МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ   
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)»**

Факультет «Управление»

Кафедра «Автоматизированные системы управления»

09.03.02 – «Информационные системы и технологии»

**Допустить к защите**

Зав. кафедрой «АСУ»

/Максимычев О.И./

«\_\_\_\_» 2023г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ**

**БАКАЛАВРА**

**Реализация парковочного сервиса средствами**

**агентного моделирования**

|  |  |  |
| --- | --- | --- |
| Выполнил: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Бернецян А.Э. |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Волосова А.В. |
| Консультант  по экономической части: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Васильева Ю.И. |
| Консультант по БЖД: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Якубович А.Н. |

**Москва – 2023г.**

**МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)**

Кафедра «Автоматизированные системы управления»

шифр и направление подготовки:

09.03.02 – «Информационные системы и технологии»

направленность (профиль):

«Информационные системы и технологии в отраслях транспортно-дорожного комплекса»

УТВЕРЖДАЮ

зав. кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Максимычев О. И. /

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

**ЗАДАНИЕ**

**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

Бернецян Арам Эдикович

Тема ВКР: **«Реализация парковочного сервиса средствами**

**агентного моделирования»**

утверждена приказом МАДИ от «16» марта 2023 г. № 834 ст

1. Обоснование темы ВКР и перечень подлежащих разработке вопросов:

Обоснование: помощь в определении наилучших подходов и стратегий для улучшения парковочного сервиса

Перечень вопросов:

* 1. Разработать модель парковки
  2. Определить диапазоны часов для машин организаций рядом с парковкой
  3. Отобразить занятые и свободные парковочные места в режиме реального времени.
  4. Оформить бронирование места на парковке
  5. Настроить систему счетчиков для определения въезда – выезда машины.
  6. Рассчитать обеспечение условий безопасности жизнедеятельности
  7. Рассчитать экономическую эффективность

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Наименование этапа работы (раздела) | Ф.И.О.должность, уч. степень, звание руководителя/консультанта (этапа работы раздела) | Срок выполнения этапа, раздела | Подпись, дата | |
| Задание выдал | Задание принял |
| 1 | Проектирование информационных моделей и разработка алгоритмов (MODELS AND METHODS) | Волосова А.В., доцент, к.т.н. | 15.03-31.03.23 |  |  |
| 2 | Технология программной реализации разработанных моделей и алгоритмов (RESULTS) | Волосова А.В., доцент, к.т.н. | 1.04-30.04.23 |  |  |
| 3 | Обеспечение условий безопасности жизнедеятельности (LIFESAFETY) | Якубович А.Н.,  профессор, д.т.н. | 1.05-15.06.23 |  |  |
| 4 | Расчет экономической эффективности (ECONOMICS) | Васильева Ю.И.,  доцент, к.э.н. | 1.05-15.06.23 |  |  |

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Волосова А.В. /

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Бернецян А.Э. /

Оглавление

[ВВЕДЕНИЕ (INTRODUCTION) 7](#_Toc138098862)

[ГЛАВА 1. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ МОДЕЛЕЙ И РАЗРАБОТКА АЛГОРИТМОВ (MODELS AND METHODS) 10](#_Toc138098863)

[1.1. Анализ предметной области 10](#_Toc138098864)

[1.2. Выбор средств технической реализации. 12](#_Toc138098865)

[1.3. Принцип работы «Роевого алгоритма» . 13](#_Toc138098866)

[1.4. Алгоритмs поиска кратчайшего пути. 15](#_Toc138098867)

[ГЛАВА 2. ТЕХНОЛОГИЯ ПРОГРАММНОЙ РЕАЛИЗАЦИИ РАЗРАБОТАННЫХ МОДЕЛЕЙ И АЛГОРИТМОВ (RESULTS) 20](#_Toc138098868)

[2.1. Настройка виртуального мира 21](#_Toc138098869)

[2.2. Режим 1. Стационарный режим парковки 23](#_Toc138098870)

[2.2.1. Модель парковки 23](#_Toc138098871)

[2.2.2. Внешний вид агентов 26](#_Toc138098872)

[2.2.3. Поведение агентов 27](#_Toc138098873)

[2.2.4. Заезд на парковочное место 30](#_Toc138098874)

[2.2.5. Управление временем на парковке 32](#_Toc138098875)

[2.2.6. Выезд с парковки 34](#_Toc138098876)

[2.2.7. Применение «Роевого алгоритм» 34](#_Toc138098877)

[2.3. Режим 2. Бронирование онлайн 37](#_Toc138098878)

[2.3.1. Модель парковки 39](#_Toc138098879)

[2.3.2. Обработчик нажатия мыши 42](#_Toc138098880)

[2.3.3. Бронирование места и оплата 45](#_Toc138098881)

[2.3.4. Применение алгоритма А\* 47](#_Toc138098882)

[ГЛАВА 3. РАСЧЕТ ОБЕСПЕЧЕНИЯ УСЛОВИЙ БЕЗОПАСНОСТИ ЖИЗНИДЕЯТЕЛЬНОСТИ (LIFESAFETY) 52](#_Toc138098883)

[3.1. Исходные данные 52](#_Toc138098884)

[3.2. Определение расчетных концентраций оксида углерода в воздухе рабочей зоны. 53](#_Toc138098885)

[3.3. Анализ условий труда и определение класса и уровня вредности 58](#_Toc138098886)

[3.4. Организация и проведение контроля содержания вредных веществ в воздухе рабочей зоны. 60](#_Toc138098887)

[3.5. Мероприятия по обеспечению безопасных условий труда. 62](#_Toc138098888)

[ГЛАВА 4. РАСЧЕТ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ (ECONOMICS) 64](#_Toc138098889)

[4.1. Расчет затрат на разработку ИС. 64](#_Toc138098890)

[4.1.1. Затраты на заработную плату разработчиков. 64](#_Toc138098891)

[4.1.2. Страховые взносы на заработную плату разработчиков. 67](#_Toc138098892)

[4.1.3. Затраты на материалы. 67](#_Toc138098893)

[4.1.4. Затраты на электроэнергию. 68](#_Toc138098894)

[4.1.5. Амортизация технических средств. 69](#_Toc138098895)

[4.1.6. Накладные расходы по разработке. 69](#_Toc138098896)

[4.2. Расчёт затрат на внедрение ИС 70](#_Toc138098897)

[4.2.1. Затраты на приобретение технических средств, стоимостью более 100 тысяч рублей. 70](#_Toc138098898)

[4.2.2. Затраты на подготовку персонала. 70](#_Toc138098899)

[4.3. Анализ спроса и прогнозирование дохода и расходов разработчика. 71](#_Toc138098900)

[4.3.1. Анализ спроса на информационную систему. 71](#_Toc138098901)

[4.3.2. Прогноз дохода разработчика от продажи прав на информационную систему. 72](#_Toc138098902)

[4.3.3. Прогноз расходов разработчика от продажи прав на информационную систему. 72](#_Toc138098903)

[4.4. Расчет экономической эффективности проекта. 73](#_Toc138098904)

[4.4.1. Финансовый план (расчет эффективности инвестиций в разработку и внедрение системы). 73](#_Toc138098905)

[4.4.2. Расчет ЧДД проекта. 73](#_Toc138098906)

[ЗАКЛЮЧЕНИЕ (CONCLUSIONS) 77](#_Toc138098907)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ (REFERENCES) 78](#_Toc138098908)

# ВВЕДЕНИЕ (INTRODUCTION)

В современных городах существуют серьезные проблемы, связанные с парковкой. Рост числа автомобилей и ограниченность пространства создают сложности в обеспечении удобства и эффективности движения транспорта. Парковочные сервисы играют важную роль в решении этих проблем, однако несмотря на наличие множества решений, вопросы эффективности и оптимального использования парковочных ресурсов остаются актуальными вызовами для городских властей и предпринимателей.

Сегодня становится все более сложно управлять потоком автомобилей и обеспечить каждому водителю место для парковки. С увеличением числа автомобилей и недостаточностью парковочных зон города оказываются под давлением. Это приводит к проблемам с транспортной инфраструктурой, пробкам на дорогах и неудовлетворенности автовладельцев.

Парковочные сервисы должны быть эффективными и гибкими, чтобы адаптироваться к постоянно меняющимся условиям городской среды. Оптимальное использование парковочных ресурсов является ключевым фактором для обеспечения удобства и снижения проблем с парковкой.

К сожалению, на сегодняшний день не существует универсального решения, которое было бы применимо ко всем городам и ситуациям. Каждый город имеет свои уникальные особенности, и эффективное решение проблем парковки требует индивидуального подхода. Разработка хорошего плана действий для улучшения парковки может отнимать много сил и времени. Однако, несмотря на это, нет никакой гарантии, что придуманный план будет идеальным для конкретной парковки.

С появлением и развитием технологий искусственного интеллекта, было разработано множество способов использования этой технологии. Компьютерные технологии и программные коды позволили достичь невероятных результатов, создавая подобие человеческого интеллекта. Теперь искусственный разум все более глубоко проникает в нашу повседневную жизнь, упрощая ее и повышая качество.

В своей работе я хочу провести исследование, направленное на улучшение парковочного сервиса. Одно из направлений искусственного интеллекта, методы которого могут быть применены, - это агентное моделирование. Агентное моделирование представляет собой мощный инструмент для анализа сложных систем и управления ими. С его помощью можно изучить, как различные стратегии могут повлиять на парковочный сервис и как сделать его более эффективным.

Агентное моделирование позволяет создавать виртуальных агентов, которые могут взаимодействовать с окружающей средой и другими агентами. Эти виртуальные агенты могут имитировать поведение реальных людей или других объектов, принимая во внимание различные факторы и принимая решения в реальном времени. Используя агентное моделирование, можно создать компьютерную модель парковочного сервиса, которая учитывает особенности конкретного города, его инфраструктуру, потоки движения и поведение водителей.

С помощью агентного моделирования можно проводить различные симуляции и эксперименты, чтобы выявить оптимальные стратегии управления парковочным сервисом. Можно изучить, как изменение цен на парковку, введение ограничений или использование новых технологий может повлиять на доступность парковочных мест, время поиска места, пропускную способность и удовлетворенность пользователей. Таким образом, агентное моделирование может помочь определить наилучшие подходы и стратегии для улучшения парковочного сервиса.

Использование искусственного интеллекта и агентного моделирования в парковочных сервисах предоставляет новые возможности для оптимизации и улучшения эффективности системы. Благодаря этим инновационным методам можно создать более удобное и эффективное парковочное пространство, что приведет к улучшению мобильности и комфорта жителей города.

Цель данной дипломной работы заключается в разработке модели для реального парковочного сервиса.

Разработанная модель позволит решить такие задачи оптимизации парковочного сервиса, как:

- структурная организация парковочных мест;

- управление движением автомобилей;

- реализация сервиса бронирования и оплаты паркинг, в том числе и через мобильное приложение;

- реализация паркинга для льготных категорий пользователей.

Моя работа имеет практическую значимость, поскольку может помочь городским властям и предпринимателям улучшить парковочные сервисы и сделать их более удобными для автовладельцев. Я надеюсь, что моя работа позволит принять более обоснованные решения и оптимизировать использование парковочных ресурсов в городах.

Для достижения цели необходимо выполнить следующие задачи:

1. Разработать модель парковки
2. Определить диапазоны часов для машин организаций рядом с парковкой
3. Отобразить занятые и свободные парковочные места в режиме реального времени.
4. Оформить бронирование места на парковке
5. Настроить систему счетчиков для определения въезда – выезда машины.
6. Настроить бронирование и оплату парковочного места.

# ГЛАВА 1. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ МОДЕЛЕЙ И РАЗРАБОТКА АЛГОРИТМОВ (MODELS AND METHODS)

1.1. Анализ предметной области

Объектом нашего исследования является подземная парковка, расположенная по адресу Москва, район Аэропорт, Кочновский проезд, 4к2. Парковка является подземным сооружением больших размеров, предназначенным для стационарного хранения и стоянки автомобилей. Вход на подземную парковку осуществляется снаружи здания бизнес-центра.



Рисунок 1.1. Въезд на парковку.

Парковочные места на данной парковке отделены специальной разметкой, которая определяет границы каждого места и обеспечивает правильное размещение автомобилей. Внутри парковки можно заметить, что места для парковки автомобилей расположены в рядах, обозначенных широкими проездами между ними.



Рисунок 1.2. Парковка изнутри.

На схеме парковки можно увидеть, что места для автомобилей выстроены в прямоугольные ряды с промежутками между ними для движения автомобилей.

Каждое парковочное место имеет определенные размеры и обозначение, которое может быть нанесено на пол или стены в виде номеров или символов. Это позволяет водителям легко найти свое место и правильно припарковаться.

Также внутри парковки могут быть установлены дополнительные элементы, такие как освещение, указатели направления движения, информационные таблички или индикаторы свободных мест. Это помогает улучшить навигацию и обеспечить более удобное использование парковочных мест. Схема парковки включает в себя 6 рядов с местами для парковки, причем каждый ряд содержит от 12 парковочных мест. Кроме того, между каждыми 3 парковочными местами находится опорная стенка.

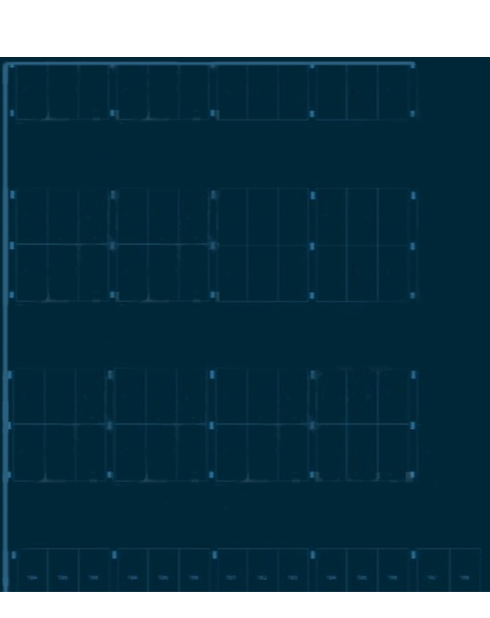


Рисунок 1.3. Макет парковки.

1.2. Выбор средств технической реализации.

Среди самых популярных средств агентного моделирования в научном сообществе можно выделить следующие:

* NetLogo. Является одним из наиболее широко используемых инструментов агентного моделирования. Он предоставляет простой и доступный интерфейс, поддерживает графическую визуализацию моделей и имеет активное сообщество пользователей и разработчиков.
* AnyLogic. мощная и гибкая среда моделирования, которая позволяет создавать агентно-ориентированные модели, а также модели на основе других подходов, таких как дискретно-событийное моделирование и системная динамика. Она широко применяется в академических и коммерческих исследованиях.
* Repast: Популярная среда моделирования агентного моделирования и социального моделирования. Она предоставляет богатый набор инструментов и библиотек для создания разнообразных моделей агентов и сложных систем.
* GAMA: Открытая платформа для моделирования и разработки агентно-ориентированных моделей сложных систем. Она отличается гибкостью и поддерживает различные типы агентов и их взаимодействие.
* Mason: Популярная библиотека Java для агентного моделирования. Она проста в использовании, позволяет создавать модели агентов и правила взаимодействия с минимальными усилиями.

Программа NetLogo выбрана для разработки макета подземной парковки и анализа парковочного сервиса из-за ее высокой популярности и широкого признания в сообществе исследователей. Благодаря простому и интуитивному интерфейсу, NetLogo позволяет легко создавать модели агентов без необходимости глубоких знаний программирования. Однако, несмотря на свою простоту, программа обладает мощными возможностями визуализации данных, что позволяет наглядно представить динамику парковочного сервиса и анализировать результаты моделирования. Этот сочетание популярности, интуитивного интерфейса и возможности визуализации делает NetLogo превосходным выбором для исследования и улучшения парковочного сервиса.

1.3. Принцип работы «Роевого алгоритма» .

Агентное моделирование является мощным инструментом для изучения и имитации поведения коллективных систем, таких как стаи птиц, рои рыб, социальные сети и многие другие. В рамках агентного моделирования роевой алгоритм играет важную роль, позволяя имитировать сложное поведение коллектива на основе простых правил и взаимодействия между его индивидуальными частями.



Рисунок 1.4. Роевой алгоритм на примере стаи птиц

Понятие роевого алгоритма:

Роевой алгоритм - это алгоритм, вдохновленный поведением коллективных систем в природе, таких как рои птиц, рыб или насекомых. Он базируется на идее совместного взаимодействия множества индивидуальных агентов, каждый из которых следует простым правилам поведения. В результате совместных действий агентов формируется эмерджентное поведение, то есть поведение всего коллектива, которое может быть сложным и координированным.

Основные шаги принципа работы роевого алгоритма:

* Инициализация: Начальное состояние роя определяется позицией и скоростью каждого агента. Обычно это случайно выбираемые значения в заданном диапазоне.
* Определение локального окружения: Каждый агент определяет свое локальное окружение, то есть ближайших соседей, с которыми он будет взаимодействовать. Это может быть определено, например, как агентов, находящихся в определенном радиусе вокруг данного агента.
* Обновление скорости и позиции: Агенты обновляют свою скорость и позицию на основе информации о своем окружении и правилах движения. Например, агент может стремиться двигаться в направлении средней скорости своих соседей или избегать столкновений с другими агентами.
* Взаимодействие и обмен информацией: Агенты обмениваются информацией друг с другом, чтобы координировать свои действия и адаптироваться к изменяющейся среде. Это может быть сигналы об опасности, обнаруженные ресурсы или просто обновленные позиции и скорости.
* Итерационный процесс: Шаги 3 и 4 повторяются в цикле до достижения определенного условия остановки, например, фиксированного числа итераций или достижения желаемого состояния.

В данной исследовательской работе рассматривается возможность применения метода роевого алгоритма с целью обеспечить безопасное перемещение агентов на парковке, предотвращая столкновения и поддерживая оптимальное расстояние между ними. Необходимо разработать эффективную стратегию, которая позволит агентам успешно маневрировать и выбирать оптимальные места для парковки, учитывая при этом безопасность и избегая столкновений с другими агентами. Использование роевого алгоритма позволит нам скоординировать действия агентов и обеспечить правильное управление их движением на парковке, минимизируя риски возникновения аварийных ситуаций.

1.4. Алгоритмs поиска кратчайшего пути.

Бывают ситуации, когда владелец автомобиля резервирует место на парковке, которую он никогда раньше не посещал. В таких случаях, к сожалению, существует риск потеряться в сложной схеме парковки. Для решения этой проблемы может понадобиться алгоритм поиска пути. Существует несколько алгоритмов поиска кратчайшего пути, которые могут помочь владельцу автомобиля найти оптимальный маршрут к его зарезервированному месту парковки.

Рассмотрим наиболее известные алгоритмы, которые применяются для поиска кратчайшего пути:

* Алгоритм Дейкстры
* Алгоритм Беллмана-Форда
* Алгоритм Флойда-У оршелла
* Алгоритм Джонсона
* Алгоритм A\*

Алгоритм Дейкстры - это алгоритм поиска кратчайшего пути во взвешенном графе без отрицательных весов ребер. Он был разработан нидерландским ученым Эдсгером Дейкстрой в 1956 году. Алгоритм Дейкстры широко применяется в различных областях, связанных с оптимизацией пути или передачей данных, таких как маршрутизация в компьютерных сетях, графовые базы данных, транспортные системы и другие.

Алгоритм Дейкстры основан на жадной стратегии выбора следующей вершины для раскрытия. Он работает пошагово, постепенно наращивая длину известного кратчайшего пути от начальной вершины до каждой другой вершины графа.

Плюсы:

* Гарантированно находит кратчайший путь от начальной вершины до всех остальных вершин во взвешенном графе без отрицательных весов ребер.

Минусы:

* Не работает с графами, содержащими ребра с отрицательными весами.
* Не учитывает эвристическую информацию о целевой вершине.

Алгоритм Беллмана-Форда - это алгоритм поиска кратчайшего пути от одной вершины до всех остальных вершин во взвешенном ориентированном или неориентированном графе. Он был разработан Ричардом Беллманом и Лестером Фордом в 1958 году.

Алгоритм Беллмана-Форда основан на идее динамического программирования и использует итеративный подход для нахождения кратчайшего пути. Он может обрабатывать графы с ребрами положительного или отрицательного веса, а также обнаруживать наличие отрицательных циклов.

Плюсы:

* Работает с графами, содержащими ребра с отрицательными весами.

Минусы:

* Сложность времени выполнения составляет O(V3), где V - количество вершин.
* Требует большого объема памяти для хранения матрицы расстояний между всеми парами вершин.
* Медленно работает на больших графах

Алгоритм Флойда-Уоршелла - это алгоритм для нахождения кратчайших путей между всеми парами вершин во взвешенном ориентированном или неориентированном графе. Он был разработан Робертом Флойдом и Стивеном Уоршеллом в 1962 году.

Основная идея алгоритма Флойда-Уоршелла состоит в том, чтобы построить матрицу расстояний, где каждый элемент [i][j] представляет кратчайшее расстояние между вершинами i и j. Алгоритм использует динамическое программирование и работает с графами любого типа, включая графы с отрицательными весами ребер.

Плюсы:

* Находит кратчайшие пути между всеми парами вершин во взвешенном графе.

Минусы:

* Сложность времени выполнения составляет O(V3), где V - количество вершин.
* Требует большого объема памяти для хранения матрицы расстояний между всеми парами вершин.

Алгоритм Джонсона - это алгоритм для нахождения кратчайших путей между всеми парами вершин во взвешенном ориентированном графе, даже если граф содержит ребра с отрицательными весами. Алгоритм был разработан Майклом Джонсоном в 1977 году.

Основная идея алгоритма Джонсона заключается в добавлении фиктивной вершины и изменении весов ребер таким образом, чтобы граф не содержал отрицательных циклов. Затем используется модификация алгоритма Беллмана-Форда для нахождения кратчайших путей из фиктивной вершины до всех остальных вершин. Наконец, с помощью алгоритма Дейкстры на основе этих измененных весов ребер находятся кратчайшие пути между всеми парами вершин.

Плюсы:

* Работает с графами, содержащими ребра с отрицательными весами.
* Позволяет находить кратчайшие пути между всеми парами вершин во взвешенном графе.

Минусы:

* Сложность времени выполнения составляет O(Velog(V)), где V - количество вершин, E - количество ребер.
* Медленно работает на больших графах.

Алгоритм A\* (A-star) - это информированный алгоритм поиска кратчайшего пути от начальной вершины к целевой вершине в графе. Он был разработан Питером Хартом, Нилом Нильсоном и Бертраном Раффелем в 1968 году. Алгоритм A\* является комбинацией алгоритма Дейкстры и эвристического поиска.

Основная идея этого алгоритма заключается в оценке эвристической функцией стоимости оставшегося пути от текущей вершины до целевой вершины. Он использует комбинацию стоимости пути от начальной вершины до текущей и эвристической оценки стоимости оставшегося пути для выбора следующей вершины для раскрытия. Это позволяет алгоритму A\* находить оптимальные пути с учетом информации о цели.

Плюсы:

* Использует эвристическую информацию для эффективного поиска кратчайшего пути.
* Может быть легко модифицирован для учета различных ограничений и факторов
* Часто работает быстрее других алгоритмов на практике.

Минусы:

* Не гарантирует нахождение оптимального пути во всех случаях.
* Не подходит для графов с отрицательными весами.

В конечном итоге я выбрал алгоритм А\*. Сделал я это по нескольким причинам. Прежде всего, он обладает внушительными достоинствами, которые делают его привлекательным для моих целей. Алгоритм А\* известен своей эффективностью и скоростью в нахождении оптимального пути в графе или сетке.

Кроме того, недостатки алгоритма А\* будут перекрываться программной средой, в которой он будет реализован. Это означает, что я смогу оптимизировать его работу и минимизировать его слабые стороны с помощью дополнительных модификаций и настройки. В итоге, я считаю, что выбор алгоритма А\* является отличным решением для моей задачи.

# ГЛАВА 2. ТЕХНОЛОГИЯ ПРОГРАММНОЙ РЕАЛИЗАЦИИ РАЗРАБОТАННЫХ МОДЕЛЕЙ И АЛГОРИТМОВ (RESULTS)

Создание макета парковки высокого качества и соответствующего реальным размерам и характеристикам объектов играет важную роль в разработке и анализе парковочного сервиса. Чтобы достичь наилучших результатов, необходимо учесть следующие этапы:

1. Настройка мира: После создания проекта необходимо настроить среду, в которой будет размещен макет парковки. Этот этап включает выбор правил и параметров для виртуальной среды, в которой будут размещены объекты парковки.
2. Модель парковки: Сначала необходимо разработать проект макета парковки. Это включает определение масштаба, размеров и расположения парковочных мест, учет разметки, наличие опорных стенок и других характеристик парковки.
3. Вид агентов: При разработке макета парковки важно выбрать тип агента, который будет использоваться для моделирования поведения водителей и управления парковочным сервисом. Также необходимо определить характеристики и модель агента.
4. Поведение агентов: Кодирование является важным шагом в разработке макета парковки, поскольку через код можно создать виртуальный мир и определить поведение агентов.

И раз уж на то пошло, то я решил внутри одной модели создать два режима работы. Первый из них будет стационарным и предоставит возможность наблюдать за парковочным сервисом в целом, воссоздавая реальную ситуацию. В этом режиме можно будет наблюдать, как машины заезжают на парковку, занимают свободные места и в конечном итоге покидают ее.

А второй режим будет включать в себя функциональность бронирования места и оплаты. Тут я воссоздаю процесс бронирования парковочного места, где пользователи могут выбирать доступные места, резервировать их и производить оплату за использование. Это позволит им планировать заранее и обеспечит удобство использования парковки.

Таким образом, два режима работы внутри модели позволят мне изучить парковочный сервис как в его обычном функционировании, так и в контексте бронирования и оплаты мест. Это поможет мне лучше понять и оптимизировать процессы парковки и создать удобную среду для пользователей.

2.1. Настройка виртуального мира

При настройке виртуального мира, вы должны учесть несколько базовых параметров, которые определяют его общий вид и характеристики. Эти параметры включают размер мира, размер патча мира и шрифт.

Размер мира определяет пространственные размеры вашего виртуального окружения. Он может быть задан в терминах количества патчей или физических единиц измерения. Размер мира может варьироваться от небольших размеров для узких симуляций до значительных размеров для более широких и сложных сред.

Размер патча мира определяет физические размеры каждого патча в вашем виртуальном мире. Патчи являются базовыми строительными блоками вашей симуляции и могут быть представлены в виде квадратов или шестиугольников. Размер патча может быть настроен таким образом, чтобы соответствовать вашим потребностям и масштабу модели.

Шрифт относится к стилю и размеру текста, используемого для отображения информации в виртуальном мире. Вы можете выбрать подходящий шрифт, который обеспечит четкое и удобочитаемое отображение текста для ваших агентов и других элементов в симуляции.

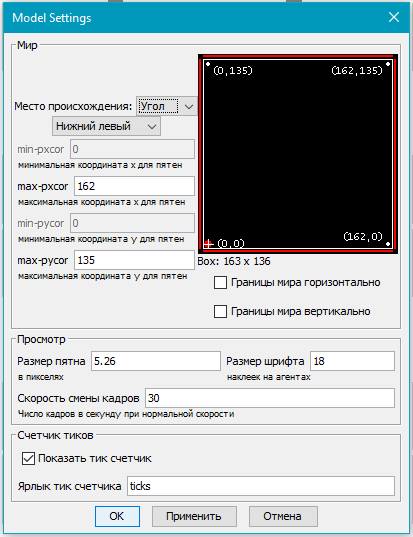


Рисунок 2.1. Настройки модели виртуального мира.

2.2. Режим 1. Стационарный режим парковки

Цель стационрарного режима состоит в том, чтобы точно отразить текущие процессы парковки. Я провел исследование и разработал модель, которая демонстрирует парковочный сервис в его текущем состоянии. В этой модели вы можете наблюдать, как агенты, выступающие в роли автоводителей, заезжают на парковку и занимают свободные парковочные места. Благодаря настройке поведения агентов, они способны выполнять те же действия, что и человек. От времени будет зависеть, как долго будет агент находиться на парковочном месте.



Рисунок 2.2. Работа режима 1.

2.2.1. Модель парковки

На данном этапе я приступаю к созданию макета самой парковки, включая отображение стен и размещение мест для парковки автомобилей. Для этого я буду использовать команды окраски патчей в коде.

При обращении ко всем патчам (Patches) в модели, в можно использовать команду "set pcolor Color" для того, чтобы окрасить их в нужный цвет. Однако, если нужно выбирать только определенные патчи на основе определенного условия (например их расположения), то нужно использовать условные выражения в коде. В данной работе я буду опираться на координаты Х и У патчей. Чтобы обратиться к Х и У координатам нужно использовать их собственные параметры pxcor и pycor.

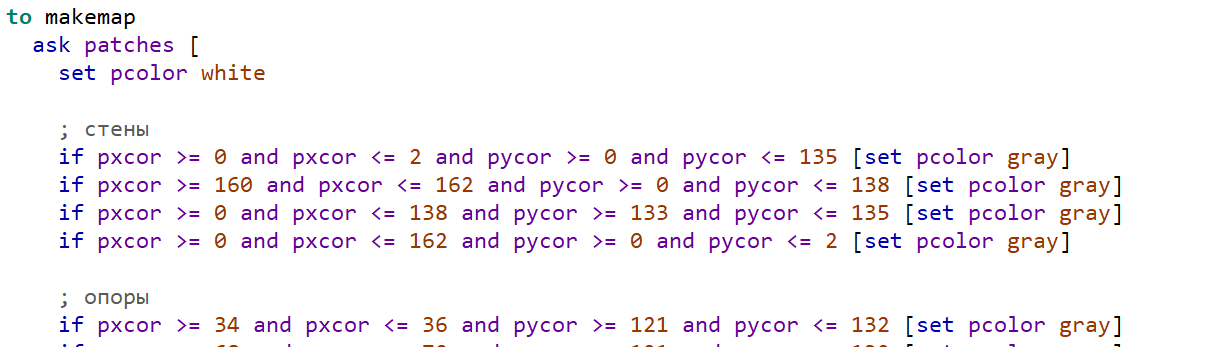


Рисунок 2.3. Пример кода, позволяющий окрашивать виртуальный мир.

Чтобы обозначить места для парковки, я применю светло-серый цвет к соответствующим патчам. Именно на них должны будут заезжать будущие агенты чтобы припарковать свою машину. Это позволит наглядно выделить области, предназначенные для стоянки автомобилей. Также я буду использовать серый цвет для окраски патчей, представляющих стены парковки. Это поможет визуально отделить их от остальных элементов макета.

Каждый патч, представляющий парковочное место, может иметь различные статусы, которые влияют на поведение агентов. Ниже приведены некоторые из статусов, которые присвоены патчам:

Free? Этот статус указывает, занято ли парковочное место текущим

агентом. Если место занято, агенты будут искать другие свободные места

для парковки. Если же оно свободно, агент может выбрать его чтобы припарковаться

VIPplace. Этот статус указывает на то, является ли парковочное место VIP-местом. Вип-места могут иметь особые правила или привилегии, например, возможность использования их только определенными категориями агентов или установленные ограничения времени парковки. Статус никогда не меняется.

Таблица 2.1. Анализ статуса места free?

|  |  |
| --- | --- |
| **Событие** | **Статус места** |
| Создание мира | True |
| Занятие места агентом | False |
| Освобождение места агентом | True |

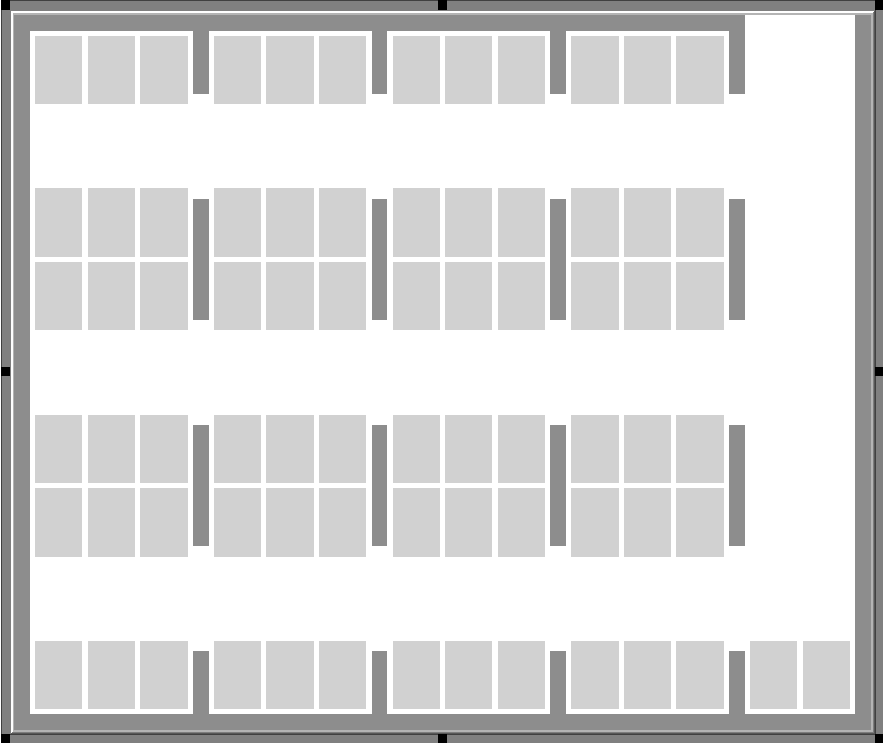


Рисунок 2.4. Окрашенный виртуальный мир для режима 1.

Применение различных цветов позволяет наглядно отобразить структуру парковки и обозначить основные элементы. Таким образом, я создаю виртуальную модель парковки, которая будет максимально приближена к реальному объекту.

2.2.2. Внешний вид агентов

При выборе формы для представления автомобилей в нашей модели, пришлось учесть ограничения обычной формы "Car", которая может быть видна только сбоку и имеет ограниченное количество направлений обзора. По умолчанию, параметр Rotatable для этой формы отключен, что ограничивает нашу способность определить направление движения автомобиля.

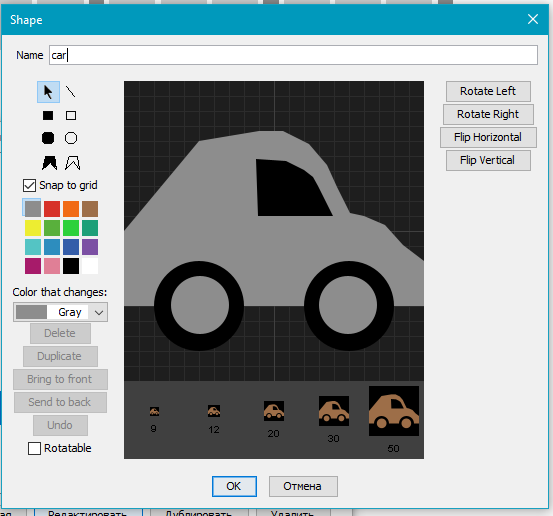


Рисунок 2.5. Вид агента "Car".

Для того чтобы преодолеть это ограничение, я обратился к библиотеке форм и нашел другой тип агента, который лучше соответствует моим требованиям. Я выбрал форму "CarTop", которая позволяет видеть автомобиль сверху и имеет возможность разворачиваться в любом направлении. Это дает возможность точно определить направление движения автомобиля и отслеживать его перемещения в разрабатываемой мной модели.

Таким образом, выбор формы агента "CarTop" помогает точнее представлять движение автомобилей и соответствует требованиям отображения направления движения агента. Отображение направления движения агента является важным моментом для нас, поэтому не стоит пренебрегать этим аспектом.

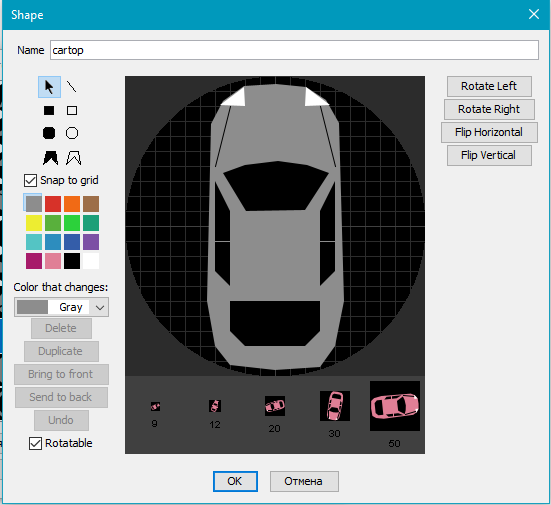


Рисунок 2.6. Вид агента "CarTop".

Таким образом, выбор остановился на агенте "CarTop", который позволяет точно отображать направление движения автомобилей в нашем макете парковки. Благодаря этому выбору я смогу более реалистично моделировать и анализировать поведение автомобилей в парковочном сервисе.

2.2.3. Поведение агентов

Поведение агентов на карте является ключевым аспектом данной работы. Корректная настройка кода позволяет агентам обрабатывать информацию о местности, анализировать своё окружение, определять доступные маршруты и принимать решения на основе этих данных. Это включает в себя умение выбирать наилучший путь для перемещения, избегать препятствий, принимать во внимание других агентов и действовать согласованно с заданными правилами или целями. И всё это они делают не требуя вмешательства человека. Таким образом, агенты могут имитировать действия реального человека, что является одним из применений искусственного интеллекта.

При разработке алгоритма поведения агентов, можно использовать различные стратегии и правила для выбора места парковки. Некоторые из возможных подходов могут включать оценку доступных мест на основе их расстояния от входа, предпочтение ближайших мест или выбор наиболее удобного места в соответствии с определенными критериями, такими как близость к выезду, размеры или чистота места.

После выбора свободного места, агенты перемещаются на него и занимают его. Нужно заметить, что агенты должны учитывать доступность места, чтобы избежать коллизий или занятия уже занятых мест.

В текущей работе поведение агентов организовано с использованием концепции статусов. Каждому агенту присваивается начальный статус "0" при въезде на парковку. В этом режиме агенты спокойно движутся по макету парковки в поисках своего парковочного места. Они следуют установленным правилам движения и взаимодействия с другими агентами.

По достижении парковочного места, у агента происходит изменение их статуса работы. Это может быть связано с заездом на парковочное место, освобождением места или выполнением других действий на парковке. При смене статуса агента, внутренние переменные и параметры обновляются в соответствии с новым режимом работы.

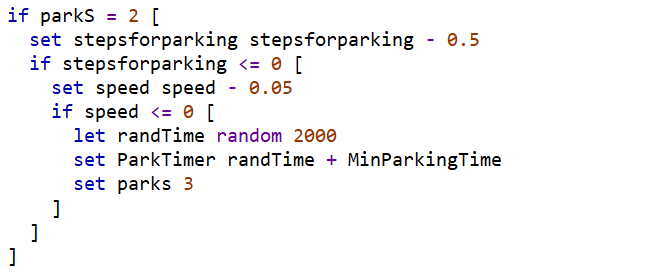


Рисунок 2.7. Смена внутренних параметров в зависимости от статуса.

Таблица 1.2. Анализ статуса агента Parks

|  |  |
| --- | --- |
| Событие | Статус агента |
| Создание агента | Set Parks = 0 |
| Достижение координаты У для поворота в сторону парковочного места | Set Parks = 1 |
| Достижение координаты Х для заезда на парковочное место | Set Parks = 2 |
| Успешная парковка | Set Parks = 3 |
| Выход с парковочного места | Set Parks = 4 |

Использование статусов позволяет эффективно управлять поведением агентов и контролировать их действия на макете парковки. Каждый статус может иметь свои особенности и правила, определяющие дальнейшие действия агента. Это помогает создать более реалистичную модель парковки и учесть различные сценарии работы автомобилей на парковке.

2.2.4. Заезд на парковочное место

При разработке алгоритма поведения агентов для парковки, можно использовать различные стратегии и правила. На созданной модели парковки предполагается, что агенты заезжают с правого верхнего угла. Они движутся вниз вдоль дороги и разворачиваются в нужном направлении, чтобы подъехать к своему парковочному месту.

Процесс создания агента начинается с вызова команды:

create-turtles 1

После создания агента, можно произвести настройку его внутренних параметров, которые определяют его внешний вид и поведение. Например, можно задать цвет агента, используя команду "set color", и указать его размер с помощью команды "set size". Также, можно задать цель парковки агента, используя команду "set target", чтобы определить, к какому конкретному месту агент должен двигаться.

Когда агент приезжает на парковку, он инициирует процесс поиска свободного парковочного места. Агент осматривает каждый патч парковки и проверяет внутренний параметр, отвечающий за занятость места в данный момент. Если агент обнаруживает свободное место, он начинает двигаться в его направлении.

Если появившийся агент является VIP-персоной, то у него есть особая привилегия занимать льготные места на парковке. В таком случае, при выборе места для парковки, льготные места получают приоритет, так как они имеют меньшую вероятность быть занятыми.

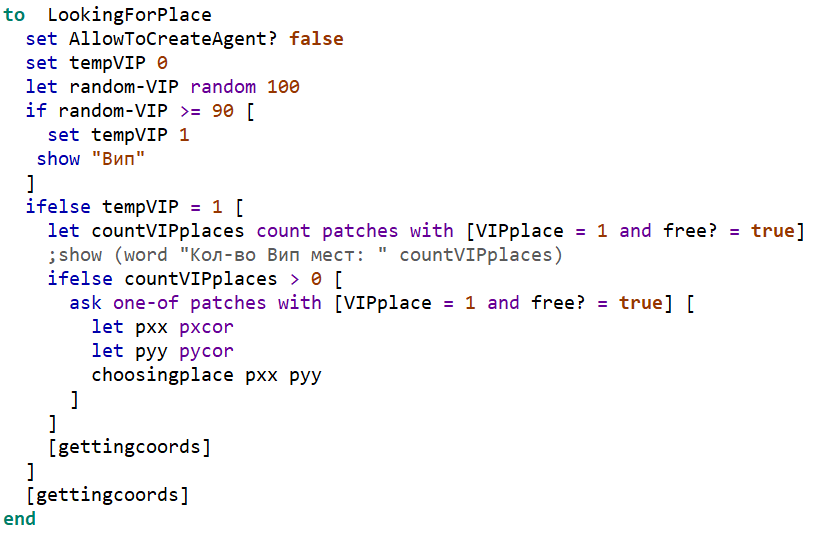


Рисунок 2.8. Функция поиска свободного места.

Для перемещения агента в модели используется команда Forward X,   
где X представляет собой количество патчей, которое агент преодолеет за один шаг. Однако, вместо использования фиксированного значения "X", можно использовать динамическую переменную, которая будет определять скорость агента.

Например, можно ввести переменную "speed", которая будет представлять скорость агента. Значение этой переменной может изменяться в зависимости от определенных условий или правил. Например, если агент сталкивается с препятствием или другим агентом, то его скорость может быть уменьшена. Соответственно, при выполнении команды "forward speed" агент будет перемещаться на расстояние, определяемое текущим значением переменной "speed".

Для выбора места парковки можно использовать разные алгоритмы. Некоторые из возможных подходов включают оценку доступных мест на основе их расстояния от входа. Агент может предпочитать ближайшие места, чтобы сократить время и расстояние до своего места парковки.

Также можно реализовать алгоритм выбора наиболее удобного места, учитывая определенные критерии. Например, агент может предпочитать место, близкое к выезду, чтобы снизить время, необходимое для покидания парковки. Он также может учитывать размеры места, чтобы убедиться, что его автомобиль поместится комфортно. Критерием выбора места может быть также его чистота или наличие дополнительных удобств, таких как близость к входу в здание.

2.2.5. Управление временем на парковке

В модели можно предусмотреть учет времени, на которое агент будет занимать парковочное место. Это позволит учесть длительность парковки и освободить место для других автомобилей после истечения указанного времени. Допустим, у нас есть два сценария: один, когда агент приезжает на парковку в ночное время, чтобы оставить машину и пойти домой спать, и другой, когда агент приезжает днем, чтобы зайти в магазин рядом.

В первом случае, агенту потребуется длительное время нахождения на парковке, так как его цель - оставить машину на длительный период. Можно установить соответствующую продолжительность времени для этого агента, чтобы он оставался на парковке в течение указанного времени, не совершая активных действий. Это позволит эмулировать ситуацию, когда машина стоит на парковке весь ночной период, пока владелец спит дома.

Во втором случае, когда агент приезжает днем для кратковременного посещения магазина, можно установить более короткое время занятия парковочного места. После истечения этого времени, агент заберет свою машину и освободит место для других автомобилей.

Для воплощения идеи отображения модельного времени в вашей парковке была разработана функция "SetTime". Эта функция отвечает за установку и обновление модельного времени в системе парковки.

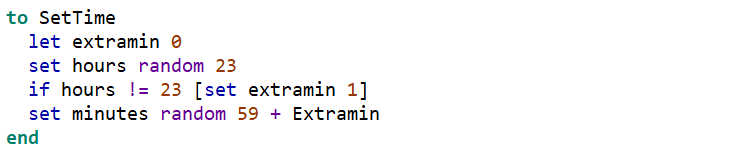
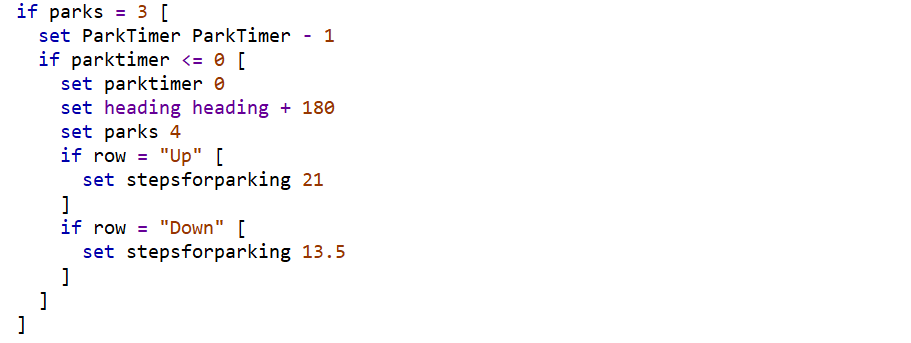


Рисунок 2.9. Функция определения модельного времени.

В данной работе при въезде на парковочное место агенту присваивается значение во внутренний параметр "parktimer". Это значение представляет собой время, на которое агент занимает парковочное место. В процессе работы модели время идет своим ходом.

При истечении времени, заданного в "parktimer", происходит изменение статуса агента. Это изменение статуса сигнализирует о необходимости освободить парковочное место. Агент освобождает свое место и двигается к выходу из парковки.

Рисунок 2.10. Функция освообождения парковочного места по исечению времени parktimer.

Таким образом, учет времени позволяет управлять доступностью парковочных мест и эффективно использовать их ресурсы. Это также добавляет реалистичность в модель, учитывая различные ситуации, когда агенты могут занимать парковку на разные промежутки времени в зависимости от своих потребностей или задач.

2.2.6. Выезд с парковки

Когда время, отведенное агенту на парковочном месте, истекает, он должен покинуть парковку. Этот процесс состоит из двух основных шагов.

Первым шагом является освобождение парковочного места, чтобы оно могло быть использовано другими автомобилями. Агент сигнализирует о своем намерении освободить место, и внутренние параметры, связанные с состоянием парковочного места, обновляются. Например, статус места изменяется на "свободное", и информация о состоянии места становится доступной для других агентов или системы управления парковкой.

После освобождения места агент начинает движение к выходу с парковки. Он выбирает оптимальный маршрут, и с помощью команды "Forward" или других соответствующих команд перемещается по дорогам парковки в направлении выхода. В процессе движения агент может взаимодействовать с другими автомобилями или препятствиями на пути, принимая решения в соответствии с заданными правилами и стратегией.

Таким образом, агент, с истечением времени, активно освобождает парковочное место и направляется к выходу, чтобы завершить свое пребывание на парковке. Это позволяет эффективно управлять использованием мест и обеспечивает непрерывную доступность парковки для других автомобилей.

2.2.7. Применение «Роевого алгоритм»

На данном этапе, когда я настроил движение агентов по парковке, пришло время подумать о том, что во время движения один агент может наехать на другого. При пропытках решить этоу проблему я пришел к выводу, что нужно использовать "роевой алгоритм".

Использование роевого алгоритма для контроля расстояния и предотвращения столкновений между агентами является разумным подходом. Роевые алгоритмы, такие как алгоритм роя частиц (PSO) или алгоритм роя пчел (ABC), имеют механизмы, которые позволяют агентам совместно работать и обмениваться информацией для достижения определенной цели или решения задачи.

В случае управления движением агентов на карте, роевой алгоритм может быть применен для контроля расстояний между агентами. Это может включать определение оптимального расстояния, которое должно быть поддерживаемо между агентами, и разработку механизмов, которые позволяют агентам подстраивать свою скорость или траекторию, чтобы избегать столкновений.

Например, в алгоритме роя частиц агенты движутся по пространству, обновляя свое положение и скорость на основе лучших результатов, которые они исследовали. В контексте предотвращения столкновений, агенты могут обмениваться информацией о своем положении и скорости, а также оценивать расстояния до ближайших агентов. Если расстояние становится слишком маленьким, агенты могут регулировать свою скорость или изменять свою траекторию, чтобы избежать столкновений.

Таким образом, роевой алгоритм может помочь агентам контролировать расстояние между ними и принимать меры предосторожности, чтобы избежать столкновений. Это способствует безопасности и эффективности движения агентов на карте.

Однако при использовании данного алгоритма возникает проблема с производительностью модели при наличии трех или более агентов на сцене. Это связано с тем, что сложность вычислений и управления увеличивается с увеличением количества агентов. Принимая всё это было принято решение упростить алгоритм. С помощью приведенного ниже кода была разработана функция, которая осуществляет поиск агентов, находящихся перед текущим агентом. Это позволяет в случае опасности агенту резко снизить скорость в целях уменьшения вероятности катастрофы.

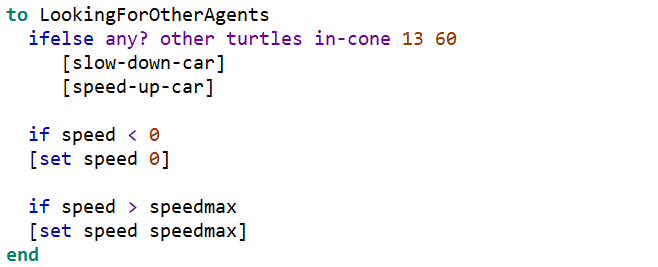


Рисунок 2.11. Функция поиска агентов перед собой.

На примере выполненной работы наблюдается ситуация, когда красный агент въезжает на парковку, но внезапно перед ним возникает синий агент. Это создает потенциальную угрозу столкновения между ними. Однако благодаря использованию метода роевого алгоритма, красный агент обнаруживает синего перед собой и быстро принимает меры для предотвращения столкновения. Он замедляет свою скорость, чтобы уступить дорогу и обеспечить безопасное прохождение синему агенту.

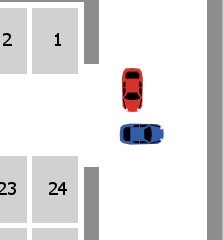


Рисунок 2.12. Возможное столкновение агентов

График скорости агентов позволяет наблюдать, как скорость красного агента рехко упала при приближении к синему агенту. Это служит доказательством работоспособности роевого алгоритма, который помогает контролировать расстояние и избегать столкновений между агентами.



Рисунок 2.13. Проверка скоростей агентов по графику.

2.3. Режим 2. Бронирование онлайн

Бронирование онлайн. В этой главе своей работы я буду рассматривать возможный способ реализации брони парковочного места с использованием онлайн-сервиса. В настоящее время отсутствуют реальные возможности для пользователей бронировать парковочное место онлайн, что представляет большой потенциал для разработки и внедрения новой модели.

В данном режиме реального времени пользователи имеют возможность наблюдать актуальную информацию о доступных свободных парковочных местах. Это позволяет пользователям выбирать и бронировать конкретное место на основе его доступности и предпочтений.

Система отображает на виртуальной карте парковку и помечает свободные и занятые места. С помощью цветовой индикации или других визуальных маркеров пользователи могут легко определить, какие места доступны для бронирования. Например, свободные места могут быть обозначены зеленым цветом. Пользователи могут щелкнуть на свободное место на карте, чтобы потом забронировать его

Создание такой модели позволит протестировать эффективность и удобство онлайн-бронирования места, а также определить потенциальные проблемы и недостатки.

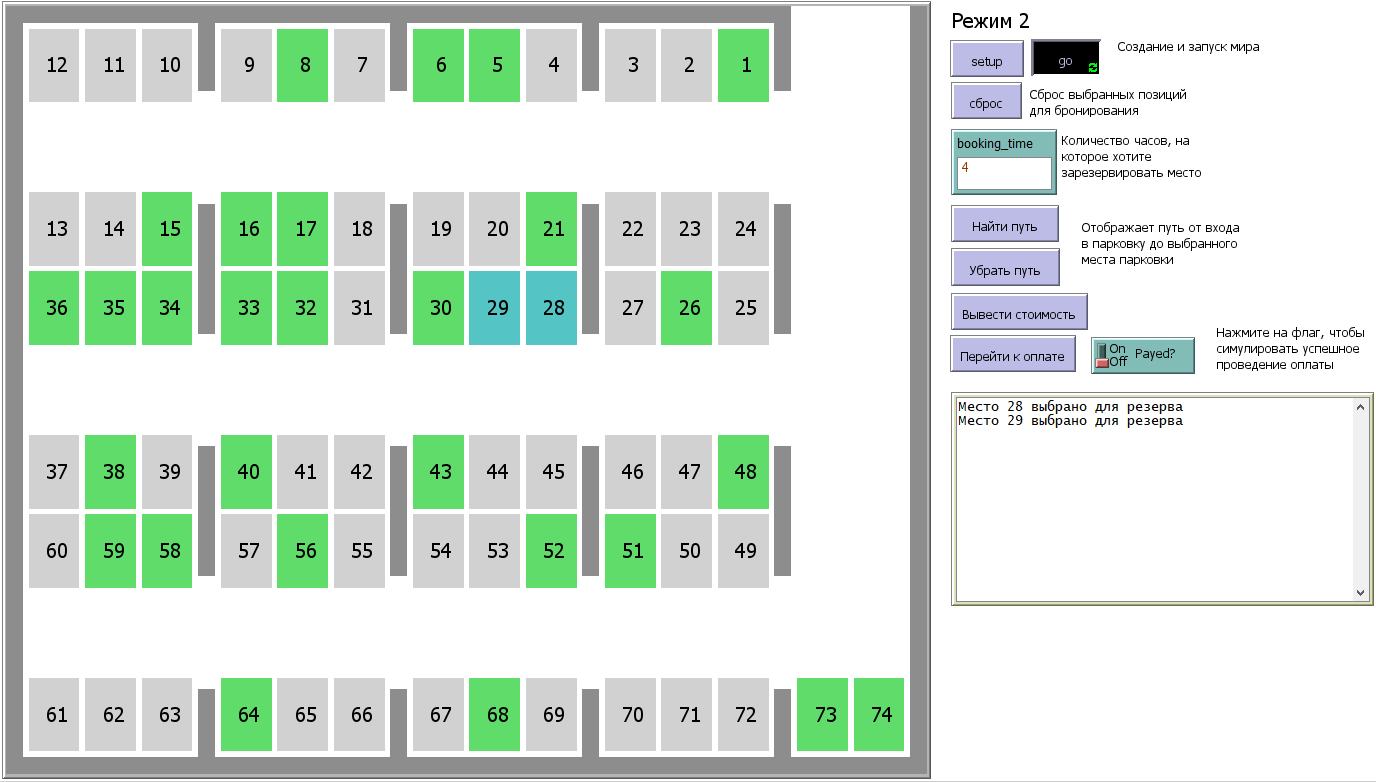


Рисунок 2.14. Работа режима 1.

Разработанный интерфейс модели может служить примером для разработки интерфейса мобильного приложения. Он предоставляет удобный способ взаимодействия с пользователем и обеспечивает эффективную функциональность, необходимую для создания приложения для бронирования мест онлайн.

Возможности интерфейса модели включают в себя:

1. Пользовательская аутентификация: Интерфейс обеспечивает возможность регистрации и входа пользователей в систему с использованием логина и пароля или других методов аутентификации, таких как отпечаток пальца или распознавание лица.
2. Поиск и просмотр доступных мест: Пользователи могут использовать интерфейс для поиска доступных мест, используя различные критерии, такие как местоположение, дата и время, тип места и другие параметры.
3. Бронирование места: После нахождения подходящего места, пользователь может выбрать его и забронировать через интерфейс. Это может включать указание даты, времени и других деталей бронирования.
4. Управление бронированиями: Интерфейс предоставляет пользователю возможность просматривать, изменять или отменять свои существующие бронирования. Это позволяет пользователям легко управлять своими планами и вносить необходимые изменения.
5. Оплата: Интерфейс может интегрироваться с платежными системами, чтобы пользователи могли производить оплату за забронированные места прямо из приложения. Это обеспечивает удобство и безопасность процесса оплаты.
6. Уведомления: Интерфейс может предоставлять возможность отправки уведомлений пользователям о подтверждении бронирования, изменениях в расписании или других важных обновлениях.
7. Персонализация: Интерфейс может предлагать функции персонализации, позволяющие пользователям сохранять предпочтения и получать рекомендации на основе их предыдущих действий и интересов.

В целом, разработка мобильного приложения на основе данного интерфейса модели может предоставить удобное и эффективное решение для бронирования мест онлайн, обеспечивая удобство пользователей и повышая их удовлетворенность использованием парковки.

2.3.1. Модель парковки

На данном этапе требуется разработать модель парковки, используя существующий макет из первого режима и внести необходимые изменения, чтобы определенные места были заняты и не отображались при бронировании.

Необходимо добавить новую функцию, которая будет управлять состоянием парковочных мест и определять, какие места должны быть заняты и не отображаться при бронировании. Эта функция будет играть важную роль в контроле доступности парковочных мест для бронирования. Таким образом можно имитировать настоящую ситуацию на парковке. Такая ситуация будет создаваться при запуске мира.

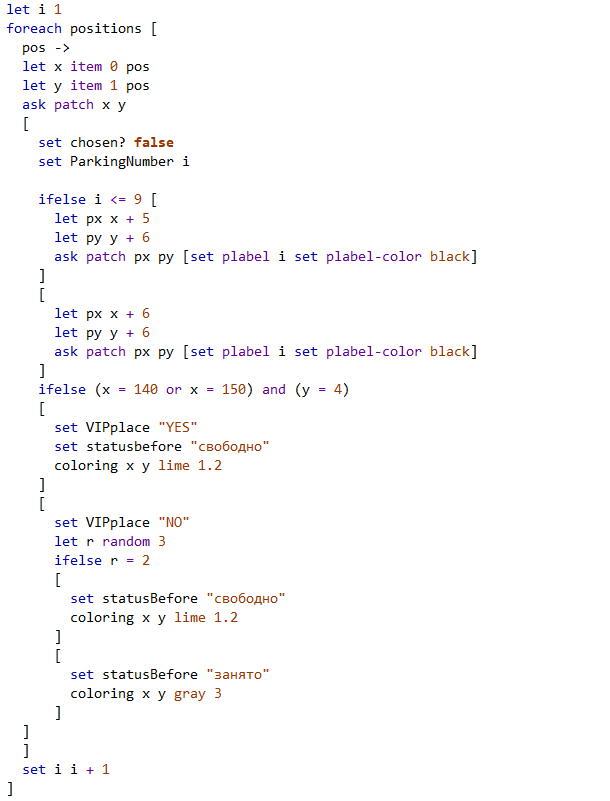


Рисунок 2.15. Функция определения доступных мест для бронирования.

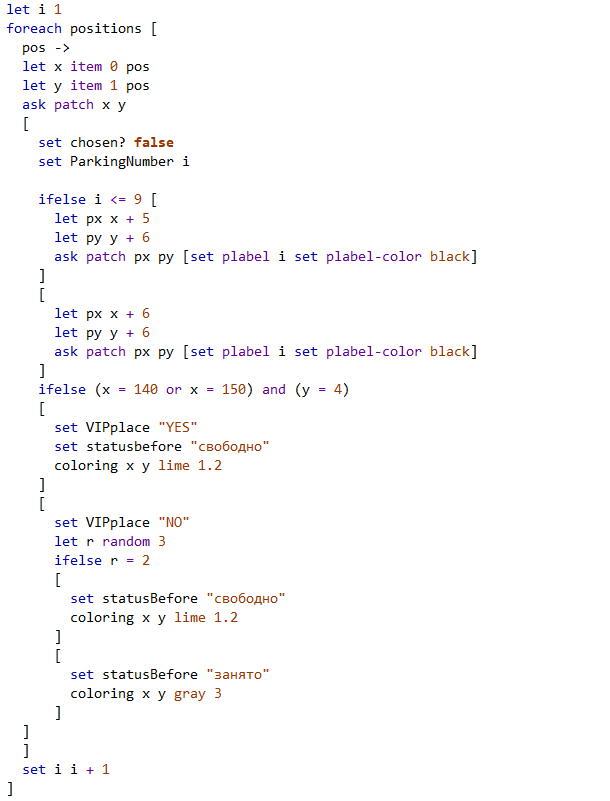


Рисунок 2.15. Функция определния доступных мест для бронирования. (Продолжение)

Таким образом, при создании модели парковочного сервиса с режимом бронирования места, мы получаем поле, которое отображает состояние парковки. В этой модели отсутствуют агенты, занимающиеся направлением и управлением движением автомобилей.

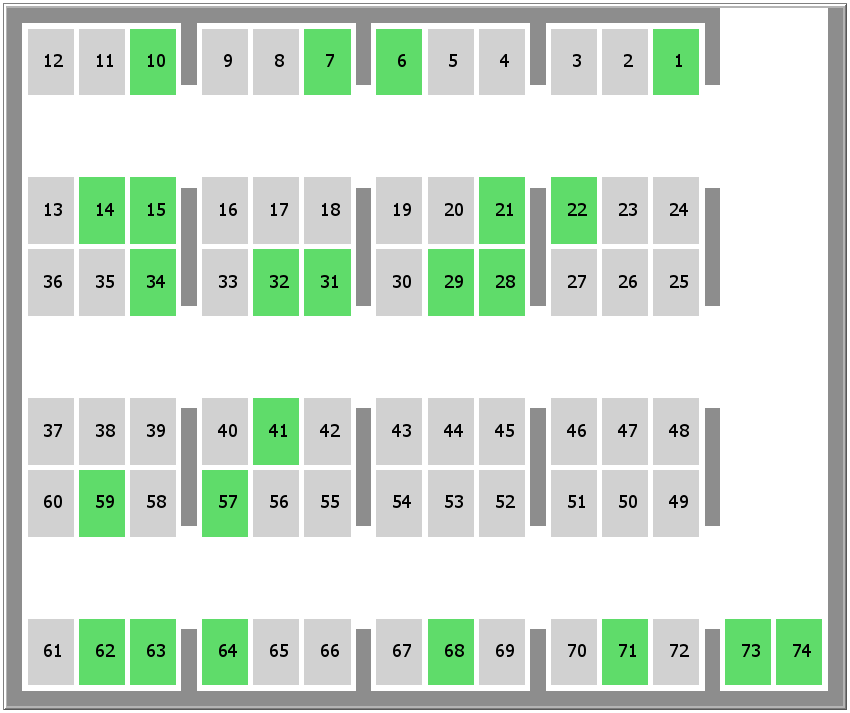


Рисунок 2.16. Окрашенный виртуальный мир для режима 2.

2.3.2. Обработчик нажатия мыши

Для реализации функциональности выбора парковочного места путем нажатия на него, необходимо создать обработчик нажатия в программе.

Обработчик нажатия будет реагировать на действия пользователя и выполнять соответствующие операции при выборе места на парковке. В данном случае, при нажатии на свободное место (оно будет отмечено зеленым местом), произойдет его бронирование.

Функция mouse-down? - это функция, которая возвращает значение истина или ложь в зависимости от того, зажата ли в данный момент кнопка мыши. Она может быть полезна при разработке обработчика нажатия по мыши на место парковки.

Чтобы построить функцию, которая обрабатывает нажатие на левую кнопку мыши (ЛКМ) вместо зажатия, необходимо добавить переменную mouse-clicked?. Эта переменная будет использоваться для отслеживания состояния нажатия на ЛКМ.

После того, как кнопка ЛКМ будет отпущена, значение переменной mouse-clicked? должно быть установлено на False, чтобы указать, что нажатие было завершено и функции перестали выполняться.

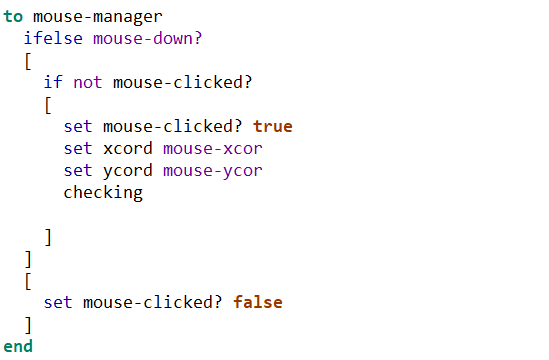


Рисунок 2.17. Обработчик нажатия кнопки мыши.

Чтобы реализовать функцию выбора свободных мест в качестве желаемых для бронирования, при нажатии на них, необходимо разработать функцию смены цвета на любой другой. В данном случае, мы будем менять зеленый цвет на голубой.

Чтобы реализовать функцию изменения цвета места, потребуется разработать функцию определения координат места на парковке. Определение координат места может быть довольно простым, особенно если мы используем функции mouse-xcor и mouse-ycor, которые позволяют мне получить координаты мыши в момент нажатия на место.

Внутри обработчика события, можно использовать функции mouse-xcor и mouse-ycor для получения координат мыши в момент нажатия. Например, x = mouse-xcor() и y = mouse-ycor().

Следующим шагом будет определение, находится ли патч с координатами X и Y внутри области с свободным парковочным местом. Если патч находится внутри этой области, мы будем менять цвет всего парковочного места на голубой, а номер места отправим в корзину для дальнейших действий.

Было принято решение реализовать функциональность, позволяющую снимать временное бронирование с выбранного (голубого) места, если пользователь снова нажал на него. Это полезно в случае, когда пользователь ошибочно добавил место в список для бронирования.А

Для реализации новых идей необходимо модифицировать функцию занятия парковочного места. Теперь каждое место будет иметь два внутренних параметра: "занято" и "выбрано для бронирования".

Первый параметр, "занято", будет отображать текущее состояние места, указывая, занято оно или нет. Начальное значение этого параметра можно установить на "False" для всех парковочных мест.

Второй параметр, "выбрано для бронирования", будет отражать, было ли данное место выбрано пользователем для бронирования. По умолчанию, начальное значение этого параметра также может быть установлено на "False" для всех парковочных мест.

При выполнении действий по занятию парковочного места, необходимо учитывать оба параметра. Если место уже занято (параметр "занято" равен "True"), то пользователь не сможет выбрать его для бронирования.

В случае, когда место свободно (параметр "занято" равен "False"), пользователь может выбрать его для бронирования, установив параметр "выбрано для бронирования" в значение "True". После этого место будет отображаться в голубом цвете (или любом другом обозначении, выбранном для выбранных мест) для обозначения его статуса.

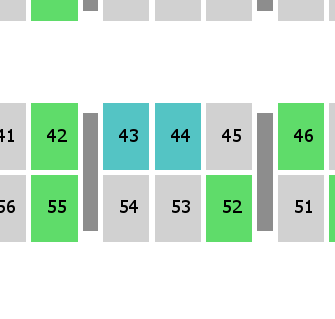


Рисунок 2.18. Окрашивание места при его выборе для бронирования.

Таким образом, с добавлением двух внутренних параметров для каждого места - "занято" и "выбрано для бронирования", вы сможете отслеживать и управлять состоянием каждого места. Это позволит реализовать новые функции, такие как предотвращение выбора уже занятого места для бронирования и обозначение выбранных мест для удобства пользователя.

2.3.3. Бронирование места и оплата

Как вы упомянули ранее, выбранные для бронирования места будут сохраняться в отдельном списке, который будет отслеживать количество выбранных мест для парковки.

Для реализации этой функциональности, вам потребуется создать отдельный список, который будет служить контейнером для хранения выбранных мест. В этом списке будут содержаться номера или другие идентификаторы выбранных мест.

Когда пользователь выбирает место для бронирования, мы добавляем его номер в список "Purchasing\_Cart" (корзину покупок). Этот список будет служить контейнером для хранения номеров мест, которые были выбраны для бронирования.

При добавлении выбранного места в "Purchasing\_Cart", мы просто используем функцию append() для добавления номера места в список. Если пользователь нажмет на голубую ячейку, будет считаться, что пользователь отказывается от брони этого места. В таком случае в массиве «Корзина покупок» удаляется номер парковочного места.

Добавление или удаление места из корзины сопровождается текстом, который выводится на экран. Он сообщает, какое место, и какое действие было произведено по отношению к нему.

При добавлении или удалении места из корзины (списка) сопровождаются текстовые сообщения, которые выводятся на экран, чтобы информировать пользователя о произведенном действии и соответствующем месте.

Так при добавлении места в корзину выводится сообщение вроде: "Место X выбрано для бронирования". Здесь "X" - это номер выбранного места, который был добавлен в корзину.

В противном случае – при удалении места из корзины выводится следующее сообщение: "Место X. Отказ от бронирования"

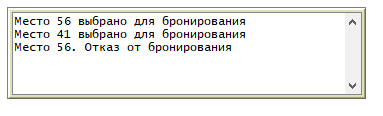


Рисунок 2.19. Выведение информации о работе со списком мест.

Таким образом, текстовые сообщения, которые выводятся на экран, являются важным сопровождающим элементом добавления или удаления места из корзины, обеспечивая информационную обратную связь для пользователя.

Помимо мест пользователь вправе выбрать время, на которое он собирается оплатить парковочное место. Если мы решили, что выбранные места окончательны для бронирования, то мы переходим к процессу оплаты. Для начала необходимо узнать количество забронированных мест и вычислить общую стоимость.

Для определения количества забронированных мест можно использовать функцию length, которое возвращает числоэлементов из списка (в нашем случае число парковочных мест, которые мы хотим забронировать).

Далее, для определения общей стоимости бронирования мест нужно умножить количество забронированных мест на стоимость одного места и время, на которое будет произведено бронирование.Вся информация конечно же должна быть высвечена на экран пользователя

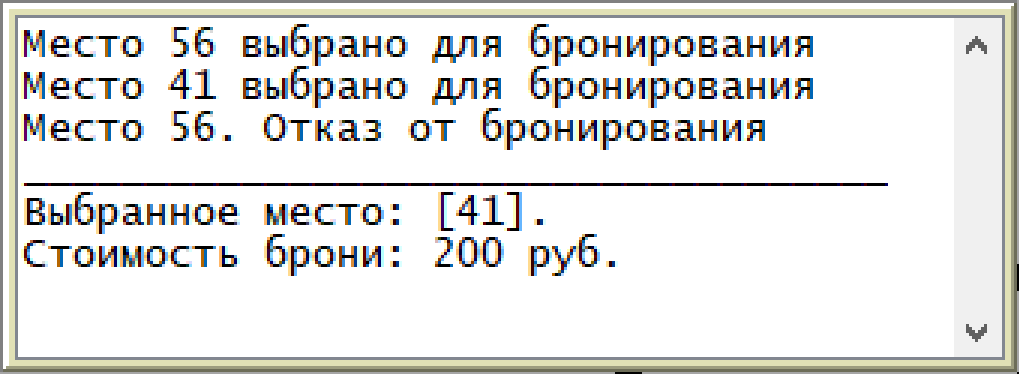


Рисунок 2.20. Вывод информации о текущем бронировании.

После того, как пользователь рассмотрел свои выбранные места и общую стоимость бронирования, он готов перейти к процессу оплаты.

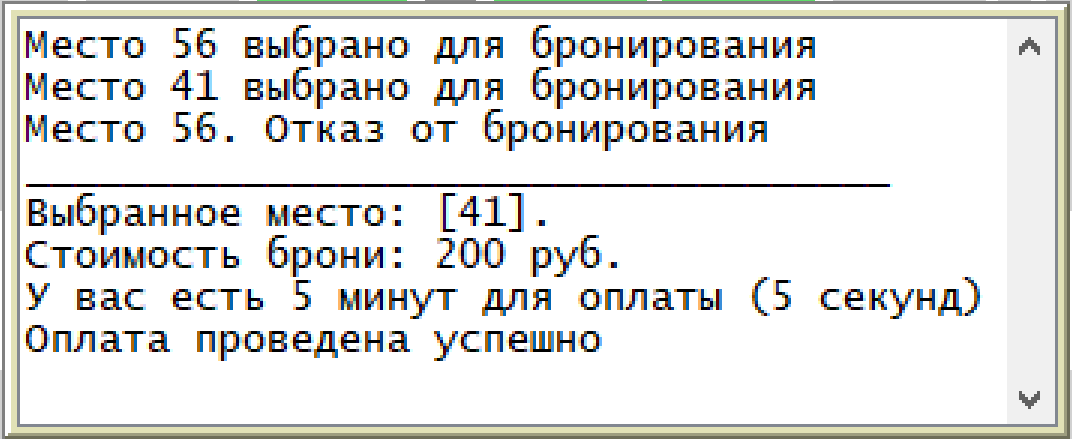


Рисунок 2.21. Успешное бронирование.

2.3.4. Применение алгоритма А\*

В программной реализации алгоритма, принимающей на вход две пары координат (начало и цель), процесс работы выглядит следующим образом:

1. Инициализация:
   1. Установить начальный патч и целевой патч.
   2. Установить начальные значения стоимости для всех патчей.
2. Вычисление стоимости пути:
   1. Вычислить стоимость пути от начального патча к его соседним патчам.
   2. Для каждого соседнего патча вычислить стоимость пути от начального патча до текущего патча.
3. Обновление стоимости и пути:
   1. Если новая стоимость пути от начального патча до текущего патча оказывается оптимальнее стоимости других возможных путей к этому патчу, то обновить стоимость пути и запомнить новый путь во внутренней переменной патча.
4. Поиск кратчайшего пути:
   1. Продолжать вычислять и обновлять стоимости и пути для всех соседних патчей, выбирая следующий наименее затратный патч.
   2. Повторять этот процесс, пока не будет достигнут целевой патч или не будут рассмотрены все доступные патчи.
5. Формирование кратчайшего пути:
   1. По завершении алгоритма, если целевой патч был достигнут, сформировать кратчайший путь, следуя обратно от целевого патча к начальному патчу.

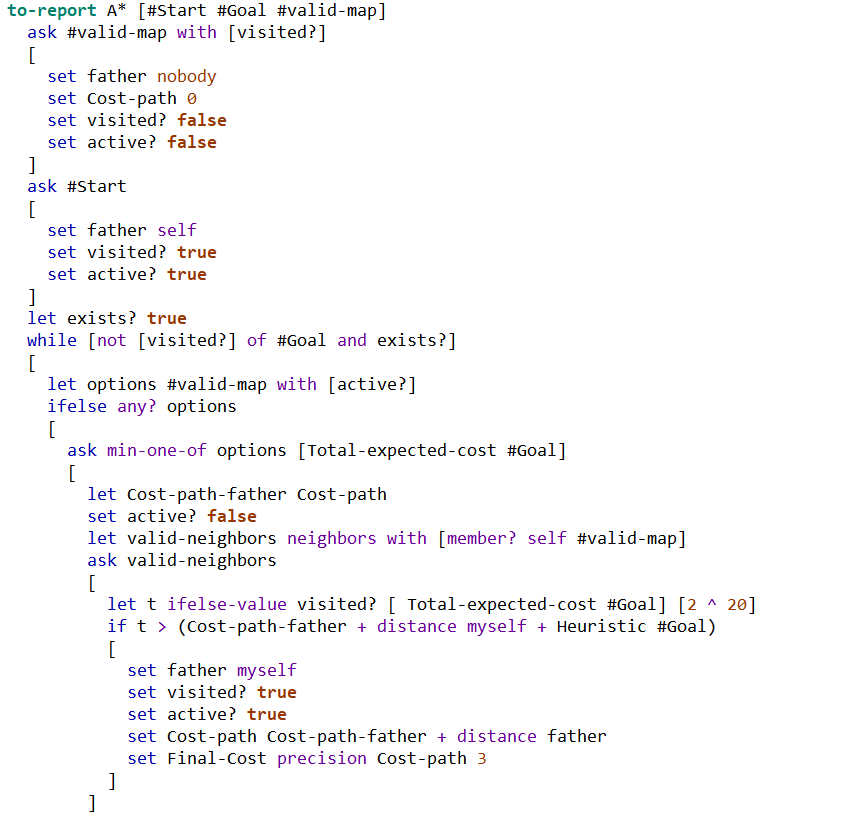


Рисунок 2.22. Функция поиска кратчайшего пути по алгоритму А\*.

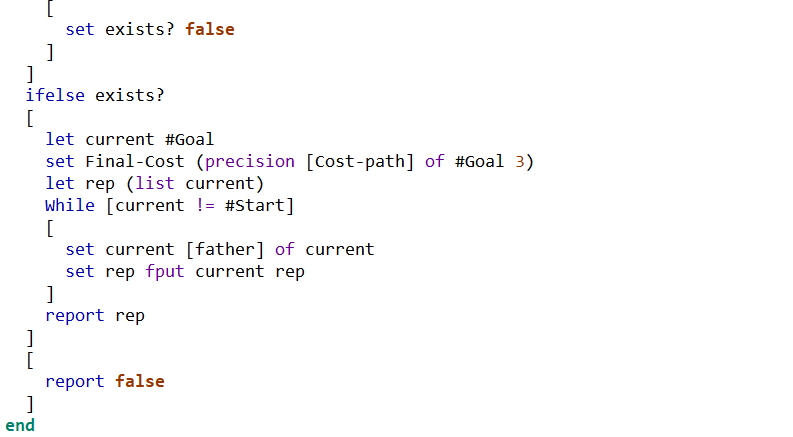


Рисунок 2.22 Функция поиска кратчайшего пути по алгоритму А\*. (Продолжение)

Таким образом, в результате работы алгоритма будет найден кратчайший путь от начального патча к целевому патчу, основываясь на вычисленных стоимостях пути и обновленных путях через каждый патч. Путь будет «вырисован» благодаря маленьким агентам с формой круга.

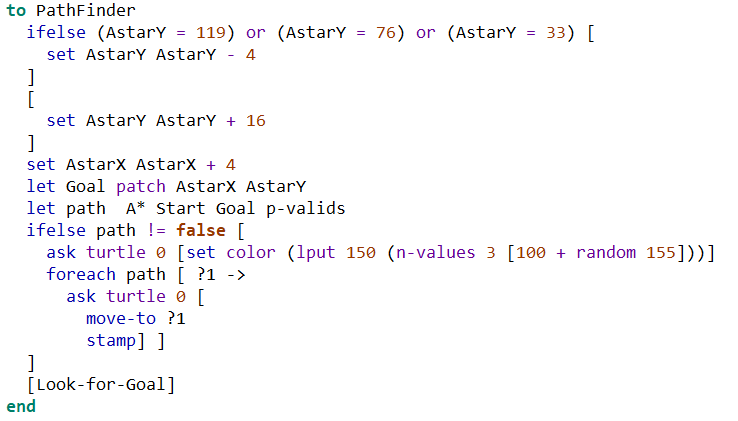


Рисунок 2.23. Вызов функции поиска кратчайшего пути по алгоритму А\*.

В моей работе алгоритм осуществляет проверку всех возможных путей к цели, перемещаясь по патчам, которые не были раскрашены в серый цвет (то есть не являются стенами). Исходно все дороги были обозначены белым цветом. Только патчи с белым цветом будут рассматриваться в алгоритме A\*.

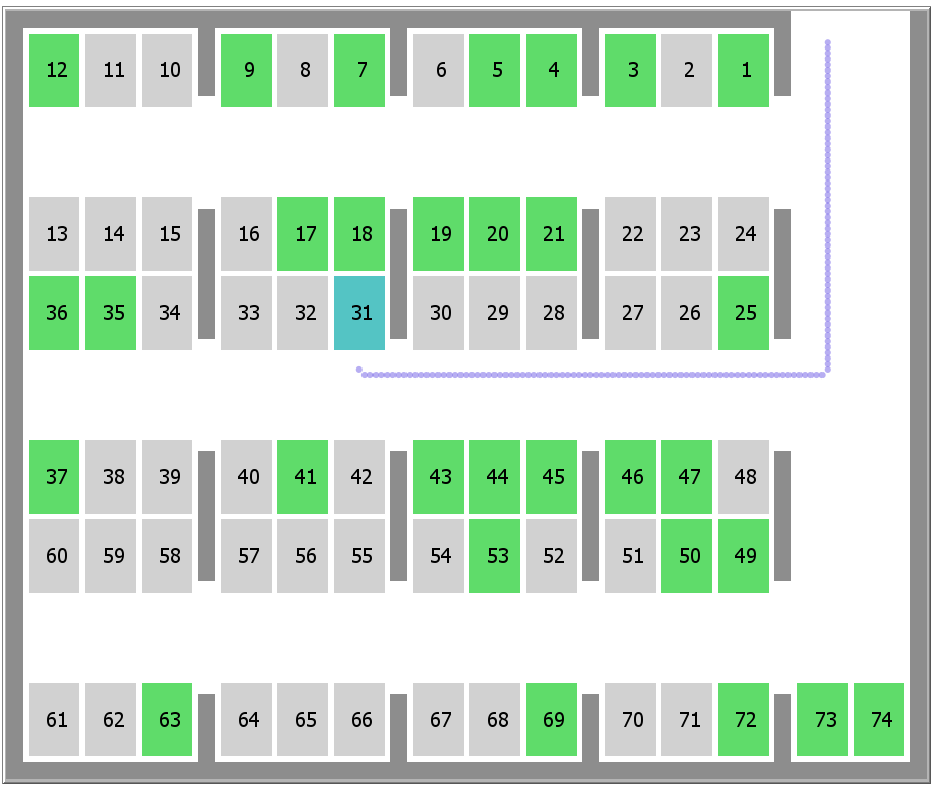


Рисунок 2.24. Результат работы поиска кратчайшего пути по алгоритму А\*.

# ГЛАВА 3. РАСЧЕТ ОБЕСПЕЧЕНИЯ УСЛОВИЙ БЕЗОПАСНОСТИ ЖИЗНИДЕЯТЕЛЬНОСТИ (LIFESAFETY)

3.1. Исходные данные

В воздухе рабочей зоны производственного помещения обнаружено вредное вещество - оксид углерода. Производственный процесс в течение одной смены разделен на 5 технологических операций, каждая из которых характеризуется различными объемами поступления вредного вещества в воздух рабочей зоны. Продолжительность каждой технологической операции составляет 65, 140, 145, 50 и 80 минут соответственно. Общая продолжительность смены составляет 480 минут или 8 часов.

Для оценки уровня вредности и опасности условий труда в данных помещениях были проведены измерения содержания вредного вещества. В каждой технологической операции было выполнено 5 измерений, где первое значение представляет измеренную концентрацию вредного вещества в мг/м³, а второе значение обозначает время выполнения измерения в минутах.

Результаты замеров:

1. (8.481-10),(15.392-36),(5.87-12),(14.254-13),(5.894-7)
2. (15.242-31),(7.707-16),(0.513-33),(10.892-11),(4.282-23)
3. (15.195-32),(9.076-5),(3.178-11),(14.751-15),(9.251-19)
4. (3.933-21),(16.054-22),(11.072-22),(6.246-21),(1.57-19)
5. (14.302-16),(9.905-13),(5.638-9),(1.499-5),(14.747-34)

3.2. Определение расчетных концентраций оксида углерода в воздухе рабочей зоны.

Фиксируем в таблице измеренные значения концентрации оксида углерода для каждой отдельной технологической операции:

Таблица 3.1. Результаты контроля содержания оксида углерода в воздухе рабочей зоны.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Операция** | | **Замер** | | **Операция** | | **Замер** | |
| **№** | **Мин** | **Мин** | **Мг/м3** | **№** | **Мин** | **Мин** | **Мг/м3** |
| 1 | 65 | 10 | 8,481 | 4 | 50 | 21 | 3,933 |
| 36 | 15,392 | 22 | 16,054 |
| 12 | 5,87 | 22 | 11,072 |
| 13 | 14,254 | 21 | 6,246 |
| 7 | 5,894 | 19 | 1,57 |
| 2 | 140 | 31 | 15,242 | 5 | 80 | 16 | 14,302 |
| 16 | 7,707 | 13 | 9,905 |
| 33 | 0,513 | 19 | 5,638 |
| 11 | 10,892 | 5 | 1,499 |
| 23 | 4,282 | 34 | 14,747 |
| 3 | 145 | 32 | 15,195 |
| 5 | 9,076 |
| 11 | 3,178 |
| 15 | 14,751 |
| 19 | 9,251 |

Для каждой операции находим среднее содержание оксида углерода в воздухе по формуле (3.1):

где:

1. – среднее содержание оксида углерода во время i-й технологической операции;
2. – концентрация оксида углерода в n-м замере, выполненном во время i-й операции;
3. – время выполнения n-го замера (время экспозиции)

Среднее содержание оксида углерода в воздухе во время выполнения первой технологической операции:

Среднее содержание оксида углерода в воздухе во время выполнения технологических операций 2-5:

Среднесменное содержание оксида углерода в воздухе определяется по формуле (3.2):

где:

1. – среднесменное содержание (концентрация) оксида углерода;
2. – средняя концентрация оксида углерода во время i-й операции
3. – продолжительность i-й операции

На основании данных о продолжительности технологических операций из таблицы 3.1 и предварительно полученных значений средних концентраций оксида углерода - , мы вычисляем среднюю концентрацию оксида углерода за смену.

Для вычисления максимальной концентрации (которая будет наблюдаться с вероятностью не менее 5%), мы начинаем с определения общего времени всех измерений.

Затем мы составляем таблицу, где все 25 значений концентрации (обозначаемых как и приведенных в таблице 3.1) упорядочены в порядке убывания. В третьем столбце таблицы 3.2 указывается время экспозиции (обозначаемое как и взятое из таблицы 3.1) для каждой концентрации.

Таблица 3.2. Упорядоченный ряд замеров концентрации оксида углерода.

| **n** |  |  |  |  |
| --- | --- | --- | --- | --- |
| *1* | *2* | *3* | *4* | *5* |
| 1 | 16,05 | 22 | 4,82 | 4,8 |
| 2 | 15,39 | 36 | 7,89 | 12,7 |
| 3 | 15,24 | 31 | 6,80 | 19,5 |
| 4 | 15,20 | 32 | 7,02 | 26,5 |
| 5 | 14,75 | 15 | 3,29 | 29,8 |
| 6 | 14,75 | 34 | 7,46 | 37,3 |
| 7 | 14,30 | 16 | 3,51 | 40,8 |
| 8 | 14,25 | 13 | 2,85 | 43,6 |
| 9 | 11,07 | 22 | 4,82 | 48,5 |
| 10 | 10,89 | 11 | 2,41 | 50,9 |
| 11 | 9,91 | 13 | 2,85 | 53,7 |
| 12 | 9,25 | 19 | 4,17 | 57,9 |
| 13 | 9,08 | 5 | 1,10 | 59,0 |
| 14 | 8,48 | 10 | 2,19 | 61,2 |
| 15 | 7,71 | 16 | 3,51 | 64,7 |
| 16 | 6,25 | 21 | 4,61 | 69,3 |
| 17 | 5,89 | 7 | 1,54 | 70,8 |
| 18 | 5,87 | 12 | 2,63 | 73,5 |
| 19 | 5,64 | 9 | 1,97 | 75,4 |
| 20 | 4,28 | 23 | 5,04 | 80,5 |
| 21 | 3,93 | 21 | 4,61 | 85,1 |
| 22 | 3,18 | 11 | 2,41 | 87,5 |

Таблица 3.2. Упорядоченный ряд замеров концентрации оксида углерода. (Продолжение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 23 | 1,57 | 19 | 4,17 | 91,7 |
| 24 | 1,50 | 5 | 1,10 | 92,8 |
| 25 | 0,51 | 33 | 7,24 | 100,0 |
|  | | 456 | 100,00 | - |

Относительную частоту (вероятность) наблюдения каждого из значений (четвертый столбец табл. 3.2) вычисляю по формуле (3.3):

где:

1. – относительная частота наблюдения n-й концентрации оксида углерода;
2. – время экспозиции n-й концентрации;
3. – суммарное время всех замеров (в нашем случае = 456 мин)

Для контроля правильности вычислений я определяею суммарную вероятность по всем строкам таблицы 3.2. Эта сумма должна быть равной 100%, что подтверждает отсутствие арифметических ошибок. В последний (пятый) столбец таблицы я заполняю накопленные вероятности (обозначаемые как ) которые вычисляются путем суммирования значений в текущей строке и всех предыдущих строках (согласно формуле 3.4).

Значение для самой низкой концентрации (в строке 25) должно равняться 100%.

Максимальная концентрация соответствует накопленной вероятности = 5%. Поскольку такое значение в табл. 3.2 отсутствует, определяется по интерполяции между значениями концентраций в строке n = 1 (Где = 4,8%) и n = 2 (Где = 12,7%). Формула линейной интерполяции имеет вид (3.5):

где:

1. – концентрация, соответствующая накопленной вероятности ;
2. – концентрации, ближайшие к сверху и снизу соответственно;
3. – накопленные вероятности, ближайшие к сверху и снизу соответственно.

Для максимальной концентрации, когда , по табл. 3.2 принимаем и ; соответственно и . Величина максимальной концентрации:

Для установления максимально допустимых интервалов времени, в течение которых необходимо контролировать (производить измерения) содержание оксида углерода в воздухе рабочей зоны, требуется рассчитать коэффициент вариации для числового ряда концентраций ().Сначала мы определяем медианное значение (обозначаемое как ), при котором накопленная вероятность составляет 50%, используя формулу (3.5). Для этого выполняется интерполяция между 9-й и 10-й строками таблицы 3.2, в которых накопленная вероятность составляет 48,5% и 50,9% соответственно.

Затем надо вычислить стандарт отклонения ряда значений от своего среднего (медианного) значения (3.6):

где:

1. – стандарт;
2. – общее количество значений в ряде

Для ряда из 25 значений, приведенного в табл. 3.1 и табл. 3.2:

Коэффициент определяется по формуле (3.7)

где:

1. – коэффициент вариации;
2. – стандарт отклонения значений;
3. – среднее (медианное) значение ряда.

Коэффициент вариации для измеренных значений концентрации оксида углерода:

3.3. Анализ условий труда и определение класса и уровня вредности

По прил. 5 определяем предельно допустимые концентрации для оксида углерода: (максимальная), (среднесменная).

Коэффициенты превышения ПДК в воздухе рабочей зоны для оксида углерода определяем на основании расчетов п. 3.2.:

Заносим в табл. 3.3 результаты расчета концентрации вредного вещества и дополняем ее нормативными показателями из прил. 5:

Таблица 3.3 Результаты расчета концентрации вредного вещества и их сравнение с ПДК.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Вещество** | **ПДК, мг/м3** | | **Наблюдаемая концентр-я, мг/м3** | | **К-т превышения** | |
| макс | сс | макс | сс | макс | сс |
| 1 | Оксид углерода | 20 | - | 16,05 | 10,12 | 0,80 | - |

Согласно приложению 5 (столбец 7), оксид углерода - это вещество, которое имеет направленный механизм действия и требует автоматического контроля его содержания в воздухе. Следовательно, его присутствие в воздухе рабочей зоны учитывается во 2-й строке таблицы "Классы условий труда в зависимости от содержания вредных веществ в воздухе рабочей зоны" (таблица 1 в "Руководстве по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда").

Определяем класс условий труда в зависимости от содержания вредного вещества в воздухе рабочей зоны (приложение 6). Коэффициент превышения максимальной концентрации соответствует классу условий труда 2 ( ); В виду отсутствия значения Nсс устанавливается «Допустимый» класс условий труда, в данном случае это класс 2. Результат приведен в табл. 3.4

Таблица 3.4. Класс условий труда в зависимости от содержания в воздухе рабочей зоны вредных веществ.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вредные вещества** | | | **Класс условий труда** | | | | | |
| Допустимый | Вредный | | | | Опасный |
| 2 | 3.1 | 3.2 | 3.3 | 3.4 | 4 |
| 1 | | | 2 | 3 | 4 | 5 | 6 | 7 |
| Особенност и действия на организм | вещества опасные для развития острого отравления | с остронаправленным механизмом действия | 0,80 --- | --- | --- | --- | --- | --- |

Оценка условий труда, учитывающая комбинированное воздействие факторов, осуществляется путем измерения всех вредных факторов, а также учета суммарных эффектов при комбинированном действии химических веществ, биологических факторов и электромагнитных излучений различных частотных диапазонов. Результаты оценки вредных факторов рабочей среды и трудового процесса представлены в таблице 3.5.

Таблица 3.5. Итоговая таблица по оценке условий труда работника по степени вредности и опасности.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Факторы** | **Класс условий труда** | | | | | | |
| Оптимальный | Допустимый | Вредный | | | | Опасный(экстремальный) |
| 1 | 2 | 3.1 | 3.2 | 3.3 | 3.4 | 4 |
| Химический | -- | + | -- | -- | -- | -- | -- |
| **Общая оценка условий труда** | -- | + | -- | -- | -- | -- | -- |

Таким образом, класс условий труда определен как допустимый (класс 2).

3.4. Организация и проведение контроля содержания вредных веществ в воздухе рабочей зоны.

Для решения вопроса о полноте контроля требуется составить список веществ, которые могут выделяться в воздух рабочей зоны в процессе технологического процесса. Для этого требуется следующая информация.:

* о вредных веществах, которые используются в технологическом процессе, включая их физическое состояние, летучесть и другие характеристики, а также соответствие этих веществ нормативно-технической документации (например, сертификаты, технические условия, ГОСТы и т. д.).
* о химических реакциях, которые происходят на всех этапах технологического процесса, включая возможность образования промежуточных и побочных продуктов, а также качественный состав продуктов деструкции, гидролиза, пиролиза и других возможных превращений.
* возможность сорбции химических веществ на частицах пыли, строительных конструкциях и оборудовании, а также возможность их последующей десорбции.

При разработке плана контроля следует учесть следующее:

* особенности технологического процесса, такие как его непрерывный или периодический характер, температурный режим, количество выделяющихся вредных веществ и другие факторы.
* физико-химические свойства контролируемых веществ, такие как их агрегатное состояние, плотность, давление пара, летучесть и другие, а также возможность превращения этих веществ в результате окисления, деструкции, гидролиза и других процессов.
* класс опасности вещества и его особенность воздействия на организм.
* планировку помещений, включая этажность здания, наличие межэтажных проемов, связь со смежными помещениями и другие факторы.
* количество и вид рабочих мест (постоянные, непостоянные, аналогичные);
* фактическую продолжительность пребывания работника на рабочем месте в течение смены.

В зависимости от класса опасности вредного вещества рекомендуется следующая периодичность контроля: веществ 1 класса опасности – не реже 1 раза в 10 дней; 2 класса – 1 раз в месяц; 3 класса – 1 раз в 3 месяца; 4 класса – 1 раз в 6 месяцев. В зависимости от стабильности концентраций (величины стандартного геометрического отклонения ): при ≤ 3 – не реже 1 раза в год, при от 3 до 6 – не реже одного раза в полугодие, при > 6 – не реже 1 раза в квартал.

Стандартное геометрическое отклонение определяется по формуле (3.8):

где:

1. - среднесменная концентрация;
2. - среднее (медианное) значение ряда концентраций.

Определяем величину стандартного геометрического отклонения для рассматриваемого вредного вещества (оксида углерода):

Периодичность контроля назначаем по наиболее жесткому требованию:

* для веществ 3 класса опасности (к которым относится оксид углерода – столбец 6 прил. 5) – не реже 1 раза в 3 месяца;
* при стабильности концентраций, характеризуемой стандартным геометрическим отклонением (≤ 3) – не реже 1 раза в год.

Окончательно принимаем периодичность контроля содержания вредных веществ в воздухе рабочей зоны равной 6 месяцев.

3.5. Мероприятия по обеспечению безопасных условий труда.

Допустимые условия труда – это такие условия, при которых уровни вредных факторов не вызывают заметных изменений в здоровье и не повышают риск заболеваний при длительном контакте.

Один из способов очистки воздуха от оксида углерода (CO) – это использование специальных устройств, таких как кондиционеры. Хорошая система вентиляции позволяет обновлять воздух в помещении и удалять оксид углерода и другие вредные вещества. Она может использоваться как в домах, так и в больших промышленных сооружениях. Хорошим решением будет настенная сплит-система DEXP AC-CH7ONF:



Рисунок 3.1. Настенная сплит-система DEXP AC-CH7ONF

# ГЛАВА 4. РАСЧЕТ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ (ECONOMICS)

4.1. Расчет затрат на разработку ИС.

В состав затрат на разработку системы (КР) входят:

* затраты на заработную плату разработчиков, ;
* отчисления в СФР, .;
* затраты на материалы (без НДС),;
* затраты на электроэнергию (без НДС), .;
* амортизационные отчисления, .;
* накладные расходы, ;
  + 1. Затраты на заработную плату разработчиков.

Для определения затрат на заработную плату разработчиков была подготовлена таблица 4.1, содержащая список работ и их предполагаемую трудоемкость, в которой:

* - минимальная трудоемкость работы при наиболее благоприятных условиях;
* - максимальная трудоемкость работы при наиболее неблагоприятных условиях ее выполнения;
* – Ожидаемая трудоемкость выполнения i-й работы, определяемой по формуле (4.1):

Таблица 4.1. Ожидаемая трудоемкость работ.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование работ |  |  |  |
| *1* | *2* | *3* | *4* |
| Сбор требований | 5 | 7 | 5,8 |
| Анализ предметной области | 3 | 7 | 4,6 |
| Проектирование макета парковочных мест | 3 | 6 | 4,2 |
| Создание агентов для ИС | 2 | 4 | 2,8 |

Таблица 4.1. Ожидаемая трудоемкость работ. (Продолжение)

|  |  |  |  |
| --- | --- | --- | --- |
| *1* | *2* | *3* | *4* |
| Программирование алгоритма А\* | 5 | 20 | 11 |
| Разработка функции бронирования мест | 5 | 15 | 9 |
| Настройка поведения агентов | 5 | 15 | 9 |
| Тестирование | 1 | 5 | 2,6 |
| Разработка документации | 1 | 5 | 2,6 |
| Итого: | | | 51,6 |

Дополнительно, для определения затрат на заработную плату разработчиков, была подготовлена таблица 4.2, содержащая информацию о составе работ, их трудоемкости и продолжительности, в которой:

* – Ожидаемая трудоемкость выполнения i-й работы
* - число исполнителей i–й работы, человек
* – продолжительность выполнения i-й работы, в рабочих днях, определяемой по формуле (4.2):

Где:

* - доля дополнительных работ, выполняемых данной группой исполнителей, данный коэффициент может принимать значения от 0 до 0,4;
* - коэффициент использования разработчиками сменного времени, определяемый как .

Таблица 4.2. Состав, трудоемкость и продолжительность работ.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование работ |  |  |  |
| Сбор требований | 5,8 | 1 | 8,53 |
| Анализ предметной области | 4,6 | 1 | 6,76 |
| Проектирование макета парковочных мест | 4,2 | 1 | 6,18 |
| Создание агентов для ИС | 2,8 | 1 | 4,12 |
| Программирование алгоритма А\* | 11 | 1 | 16,18 |
| Разработка функции бронирования мест | 12 | 1 | 13,24 |
| Настройка поведения агентов | 12 | 1 | 13,24 |
| Тестирование | 12 | 1 | 3,82 |
|  | | | 75,88 |

Затраты на заработную плату разработчиков рассчитываются по следующей формуле (4.3):

Где:

* - коэффициент, учитывающий дополнительную заработную плату. Его принимают 1,1;
* - продолжительность выполнения работ по разработки ИС в днях;
* - среднедневная ставка заработной платы. Принимаем значение

Результат расчета затрат на заработную плату разработчиков представлен в табл. 4.5.

* + 1. Страховые взносы на заработную плату разработчиков.

Отчисления в СФР от затрат на заработную плату рассчитывают с учетом ставки по состоянию на 01.01.23 г. 30%, по формуле (4.4):

Результат расчетов страховых взносов на заработную плату разработчиков представлены в табл. 4.5.

* + 1. Затраты на материалы.

Была подготовлена таблица 4.3, в которой учтены расходные материалы и технические средства, стоимость которых составляет менее 100 тысяч рублей.

Таблица 4.3. Цена материалов, покупных изделий и технических средств для разработки системы.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование материалов | Кол-во единиц | Цена, руб./ед | Сумма рублей |
| Бумага формата А4 | 120 | 12 | 1440 |
| Катриджи для принтера | 3 | 2000 | 6000 |
| Итого | | | 7440 |
| Наименование ТС, стоимостью менее 100 тыс. руб | Кол-во единиц | Цена, руб./ед | Сумма рублей |
| Ноутбук THUNDEROBOT 911ST | 1 | 78 000 | 78 000 |
| Принтер HIPER P-1120 | 1 | 13000 | 13 000 |
| Итого | | | 91 000 |
| Итого | | | 98 440 |

Для расчета затрат на расходные материалы и технические средства, стоимость которых составляет менее 100 тысяч рублей, требуется использовать следующую формулу (4.5):

Результат расчета затрат на расходные материалы и технические средства менее 100 тыс. руб. представлены в табл. 4.5.

* + 1. Затраты на электроэнергию.

Затраты на электроэнергию рассчитываются по формуле (4.6):

Где:

* - тариф на электроэнергию без НДС, руб./кВтч;
* - потребляемая используемыми на i-й работе техническими средствами мощность, кВт;
* - время выполнения i-й работы в часах. = (из табл. 4.3) \* (среднее время работы техники в сутки)

Результат расчетов затрат на электроэнергию представлен в таблице 4.4.

Таблица 4.4. Затраты на электроэнергию.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование работ | Время выполнения i-й работы, ч | Потребляемая техническими средствами мощность, кВт | кВт/ч |
| *1* | *2* | *3* | *4* |
| Сбор требований | 68,24 | 0,06 | 4,09 |
| Анализ предметной области | 54,12 | 0,06 | 3,25 |
| Проектирование макета парковочных мест | 49,41 | 0,06 | 2,96 |
| Создание агентов для ИС | 32,94 | 0,06 | 1,98 |
| Программирование алгоритма А\* | 129,41 | 0,06 | 7,76 |
| Разработка функции бронирования мест | 105,88 | 0,06 | 6,35 |

Таблица 4.4. Затраты на электроэнергию. (Продолжение)

|  |  |  |  |
| --- | --- | --- | --- |
| *1* | *2* | *3* | *4* |
| Настройка поведения агентов | 105,88 | 0,06 | 6,35 |
| Тестирование | 30,59 | 0,06 | 1,83 |
| Разработка документации | 30,59 | 0,06 | 1,83 |
|  | | | 36,42 |
|  | | | 234,20 |

* + 1. Амортизация технических средств.

Так как в проекте не применяется оборудование, стоимостью более 100 тысяч рублей, расчёт амортизационных отчислений не является обязательным.

Это связано с тем, что амортизационные отчисления применяются для учёта изнашивания и уменьшения стоимости основных средств компании, которые используются для производства товаров или услуг. В данном случае, в связи с отсутствием значительных объектов основных средств в проекте, амортизационные отчисления не являются необходимыми для финансового учёта.

* + 1. Накладные расходы по разработке.

Под термином "накладные расходы" понимаются определенные расходы, которые возникают у разработчика. Эти расходы включают следующие составляющие: коммунальные платежи, такие как платежи за водоснабжение, отопление и освещение; затраты на заработную плату административно-управленческого персонала, вспомогательного персонала и обслуживающего персонала, включая страховые взносы; контрагентские расходы, которые включают оплату работ, выполненных сторонними организациями, налоги, начисленные в соответствии с законодательством, арендные платежи, а также расходы на техническое обслуживание и ремонт технических средств.

Данные расходы вычисляются по формуле ():

Результат вычисления накладных расчетов представлены в табл. 4.5.

Таблица 4.5. Затраты на разработку ИС.

|  |  |  |
| --- | --- | --- |
| Статьи затрат | Условные обозначения | Сумма, руб. |
| Затраты на заработную плату |  | 208676,47 |
| Отчисления во внебюджетные фонды |  | 62602,94 |
| Затраты на материалы |  | 98440 |
| Затраты на электроэнергию |  | 234,20 |
| Накладные расходы |  | 184976,81 |
| **Итого затрат на разработку** |  | **554930,42** |

* 1. Расчёт затрат на внедрение ИС

В состав затрат на внедрение следует включать:

* затраты на приобретение дополнительного оборудования для внедрения ИС, стоимостью более 100 тысяч рублей;
* затраты на подготовку персонала для работы с ИС, руб.
  + 1. Затраты на приобретение технических средств, стоимостью более 100 тысяч рублей.

При создании информационной системы удалось обойтись затратами на технические средства, не превышающими 100 тысяч рублей. Следовательно руб.

* + 1. Затраты на подготовку персонала.

Затраты на подготовку персонала рассчитываются по формуле ()

Где Цподг – цена подготовки одного специалиста. Цподг = 15 000 руб.;

Rспец – количество подготавливаемых специалистов. Rспец = 1 чел.;

Затраты на внедрение информационной системы расчитываются по формуле ():

Результаты расчета затрат на внедрение информационной системы представлены в таблице 4.6.

Таблица 4.6. Затраты на внедрение информационной системы.

|  |  |  |
| --- | --- | --- |
| Статьи затрат | Условные обозначения | Сумма, руб. |
| Затраты на ТС стоимостью более 100 тыс руб. |  | 0 |
| Затраты на подготовку персонала |  | 15 000 |
| **Затраты на внедрение** |  | **15 000** |

* 1. Анализ спроса и прогнозирование дохода и расходов разработчика.
     1. Анализ спроса на информационную систему.

Разработанная система предназначена для прогнозирования текущих и возможных событий на предприятии, связанных с парковочными услугами. Она обладает высоким потенциалом и будет востребована компаниями, специализирующимися на парковочном сервисе. Система на базе агентного моделирования предлагает комплексный подход, моделируя поведение различных участников парковочного процесса. Она поможет компаниям оценивать текущую ситуацию, прогнозировать спрос, планировать ресурсы и повышать эффективность операций. Также система способна предсказывать возможные проблемы и принимать проактивные меры для их предотвращения.

На начальном этапе не предполагается высокий спрос на нашу систему, так как она представляет собой инновационное решение для отображения парковочного сервиса с применением агентного моделирования. Предполагается начать продажи на российском рынке при стоимости разработанного продукта в 150000 и объеме продаж 12 шт./год.

Внедрение данной системы потребует переработки многих аспектов, на которых базируются существующие компании, предоставляющие парковочные услуги. Однако, благодаря ее высокой эффективности, я уверен, что со временем она сможет завоевать доверие и заинтересовать компании

* + 1. Прогноз дохода разработчика от продажи прав на информационную систему.

Для расчета возможного дохода разработчика от продажи ИС на внешнем рынке используется формула (4.10):

Где:

* - цена единицы ИС, руб./шт.;
* - объем продаж ИС за год, шт.

В результате возможный доход разработчика от продажи ИС на внешнем рынке составит:

* + 1. Прогноз расходов разработчика от продажи прав на информационную систему.

Расчет доходов и расходов указаны в таблице 4.7.

Таблица 4.7. Годовые расходы разработчика.

|  |  |
| --- | --- |
| **, руб./год, в т.ч.** | **1 213 493,04** |
| зарплата продавца (55000\*12) | 660 000 |
| Страховые взносы с ЗП продавца | 198 000 |
| Расходы на интернет и прочие услуги связи | 50 000 |
| Банковские расходы | 10 000 |
| Аренда помещения (20000\*12) | 240 000 |
| Амортизационные отчисления, | 55 493,04 |

* 1. Расчет экономической эффективности проекта.
     1. Финансовый план (расчет эффективности инвестиций в разработку и внедрение системы).

Для составления финансового плана на двухлетний период (t = 0,1,2) были рассчитаны следующие параметры:

* Балансовая прибыль в t-м году:
* Налог на прибыль в t-м году:

Где:

* + - ставка налога на прибыль,
* Читая прибыль в t-м году:

Результаты расчетов представлены в таблице 4.8.

Таблица 4.8. Прогноз доходов и расходов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| t |  |  |  |  |  |
| 0 | 01 | 0 | 0 | 0 | 0 |
| 1 | 1 800 000,00 | 1 213 493,04 | 586 506,96 | 117 301,39 | 469 205,57 |
| 2 | 1 800 000,00 | 1 213 493,04 | 586 506,96 | 117 301,39 | 469 205,57 |

* + 1. Расчет ЧДД проекта.

Один из основных показателей для оценки эффективности реализации проекта или инвестиций является чистый дисконтированный доход (ЧДД). Для достижения сопоставимости экономических показателей, полученных в разные годы, с моментом начала реализации проекта, используется процесс дисконтирования. При этом показатели умножаются на коэффициент дисконтирования.

Коэффициент дисконтирования рассчитывается с использованием следующей формулы: (4.14):

Где:

* - коэффициент дисконтирования;
* - расчетный период (в годах), ;
* E представляет собой норму дисконта, которая может быть постоянной для всех этапов расчета или переменной. Норма дисконта является основным экономическим показателем, используемым для оценки эффективности бакалаврской работы. В данном случае, норма дисконта составляет ключевую ставку Центрального Банка Российской Федерации + 3 процентных пункта, и равна 0,11. Начиная с 19 февраля 2022 года, ключевая ставка составляет 7,5%.

Чистый дисконтированный доход (ЧДД) представляет собой сумму текущих эффектов за весь расчетный период, приведенную к начальному шагу, или разницу между интегральными результатами и интегральными затратами. Если используется постоянная норма дисконта, ЧДД рассчитывается с помощью следующей формулы (4.15):

При этом величины и принимаются постоянными:

Результаты расчетов представлены в таблице 4.9.

Таблица 4.9. Расчет ЧДД проекта.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование | Условные обозначения | Годы расчетного периода | | |
| 0 | 1 | 2 |
| *1* | *2* | *3* | *4* | *5* | *6* |
| 1 | Капиталовложения в проект |  | 554 930,42 | 0 | 0 |

Таблица 4.9. Расчет ЧДД проекта. (Продолжение)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *1* | *2* | *3* | | *4* | | *5* | *6* |
| 2 | Чистая прибыль |  | | 0 | | 469 205,57 | 469 205,57 |
| 3 | Амортизационные вычисления |  | | 0 | | 0 | 0 |
| 4 | Итог от операционной деятельности |  | | 0 | | 469 205,57 | 469 205,57 |
| 5 | Итог от инвестиционной и операционной деятельности |  | | -554 930,42 | | 469 205,57 | 469 205,57 |
| 6 | Чистый доход, ЧД |  | | -554 930,42 | | -85 724,86 | 383 480,71 |
| 7 | Коэффициент дисконтирования |  | | 1 | | 0,901 | 0,812 |
| 8 | Дисконтированные капитальные вложения |  | | 554 930,42 | | 0 | 0 |
| 9 | Дисконтированная прибыль |  | | 0 | | 422 707,72 | 380 817,76 |
| 10 | Дисконтированный итог |  | | -554 930,42 | | 422 707,72 | 380 817,76 |
| 11 | Чистый дисконтированный доход, ЧДД | |  | | -554 930,42 | -132 222,71 | 248 595,06 |

Исходя из расчетов:

* руб.
* Простой срок окупаемости – 2 года
* Дисконтированный срок окупаемости – 2 года
* Индекс доходности дисконтированных инвестиций (ИДД) вычисляется по формуле (4.18):

И его значение

Так как и – проект признается эффективным.

# ЗАКЛЮЧЕНИЕ (CONCLUSIONS)

В рамках выпускной квалификационной работы была достигнута поставленная цель путем выполнения следующих задач:

1. Разработана модель парковки в Netlogo с помощью методов жадного алгоритма поиска кратчайшего пути А\* и роевого алгоритма
2. Определение диапазона часов для машин организаций рядом с парковкой
3. Реализовано отображение занятых и свободных парковочных мест в режиме реального времени.
4. Реализовано бронирование парковочного места
5. Реализована настройка системы счетчиков для определения въезда – выезда машины.

В результате этой работы я создал модель парковочного сервиса в среде агентного моделирования Netlogo. Эта модель является полноценным отражением всех аспектов парковочного сервиса. Было уделено особое внимание деталям и функционалу, чтобы обеспечить максимально реалистичное представление процесса парковки и организации сервиса.

Интерфейс, который мы разработали для этой модели, представляет собой пример интерфейса мобильного приложения. Мы стремились сделать его интуитивно понятным и удобным для пользователей, чтобы они могли легко освоить приложение и взаимодействовать с парковочным сервисом.

В целом, созданная модель и ее интерфейс являются не только репрезентацией парковочного сервиса, но и полезным инструментом для анализа и оптимизации работы парковочных систем.

# СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ (REFERENCES)

1. Ивашкин Ю. А. Мультиагентное имитационное моделирование больших систем: учеб. пособие / Ю. А. Ивашкин. - М.: МГУПБ, 2008. - 230 с.
2. Мезенцев К.Н. Мультиагентное моделирование в среде NetLogo: учеб. пособие /К.Н. Мезенцев. - М.: МАДИ, 2014. - 169 с.
3. Мезенцев К.Н. Моделирование в примерах и задачах в среде AnyLogic / К.Н. Мезенцев. - Lap LAMBERT Academic Publishing, 2013 - 205 с.
4. Мезенцев, К. Н. Системный подход и Имитационное моделирование / К. Н. Мезенцев // Информационные технологии в образовании, науке, технике и гуманитарной сфере : Межвузовский сборник трудов. Том Выпуск № 5. – Москва : Московский гуманитарный университет, 2014. – С. 191-195. – EDN VIVLXF.
5. Prihozhy, А. А. All pairs shortest paths search in large graphs / А. А. Prihozhy, O. N. Karasik // Big Data and Advanced Analytics. – 2021. – No. 7-1. – P. 39-49. – EDN QHLNUY.
6. Изотова, Т. Ю. Обзор алгоритмов поиска кратчайшего пути в графе / Т. Ю. Изотова // Новые информационные технологии в автоматизированных системах. – 2016. – № 19. – С. 341-344. – EDN VTZOVJ.
7. Рассел С., Норвиг П. Искусственный интеллект. Современный подход. Вильямс. 2006 (2007) - 408 с.
8. Карпенко, А. П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой / А. П. Карпенко. – Москва : Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), 2014. – 448 с. – ISBN 978-5-7038-3949-2. – EDN VCPWAJ.
9. Коцюбинская, С. А. Алгоритмы роевого интеллекта. Алгоритм роя частиц / С. А. Коцюбинская // Modern Science. – 2020. – № 3-2. – С. 267-271. – EDN JWARUO.