Программа нахождения кратчайших путей для игровых приложения

Выполнил:
ст. гр. ПИ-12-1 Пиляев Д.В.

Руководитель:
проф. Качко Е.Г.

Цель работы

Для большого числа игр поиск оптимальных путей является необходимой и важной частью, например для таких игр как: “Dota 2”, “Planetary Annihilation”, “Dragon Age”.

Целью аттестационной работы является разработка оптимизированной библиотеки для нахождения путей в игровых приложениях с последующей возможной интеграцией этой библиотеки в существующие и разрабатываемые игры.

Задача создания оптимизированной библиотеки для нахождения путей является актуальной проблемой рассмотренной в данной работе.

Анализ предметной области

Задача нахождения кратчайшего пути – поиск оптимального и короткого пути между двумя точками. Такая задача возникает при оптимизация перевозки грузов и пассажиров, навигация роботов, навигация ИИ и игрока в компьютерных играх.

Область поиска может быть представлена в разном виде, что влияет на выбор алгоритмов и их работу. В данной работе была выбрана квадратная сетка.

Анализ предметной области

Алгоритмы

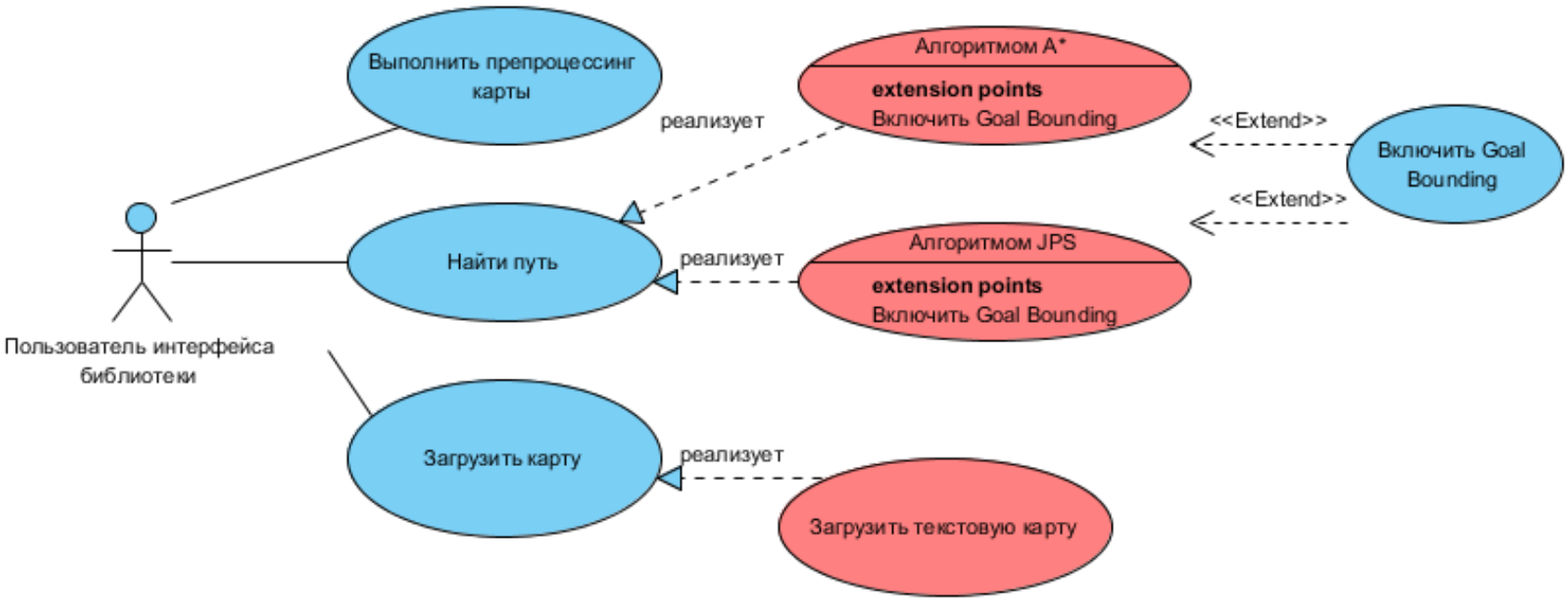
При анализе были выбраны следующие алгоритмы:

- A* – простой и универсальный алгоритм;
- Jump Point Search (JPS) – приспособлен для квадратной сетки, в 10 раз быстрее A*;
- Goal Bounding – алгоритм препроцессинга карты для ускорения A* и JPS.

Так же были рассмотрены алгоритмы поиска путей HPA*, HAA* и алгоритм препроцессинга RSR.

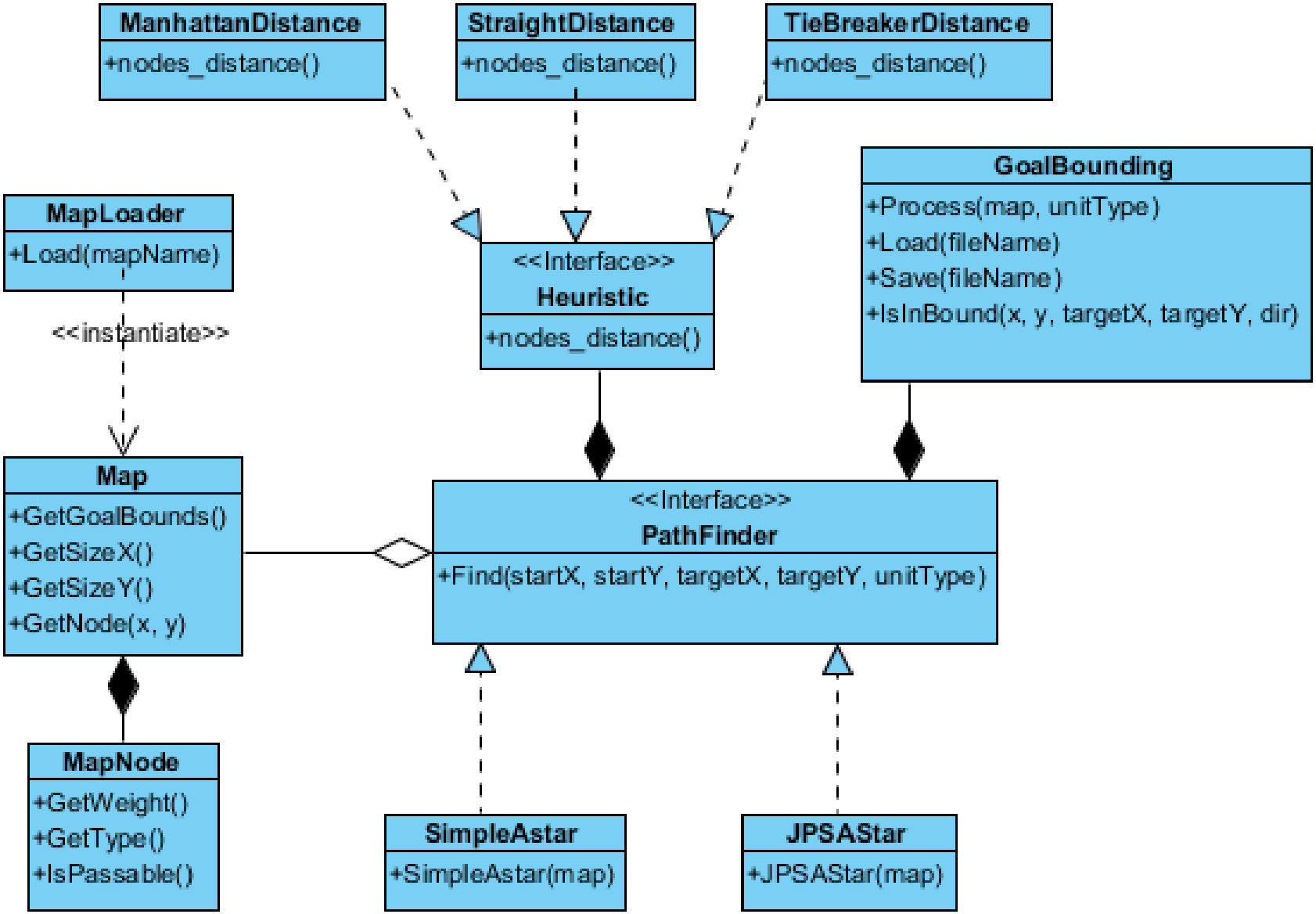
Проектирование

Диаграмма вариантов использования



Проектирование

Диаграмма классов



Алгоритм A*

A* является вариацией алгоритма Дейкстры и использует эвристическую функция для ускорения работы.

Алгоритм минимизирует функцию точки $f(n) = g(n) + h(n)$, где $g(n)$ – стоимость пути до точки, а $h(n)$ – эвристическая оценка стоимости прохождения до конца пути.

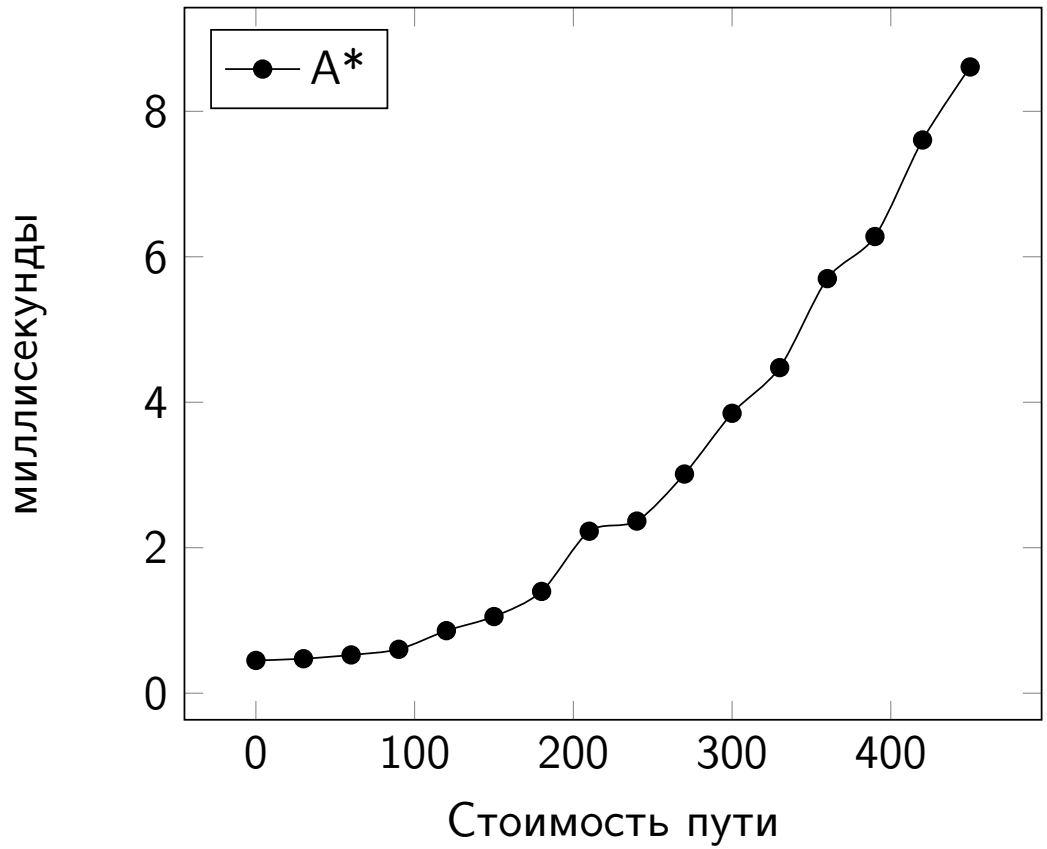
Алгоритм останавливается когда точкой с наименьшей стоимостью является конечная точка, или список для рассмотрения пуст.

Алгоритм A*

Визуализация

Алгоритм A*

Результаты на карте AR0011SR (512x512)

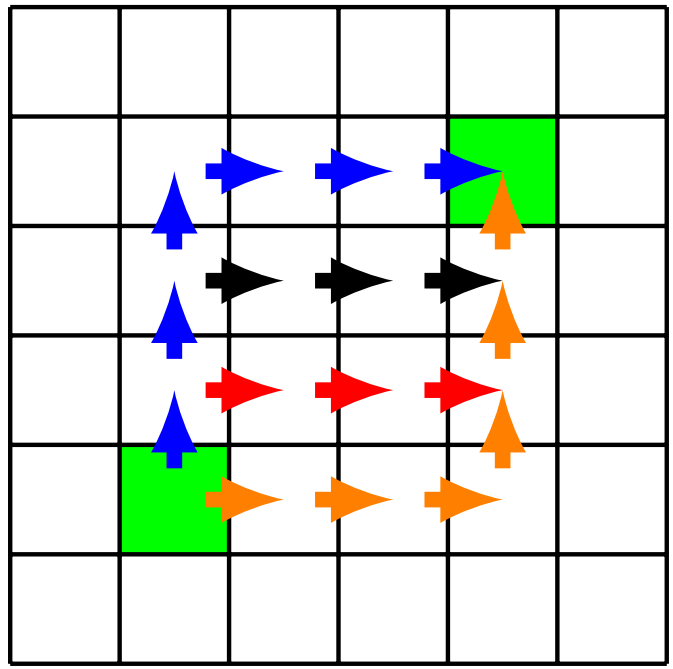


Длина	мс
0	0.45 мс
50	0.52 мс
100	0.74 мс
150	1.12 мс
200	2.15 мс
250	2.69 мс
300	3.90 мс
350	5.60 мс
400	7.03 мс
450	8.60 мс

По результатам видно, что время выполнения одного поиска пути для A* находится от 0.5 до 9 миллисекунд. Такое время слишком велико для игр.

Алгоритм JPS

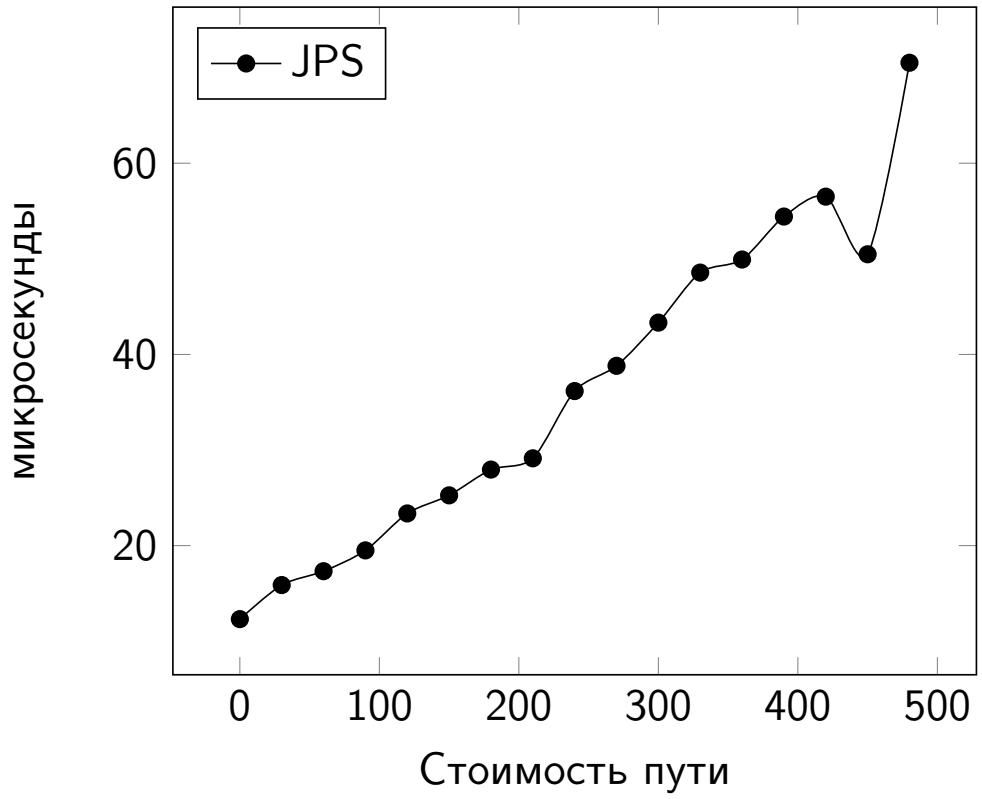
Jump Point Search – эффективная техника для нахождения и отброса симметричных путей, основана на алгоритме A*. Избавляет A* от необходимости рассматривать симметричные пути и точки, которые не могут входить в оптимальный маршрут.



Симметричные пути

Алгоритм JPS

Результаты на карте AR0011SR (512x512)



Длина	мс
0	0.060 мс
50	0.0895 мс
100	0.110 мс
150	0.145 мс
200	0.160 мс
250	0.232 мс
300	0.277 мс
350	0.379 мс
400	0.536 мс
450	0.668 мс

По результатам JPS на порядок быстрее A*.

Алгоритм Gaol Bounding

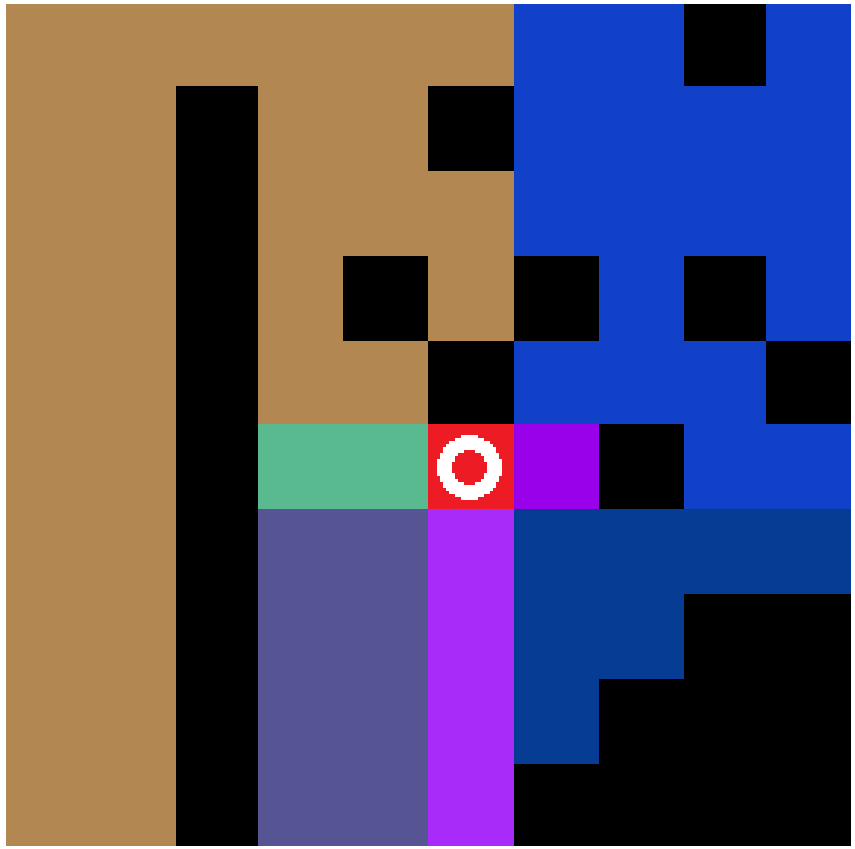
Goal Bounding – техника отброса заранее неподходящих направлений, которая позволяет значительно ускорить поиск пути. Включает оффлайн и онлайн этапы.

- Оффлайн этап – обработка карты с целью найти и сохранить для каждой точки направления с кратчайшими путями;
- Онлайн этап – проверка может ли являться путь в выдранном направлении кратчайшим.

Является крайне затратным по времени для карт размером более 512 на 512, однако очень хорошо поддаётся параллелизации.

Алгоритм Gaol Bounding

Визуализация



Каждый цвет показывает в каком направлении следует идти что бы дойти до покрашенной точки по кратчайшему пути.

Общее сравнение

Cost	A*	A*GB	от A*	JPS	от A*	JPS+GB	от A*
0	0,36	0,16	43,80%	0,024	06,50%	0,005	01,28%
60	0,48	0,28	58,10%	0,065	13,33%	0,012	02,45%
120	0,72	0,33	46,63%	0,103	14,27%	0,016	02,27%
180	1,16	0,39	33,87%	0,150	12,84%	0,021	01,76%
240	1,78	0,46	25,93%	0,205	11,48%	0,025	01,38%
300	2,63	0,53	20,28%	0,275	10,43%	0,029	01,12%
360	3,74	0,58	15,53%	0,358	09,57%	0,033	00,89%
420	4,85	0,54	11,31%	0,481	09,90%	0,039	00,80%
480	4,97	0,41	08,38%	0,569	11,44%	0,040	00,81%
540	5,32	0,45	08,52%	0,828	15,56%	0,050	00,93%

Алгоритм Goal Bounding ускоряет A* в 2 – 10 раз, а JPS в 5 – 15 раз. JPS с Goal Bounding быстрее обычного A* в 40 – 100 раз.

Выводы

Во время выполнения работы были проанализированы и реализованы алгоритмы нахождения кратчайшего пути A*, JPS и Goal Bounding. Была спроектирована, реализована и протестирована библиотека включающая указанные алгоритмы.

Из рассмотренных алгоритмов самым быстрым является JPS с Goal Bounding, однако при этом он является самым не универсальным.

Для карт с одинаковой стоимостью прохождения по клеткам наилучшим выбором оказался JPS.

Для карт с разной стоимостью прохождения по клеткам наилучшим будет алгоритм A* с или без Goal Bounding.