Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Кубанский государственный университет»

Кафедра информационных технологий

**ОТЧЕТ**

о выполнении лабораторной работы №8

по дисциплине «Технологии проектирования программного обеспечения»

Выполнил: ст. гр. 4ИТ

Хижний Е.Г.

Проверил: доц. каф. ИТ

Полетайкин А.Н.

Краснодар

2021

**Тема**: Документирование и развертывание ИС

**Цель**: Освоение методики документирования ПС.

**Тема проекта**: Разработка имитационной модели ОДД на участке УДС крупного города.

### Задание.

1. Собрать документированные материалы, разработанные при выполнении лабораторных работ №1–7 в единый документ с разбивкой на разделы.
2. Описать назначение, технические характеристики, принцип работы и меры безопасности при эксплуатации ПС.
3. Составить руководство пользователя.

**Анализ предметной области**

Описание объекта:Одной из основных задач департамента транспорта и дорожного хозяйства администрации муниципального (далее – дептранс) наряду с предоставлением транспортных услуг населению по организации транспортного обслуживания в границах муниципального образования город Краснодар является осуществление дорожной деятельности и деятельности по организации дорожного движения на территории муниципального образования город Краснодар в отношении автомобильных дорог общего пользования местного значения.

Описание процесса:Дептранс имеет потребность в грамотной организации дорожного движения. Приложение может моделировать организацию дорожного движения на участке улично-дорожной сети города Краснодара. Моделирование может происходить при различных погодных условиях, при различных дорожных ситуациях: ремонт, авария и т. д. Данное приложение может помочь различным службам в понимании того, как те или иные события будут влиять на дорожную ситуацию.

Для представления бизнес-процесса используется модель «Черный ящик». Он выглядит следующим образом:

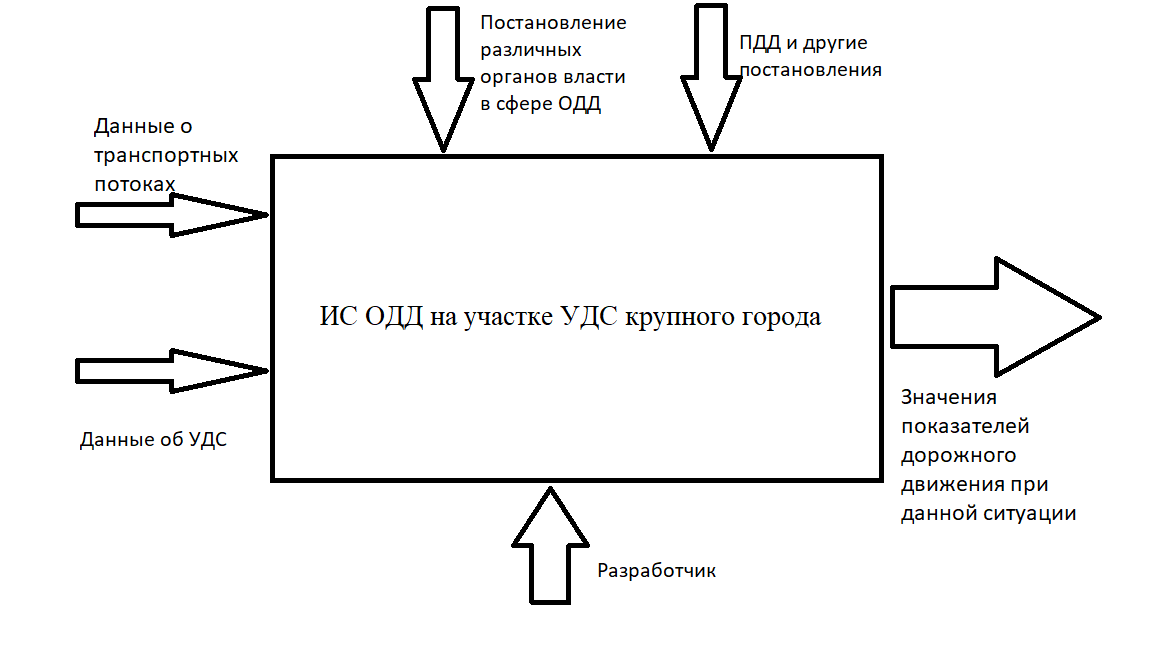


Рисунок 1 – Черный ящик

В «Черном ящике» входные потоки представляют собой данные о транспортных потоках и об улично-дорожной сети.

Управляющий поток представлен ПДД, уголовный и административный кодексы и постановлениями различных органов власти в сфере ОДД.

В функциональном потоке отображено действующее лицо: разработчик.

Выходным потоком является значение показателей дорожного движения при данной ситуации.

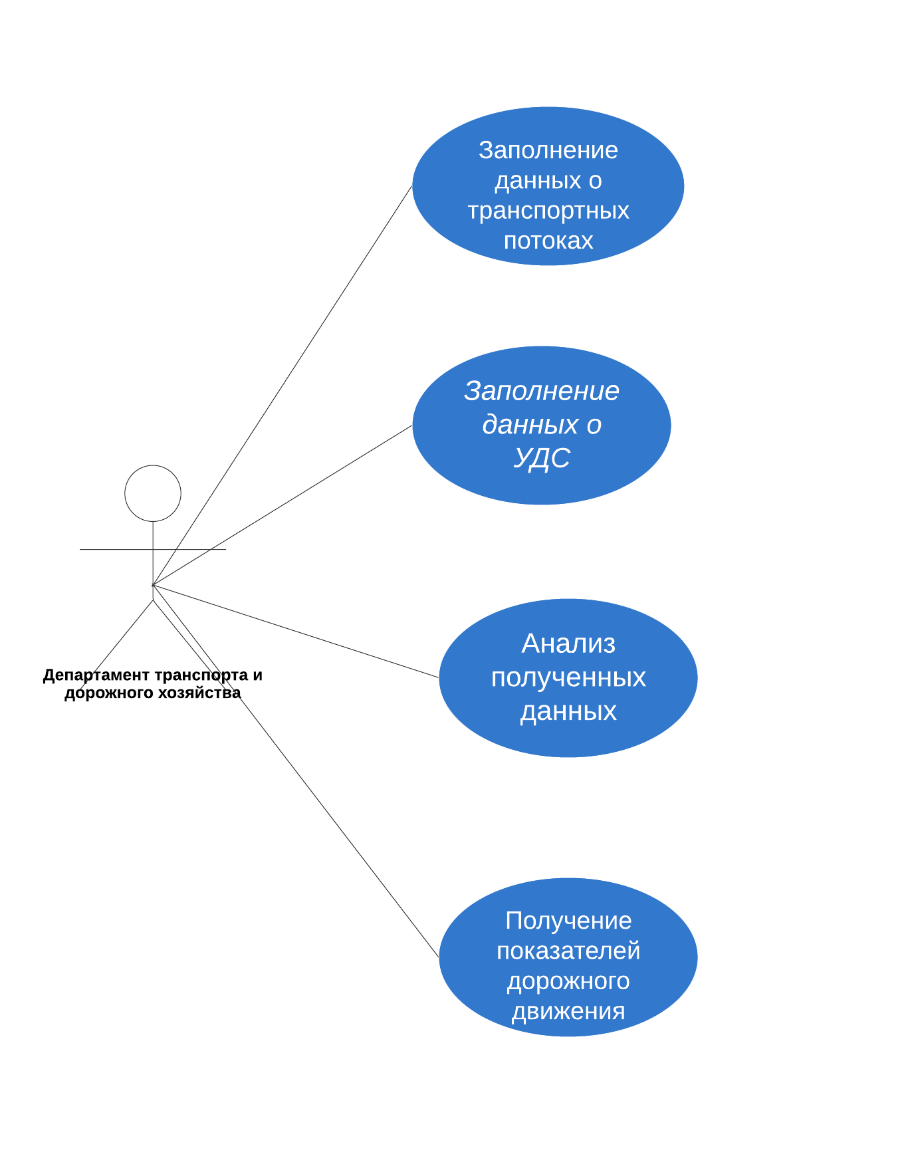


Рисунок 2 – Модель вариантов использования

В данной системе можно выделить следующие подпроцессы:

– заполнение данных о транспортных потоках;

– заполнение данных о УДС;

– анализ полученных данных;

– получение показателей дорожного движения.

Описание основных бизнес-процессов

– процесс заполнения информации. Информация о транспортных потоках и УДС ;

– процесс анализа полученной информации. Получение необходимых данных для составления ИМ;

– процесс применения методов и алгоритмов для построения ИМ;

– процесс получение показателей дорожного движения. Применяя ИМ, полученную на предыдущем шаге, получим необходимые значения.

На вход программа получает данные о транспортных потоках и информацию о дорожной ситуации в конкретный момент времени. Также передается информация, которую нудна смоделировать, такая как, например погодные условия.

На выходе получаем ИМ, которая моделирует ОДД при полученных во входных данных условиях.

**Анализ существующих компьютерных разработок**

**1)Библиотека дорожного движения AnyLogic**

Библиотека дорожного движения AnyLogic позволяет детально планировать, проектировать и моделировать транспортные потоки с учетом индивидуального поведения каждого водителя.

Алгоритмы библиотеки настроены в соответствии с правилами дорожного движения — учет ограничения скорости, «уступи дорогу» и др. В то же время, в моделях дорожного движения каждое транспортное средство представляется в виде агента, который имеет индивидуальные физические параметры и поведенческие шаблоны. Кроме того, у пользователя есть возможность создавать в модели двухмерную и трехмерную анимацию транспортных средств и окружающих объектов. Всё это вместе делает модели дорожного движения наглядными и гибкими.

Библиотека дорожного движения — это инструмент планирования и организации транспортных потоков. В моделях дорожного движения имитируется перемещение машин по улицам и автомагистралям, включая такие элементы как перекрестки, пешеходные переходы, круговое движение, автостоянки и остановки общественного транспорта. Возможности библиотеки позволят решить следующие задачи:

* Планирование дорог и автомагистралей;
* Оценка загруженности и пропускной способности дорог;
* Оптимизация фаз светофоров;
* Интеграция общественных зданий в дорожную сеть.

Библиотека предоставляет пользователю инструменты для моделирования перекрестков неравнозначных дорог, светофоров, пешеходных переходов, автобусных остановок и автостоянок.

**2) Транспортное моделирование от компании "Промтерра"**

Проектно-изыскательская Компания "Промтерра" разрабатывает комплекс инженерных решений по моделированию транспортных потоков для автомобильных дорог как в Москве, так и в других субъектах нашей страны.

На этапе подготовки проектного решения проведение моделирования транспортных потоков подразумевает целый ряд работ, направленных на изучение дорожной ситуации с выделением проблемных участков автомобильных дорог. На основании такого исследования создается компьютерная модель транспортного потока с виртуальной проекцией движения на объекте.

При составлении математической модели обязательно учитывают не только реальную ситуацию передвижения транспорта по временным интервалам, но и будущую загруженность с перспективой развития смежных городских проектов транспортной сети и застройки населенного пункта. Моделирование дает возможность рассчитать среднюю скорость на сложном участке, при этом предложить решения, для увеличения интенсивности движения.

Транспортная имитация включает в себя моделирование транспортных процессов и систем движения на автодорогах как городского типа, так и на трассах федерального значения, соединяющих разные регионы страны в единую сеть.

При условии соблюдения скоростного режима и других требований, даже в моменты пиковой нагрузки - моделирование транспортных потоков, проведенное нашими специалистами с большим опытом таких работ, гарантирует низкий уровень аварийных ситуаций на дороге, долгий срок службы асфальтового покрытия и объектов инфраструктуры.

На основе проектного транспортного моделирования, для увеличения пропускной способности на создаваемом дорожном объекте, в компании проводят компьютерную оптимизацию движения транспорта.

Ниже приведена таблица, основанная на заявленных разработчиками показателей системы.

Таблица 1. Сравнение «AnyLogic» и Транспортное моделирование от компании "Промтерра"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Библиотека дорожного движения AnyLogic** | | **Транспортное моделирование от компании "Промтерра"** | |
| Назначение | Детальное планирование, проектировка и моделирование транспортные потоков с учетом индивидуального поведения каждого водителя. | | Создание рабочей модели дорожного движения (транспортная имитация), соответствующего движению в реальных условиях на автомобильных дорогах. | |
| Эффективность | Есть (Легкий интерфейс, не требующий большого количества времени на освоение) | | Есть (Легкий интерфейс, не требующий большого количества времени на освоение) | |
| Гибкость | Да (можно самому вносить данные для моделирования) | | Нет (все расчеты происходят на стороне, потребитель сразу получает готовую модель) | |
| Открытость | Нет (проприетарное ПО) | | Нет (проприетарное ПО) | |
| Наглядность интерфейса | Есть (дружелюбный интерфейс) | | Есть (простой интерфейс) | |
| Сопровождаемость | | Частично (возможно внесение данных пользователем) | | Нет (данные строго по трафику) |
| Стоимость | | Высокая (от 13 тыс. рублей) | | Высокая (от 400 тыс. рублей до 550 тыс. рублей) |
| Эффективность | | Высокая (полученная модель согласно данным от пользователя или основе реальных данных о трафике) | | Высокая (полученная модель на основе реальных данных о трафике) |

**Техническое задание на создание программного продукта**

Данное программное обеспечивание направленно на создание правдоподобной трехмерной модели организации дорожного движения (ОДД) на участке улично-дорожной сети (УДС) крупного города.

*Целью* данного программного продукта является помощь сотрудникам Центра организации дорожного движения (ЦОДД) оценивать поведение транспортных потоков на различных участках УДС при различных внешних факторах. Человек не способен учесть все возможные факторы и просчитать все возможные варианты развития событий, из-за чего данную задачу логичнее всего доверить компьютеру. Он сможет выполнять эту задачу быстрее, тем самым сократит временные затраты сотрудников ЦОДД.

Структура программы: Подсистема получения информации должна предоставить удобный интерфейс для ввода данных о транспортных потоках, участке УДС, различных происшествиях на участке УДС. Эта информация сохраняются в базе данных, при необходимости сотрудник ЦОДД может изменить и удалить эти данные. Подсистема построения модели берет данные о транспортных потоках, участке УДС, различных происшествиях на участке УДС и на основе этих данных и некоторых математических и геометрических методов строит трехмерную модель. Подсистема вывода отвечает за просмотр получившейся модели и должна предоставлять удобный интерфейс взаимодействия с нею.

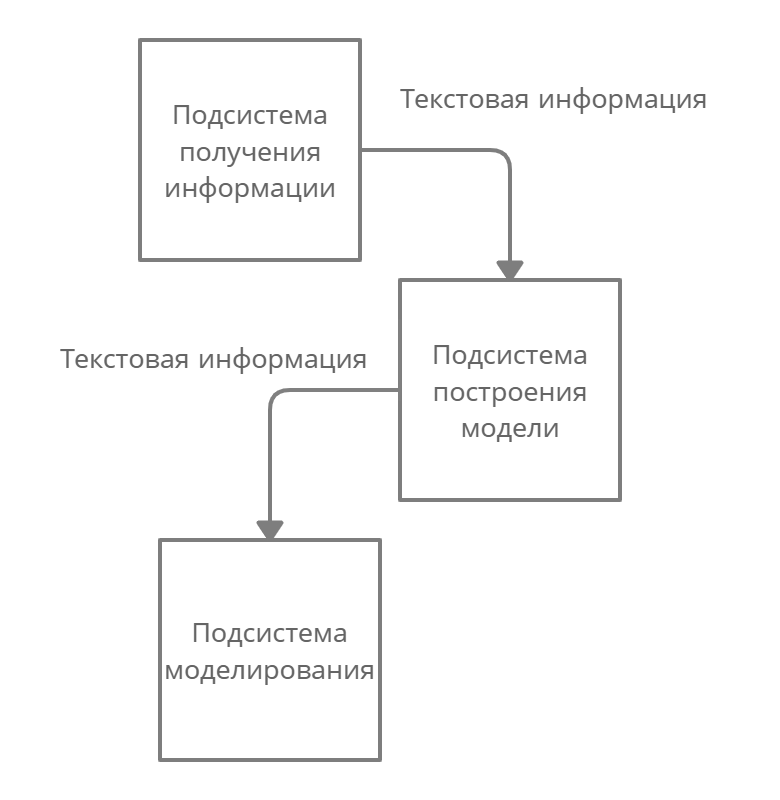


Рисунок 3 – Структура взаимодействия между подсистемами и базой данных

Выделим следующие задачи:

1. Заполнение, изменение или удаление информации о транспортных потоках, участках УДС, различных происшествиях на участке УДС в базу данных;
2. Анализ данных;
3. Построение модели.

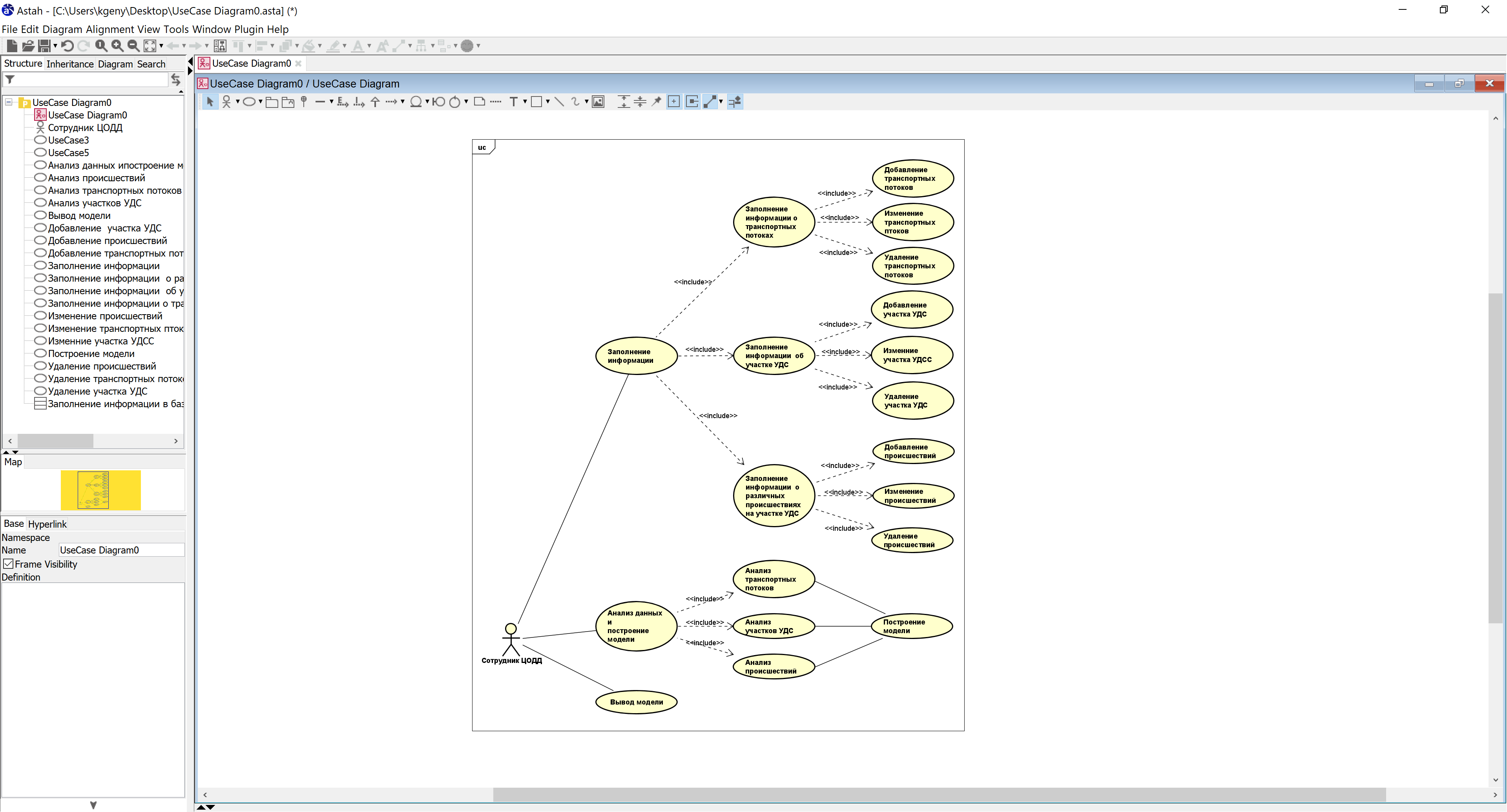


Рисунок 4 – Диаграмма вариантов использования

Требования к задаче “Заполнение актуальной информации в базу данных”: для реализации данной задачи программа должна предоставить пользователю удобный интерфейс для занесения, изменения и удаления информации о транспортных потоках, участках УДС, различных происшествиях на участке УДС.

Требования к задаче “Анализ транспортных потоков: в данной задаче необходим анализ транспортный поток. Для этого надо проанализировать поведение каждого отдельно взятого объекта транспортного потока, будь то автомобиль, мотоцикл, трамвай или другое. У каждого ТС есть цель: доехать до конца своего маршрута. Каждый водитель будет добиваться данной цели разными способами, используя разный алгоритм или поведение. Поведенческие принципы пользователей транспортной сети предполагают следующие две возможные ситуации, получившие название *1-го и 2-го* *принципа Вардропа* соответственно:

1. Пользователи сети независимо друг от друга выбирают маршруты следования, соответствующие их минимальным транспортным расхода. То есть каждый стремится достигнуть конечного пункта своей поездки как можно выгоднее для себя и из имеющихся возможных вариантов следования выбирает тот маршрут, по которому будет нести минимальные затраты (временные, финансовые, моральные и т. п.) на проезд.
2. Пользователи сети выбирают маршруты следования исходя из минимизации общих транспортных расходов в сети. Предполагает централизованное управление движением в сети. Такое поведение присуще транспортному средству.

Подавляющее большинство участников дорожного движения любого города составляют легковые автомобили, совершающие преимущественно маятниковые поездки: место проживания— место работы и обратно. Именно такие поездки создают пиковые нагрузки на УДС, вызывают основные потери времени и других ресурсов, повышают аварийность и усложняют социально-экономическую ситуацию

Таким образом, *данную задачу можно свести к решению транспортной задачи для каждого субъекта транспортного потока.*

Требования к задаче “Анализ УДС”: Для данной задачи необходимо проанализировать данные об УДС, ее состоянии, технических характеристиках, размерах, ограничений на ней, об регулирующих объектах(знаки и/или светофорные объекты). Также данная задача включает в себя анализ происшествий, например ДТП, природных происшествий или другое. Данная задача необходима для моделирования УДС при различных ситуациях.

Требования к задаче “Построение модели”: ЦОДД наиболее всего заинтересован в моделировании в утренне-вечерние часы пик.

Модель должна строиться средствами Unity в сцене.

Данная задача включает подзадачу вывода построенной модели. Она решается путем вывода построенной сцены и возможностью пользователя взаимодействовать с ней.

Требования к выходным данным: программа должна выводить трехмерную модель. Пользователь должен иметь возможность взаимодействия с ней.

Требования к интерфейсу программы: интерфейс должен быть прост и удобен в использовании для непродвинутого пользователя.

Требования к программному обеспечению:требуемая OC: Windows, macOS, Linux. Требуемый язык программирования: С#, редактор кода для языка: VSCode c расширением для Unity. Требуемый игровой движок: Unity. СУБД: MySQL.

**Проектирование функциональной структуры программного продукта: функционально-ориентированный подход**

Для разработки контекстной диаграммы в нотации IDEF0, используем модель «черный ящик», который был разработан при написании лабораторной работы №1 и немного дополнен в этой лабораторной работе.



Рисунок 5 – Модель «Черный ящик»

Декомпозируем диаграмму «черного ящика» и выполним разработку контекстной диаграммы.



Рисунок 6 – Диаграмма декомпозиции A0

****

Рисунок 7 – Декомпозиция подзадачи

“Заполнение актуальной информации в базу данных”

Таблица 2. Активности диаграммы декомпозиции заполнение данных

|  |  |
| --- | --- |
| Наименование действия | Определение действия |
| Заполнение информации о транспортных потоках, участках УДС | Сотрудник ЦОДД заполняет данные о транспортных потоках, участках УДС |
| Сохранение данных в соответствующие таблицы базы данных | Программа сохраняет введенные ранее данные в соответствующие таблицы базы данных |

Таблица 3. Описание элементов функциональной модели

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование стрелки | Источник стрелки | Тип стрелки источника | Приемник стрелки | Тип стрелки приемника |
| Данные о транспортных потоках | Внешняя граница | Output | Заполнение информации о транспортных потоках, участках УДС | Input |
| Данные об УДС | Внешняя граница | Output | Заполнение информации о транспортных потоках, участках УДС | Input |
| Различные ФЗ | Внешняя граница | Output | Заполнение информации о транспортных потоках, участках УДС | Control |
| Различные Постановления Правительства | Внешняя граница | Output | Заполнение информации о транспортных потоках, участках УДС | Control |
| ПДД,  АК,  УК | Внешняя граница | Output | Заполнение информации о транспортных потоках, участках УДС | Control |
| Различные  ГОСТ'ы,  СНИП'ы | Внешняя граница | Output | Заполнение информации о транспортных потоках, участках УДС | Control |
| Градостроительный  кодекс | Внешняя граница | Output | Заполнение информации о транспортных потоках, участках УДС | Control |
| Сотрудник ЦОДД | Внешняя граница | Mechanism | Заполнение информации о транспортных потоках, участках УДС | Mechanism |
| Внесенные данные | Заполнение информации о транспортных потоках, участках УДС | Output | Заполнение информации о транспортных потоках, участках УДС | Input |
| Данные о транспортных потоках | Сохранение данных в соответствующие таблицы базы данных | Output | Внешняя граница | Input |
| Данные об УДС | Сохранение данных в соответствующие таблицы базы данных | Output | Внешняя граница | Input |



Рисунок 8 – Декомпозиция подзадачи “Анализ данных”

Таблица 4. Активности диаграммы декомпозиции заполнение данных

|  |  |
| --- | --- |
| Наименование действия | Определение действия |
| Формирование запроса на анализ | Программа получает на вход данные и запрос на анализ и формирует данные для последующего анализа |
| Анализ УДС | Программа подготавливает данные для анализа |
| Анализ транспортных потоков | Программа анализирует транспортные потоки |
| Анализ отдельных ТС | Программа анализирует отдельные ТС |

Таблица 5. Описание элементов функциональной модели

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование стрелки | Источник стрелки | Тип стрелки источника | Приемник стрелки | Тип стрелки приемника |
| Данные об УДС | Внешняя граница | Output | Формирование запроса на анализ | Input |
| Данные о транспортных потоках | Внешняя граница | Output | Формирование запроса на анализ | Input |
| Запрос на создание модели | Внешняя граница | Output | Формирование запроса на анализ | Input |
| Данные об УДС | Формирование запроса на  анализ | Output | Анализ УДС | Input |
| Данные о транспортных потоках | Формирование запроса на анализ | Output | Анализ транспортных потоков | Input |
| Данные об  отдельном ТС | Анализ транспортных потоков | Output | Анализ отдельных ТС | Input |
| Проанализированные данные об УДС | Анализ УДС | Output | Внешняя граница | Input |
| Проанализированные данные о транспортных потоках | Анализ отдельных ТС | Output | Внешняя граница | Input |



Рисунок 9 – Декомпозиция подзадачи “ Построение модели”

Таблица 6. Активности диаграммы создания модели

|  |  |
| --- | --- |
| Наименование действия | Определение действия |
| Моделирование УДС | Программа моделирует УДС |
| Моделирование транспортных потоков | Программа моделирует транспортные потоки путем моделирования отдельных ТС |
| Моделирование отдельных ТС | Программа моделирует отдельные ТС |
| Вывод модели | Программа производит вывод построенной модели |

Таблица 7. Описание элементов функциональной модели

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование стрелки | Источник стрелки | Тип стрелки источника | Приемник стрелки | Тип стрелки приемника |
| Проанализированные данные об УДС | Внешняя граница | Output | Моделирование УДС | Input |
| Проанализированные данные о транспортных потоках | Внешняя граница | Output | Моделирование транспортных потоков | Input |
| Проанализированные данные об отдельных ТС | Моделирование транспортных потоков | Output | Моделирование отдельных ТС | Input |
| Модель  транспортных  потоков | Моделирование отдельных ТС | Output | Вывод модели | Input |
| Модель УДС | Моделирование отдельных ТС | Output | Вывод модели | Input |
| Значение показателей  дорожного движения  в данной ситуации | Вывод модели | Output | Внешняя граница | Input |

**Проектирование функциональной структуры программного продукта: объектно-ориентированный подход.**

Таблица 8. Абстракции подсистемы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракция | Тип | Описание |
| 1 | Транспортное средство | Абстракция сущности | Транспортное средство, участвующее в дорожном движении на определенном участке улично-дорожной сети |
| 2 | Транспортный поток | Абстракция сущности | Совокупность транспортных средств, одновременно участвующих в дорожном движении на определенном участке улично-дорожной сети |
| 3 | Улично-дорожная сеть | Абстракция сущности | Комплекс объектов транспортной инфраструктуры, предназначенной для движения транспортных средств и пешеходов. Представляет собой взаимосвязанную систему городских улиц и автомобильных дорог |
| 4 | Запрос на создание модели | Абстракция сущности | Поступающий извне запрос на составление модели |
| 5 | Сотрудник ЦОДД | Абстракция сущности | Физическое лицо, вносящее изменения в базу данных, а также выполняющее запуск работы программной системы |
| 6 | Трехмерная модель | Абстракция сущности | Трехмерное представление модели, построенное по исходным данным |
| 7 | Создание модели | Абстракция поведения | Создание трехмерной модели из исходных данных |
| 8 | Показатели дорожного движения | Абстракция сущности | Поведение трехмерной модель при введенных данных |
| 9 | Анализ данных | Абстракция поведения | Анализ полученных данных для построения модели |
| 10 | Диалоговые окна | Абстракция интерфейса | Способ взаимодействия программной системы и сотрудника ЦОДД |

Таблица 9. Классификация абстракций

|  |  |  |
| --- | --- | --- |
| № | Класс | Список абстракций |
| 1 | Люди | Сотрудник ЦОДД |
| 2 | Места |  |
| 3 | Предметы | Транспортный поток, улично-дорожная сеть, трехмерная модель, диалоговые окна |
| 4 | Организации |  |
| 5 | Концепции | Показатели дорожного движения |
| 6 | События | Запрос на создание модели, анализ данных, создание модели |

Таблица 10. Абстракции подсистемы и их поведение

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракция | Поведение | Описание поведения |
| 1 | Сотрудник ЦОДД | Управление работой ПО | Запуск программной системы  Занесение актуальной информации в базу данных |
| 2 | Создание модели | Создание модели | Создание трехмерной модели из исходных данных |
| 3 | Трехмерная модель | Вывод модели | Вывод построенной трехмерной модели |
| 4 | Диалоговое окно | Передача данных или запуск | Предоставляет данные и запросы от сотрудника ЦОДД к соответствующим подсистемам |

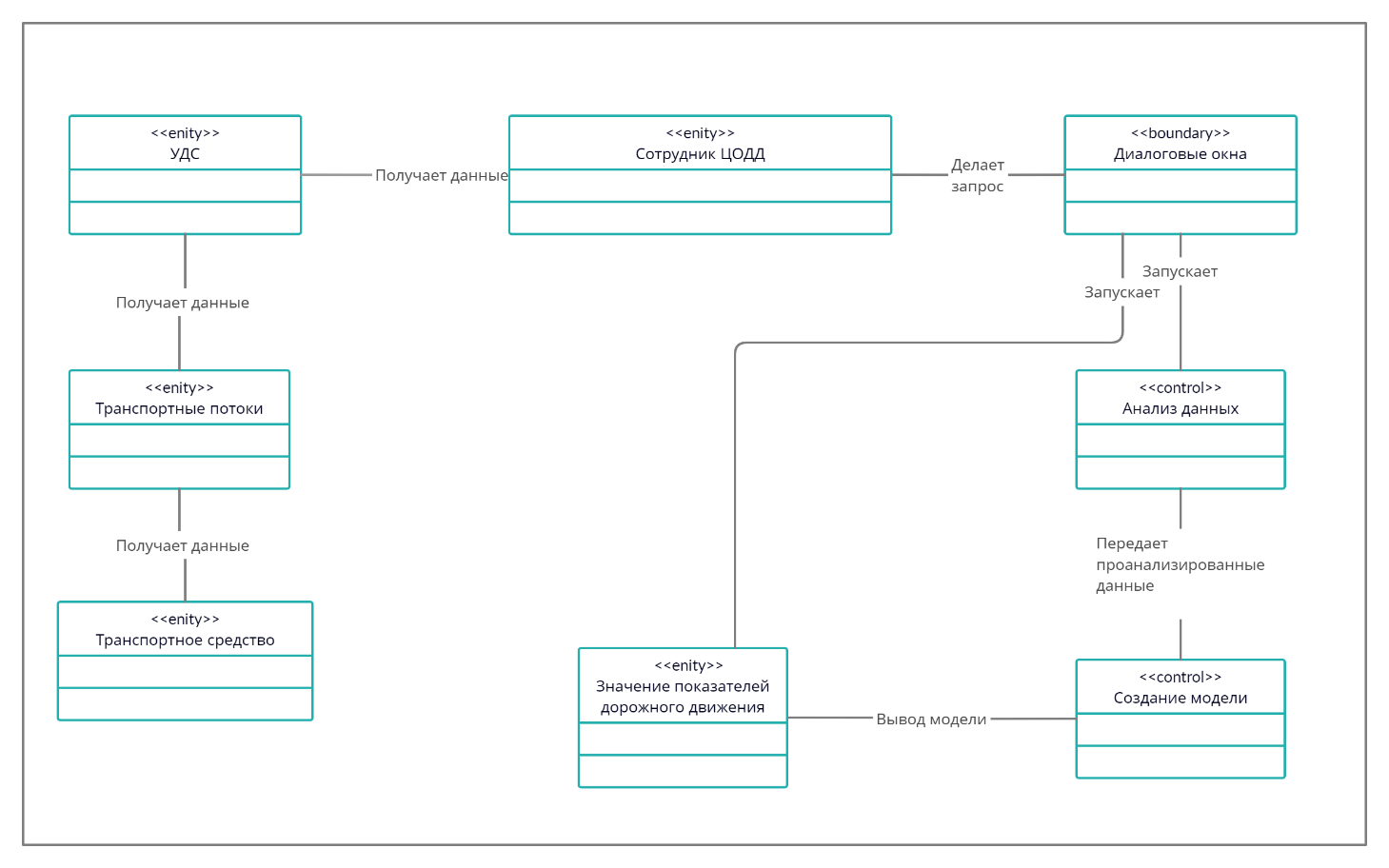


Рисунок 10 – Диаграмма классов UML

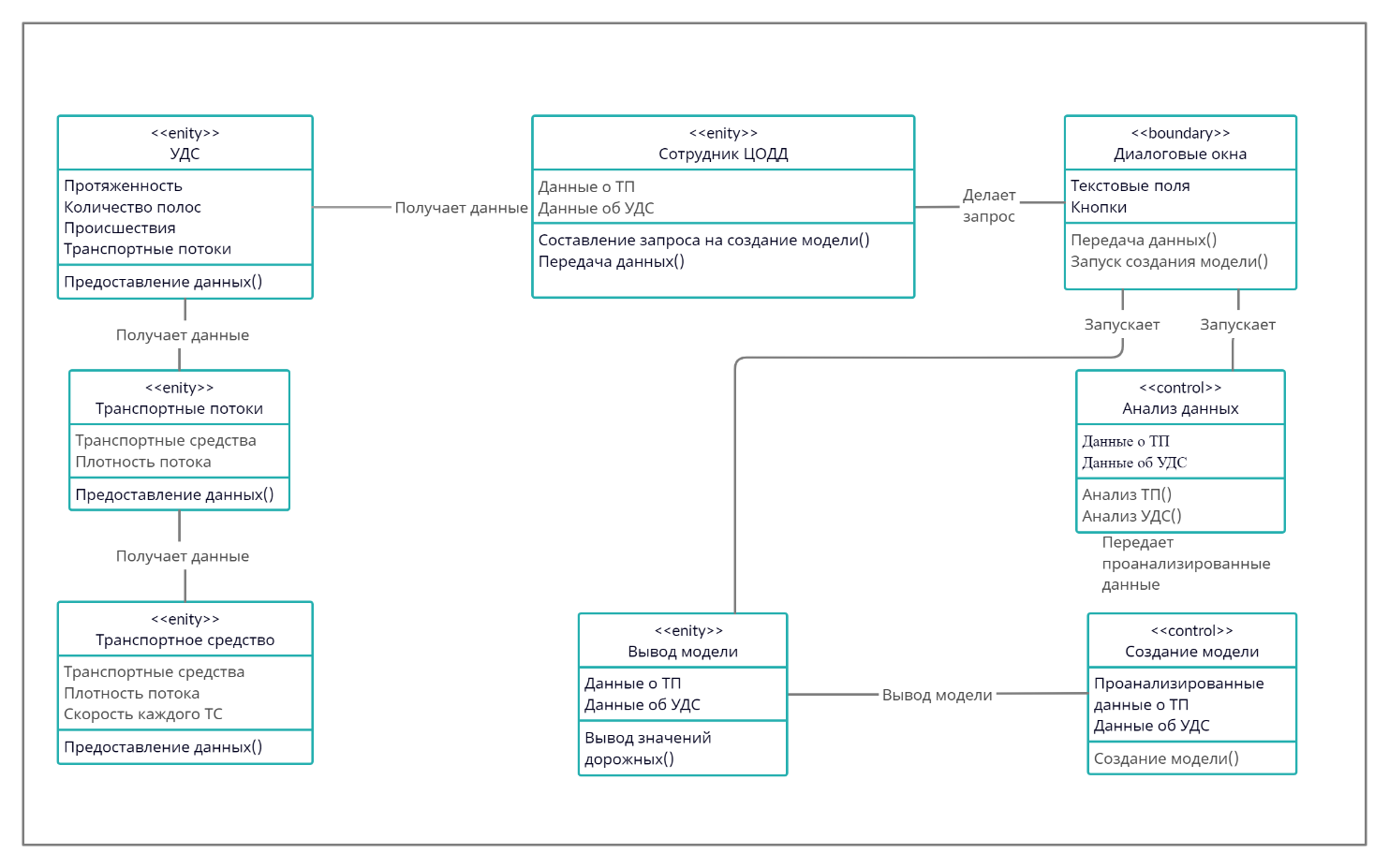


Рисунок 11 – Расширенная диаграмма классов UML

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 – Диаграмма состояний для класса “Данные о транспортных потоках”

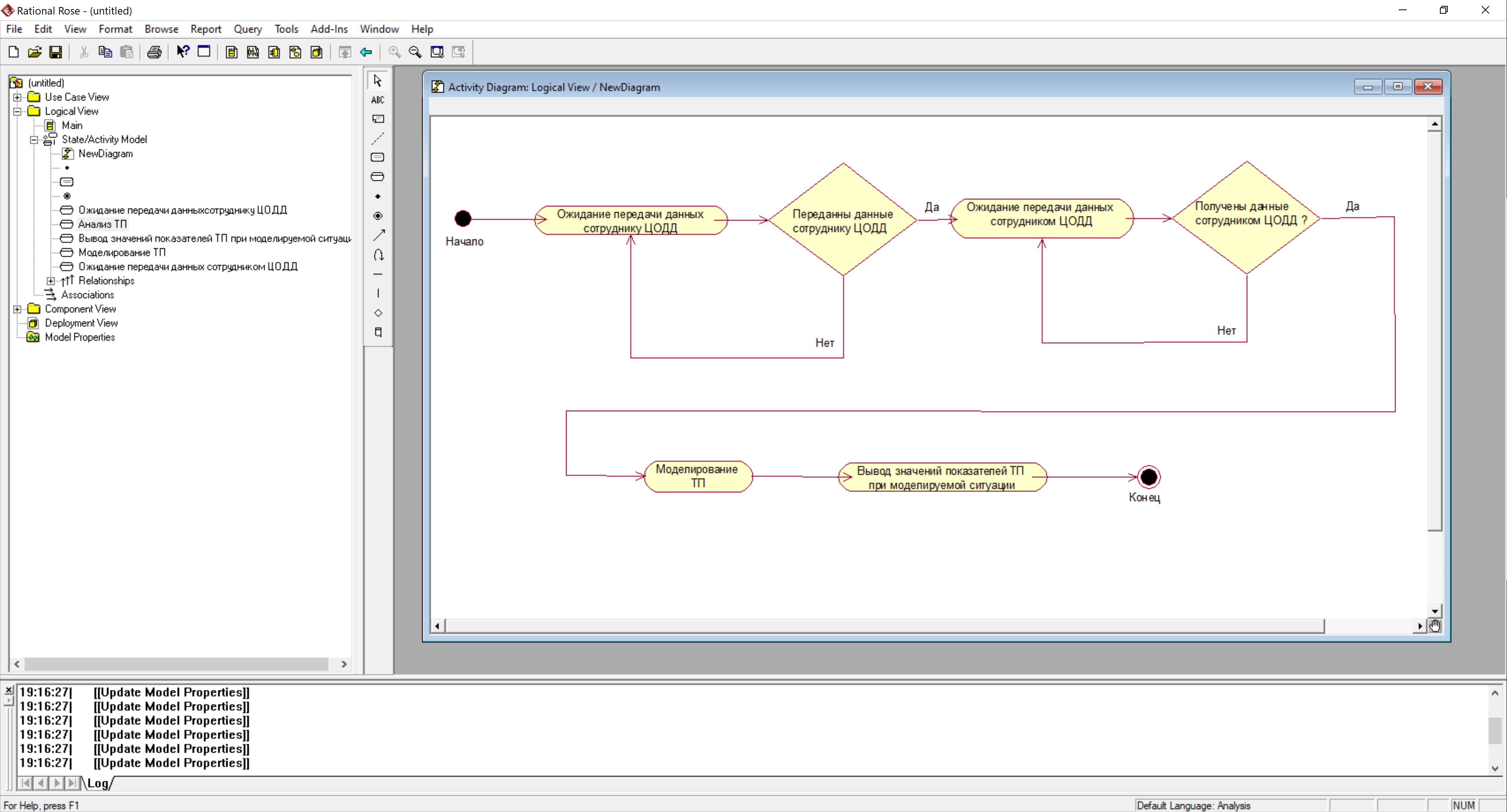


Рисунок 13 – Диаграмма деятельности для класса “Данные о транспортных потоках”

Таблица 11. Список объектов для каждого потока событий

|  |  |  |  |
| --- | --- | --- | --- |
| № п. п | Прецедент | Объект | Описание объекта |
| 1.1 | Ожидание передачи данных сотруднику ЦОДД | Сотрудник ЦОДД | Ожидает получение информации о ТП |
| 1.2 | Ожидание передачи данных сотруднику ЦОДД | Информация об УДС | Хранит информацию о начальных данных ТП |
| 2.1 | Ожидание передачи данных сотрудником ЦОДД | Сотрудник ЦОДД | Передает данные системе для моделирования |
| 2.2 | Ожидание передачи данных сотрудником ЦОДД | Алгоритмы моделирования | Ожидает получение данных от сотрудника ЦОДД |
| 3.1 | Моделирование ТП | Алгоритмы моделирования | Ожидает получения проанализированных данных |
| 3.2 | Моделирование ТП | Имитационная модель | Построенная имитационная модель по полученным данным |
| 4.2 | Вывод значение показателей дорожного движения при моделируемой ситуации | Имитационная модель | Построенная имитационная модель по полученным данным |
| 4.2 | Вывод значение показателей дорожного движения при моделируемой ситуации | Значения показателей | Значения показателей дорожного движения при заданной ситуации |

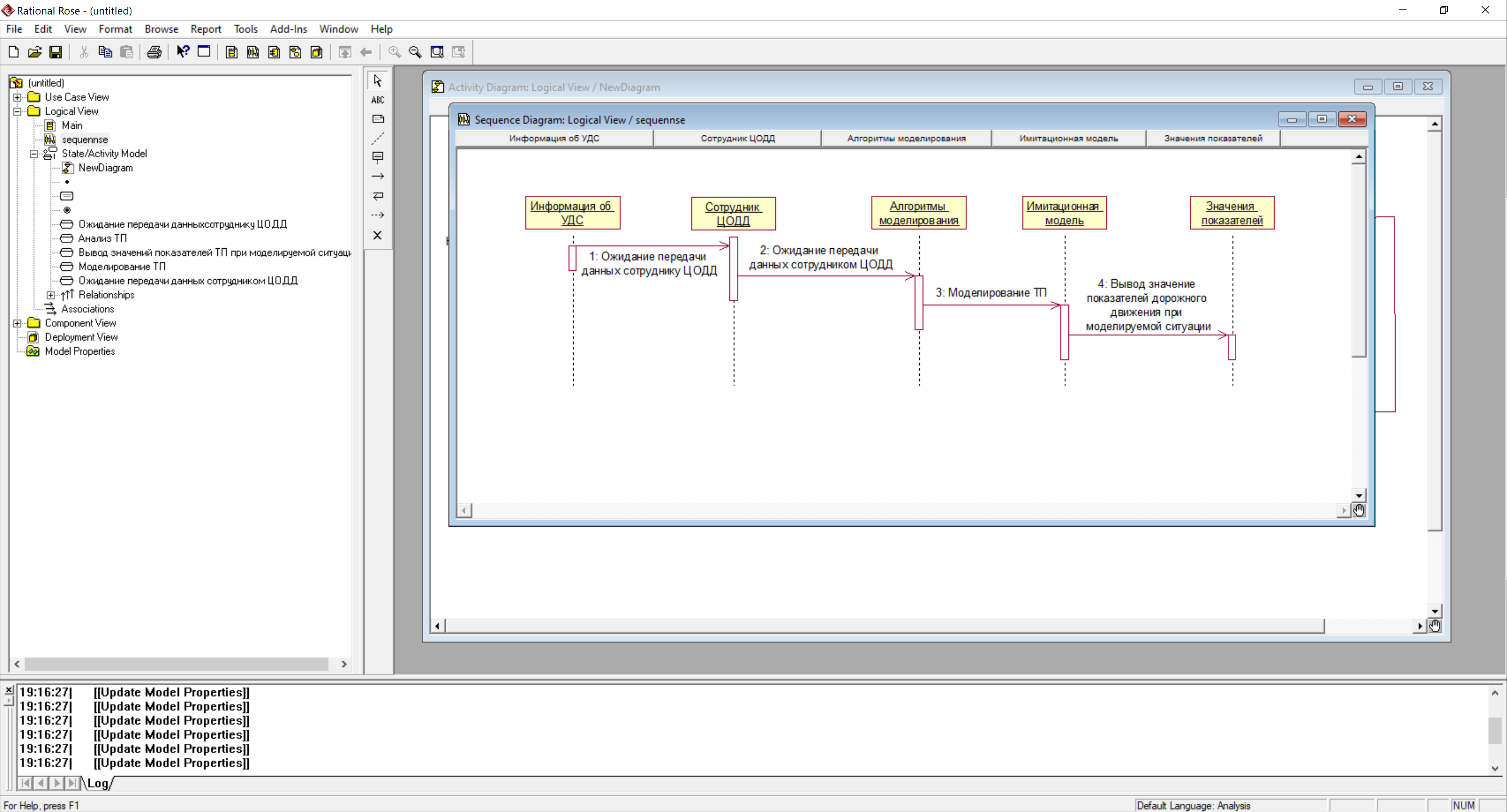


Рисунок 14 – Диаграмма последовательности для перечисленных прецедентов

**Проектирование базы данных программной системы**

На основе анализа предметной области была выделена следующая информация:

* Информация об объектах поля. Представляет собой информацию об объектах, которые являются объектами окружения (здания, дороги и т.д.). Содержит название объекта, название префаба (объекта, на котором основан данный объект), его координаты (x, y, z) и поворот по осям координат();
* Информация об объектах транспортного потока. Представляет собой информацию об объектах, которые непосредственно являются объектами транспортного потока. Содержит название объекта, его координаты (x, y, z) и поворот по осям координат() и скорость в момент сохранения;
* Информация об участке УДС. Представляет собой информация об возможных направления на данном участке. Направления является набором точек с координатами (x, y, z);
* Информация о препятствиях. Представляет собой информацию об объектах, которые являются объектами препятствий (аварии, дорожные работы, припаркованные автомобили и т.д.). Содержит название объекта, название префаба, его координаты (x, y, z) и поворот по осям координат();

Для хранения информации об объектах поля, участке УДС нет необходимости хранить в базе данных, так как данные объекты являются постоянными и нет необходимости их дублировать

Хранить непостоянную информацию в Unity принято не через базы данных, а через JSON-файлы. Поэтому при хранении информации об объектах транспортного потока и о препятствиях будем хранить в JSON-файле.

При помощи CASE-средства разработана ER-диаграмма логической и физической модели данных.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 15 – ER-диаграмма логической модели данных

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 16 – ER-диаграмма физической модели данных

Таблица 12. Сущности физической модели данных

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сущность | Атрибуты | Описание |
| Оперативные | | | |
| 1 | Транспортный поток | ID, Название, Имя\_префаба, Координата\_X, Координата\_Y, Координата\_Z, Поворот\_по\_оси\_X, Поворот\_по\_оси\_Y, Поворот\_по\_оси\_Z, Скорость | Информация об объектах транспортного потока |
| 2 | Препятствия | ID, Название, Имя\_префаба, Координата\_X, Координата\_Y, Координата\_Z, Поворот\_по оси\_X, Поворот\_по оси\_Y, Поворот\_по оси\_Z, Тип\_препятствия | Информация о цифровой компетенции |

Таким образом JSON-формат для обеих таблиц будет иметь следующий вид:

{

"Транспортные потоки":

[

{

"id":1,

"Название":"Volga1",

"Имя\_префаба":"Volga",

"Координата\_X":25,

"Координата\_Y":0.1,

"Координата\_Z":45,

"Поворот\_по\_оси\_X": 0,

"Поворот\_по\_оси\_Y": 90,

"Поворот\_по\_оси\_Z": 0,

"Скорость": 35.4

}

],

"Препятствия":

[

{

"id":1,

"Название":"Ремонт3.1",

"Имя\_префаба":"Ремонт3",

"Координата\_X":45,

"Координата\_Y":0.1,

"Координата\_Z":45,

"Поворот\_по\_оси\_X": 0,

"Поворот\_по\_оси\_Y": 0,

"Поворот\_по\_оси\_Z": 0,

"Тип\_препятствия": "Ремонт"

}

]

}

Описание таблиц БД приведено в таблицах 14, 15, краткое описание этих таблиц сведено в таблицу 16, описание связей между таблицами отсутствует.

Таблица 14. Структура таблицы «Транспортный поток»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Значение по умолчанию | Примечание |
| ID | ЧИСЛО | - | - | - | Обязательное поле, первичный ключ |
| Название | СТРОКА | - | - | - | Обязательное поле |
| Имя\_префаба | СТРОКА | - | - | - | Обязательное поле |
| Координата\_X | ЧИСЛО | - | - | - | Обязательное поле |
| Координата\_Y | ЧИСЛО | - | - | - | Обязательное поле |
| Координата\_Z | ЧИСЛО | - | - | - | Обязательное поле |
| Поворот\_по\_оси\_X | ЧИСЛО | - | - | - | Обязательное поле |
| Поворот\_по\_оси\_Y | ЧИСЛО | - | - | - | Обязательное поле |
| Поворот\_по\_оси\_Z | ЧИСЛО | - | - | - | Обязательное поле |
| Скорость | ЧИСЛО | - | - | - | Обязательное поле |

Таблица 15. Структура таблицы «Препятствия»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Размер | Условие на значение | Значение по умолчанию | Примечание |
| ID | ЧИСЛО | - | - | - | Обязательное поле, первичный ключ |
| Название | СТРОКА | - | - | - | Обязательное поле |
| Имя\_префаба | СТРОКА | - | - | - | Обязательное поле |
| Координата\_X | ЧИСЛО | - | - | - | Обязательное поле |
| Координата\_Y | ЧИСЛО | - | - | - | Обязательное поле |
| Координата\_Z | ЧИСЛО | - | - | - | Обязательное поле |
| Поворот\_по\_оси\_X | ЧИСЛО | - | - | - | Обязательное поле |
| Поворот\_по\_оси\_Y | ЧИСЛО | - | - | - | Обязательное поле |
| Поворот\_по\_оси\_Z | ЧИСЛО | - | - | - | Обязательное поле |
| Тип\_препятствия | СТРОКА | - | - | - | Обязательное поле |

|  |  |  |
| --- | --- | --- |
| № п/п | Имя таблицы | Описание |
| 1 | Транспортный поток | Информация об объектах транспортного потока |
| 2 | Препятствия | Информация о препятствиях |

Таблица 16. Список разработанных таблиц

**Разработка программного обеспечения ПС**

В соответствии с требованиями технического задания выбраны средства разработки, представленные на рис. 17.

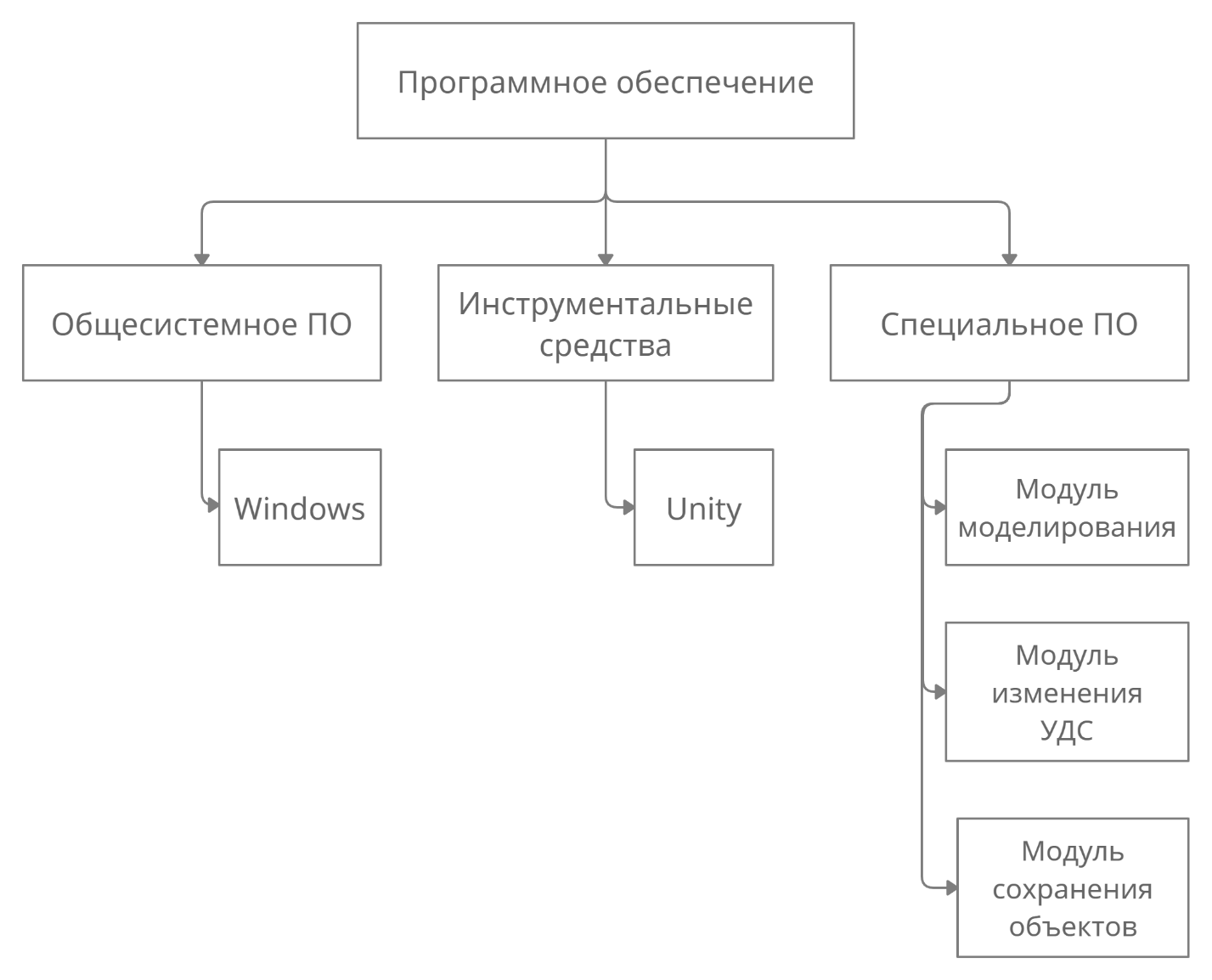


Рисунок 17– Структурная схема общесистемного программного обеспечения.

В соответствии с требованиями, описанными в техническом задании, было разработано программное обеспечение. Описание разработанных компонентов представлено в таблице 17.

Таблица 17. Перечень разработанных компонентов приложения

|  |  |  |
| --- | --- | --- |
| № | Имя компонента | Описание |
| 1 | Модуль моделирования | Производит моделирование и вывод модели |
| 2 | Модуль изменение УДС | Вносит изменения в модель |
| 3 | Модуль сохранения объектов | Сохраняет состояние объектов в момент сохранения |

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание

Рисунок 18 – Модули «Моделирование», «Изменение УДС», «Сохранение объектов»

На рис.18:

* Модуль «Моделирование»:
  + выводная модель;
  + вывод «Средняя скорость потока»;
  + кнопка «Пауза»;
* Модуль «Изменение УДС»:
  + «Добавить аварию»;
  + «Добавить дорожные работы»;
* Модуль «Сохранение объектов»:
  + кнопка «Сохранить».

Структурная схема приложения изображена на рис. 19.

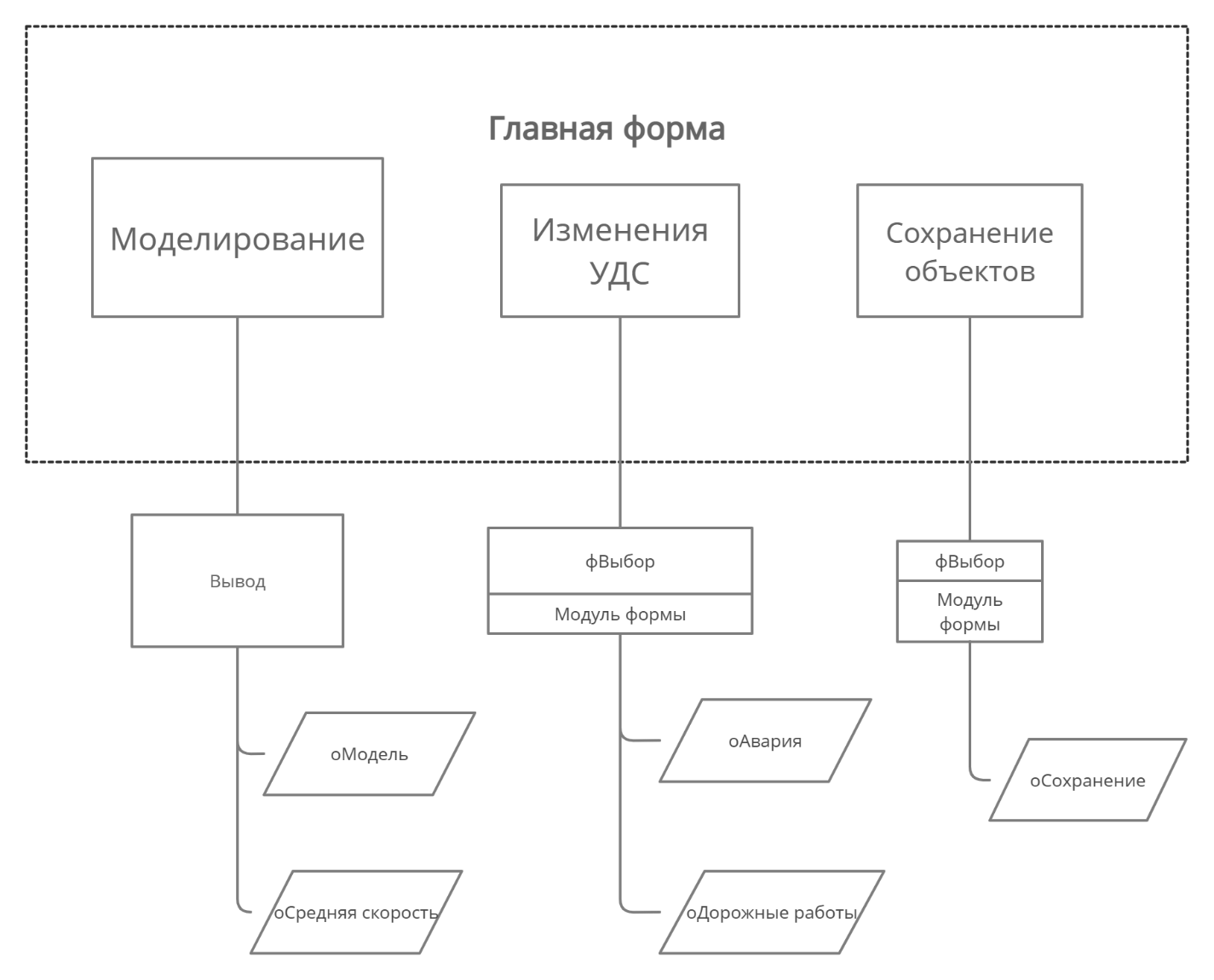


Рисунок 19 – Структурная схема приложения

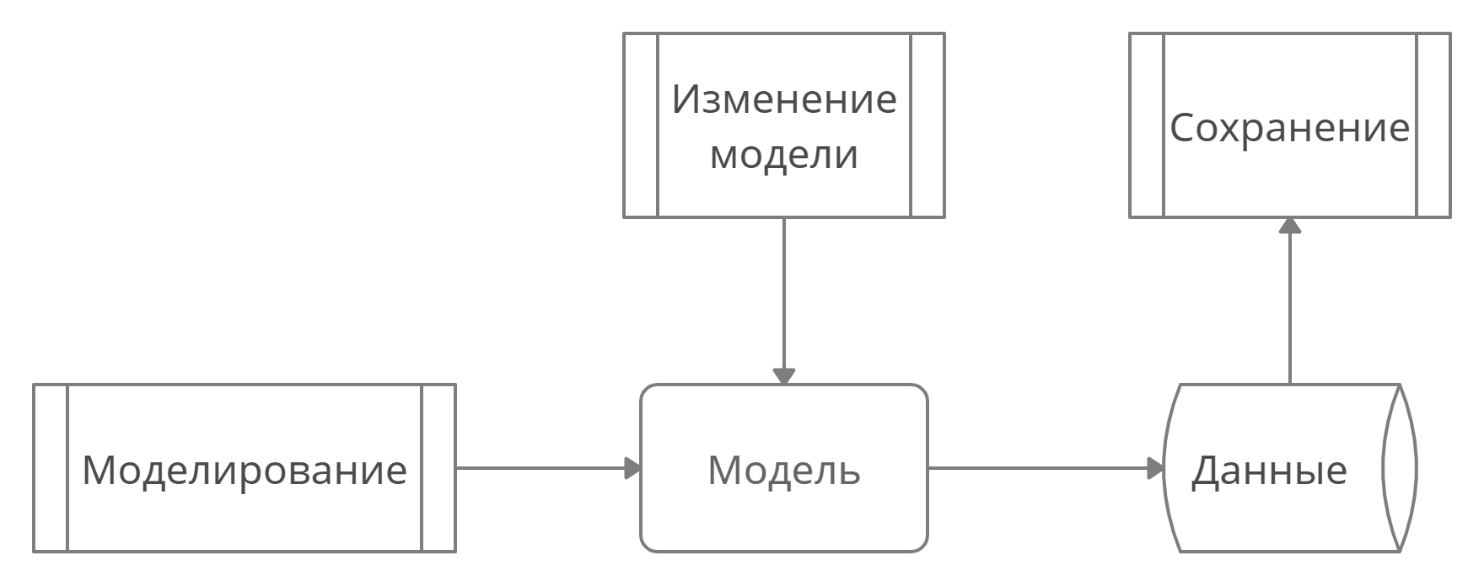


Рисунок 20 – Схема взаимодействия компонентов специального ПО

При компиляции проекта синтаксических и логических ошибок не обнаружено. Результат работы совпадает с ожидаемым.

Рефакторинг кода не понадобился, так как поведение всех компонентов изначально задано как циклическое

**Тестирование программной системы**

В качестве нагрузочного теста был выполнен подсчет FPS(частота кадров в секунду) для модели с количеством объектов 1, 5, 10, 15, 20. Данные по тестированию приведены на рис. 21.

Рисунок 21 ­– График среднего FPS в зависимости от количества объектов

Для тестирования модуля «Моделирование» был разработан следующий скрипт тестирования, результат тестирования представлен на рис. 22, 23.

public class NewTestScript

{

    [UnityTest]

    // класс теста

    public IEnumerator createModel()

    {

        yield return null;

    }

}

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 22 ­– Результат 1 для теста модуля «Моделирование»

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 23 ­– Результат 2 для теста модуля «Моделирование»

Для тестирования модуля «Создание препятствий» был разработан следующий скрипт тестирования, результат тестирования представлен на рис. 24, 25.

public class Test

{

    [UnityTest]

    // класс теста

    public IEnumerator CarSpawnerTest()

    {

        // Создаем пустой объукт

        GameObject go = new GameObject();

        // Делаем его объектом класса CarSpawner

        CarSpawner cs = go.AddComponent<CarSpawner>();

        // Вызываем необходимую для теста функцию

        cs.spawnCar();

        yield return null;

    }

}

public class CarSpawner : MonoBehaviour

{

    // проверяемая функция

    public void spawnCar() {

        // Загружаем объект "авария"

        GameObject carPrefabs = (GameObject)Resources.Load("Prefabs/Crash 1", typeof(GameObject));

        //ставим на позицию

        carPrefabs.transform.position = GameObject.Find("Spawner").transform.position;

        //Создаем экземпляр

        Instantiate(carPrefabs);

    }

}

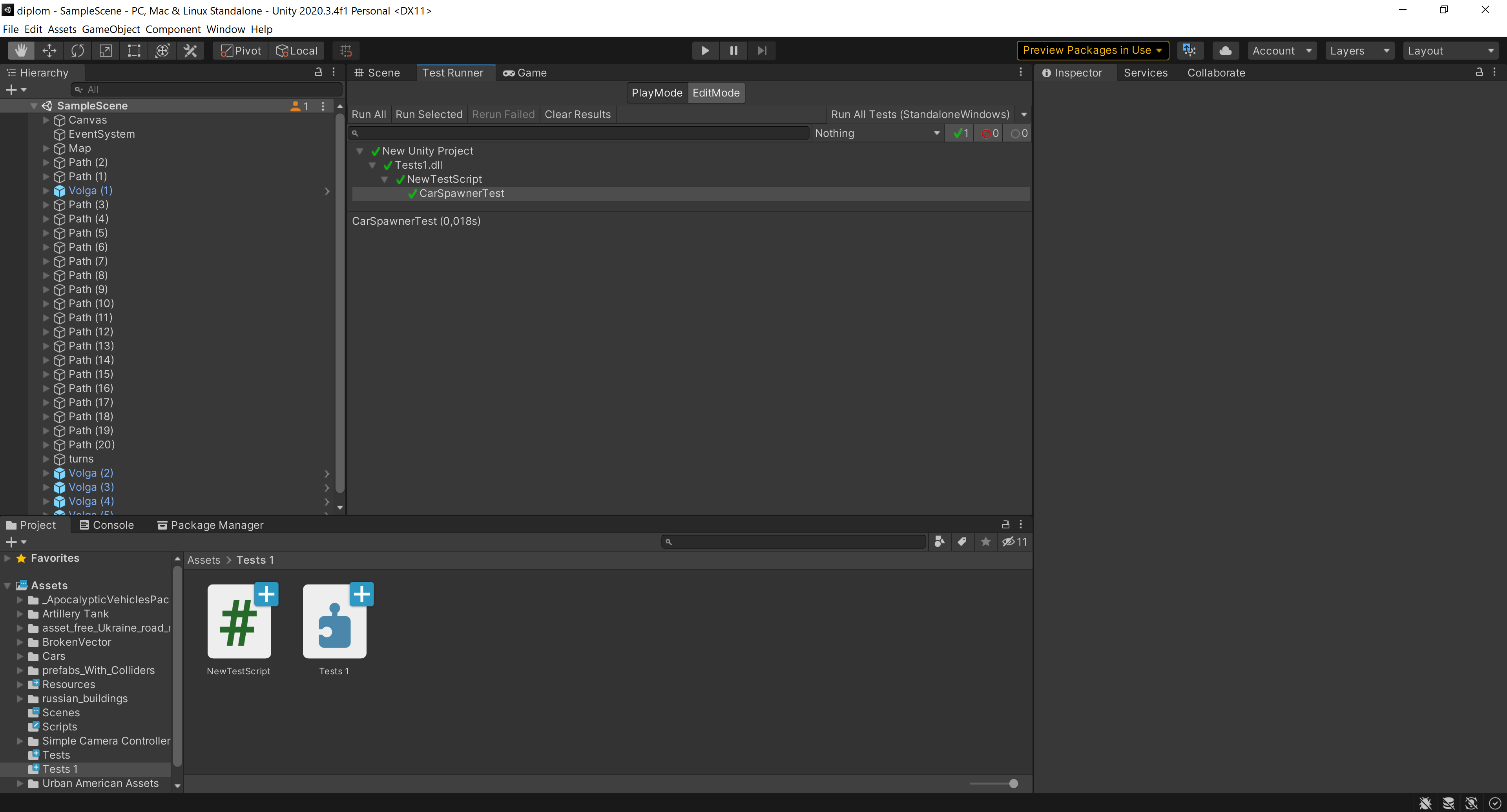


Рисунок 24 ­– Результат 1 для теста модуля «Создание препятствий»

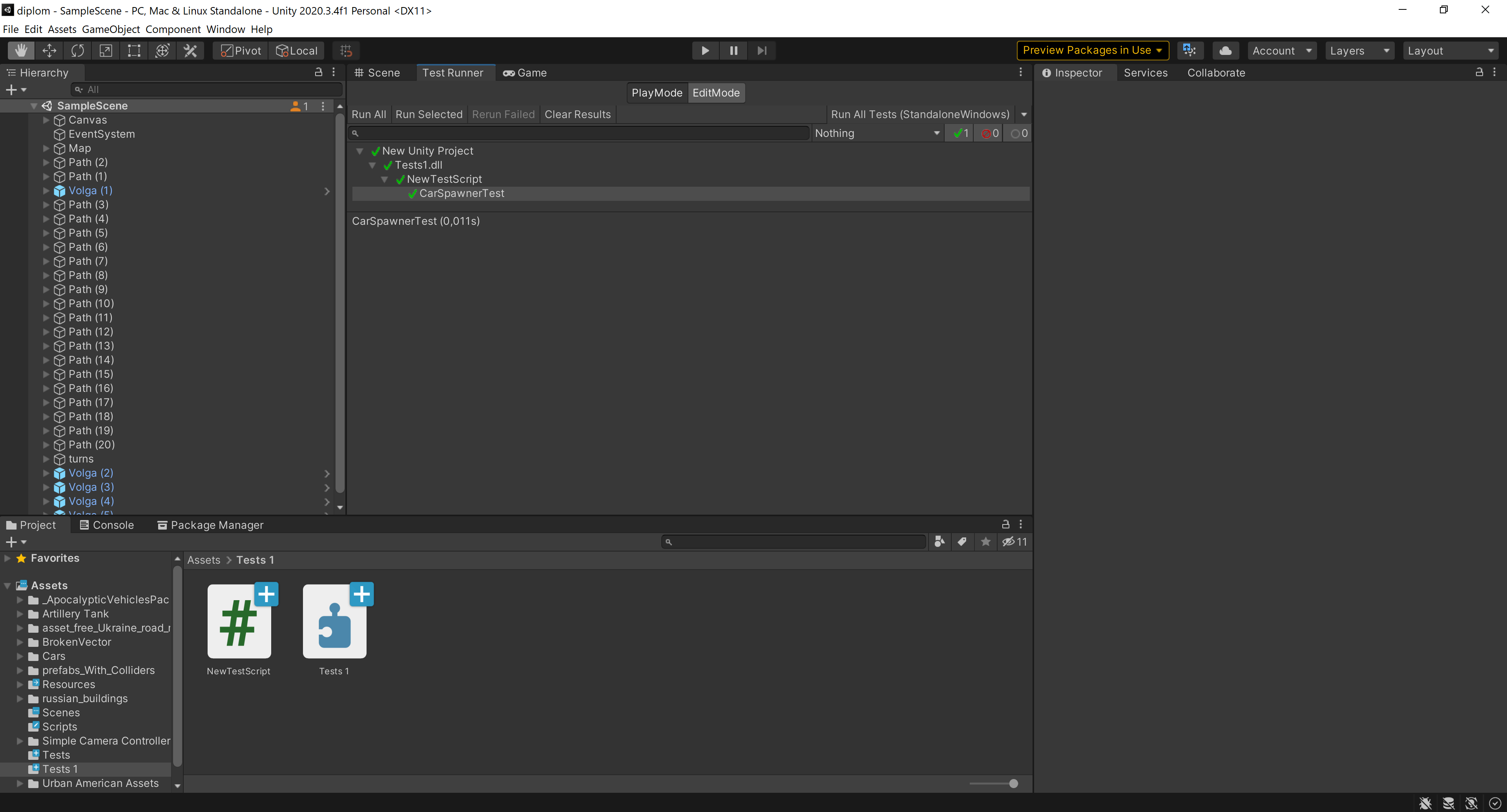


Рисунок 25 ­– Результат 2 для теста модуля «Создание препятствий»

Для тестирования модуля «Сохранение объектов» был разработан следующий скрипт тестирования, результат тестирования представлен на рис.26, 27.

public class NewTestScript

{

    [UnityTest]

    // класс теста

    public IEnumerator saveTest()

    {

        GameObject go = new GameObject();

        // Делаем его объектом класса AutoSave

        AutoSave cs = go.AddComponent<AutoSave>();

        // Вызываем необходимую для теста функцию

        cs.Save();

        yield return null;

    }

}

public class AutoSave : MonoBehaviour

{

    // Создаем массив объектов

    private GameObject[] cars;

    public void Save() {

    // Находим все объекты типа "Car"

        cars = GameObject.FindGameObjectsWithTag ("Car");

    // Инициализируем список для вывода в Json типа "Data"

        List<Data> datas = new List<Data>();

        foreach(GameObject car in cars){

    // Инициализируем объект типа "Data" для хранения информации об одном объекте

            Data data = new Data(car.name, car.transform.position.x, car.transform.position.y, car.transform.position.z, car.transform.rotation.x, car.transform.rotation.y, car.transform.rotation.z);

// Добавляем объект в список

            datas.Add(data);

        }

// Создаем строку JSON из списка типа "Data"

        string json = JsonUtility.ToJson(datas);

// Создаем выходной файл

        System.IO.File.WriteAllText(Application.persistentDataPath + "/data.json", json);

    }

}

[System.Serializable]

// Класс для хранения информации об одном объекте

public class Data{

    public string name;

    public float posX, posY, posZ, rotX, rotY, rotZ;

    public Data(string name, float posX, float posY, float  posZ, float  rotX, float rotY, float rotZ){

        this.name=name;

        this.posX=posX;

        this.posY=posY;

        this.posZ=posZ;

        this.rotX=rotX;

        this.rotY=rotY;

        this.rotZ=rotZ;

    }

}

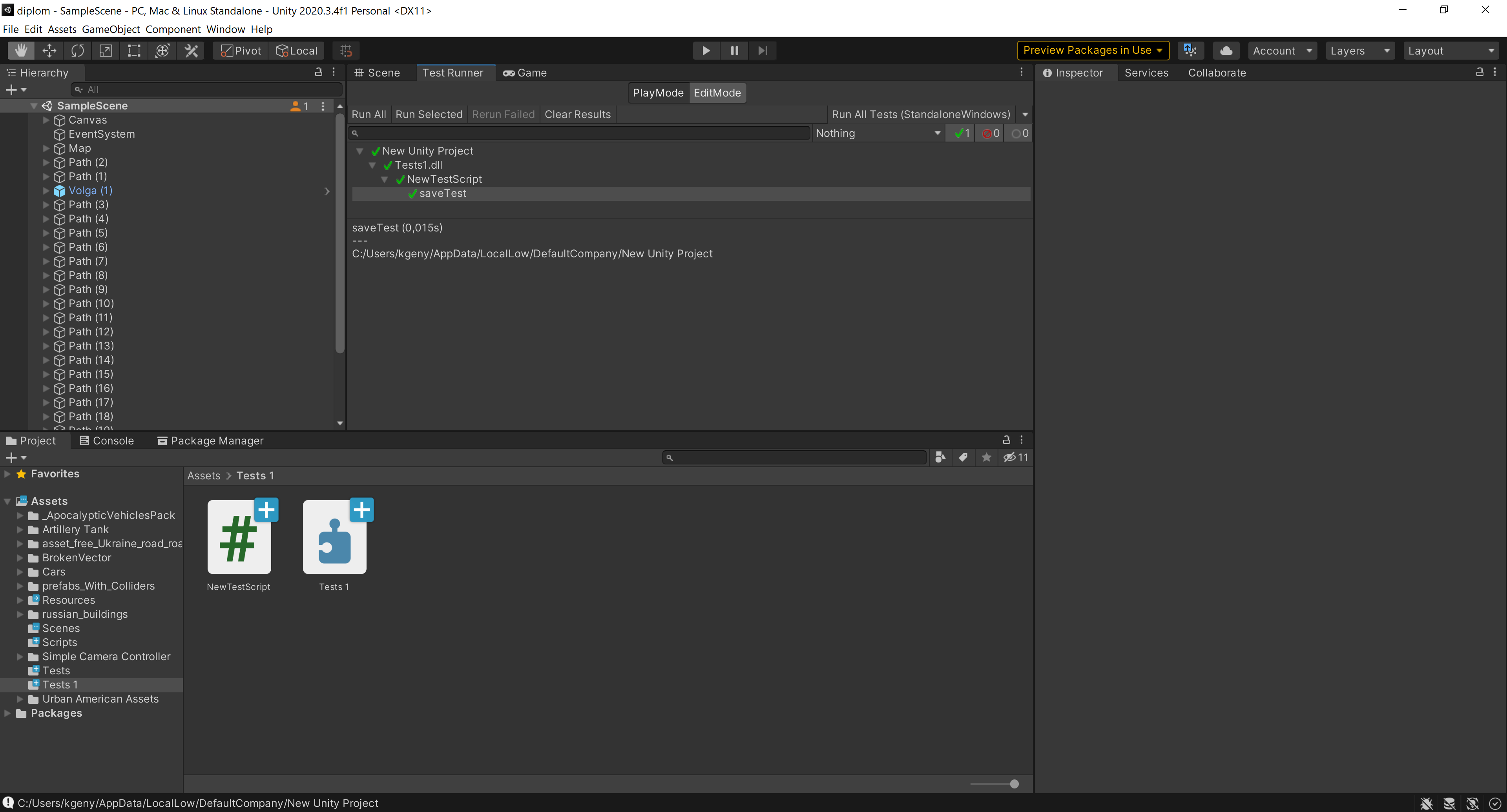


Рисунок 26 ­– Результат 1 для теста модуля «Сохранение объектов»

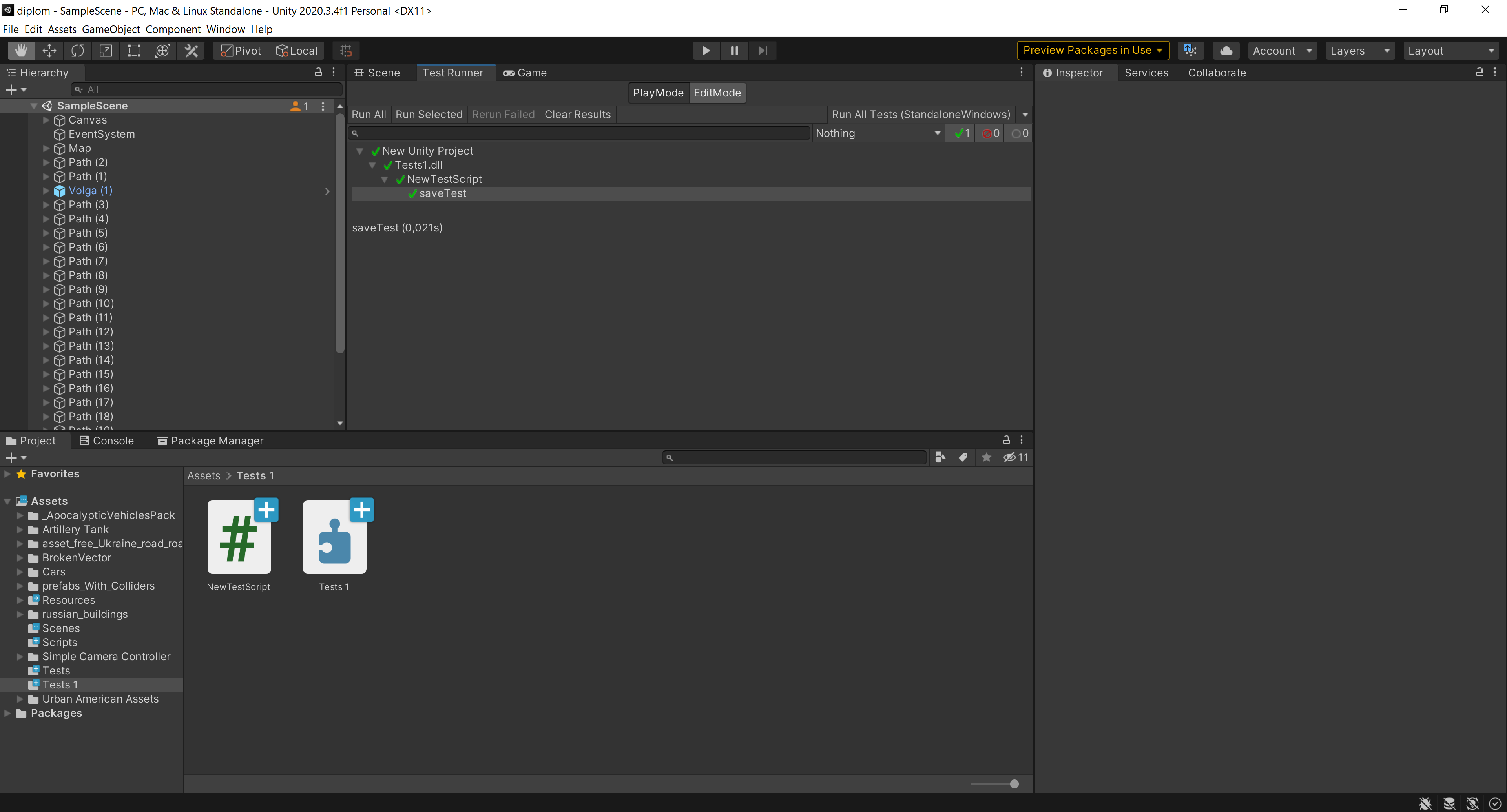


Рисунок 27 ­– Результат 2 для теста модуля «Сохранение объектов»

**Вывод**: в данной лабораторной работе были освоены методики документирования ПС.

**Назначение**: моделирование организации дорожного движения на участке УДС.

**Технические характеристики**: Windows 7 и выше.

**Принцип работы**: Приложение моделирует ОДД с возможностью добавление на участки УДС аварий и дорожных работ.

**Руководство пользователя**

При открытии приложения появляется главное окно, как показано на рис. 28.

Для перемещения по карте используются кнопки W,A,S,D для передвижения вперед, влево, назад и вправо соответственно. Для поворота камеры налево используется кнопка Q, направо ­– E.

Для добавления аварии или дорожной работы нажмите на кнопку «Добавить аварию» или «Добавить дорожную работу» соответственно, для ее перемещения нажмите и удерживайте на аварию или дорожную работу правой кнопкой мыши и перемещайте по модели, для поворота направо или налево при удержании правой кнопкой мыши на кнопки С и Z соответственно.

Для остановки всего моделирования, нажмите на кнопку «Пауза».

Для сохранения текущего состояние модели, нажмите на кнопку «Сохранить».

В окне «Средняя скорость потока» выводится информация о средней скорости потока.

Для выхода из приложения нажмите на красный крестик в правом верхнем углу.