Тестовое задание: алгоритмы поиска пути на грид-графе

Цель

Реализовать и сравнить классические алгоритмы поиска пути на двумерном грид-графе для одного агента с поддержкой различных эвристик и параметров планирования.

Постановка задачи

Необходимо реализовать систему планирования пути на **2D grid-графе** для одного агента со следующими требованиями:

1. Алгоритмы поиска

Реализовать следующие алгоритмы:

- **Dijkstra**
- **BFS** (поиск в ширину)
- **A*** (A-star)
- **WA*** (Weighted A-star)

Алгоритмы должны корректно находить путь от стартовой вершины к целевой при наличии препятствий.

2. Эвристики

Для алгоритмов A* и WA* необходимо реализовать поддержку следующих эвристик:

- Манхеттенское расстояние
- Евклидово расстояние
- Диагональное расстояние (Octile distance)

Для **WA*** должен поддерживаться настраиваемый **вес эвристики** $w \geq 1$.

3. Типы связности графа

Поддержать два варианта графа:

- **4-связный** (up, down, left, right)
- **8-связный** (включая диагонали)

Для **8-связного графа** необходимо **запретить срезание углов препятствий**, т.е. диагональный переход разрешён только если обе ортогональные клетки свободны.

4. Конфигурация

Выбор параметров должен осуществляться через конфигурацию (файл или аргументы), включая:

- используемый алгоритм (Dijkstra / A* / WA* / BFS)
- тип эвристики
- вес эвристики (для WA*)
- тип связности графа (4 / 8)

Формат конфига — на усмотрение кандидата (JSON / YAML / CLI).

5. Архитектура и технологии

- Основная логика алгоритмов должна быть реализована на **C++**
- Допускается и рекомендуется реализовать обвязку и запуск экспериментов через **Python**
- Для связи C++ и Python рекомендуется использовать **pybind11**
- Код должен быть структурирован и расширяем

6. Тестирование и эксперименты

Необходимо провести тестирование реализованных алгоритмов на картах и задачах из набора:

- **MovingAI Benchmark** (https://movingai.com/benchmarks/grids.html)

Для тестирования:

- выбрать несколько карт разных размеров
- запустить разные алгоритмы и конфигурации
- собрать и сравнить метрики

Ожидаемые метрики

Минимальный набор:

- длина найденного пути
- время планирования
- количество раскрытых вершин (expanded nodes)

Дополнительно (по желанию):

- suboptimality (для WA*)
- визуализация пути

Ожидаемый результат

В результате выполнения задания ожидается:

1. Репозиторий с исходным кодом (C++ + Python)
2. Инструкция по сборке и запуску
3. Краткий отчёт с результатами экспериментов и наблюдениями:
    - сравнение A* и WA*
    - влияние эвристики
    - влияние связности графа

Критерии оценки

- Корректность алгоритмов
- Качество архитектуры и читаемость кода
- Аккуратность работы с эвристиками и ограничениями (corner cutting)
- Качество экспериментов и анализа
- Общий инженерный уровень решения

Полезные ссылки

https://theory.stanford.edu/~amitp/GameProgramming/