Министерство образования и науки Российской Федерации

ФГБ ОУ ВПО «Волгоградский государственный технический университет»

Кафедра Системы автоматизированного проектирования

и поискового конструирования

|  |  |
| --- | --- |
|  | Утверждаю  Зав. кафедрой САПР и ПК  М. В. Щербаков  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, инициалы и фамилия)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к выпускной работе бакалавра на тему

(наименование работы)

Разработка модуля генерации «2D прогулки»

Автор \_\_\_\_\_И.А.Волосникова\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

Обозначение ВРБ — 40 461 806 — 10.27— -- — 19.81

(код документа)

Группа ИВТ-465

(шифр группы)

Направление 09.03.01 — Информатика и вычислительная техника

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель работы О. А. Шабалина

(подпись и дата подписания) (инициалы и фамилия)

Консультанты по разделам:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(краткое наименование раздела) (подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(краткое наименование раздела) (подпись и дата подписания) (инициалы и фамилия)

Нормоконтролер\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Волгоград 2020

Аннотация

Данный документ содержит постановку цели и задач бакалаврской работы, анализ существующих решений по генерации 2D маршрута, описание проектирования и разработки модуля генерации «2D прогулки», экранные формы модуля и описание процесса тестирования.

Abstract

This document contains the statement of the goals and objectives of the bachelor’s work, the analysis of existing applications for generating a 2D route, a description of the design and development of the module for generating «2D walkthrough», module screen forms and a description of the testing process.

Оглавление

Введение 6

1. Анализ способов генерации 2D маршрута на карте закрытого пространства 7

1.1. Описание задачи генерации маршрута 7

1.2. Способы представления пространства 8

1.2.1. Сетка 8

1.2.2. Полигональная карта 9

1.2.3. Навигационная сетка 9

1.3. Анализ алгоритмов поиска пути 10

1.3.1. Алгоритмы поиска кратчайшего пути между двумя вершинами 10

1.3.2. Алгоритмы решения задачи коммивояжера 11

1.4. Решения по навигации в закрытом пространстве 14

1.4.1. Навигация в онлайн режиме 15

1.4.2. Навигация в офлайн режиме 15

1.4.2.1. Кунсткамера – мобильный гид 15

1.4.2.2. Мобильный гид по Музею А.С.Пушкина 17

1.4.2.3. Shopping Guide – карты торговых центров Москвы 18

1.4.2.4. NaviMail – навигация в торговых центрах 19

1.4.2.5. Бот «Картография» 21

1.5. Формулирование требований к приложению 21

1.6. Выводы 23

2. Разработка алгоритмов генерации «2D прогулки» 24

2.1. Разработка алгоритма построения маршрута 24

2.2. Разработка алгоритма визуализации маршрута 27

2.3. Разработка алгоритма моделирования прохождения маршрута 28

2.4. Выводы 28

3. Проектирование модуля генерации «2D прогулки» 28

3.1. Описание концепции модуля генерации «2D прогулки» 28

3.2. Описание формата входных и выходных данных 31

3.3. Проектирование графического интерфейса пользователя 32

3.4. Описание сценариев использования 32

3.5. Выводы 33

4. Реализация модуля генерации «2D прогулки» 33

4.1. Выбор средств реализации 33

4.2. Описание этапов реализации модуля 33

Заключение 34

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 35

# Введение

Одной из отличительных особенностей современной жизни является большое количество магазинов, торговых и культурных центров. Зачастую они занимают большую площадь, имеют много отделов или залов, и поэтому быстро сориентироваться в них и найти нужное место становится затруднительно. Для помощи в решении этих задач существуют интерактивные информационные стенды [1], системы указателей, справочные службы. Однако они предназначены в основном для людей, не имеющих ограничений по состоянию здоровья.

Для людей с физическими ограничениями трудности при взаимодействии с системами навигации постепенно уменьшаются: системы указателей дублируются с использованием шрифта Брайля, информационные стенды имеют функции голосовых подсказок. Но для людей с ограниченными интеллектуальными возможностями (People with Intellectual Disabilities, PID) этот аспект повседневной жизни все еще представляет трудности, поскольку некоторые группы PID не умеют читать и писать, многие испытывают трудности при использовании карт и коммуникации с незнакомыми людьми.

Наиболее распространенным решением для поддержки таких людей является их сопровождение и помощь в ориентировании в незнакомой обстановке. Однако сейчас все большее значение приобретает задача социализации PID, развитие и поддержка их самостоятельности. Многие PID стремятся к независимости от опекунов в повседневных действиях, активно участвуют в социальной жизни, используют новые технологии, в том числе и мобильные приложения. Поэтому другим способом помощи может быть использование специализированных мобильных приложений, разработанных для навигации в закрытом пространстве.

Ориентирование и составление маршрута в закрытом пространстве является частью многих действий: совершение покупок в супермаркете, посещение торгового центра, культурного или больничного комплекса. Поэтому является актуальной задача разработки модуля для генерации и визуализации маршрута на карте закрытого пространства, который можно использовать при создании конфигурируемых приложений для PID.

1. Анализ способов генерации 2D маршрута на карте закрытого пространства
   1. Описание задачи генерации маршрута

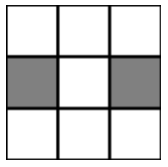
В общем виде задачу генерации маршрута на карте закрытого пространства можно определить следующим образом. Пространство представляется в виде 2D или 3D модели помещения, имеющего один или несколько входов и выходов. В пространстве находятся объекты – 2D или 3D модели предметов, имеющие собственные параметры и поведение. Для построения маршрута пространство формализуется в виде графа. Вершинам графа соответствуют точки в пространстве, расположенные в местах, свободных от объектов, а ребрам – путь между двумя точками, свободный от препятствий. Для определения расположения вершин существует несколько способов, и выбор одного из них зависит от способа представления пространства и поставленной задачи. Более подробно способы представления пространства рассмотрены в п.1.2.

Далее происходит генерация маршрута. Для его построения задаются начальная и конечная точки, а также – опционально – промежуточные точки, через которые должен проходить маршрут. Маршрут представляет собой ломаную линию, соединяющую заданные точки в обход препятствий, и строится на основании графа с использованием алгоритмов поиска оптимального пути. Если в маршруте есть промежуточные точки, порядок обхода которых не важен, задача генерации маршрута является незамкнутым вариантом задачи коммивояжера. Более подробно алгоритмы поиска оптимального пути и алгоритмы решения задачи коммивояжера рассмотрены в п.1.3. Затем построенный маршрут визуализируется на карте пространства.

Генерация маршрута может происходить в онлайн и офлайн режиме. В онлайн режиме доступно определение текущего местоположения и указание его в качестве начальной или конечной точки для маршрута, а также отслеживание прохождения маршрута. В закрытых помещениях онлайн режим реализуется с использованием дополнительного оборудования – Wi-Fi передатчиков или Bluetooth маячков. В режиме офлайн отслеживание текущего местоположения недоступно, прохождение маршрута моделируется в виде описания его участков (длина прямых участков, повороты) либо в виде виртуального тура (virtual tour). Более подробно существующие решения описаны в п.1.4.

* 1. Способы представления пространства
     1. Сетка

Сетка – это наиболее простой способ хранения информации о пространстве. Карта пространства разбивается на небольшие участки простой формы (например, квадраты или шестиугольники), каждый из которых может быть свободным для передвижения, а может являться препятствием (рис.1). Данный способ представления не подходит для больших и сложных пространств, так как требует большой объем памяти для хранения данных.



а) схематическое изображение



б) реализация карты

Рисунок 1 – Представление пространства в виде сетки

* + 1. Полигональная карта

В качестве полигональной карты может использоваться граф видимости. Для его построения препятствия представляются в виде полигонов, вершинами графа являются вершины полигонов, а ребрами – прямые пути между ними, не содержащие препятствий (рис.2). При таком представлении количество ребер может быть довольно большим (до n2 ребер для n вершин), что делает метод очень затратным, однако, в сравнении с сеткой, большое количество ребер может ускорить процесс поиска пути.



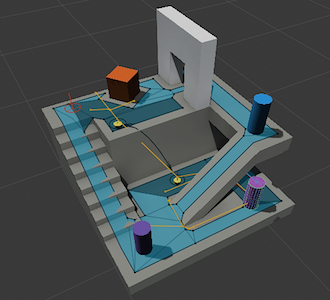
Рисунок 2 – Представление пространства в виде полигональной карты

* + 1. Навигационная сетка

В этом варианте представления для обозначения доступного для передвижения пространства используются непересекающиеся полигоны. Для формализации пространства в виде графа достаточным набором точек являются центры полигонов, однако это не всегда удобно, поэтому могут быть использованы точки на гранях полигонов или в их вершинах, а также скомбинированы все перечисленные варианты (рис.3).



а) схематическое изображение



б) реализация карты

Рисунок 3 – Представление пространства в виде навигационной сетки

* 1. Анализ алгоритмов поиска пути
     1. Алгоритмы поиска кратчайшего пути между двумя вершинами

Алгоритм Дейкстры позволяет найти кратчайший путь от одной вершины графа до всех остальных, применяется только для графов без отрицательных ребер.[1] Алгоритм начинает работу со стартовой точки, на каждом шаге рассматривается ближайшая нерассмотренная вершина, а все смежные с ней добавляются к множеству вершин, подлежащих рассмотрению. Алгоритм обеспечивает нахождение кратчайшего пути от стартовой до целевой вершины. Данный алгоритм считается одним из самых простых и подходит для графов с небольшим количеством вершин.

Алгоритм Беллмана-Форда также предназначен для поиска кратчайшего пути во взвешенном графе, однако, в отличие от алгоритма Дейкстры, допускает наличие ребер с отрицательным весом. Алгоритм использует полный перебор всех вершин графа, что приводит к большим затратам времени и памяти.

Алгоритм Флойда-Уоршелла является динамическим алгоритмом для поиска кратчайших расстояний между всеми вершинами взвешенного ориентированного графа. При применении его к решению задач на значительно разветвленном графе требуются значительные затраты вычислительных ресурсов.

Алгоритм Ли, или волновой алгоритм, работает на дискретном рабочем поле, представляющем собой ограниченную замкнутой линией фигуру, разбитую на прямоугольные ячейки. Множество всех ячеек разбивается на проходимые, непроходимые, стартовые и финишные. Алгоритм предназначен для поиска кратчайшего пути от стартовой ячейки к финишной, если он существует.

Алгоритм поиска А\* позволяет найти кратчайший путь от стартовой вершины к целевой, используя алгоритм поиска по первому наилучшему совпадению на графе.[2] Алгоритм А\* является расширением алгоритма Дейкстры, более высокая производительность достигается за счет того, что порядок обхода вершин определяется эвристической функцией f(x) = h(x) + g(x), где h(x) – функция эвристической оценки расстояния от рассматриваемой вершины к конечной, g(x) – функция стоимости достижения рассматриваемой вершины из начальной (может быть как эвристической, так и нет). Алгоритм А\* обеспечивает нахождение оптимального пути. Временная сложность алгоритма зависит от эвристики. В худшем случае, число вершин, исследуемых алгоритмом, растет экспоненциально по сравнению с длиной оптимального пути.

* + 1. Алгоритмы решения задачи коммивояжера

Алгоритмы для решения задачи коммивояжера можно разделить на точные и неточные. Точные алгоритмы включают в себя перебор всех возможных вариантов, при небольшом количестве вершин решения могут быть найдены быстро, однако при увеличении количества вершин время работы точных алгоритмов возрастает экспоненциально.[3] Алгоритмы неточной группы обычно находят потенциально неоптимальные решения, но за значительно меньшее время.

Один из самых очевидных методов решения задачи коммивояжера – метод полного перебора или грубой силы, заключающийся в рассмотрении всех вариантов путей (гамильтоновых контуров) и нахождения пути с минимальным весом. Данный метод гарантирует решение задачи, однако неэффективен при работе с большим объемом данных, т.к. в нем требуется найти вес (n – 1)! гамильтоновых контуров, где n – число вершин графа.[4]

Метод ветвей и границ часто используется для нахождения оптимального решения задач комбинаторной оптимизации. Одним из главных недостатков метода является необходимость целиком решать задачи линейного программирования во всей области допустимых решений.[5] Несмотря на то, что алгоритм является надежным методом нахождения оптимального решения, для задач с большим объемом данных метод ветвей и границ является очень трудоемким.

Алгоритм Гомори, или метод отсечений, используется для построения точных или приближенных задач, особенно часто встречается в сочетании с методом ветвей и границ и тогда называется методом ветвей и отсечений. [6] Оба метода основаны на решении последовательности подзадач линейного программирования, однако метод отсечений требует меньшего объема памяти для хранении дерева решений.

Общим для всех описанных выше алгоритмов является высокая трудоемкость и значительное увеличение времени, объема памяти и вычислительных ресурсов, необходимых для нахождения решения, при увеличении числа вершин в исходном графе.

Самый распространенный способ измерить эффективность эвристического алгоритма для решения задачи коммивояжера – это сравнить результаты с нижней гранью Хелд-Карпа, т.е. сравнить с решением, найденным за полиномиальное время при помощи симплекс-метода. Это решение на 0,8 % хуже оптимального маршрута.[7]

Одним из самых простых эвристических методов решения задачи коммивояжера является алгоритм ближайшего соседа. Главное правило алгоритма – всегда выбирать ближайшую к текущей вершину. В общем случае трудоемкость решения не превышает O(n2), где n – количество вершин в графе. Нижняя граница стоимости оптимального маршрута на 10% выше нижней грани Хелд-Карпа.[8]

Важным недостатком алгоритма ближайшего соседа является то, что в качестве оптимального часто принимается локально оптимальное решение. Этой проблемы позволяет избежать алгоритм поиска с запретами. Данный метод позволяет переходить от одного локально оптимального решения к другому в поиске глобального оптимума, после перехода ребро попадает в список запретов и повторно не используется, кроме тех случаев, когда оно может улучшить построенный оптимальный путь. На практическом уровне запрещенный набор сохраняется как комбинация ранее использованных ребер, которая позволяет построить дальнейший путь относительно текущего решения и соседних узлов. [9] Главным недостатком этого метода служит его время выполнения: трудоемкость алгоритма оценивается как О(n3), где n – количество вершин в графе. [10]

Еще одним распространенным решением является жадный алгоритм. Суть алгоритма заключается в переборе всех возможных ребер графа, исходящих из конкретной вершины. Среди рассматриваемых ребер выбирается то, у которого минимален вес перехода в другую вершину графа. После чего происходит сам переход, и алгоритм повторяется из новой вершины. Процедура работает до тех пор, пока не будет построен гамильтонов цикл, который и считается оптимальным маршрутом. Вес найденного маршрута выше нижней границы Хелд-Карпа на 15-20%. [10]

В основе генетического алгоритма лежат природные процессы естественного отбора, в ходе которого выживают особи с наибольшей приспособленностью к условиям окружающей среды. Задача оптимизации обозначается как задача нахождения функции приспособленности. В случае применения генетического алгоритма для решения задачи поиска оптимального пути, характеристикой приспособленности является мера длины маршрута. Существует несколько реализаций классического генетического алгоритма, различающихся подходом к реализации его шагов. Также как и в алгоритме поиска с запретами, при большом количестве узлов значительно возрастает трудоемкость алгоритма.

Исходя из сравнения алгоритма ближайшего соседа, генетического и жадного алгоритма [11], генетический алгоритм можно назвать более точным для небольшого объема данных, однако его трудоемкость и время, необходимое для поиска решения, значительно превышают те же показатели для двух других алгоритмов. При увеличении числа вершин графа генетический алгоритм проигрывает двум другим алгоритмам по всем параметрам. Результаты сравнения алгоритмов представлены в Таблице 1.

Таблица 1 – Сравнение алгоритма ближайшего соседа, генетического и жадного алгоритмов при решении задачи коммивояжера

|  |  |  |  |
| --- | --- | --- | --- |
| Алгоритм | Длина оптимального маршрута | Затраченное время | Количество итераций |
| Количество вершин - 100 | | | |
| Алгоритм ближайшего соседа | 26664 | 2,5 | 100 |
| Генетический алгоритм | 25479 | 45 | 10000 |
| Жадный алгоритм | 23311 | 0,07 | 18 |
| Количество вершин - 1000 | | | |
| Алгоритм ближайшего соседа | 83938 | 95,5 | 1000 |
| Генетический алгоритм | 282866 | 468 | 10000 |
| Жадный алгоритм | 72801 | 127 | 151 |

* 1. Решения по навигации в закрытом пространстве
     1. Навигация в онлайн режиме

Технология навигации и обнаружения объектов (устройств, людей) внутри зданий носит название indoor-навигации. В закрытых помещениях использование спутниковой системы навигации (GPS) недоступно: ослабление и рассеяние сигнала приводит к низкой точности позиционирования, к которой внутри зданий предъявляются повышенные требования. Вместо этого в помещении устанавливаются Wi-Fi передатчики или Bluetooth маяки, определение местоположения устройства осуществляется путем обработки их сигнала, а также данных от внутренних датчиков смартфона – гироскопа, акселерометра, барометра, компаса. Возможно использование двух методов определения местоположения: трилатерация (для этого указывается расположение маяков) или метод «цифровых отпечатков» (для этого производится замер радиокарты – в разных точках карты измеряется мощность сигнала от окружающих маяков, чтобы затем можно было сравнить значения в реальном времени со значениями, привязанными к карте помещения). Такой способ навигации требует установки дополнительного аппаратного обеспечения в помещениях, поэтому не будет рассматриваться в рамках данной работы.

* + 1. Навигация в офлайн режиме

Решения по офлайн навигации в закрытом пространстве разрабатываются в основном для объектов культуры, торговых комплексов или складских помещений. Последняя категория предназначена для решения специализированных задач логистики, не принадлежащих к повседневным задачам ориентирования в закрытом пространстве, поэтому не будет рассмотрена в рамках данной работы.

* + - 1. Кунсткамера – мобильный гид

Данный мобильный гид предоставляет информацию о расположении залов и экспонатов в музее, описание и изображения экспонатов.[12] В приложении есть возможность изучить карту всех этажей музея, прослушать аудио экскурсию по выбранной экспозиции, однако нет функции поиска экспоната или зала, а также нет возможности проложить маршрут между точками внутри музея. Для использования приложения необходим навык чтения. Экранные формы приложения представлены на рис.4 – 5.



Рисунок 4 – Карта музея

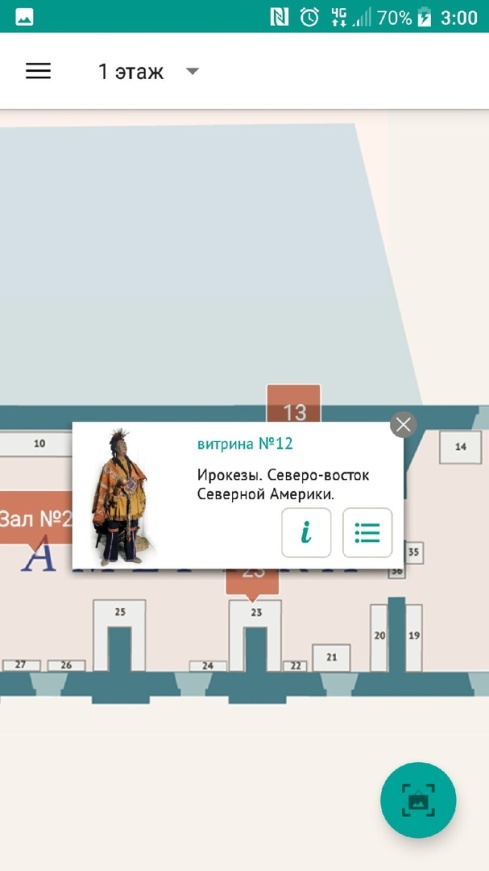


Рисунок 5 – Просмотр информации по экспонату

* + - 1. Мобильный гид по Музею А.С.Пушкина

В данном мобильном гиде можно найти карту музея с указанием расположения залов, есть возможность поиска объектов (в т.ч. голосовой поиск), а также прокладывания маршрута между двумя залами с текстовым описанием прохождения и отображением на карте.[13] Нет возможности проложить маршрут через несколько точек. Для использования приложения необходим навык чтения. Экранные формы приложения представлены на рис.6 – 7.

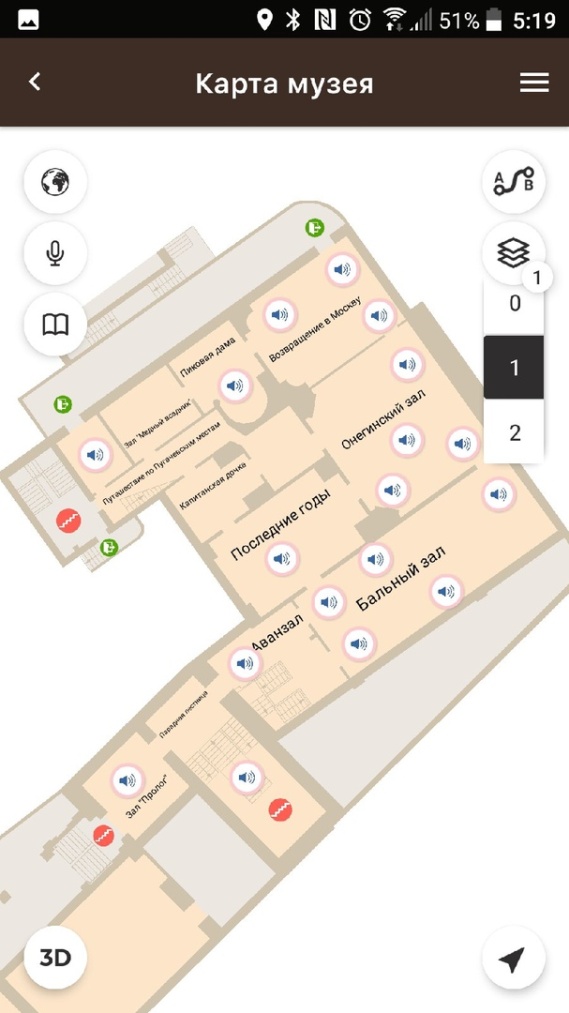


Рисунок 6 – Карта музея

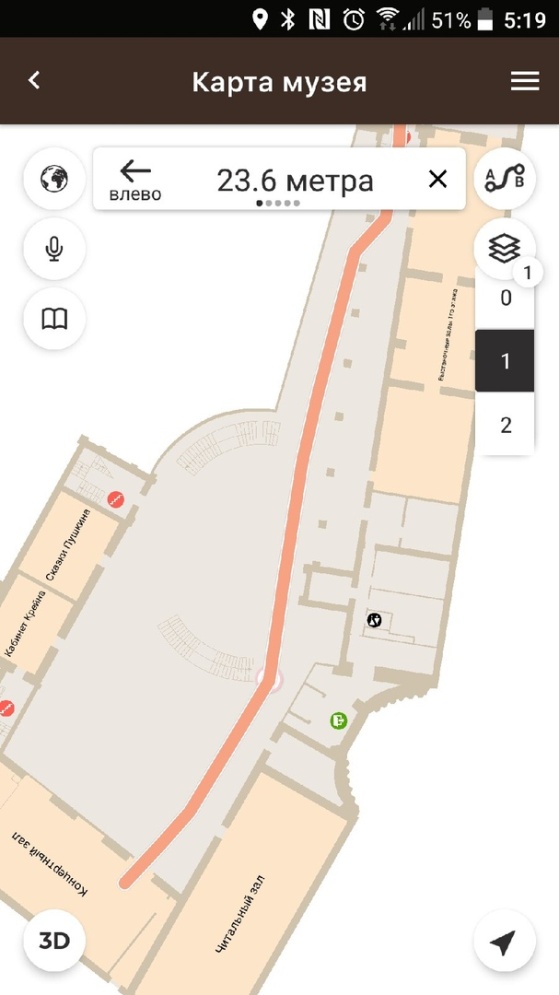


Рисунок 7 – Прокладывание маршрута

* + - 1. Shopping Guide – карты торговых центров Москвы

Мобильное приложение Shopping Guide содержит карты торговых центров Москвы.[14] Доступны функции поиска объекта на карте, построение маршрута между двумя точками, отображение маршрута на карте. Описание прохождения маршрута и разбиение его на участки недоступно. Приложение содержит карты 45 торговых центров. Для использования приложения необходим навык чтения. Экранные формы приложения представлены на рис.8 – 9.

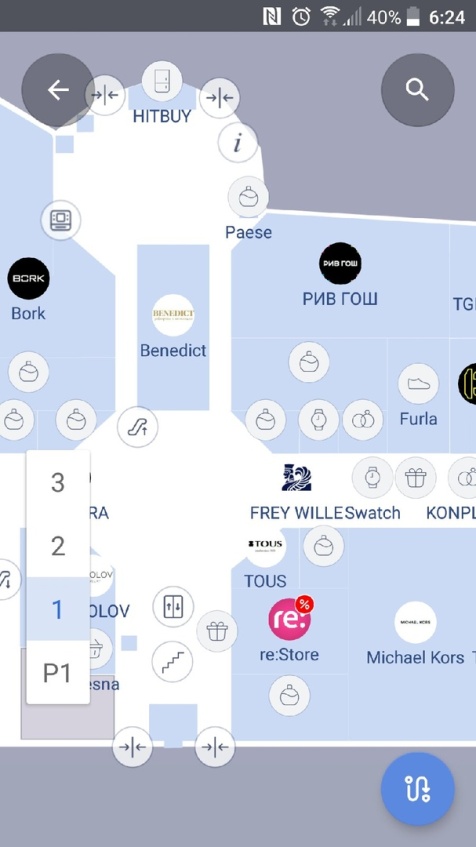


Рисунок 8 – Карта торгового центра

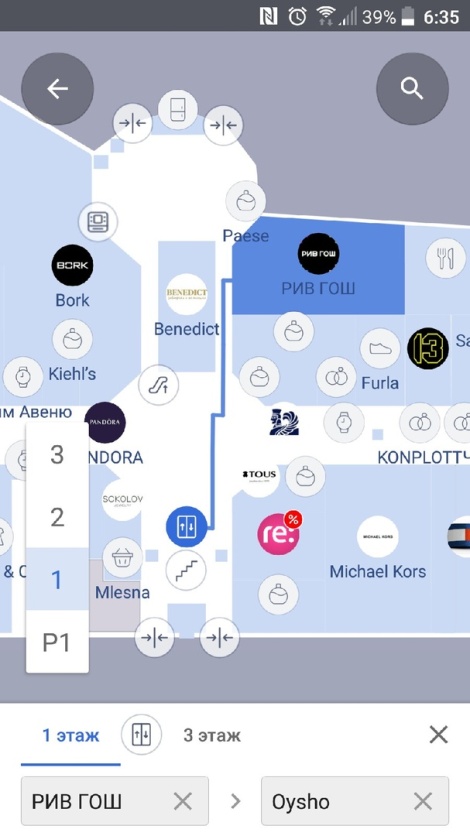


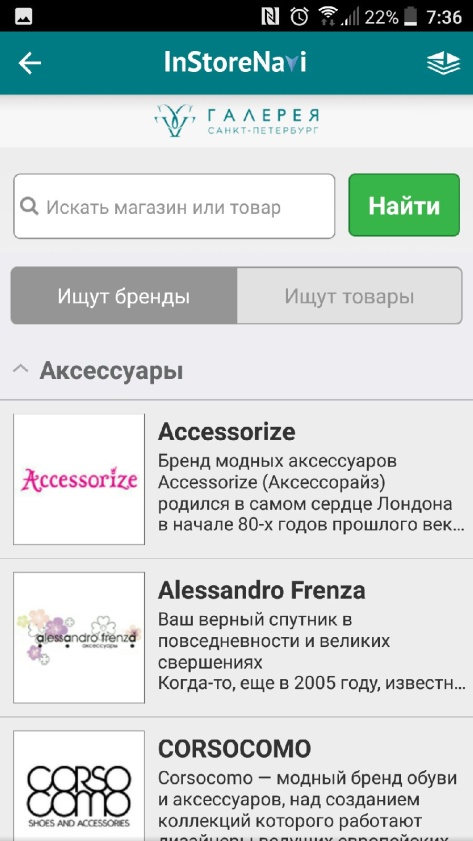
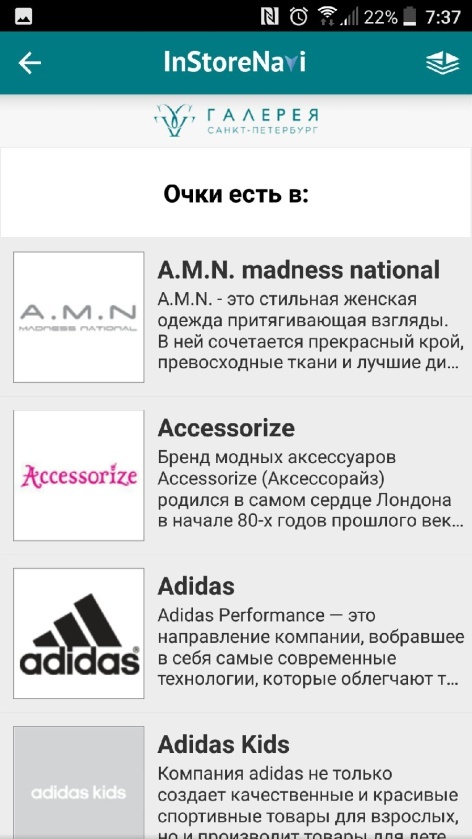
Рисунок 9 – Прокладывание маршрута

* + - 1. NaviMail – навигация в торговых центрах

Мобильное приложение содержит карты двух торговых центров Москвы.[15] Доступна функция поиска объекта на карте (по названию и по типу товаров), однако функция прокладывания маршрута недоступна. Для использования приложения необходим навык чтения. Экранные формы приложения представлены на рис.10 – 11.



Рисунок 10 – Карта торгового центра

а) поиск по названию б) поиск по товару

Рисунок 11 – Поиск объекта

* + - 1. Бот «Картография»

«Картография» – это текстовый бот, разработанный для ВКонтакте, Facebook и Telegram, и предназначенный для навигации внутри торговых центров.[16] Доступны функции просмотра карты 5 торговых центров, поиска объекта по названию и категории, построение маршрута между двумя объектами, визуализация его на карте и текстовое описание его прохождения. Для использования бота необходимы навыки чтения и письма. Экранная форма бота представлена на рис.12.

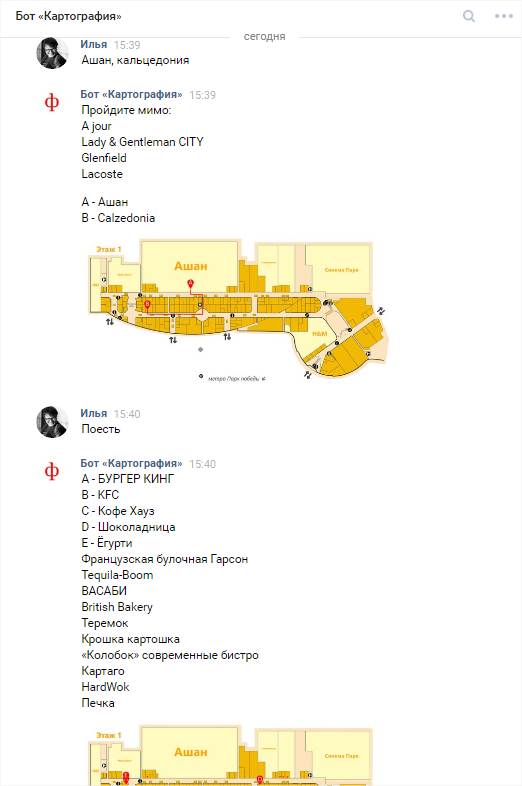


Рисунок 12 – Поиск объекта и построение маршрута

* 1. Формулирование требований к приложению

Анализ существующих решений по генерации 2D маршрута на карте закрытого пространства показал, что основными функциями приложений являются:

* поиск объекта на карте пространства;
* генерация маршрута, проходящего через два и более объекта;
* визуализация маршрута в виде линии на карте;
* разделение маршрута на участки и последовательное отображение участков;
* описание маршрута (словесное или пиктографическое).

Процесс генерации, визуализации и моделирования прохождения маршрута на карте 2D пространства обозначим как «2D прогулка» («2D walkthrough»).

Применение приложений для поддержки ориентирования PID накладывает несколько ограничений, связанных с особенностью данной категории пользователей. Во-первых, многие категории PID не владеют навыками чтения и письма, поэтому эти навыки не должны быть обязательными для использования приложения. Во-вторых, приложение должно быть доступно для использования в режиме офлайн, чтобы обеспечить возможность обучения PID навыку ориентирования дома или в медицинском учреждении, под контролем опекуна. Таким образом, PID смогут подготовиться к самостоятельному посещению общественных мест и ориентированию в них, а также приобрести базовые навыки работы с картой. В-третьих, должна быть возможность прокладывать маршрут, проходящий через несколько точек (например, на основании списка покупок). Для многих категорий PID самостоятельное посещение общественных мест может вызвать волнение, долгое концентрирование внимания на одной задаче (следование маршруту) может быть затруднительным, поэтому прокладываемый маршрут должен быть как можно более коротким, чтобы сократить время его прохождения.

Исходя из описанных требований, оценим существующие решения по следующим критериям:

* навыки, необходимые для использования;
* режим работы (онлайн или офлайн);
* возможность составления простого (между двумя точками) и сложного (проходящего через три и более точки) маршрута.

Результаты анализа представлены в Таблице 2.

Таблица 2 – Анализ решений по генерации 2D маршрута

|  |  |  |  |
| --- | --- | --- | --- |
| Название решения | Критерии анализа решений | | |
| Навыки, необходимые для использования | Режим работы | Функция составления маршрута |
| Кунсткамера – мобильный гид | Чтение | Оффлайн | Нет |
| Мобильный гид по Музею А.С.Пушкина | Чтение | Онлайн и оффлайн | Простой маршрут |
| Shopping Guide | Чтение | Оффлайн | Простой маршрут |
| NaviMail | Чтение | Онлайн и оффлайн | Нет |
| Бот «Картография» | Чтение и письмо | Онлайн | Простой маршрут |

* 1. Выводы

При анализе существующих решений по навигации в закрытом пространстве было выявлено, что некоторые из них имеют только функцию поиска объекта на карте, но не позволяют построить маршрут. Те приложения, которые имеют функцию построения маршрута, позволяют строить его только между двумя точками, без промежуточных остановок. Функция построения сложного маршрута является ключевой для приложения генерации «2D прогулки», поскольку позволит PID заранее изучить весь путь и чувствовать себя увереннее при реальном посещении общественных мест.

Также использование всех рассмотренных решений требует владения навыком чтения, что делает их недоступными или очень сложными для многих категорий PID.

Таким образом, задача разработки модуля для мобильного приложения, позволяющего генерировать и визуализировать 2D маршрут на карте закрытого пространства, а также моделировать его прохождение, с адаптированным для PID интерфейсом, является актуальной.

Целью данной работы является разработка модуля генерации «2D прогулки».

Для достижения поставленной цели необходимо решить следующие задачи:

* разработать алгоритм построения оптимального маршрута;
* разработать алгоритм визуализации маршрута;
* разработать алгоритм моделирования прохождения маршрута;
* разработать интерфейс модуля, ориентированный на PID, не владеющих навыками чтения и письма;
* реализовать модуль генерации «2D прогулки»;
* протестировать полученный модуль и, при необходимости, исправить выявленные при тестировании недостатки.

1. Разработка алгоритмов генерации «2D прогулки»
   1. Разработка алгоритма построения маршрута

Основная функция модуля – построение маршрута по списку заданных точек. Общий вид алгоритма построения маршрута представлен на рис.13.

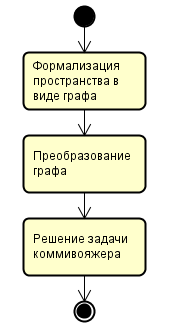


Рисунок 13 – Алгоритм построения маршрута

Первым этапом реализации данной функции является формализация пространства в виде взвешенного графа. Алгоритм формализации представлен на рис.14. Всем объектам, находящимся в пространстве, ставятся в соответствие плоские горизонтальные полигоны. Множество вершин графа составляют вершины полигонов и точки, заданные для построения маршрута. Далее вершины графа соединяются ребрами в том случае, если между точками на карте, которым соответствуют вершины, существует прямой путь, не пересекающий объекты-препятствия. Пары вершин для соединения ребром выбираются таким образом, чтобы полученный граф являлся плоским. Весом ребра является длина пути между точками. Таким образом, все пространство представляется в виде навигационной сетки.

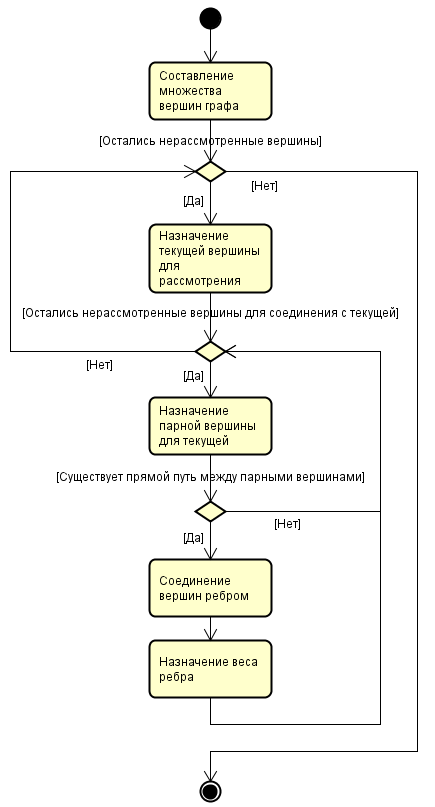


Рисунок 14 – Алгоритм формализации пространства в виде графа

Затем полученный граф преобразуется таким образом, чтобы во множестве вершин остались только точки, заданные для построения маршрута. Для этого между каждой парой точек находится оптимальный путь в исходном графе, и они соединяются ребром с весом, равным длине найденного пути. Исходя из анализа алгоритмов нахождения оптимального пути между двумя точками, произведенного в п.1.3.1, наиболее эффективным является алгоритм А\*. Если задано несколько альтернативных начальных и/или конечных точек, ребра строятся только для соединения каждой из них с промежуточными точками маршрута, но не между собой. Алгоритм преобразования представлен на рис.15.

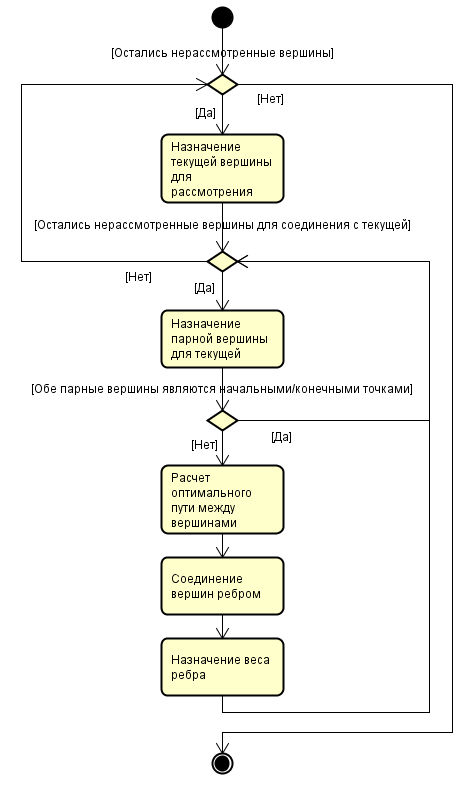


Рисунок 15 – Алгоритм преобразования графа

Для полученного графа решается незамкнутая задача коммивояжера на подмножестве вершин, которым соответствуют промежуточные точки маршрута. При этом общая длина пути рассчитывается вместе с длиной пути от начала обхода до ближайшей начальной точки и от конца обхода до ближайшей конечной точки. Поскольку при разработке модуля необходимо учитывать не только оптимальность найденного маршрута, но и быстродействие модуля, обеспечивающее удобство его использования, на данном этапе не применимы точные методы нахождения оптимального пути, т.к. они требуют значительных временных затрат. Согласно анализу, проведенному в п.1.3.2, наиболее подходящим для решения данной задачи является жадный алгоритм.

Таким образом, строится близкий к оптимальному маршрут через все указанные точки.

* 1. Разработка алгоритма визуализации маршрута

Построенный маршрут разбивается на участки – отрезки пути между двумя соседними точками маршрута. Во избежание рассеяния внимания пользователя при работе с картой, маршрут отображается не целиком, а только его текущий участок. Отрезок пути, соответствующий участку маршрута, изображается яркой контрастной линией на карте пространства. Конец участка маршрута отображается ярким маркером для удержания внимания пользователя.

Поскольку ребрам между каждой парой точек в модифицированном графе, получение которого описано в п.2.1, соответствует оптимальный путь, найденный на исходном графе, участок пути представляет собой ломаную линию, отрезками которой являются ребра, составляющие найденный оптимальный путь между двумя точками.

Переключение между участками маршрута происходит тогда, когда в процессе моделирования прохождения маршрута пользователь достигает конечной точки участка. Пройденный участок маршрута и его конечная точка удаляются с карты, затем отображается следующий участок маршрута и его конечная точка.

* 1. Разработка алгоритма моделирования прохождения маршрута

Для моделирования прохождения маршрута на карту пространства добавляется игровой персонаж. Игровой персонаж – это подвижный объект, перемещением которого управляет пользователь. Игровой персонаж необходим для того, чтобы пользователь мог отождествить себя с ним и таким образом изучить и запомнить порядок прохождения маршрута.

Управление персонажем осуществляется с помощью кнопок движения вперед, назад, влево и вправо. Длина пути, пройденного персонажем, запоминается. После достижения игровым персонажем конечной точки текущего участка маршрута пользователю выводится сообщение с длиной пройденного пути, длиной оптимального пути и графической оценкой успешности прохождения (например, улыбающийся смайлик, если длина пройденного пути не превышает оптимальный более чем на 50%, и грустный смайлик в обратном случае). Численная информация предназначена для опекуна пользователя, который может контролировать процесс изучения маршрута, а графическая – для самого пользователя. После достижения конечной точки последнего участка маршрута пользователю выводится аналогичное сообщение с общей оценкой прохождения маршрута.

* 1. Выводы

Разработанные алгоритмы позволяют реализовать три основные функции модуля: генерацию маршрута, близкого по длине к оптимальному, визуализацию его на карте пространства и моделирование его прохождения.

Для привлечения интереса PID и концентрации на задаче ориентирования, моделирование прохождения маршрута было решено реализовать в виде игры с управляемым персонажем.

1. Проектирование модуля генерации «2D прогулки»
   1. Описание концепции модуля генерации «2D прогулки»

Модуль предназначен для развития и поддержки навыка ориентирования в закрытом пространстве с помощью карты. Пространство представляет собой модель помещения с одним или несколькими входами и выходами, которое заполнено объектами. Модель пространства представлена на рис.16.

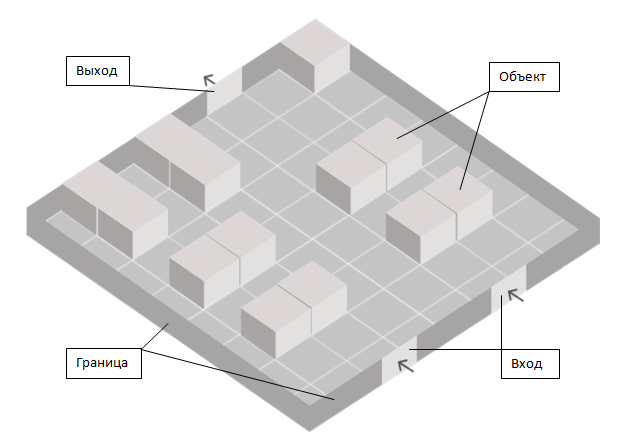


Рисунок 16 – Модель пространства

Задача ориентирования состоит из следующих этапов:

* задание точек, которые необходимо обойти;
* генерация маршрута от входа к выходу, проходящего через заданные точки;
* отображение маршрута на карте;
* прохождение маршрута.

Модуль предназначен для интеграции с мобильными приложениями для PID, которые могут осуществлять поддержку различных повседневных процессов и обучение разным навыкам. Поскольку задача ориентирования в закрытом пространстве является частью многих процессов, генерация маршрута должна осуществляться на любом заданном пространстве и для любого списка точек обхода. Таким образом, идентификатор пространства и список точек, через которые необходимо построить маршрут, должны быть входными данными для модуля.

Объекты на карте пространства могут иметь название, в таком случае через них может быть проложен маршрут. Если объект не имеет названия, он является препятствием и не может быть включен в маршрут. Для задания списка точек используются названия объектов.

Модуль должен хранить модели доступных пространств. Добавление новых моделей возможно двумя способами: вручную или с помощью специально разработанного редактора. Реализация приложения для добавления моделей пространства не входит в задачи данной работы, поэтому будет использоваться только ручное добавление.

Построение маршрута и разбиение его на участки должно осуществляться до отображения карты пространства пользователю. Во время взаимодействия пользователя с модулем на карте пространства должен отображаться только текущий участок маршрута. Модель карты пространства с размеченным маршрутом представлена на рис.17.

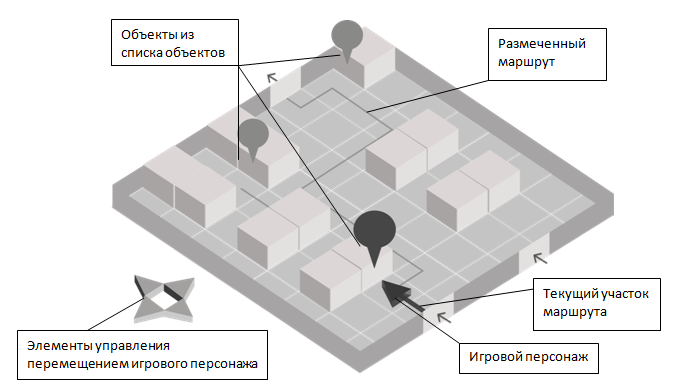


Рисунок 17 – Модель карты пространства

После завершения прохождения маршрута и вывода информации о результате прохождения, модуль должен сохранять следующие данные о прохождении маршрута:

* дата и время начала прохождения маршрута;
* дата и время окончания прохождения маршрута;
* идентификатор пространства;
* описание сгенерированного маршрута в виде порядка точек обхода;
* общую результативность прохождения (соотношение длины пути, пройденного пользователем в процессе моделирования прохождения маршрута, к длине построенного оптимального (близкого к оптимальному) пути).
  1. Описание формата входных и выходных данных

Входными данными для модуля являются:

* идентификатор пространства: строка;
* названия точек обхода: список строк.

В модуле должны быть реализованы два способа передачи входных данных: в качестве параметров функции вызова модуля и с помощью текстового файла в формате JSON. Во втором случае файл должен иметь название «input.json» и содержать данные в следующем формате:

{

«space»: <идентификатор пространства>,

«waypoints»:

[

<название точки 1>,

< название точки 2>,

…

< название точки N>

]

}

Выходными данными для модуля являются:

* дата начала прохождения маршрута: дата;
* время начала прохождения маршрута: время;
* дата окончания прохождения маршрута: дата;
* время окончания прохождения маршрута: время;
* идентификатор пространства: строка;
* описание маршрута: список строк;
* длина сгенерированного пути: десятичная дробь;
* длина пути, пройденного пользователем: десятичная дробь.

Выходные данные должны быть сохранены в текстовый файл формата JSON

* 1. Проектирование графического интерфейса пользователя
  2. Описание сценариев использования

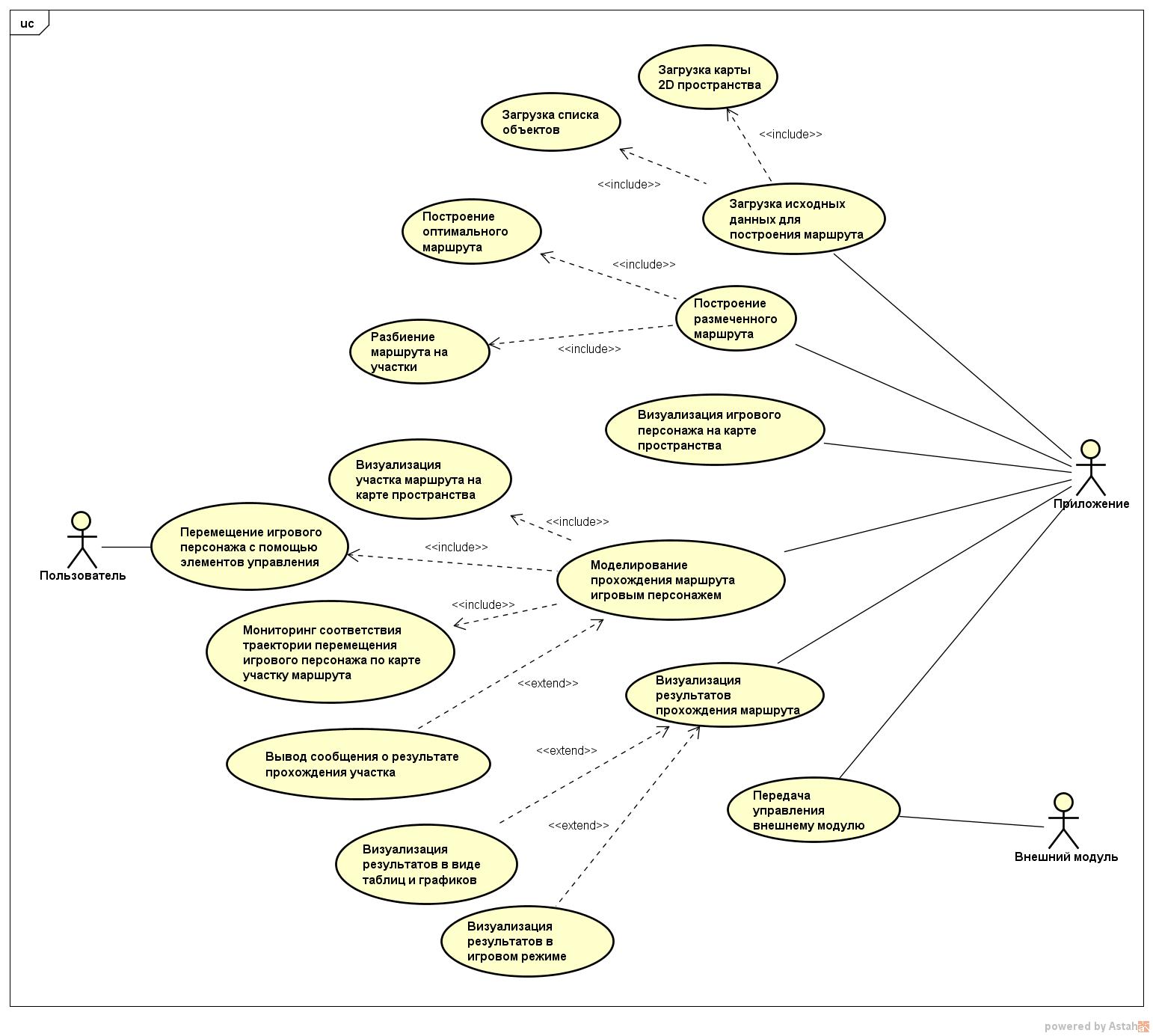


Рисунок 18 – Диаграмма прецедентов

* 1. Выводы

1. Реализация модуля генерации «2D прогулки»
   1. Выбор средств реализации
   2. Описание этапов реализации модуля

(Первый этап – основные функции (нахождение и визуализация маршрута, моделирование его прохождения), второй – создание интерфейса (выбор тестового примера(?), добавление экранов результата прохождения участка и всего маршрута), третий этап - релиз – графическое оформление (добавление моделей объектов для имитации обстановки, например, магазина))

* 1. Создание первой версии модуля
  2. Создание второй версии модуля
  3. Создание релизной версии модуля
  4. Выводы

1. Тестирование модуля генерации «2D прогулки»

# Заключение

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Directorix. Интерактивная навигация [Электронный ресурс]. – Режим доступа: http://directorix.ru/functionality/ (дата обращения: 01.12.2019)
2. Изотова Т.Ю. Обзор алгоритмов поиска кратчайшего пути в графе // Новые информационные технологии в автоматизированных системах. 2016. №19.
3. Introduction to the A\* Algorithm [Электронный ресурс]. – Режим доступа: https://www.redblobgames.com/pathfinding/a-star/introduction.html (дата обращения: 04.12.2019).
4. Левитин А. Алгоритмы. Введение в разработку и анализ. Вильямс, 2006. 35–36 с.
5. Tannenbaum P. Excursions in Mathematics. University of Kansas, 2011. P. 25.
6. Tollis I. G. Algorithms and Complexity. University of Crete, 2000. P. 140–146.
7. Захарова Е.М., Минашина И.К. Обзор методов многомерной оптимизации // Информационные процессы, 2014. Том 14, № 3. стр. 265– 266.
8. Johnson D. S., McGeoch L. A., Rothberg E. E. Asymptotic Experimental Analysis for the Held-Karp Traveling Salesman Bound. AT&T Bell Laboratories, 1999 P. 1–2.
9. Alsalibi B.A., Jelodar M.B., Venkat I. A Comparative Study between the Nearest Neighbor and Genetic Algorithms: A revisit to the Traveling Salesman Problem // International Journal of Computer Science and Electronics Engineering (IJCSEE), 2013. Vol. 1, Issue 1.
10. Basu S. Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey. Indian Institute of Management Calcutta, 2012. P. 1–8.
11. Nilsson C. Heuristics for the Traveling Salesman Problem. Linkoping University, 2011. P. 1–6.
12. Abdulkarim H.A., Alshammari I. F. Comparison of Algorithms for Solving Traveling Salesman Problem. // International Journal of Engineering and Advanced Technology, 2015. Vol.4, Issue 6. P. 76–79.
13. Кунсткамера. Гид по музею [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=com.msl.kunstkamera (дата обращения: 10.12.2019).
14. Музей Пушкина музею [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=pro.indoorsnavi.puskinmuseum (дата обращения: 10.12.2019).
15. Shopping Guide – карты торговых центров Москвы [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=com.infoshopping.app (дата обращения: 12.12.2019).
16. NaviMail – навигация в торговых центрах [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=com.smartlabs.instorenavi (дата обращения: 13.12.2019).
17. Бот «Картография» [Электронный ресурс]. – Режим доступа: https://vk.com/kartografia\_bot (дата обращения: 13.12.2019).