Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Кафедра «Системы автоматизированного проектирования  
и поискового конструирования»

|  |  |  |
| --- | --- | --- |
| Согласовано |  | Утверждаю |
|  |  | Зав. кафедрой САПР и ПК |
|  |  |  |
|  |  | М.В. Щербаков |
| (подпись) (инициалы, фамилия) |  | (подпись) (инициалы, фамилия) |
| «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. |  | «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. |

Пояснительная записка

к выпускной квалификационной работе бакалавра

Разработка модуля генерации «2D прогулок»

(наименование работы)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Автор |  |  | Волосникова Инга Александровна | |
|  | (подпись и дата подписания) |  | (фамилия, имя, отчество) | |
| Обозначение | ВРБ-40 461 806–10.27–03–20.81 | | |  |
|  | (обозначение документа) | | |  |
| Группа | ИВТ-465 |  | | |
|  | (шифр группы) |  | | |
| Направление | 09.03.01 «Информатика и вычислительная техника» | | | |
|  |  | | | |
|  | (код и наименование направления, наименование программы (профиля)) | | | |
| Руководитель работы |  |  | Шабалина Ольга Аркадьевна | |
|  | (подпись и дата подписания) |  | (фамилия, имя, отчество) | |

Консультанты по разделам:

|  |  |  |
| --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |

Нормоконтролер

(подпись, дата подписания) (инициалы и фамилия)

Волгоград 2020 г.

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Кафедра «Системы автоматизированного проектирования  
и поискового конструирования»

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой САПР и ПК |
|  |  |
|  | М.В. Щербаков |
|  | (подпись) (инициалы, фамилия) |
|  | «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Задание на | | | выпускную квалификационную работу бакалавра | | | | | | | | |
|  | | | (наименование работы) | | | | | | | | |
| Студент | | | Волосникова Инга Александровна | | | | | | | | |
|  | | | (фамилия, имя, отчество) | | | | | | | | |
| Код кафедры | | | 10.27 | | |  | | Группа | ИВТ-465 |  | |
|  | | | | | |  | |  |  |  | |
| Тема | | Разработка модуля генерации «2D прогулок» | | | | | | | | | |
|  | |  | | | | | | | | | |
| Утверждена приказом по ВолгГТУ | | | | | | | | | | | |
| от | 07.10.2019 г. | | | № | 1238-ст | |  | | | | |
|  | (дата подписания) | | |  | | |  | | | | |
| Срок предъявления готовой работы | | | | | | | | 01 июня 2020 г. | |  | |
|  | | | | | | | | (дата подписания) | |  | |
| Исходные данные для выполнения работ | | | | | | | | | | | |
| Задание на выпускную работу бакалавра, выданное научным руководителем | | | | | | | | | | |
| от кафедры САПР и ПК, утверждённое приказом ректора университета | | | | | | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |

Содержание основной части пояснительной записки

|  |  |
| --- | --- |
|  | Анализ предметной области и существующих решений |
|  | Проектирование программы |
|  | Реализация программы |
|  | Тестирование программы |
|  | Заключение |
|  | Список использованных источников |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Перечень графического материала

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель работы |  |  | О.А.Шабалина |
|  | (подпись и дата подписания) |  | (инициалы и фамилия) |

Консультанты по разделам:

|  |  |  |
| --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |

Аннотация

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

Содержание

Введение 8

1 Анализ способов генерации 2D маршрута на карте закрытого пространства 9

1.1 Задача построения «2D прогулки» 9

1.2 Способы представления закрытого пространства 10

1.2.1 Сетка 10

1.2.2 Полигональная карта 10

1.2.3 Навигационная сетка 11

1.3 Способ построения маршрута в закрытом пространстве 12

1.3.1 Способы формализации закрытого пространства 12

1.3.2 Описание задачи построения маршрута 13

1.4 Анализ алгоритмов поиска пути 13

1.4.1 Алгоритмы поиска кратчайшего пути между двумя вершинами 13

1.4.2 Алгоритмы решения задачи коммивояжера 15

1.5 Анализ программных решений по навигации в закрытом пространстве 17

1.5.1 Приложения indoor-навигации 17

1.5.2 Мобильный гид «Кунсткамера» 18

1.5.3 Мобильный гид по Музею А.С.Пушкина 20

1.5.4 Мобильное приложение «Shopping Guide» 21

1.5.5 Мобильное приложение «NaviMail» 23

1.5.6 Чат-бот «Картография» 24

1.6 Сравнение существующих программных решений 25

1.7 Выводы 28

2 Разработка алгоритмов генерации 2D маршрута 29

2.1 Формализация задачи генерации 2D маршрута 29

2.1.1 Формализация описания пространства 29

2.1.2 Формализация построения маршрута 30

2.1.3 Формализация визуализации маршрута 31

2.2 Разработка алгоритмов построения графа 32

2.3 Выбор алгоритма построения маршрута 35

2.4 Разработка алгоритма визуализации маршрута 36

2.5 Разработка алгоритма моделирования прохождения маршрута 38

2.6 Выводы 40

3 Проектирование модуля генерации «2D прогулок» 40

3.1 Проектирование функциональности модуля 40

3.2 Описание сценариев использования 43

3.3 Проектирование графического интерфейса 46

3.4 Выводы 51

4 Реализация модуля генерации «2D прогулок» 51

4.1 Выбор средств реализации 51

4.2 Описание структуры входных и выходных данных 52

4.3 Описание результатов реализации модуля 52

4.4 Тестирование модуля 55

4.5 Выводы 60

Заключение 60

Список использованных источников 61

Приложение А Техническое задание 63

Приложение Б Содержание файла выходных данных 64

# 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, визуализации и моделирования прохождения маршрута на карте закрытого пространства, т.е. модуля генерации «2D прогулок», который можно использовать при создании конфигурируемых приложений для PID.

1. Анализ способов генерации 2D маршрута на карте закрытого пространства
   1. Задача построения «2D прогулки»

Построение «2D прогулки» («2D walkthrough») – это построение 2D маршрута на карте закрытого пространства; является способом навигации. Процесс генерации 2D маршрута включает в себя следующие этапы:

* отображение карты закрытого пространства;
* построение маршрута через две и более заданные на карте точки;
* изображение маршрута на карте пространства;
* моделирование прохождения маршрута.

Каждый из этапов генерации 2D маршрута может быть реализован различными способами.

* 1. Способы представления закрытого 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).

|  |  |
| --- | --- |
|  | https://civgames.files.wordpress.com/2010/05/civ5_s017.jpg |
| а) схематическое изображение | б) реализация карты |

Рисунок 1 – Представление пространства в виде сетки

* + 1. Полигональная карта

Для создания полигональной карты препятствия представляются в виде полигонов. Вершины полигонов соединяются между собой отрезками в том случае, если между ними есть прямой путь, не содержащий препятствий (т.е. вершины являются видимыми друг для друга). При таком представлении количество отрезков может быть до n2 для n вершин (рис.2).



Рисунок 2 – Представление пространства в виде полигональной карты

* + 1. Навигационная сетка

Для создания навигационной сетки доступное для передвижения пространство представляется в виде непересекающихся полигонов. Достаточным набором точек для описания пространства являются центры полигонов, однако могут использоваться точки на гранях или в вершинах полигонов, а также комбинация нескольких вариантов (рис.3).

|  |  |
| --- | --- |
|  | Unity - Manual: Navigation and Pathfinding |
| а) схематическое изображение | б) реализация карты |

Рисунок 3 – Представление пространства в виде навигационной сетки

* 1. Способ построения маршрута в закрытом пространстве
     1. Способы формализации закрытого пространства

Для построения маршрута пространство должно быть формализовано в виде графа. Если необходимо построение оптимального маршрута, граф должен быть взвешенным. Весом ребра может являться длина пути между двумя инцидентными вершинами (если осуществляется поиск маршрута, оптимального по длине) или время перехода между ними (если осуществляется поиск маршрута, оптимального по времени прохождения).

Если пространство представлено в виде сетки, каждому доступному для перемещения элементу сетки ставится в соответствие вершина графа. Затем каждая вершина соединяется ребром с каждой соседней ей вершиной, если она существует (у каждой вершины может быть до 4 соседних, если допустимо передвижение только по горизонтали и вертикали, и до 8 соседних, если допустимо также передвижение по диагонали).

Представление пространства в виде полигональной карты является формализованным: вершины полигонов являются вершинами графа, а отрезки, соединяющие вершины – ребрами графа.

Для формализации пространства, представленного в виде навигационной сетки, в качестве вершин графа принимается набор точек, описывающих пространство. Вершины соединяются ребрами в том случае, если между точками существует прямой путь, не выходящий за границы полигонов (т.е. проходящий по доступному для передвижения пространству).

* + 1. Описание задачи построения маршрута

Маршрут на карте пространства соответствует пути на графе, в виде которого формализовано пространство. Для построения маршрута задается список точек, через которые должен проходить маршрут: начальная и конечная точки, а также множество промежуточных точек. Порядок обхода промежуточных точек может быть жестко заданным или произвольным. Множество промежуточных точек может быть пустым. Если заданные точки не соответствуют вершинам графа, они добавляются к множеству вершин и соединяются ребром с ближайшей вершиной по правилам, аналогичным тем, что были применены при формализации пространства в виде графа.

Чаще всего производится поиск оптимального маршрута через заданные точки. Для нахождения оптимального пути между двумя последовательными точками маршрута используются алгоритмы поиска кратчайшего пути между двумя вершинами. Если порядок обхода промежуточных точек не задан, нахождение маршрута является незамкнутым вариантом задачи коммивояжера. Для построения маршрута в этом случае находятся кратчайшие пути между каждой парой заданных точек, а затем используются алгоритмы решения задачи коммивояжера.

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

Алгоритм Дейкстры [2] позволяет найти кратчайший путь от одной вершины графа до всех остальных, применяется только для графов без отрицательных ребер. 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 [2] 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 [2] 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 [2], 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\* [3] позволяет найти кратчайший путь от стартовой вершины к целевой, используя алгоритм поиска по первому наилучшему совпадению на графе. 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 [4]. 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 [5].

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 [6]. 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 [7]. Оба метода основаны на решении последовательности подзадач линейного программирования, однако метод отсечений требует меньшего объема памяти для хранения 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 [8].

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 [9].

Важным недостатком алгоритма ближайшего соседа является то, что в качестве оптимального часто принимается локально оптимальное решение. Этой проблемы позволяет избежать алгоритм поиска с запретами [10]. 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. Главным недостатком этого метода служит его время выполнения: трудоемкость алгоритма оценивается как О(n3), где n – количество вершин в графе [11].

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% [11].

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.

* 1. Анализ программных решений по навигации в закрытом пространстве
     1. Приложения indoor-навигации

Технология навигации и обнаружения объектов (устройств, людей) внутри зданий носит название indoor-навигации [12]. 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 приложения, использующие данную технологию, не рассматриваются в рамках данной работы.

* + 1. Мобильный гид «Кунсткамера»

Мобильный гид «Кунсткамера» [14] разработан российской компанией ООО «КАМИС». Гид представляет собой свободно распространяемое мобильное приложение для платформ Android и iOS.

В приложении реализованы следующие функции:

* отображение карты музея;
* отображение описания и изображения экспоната при нажатии на соответствующий ему объект на карте;
* воспроизведение аудио экскурсии по выбранной экспозиции.

Маршрут между объектами составляется неявным образом (без отображения на карте) путем озвучивания номеров экспонатов во время аудио экскурсии. Возможность проложить маршрут между двумя и более произвольно заданными точками отсутствует. Для использования приложения необходим навык чтения. Экранные формы приложения представлены на рис.4–5.



Рисунок 4 – Карта музея

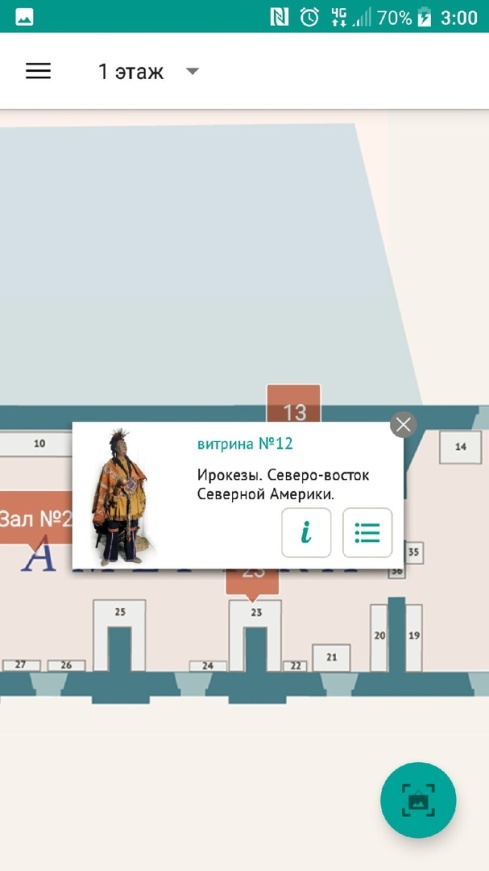


Рисунок 5 – Просмотр информации по экспонату

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

Мобильный гид [15] разработан российской компанией Indoors Navigation. Гид представляет собой свободно распространяемое мобильное приложение для платформ Android и iOS.

В приложении реализованы следующие функции:

* отображение карты музея;
* воспроизведение аудио экскурсии по выбранной экспозиции;
* построение и визуализация маршрута между двумя точками, заданными пользователем;
* навигация по маршруту с использованием описания участков маршрута или элементов дополненной реальности (AR).

Возможность проложить составной маршрут (между тремя и более произвольно заданными точками) отсутствует. Для использования приложения необходим навык чтения. Экранные формы приложения представлены на рис.6–7.

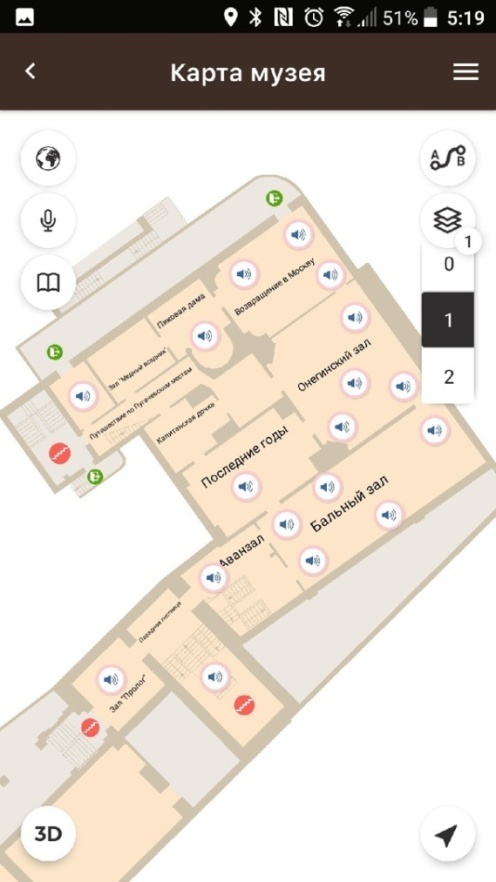


Рисунок 6 – Карта музея

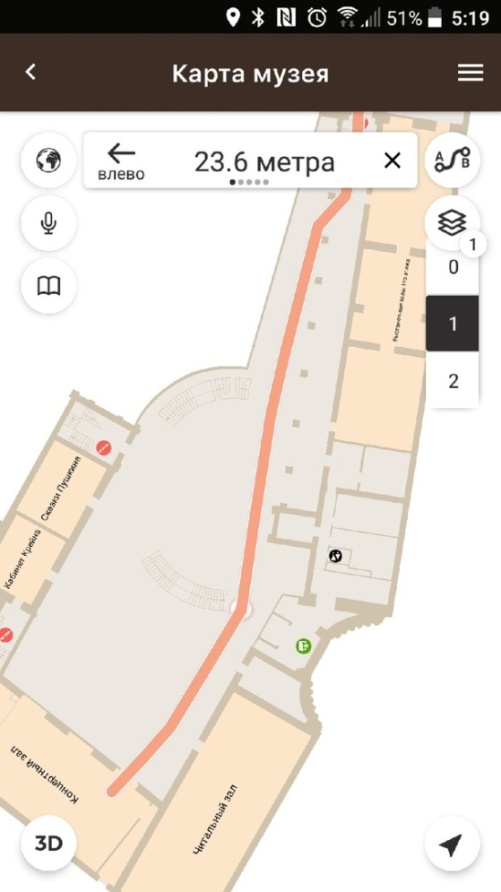


Рисунок 7 – Прокладывание маршрута

* + 1. Мобильное приложение «Shopping Guide»

Мобильное приложение «Shopping Guide» [16] разработано российской компанией NaviGuide и свободно распространяется для платформ Android и iOS.

В приложении реализованы следующие функции:

* выбор торгового центра;
* отображение карты торгового центра;
* поиск объекта на карте;
* построение и визуализация маршрута между двумя точками, заданными пользователем.

В приложении доступны карты 45 торговых центров. Возможность проложить составной маршрут (между тремя и более произвольно заданными точками) отсутствует. Для использования приложения необходим навык чтения. Экранные формы приложения представлены на рис.8–9.

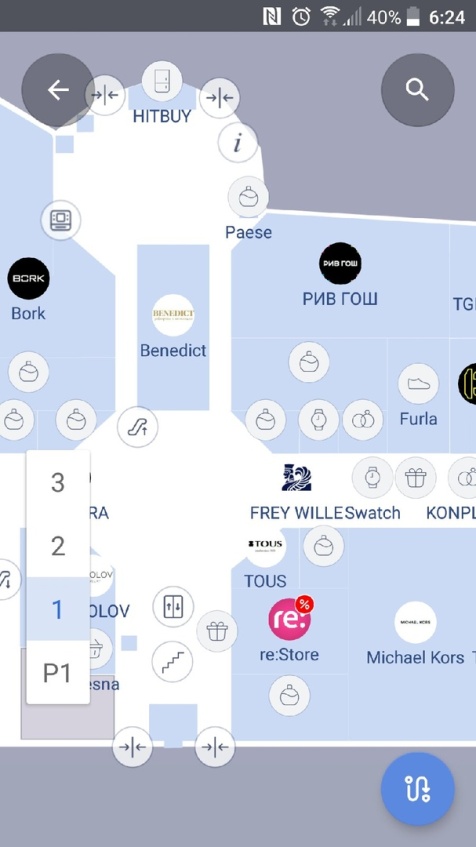


Рисунок 8 – Карта торгового центра

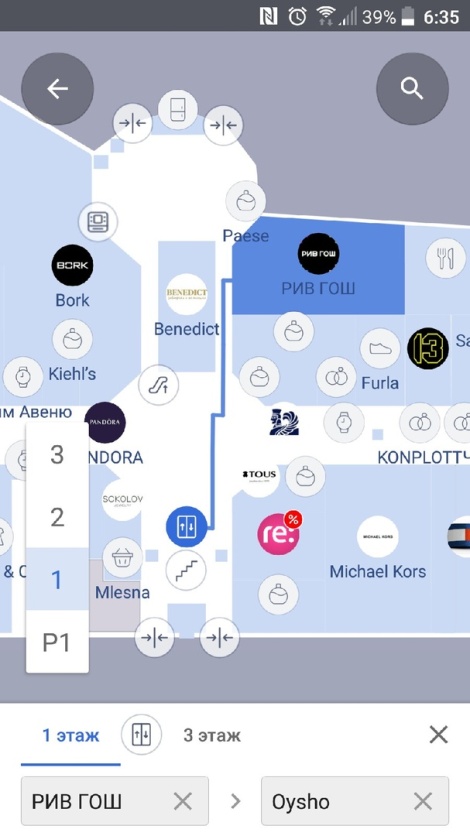


Рисунок 9 – Прокладывание маршрута

* + 1. Мобильное приложение «NaviMail»

Мобильное приложение «NaviMail» [17] разработано российской компанией SmartLabs и свободно распространяется для платформ Android и iOS.

В приложении реализованы следующие функции:

* выбор торгового центра;
* отображение карты торгового центра;
* поиск объекта на карте.

В приложении доступны карты 2 торговых центров. Возможность проложить маршрут между двумя и более произвольно заданными точками отсутствует. Для использования приложения необходим навык чтения. Экранные формы приложения представлены на рис.10–11.



Рисунок 10 – Карта торгового центра

|  |  |
| --- | --- |
| https://sun9-71.userapi.com/c857028/v857028173/183420/Ll3PJcLx_as.jpg | https://sun9-70.userapi.com/c206728/v206728173/10df3b/gHHk0-9KMVk.jpg |
| а) поиск по названию | б) поиск по товару |

Рисунок 11 – Поиск объекта

* + 1. Чат-бот «Картография»

Чат-бот «Картография» [18] разработан российской компанией MOLOKO для социальных сетей ВКонтакте и Facebook и мессенджера Telegram.

Чат-бот реализует следующие функции:

* выбор торгового центра;
* отображение карты торгового центра;
* поиск объекта на карте;
* построение и визуализация маршрута между двумя точками, заданными пользователем;
* текстовое описание маршрута.

В приложении доступны карты 5 торговых центров. Возможность проложить составной маршрут (между тремя и более произвольно заданными точками) отсутствует. Для использования приложения необходимы навыки чтения и письма. Пример работы чат-бота представлен на рис.12.

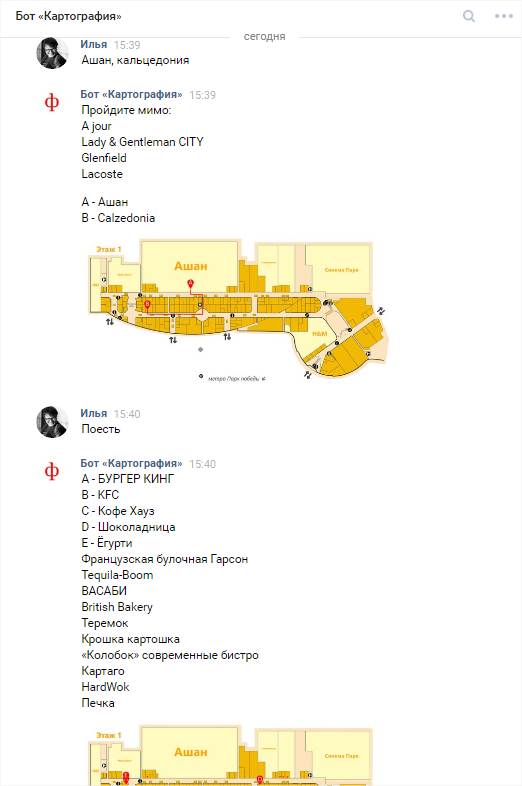


Рисунок 12 – Поиск объекта и построение маршрута

* 1. Сравнение существующих программных решений

Критерии оценки существующих программных решений должны учитывать не только их функциональность, но возможность их применения PID.

Данная категория пользователей обладает следующими особенностями. Во-первых, многие 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. Наконец, выбор доступных карт должен быть как можно более широким.

Таким образом, оценка существующих решений должна быть произведена по следующим критериям:

* количество доступных карт пространства;
* наличие функции поиска объектов;
* наличие функции составления маршрута;
* режим работы (онлайн или офлайн);
* навыки, необходимые для использования.

Результаты сравнения представлены в Таблице 1.

Таблица 1 – Анализ программных решений по навигации в закрытом пространстве

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название решения | Критерии анализа решений | | | | | | | |
| Кол-во доступ-ных карт | Функция поиска объектов | | Функция составле-ния маршрута | | Режим работы | | Необходи-мые навыки |
| Мобильный гид «Кунсткамера» | 1 | Нет | | Нет | | Оффлайн | | Чтение |
| Мобильный гид по Музею А.С.Пушкина | 1 | Есть | Есть | | Онлайн и оффлайн | | Чтение | |
| Мобильное приложение «Shopping Guide» | 45 | Есть | Есть | | Оффлайн | | Чтение | |
| Мобильное приложение «NaviMail» | 2 | Есть | Нет | | Онлайн и оффлайн | | Чтение | |
| Чат-бот «Картография» | 5 | Есть | Есть | | Онлайн | | Чтение и письмо | |

Часть рассмотренных решений жестко привязаны к пространству, внутри которого осуществляется навигация. Остальные предоставляют возможность выбора из нескольких пространств, однако исходный код всех решений является закрытым, и добавление новых объектов возможно только разработчиками.

Некоторые из рассмотренных решений не имеют функции построения маршрута, остальные выполняют только построение простого маршрута, т.е. соединяющего две точки. Функция генерации составного маршрута (с несколькими промежуточными точками) во всех рассмотренных решениях отсутствует.

* 1. Выводы

Задача генерации «2D прогулки» была декомпозирована на несколько подзадач: отображение карты закрытого пространства, построение и визуализация маршрута, моделирование его прохождения. Для каждой подзадачи были рассмотрены существующие способы ее решения.

Анализ существующих программных решений по навигации в закрытом пространстве показал, что они не ориентированы на PID, поскольку для их использования нужны навыки чтения и письма, список пространств, для которых доступны функции навигации, ограничен, а функция прокладывания составного маршрута недоступна.

Таким образом, актуальной является 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.

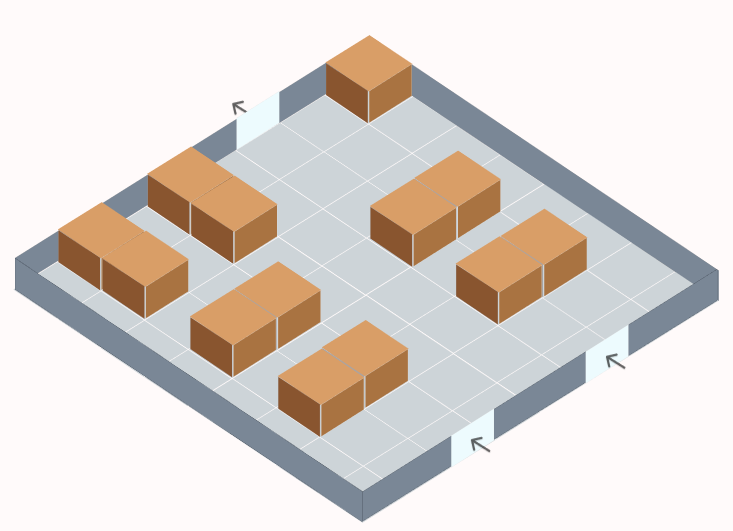
Целью данной работы является разработка модуля генерации «2D прогулок».

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 прогулок»;
* протестировать полученный модуль и, при необходимости, исправить выявленные при тестировании недостатки.

1. Разработка алгоритмов генерации 2D маршрута
   1. Формализация задачи генерации 2D маршрута
      1. Формализация описания пространства

Разработан следующий способ формализации задачи генерации 2D маршрута на карте закрытого пространства. 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 (рис.13).



Выход

Объекты

Вход

Граница пространства

Рисунок 13 – Модель пространства

* + 1. Формализация построения маршрута

Для построения маршрута пространство формализуется в виде графа (рис.14).

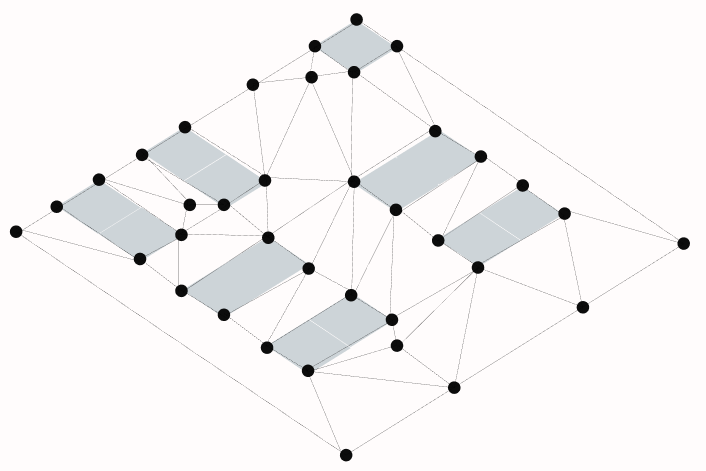
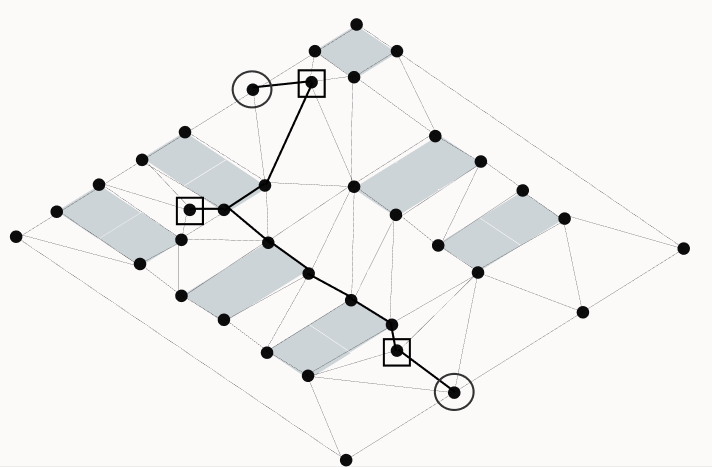


Рисунок 14 – Представление пространства в виде графа

Маршрут задается набором точек на графе, состоящем из начальной, конечной и множества промежуточных точек. В частном случае множество промежуточных точек может быть пустым, в этом случае маршрут является простым. Маршрут представляет собой путь на графе, проходящий через все заданные точки ровно один раз (рис.15).



Маршрут

Промежуточные точки

Конечная точка

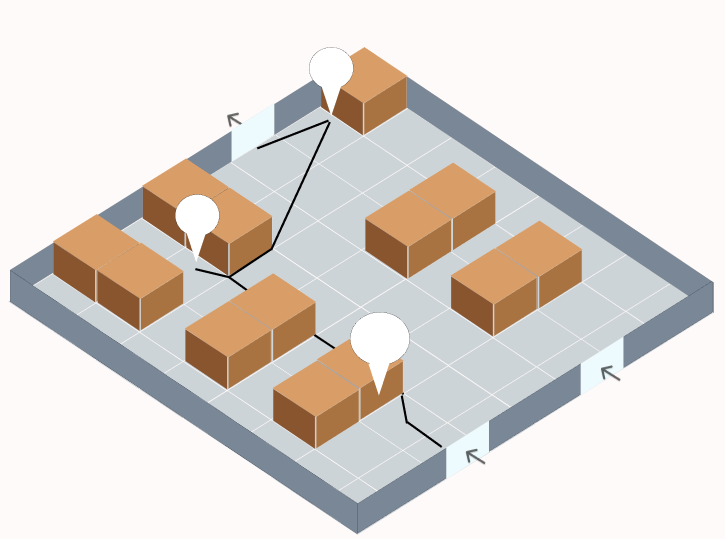
Начальная точка

Рисунок 15 – Маршрут обхода точек

В такой постановке задачи может быть несколько путей. Чаще всего ставится задача поиска оптимального пути. Для его построения используются рассмотренные алгоритмы поиска оптимального пути и алгоритмы решения задачи коммивояжера.

* + 1. Формализация визуализации маршрута

Построенный маршрут визуализируется на карте пространства в виде ломаной линии, отрезкам которой соответствуют ребра графа, составляющие найденный путь (рис.16).



Размеченный маршрут

Заданные для построения маршрута точки

Рисунок 16 – Модель карты пространства

* 1. Разработка алгоритмов построения графа

Постановка задачи генерации «2D прогулки» предполагает, что пространство, в котором осуществляется навигация, имеет большую площадь и сложную структуру, поэтому представление пространства в виде сетки потребует большого объема памяти для хранения данных. В случае представления пространства в виде полигональной карты, хранение большого числа ребер также потребует значительных затрат ресурсов. Навигационная сетка является наиболее подходящим способом представления пространства: количество опорных точек может быть сокращено для уменьшения объема памяти, требуемой для хранения информации о пространстве.

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. Алгоритм формализации представлен на рис.17.

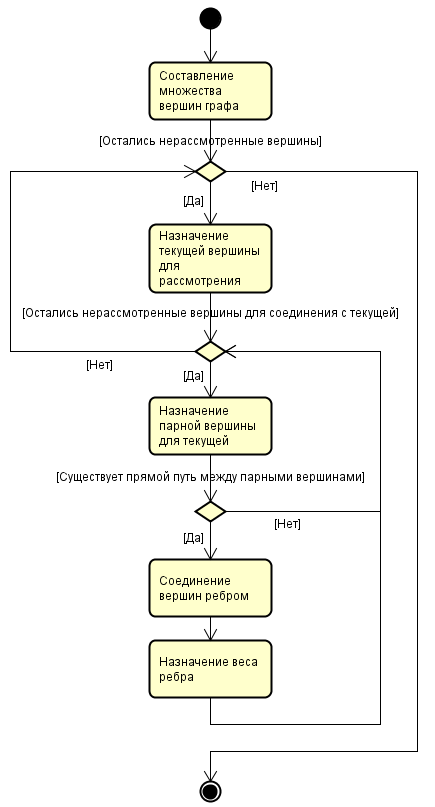


Рисунок 17 – Алгоритм построения графа

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. Ребра, составляющие найденный путь, стягиваются в одно ребро с весом, равным длине пути. Исходя из анализа алгоритмов нахождения оптимального пути между двумя точками, для использования принят алгоритм А\* как наиболее эффективный. 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 стягивания ребер графа представлен на рис.18.

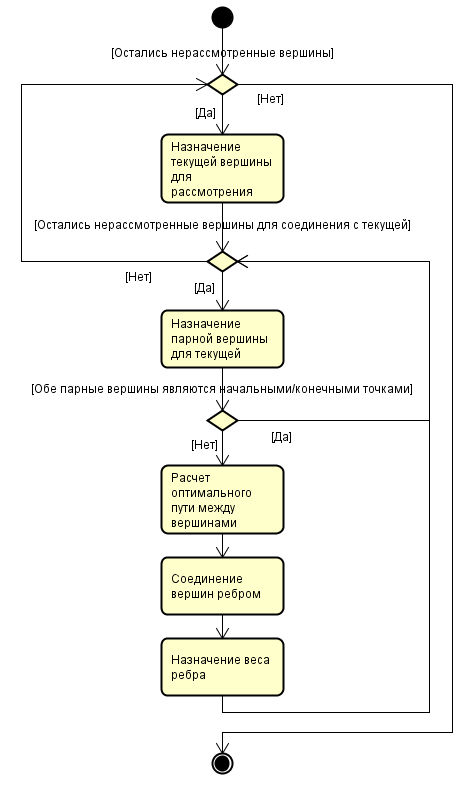


Рисунок 18 – Алгоритм стягивания ребер графа

* 1. Выбор алгоритма построения маршрута

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.

Исходя из сравнения алгоритма ближайшего соседа, генетического и жадного алгоритма [13], 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 2.

Таблица 2 – 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 |

Поскольку при разработке модуля необходимо учитывать не только оптимальность найденного маршрута, но и быстродействие, обеспечивающее удобство использования, на данном этапе не применимы точные методы нахождения оптимального пути, т.к. они требуют значительных временных затрат даже при небольшом количестве вершин. Согласно проведенному анализу, 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 на цели передвижения.

Поскольку ребрам между каждой парой точек в модифицированном графе, получение которого описано в п.2.2, 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. Алгоритм визуализации маршрута представлен на рис.16.

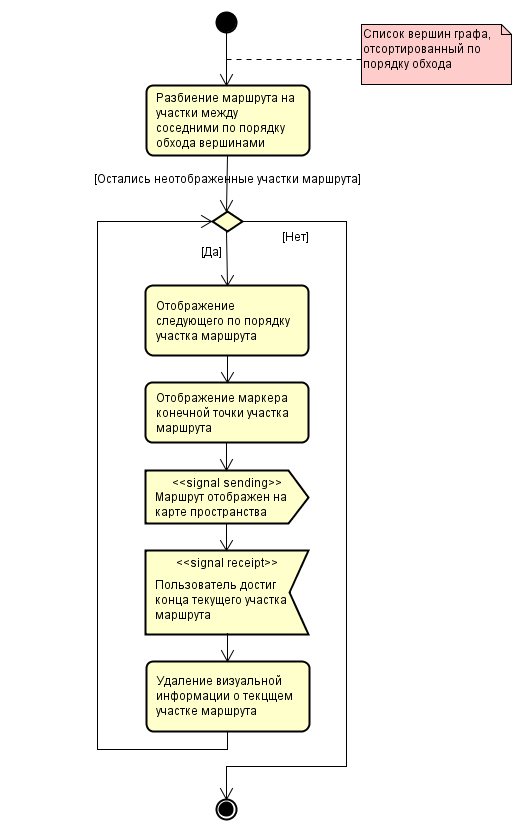


Рисунок 16 – Алгоритм визуализации маршрута

* 1. Разработка алгоритма моделирования прохождения маршрута

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

Для оценки успешности прохождения маршрута составляется отношение длины пройденного персонажем пути к длине оптимального маршрута. Результат визуализируется в виде смайликов. В зависимости от значения полученного отношения, возможны следующие результаты:

* меньше или равно 1.5: смайлик «большой палец вверх»;
* больше 1.5: смайлик «большой палец вниз».

После достижения конечной точки последнего участка маршрута пользователю выводится сообщение с общей оценкой прохождения маршрута в виде списка смайликов-оценок за каждый пройденный участок. Алгоритм моделирования прохождения маршрута представлен на рис.17.

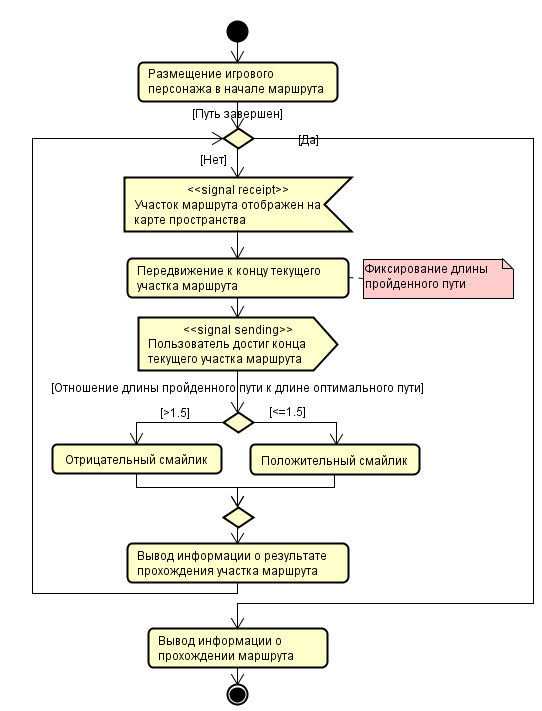


Рисунок 17 – Алгоритм моделирования прохождения маршрута

* 1. Выводы

Таким образом, разработан алгоритм генерации «2D прогулки», состоящий из следующих этапов:

* формализация пространства в виде графа;
* нахождение оптимальных путей между парами заданных вершин;
* решение задачи коммивояжера;
* визуализация найденного пути;
* моделирование прохождения найденного пути.

Для решения задачи формализации пространства выбран способ представления пространства в виде навигационной сетки. Нахождение оптимальных путей на полученном графе осуществляется с помощью алгоритма А\*.

Решение задачи коммивояжера осуществляется с помощью жадного алгоритма, что позволяет найти близкий к оптимальному маршрут за время, допустимое для сохранения быстродействия и удобства использования модуля.

Визуализация построенного маршрута осуществляется поэтапно в процессе моделирования прохождения маршрута. Алгоритм моделирования прохождения маршрута реализован с применением игровых технологий.

1. Проектирование модуля генерации «2D прогулок»
   1. Проектирование функциональности модуля

В соответствии с анализом задачи генерации «2D прогулки» и требований к ее решению, определен список функций, выполнение которых должен обеспечить разрабатываемый модуль:

* загрузка и отображение выбранной карты пространства;
* загрузка списка объектов для построения маршрута;
* построение размеченного маршрута;
* отображение игрового персонажа на карте пространства;
* отображение текущего участка маршрута на карте пространства;
* моделирование передвижения игрового персонажа по маршруту;
* отображение результатов моделирования прохождения маршрута;
* сохранение результатов «2D прогулки».

Требования к модулю приведены в Техническом задании (Приложение А).

Модель пространства должна иметь визуальное сходство с помещением, для которого генерируется «2D прогулка». Масштаб выбирается таким образом, чтобы пользователь мог ассоциировать виртуальную «прогулку» с настоящей. Для изображения пространства может быть использована 3D модель помещения в изометрической проекции. Таким образом, будет получена псевдотрехмерность, или 2,5D. Это позволит создать более реалистичную модель помещения, при этом перемещение будет возможно только в горизонтальной плоскости, т.е. в плоскости карты пространства, что не нарушает принцип «2D прогулки».

Если объект на карте пространства имеет название, через него может быть проложен маршрут. Объект, не имеющий названия, является препятствием и не может быть включен в маршрут. В качестве идентификатора объекта для включения его в список объектов для построения маршрута используется название объекта.

Построение маршрута должно осуществляться до отображения карты пространства пользователю. Карта должна отображаться вместе с первым участком построенного маршрута и игровым персонажем. Игровой персонаж представляет собой анимированную модель человека. Элементы управления должны позволять перемещать персонажа произвольно в плоскости карты пространства. Персонаж не должен иметь возможность перемещаться сквозь объекты-препятствия. Путь, пройденный персонажем на каждом участке маршрута, фиксируется для определения успешности прохождения и суммируется для сохранения результатов моделирования прохождения маршрута.

Разрабатываемый модуль должен быть интегрируемым для подключения его к различным приложениям для помощи PID. Список объектов для построения маршрута должен передаваться в модуль при его вызове. Результат генерации «2D прогулки» должен сохраняться в файл для возможности его анализа и использования другими модулями.

Модуль должен хранить модели пространств, доступные для генерации «2D прогулки». На этапе разработки, которому посвящена данная работа, добавление новых моделей должно осуществляться вручную разработчиком. Дальнейшее развитие модуля предполагает создание инструмента, с помощью которого функция добавления новых моделей будет доступна для пользователей (опекунов или членов семьи PID, использующих приложение).

Описанные функции модуля представлены в виде диаграммы прецедентов на рис.18.

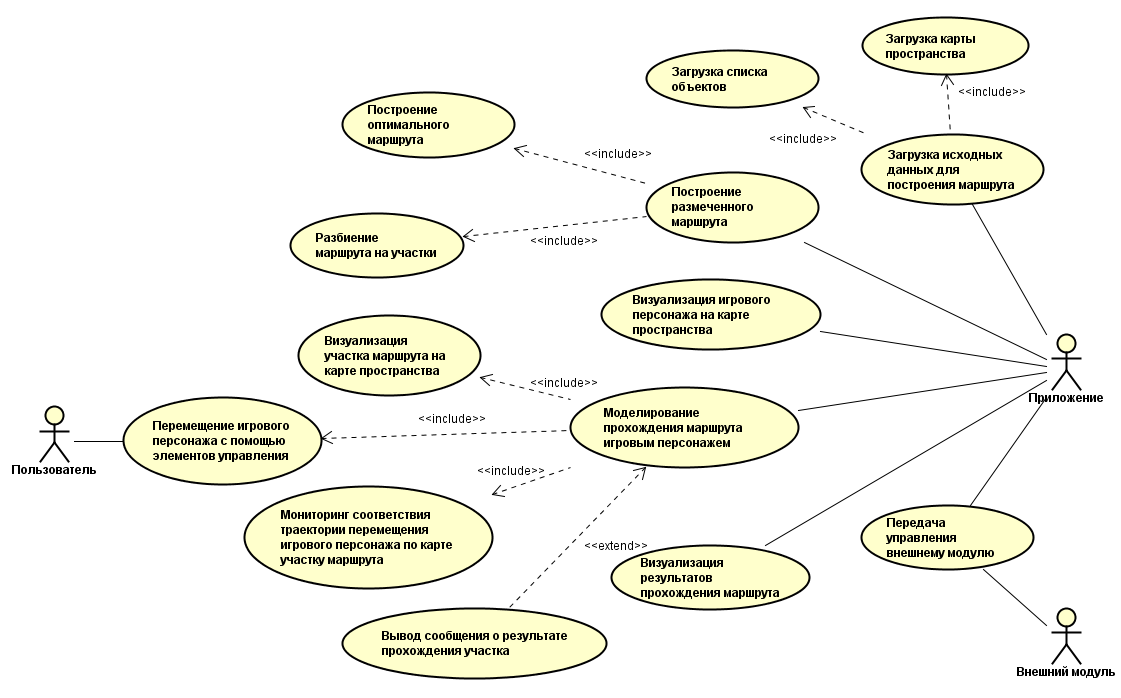


Рисунок 18 – Диаграмма прецедентов

* 1. Описание сценариев использования

Для тестирования работы модуля при добавлении или изменении моделей пространств, а также для обеспечения возможности его автономного использования, принято решение реализовать дополнительную возможность составления списка объектов и выбора модели пространства в рамках работы модуля. Данная функция предназначена для опекунов PID, использующих приложение, и для ее использования необходим навык чтения.

Таким образом, при работе с модулем возможны следующие сценарии использования:

* использование модуля внутри мобильного приложения (PID используют модуль самостоятельно)
* автономное использование модуля (PID используют модуль совместно с опекуном).

Описание сценария использования модуля в рамках приложения приведено в Таблице 3.

Описание сценария настройки модуля при автономном использовании приведено в Таблице 4.

Таблица 3 – Описание сценария «Использование модуля в рамках приложения»

|  |  |  |  |
| --- | --- | --- | --- |
| Название сценария | | Использование модуля в рамках приложения | |
| Пользователь | | PID | |
| Предусловие | | Список объектов составлен и передан в модуль программно при работе приложения | |
| Описание сценария | | | |
| Шаг | Действие пользователя | | Реакция системы |
| 1 | Нажатие на кнопку перехода к модулю | | Отображение карты пространства, игрового персонажа и первого участка маршрута |
| 2 | Нажатие на кнопку управления персонажем | | Воспроизведение анимации передвижения персонажа |
| 3 | Перемещение персонажа на конечную точку текущего участка | | Блокировка элементов управления персонажем, затемнение карты пространства и отображение поверх нее окна с рейтингом, длиной пройденного пути, длиной оптимального пути и кнопкой закрытия окна |
| 4 | Нажатие на кнопку закрытия окна | | Закрытие окна, разблокировка элементов управления персонажем |
| 5 | Перемещение персонажа на конечную точку маршрута | | Переход на экран с итоговым рейтингом, временем прохождения, длиной пройденного пути, кнопками выхода и повторного прохождения |
| 6-1 | Нажатие на кнопку повторного прохождения | | Аналогично шагу 1 |
| 6-2 | Нажатие на кнопку выхода | | Передача управления следующему модулю приложения |

Таблица 4 – Описание сценария «Настройка модуля при автономном использовании»

|  |  |  |  |
| --- | --- | --- | --- |
| Название сценария | | Настройка модуля при автономном использовании | |
| Пользователь | | Опекун PID | |
| Предусловия | | Модуль запущен автономно, открыт экран основного меню | |
| Описание сценария | | | |
| Шаг | Действие пользователя | | Реакция системы |
| 1 | Нажатие на кнопку выбора тестовых данных | | Переход к меню выбора модели пространства и составления списка объектов |
| 2 | Нажатие на название пространства в списке | | Выделение названия выбранного пространства и снятие предыдущего выделения |
| 3 | Нажатие на название объекта в списке | | Выделение названия выбранного объекта |
| 4 | Нажатие на кнопку возврата в основное меню | | Переход к основному меню |
| 5 | Нажатие на кнопку запуска | | Передача названия выбранного пространства и списка объектов в функцию генерации маршрута |

После настройки приложения происходит переход на экран отображения карты пространства, дальнейшее использование модуля PID аналогично сценарию 1, за исключением шагов 6-1 и 6-2. На шаге 6-1 происходит возврат к экрану основного меню. На шаге 6-2 происходит закрытие модуля.

* 1. Проектирование графического интерфейса

Графический интерфейс состоит из следующих экранов:

* экран основного меню настройки;
* экран выбора тестовых данных;
* экран моделирования прохождения маршрута;
* экран результатов прохождения маршрута.

Экран основного меню настройки содержит:

* кнопку запуска модуля;
* кнопку перехода на экран выбора тестовых данных;
* кнопку выхода.

Макет экрана представлен на рис.19.

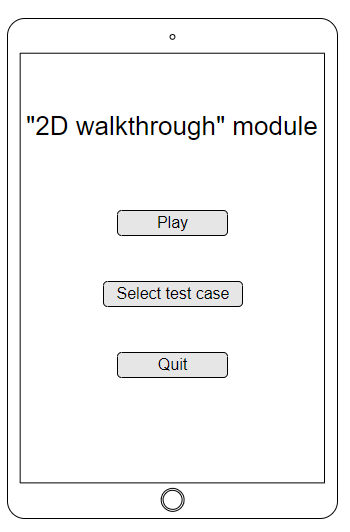


Рисунок 19 – Макет экрана основного меню

Экран выбора тестовых данных содержит:

* список моделей пространств;
* список объектов;
* кнопку возврата в основное меню.

Макет экрана представлен на рис.20.

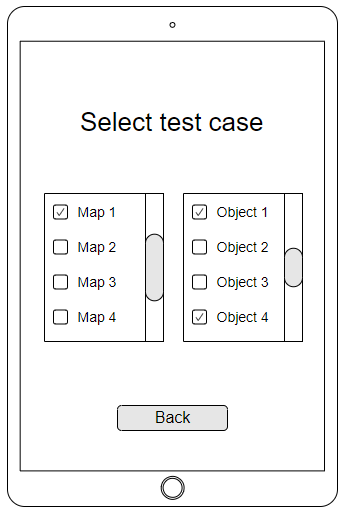


Рисунок 20 – Макет экрана выбора тестовых данных

Экран моделирования прохождения маршрута содержит:

* изображение карты пространства с игровым персонажем;
* кнопки управления персонажем;
* навигационную панель.

Макет экрана представлен на рис.21.

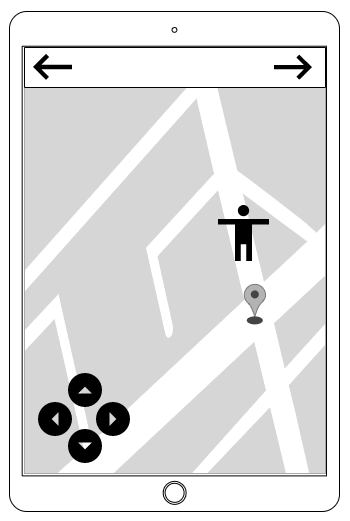


Рисунок 21 – Макет экрана моделирования прохождения маршрута

На экране моделирования прохождения маршрута поверх карты пространства выводится окно результатов прохождения участка маршрута, которое содержит:

* оценку результата прохождения в виде смайлика;
* кнопку закрытия окна.

Макет окна представлен на рис.22.

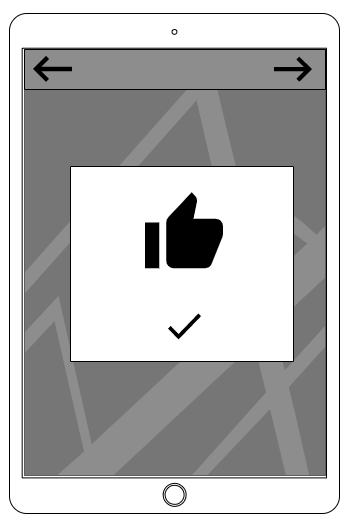


Рисунок 22 – Макет окна результатов прохождения

Экран результатов моделирования содержит:

* оценку результата прохождения в виде списка смайликов;
* кнопку повторного прохождения;
* кнопку выхода.

Макет экрана представлен на рис.23.

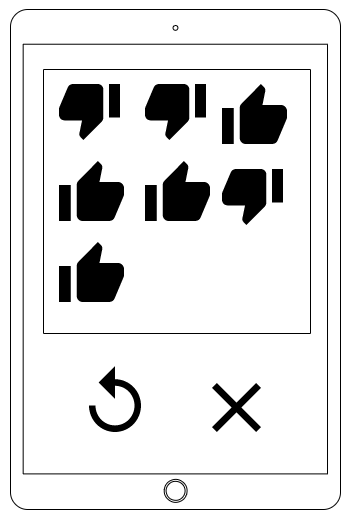


Рисунок 23 – Макет экрана результатов

Схема графического интерфейса представлена на рис.24.

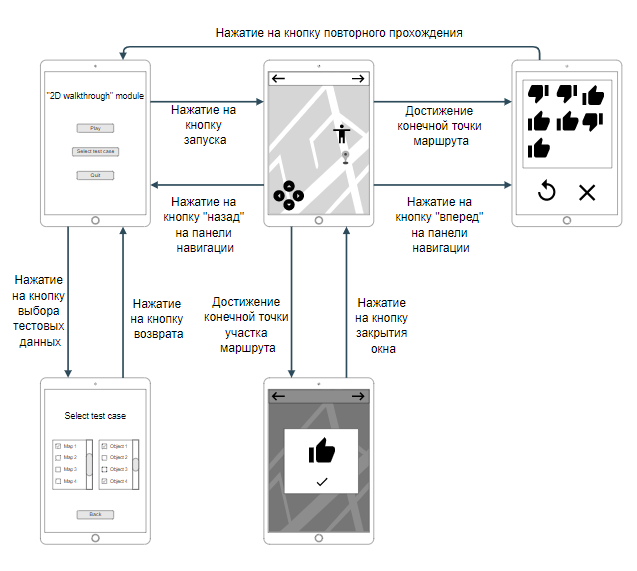


Рисунок 24 – Схема графического интерфейса

* 1. Выводы

На основании анализа задачи генерации «2D прогулки» определены требования к функциям модуля и особенности их реализации, спроектирован графический интерфейс модуля.

Принято решение о необходимости реализации двух режимов использования модуля – интеграция в приложение и автономная работа. Описаны сценарии работы в обоих режимах, первый из которых предназначен для самостоятельной работы PID, а второй требует участия опекуна.

1. Реализация модуля генерации «2D прогулок»
   1. Выбор средств реализации

Разработка модуля осуществляется для мобильных устройств на базе операционной системы Android версии 4.4 и выше. Существующее мобильное приложение для PID [20] разработано на платформе Unity, поэтому всю дальнейшую разработку, включая данный модуль, было решено вести с использованием среды разработки Unity Personal 2018. Данная платформа является одним из наиболее удобных и многофункциональных средств разработки игровых приложений. Исходные коды модуля написаны на языке высокого уровня C#. Для компиляции приложения используется Android SDK.

При создании карт пространства использованы модели объектов, распространяемые через Unity Asset Store.

* 1. Описание структуры входных и выходных данных

Для передачи входных данных в модуль, их хранения и обработки в процессе выполнения функций модуля, а также для хранения информации о прохождении маршрута, реализован статический класс WayDescription. Входные данные передаются в следующем формате:

* идентификатор пространства: string;
* список объектов для построения маршрута: HashSet<string>.

Выходными данными для модуля являются:

* дата и время начала прохождения маршрута: DateTime;
* дата и время окончания прохождения маршрута: DateTime;
* идентификатор пространства: string;
* описание участков маршрута: List<WalkPartModellingResult>:
* флаг завершения прохождения маршрута: bool.

Формат описания участка маршрута WalkPartModellingResult:

* конечный объект участка: string;
* время прохождения участка: TimeSpan;
* длина пройденного пути: float;
* длина оптимального пути: float;

Выходные данные сохраняются в файл формата JSON. Пример содержания файла выходных данных приведен в Приложении Б.

* 1. Описание результатов реализации модуля

Реализация модуля проводилась в три этапа.

На первом этапе были реализованы алгоритмы поиска маршрута и его визуализации на карте пространства, добавлен игровой персонаж и элементы управления его передвижением.

На втором этапе был создан графический интерфейс модуля: меню для автономного использования модуля с возможностью настройки набора тестовых данных, навигационная панель на основном экране, позволяющая переключаться между экранами модуля, окно с результатами прохождения участка маршрута и экран вывода результатов прохождения маршрута целиком.

На третьем этапе были добавлены модели объектов для имитации реального помещения (в качестве примеров реализованы модели магазинов).

Набор файлов модуля может быть интегрирован в любое мобильное приложение, разрабатываемое на платформе Unity 2018 и выше.

Экранные формы модуля представлены на рис.24-29.

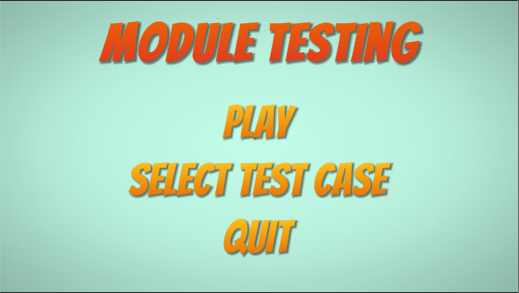


Рисунок 24 – Экран основного меню при автономной работе модуля



Рисунок 25 – Экран выбора тестовых данных



Рисунок 26 – Модель пространства



Рисунок 27 – Визуализация текущего участка маршрута

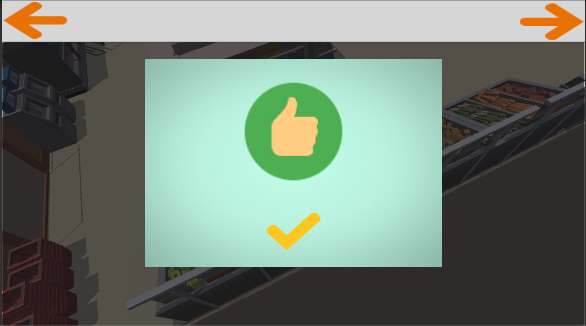


Рисунок 28 – Оценка прохождения участка маршрута



Рисунок 29 – Экран вывода информации о прохождении маршрута

* 1. Тестирование модуля

Для проведения тестирования была создана тестовая группа, состоящая из PID и их опекунов. Описание тестовой группы приведено в Таблице 5.

Таблица 5 – Описание тестовой группы

|  |  |  |
| --- | --- | --- |
| Пользователь | Возраст | Категория |
| Мария | 33 года | Опекун |
| Елена | 44 года | Опекун |
| Андрей | 12 лет | PID (задержка умственного развития) |
| Антон | 16 лет | PID (расстройство аутистического спектра) |

При проведении тестирования опекуны выполняли настройку тестовых данных, PID выполняли прохождения маршрута. Участие опекунов было также необходимо для поддержки и объяснения задач PID при взаимодействии с модулем.

Тестовый сценарий, выполняемый опекунами, приведен в Таблице 6. Тестовый сценарий, выполняемый PID, приведен в Таблице 7. Тестовые сценарии выполнялись последовательно, т.е. PID приступали к тестированию модуля после того, как их опекуны выполняли настройку тестовых данных и запускали моделирование прохождения маршрута. Таким образом, тестовый сценарий PID выполняется при открытом основном экране модуля.

Таблица 6 – Тестовый сценарий опекуна

|  |  |  |  |
| --- | --- | --- | --- |
| Шаг | Действие пользователя | Ожидание | Реакция системы |
| 1 | Запуск модуля | Должен открыться экран основного меню автономной работы модуля | Отображение экрана основного меню |
| 2 | Нажатие на кнопку выбора тестовых данных | Должен открыться экран меню выбора тестовых данных | Отображение экрана выбора тестовых данных с выбранными значениями по умолчанию |
| 3 | Выбор помещения | Выбранное помещение должно быть выделено, предыдущее выделение должно быть снято | Чекбокс с названием выбранного помещения отмечается, предыдущая отметка снимается |
| 4 | Выбор объекта | Выбранный объект должен быть выделен | Чекбокс с названием выбранного объекта отмечается |
| 5 | Нажатие на кнопку возврата в основное меню | Должен открыться экран основного меню автономной работы модуля | Отображение экрана основного меню |
| 6 | Нажатие на кнопку запуска | Должен открыться основной экран модуля | Генерация маршрута, отображение первого участка маршрута на карте пространства, перемещение игрового персонажа в начальную точку маршрута, отображение основного экрана модуля |

Таблица 7 – Тестовый сценарий PID

|  |  |  |  |
| --- | --- | --- | --- |
| Шаг | Действие пользователя | Ожидание | Реакция системы |
| 1 | Нажатие на кнопку управления персонажем | Персонаж должен переместиться в направлении, выбранном пользователем | Отображение анимации персонажа, перемещение персонажа в выбранном на правлении на расстояние, пропорциональное продолжительности нажатия на кнопку управления, подсчет длины пути, пройденного персонажем |
| 2 | Перемещение персонажа на конечную точку участка маршрута | Должно открыться окно с информацией о прохождении участка маршрута | Скрытие элементов управление персонажем, затемнение экрана, вывод длины пути, пройденного персонажем, длины оптимального пути, определение и вывод рейтинга, отображение окна информации |
| 3 | Нажатие на кнопку закрытия окна информации | Окно с информацией о прохождении участка маршрута должно закрыться, на карте должен быть отображен следующий участок маршрута | Отображение следующего участка маршрута и элементов управления персонажа, закрытие окна информации и отмена затемнения экрана |
| 4 | Перемещение персонажа на конечную точку маршрута | Должен открыться экран с информацией о прохождении маршрута | Отображение экрана результатов прохождения маршрута, вывод длины пути, пройденного персонажем, вывод времени, затраченного на прохождение маршрута, определение и вывод рейтинга, сохранение результатов в файл |
| 5-1 | Нажатие на кнопку повтора | Должен открыться экран основного меню автономной работы модуля | Отображение экрана основного меню |
| 5-2 | Нажатие на кнопку выхода | Модуль должен прекратить работу | Закрытие модуля |

Результаты тестирования:

1. опекуны:

* настройка модуля не вызвала сложностей;
* интерфейс был интуитивно понятен;

1. PID:

* управление персонажем и ориентирование в виртуальном пространстве были понятны;
* было отмечено сходство с процессом реального посещения магазина.

* 1. Выводы

В модуле были реализованы все спроектированные функции. Был создан базовый набор моделей пространств. Созданный модуль был протестирован с участием PID и их опекунов. Результаты тестирования показали, что все функции модуля успешно реализованы, интерфейс модуля понятен пользователям.

# Заключение

В рамках данной бакалаврской работы был разработан модуль генерации «2D прогулки». Был проведен анализ и декомпозиция задачи генерации 2D маршрута на карте закрытого пространства, проанализированы методы ее решения, а также существующие программные решения по навигации в закрытом пространстве. Были разработаны и реализованы алгоритмы генерации 2D маршрута. Созданный модуль был протестирован с участием PID и их опекунов. Тестирование показало, что все функции модуля успешно реализованы, интерфейс интуитивно понятен.

Дальнейшая работа над модулем будет производиться в следующих направлениях:

* расширение набора моделей пространств;
* создание инструмента для добавления моделей пространств пользователями;
* подключение модуля к веб-системе управления мобильными приложениями для PID.

# 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. Indoor navigation: knowledge [Электронный ресурс]. – Режим доступа: https://www.indoornavigation.com/knowledge (дата обращения: 09.12.2019).
13. 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.
14. Кунсткамера. Гид по музею [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=com.msl.kunstkamera (дата обращения: 10.12.2019).
15. Музей Пушкина. Гид по музею [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=pro.indoorsnavi.puskinmuseum (дата обращения: 10.12.2019).
16. 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).
17. 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).
18. Бот «Картография» [Электронный ресурс]. – Режим доступа: https://vk.com/kartografia\_bot (дата обращения: 13.12.2019).
19. Bycer J. Game Design Deep Dive: Platformers // CRC Press Taylor & Francis Group.- 2019
20. Воронина, А.А. Мобильное приложение для поддержки процесса совершения покупок для людей с ограниченными интеллектуальными возможностями / А.А. Воронина, О.А. Шабалина // Информационные технологии в науке, управлении, социальной сфере и медицине: матер. V Междунар.науч. конф. – Томск, 2018. – С. 179 – 184.

Приложение А

Техническое задание

Приложение Б

Содержание файла выходных данных

{   "SpaceLabel": "ShopB",

    "BeginningTime": "15.05.2020 18:20:45",

    "EndingTime": "15.05.2020 18:21:21",

    "Path": [

        {

            "EndPoint": "Cucumber",

            "Time": "0m 1s 995ms",

            "PathPartLength": 7.55806303024292,

            "OptimalPathPartLength": 7.55806303024292

        },

        {

            "EndPoint": "Carrot",

            "Time": "0m 4s 346ms",

            "PathPartLength": 12.617047309875489,

            "OptimalPathPartLength": 10.880000114440918

        },

        {

            "EndPoint": "Cereals",

            "Time": "0m 7s 228ms",

            "PathPartLength": 30.71002960205078,

            "OptimalPathPartLength": 16.866239547729493

        },

        {

            "EndPoint": "Cheese",

            "Time": "0m 5s 279ms",

            "PathPartLength": 17.904682159423829,

            "OptimalPathPartLength": 13.421821594238282

        },

        {

            "EndPoint": "Exit",

            "Time": "0m 8s 21ms",

            "PathPartLength": 39.811885833740237,

            "OptimalPathPartLength": 31.1292724609375

        }

    ],

    "IsPathCompleted": true }