5. Поиск в глубину.
6. Алгоритм Дейкстры.
7. A* поиск.

Метод проб и ошибок заключается в том, чтобы столкнувшись с препятствием отойти назад, повернуться вправо или влево на 45 или 90 градусов и переместиться на некоторое расстояние. После этого, проверить направление на целевую точку и повторить попытку добраться к ней по прямой.

Метод обхода по контуру состоит в том, что объект перемещается вокруг препятствия и периодически проверяет, пересекает ли отрезок между объектом и целью препятствие. Если не пересекает, то объект продолжает движение по направлению к своей цели. В противном случае, объект продолжает обход препятствия.

Для имитации более интеллектуального поведения объекта в игровом мире заранее создаются пути вокруг препятствий, которые представляют собой массивы точек или векторов. Когда объекту необходимо обойти препятствие, он по запросу получает кратчайший путь. Таким образом, траектория движения объекта представляется более реальной. Наличие известных путей обхода является основой поиска путей с использованием промежуточных пунктов. Этот метод используется в играх со сложными мирами и препятствиями разных типов. В игровом мире создается сеть путей, соединяющая все важные точки карты. Узлы этой сети являются промежуточными пунктами. Соответственно, дуги сети представляют вектора движения и расстояние между узлами. При наличии такой сети объект всегда может получить кратчайший путь к целевой точке.

Алгоритмы поиска в ширину, в глубину, A* поиск и алгоритм Дейкстры основаны на использовании графов и позволяют найти кратчайший путь между двумя вершинами графа. Они являются классическими алгоритмами поиска и хорошо известны. В играх, как правило, используются как их модификации (как правило в сторону упрощения), так и составные алгоритмы поиска.

Наиболее сложные игры требующие для своего выполнения серьезных вычислительных ресурсов могут использовать такие технологии, как нейронные сети, нечеткую логику и генетические алгоритмы. Эти технологии позволяют имитировать «осознанное» поведение игровых объектов, адаптировать их к изменяющимся условиям игрового окружения, принимать решения в условиях неопределенности.

Искусственные нейронные сети имитируют работу мозга. Информация передается между нейронами, а структура и вес нервных окончаний определяют поведение сети. Для искусственного интеллекта персонажа игры невозможно задать поведение, которое охватывало бы все возможные в окружающей среде ситуации. Обучение нейронной сети на ограниченном количестве примеров, позволяет ей самостоятельно генерировать поведение во всех прочих ситуациях [4].

Нечеткая логика позволяет программам работать в некотором диапазоне различных степеней истины. Эта модель подходит для алгоритмов принятия решений, так как человеческая логика сама по себе является приблизительной. Генетический алгоритм

представляет собой технику оптимизации, которая моделирует феномен естественной эволюции [4].

Нейронные сети, нечеткая логика и генетические алгоритмы являются сложными и интересными технологиями и заслуживают отдельной статьи. Использование этих технологий в рамках игровых алгоритмов позволяет получить достаточно сложный игровой искусственный интеллект, но требует значительных вычислительных ресурсов.

Таким образом, технологии искусственного интеллекта, применяемые в компьютерных играх представляют собой оптимизированный, упрощенный вариант. Такая оптимизация под конкретные задачи, позволяет в условиях ограниченной вычислительной мощности имитировать интеллект игровых персонажей на достаточно высоком уровне.

Библиографический список:

1. Электронный ресурс:

https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0

Интеллектуальная система — Википедия.

2. Андре Ламот Программирование игр для Windows. Советы профессионала, 2-е изд.

:Перевод с англ. - М. : Издательский дом «Вильямс», 2004. - 880 с. ISBN 5-8459-0422-6 (рус.)

3. Электронный ресурс:

<https://habr.com/company/intel/blog/265679/>

Создание искусственного интеллекта для игр — от проектирования до оптимизации/Блог компании Intel/Хабр

4. Джонс М. Т. Программирование искусственного интеллекта в приложениях. Пер. С англ.

Осипов А. И. - М.: ДМК Пресс, 2004. - 312 с. ISBN 5-94074-275-0