

Постановка задачи

- Найти путь от точки А к точке Б значит найти последовательность действий проделав которые можно переместиться на игровом поле из А в Б
- Действия не противоречат правилам игры

Виды карт

- Карта дискретна. Нужно найти путь на графе
- Карта непрерывна. Нужно найти векторную функцию
- Комбинация двух подходов: об этом позже

Поиск пути на графе

- Найти путь значит найти последовательность рёбер, от исходной вершины к искомой
- Классические алгоритмы: Дейкстры, Флойда,
- Волновой алгоритм
- A^* - наиболее распространённый в играх

A* Общие сведения

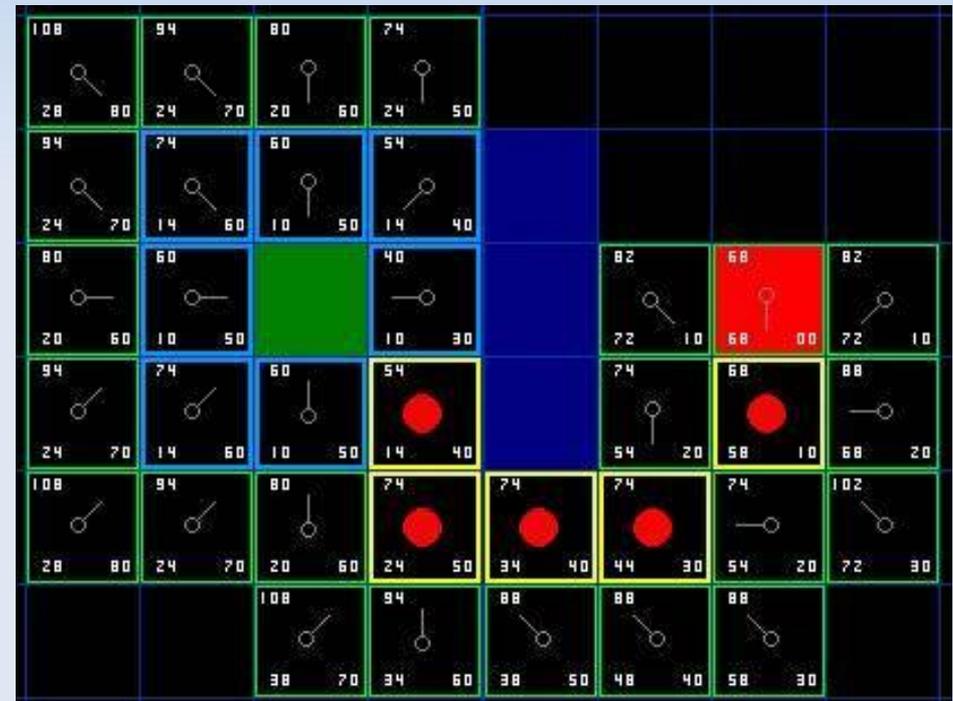
- Впервые упомянут в 1968 году Питером Хартом Нильсом Нильсоном и Бертраном Рафаэлем.
- Является эвристическим
- Всегда находит решение, если оно существует
- По сути является обобщённым алгоритмом Дейкстры
- Лёгок в реализации

A* Описание алгоритма

- Алгоритм A* оперирует с двумя списками: открытым и закрытым. В открытый список помещаются клетки, которые нужно проверить. В закрытые те, что уже не нужно проверять
- То, какую клетку проверить первой определяется по значению $F=G+N$. G – Стоимость передвижения из стартовой клетки в исходную с учётом данной N – эвристическая функция

A* описание алгоритма

- 1) Добавляем стартовую клетку в открытый список.
- 2) Повторяем следующее:
 - а) Ищем в открытом списке клетку с наименьшей стоимостью F. Делаем ее текущей клеткой.
 - б) Помещаем ее в закрытый список. (И удаляем с открытого)
- с) Для каждой из соседних 8-ми клеток ...
 - Если клетка непроходимая или она находится в закрытом списке, игнорируем ее. В противном случае делаем следующее.
 - Если клетка еще не в открытом списке, то добавляем ее туда. Делаем текущую клетку родительской для этой клетки. Рассчитываем стоимости F, G и H клетки.
 - Если клетка уже в открытом списке, то проверяем, не дешевле ли будет путь через эту клетку. Для сравнения используем стоимость G. Более низкая стоимость G указывает на то, что путь будет дешевле. Если это так, то меняем родителя клетки на текущую клетку и пересчитываем для нее стоимости G и F. Если вы сортируете открытый список по стоимости F, то вам надо отсортировать свесь список в соответствии с изменениями.
- д) Останавливаемся если:
 - Добавили целевую клетку в открытый список, в этом случае путь найден.
 - Или открытый список пуст и мы не дошли до целевой клетки. В этом случае путь отсутствует.
- 3) Сохраняем путь. Двигаясь назад от целевой точки, проходя от каждой точки к ее родителю до тех пор, пока не дойдем до стартовой точки. Это и будет наш путь

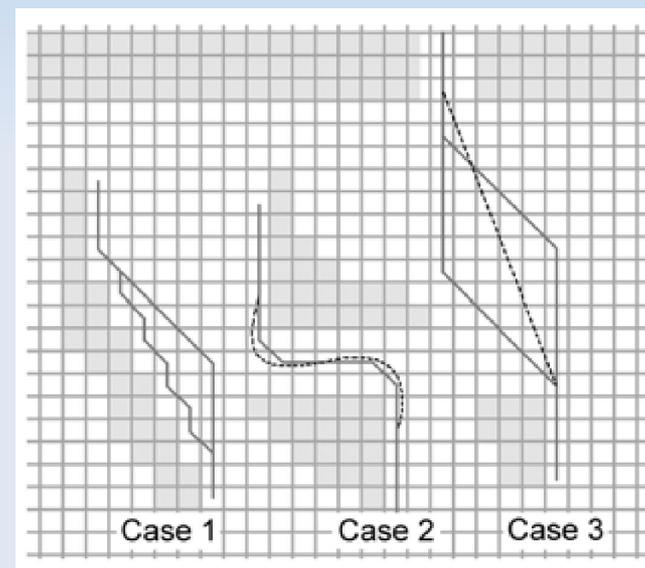


Проблемы A*

- Сглаживание пути
- Рост потребляемой памяти
- Одновременный поиск

Проблемы A*: сглаживание пути

- Неестественная траектория
- Зигзагообразная траектория
- Путь по центрам клеток



Проблемы A*: память и скорость

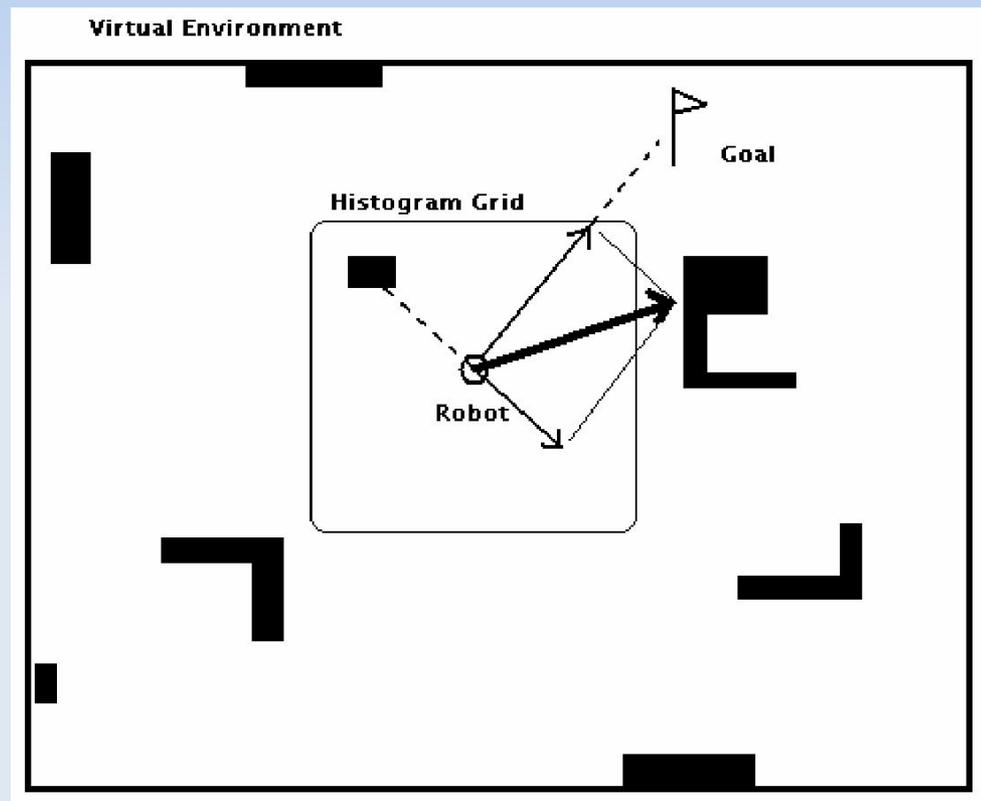
- Улучшаем алгоритм
- Улучшаем эвристику
- Укрупняем клетки(двухпроходный алгоритм)
- Комбинируем с другими алгоритмами

Метод потенциальных полей

- Препятствия отталкивают, цель притягивает
- Основная проблема – локальные минимумы
- Локальные минимумы обходим итерациями
- Не подходит для больших расстояний
- Быстрый

Метод виртуальных отталкивающих клеток

- Вариант предыдущего метода
- Препятствия режутся на клетки
- Отталкивают клетки, которые близко
- Более плавное движение



Комбинируем методы поиска

- Делим мир на локации
- От локации к локации – A^*
- Внутри локации метод потенциальных полей
- Локации это выпуклые фигуры или фигуры содержащие точку видимости
- Точка видимости, это точка из которой можно провести отрезок в любую точку локации целиком находящийся в ней

Вопросы

Нужна помощь – пиши на мыло

- vkozhaev@gmail.com