МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСИС»**

Институт компьютерных наук

Кафедра инженерной кибернетики

Направление подготовки: «09.04.03 Прикладная информатика»

Квалификация: **магистр**

Группа: **МПИ-22-1**

**ОТЧЕТ**

**по домашнему заданию**

на тему:

**« *Интеллектуальное имитационное агентное моделирование мультисервисной зоны общественного питания с потоками клиентов высокой интенсивности (на примере фуд-кортов)*»**

учебная дисциплина

«ИНТЕЛЛЕКТУАЛЬНЫЕ МУЛЬТИАГЕНТНЫЕ СИСТЕМЫ»

**Выполнил:**

**Студент: Аскаров Б.Р.**

**Проверил:** доц., к.т.н. А.С. Кожаринов

**Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва 2023**

**СОДЕРЖАНИЕ**

[1. ПОСТАНОВКА ЗАДАЧИ 3](#_Toc154612491)

[2. ОПИСАНИЕ ИМИТАЦИОННОЙ МОДЕЛИ 6](#_Toc154612492)

[3. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ 16](#_Toc154612493)

[4. ВЫВОД 18](#_Toc154612494)

# ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать агентную имитационную модель предметной области S с использованием системы имитационного моделирования AnyLogic©. Модель должна включать взаимодействие участников предметной области в соответствии с заданными параметрами. Основные процессы функционирования модели должны быть реализованы с использованием дискретно-событийного моделирования систем массового обслуживания.

Требования к модели:

1) Модель должна содержать трехмерную анимацию, созданную с использованием стандартных средств анимации AnyLogic©. Анимация должна отражать моделируемые процессы и особенности поведения и взаимодействия агентов.

2) В модели должно быть реализовано не менее 4 классов агентов с собственными графами состояний. Для посетителей (клиентов) должно быть создано не менее 3 классов агентов, а для персонала РД - не менее 1 класса агентов. Каждый класс агентов должен содержать не менее 5 простых состояний в диаграмме состояний.

3) Для каждого класса агентов должно быть реализовано не менее одного способа коммуникации (взаимодействия) с другими классами агентов.

4) Используя разработанную агентную имитационную модель, необходимо определить значения общей интенсивности λ входного потока посетителей в систему S и интенсивностей обслуживания µi (i=1,…, M) для каждого устройства в системе S. Значения должны быть определены таким образом, чтобы в установившемся (стационарном) режиме работы системы S среднее время пребывания посетителей в очередях достигло заданной величины среднего времени ожидания (режим "насыщения" объекта).

Общие параметры модели:

- Период моделирования: с 7:00 до 24:00.

- Единица модельного времени: минута.

- Минимальное время обслуживания одного посетителя в любом многоканальном ПБП: 5 минут.

- Минимальное время, затрачиваемое простым посетителем на прием пищи: 10 минут.

- Доля посетителей, покидающих РД сразу после покупки без приема пищи: 20%.

- На начало моделирования в системе S нет заявок.

Для данной задачи имеется следующая конфигурация обслуживания заявок:

* Общее количество устройств обслуживания: M = 5.

Количество каналов в устройствах ПОЗ:

* + Для устройств ПОЗ1 и ПОЗ2: n1 = n2 = 2.
  + Для устройств ПОЗ3 и ПОЗ4: n3 = n4 = 3.
  + Для устройства ПОЗ5: n5 = 4.

При этой конфигурации обслуживания, среднее время ожидания посетителей в очередях составляет Tсредн\_ож = 6.00 минут.

Исходные данные представлены в таблице 1.

Таблица 1. Исходные данные индивидуального задания, Вариант №8.

|  |  |  |
| --- | --- | --- |
| Количество пунктов об- служивания заявок, 𝑴 | Количество каналов в устройства ПОЗ, 𝒏𝒊 | Среднее время ожидания,  𝑻̅ож, [мин] |
| 5 | 𝐧𝟏 = 𝒏𝟐 = 2  𝒏𝟑 = 𝒏𝟒 = 𝟑  𝒏5 = 4 | 6,00 |

Также разработанная ИМ в процессе своего функционирования должна определять множество других специальных статистик для данной системы, представленных в таблице 2.

Таблица 2. Определяемые характеристики и показатели эффективности СМО.

|  |  |  |
| --- | --- | --- |
| **Название характеристики (показателя) СМО** | **Способ отображения** | |
| **Число** | **График (диаграмма)** |
| Интенсивность входного потока | + | - |
| Интенсивность выходного потока обслуженных заявок | + | - |
| Интенсивность выходного потока необслуженных заявок | + | - |
| Количество заявок, вошедших в систему | + | - |
| Общее число обслуженных заявок | + | - |
| Процент обслуженных заявок | + | + |
| Общее число заявок, покинувших систему необслуженными | + | - |
| Процент заявок, покинувших систему необслуженными | + | + |
| Общее число переходов из одного ПОЗ в другой | + | - |
| Среднее число заявок в системе в целом | + | + |
| Среднее число заявок, находящихся на обслуживании | + | + |
| Среднее время пребывания заявки в системе | + | + |

# ОПИСАНИЕ ИМИТАЦИОННОЙ МОДЕЛИ

В процессе разработки было принято решение сделать модель максимально информативной и понятной для визуального восприятия. Для этого было использовано изображение в качестве основы для создания ресторанного дворика. На этом изображении представлены все необходимые области, которые были использованы для построения модели (см. рисунок 1).

Вместо использования абстрактных элементов или символов, использование изображения помогает создать более наглядную модель, которая легко узнаваема и понятна для зрителя.



Рисунок 1 – Изображение, являющееся основой для ресторанного дворика

В процессе разработки модели был использован подход поэтапного построения. Первым этапом было наполнение карты различными объектами из палитры инструментов AnyLogic©, которые затем играли ключевую роль в модели. На этом этапе были определены и размещены следующие элементы:

1. Линия входа и выхода из системы - определена область, через которую агенты входят и выходят из модели.
2. Область группировки агентов - создана область, где агенты могут сгруппироваться перед входом в систему или после выхода из нее.
3. Сервисные места, очереди и области ожидания - определены места, где агенты могут получить обслуживание, а также очереди и области ожидания, где агенты могут ожидать своей очереди.
4. Области столов для приема пищи - созданы области, где агенты могут занять столы и употребить пищу.
5. Слой карты плотности - добавлен слой на карте, который позволяет анализировать загруженность отдельных узлов системы.

Таким образом, на первом этапе моделирования были определены и размещены указанные элементы, которые являются основой для дальнейшего развития модели и выполнения поставленных задач (рис. 2)

На втором этапе разработки произошёл анализ созданной карты ресторанного дворика, а также были созданы различные типы агентов, такие как "Дети", "Взрослые", "Пенсионеры" и "Уборщицы". Каждый из этих агентов имеет уникальный внешний вид, что значительно расширяет возможности для индивидуальной настройки каждого класса.

Для установки общих параметров и определения логики для всех агентов был создан обобщающий агент "Посетитель". Этот класс агента выступает в роли "родителя" для всех ранее созданных агентов посетителей. В этом классе определены общие свойства и методы, а также содержится диаграмма состояний, описывающая поведение посетителя в различных ситуациях (см. рисунок 3).

Использование обобщающего агента "Посетитель" позволяет установить общие характеристики и поведение для всех агентов посетителей в модели. Все созданные типы агентов наследуют эти общие свойства и методы, но также могут иметь свои собственные уникальные особенности и поведение. Это обеспечивает гибкость и настраиваемость модели для различных категорий посетителей.



Рисунок 2 – Ресторанный дворик, наполненный объектами из палитры Anylogic©



Рисунок 3 – Параметры и диаграмма состояний для класса “Посетитель”

Для агента "Посетитель" были созданы следующие параметры:

- isLeader: устанавливается в значение true (истина), когда посетитель является лидером группы, с которой он пришёл.

- timeInQueue: количество минут, проведенных агентом в очереди.

- change\_tries: количество оставшихся попыток, которые агент может сменить очередь, в которой он находится.

- queue\_patience: максимальное количество минут, которое клиент может выдержать в очереди, прежде чем попытаться выбрать другую очередь.

Кроме того, агент "Посетитель" следует диаграмме состояний, описанной ниже:

- state: состояние по умолчанию, где выполняются первоначальные действия над клиентом. Затем состояние разветвляется, и агенты следуют определенному алгоритму:

- В состояние "followLeader", если агент не является лидером группы. Это конечное состояние.

- В состояние "chooseRest", если агент является лидером группы, где происходит выбор точки быстрого питания.

- chooseRest: состояние, в котором клиент выбирает точку быстрого питания.

- inQueue: состояние, в котором клиент находится в очереди и циклически переходит в самого себя каждую минуту для подсчета времени, проведенного в очереди. Это состояние может перейти в:

- Развилку, определяющую возможность смены очереди:

- Если у клиента остались попытки, он возвращается к состоянию выбора очереди.

- Если у клиента нет попыток, он переходит в состояние "терпение закончилось" (endOfPatience), где клиент покидает очередь и уходит из ресторанного дворика с помощью состояния "leave".

- Состояние "onService", где клиент находится на этапе обслуживания.

- Состояние "serviced", где клиент обслужен и выбирает, поесть ли в зале или взять еду с собой:

- Состояние "leave", где клиент решает взять покупку с собой и покидает здание.

- Состояние "eating", где клиент решает поесть еду в зале и затем уходит (переход в состояние "leave").

Агент "Уборщица" следует диаграмме состояний, описанной ниже:

- state: состояние по умолчанию, где выполняются первоначальные действия над агентом.

Агенты следуют определенному алгоритму:

- В состояние "Wait", при начале рабочей смены.

- В состояние "GoToTable", если появляется стол, свободный от посетителей.

- Cleaning: состояние, в котором уборщица протирает свободный стол.

- GoingToWaitArea: состояние, в котором уборщица возвращается в зону ожидания, после мытья тряпки.

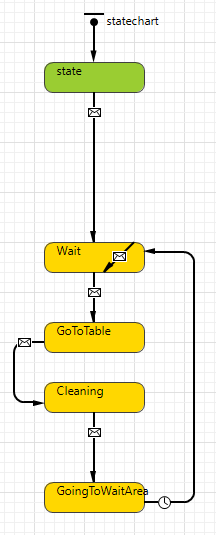


Рисунок 4 – Диаграмма состояний для класса “Уборщица”

На следующем этапе разработки модели была создана логика работы, используя блоки из палитры "Пешеходная библиотека" в среде моделирования AnyLogic.

Общая структура имитационной модели в AnyLogic выглядит следующим образом (см. рисунок 5, 6).

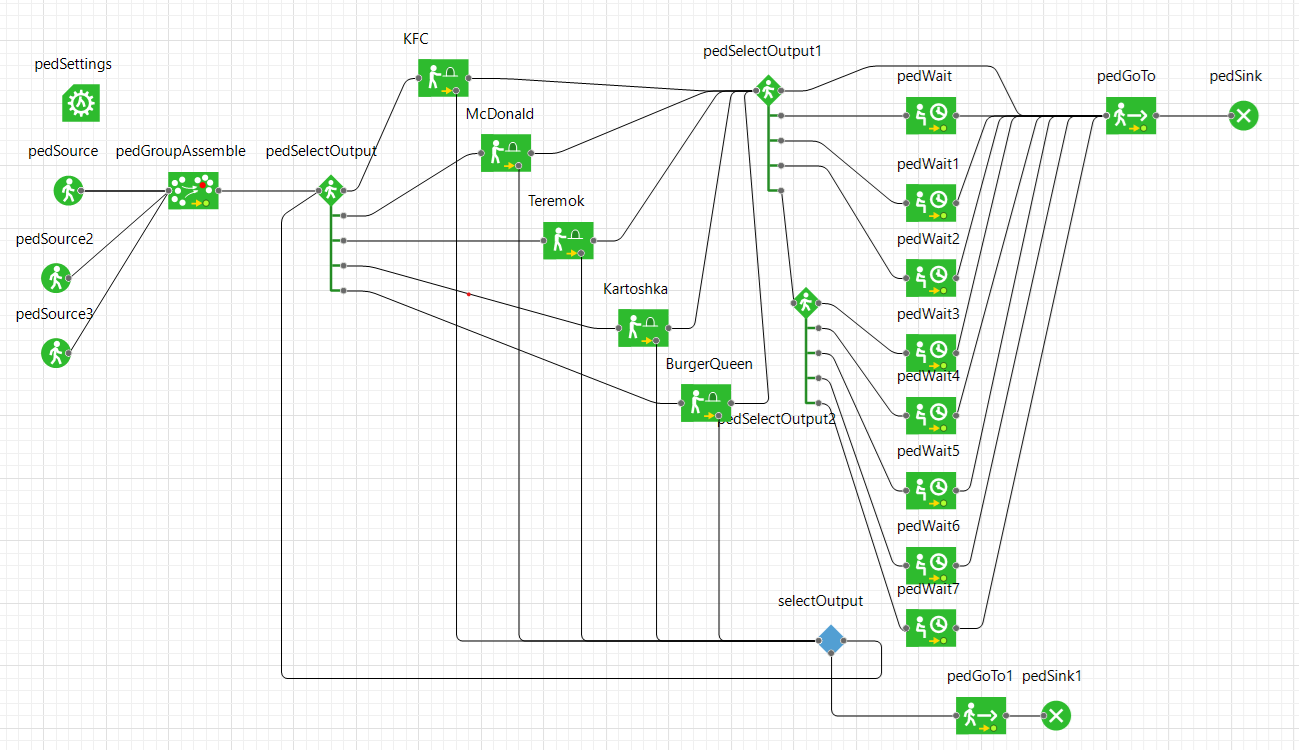


Рисунок 4 – Структура разработанной модели поведения посетителей

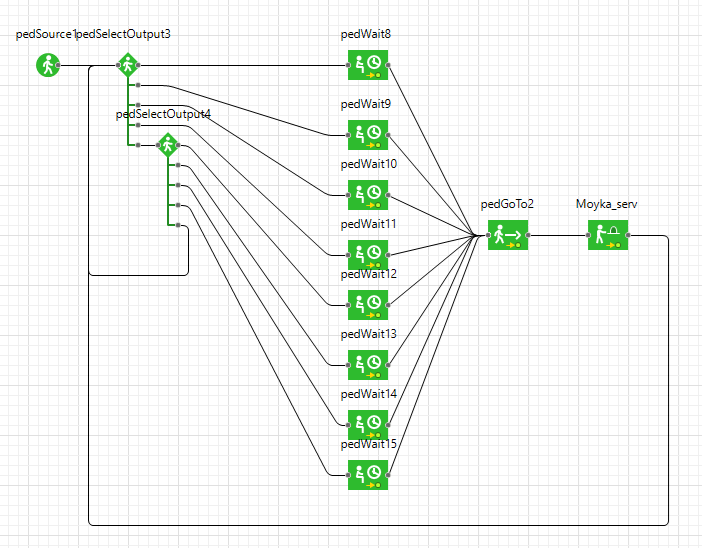


Рисунок 5 – Структура разработанной модели поведения уборщиц

Чтобы наблюдать за происходящим со стороны, создан отдельный 3D, 2D виды и вид статистики. (рис 6, 7, 8). Переключение между видами осуществляется с помощью меню, расположенного в верхней части окна отображения.



Рисунок 6 – 3D-вид разработанной модели.

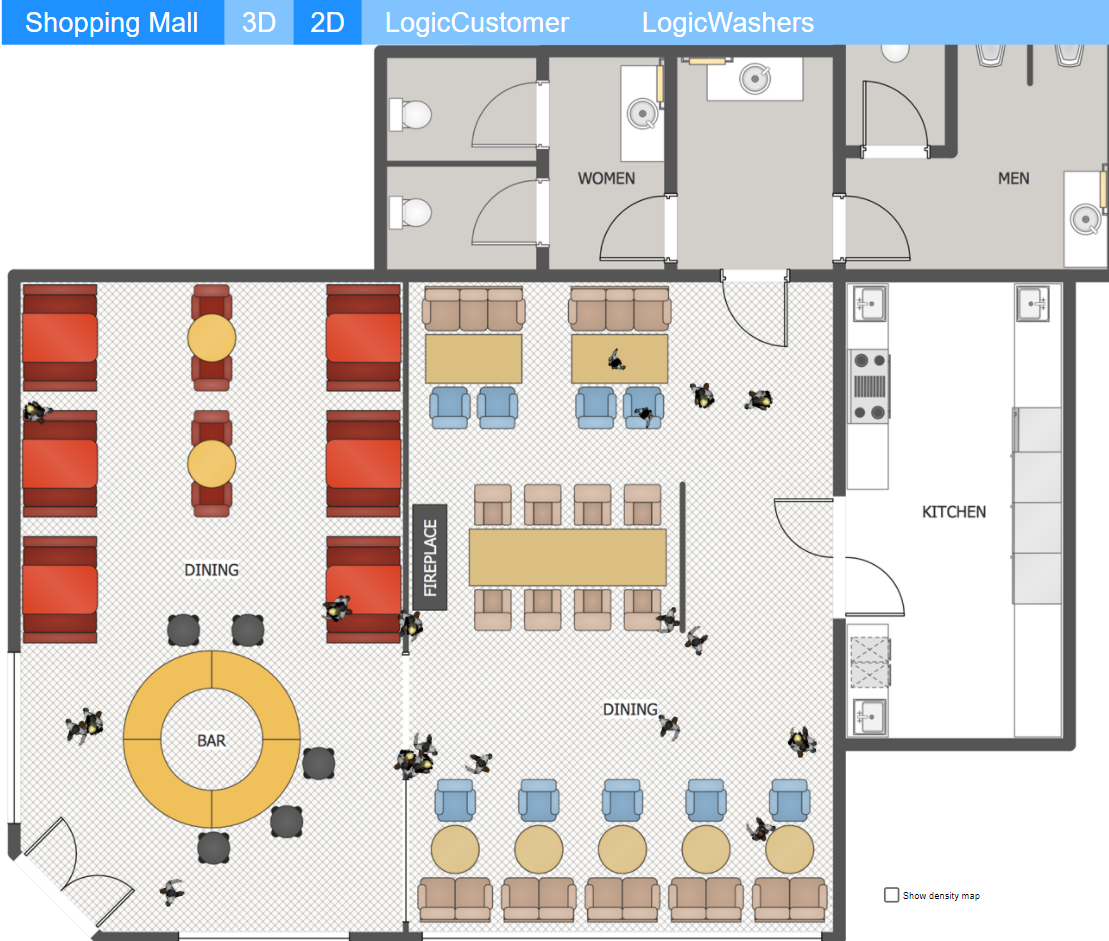


Рисунок 7 – 2D-вид разработанной модели.

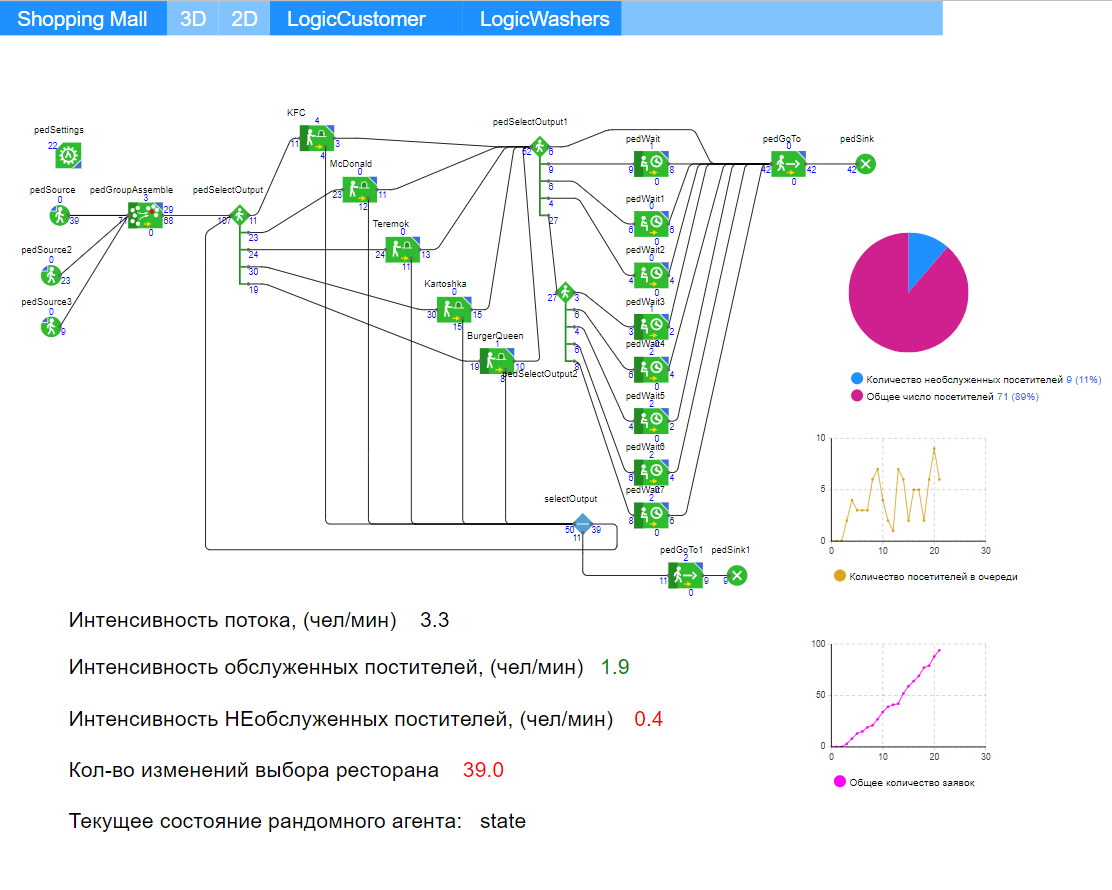


Рисунок 8 – Вид “Статистика”.

# РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Для мониторинга результатов моделирования была создана статистика, которая отслеживает заданные в начале задачи критерии. Для этого был создан отдельный вид, который можно открыть, нажав соответствующую кнопку. Статистика собирается и отображается в режиме реального времени во время работы модели. Рассмотрим показатели, полученные после продолжительного времени работы модели (см. рисунок 9).

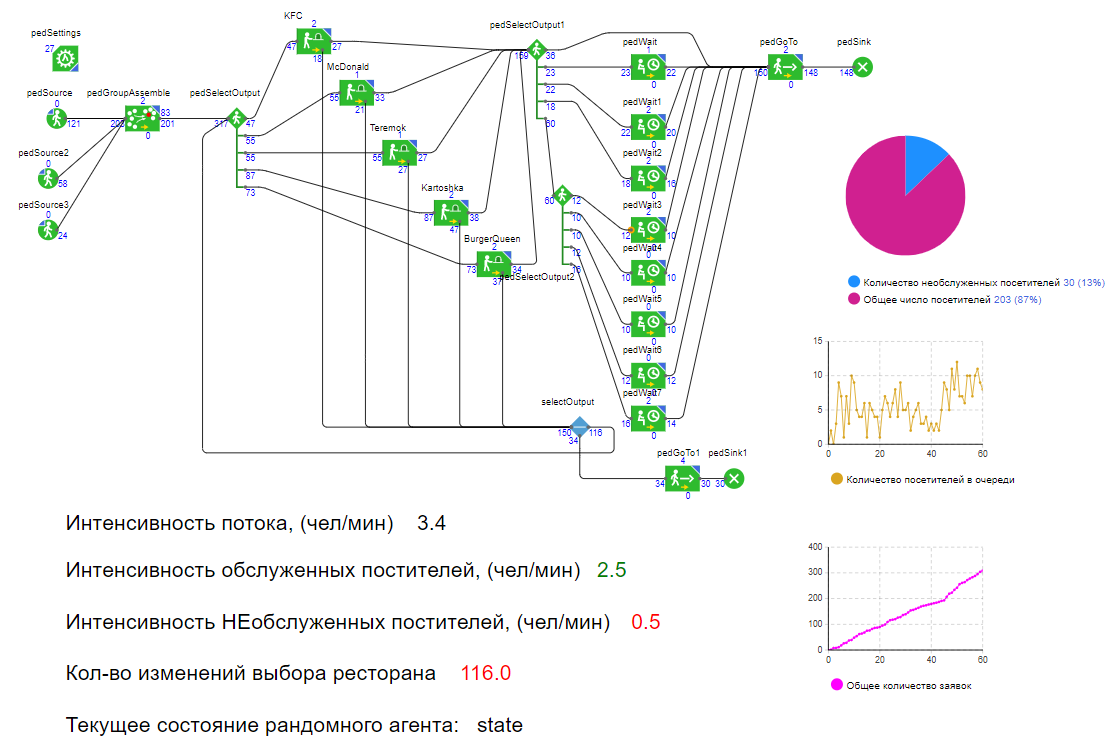


Рисунок 9 – Статистика модели, собранная за длительное время.

Основные данные представлены в виде текста с пояснениями и указанием единицы измерения, а некоторые данные отображаются в виде гистограмм. Для оценки загруженности каждой отдельной зоны объекта была добавлена функция включения тепловой карты (карты плотности), которая позволяет контролировать и выявлять зоны, требующие особого внимания или дальнейшего рассмотрения (см. рисунок 10).

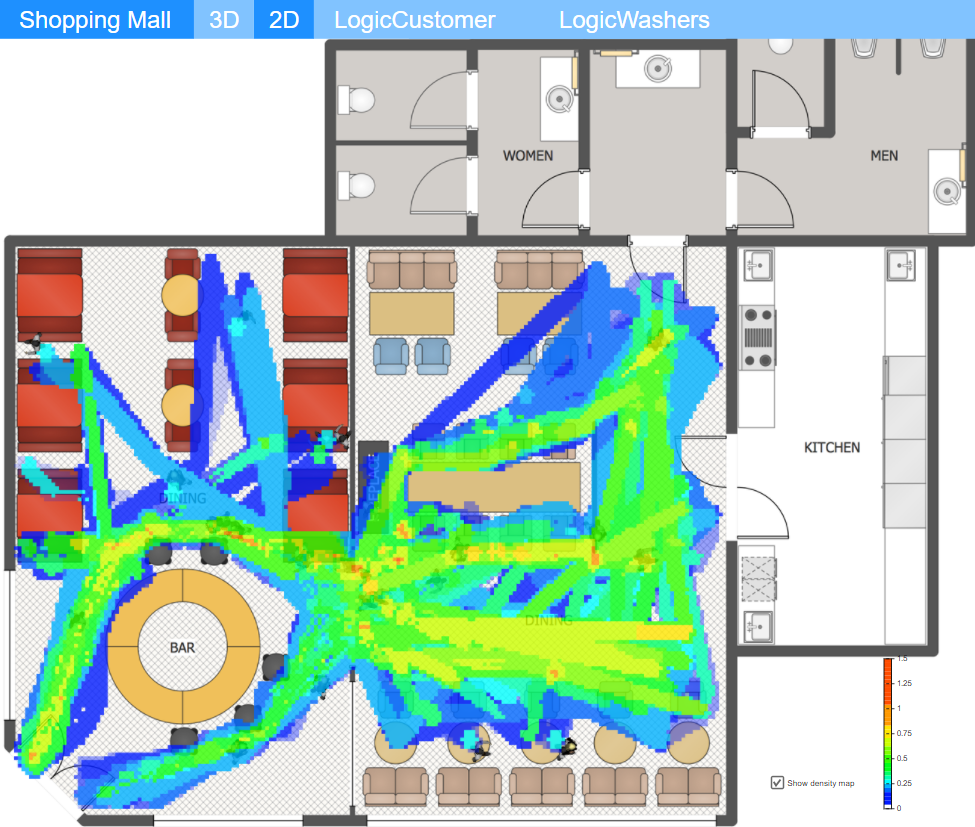


Рисунок 10– Тепловая карта спустя длительное время работы модели.

# ВЫВОД

В ходе данного задания была разработана имитационная модель ресторанного дворика. Использовалось специально созданное изображение, которое включает все необходимые элементы: входы, выходы, области обслуживания и зоны питания. Для агентов-посетителей была разработана диаграмма состояний, обеспечивающая правильную работу модели с агентами на протяжении всего процесса.

Анализ собранной статистики позволяет сделать следующий вывод о работе модели. Она демонстрирует стабильную работу без больших скоплений людей при установленной загрузке. Иногда возникают "часы пик", но они быстро устраняются.

В качестве решений для предотвращения скоплений были предложены следующие подходы:

1. Расстановка столов и посадочных мест: Оптимизация расстановки столов и стульев с целью создания достаточного пространства между ними, что обеспечит комфорт и свободу движения посетителям, и снизит скопления.
2. Временные интервалы приема пищи: Разделение времени на несколько интервалов (например, завтрак, обед и ужин), чтобы равномерно распределить поток посетителей в течение дня и снизить скопления в пиковые часы.
3. Разнообразие предложения: Предоставление разнообразных кулинарных вариантов на фуд-корте, чтобы привлечь больше посетителей и распределить их по различным местам, снижая скопления в конкретных точках.
4. Оптимизация процесса обслуживания: Использование самообслуживания или автоматизированных систем заказа и оплаты, что позволит сократить время обслуживания каждого посетителя, увеличить пропускную способность и снизить скопления.
5. Мониторинг и управление потоком посетителей: Реализация системы мониторинга и управления потоком посетителей на фуд-корте, включая счетчики посетителей, технологии определения очередей и уведомления для посетителей о доступных местах.

Применение данных решений поможет создать более комфортные условия для посетителей фуд-корта, распределить поток посетителей равномерно и снизить скопления в определенные периоды времени. Это также способствует более эффективной работе фуд-корта и повышению удовлетворенности посетителей.