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

\_\_\_\_\_\_\_\_\_ д.т.н. Щербаков М.В.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Модуль генерации «2D прогулок»

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ЛИСТОВ - 20

|  |  |
| --- | --- |
|  | Научный руководитель  к.т.н., доц. Шабалина О.А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2020 |
| Нормоконтролер  к.т.н., доц. Шабалина О.А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2020 | Исполнитель  студентка группы ИВТ-465  Волосникова И.А.     «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2020 |

Аннотация

Техническое задание на программу «Модуль генерации «2D прогулки». Составлено и оформлено согласно ГОСТ 19.201-78. Страниц – 20.

Содержит основания и назначение разработки, основные сведения об объекте разработки, требования к программе и программной документации, стадии и этапы разработки, порядок контроля и приемки.

Глоссарий

Объект – 2D модель предмета, имеющая собственные параметры и поведение.

2D пространство – 2D модель помещения с входом (входами) и выходом (выходами), которое заполнено объектами.

Маршрут – путь, проходящий через все заданные объекты только один раз, началом которого является вход (один из входов) в 2D пространство, а окончанием – выход (один из выходов) из него.

Оптимальный маршрут – кратчайший маршрут обхода объектов в 2D пространстве.

Список объектов – непустое подмножество объектов, находящихся в 2D пространстве, через которые должен быть проложен маршрут.

Участок маршрута – часть маршрута, заключенная между двумя объектами, находящимися один за другим по порядку обхода (или входом и первым по порядку обхода объектом, или последним по порядку обхода объектом и выходом). Весь маршрут состоит из N+1 участков, где N – число объектов, которые необходимо обойти.

Размеченный маршрут – оптимальный маршрут, разбитый на участки.

Карта 2D пространства – изображение 2D пространства с построенным размеченным маршрутом.

Текущий участок маршрута – участок маршрута, отображаемый в данный момент на карте 2D пространства.

Игровой персонаж – объект, перемещением которого в 2D пространстве управляет пользователь.

Содержание

1 Введение 6

1.1 Наименование программы 6

1.2 Характеристика области применения программы 6

2 Основания для разработки 6

2.1 Документы, на основании которых ведется разработка 6

2.2 Организация, утвердившая документ, и дата его утверждения 6

2.3 Наименование разработки 6

3 Назначение разработки 7

4 Требования к программе или программному изделию 7

4.1 Требования к функциональным характеристикам программы 7

4.1.1 Требования к составу выполняемых функций 7

4.1.2 Организация входных и выходных данных 9

4.2 Требования к надежности 9

4.2.1 Требования к обеспечению надежного и устойчивого функционирования программы 9

4.2.2 Время восстановления после отказа 9

4.2.3 Отказы из-за некорректных действий пользователя 10

4.3 Условия эксплуатации 10

4.3.1 Климатические условия эксплуатации 10

4.3.2 Требования к численности и квалификации персонала 10

4.4 Требования к составу и параметрам технических средств 10

4.5 Требования к информационной и программной совместимости 11

4.5.1 Требования к исходным кодам и языкам программирования 11

4.5.2 Требования к программным средствам, используемым программой 11

5 Требования к программной документации 11

6 Стадии и этапы разработки 11

7 Порядок контроля и приемки 12

7.1 Виды испытаний 12

7.2 Общие требования к приемке работы 12

Приложение А Описание интерфейса программы 13

Приложение Б Описание 2D пространства 15

Приложение В Описание карты 2D пространства 16

Приложение Г Формат описания модели пространства 17

Приложение Д Диаграмма прецедентов 19

Приложение Е Структура файла выходных данных 20

Введение

Наименование программы

Наименование ­­– «Модуль генерации «2D прогулок». Краткое наименование – программа.

## Характеристика области применения программы

Программа представляет собой функциональный модуль для использования в мобильных приложениях на платформе Android. Программа предполагает использование в медицинских и реабилитационных центрах для людей с ограниченными интеллектуальными возможностями, а также в семьях, в которых есть люди с ограниченными интеллектуальными возможностями.

Основания для разработки

Документы, на основании которых ведется разработка

Разработка ведется на основании задания на бакалаврскую работу.

Организация, утвердившая документ, и дата его утверждения

Документ утвердила д.т.н., доц. кафедры САПР и ПК Шабалина О.А. Дата утверждения документа: «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Наименование разработки

Наименование темы разработки – «Разработка модуля генерации «2D прогулок».

Назначение разработки

Разрабатываемая программа реализует генерацию, визуализацию и моделирование прохождения маршрута на карте 2D пространства. Программа предназначена для развития и поддержки навыка ориентирования в пространстве с помощью карты.

Требования к программе или программному изделию

Программа должна обеспечивать процесс генерации оптимального маршрута, визуализации его на карте 2D пространства и моделирования его прохождения.

Описание интерфейса программы приведено в Приложении А.

Описание 2D пространства приведено в Приложении Б.

Описание карты 2D пространства приведено в Приложении В.

Требования к функциональным характеристикам программы

Требования к составу выполняемых функций

Программа должна обеспечить выполнение следующих функций:

* загрузка исходных данных для построения маршрута:

1. загрузка карты 2D пространства (описание формата представления карты приведено в Приложении Г);
2. загрузка списка строковых идентификаторов объектов, через которые должен быть проложен маршрут (описание формата представления объектов приведено в Приложении Г).

* построение размеченного маршрута:

1. формализация 2D пространства в виде взвешенного неориентированного графа;
2. нахождение кратчайших путей между вершинами, которым соответствуют заданные для построения маршрута объекты, с использованием алгоритма A\*;
3. построение оптимального маршрута обхода вершин, которым соответствуют заданные объекты, с использованием жадного алгоритма;
4. разбиение построенного маршрута на участки.

* визуализация игрового персонажа на карте пространства:

1. визуализация антропоморфной модели игрового персонажа;
2. визуализация анимации движения (ходьбы) персонажа при перемещении его пользователем.

* моделирование прохождения маршрута игровым персонажем:

1. визуализация участка маршрута в виде ломаной линии на карте пространства;
2. визуализация конечной точки маршрута в виде маркера на карте пространства;
3. перемещение игрового персонажа в плоскости карты с помощью элементов управления;
4. подсчет длины пути, пройденного персонажем.

* передача управления внешнему модулю;
* сохранение информации о прохождении маршрута в файл JSON:

1. сохранение даты и времени начала прохождения маршрута;
2. сохранение даты и времени окончания прохождения маршрута;
3. сохранение длины пути, пройденного персонажем, и оптимальной длины для каждого завершенного участка маршрута;
4. сохранение отметки завершения прохождения маршрута.

* визуализация результатов прохождения маршрута:

1. вывод окна оценки прохождения участка маршрута на основании сравнения длины пути, пройденного пользователем, с длиной оптимального пути для текущего участка маршрута: отображение позитивного смайлика, если длина пройденного пути превышает длину оптимального пути не более чем на 50%; отображение негативного смайлика, если длина пройденного пути превышает длину оптимального пути более чем на 50%;
2. отображение результата прохождения маршрута в виде списка оценок (смайликов), полученных за пройденные участки маршрута.

Функциональность приложения в виде диаграммы прецедентов приведена в Приложении Д.

Организация входных и выходных данных

Входными данными являются идентификатор карты 2D пространства и список объектов на карте, которые необходимо обойти.

Выходными данными является JSON файл c информацией о пройденном маршруте. Структура данных, содержащихся в файле, приведена в приложении Е.

Требования к надежности

Требования к обеспечению надежного и устойчивого функционирования программы

Надежное функционирование программы должно быть обеспечено совокупностью следующих организационно-технических мероприятий:

* организацией бесперебойного питания технических средств;
* использованием лицензионного программного обеспечения.

Время восстановления после отказа

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств. Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем операционной системы, не должно превышать времени, требуемого на восстановление подачи электропитания и запуск программы.

Отказы из-за некорректных действий пользователя

Отказы средства возможны из-за некорректных действий пользователя при работе с операционной системой. Во избежание возникновения отказов средства не следует предоставлять права администратора конечному пользователю.

Условия эксплуатации

Климатические условия эксплуатации

Климатические условия должны удовлетворять требованиям, предъявленным к техническим средствам в части условий их эксплуатации.

Требования к численности и квалификации персонала

Минимальная численность персонала, требуемого для работы средства – 1 конечный пользователь.

Требования к составу и параметрам технических средств

Техническое средство должно быть мобильным устройством на платформе OS Android 4.4 и выше.

Требования к информационной и программной совместимости

Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке C#. В качестве среды разработки должна быть использована Unity Personal 2018.

Требования к программным средствам, используемым программой

Системные программные средства, используемые разрабатываемой программой, должны быть лицензионными.

Требования к программной документации

Состав программной документации должен включать в себя техническое задание на разработку и проектирование программы, пояснительную записку и исходные коды программы.

Стадии и этапы разработки

Проектирование программы должно включать в себя стадии, приведенные в Таблице 1.

Таблица 1 – Сроки выполнения работ

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование стадии | Сроки |
| 1 | Анализ требований пользователя | 03.10.2019 – 04.11.2019 |
| 2 | Разработка технического задания | 05.11.2019 – 09.12.2019 |
| 3 | Проектирование программы | 10.12.2019 – 20.01.2020 |
| 4 | Реализации программы | 21.01.2020 – 30.04.2020 |
| 5 | Тестирование программы | 01.05.2020 – 31.05.2020 |

Порядок контроля и приемки

Виды испытаний

Приемо-сдаточные испытания должны проводиться на защите выпускной работы бакалавра. Приемные мероприятия программы должны проводиться не позднее мая 2020 года.

Общие требования к приемке работы

Возможность приемки программы должна определяться соответствием всем пунктам данного технического задания.

Приложение А

Описание интерфейса программы

Интерфейс программы состоит из двух экранов: моделирования прохождения маршрута и визуализации результатов.

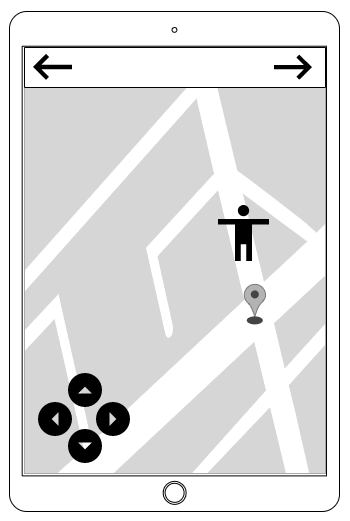
Элементы интерфейса экрана моделирования прохождения маршрута:

­­– карта 2D пространства;

­­– кнопки управления перемещением игрового персонажа;

­­– кнопки перехода между экранами (для связи с другими экранами приложения, в котором используется модуль).

Макет интерфейса представлен на рис.1.



Кнопки управления перемещением игрового персонажа

Карта 2D пространства

Кнопки перехода между экранами

Рисунок 1 – Макет интерфейса экрана моделирования прохождения маршрута

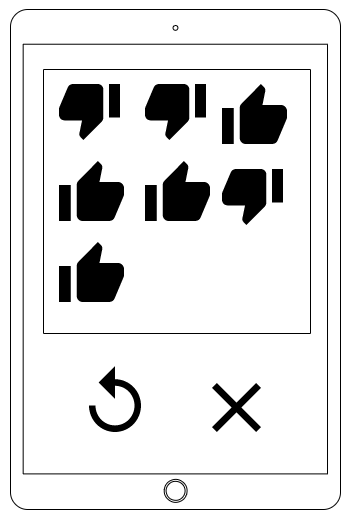
Элементы интерфейса экрана визуализации результатов:

­­– визуализация результатов прохождения маршрута;

­­– кнопка повторного прохождения маршрута;

­­– кнопки завершения прохождения маршрута.

Макет интерфейса представлен на рис.2.



Визуализация результатов прохождения маршрута

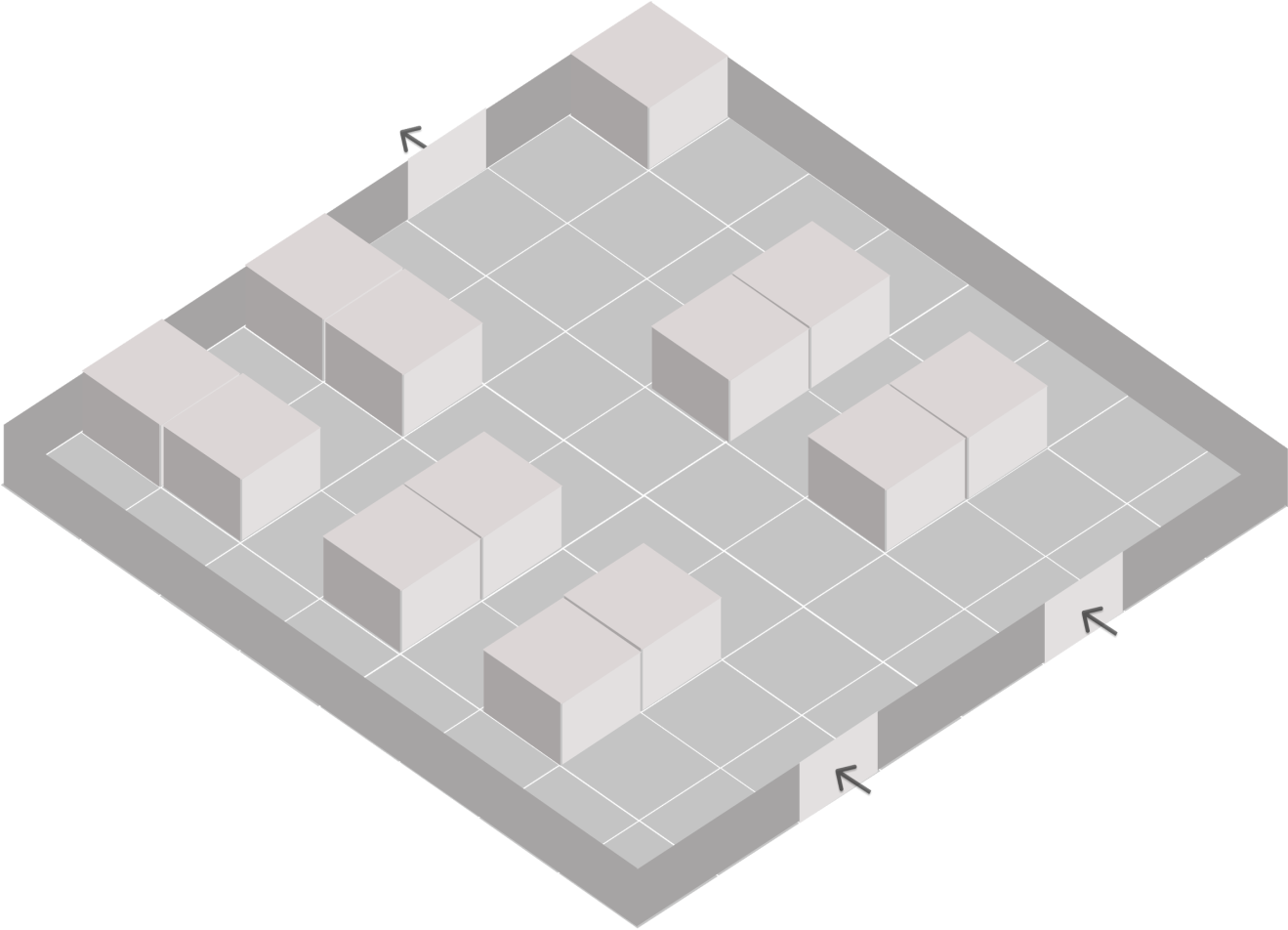
Кнопки повторного прохождения и завершения прохождения маршрута

Рисунок 2 – Макет интерфейса экрана визуализации результатов

# Приложение Б

# Описание 2D пространства

Макет и элементы 2D пространства представлены на рис.3



Вход

Объект

Выход

Граница

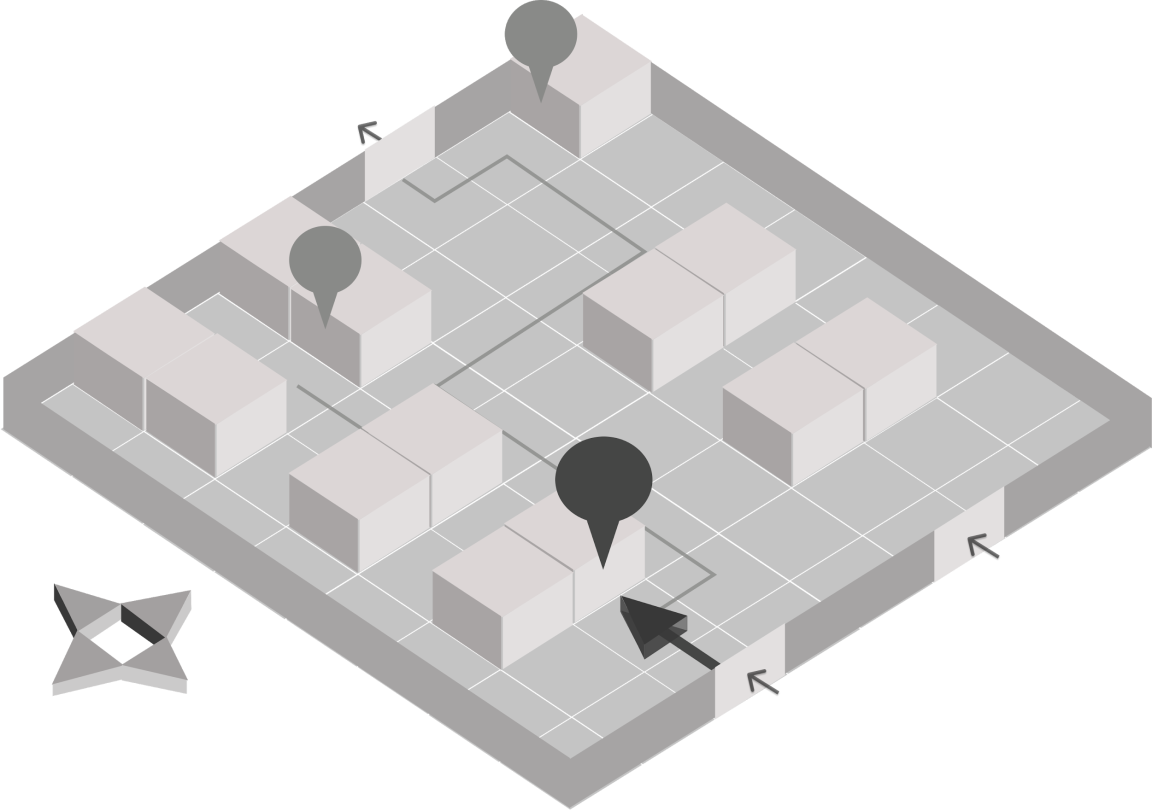
Рисунок 3 – 2D пространство

# Приложение В

# Описание карты 2D пространства

Макет и элементы карты 2D пространства представлены на рис.4

Объекты из списка объектов



Элементы управления перемещением игрового персонажа

Текущий участок маршрута

Игровой персонаж

Размеченный маршрут

Рисунок 4 – карта 2D пространства

# Приложение Г

# Формат описания модели пространства

Описание модели пространства должно осуществляться при помощи языка сериализации данных YAML. Каждый объект в пространстве должен быть записан в файл как отдельный YAML документ и представлен в описании пространства через последовательность –. Первая строка каждого документа должна содержать запись «!u!<класс объекта> &<идентификационный номер объекта>». Каждое сериализуемое свойство должно быть записано с новой строки и поименовано с префиксом «m\_»

Пример описания модели пространства:

%YAML 1.1

%TAG !u! tag:unity3d.com,2011:

--- !u!header

SerializedFile:

m\_TargetPlatform: 4294967294

m\_UserInformation:

--- !u!29 &1

Scene:

m\_ObjectHideFlags: 0

m\_PVSData:

m\_QueryMode: 1

m\_PVSObjectsArray: []

m\_PVSPortalsArray: []

m\_ViewCellSize: 1.000000

--- !u!127 &3

GameManager:

m\_ObjectHideFlags: 0

--- !u!157 &4

--- !u!1 &7

GameObject:

m\_ObjectHideFlags: 0

m\_PrefabParentObject: {fileID: 0}

m\_PrefabInternal: {fileID: 0}

importerVersion: 3

m\_Component:

- 4: {fileID: 9}

- 20: {fileID: 10}

- 92: {fileID: 15}

- 124: {fileID: 16}

m\_Layer: 0

m\_Name: Main Camera

m\_TagString: MainCamera

m\_Icon: {fileID: 0}

m\_NavMeshLayer: 0

m\_IsActive: 1

…

# Приложение Д

# Диаграмма прецедентов

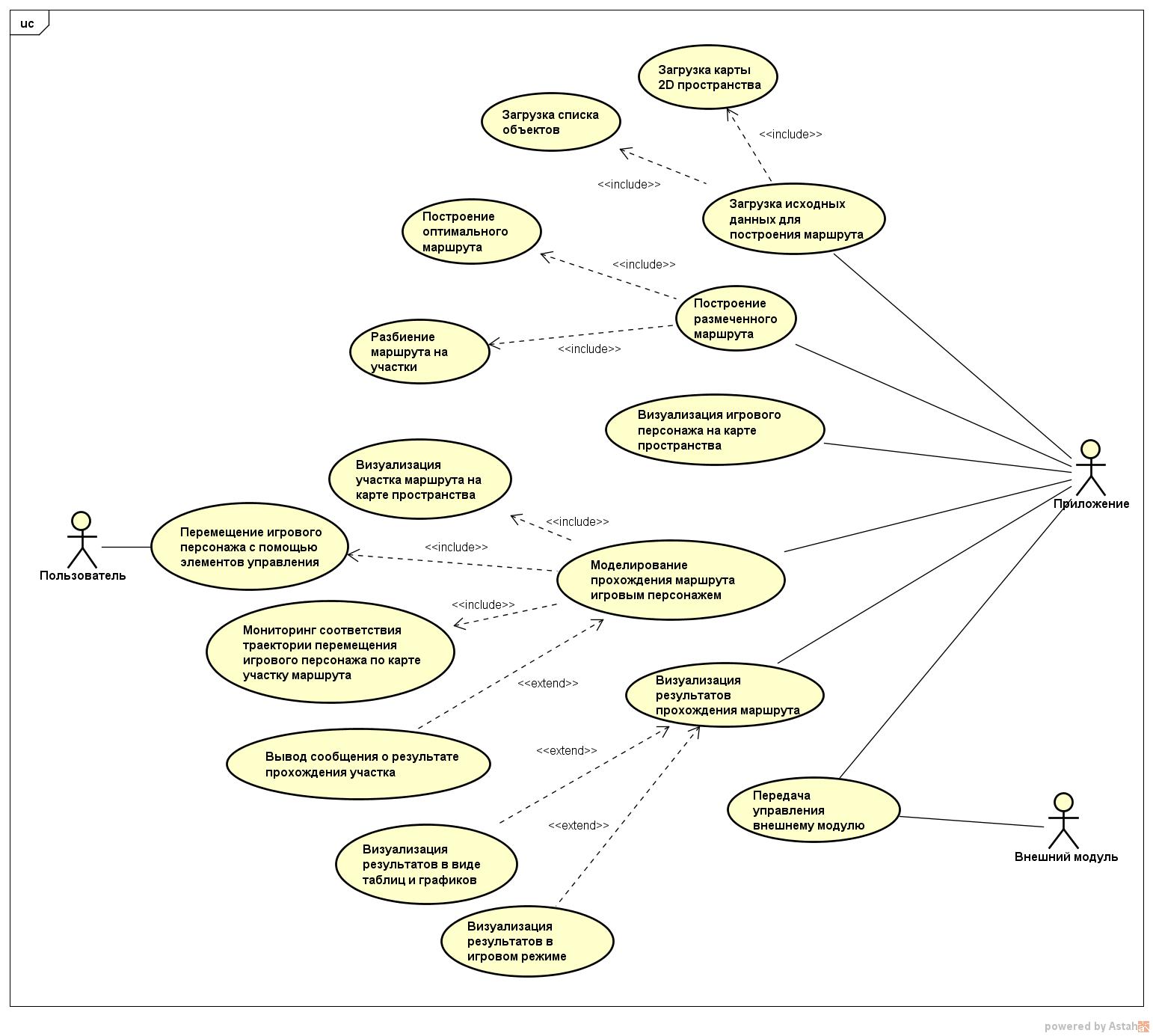


Рисунок 5 – Диаграмма прецедентов

# Приложение Е

# Структура файла выходных данных

{

"SpaceLabel": <строка >,

"BeginningTime": <дата и время >,

"EndingTime": <дата и время >,

"Path":

[

{

"EndPoint": <строка >,

"Time": <время >,

"PathPartLength": <число>,

"OptimalPathPartLength": <число>

}, …

     ],

    "IsPathCompleted": <логическое значение>

}