АЛГОРИТМ ПОИСКА ПУТИ В ИГРОВОМ ПРИЛОЖЕНИИ. АЛГОРИТМ ДЕЙКСТРЫ

Рязанцев Д.Д.,Жиляк Н.А.
Кафедра вычислительных методов и программирования, Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: {dmitryryaz, gh_nadya}@mail.ru

В статье описывается понятие алгоритма поиска пути. В частности расматривается алгоритм Дейкстры

Введение

По мере развития игр увеличивались их размеры и сложность. Виртуальные миры и карты стали больше, как и количество взаимодействующих объектов. Нередко сотни объектов одновременно ищут путь, а время обработки, отведенное на поиск пути, существенно не увеличивается. В связи с этим разработчики часто вынуждены идти на компромиссы в алгоритмах поиска пути и тратить значительное время на настройку и проверку алгоритмов.

I. Что такое алгоритм поиска пути?

Алгоритм поиска пути в игровых приложениях — это алгоритм, который используется для нахождения пути между двумя точками на игровом поле. Он может быть использован для перемещения персонажей, NPC, машин и других объектов в игре. Существует множество алгоритмов поиска пути, каждый из которых имеет свои преимущества и недостатки.

II. Разновидности алгоритмов поиска пути

Среди наиболее распространенных алгоритмов поиска пути в играх можно выделить:

1. Алгоритм Дейкстры: Этот алгоритм используется для нахождения кратчайшего пути между двумя точками на графе. Он может быть использован для поиска пути в играх, где перемещение ограничено определенными направлениями или где есть препятствия на пути. Алгоритм A*: Этот алгоритм является усовершенствованным вариантом алгоритма Дейкстры и используется для нахождения кратчайшего пути между двумя точками на графе. Он может быть использован для поиска пути в играх, где перемещение ограничено определенными направлениями или где есть препятствия на пути;
2. алгоритм поиска в ширину: Этот алгоритм используется для нахождения кратчайшего пути между двумя точками на графе. Он может быть использован для поиска пути в играх, где перемещение не ограничено определенными направлениями;
3. алгоритм поиска в глубину: Этот алгоритм используется для нахождения всех возможных путей между двумя точками на графе. Он может быть использован для поиска путей в играх, где перемещение не ограничено определенными направлениями;
4. алгоритм Ли: Этот алгоритм используется для нахождения кратчайшего пути между двумя точками на графе с единичными стоимостями ребер. Он может быть использован для поиска путей в играх, где перемещение не ограничено определенными направлениями;
5. алгоритм Флойда-Уоршелла: Этот алгоритм используется для нахождения кратчайших расстояний между всеми парами вершин в графе. Он может быть использован для поиска кратчайших расстояний между всеми объектами в игре.

Во всех этих алгоритмах соблюдается баланс между временем поиска и используемой памятью

Выбор наиболее подходящего алгоритма зависит от среды поиска пути в видеоигре и ее требований.

Существуют три свойства, которыми должны обладать алгоритмы, чтобы быть полезными на практике:

- Согласованность решений - качество решений (субоптимальность) не должно резко отличаться в разных задачах;
- адекватное время отклика - жесткие ограничения на время отклика, диктуемые игрой, должны быть соблюдены;
- Эффективность использования памяти - объем памяти для одного объекта и фиксированной памяти должен быть сведен к минимуму.

Цель - минимизировать неоптимальность, время отклика и использование памяти.

III. Алгоритм Дейкстры и его особенности

Алгоритм Дейкстры - это алгоритм на графах, который используется для нахождения

кратчайшего пути между двумя точками. Он может быть использован для поиска пути в играх, где перемещение ограничено определенными направлениями или где есть препятствия на пути. Алгоритм Дейкстры работает только для графов без ребер отрицательного веса.

Алгоритм Дейкстры начинает работу с одной из вершин графа и находит кратчайшие пути от этой вершины до всех остальных. Он работает пошагово, на каждом шаге он "посещает" одну вершину и пытается уменьшать метки. Работа алгоритма завершается, когда все вершины посещены.

Примеры игр, в которых используется алгоритм Дейкстры: Civilization, Starcraft, World of Warcraft,Diablo,Half-Life 2.

Алгоритм Дейкстры распространяется из начальной точки в ее окрестности. По мере распространения к последующим узлам он записывает предыдущие узлы и, наконец, достигает целевой точки и генерирует полный путь.

Пример процесса вычисления алгоритма Дийкстры:

Если имеется четыре узла 1, 2, 3, 4, с заданными узлами, линиями и весами (см. рис. 1) :

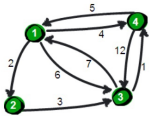

Рис. 1 – Пример работы алгоритма Дейкстры

Тогда в результате вычисления кратчайшего пути от 1 до всех точек с помощью алгоритма Дейкстры получим следующие пути: {1->2} {1->4} {1->2->3} .

На первой итерации, начиная непосредственно с узла 1, находим узел, на который он указывает, и заполняем соответствующую ячейку с весом до узла. Если путь к точке невозможен, соответствующая графа заполняется бесконечностью. Затем сравниваем веса каждого узла на первой итерации, и кратчайший путь от узла 1 к узлу 2 равен {1->2}, а вес равен 2.

Во второй итерации на основе предыдущей итерации заполнить бесконечность в ячейках, соответствующих путей. Затем расходятся по текущим покрытым узлам для поиска кратчайшего пути. Выясняется, что вес пути из {1->2->3} равен 5, что меньше исходного значения веса непосредственно из 1->2, поэтому значение веса записывается в ячейку, соответствующую 3. Наконец, сравниваются значения веса каждого узла, оставшегося на второй итерации, и кратчайший путь из узла 1 в узел 4 получается как {1->4}, а вес равен 4.

В третьей итерации, поскольку в графе осталось только четыре узла, эта итерация является последней. На основании предыдущей, покрытые в данный момент узлы расходятся в окрестности для поиска более короткого пути. Однако более короткого пути найти не удается, и остаются только узлы, которые можно выбрать, поэтому кратчайший путь из узла 1 в узел 3 получается, как {1->2->3}, а вес равен 5.

Наконец, получен кратчайший путь от узла 1 к каждому узлу.

IV. Идеи применения алгоритма Дейкстры в игровых приложениях

Конкретные идеи применения алгоритма Дейкстры для интеллектуального поиска пути в игре таковы:

1. Во-первых, на карту должны быть добавлены виртуальные горизонтальные и вертикальные пересекающиеся прямые линии, чтобы создать спецификационную сеть, похожую на шашечную сетку, тогда пересечение двух прямых линий или центр шашечной доски будет использовано в качестве узла. При этом ИИ может выполнять только четыре смежные операции - движение вверх, вниз, влево и вправо, поэтому матрица смежности графа задается только в четырех смежных направлениях каждого узла, причем те, которые выходят за пределы диапазона количества узлов, не задаются, а остальные узлы задаются до бесконечности;
2. Поскольку ИИ также необходим для управления персонажем, чтобы завершить движение по орбите тела столкновения, строки и столбцы матрицы смежности, в которых находится тело столкновения, все устанавливаются на бесконечность, то есть линия, соединяющая тело столкновения, разрывается.

Недостатки:

- Если масштаб сети слишком велик, то получение результата будет занимать много времени для получения результата;
- Для некоторых приложений, чувствительных к времени, или сервисов, работающих в реальном времени, нам необходимо сократить время работы.

V. Список литературы

1. Fast Grid-based Path Finding for Video Games / William Lee, Ramon Lawrence // University of British Columbia
2. Study of the Interception Scheme Based on A* Path Finding Algorithm in Computer Game / Zhenjie Li, Haiming Li // Journal of Computer and Communications, 2020, 8, 32-49
3. ProgrammerSought [Электронный ресурс] /Application of Dijkstra Algorithm in Game Intelligent Pathfinding - Режим доступа: https://www.programmersought.com/article/23446213813/ - Дата досутпа: 13.10.2023