

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Белгородский государственный технологический университет  
им. В.Г. Шухова»

**Институт энергетики, информационных технологий и управляющих систем**

**Кафедра информационных технологий**

**Направление подготовки 09.03.02 Информационные системы и технологии**

**Профиль образовательной программы Информационные системы и технологии**

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**на тему: «Подсистема управления поездом в симуляторе московского метрополитена»**

Студент  
Зав. кафедрой  
Руководитель  
Консультант

Богунов Артем Александрович  
доц. Старченко Денис Николаевич  
доц. Подгорный Николай Николаевич  
доц. Подгорный Николай Николаевич

**К защите допустить**

**Зав. кафедрой** \_\_\_\_\_ /Старченко Д.Н./

« \_\_\_\_\_ » \_\_\_\_\_ 2022 г.

Белгород, 2022

## Содержание

|  |    |
|--|----|
| Содержание .....   | 6  |
| Введение.....  | 8  |
| 1. Предпроектное обследование симуляторов метро .....  | 10 |
| 1.1. Постановка задачи проектирования .....  | 10 |
| 1.2. Описание предметной области .....   | 10 |
| 1.3. Цели и задачи программного средства .....   | 13 |
| 1.3.1. Цель программного средства .....  | 13 |
| 1.3.2. Задачи программного средства.....   | 13 |
| 1.4. Анализ существующих аналогов.....   | 13 |
| 1.4.1. Subway Simulator .....  | 13 |
| 1.4.2. Учебный тренажёр для машиниста .....  | 15 |
| 1.4.3. Hmmsim Metro .....  | 17 |
| 1.5. Актуальность разработки .....   | 18 |
| Вывод к разделу 1.....   | 20 |
| 2. Проектирование подсистемы управления поездом .....  | 22 |
| 2.1. Описание основного цикла работы программного средства.....                                      | 22 |
| 2.2. Методология IDEF .....  | 24 |
| 2.2.1. Диаграмма IDEF0 .....   | 25 |
| 2.3. Проектирование модели данных .....  | 37 |
| 2.3.1. Общий принцип взаимодействия .....  | 37 |
| 2.3.2. Модели системы движения .....   | 40 |
| 2.4. Выбор средства реализации .....   | 52 |
| Вывод к разделу 2.....   | 54 |
| 3. Реализация подсистемы управления поездом в симуляторе машиниста<br>московского метрополитена..... | 55 |
| 3.2. Разработка модуля управления поездом.....   | 57 |
| 3.3. Разработка модуля движения поезда по сплайну .....  | 57 |
| 3.4. Симуляция физики реального состава .....  | 58 |

|            |      |                |         |      |   |  |  |                             |      |        |  |
|------------|------|----------------|---------|------|---|--|--|-----------------------------|------|--------|--|
|            |      |                |         |      | Содержание  |  |  |                             |      |        |  |
|            |      |                |         |      |   |  |  |                             |      |        |  |
| Изм.       | Лист | № докум.       | Подпись | Дата |   |  |  |                             |      |        |  |
| Разраб.    |      | Богун А.А.     |         |      | <b>Подсистема управления<br/>поездом в симуляторе<br/>московского метрополитена</b> |  |  | Лит.                        | Лист | Листов |  |
| Руководит. |      | Подгорный Н.Н. |         |      |   |  |  |                             |      |        |  |
| Консульт.  |      | Подгорный Н.Н. |         |      |   |  |  |                             |      |        |  |
| Н. Контр.  |      | Лазебная Е.А.  |         |      |   |  |  |                             |      |        |  |
| Зав. Каф.  |      | Старченко Д.Н. |         |      |   |  |  |                             |      |        |  |
|            |      |                |         |      |   |  |  |                             |      |        |  |
|            |      |                |         |      |   |  |  | <i>БГТУ им. В.Г. Шухова</i> |      |        |  |
|            |      |                |         |      |   |  |  | <i>ИТ-42</i>                |      |        |  |

|   |    |
|---|----|
| 3.5. Создание инструментов для быстрой конфигурации поездов ..... | 59 |
| 3.6. Написание Unit-тестов .....                                  | 60 |
| Вывод к разделу 3.....  | 62 |
| Заключение .....  | 63 |
| Библиографический список .....                                    | 64 |
| ПРИЛОЖЕНИЕ 1. Техническое задание .....                           | 66 |

|            |      |                |         |      |   |  |  |                               |      |        |  |
|------------|------|----------------|---------|------|---|--|--|-------------------------------|------|--------|--|
|            |      |                |         |      | Содержание  |  |  |                               |      |        |  |
|            |      |                |         |      |   |  |  |                               |      |        |  |
| Изм.       | Лист | № докум.       | Подпись | Дата |   |  |  |                               |      |        |  |
| Разраб.    |      | Богун А.А.     |         |      | Модуль управления персоналом корпоративной информационной системы |  |  | Лит.                          | Лист | Листов |  |
| Руководит. |      | Подгорный В.В. |         |      |   |  |  |                               |      |        |  |
| Консульт.  |      | Подгорный О.В. |         |      |   |  |  | БГТУ им. В.Г. Шухова<br>ИТ-42 |      |        |  |
| Н. Контр.  |      | Лазебная Е.А.  |         |      |   |  |  |                               |      |        |  |
| Зав. Каф.  |      | Старченко Д.Н. |         |      |   |  |  |                               |      |        |  |

## Введение

Тематика метро стала актуальна с 20 века и не теряет своей актуальности до сих пор. Московский метрополитен с каждым годом разрастается, обретая новые ветки и тупиковые станции, которые связывают отдаленные участки города, предоставляя пассажирам быстрый и удобный способ передвижения и радуя свои посетителей интересным и проработанным дизайном и интерьером станций. На сегодняшний день метро является самым популярным способом передвижения в мире, это при том, что далеко не каждый город может похвастаться наличием данного транспорта.

Так же на месте не стоят и отрасли, ответственные за развитие поездостроительной промышленности. Постоянно улучшая комфорт и надежность составов метрополитена, специалисты в данной области превращают метро в один из лучших и надежных видов транспорта в заселенных мегаполисах. Хоть общий принцип движения рельсовых машин не изменен с момента их изобретения, уровень надежности и качества поездов с каждым годом только растет. В современных составах у пассажира есть все, чтобы получить максимальный комфорт от поездки.

И во всей этой предметной области, именно у водителя этих машин, ежедневно перевозящих тысячи пассажиров, одна из самых напряженных и ответственных работ, которую обычные пассажиры зачастую даже не замечают. Расширение функциональности и повышение качественных характеристик составов несомненно влечет за собой так же параллельно увеличивающуюся сложность эксплуатации. Машинист метро не просто нажимает педаль газа и тормоза, двигая состав от станции к станции, он ответственен за состояние всего поезда во время его рейса. Современные поезда имеют очень сложную и разнообразную внутреннюю систему движения и взаимной работы всех его систем.

|            |      |                |         |      |   |  |  |                                       |      |        |
|------------|------|----------------|---------|------|---|--|--|---------------------------------------|------|--------|
|            |      |                |         |      | Введение  |  |  |                                       |      |        |
|            |      |                |         |      |   |  |  |                                       |      |        |
| Изм.       | Лист | № докум.       | Подпись | Дата |   |  |  |                                       |      |        |
| Разраб.    |      | Богун А.А.     |         |      | <b>Подсистема управления<br/>поездом в симуляторе<br/>московского метрополитена</b> |  |  | Лит.                                  | Лист | Листов |
| Руководит. |      | Подгорный Н.Н. |         |      |   |  |  |                                       |      |        |
| Консульт.  |      | Подгорный Н.Н. |         |      |   |  |  | <i>БГТУ им. В.Г. Шухова<br/>ИТ-42</i> |      |        |
| Н. Контр.  |      | Лазебная Е.А.  |         |      |   |  |  |                                       |      |        |
| Зав. Каф.  |      | Старченко Д.Н. |         |      |   |  |  |                                       |      |        |

И именно машинист должен вставить во главу этой системы, взять ответственность за тысячи жизней, которые он перевозит в составе, и проехать по заданному вплоть до секунд расписанию. В метро не бывает аварий в первую очередь не потому, что все линии контролируются системой сигнализаций и светофоров, а как в раз таки из-за высокой профессиональной подготовки личного состава машинистов, проходящих строгий отбор во время подготовки к своим первым рейсам.

## 1. Предпроектное обследование симуляторов метро

### 1.1. Постановка задачи проектирования

Основной задачей проектирования является создание подсистемы, отвечающей за управление алгоритмом движения различных поездов системы программного средства в соответствии с логикой движения реальных прототипов данных поездов, что позволяет пользователю полноценно погрузиться в процесс управления и почувствовать ощущения, приближенные к ощущениям реального машиниста.

Решаемые задачи:

- Исследование предметной области, для изучения особенностей строения и работы различных моделей поездов московского метрополитена;
- Изучение аналогов разрабатываемой системы, выявление их плюсов и минусов, для учета в проектируемой подсистеме;
- Определение целей, задач, актуальности разработки новой системы;
- Формирование функциональных требований к разрабатываемой системе;
- Разработка и построение моделей данных;

Как итог, результатом проектирования должно быть техническое задание на разработку подсистемы движения и управления различными поездами московского метрополитена.

### 1.2. Описание предметной области

В последнее время набирают популярность различные симуляторы. Под симулятором будем понимать любое программное средство, визуально и первоначально выраженное как развлекательное, но несущее в себе обучающую или ознакомительную цель. Огромного успеха симуляторы достигли в сфере образования учеников начальной школы.

|            |      |                |         |      |  |  |                               |        |
|------------|------|----------------|---------|------|--|--|-------------------------------|--------|
|            |      |                |         |      | Предпроектное обследование симуляторов метро                               |  |                               |        |
| Изм.       | Лист | № докум.       | Подпись | Дата |  |  |                               |        |
| Разраб.    |      | Богунов А.А.   |         |      | Подсистема управления<br>поездом в симуляторе<br>московского метрополитена |  | Лит.                          | Лист   |
| Руководит. |      | Подгорный Н.Н. |         |      |  |  |                               | Листов |
| Консульт.  |      | Подгорный Н.Н. |         |      |  |  |                               |        |
| Н. Контр.  |      | Лазебная Е.А.  |         |      |  |  | БГТУ им. В.Г. Шухова<br>ИТ-42 |        |
| Зав. Каф.  |      | Старченко Д.Н. |         |      |  |  |                               |        |

Также набирают популярность компьютерные приложения, симулирующие настоящие, серьезные профессии, такие как: врач, архитектор, менеджер. Это очень удобный способ попробовать на себе новую роль без долгого вхождения в предметную область.

Гораздо удобнее поиграть в симулятор определенной профессии, чем отучиться на нее.

Симуляторы в большинстве своем не дают практических знаний, в этом и заключается их суть – они выполняют ознакомительную функцию, позволяющую человеку опробовать неизвестное для него ремесло и определиться, стоит ли ему в него углубляться уже в реальной жизни. Однако с развитием технологий, появилась возможность использовать симуляторы не только в ознакомительных, но и в обучающих целях. Пилоты самолетов проходят курсы обучения в специальных кабинках для симуляции полета, а в каждой автошколе стоит автомат-симулятор автомобиля, позволяющий ученику отточить навыки вождения по виртуальным дорогам, где цена ошибки – это выговор преподавателя, а не человеческая жизнь.

Программное средство представляет собой симулятор московского метрополитена, где пользователь берет на себя роль машиниста, которому предстоит управлять одним из трех поездов, представляемых как на выбор, так и в результате сюжетных условий.

Игра дает пользователю на выбор 3 режима: обучение, свободный режим и сюжет. В обучении игрок изучает основы управления определенным поездом, а также знакомится с правилами движения в метро. Взаимодействие построено таким образом, что пользователю поочередно предлагается выполнить определенное действие, параллельно рассказывая, зачем оно нужно и за что отвечает. Любые неправильные действия заблокированы. Режим направлен на первичное обучение основам управления поездом, поэтому режим сюжетных эпизодов будет открыт только после успешного прохождения всего курса обучения, который состоит из знакомства с управлением поезда, расписанием

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Предпроектное обследование симуляторов метро | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

движения поездов в метро и в депо, а также с работой с системой сигнализации московского метро.

В свободном режиме игрок может беспрепятственно передвигаться по линии метро на любом поезде.

Этот режим направлен на ознакомление со структурой московского метро и архитектурой его станций. В этом режиме пользователь может начать игровую сессию не только в качестве машиниста, но и как обычный пассажир. В этом случае по обоим линия будут идти,

в соответствии с графиком, поезда, управляемые искусственным интеллектом.

В сюжетном режиме игрок выполняет поочередно открывающиеся, небольшие эпизоды, многие из которых берут основу из реальных ситуаций, произошедших в истории московского метро. Пользователь начинает игровую сессию с заранее определенными условиями, предысторию которых узнает из описания в главном меню и выполняет какую то, четко поставленную цель, чтобы по итогу игровой сессии получить результат, пройден данный сюжетный этап или нет.

Пользователь появляется на сцене в кабине машиниста, в тупике станции. Если режим сюжетный, то игрок сам ставит расписание и предварительно выбирает тупик, в котором появится. В режимах с сюжетом игрок появляется в заранее обусловленном месте с определенным расписанием. Далее идет процесс запуска игроком поезда, установка состава в начальное состояние, выезд по маршруту. Во время процесса езды игрок должен останавливаться на станциях, на которых появляются контролируемые искусственным интеллектом пассажиры, контактирующие с поездом игрока. Пассажиры заходят и выходят на станциях, имитируя поведения реальных людей. Пока игрок едет по станциям московского метро, он либо выполняет определенные сюжетные задачи, либо ездит в свободном режиме, развозя пассажиров в роли машиниста или изучая станции и поезда московского метрополитена в роли обычного пассажира.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Предпроектное обследование симуляторов метро | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |



### 1.3. Цели и задачи программного средства

#### 1.3.1. Цель программного средства

Цель программного средства – реализовать подсистему управления поездом в симуляторе машиниста московского метрополитена.

#### 1.3.2. Задачи программного средства

Для реализации поставленной цели будут решены следующие задачи:

- Реализация считывания ввода пользователя;
- Разработка системы движения модели поезда;
- Разработка подсистем для взаимодействия с вспомогательным функционалом поезда;
- Внедрение подсистемы в симулятор;
- Связь модулей управления с визуальной частью;
- Написание unit-тестов для автоматизации тестирования системы;

### 1.4. Анализ существующих аналогов

Рассмотрим несколько аналогов среди других симуляторов метро. Основным конкурентом является тренажер для машиниста, на котором обучаются в академии машинистов московского метро. Так же были рассмотрены игры Subway Simulator и HmmSim Metro, продающиеся на той же торговой площадке, соответственно имеющие тех же потенциальных пользователей.

#### 1.4.1. Subway Simulator

Subway Simulator – мобильное и десктопное приложение-симулятор метро с простой графикой. Пользователь берет на себя управление выдуманными моделями поездов, основанными на оригинальных прототипах. Данное приложение есть и на мобильных платформах, с которых и было перенесено на ПК, что сказывается на визуальном и инструментальном качестве продукта. Так же данный продукт не сыскал сильной поддержки сообщества, что сильно ограничило ресурсы разработчиков и из-за чего количество контента очень ограничено, но что так же влияет выпуск на мобильных платформах (Рисунок 1.1).

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Предпроектное обследование симуляторов метро | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

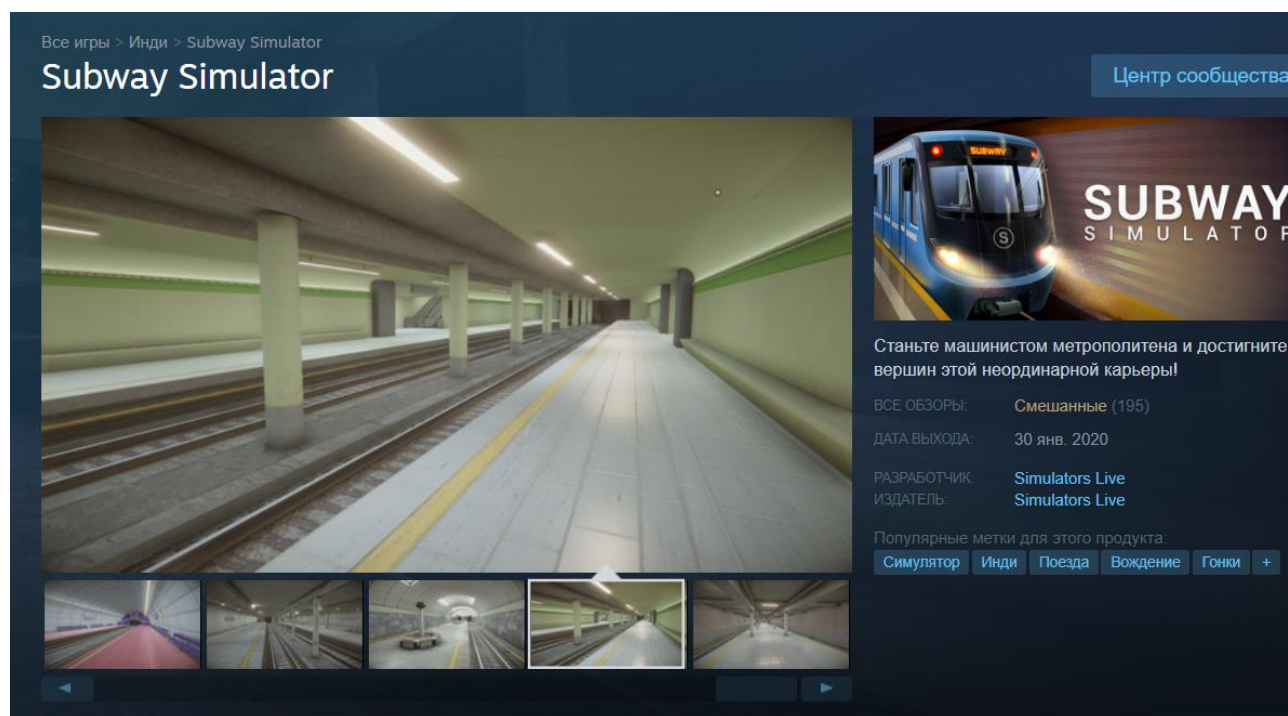


Рисунок 1.1 Начальный экран игры Subway simulator на сайте Steam

Основные особенности рассматриваемого программного продукта Subway Simulator:

Из плюсов можно выделить:

- Интересная система достижений и навыков, позволяющая отслеживать прогресс пользователя.
- Возможность играть с мобильного устройства, что увеличивает количество пользователей.

Из минусов:

- Слабый уровень графики для десктопных платформ, из-за которого понижается средний уровень комфорта и интереса к приложению.
- Нет многих функций, доступных в настоящем поезде, что снижает уровень приложения если рассматривать его с обучающей стороны.
- Ветки метро не имеют реальных аналогов, из-за чего пользователь не чувствует полного погружения и должного реализма.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Предпроектное обследование симуляторов метро | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

- Слабая поддержка от разработчиков приложения. Отсутствуют постоянные обновления и своевременные исправления найденных ошибок понижает качество продукта, а значит и интерес потенциальных пользователей.

#### 1.4.2. Учебный тренажёр для машиниста

Учебный тренажер предназначен для профессиональной подготовки машинистов и помощников машинистов к управлению вагонами метрополитена, в обстановке, максимально приближенной к условиям конкретного участка пути. Использование тренажера возможно в моторвагонных депо, учреждениях среднего и высшего профессионального образования, что и является главным минусом. Так же тренажер требует довольно дорогого обслуживания и высоких вычислительных возможностей машины, с помощью которой будет запущена визуализация движения, что накладывает на пользователя много ограничений. (Рисунок 1.2).

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Предпроектное обследование симуляторов метро | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |



Рисунок 1.2 Панель управления симулятором машиниста

Особенности:

Из плюсов можно выделить:

- Реалистичная приборная панель, полностью повторяющая панель своего прототипа. Это позволяет обучить будущего машиниста механическим движениям и с помощью мышечной памяти выучить расположение всех органов управления.
- Максимальное погружение машиниста. Ввиду того, что тренажеры находятся в университетах машинистов, то студент получает высококачественное обучение азам управления поездом под контролем системы и наставников.

Из минусов:

- Большая стоимость, вследствие чего низкая доступность. Тренажеры стоят только в обучающих учреждениях.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Предпроектное обследование симуляторов метро | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |



- Нет реализации критических ситуаций, ученики ездят по пустому туннелю в штатном режиме.
- Нет симуляции пассажиров. В текущих версиях тренажера отрабатывается только езда и управление поездом.

### 1.4.3. Hmmsim Metro

Hmmsin metro – симулятор машиниста Сеульских железных дорог. В данном приложении реализованы как часть подземных, так и часть наземных станций Сеула, которые достаточно сильно переделаны, и некоторые пути для поездов отличаются от оригинала. доступных для изучения игроком. В игре хорошая графика и поезда сделаны по реальному прототипу (Рисунок 1.3).

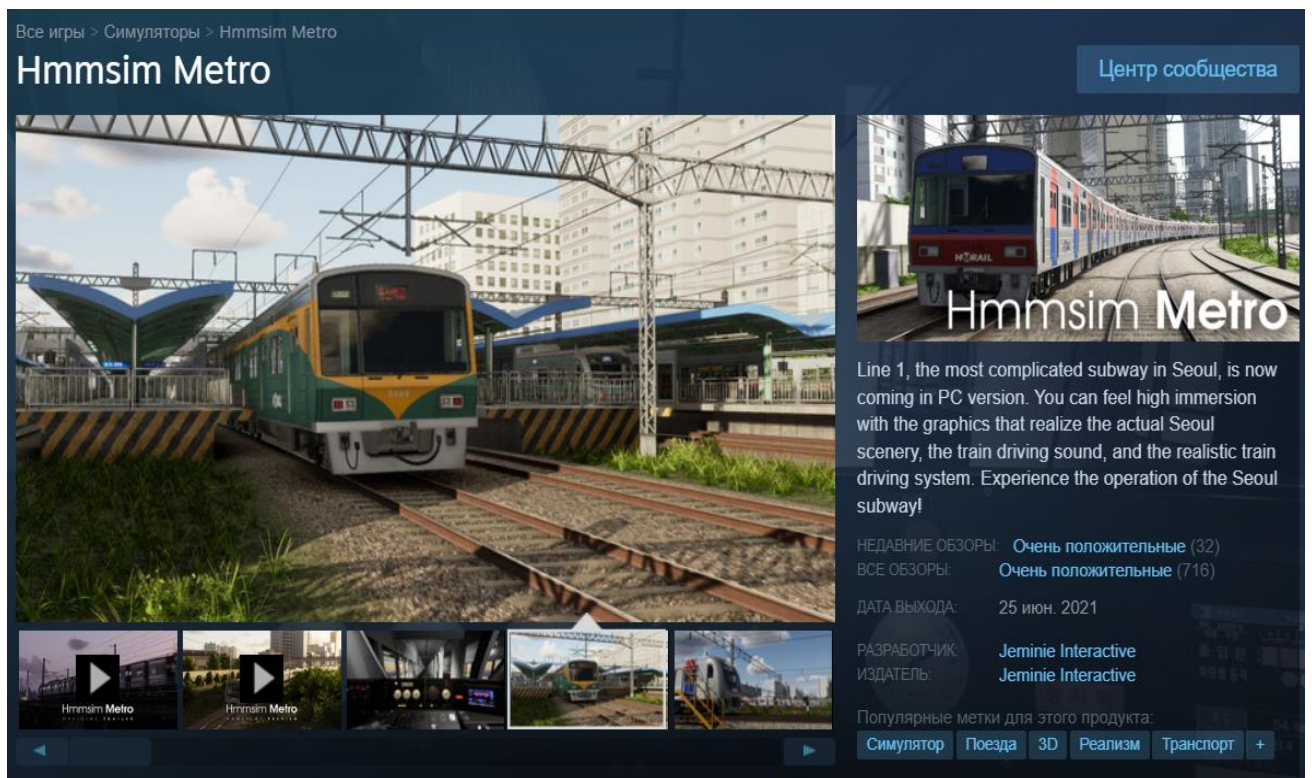


Рисунок 1.3 Начальный экран игры Hmmsin metro на сайте Steam

Можно выделить ряд особенностей:

Из плюсов можно выделить:

- В игре на выбор есть большое количество поездов.
- Хорошая графика, которая радует глаз.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Предпроектное обследование симуляторов метро | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

- Большая локация, которая не заставляет пользователя скучать на заезженных путях.

Из минусов:

- Плохая оптимизация, из-за чего не каждый пользователь сможет опробовать этот симулятор.
- Нет сценариев с экстренными ситуациями, что понижает обучающую составляющую.
- Прототипами служат локации и поезда измененного разработчиками Сеула, что не позволяет эффективно обучаться в симуляторе иностранным пользователям.

### 1.5.Актуальность разработки

Актуальность разрабатываемой системы состоит в том, что аналоги не являются доступным и качественным инструментом для осваивания азов управления поездами московского метро начинающих машинистов.

Большинство имеющихся на рынке аналогов не дают должного погружения и проработки, либо же ценой этому служат большие затраты и сложность в обслуживании. Основываясь на приведенных фактах, можно сделать вывод, что большая часть потенциальной аудитории таких проектов просто лишена возможности опробовать на себе роль машиниста и отточить навыки управления поездом метро ввиду того, что рынок просто не имеет доступных продуктов надлежащего качества.

Интеграция данного программного средства в процесс обучения начинающих машинистов повысит их уровень подготовки с помощью игры в данное приложение, что является одним из самых популярных видов досуга среди молодых людей двадцать первого века.

Совмещение обучающих и развлекательно-медитативных механик программного средства поможет популяризации профессии машиниста, которая

~~еще очень долго будет актуальна, что в свою очередь может повысить средний~~

|  |      |          |         |      |      |
|--|------|----------|---------|------|------|
| Предпроектное обследование симуляторов метро |      |          |         |      | Лист |
| Изм.   | Лист | № докум. | Подпись | Дата |      |

уровень мастерства практикующих специалистов. Так же действия игры происходят в прототипе реального московского метро, что в купе с продажами игры на интернациональных площадках приведет к популяризации русской и культуры русского метро в тематических обществах других стран.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Предпроектное обследование симуляторов метро | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

## Вывод к разделу 1

В ходе предпроектного обследования была исследована предметная область: симуляторы машиниста. Было выявлено, что игровые симуляторы, помимо своей развлекательной составляющей, могут выполнять также обучающую и познавательную роль, что в свою очередь позволяет как заинтересовать новых пользователей, так и улучшить процесс обучения тех, кто заинтересован в тренировочной стороне симулятора.

Проведенный обзор существующих аналогов разрабатываемой системы позволил доказать актуальность разработки. Так разрабатываемый модуль позволит решить спектр общих проблем, выделенных в приведенных аналогичных продуктах, а это:

- Низкий уровень доступности – большинство приведенных выше продуктов имеют либо большую цену, либо высокие требования для качественной эксплуатации.
- Недостаточный уровень покрытия различных экстренных ситуаций для машиниста – все приведенные продукты симулируют лишь управление поездом, но в них нет сценарных режимов, в которых пользователь может получить дополнительные навыки и знания для преодоления нестандартных ситуаций.
- Слабый уровень визуальной симуляции – большинство сравниваемых программных средств не повторяют реалистичных прототипов из московского метрополитена в своем окружении. Это сильно понижает уровень погружения и вовлеченности пользователя, что в свою очередь влияет как на обучающие, так и на познавательные качества приложения.

На основании исследованной предметной области были сформулирована цель разработки и определен список основных задачи, разрабатываемого модуля системы управления поездом московского метрополитена, необходимых к

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Предпроектное обследование симуляторов метро | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |



выполнению.

Результатом проведенного исследования стало разработанное техническое задание на разработку модуля управления поездом метрополитена (Приложение А).

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Предпроектное обследование симуляторов метро | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

## 2.Проектирование подсистемы управления поездом

### 2.1. Описание основного цикла работы программного средства

Взаимодействие пользователя с программным средством происходит при постоянном изменении игрового мира, который может менять свое состояние фиксированное количество раз. Это количество ограничено количеством FPS на конкретном компьютере. Так же операции считывания, обновления состояния и математические вычисления занимают определенное время. Поэтому, для предотвращения рассинхронизации отображения и реального состояния модели, все эти тяжелые операции выполняются по очереди в нативном методе движка Unity, вызываемом один раз за кадр. Это позволяет выстроить четкую последовательность взаимодействия пользователя с приложением. Данная схема изображена на рисунке 2.1.

|            |      |                |         |      |  |   |      |        |
|------------|------|----------------|---------|------|--|---|------|--------|
|            |      |                |         |      | Проектирование системы управления поездом                                  |   |      |        |
| Изм.       | Лист | № докум.       | Подпись | Дата |  |   |      |        |
| Разраб.    |      | Богунов А.А.   |         |      | Подсистема управления<br>поездом в симуляторе<br>московского метрополитена | Лит.  | Лист | Листов |
| Руководит. |      | Подгорный Н.Н  |         |      |  |   |      |        |
| Консульт.  |      | Подгорный Н.Н  |         |      |  | <i>БГТУ им. В.Г. Шухова</i><br><i>ИТ-42</i> |      |        |
| Н. Контр.  |      | Лазебная Е.А.  |         |      |  |   |      |        |
| Зав. Каф.  |      | Старченко Д.Н. |         |      |  |   |      |        |



Рис. 2.1. Цикл работы системы

## 2.2. Методология IDEF

Семейство методов проектирования и нотаций IDEF базируется на методологии ICAM (Integrated Computer Aided Manufacturing) и состоит из следующих представителей:

- IDEF0 —метод и нотация описания бизнес-процессов;
- IDEF1 —метод и нотация описания взаимосвязей между информационными потоками;
- IDEF1X —метод и нотация разработки реляционных баз данных;
- IDEF3 —метод и нотация описания технологических процессов;
- IDEF5 —метод и нотация описания онтологических исследований [6].

Методология и нотация IDEF0 необходима для моделирования систем. Для разрабатываемых проектов IDEF0 помогает определить требования и функции, которые необходимо, в дальнейшем, реализовать. К существующим системам эта методология применяется для анализа функций, которые выполняет система, и отображения механизмов, с помощью которых эти функции выполняются. Результатом применения IDEF0 к некоторой системе является её модель, которая состоит из иерархически упорядоченного набора диаграмм, текста документации. Важные элементы, из которых строятся диаграммы IDEF0, называются работа (блоки) и объекты (дуги). Объекты связывают блоки между собой и показывают взаимодействия и взаимосвязи между ними [7].

Модель может содержать четыре типа диаграмм:

- контекстную диаграмму (в каждой модели может быть только одна контекстная диаграмма);
- диаграммы декомпозиции;
- диаграммы дерева узлов;
- диаграммы только для экспозиции (FEO).

Процесс моделирования начинается с построения контекстной диаграммы. Она показывает, что будет рассматриваться как компоненты системы,

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

а что как внешние факторы.

Контекстная диаграмма является вершиной древовидной структуры диаграмм и представляет собой общее описание системы и ее взаимодействия с внешней средой.

После того, как систему описали в целом, ее разделяют на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, являются диаграммами декомпозиции. Затем проводится декомпозиция каждого большого фрагмента системы на более мелкие и так далее, пока не достигается нужный уровень подробности описания. Обычно на диаграмме представляется от трех до шести функциональных блоков, при этом количество подходящих к одному функциональному блоку (выходящих из одного функционального блока) интерфейсных дуг предполагается не более четырех.

Диаграмма дерева узлов показывает иерархическую зависимость работ, но не взаимосвязи между работами. Диаграмм деревьев узлов может быть в модели сколь угодно много. Диаграммы для экспозиции (FEO) строятся для иллюстрации отдельных фрагментов модели или альтернативной точки зрения, либо для специальных целей. Диаграммы FEO позволяют нарушить любое синтаксическое правило, потому что являются копиями стандартных диаграмм и не включаются в анализ синтаксиса [9].

### **2.2.1.Диаграмма IDEF0**

Для общего описания системы управления персоналом будет использована диаграмма IDEF0, представленная на рисунке 2.2.

Входными данными являются:

Ввод пользователя – это информация, получаемая системой от игрока, посредством перехвата сигналов с манипуляторов ввода. Управление системой

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

возможно, как одной лишь мышкой, так и в сочетании с клавиатурой. Ответственность за обработку ввода на себя берет движок Unity, благодаря чему разработчику остается только написать логику для работы с предоставляемым API. Список используемых методов приведен в таблице 1.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

Таблица 1 используемы методы API системы ввода Unity

|  |   |
|--|---|
| Input.GetButtonDown(string buttonName) | Возвращает true во время кадра, когда пользователь нажимал виртуальную кнопку, указанную в buttonName.      |
| Input.GetButton(string buttonName)     | Возвращает значение true, пока виртуальная кнопка, идентифицированная как buttonName, удерживается нажатой. |
| Input.GetButtonUp(string buttonName)   | Возвращает true для первого кадра, когда пользователь отпускает виртуальную кнопку, указанную в buttonName. |
| Input.anyKeyDown()                     | Возвращает true для первого кадра, в котором пользователь нажимает любую клавишу или кнопку мыши.           |
| GetKeyUp()                             | Возвращает true, пока пользователь удерживает нажатой клавишу, указанную по имени.                          |

Информация смежных систем – это сведения, которые получаются системой извне, от других действующих компонентов игровой сцены.

К ним относится информация от системы сигнализаций, пассажирской системы и т. д. Обработка получаемых данных проводится по push системе, т. е. обработчики смежных систем оповещают нас о изменениях тех данных, которые нам интересны.

Управляющие воздействия представлены правилами движения поездов и сценарными ограничениями.

К правилам движения поездов относятся правила реального мира, так как приложение имитирует работу реального метро. Примером может послужить то, что движение вперед производится только по правому пути либо то, что если поезда врежутся, то дальнейший процесс движения невозможен. Сценарные ограничения – это искусственно введенные правила, заставляющие игрока следовать некому сценарию, заранее продуманному разработчиком, чтобы погрузить пользователя в нестандартную ситуацию или спровоцировать неожиданное усложнение игрового процесса.

В качестве механизмов в проектируемой системе выступают подсистемы управления поездом и контроллеры системы движения. К списку подсистем управления относятся все компоненты, ответственные за передачу сигналов от пользователя во внутреннюю логику взаимодействия системы.

Они отвечают за валидацию, обработку и изменение состояния поезда. Контроллеры системы движений отвечают за преобразование входных и выходных данных, вычисление скоростных величин, применение физической симуляции. Они ответственны за движение поезда по сплайну и имитацию физики поезда. Так же к контроллерам системы движения относятся модули смежных подсистемы, которые имеют влияние на скоростные показатели поезда. Это подсистемы сигнализаций на линиях и в депо.

На выходе мы получаем результат управления. Он состоит из изменений данных подсистемы, представляющих состояние поезда. В начале следующего кадра отрисовки движок обновит состояние поезда и перерисует сцену заново (Рисунок 2.2).

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |



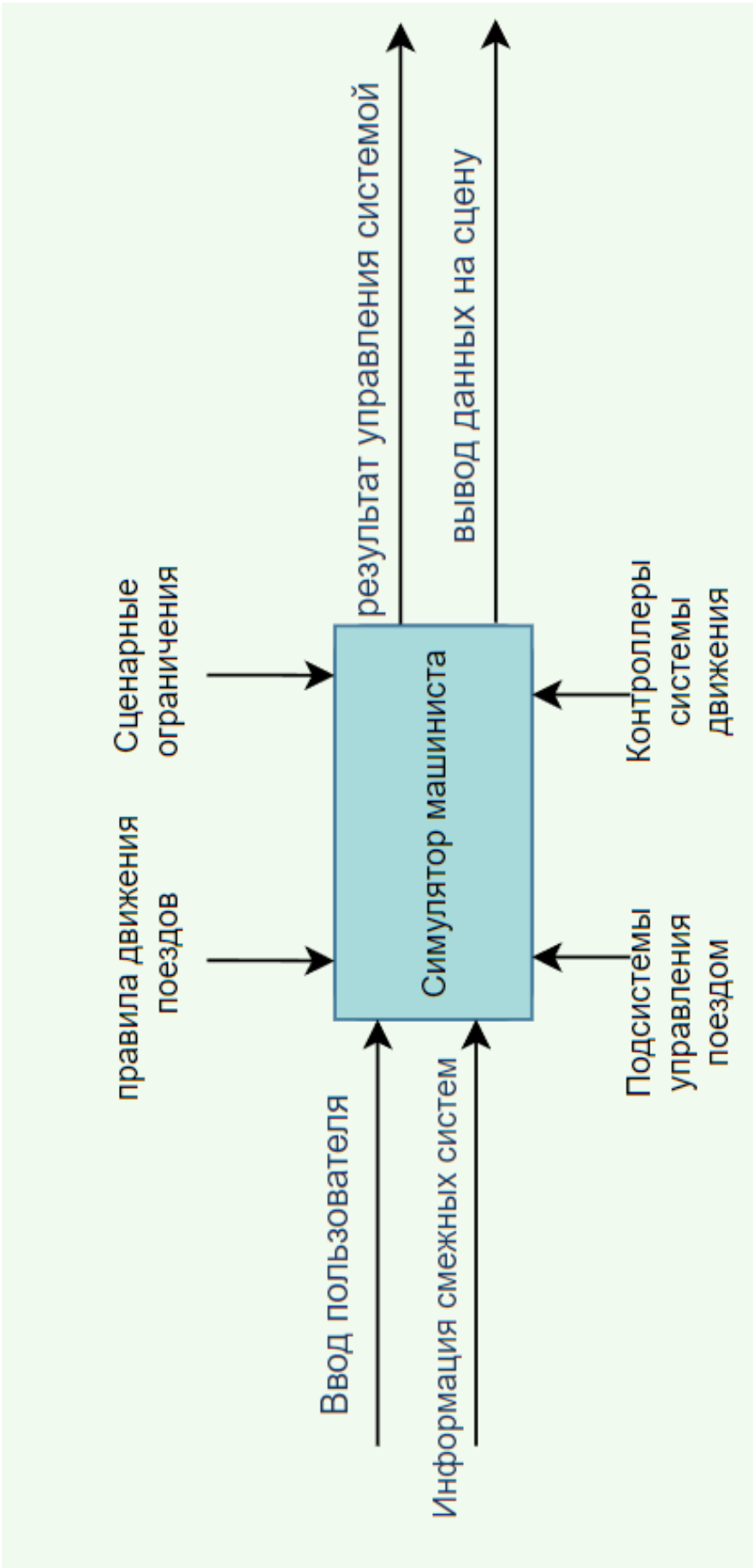


Рисунок 2.2 Контекстная диаграмма IDEF0

Декомпозируем контекстную диаграмму на следующие основные блоки (представлены на рисунке ): Валидация ввода, вспомогательные подсистемы поезда, система движения поезда, система отображения состояния.

- Валидация ввода – блок, отвечающий за обработку ввода пользователя. В данном блоке проверяется, были ли поданы какие-либо сигналы от пользователя за этот кадр отрисовки. Если да, то следует проверить правильность нажатия, допустимо ли запрашиваемое игроком действие.

- Вспомогательные системы поезда – это системы управления поездом. В них входят:

- Работа пассажирских дверей поезда
- Смена типа освещения фар поезда
- Отображение различных показателей на приборах
- Переключение батареи и реверса поезда
- Отрисовка состояния поезда на МФДУ
- Переключение экстренного тормоза
- правление ходом поезда посредством контроллера хода
- Смена камер заднего вида

- Системы движения – блок описывает систему передвижения физической и визуальной модели поезда по определенной линии сплайна, с пересчитываем ее физических свойств и величин.

Часть системы движения и вспомогательных систем так же выполняют роль механизмов в подсистеме. Это обусловлено тем, что состояние поезда зависит от многих условий, которые могут приходить как извне, так и изнутри. Так же ввиду крепкой зависимости внутренних систем разрабатываемого модуля, которая обусловлена предметной областью, многие сущности, выполняющие роль механизмов, сильно связаны с другими вспомогательными системами.

- Система отображения состояния поезда – своевременно изменяет внешний вид модели поезда согласно его состоянию и заранее подготовленным конфигурациям внешнего вида.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

Таким образом, можно сказать, что система управления берет начало с валидации пользовательского ввода, поскольку приложение запускается на ПК и требует от пользователя клавиатуру и мышь или джойстик, следует во время выполнения анализировать, имеет ли введенный сигнал переопределенное значение, а также доступен ли он в текущей игровой ситуации.

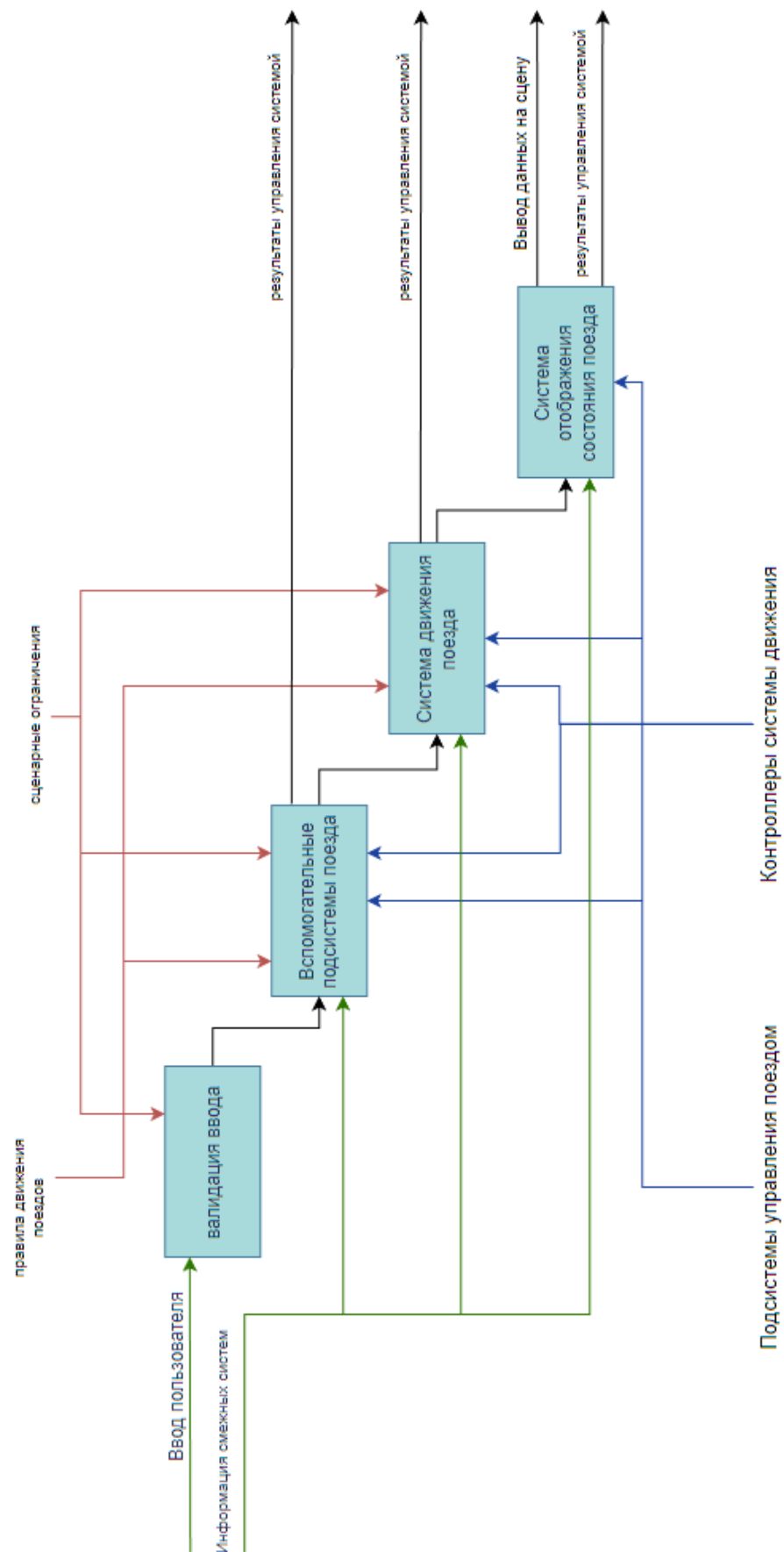


Рисунок 2.3 Декомпозиция контекстной диаграммы IDEF0

Процесс «Валидация ввода» двумя блоками. Сначала движок Unity получает необработанные сигналы от устройств ввода и расшифровывает, предоставляя к ним доступ в виде удобного API. Сущности системы валидации фильтруют только интересные системе сигналы ввода, к примеру нажатия на какую-либо кнопку в поезде, после чего распределяют эти данные по заинтересованным смежным системам, посредством подписок на изменение или прямого вызова методов и конструкторов (Рисунок 2.4).

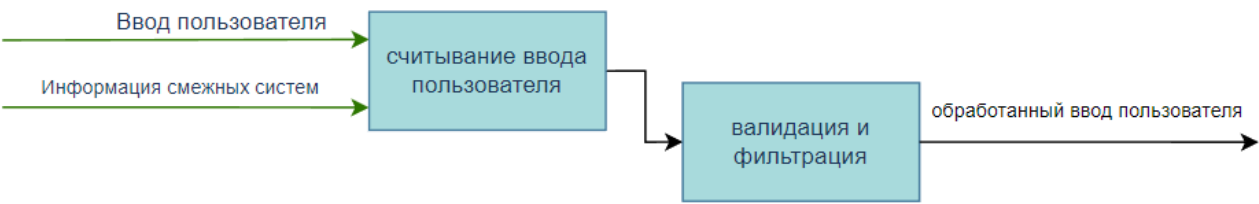


Рисунок 2.4 Декомпозиция функционального блока: валидация ввода

Процесс «Вспомогательные системы поезда» (рисунке ) отвечает за работу подсистем поезда, выполняющих вспомогательные роли, которые задействуются в определенные моменты.

На вход приходят обработанные данные из блока валидации ввода, получаемые правильными сущностями, общую логику работы можно разделить на такие составляющие:

Сначала производится валидация запрашиваемого изменения состояния, посредством сверки желаемого с текущим. Если переход в новое состояние возможен, то сущности меняют состояние поезда и оповещают все, подписанные на них подсистемы и сущности (Рисунок 2.5).

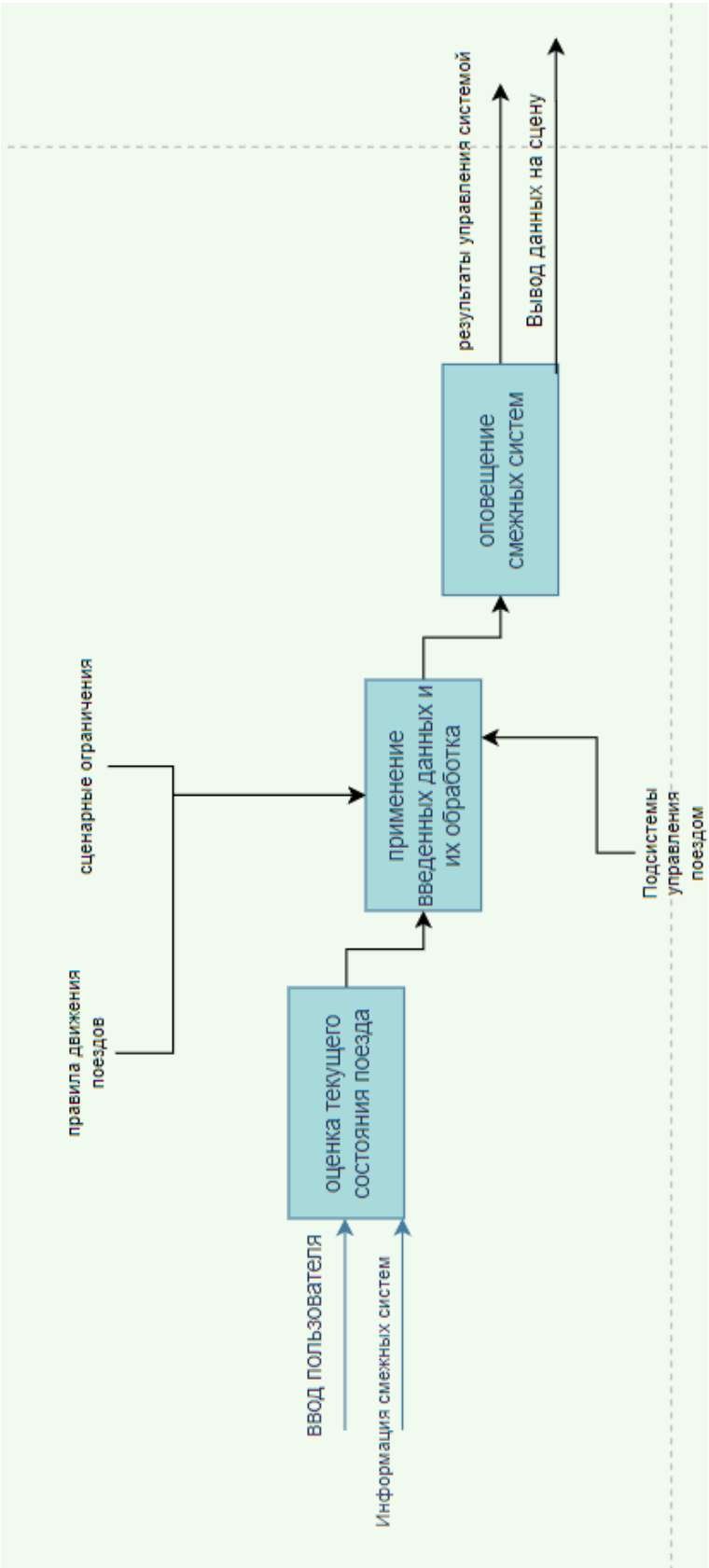


Рисунок 2.5 Декомпозиция функционального блока:  
Вспомогательные системы

Процесс «Система движения поезда» (рисунок ) состоит из таких процессов как валидация запрашиваемых действий. На вход процесса приходит информация от смежных систем и прошедший процесс «Валидация ввода» сигнал от пользователя и смежных систем. После применения изменений и смены текущего состояния, в конце кадра, после всей программной логики и перед вызовом отрисовки, производится пересчет физических величин поезда.

Частично рассматриваемый процесс выполняет действие механизма в «Вспомогательных системах», ввиду их сильной зависимости друг от друга.

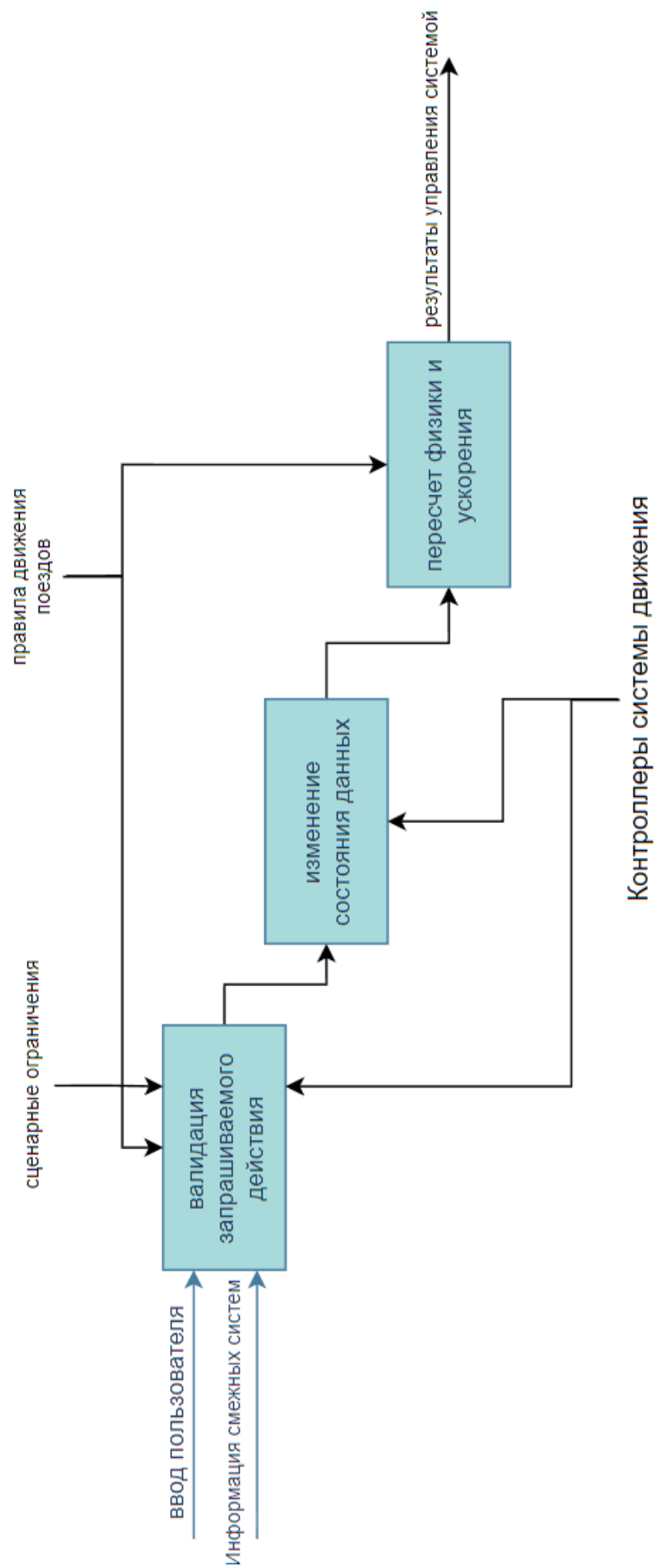


Рисунок 2.6 Декомпозиция функционального блока: Система движения поезда



## 2.3. Проектирование модели данных

### Язык моделирования UML

Унифицированный язык моделирования (UML) был создан для создания общего, семантически и синтаксически богатого языка визуального моделирования для архитектуры, проектирования и реализации сложных программных систем как структурно, так и поведенчески. UML имеет приложения, выходящие за рамки разработки программного обеспечения, такие как поток процессов в производстве.

Он аналогичен чертежам, используемым в других областях, и состоит из различных типов диаграмм. В совокупности диаграммы UML описывают границы, структуру и поведение системы и объектов внутри нее.

UML не является языком программирования, но существуют инструменты, которые можно использовать для создания кода на различных языках с использованием диаграмм UML. UML имеет прямое отношение к объектно-ориентированному анализу и проектированию.

#### 2.3.1.Общий принцип взаимодействия

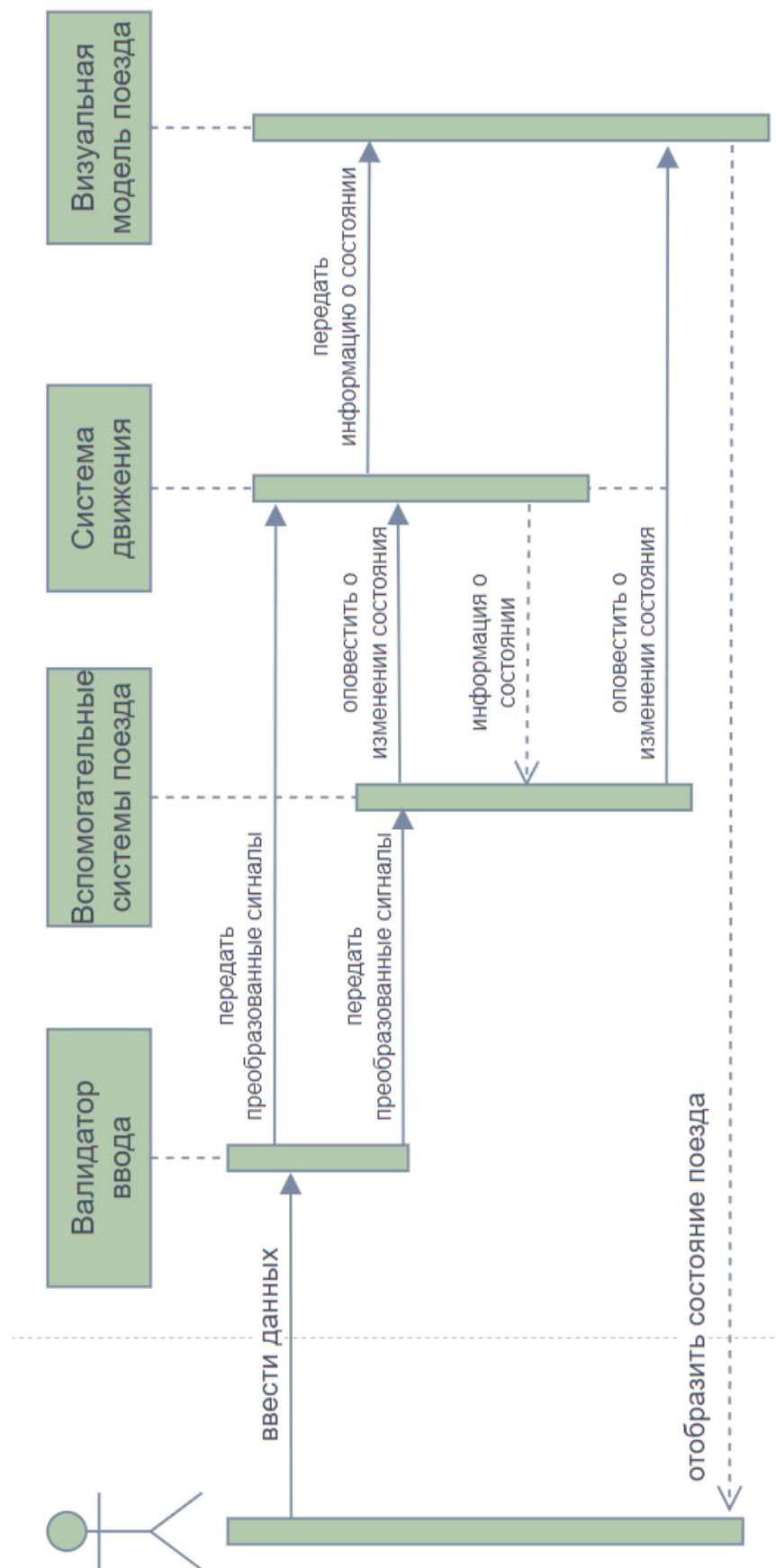
Работа всей подсистемы в общем случае продемонстрирована с помощью диаграммы последовательности.

Диаграмма последовательности или диаграмма последовательности системы показывает взаимодействия процессов, организованные во временной последовательности в области разработки программного обеспечения. Она изображает задействованные процессы и последовательность сообщений, которыми обмениваются процессы, необходимые для выполнения функциональности. Диаграммы последовательности обычно связаны с реализациями вариантов использования в модели архитектурного представления 4+1 разрабатываемой системы. Диаграммы последовательности иногда называют диаграммами событий или сценариями событий.

Пользователь управляет системой через ввод данных, представляемый взаимодействием с различными органами управления поезда. Предварительно эти данные проходят этап валидации, на котором отсекается ненужная

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

информация. Далее сигнал от пользователя идет на вспомогательные системы поезда и на систему движения, внутри которых проходит повторную валидацию уже на корректность запрашиваемых действий. В случае успеха эти данные применяются системами, далее общие изменения сопоставляются между системами и поезд изменяет свое состояние. На основе измененного состояния происходит пересчет его физических характеристик. В конце кадра, во время вызова отрисовки, так же меняется и визуальная составляющая. Отрисованные изменения воспринимаются пользователем, и он продолжает взаимодействие с уже измененной системой.



### 2.3.2. Модели системы движения

Работа системы движения предоставлена на рисунках 2.8, 2.9 и 2.10 в нотации UML для демонстрации связей классов и сущностей.

Система начинает работу с класса TrainCreator, который создает на сцене заранее собранную модель поезда. В параметрах передается модель поезда, его начальное положение и сплайн, по которому поезд начнет свой путь. После создания модели в ход вступает класс TrainInstaller, который производит внутреннюю настройку систем поезда. Он инициализирует все системы, пробрасывает сущностям требуемые им зависимости и выставляет начальное состояние поезда.

В группе «View» находятся классы, отвечающие за визуальную часть системы. Они используются для изменения положения контроллера хода – объекта, отвечающего за управление ускорением поезда.

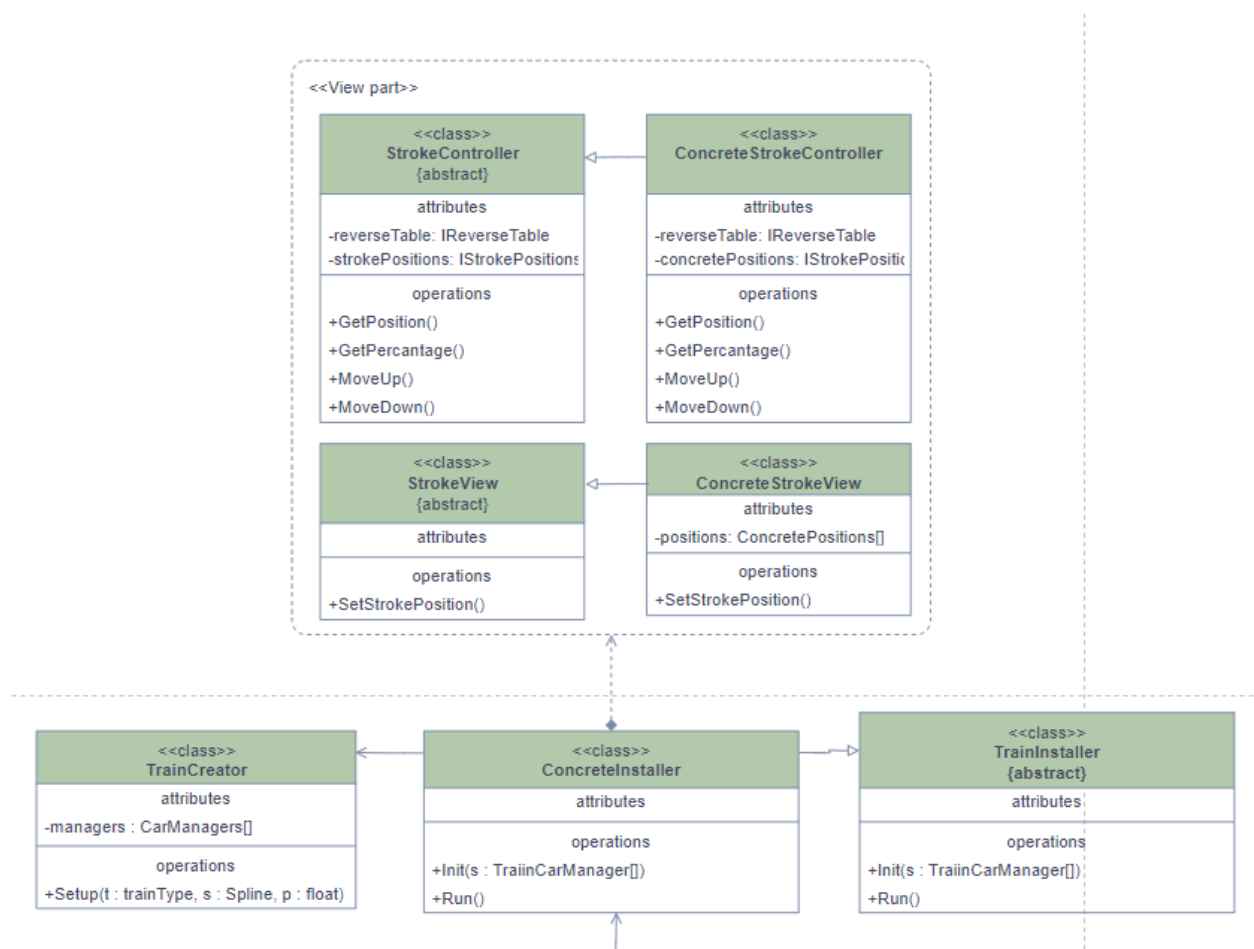


Рисунок 2.8 UML схема классов 1

В результате работы TrainInstaller'a происходит инициализация класса TrainMover, агрегирующего в себе всю логику движения поезда. Основной агрегируемой сущностью служит ISpeedReceiver, который берет на себя ответственность за перерасчёт скорости и ускорения поезда. Он в свою очередь агрегирует 3 основные сущности для движения поезда:

- Accelerator: просчитывает первоначальное ускорение поезда, исходя из входящих от пользователя данных.
- EffecentyLoss: потеря эффективности ускорения, зависящая от максимального ускорения для текущей скорости поезда, текущей мощности хода и прочих факторов.
- SlopeModifier: сущность, ответственная за применение физики уклонов поезда.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

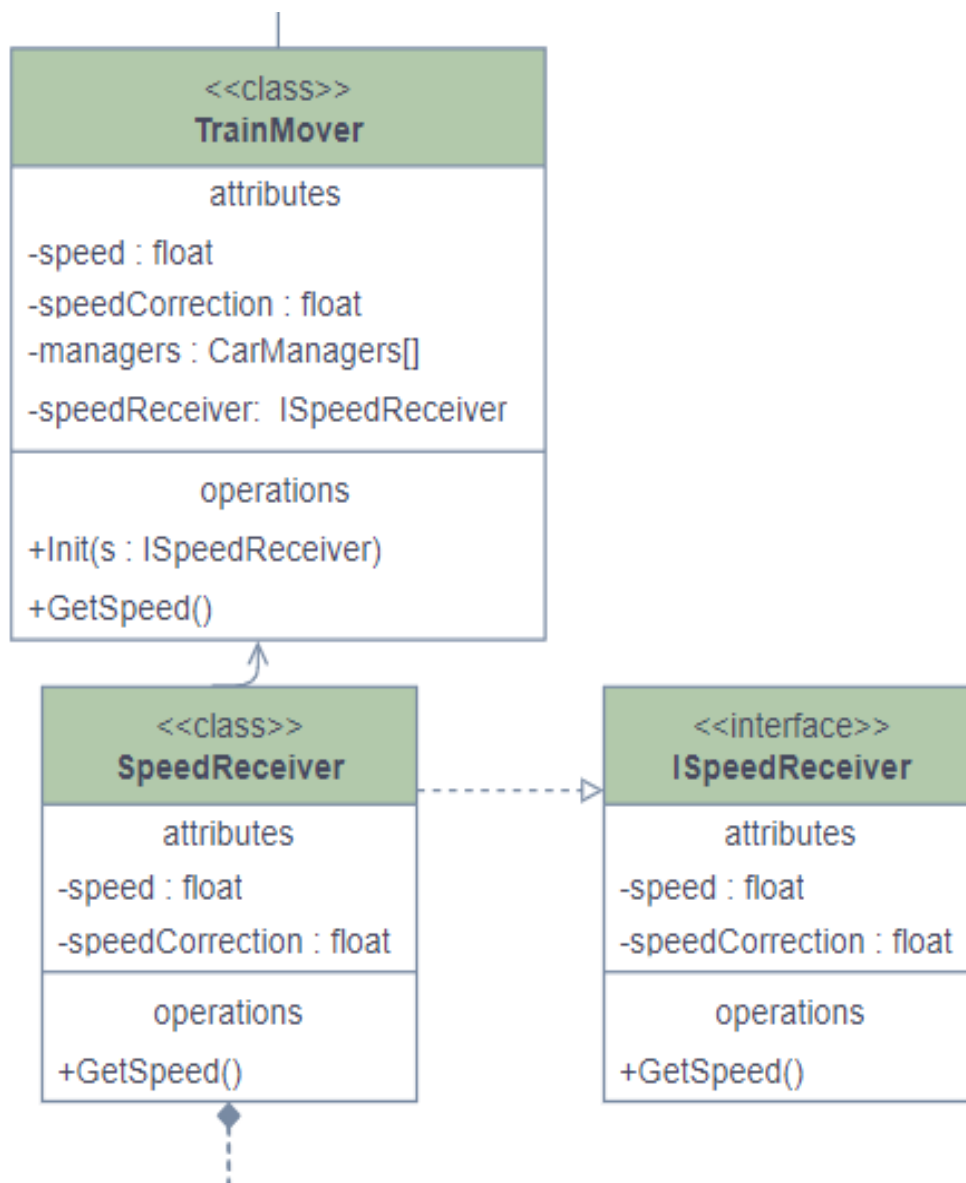


Рисунок **Ошибка! Источник ссылки не найден.**9 UML  
схема классов часть 2

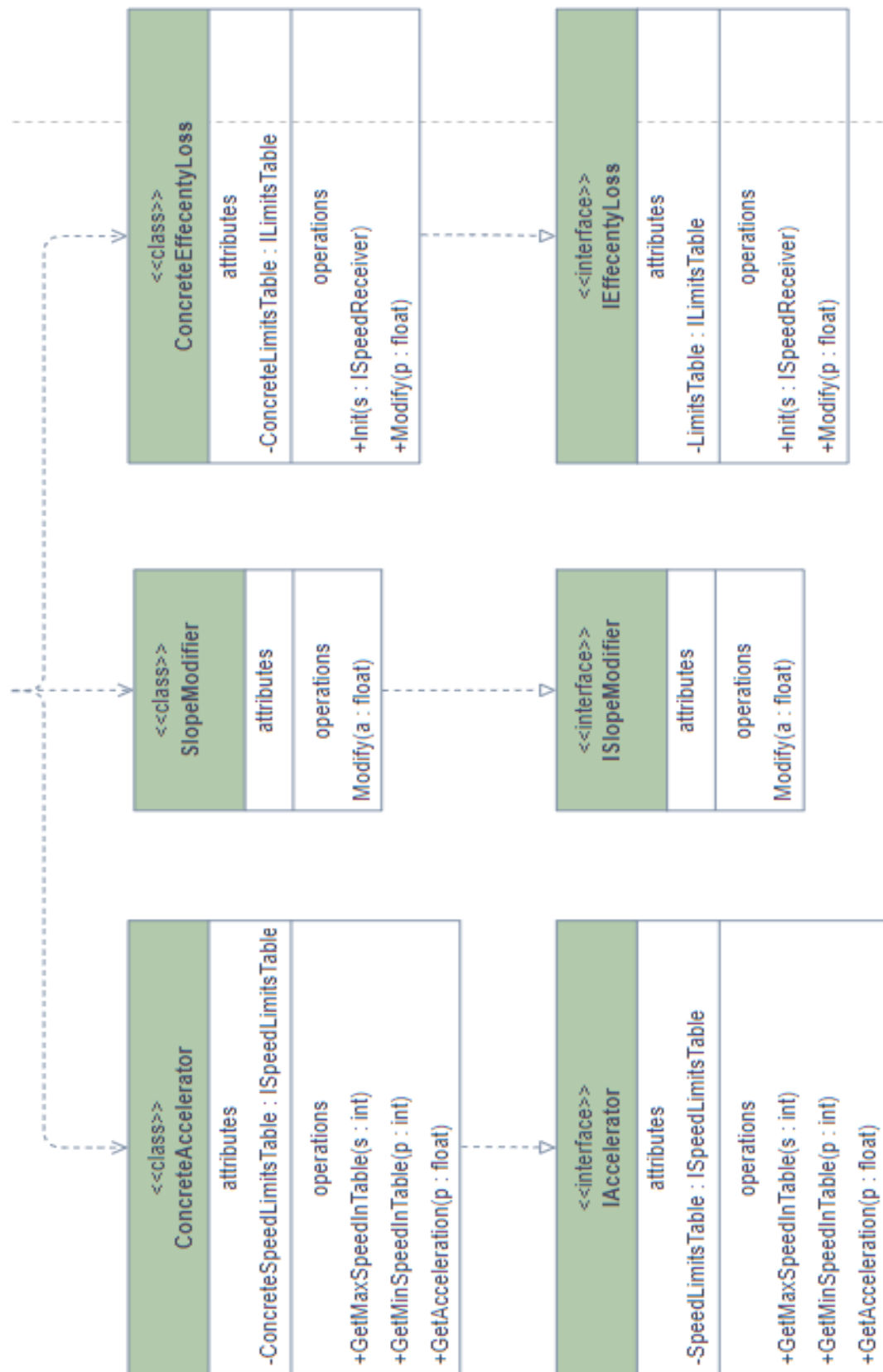


Рисунок 2.10 UML схема классов часть 3

С рассмотренной выше системой движения параллельно начинает свою инициализацию и система компонентов, включающая в себя список сущностей, унаследованных от пустого интерфейса `IComponent` для будущей фильтрации на уровне рефлексии. Для каждой главной сущности предусмотрена своя решаемая задача из общего списка:

- Работа пассажирских дверей поезда

За нее отвечает `DoorsComponent`, в котором содержится интерфейс и общая логика. Для доступа к моделям дверей поезда и их анимация используется класс `DoorsAgregator` (рисунок 2.11).

- Смена типа освещения фар поезда

Для смены освещения задних и передних фар используется `HeadlightsComponent`, имеющий ссылку на `HeadlightsAgregator` – хранилище ссылок на модели фар поезда. `HeadlightsAgregator` в свою очередь подгружает из каталога ресурсов игры файл конфигурации, заполняемый человеком, настраивавшим поезд. Файл конфигураций содержит различные настройки фар, такие как цвет, яркость, материалы (рисунок 2.11).

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |



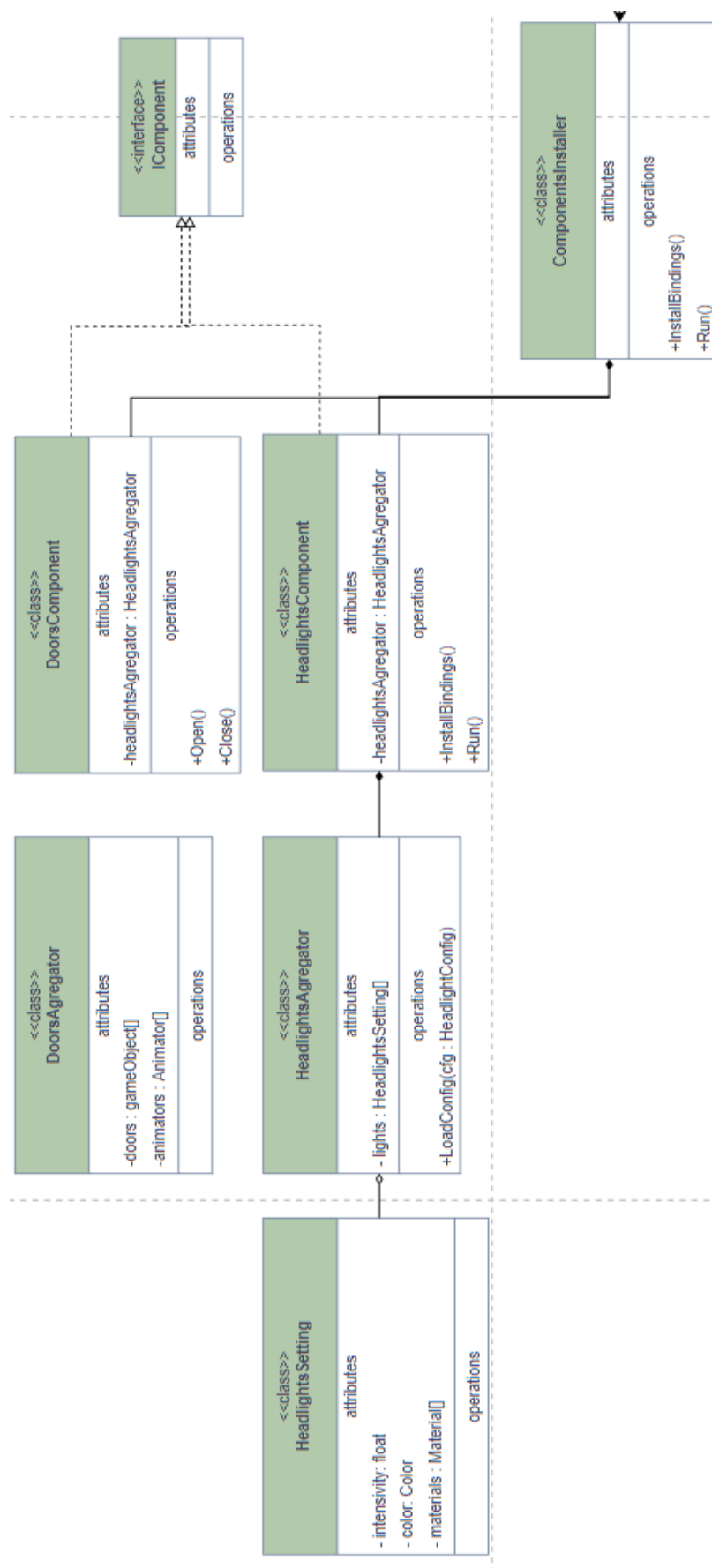


Рисунок 2.11 UML схема классов компонентов

- Отображение различных показателей на приборах

Отдельным компонентом является IndicatorsComponent, который агрегирует в себе сущности для вычисления значений на различных приборах поезда, таких как вольтметр, спидометр, барометр. Данная сущность имеет много зависимостей, так как должна получать информацию о многих системах, но поскольку в подсистеме используется IoC-контейнер, процесс создания сущности сильно упрощается (рисунок 2.12).

- Переключение батареи

Батарея поезда – устройство, отвечающее за питание его вагонов. Именно с помощью батареи машинист включает питание, приводящее поезд в движение, регулирует освещение в поезде и т. д. Эта сущность является главным условием при валидации действий по управлению поездом. За батарею отвечает BatteryComponent, получающий данные из файла конфигураций (рисунок 2.12).

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

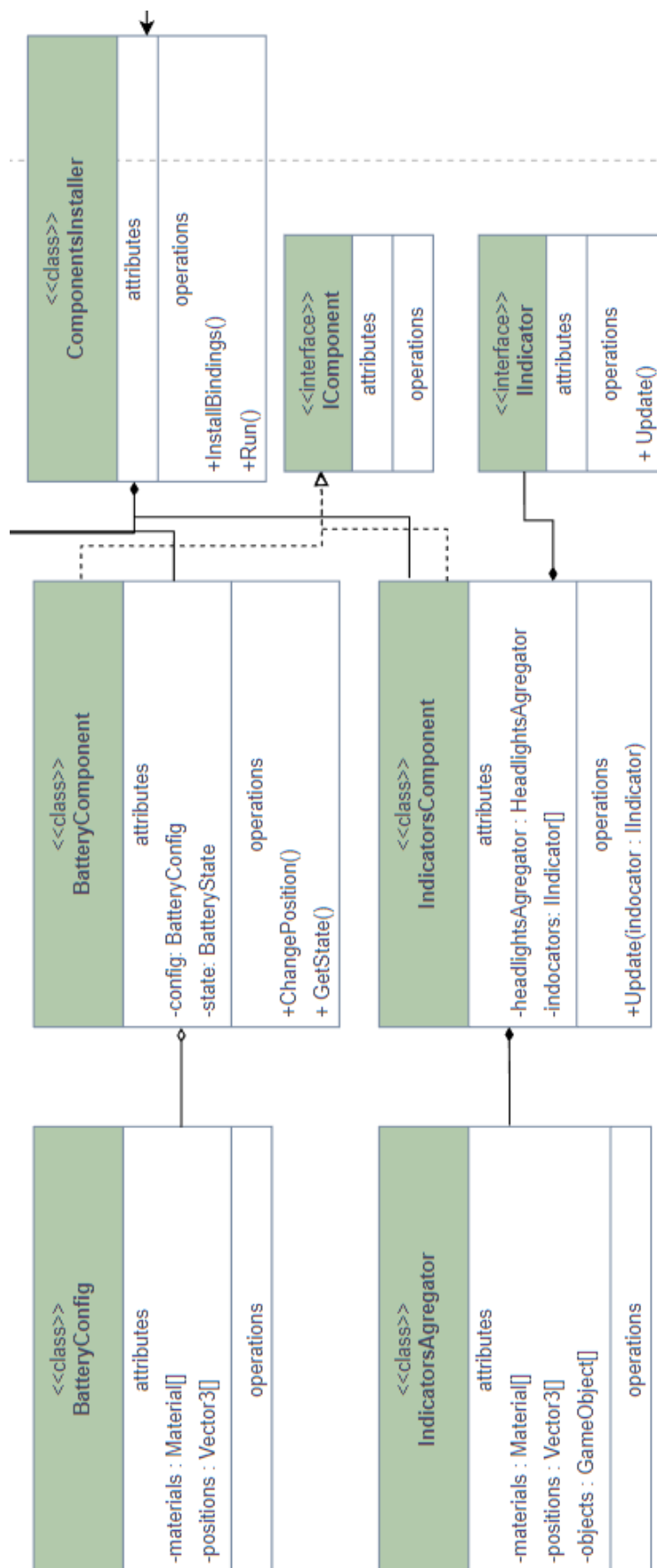


Рисунок 2.12 UML схема классов компонентов Часть 2

- Переключение реверса поезда

Если батарея отвечает за работу всего поезда, то контроллер реверса отвечает за логику всей системы движения, поскольку именно он указывает, в каком движении ехать поезду. Поскольку его устройство разнится от поезда к поезду, то при его проектировании было приложено максимальное усилие на абстракцию (рисунок 2.13).

- Отрисовка состояния поезда на МФДУ

Многофункциональный дисплей управления располагается на панели управления поездом и отражает текущее состояние всех вагонов. Для отрисовки всей графики используется конкретный класс, поскольку у разных поездов эта панель совершенно разная. В сущность `MFDUComponent` внедряются все необходимые зависимости, создаются определенные компоненты и происходит отрисовка всей графики (рисунок 2.13).

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

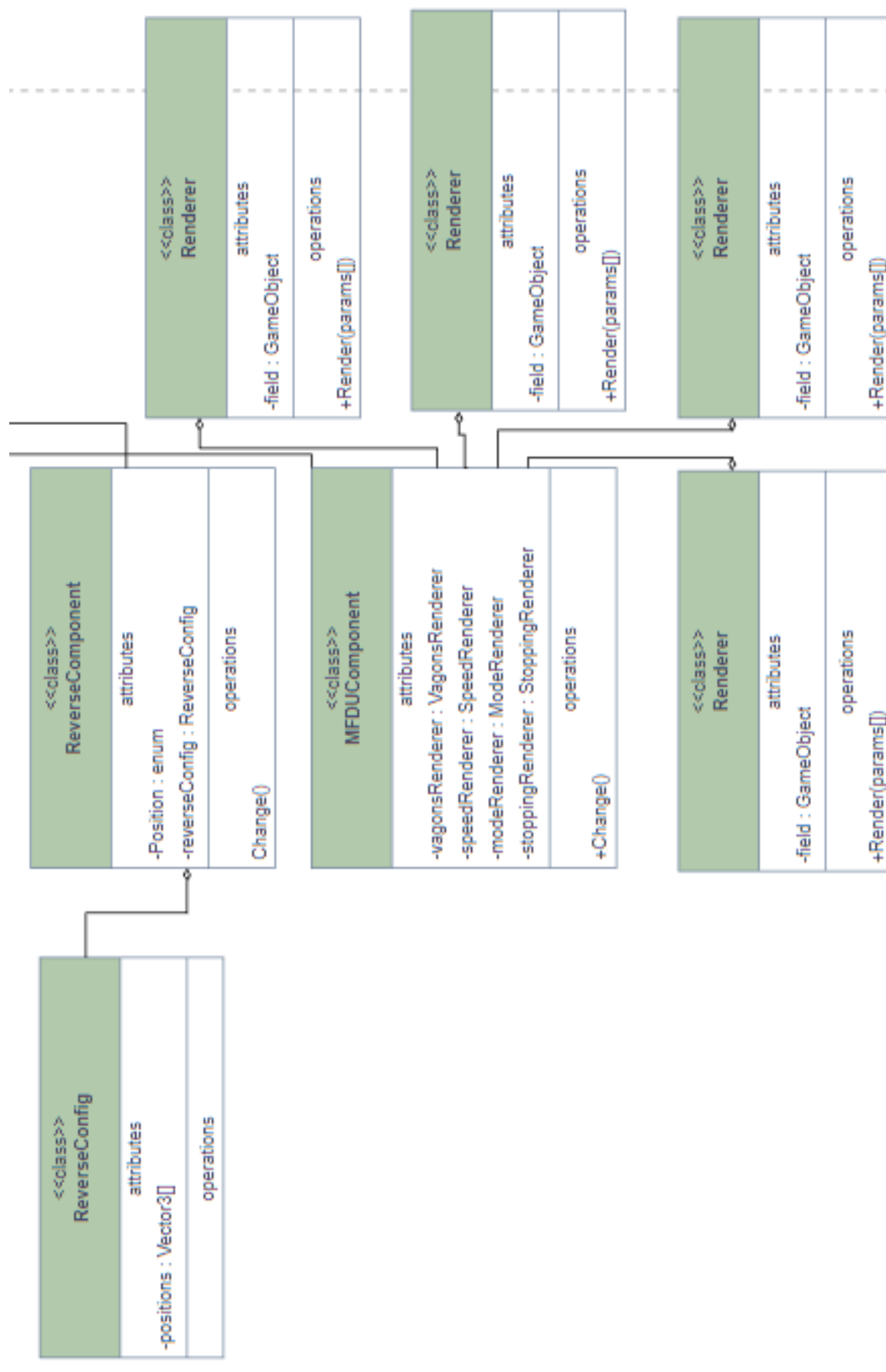


Рисунок 2.13 UML схема классов компонентов Часть 3

- Работа торможения

В поезде предусмотрено несколько механизмов экстренного торможения, которые срабатывают либо в различных критических ситуациях, либо посредством действий машиниста. Система экстренного тормоза блокирует некоторые смежные компоненты, оповещая их о своем изменении. Интерфейсом взаимодействия с компонентом является класс `StoppingComponent`, который содержит в себе `EmergencyBreak` – отвечающий за экстренное торможение, вызванное критической ситуацией, `PanelBreak` – отвечает за экстренное торможение, спровоцированное нажатием кнопки в кабине машиниста и `EmergencyPick` – за переключатель экстренного торможения в кабине (рисунок 2.14).

- Смена камер заднего вида

Отрисовка изображения с камер на сцене происходит на небольшую панель в кабине машиниста. Экран может отображать вид камеры заднего вида или внутри вагона пассажиров. Взаимодействие происходит с сущностью `OutCamerasComponent` (рисунок 2.14).

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

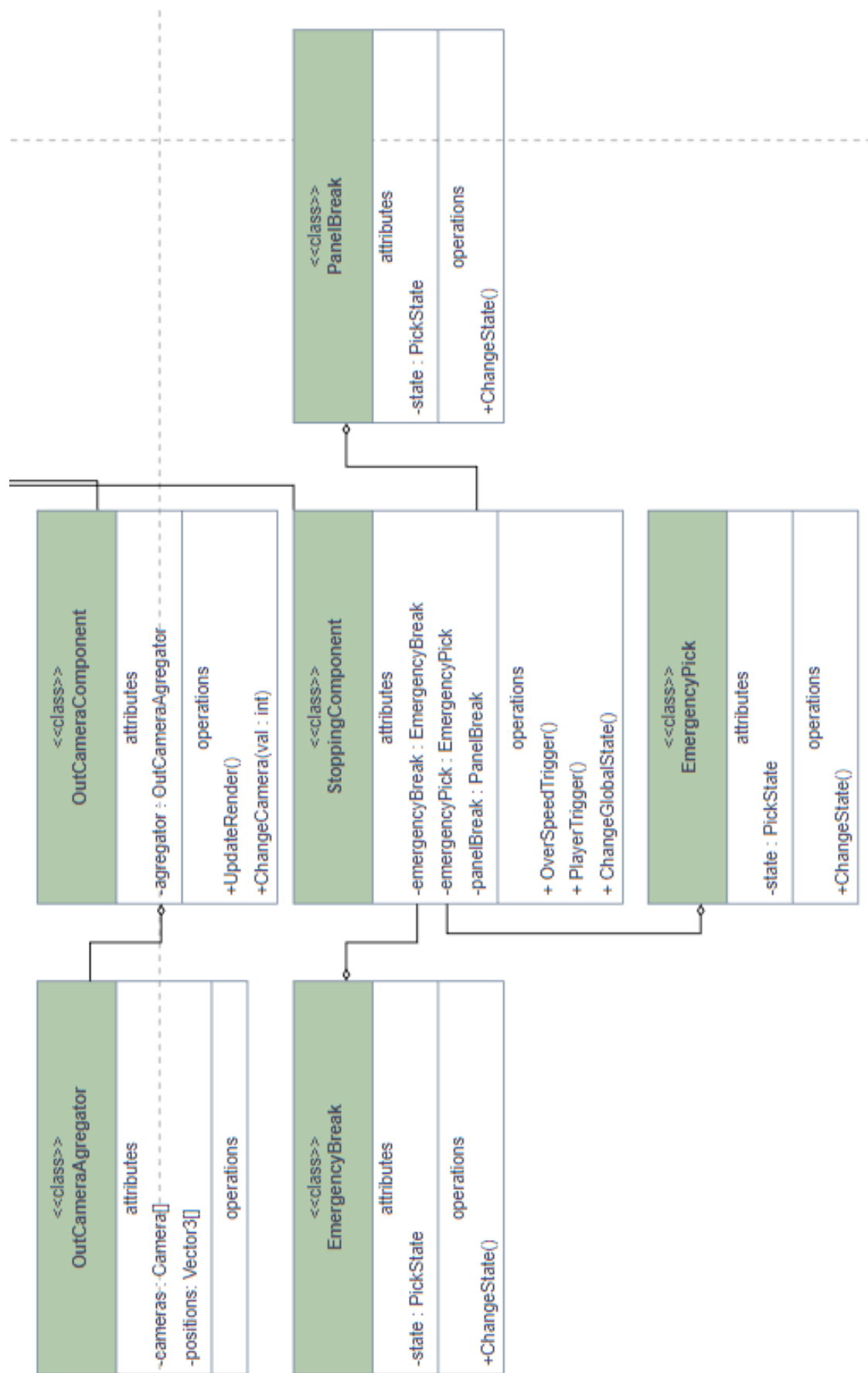


Рисунок 2.14 UML схема классов компонентов Часть 4

## 2.4. Выбор средства реализации

Объектно-ориентированный движок для разработки 2D и 3D игр и интерактивных приложений Unity.

Unity — это кроссплатформенный игровой движок, разработанный Unity Technologies, впервые анонсированный и выпущенный в июне 2005 года на Всемирной конференции разработчиков Apple Inc. С тех пор движок постепенно расширялся для поддержки различных настольных, мобильных, консольных платформ и платформ виртуальной реальности.

Unity можно использовать для создания трехмерных (3D) и двумерных (2D) игр, а также интерактивных симуляций и кинематографии. Движок был принят на вооружение отраслями, не относящимися к видеоиграм, такими как кино, автомобилестроение, архитектура, инженерия, строительство.

Основными преимуществами, которыми был обусловлен выбор именно этого движка при создании проекта, являются:

- бесплатная версия для небольших команд разработчиков
- наличие 3D визуализации во время проектирования игрового пространства
- использование приятного и мощного языка программирования – C#
- широкий спектр возможностей для настройки и проектирования
- мощный графический движок, обеспечивающий хорошую графику и приемлемый уровень оптимизации

Язык программирования C# — это современный, объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать множество типов безопасных и надежных приложений, работающих в .NET. C# уходит своими корнями в семейство языков

C# — это объектно-ориентированный, компонентно-ориентированный язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки этих концепций, что делает C# нативным языком для создания и использования программных компонентов. В Unity, начиная с версии 2017 года используется для разработки только этот язык

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |



программирования. В данном случае выбор языка обусловлен ограничениями платформы, однако выбор, сделанный разработчиками движка, имеет свои основания.

Основные преимущества языка:

- Сборка мусора автоматически освобождает память, занятую недоступными неиспользуемыми объектами.
- Типы с возможностью обнуления защищают от переменных, которые не ссылаются на выделенные объекты.
- Обработка исключений обеспечивает структурированный и расширяемый подход к обнаружению и восстановлению ошибок.
- Лямбда-выражения поддерживают методы функционального программирования.
- Синтаксис языка интегрированных запросов (LINQ) создает общий шаблон для работы с данными из любого источника.
- Языковая поддержка асинхронных операций обеспечивает синтаксис для построения распределенных систем.

Если для работы с данными и моделями будет использоваться интерфейс движка, то для программирования требуется IDE, в моем случае был выбран продукт компании JetBrains, IDE для разработки на C# под названием Rider.

В данной IDE имеется поддержка интеграции с Unity, что позволяет быстро писать, рефакторить и отлаживать код программного средства. Так же она позволяет работать с тестами и системами контроля версий прямо из интерфейса, что ускоряет работу и позволяет ограничить список используемых приложений.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

## Вывод к разделу 2

В данном разделе было осуществлено проектирование моделей предметной области с применением смешанного подхода. Для общего описания системы управления персоналом была использована диаграмма IDEF0, которая позволила отобразить в качестве входных данных системы: ввод пользователя и данные смежных систем.

Выходные данные системы содержат: вывод данных на сцену и результат управления системой. Они формируются в процессе работы пользователя с системой. Поэтому механизмом с точки зрения выбранной нотации выступают подсистемы управления поездом и контроллеры системы движения. Управляющие воздействия представлены правилами движения поездов и сценарными ограничениями.

Взаимодействие с разрабатываемым модулем было продемонстрировано также с применением схемы деятельности, показывающей цикличность и самодостаточность системы, что является показателем того, что при разработке были соблюдены Single Responsibility и Inversion of Control принципы.

Строение и организация классов показана на диаграмме классов в нотации UML. На ней можно проследить, что большинство сущностей связаны абстракциями, которые агрегируют в большие объекты, выстраивая четкое дерево инициализации, что упрощает работу с системой на программном уровне.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

### 3.Реализация подсистемы управления поездом в симуляторе машиниста московского метрополитена

#### 3.1. Создание главных сущностей для инициализации системы

Основными компонентами системы выступают инициализаторы системы движения, системы управляющей панели поезда, системы считывания действий пользователя. Эти компоненты реализованы с помощью плагина Zenject, предоставляющего доступ к внедрению зависимостей в проект. Регистрация зависимостей и является ответственностью этих инициализаторов. После регистрации зависимостей система переходит в состояние создания контекстов систем предметной области. Создаются все необходимые сервисы, происходит загрузка сцены с поездом и игроком (рис 3.1)

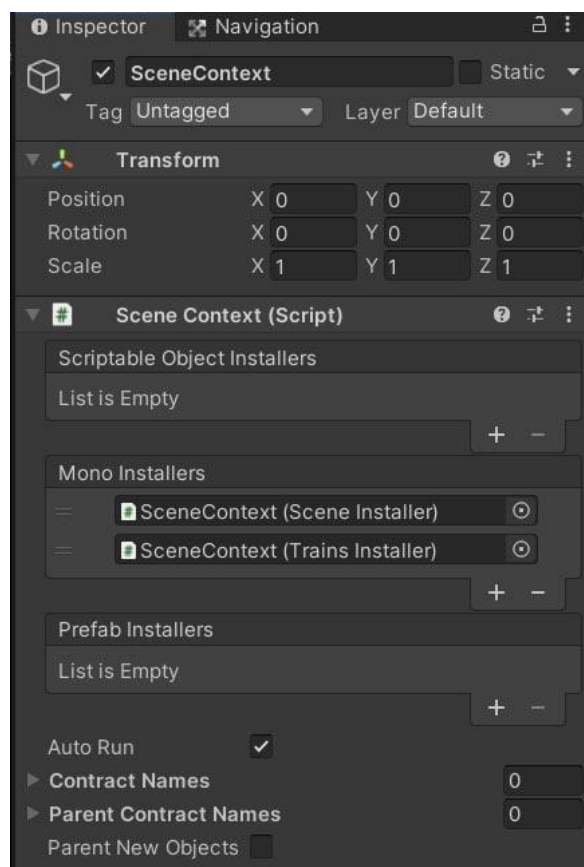


Рисунок 3.1 Ответственный за инициализацию объект

|            |      |                |         |      |   |                                       |  |      |      |        |
|------------|------|----------------|---------|------|---|---------------------------------------|--|------|------|--------|
|            |      |                |         |      | Библиографический список  |                                       |  |      |      |        |
|            |      |                |         |      |   |                                       |  |      |      |        |
| Изм.       | Лист | № докум.       | Подпись | Дата |   |                                       |  |      |      |        |
| Разраб.    |      | Богунов А.А.   |         |      | <b>Подсистема управления<br/>поездом в симуляторе<br/>московского метрополитена</b> |                                       |  | Лит. | Лист | Листов |
| Руководит. |      | Подгорный Н.Н  |         |      |   |                                       |  |      |      |        |
| Консульт.  |      | Подгорный Н.Н  |         |      |   |                                       |  |      |      |        |
| Н. Контр.  |      | Лазебная Е.А.  |         |      |   |                                       |  |      |      |        |
| Зав. Каф.  |      | Старченко Д.Н. |         |      |   |                                       |  |      |      |        |
|            |      |                |         |      |   | <i>БГТУ им. В.Г. Шухова<br/>ИТ-42</i> |  |      |      |        |

Инициализация всех систем происходит через привязку зависимостей в IoC-контейнер плагина Zenject, как показано в фрагменте кода ниже:

```
Container.Bind<DoorAgregator>()
    .ToSelf()
    .WithConcreteId("door_agregator")
    .NonLazy();

Container.Bind<IHeadlightsController>()
    .To<HeadlightsController>()
    .FromInstance(_headlightsController)
    .AsTransient()
    .Lazy();
```

Инициализация поезда начинается в классе TrainCreator, вызовом метода Setup.

```
public void Setup(TypeTrain typeTrain, CurvySpline spline, float position)
{
    _spline = spline;
    _typeTrain = typeTrain;
    _position = position;

    var trainInfo = GetSelectedTrain(typeTrain);
    _forwardCar = trainInfo.TrainCarManagers[0];
    InstallTrain(typeTrain, trainInfo);
    SetupSpline(spline, typeTrain, position);
}
```

После выполнения метода поезд встанет на определенный сплайн и на заданную позицию.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

### 3.2. Разработка модуля управления поездом

Поскольку в техническом задании было сказано, что симулятор будет имитировать не весь функционал реального поезда, что подсистема управления должна быть единой для всех поездов симулятора, а также что архитектура должна быть гибкой, легко расширяемой, для минимизации издержек при изменении деталей реализации под разные модели поездов и при добавлении нового функционала в существующие модели. Для этого каждый общий модуль был выделен в отдельную абстракцию, реализация которой уже зависела от конкретного случая. (рис 3.2)

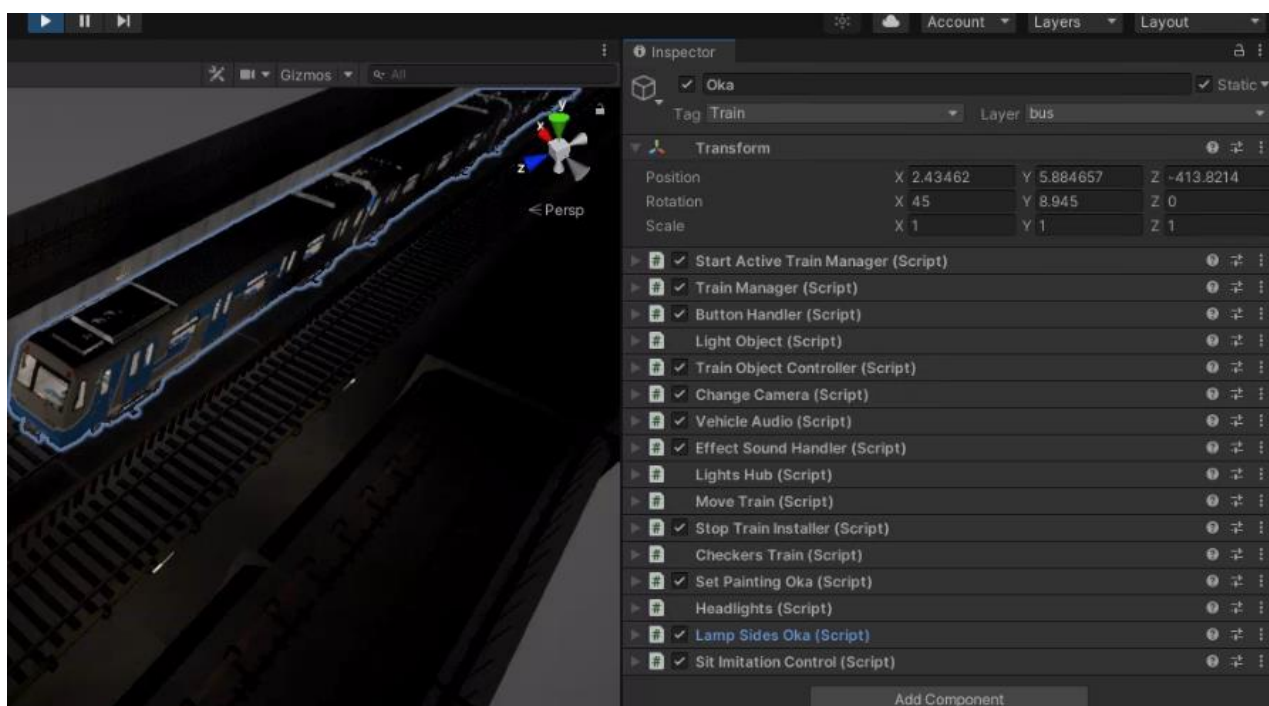


Рисунок 3.2 Объект поезда со списком сущностей

### 3.3. Разработка модуля движения поезда по сплайну

Согласно техническому заданию, поезд должен двигаться по заранее построенному дизайнером маршруту. Для этого было решено использовать плагин для Unity под названием Curvy spline. Он позволяет построить в пространстве линию, по которой можно пустить объект SplineMover, который будет с заданной скоростью перемещаться вдоль этой линии. Соответственно после того, как будет построена 3D визуализацию тоннеля, дизайнер уровней проложит вдоль этого тоннеля линию.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

Объектами SplineMover-ами выступают тележки вагонов поезда, которые и будут двигаться вдоль линий. На данном этапе нужно была разработана система, считывающая показания с системы управления движением и применяющая уже высчитанные значения скорости к тележкам поезда.

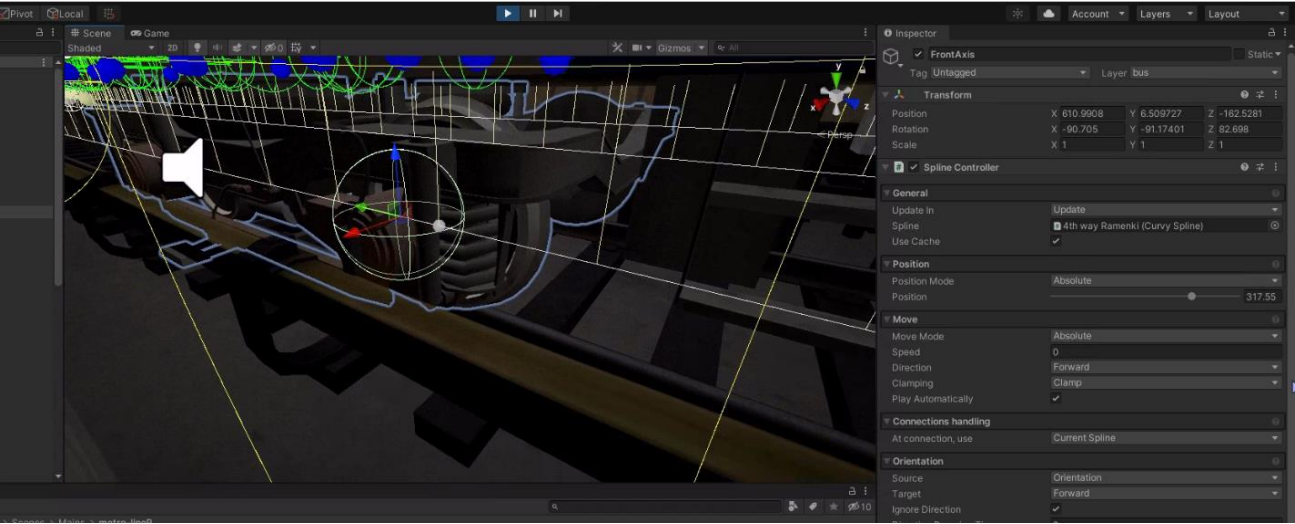


Рисунок 3.3 Поезд с навешенными сущностями

### 3.4.Симуляция физики реального состава

После тестирования системы было выявлено, что симулятор не передает ощущений реального поезда, отсутствует раскачка кабины, ускорение и замедление на склонах. Для этого были подключены компоненты Unity для симуляции физики под названием Rigidbody. Этот компонент моделирует поведение физического тела с заданными параметрами (масса, сопротивление) посредством инструментов движка. Так же была написана система уклонов, влияющая на ускорение поезда под разными углами наклона отдельных его вагонов. Алгоритм работы этой системы приведен на рисунке 3.4.

```

public float Modify(float acceleration)
{
    var listCarts = GetAllControllers();
    var countCart:int = listCarts.Count;

    float sum = 0;
    if (_moveTrain.IsForward() == (listCarts[0].Spline.Direction == SplineDirection.Forward))
    {
        sum = listCarts.Sum(selector: cart:SplineController => cart.GetSlope(listCarts[0].Spline.Direction));
    }
    else
    {
        var currentDirection = listCarts[0].Spline.Direction == SplineDirection.Forward
            ? SplineDirection.Backward
            : SplineDirection.Forward;

        sum = listCarts.Sum(selector: cart:SplineController => cart.GetSlope(currentDirection));
    }

    var average:float = sum / countCart;
    var slopeFactor:float = average * SlopeCoefficient;
    var trainPull:float = acceleration - SelfDeceleration;

    var deltaSpeed:float = Time.deltaTime == 0
        ? 0.1f * (trainPull + slopeFactor)
        : Time.deltaTime * (trainPull + slopeFactor);

    return deltaSpeed;
}

```

Рисунок 3.4 Поезд с навешенными сущностями

### 3.5.Создание инструментов для быстрой конфигурации поездов

Ввиду большого количества настраиваемых параметров, которые подгоняются геймдизайнером исходя из его ощущений, было решено разработать удобные инструменты для быстрого изменения и сохранения характеристик поездов, хранения ссылок на необходимые модели и объекты системы. Для этого были написаны расширения для редактора Unity (рисунок 3.5) и специальные сериализуемые хранилища данных, предоставляемые движком — ScriptableObjects. В них будет храниться информация о конфигурации поездов. Пример такого объекта приведен на рисунке 3.6.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

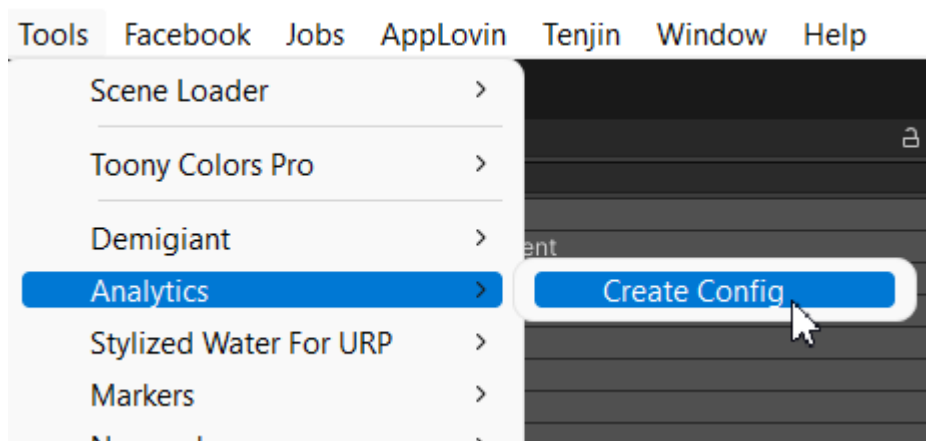


Рисунок 3.5 Поезд с навешенными сущностями

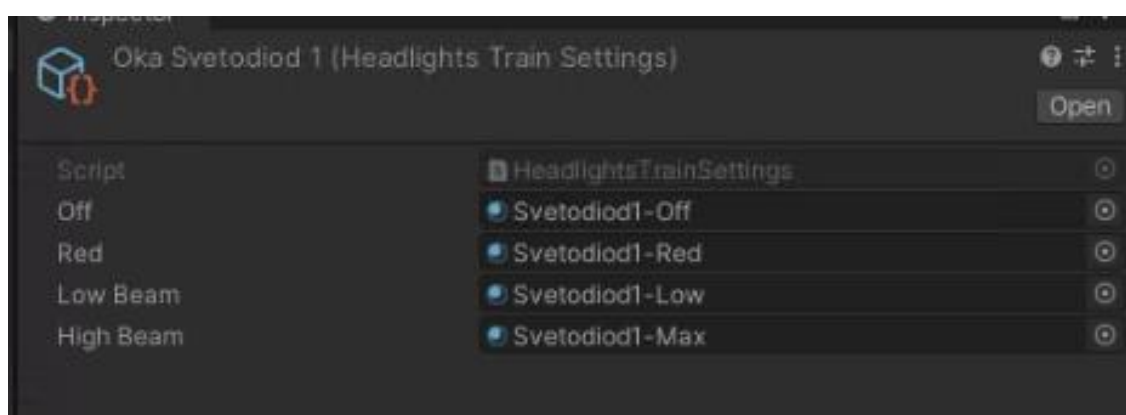


Рисунок 3.6 Поезд с навешенными сущностями

### 3.6.Написание Unit-тестов

Для оптимизации процесса тестирования изменений системы и упрощения ее поддержки был написан модуль Unit-тестов основных модулей системы, проверяющих основные сущности, ответственные за обработку математических вычислений и управление модулями движения поезда.

Для тестирования использовалась библиотека Nunit для тестирования приложений на платформе .Net. Целью было покрыть тестами минимум 70% кода подсистемы движения, основной упор делался на модули, ответственные за движение поезда. Часть списка тестов приведена на рисунке 3.7.



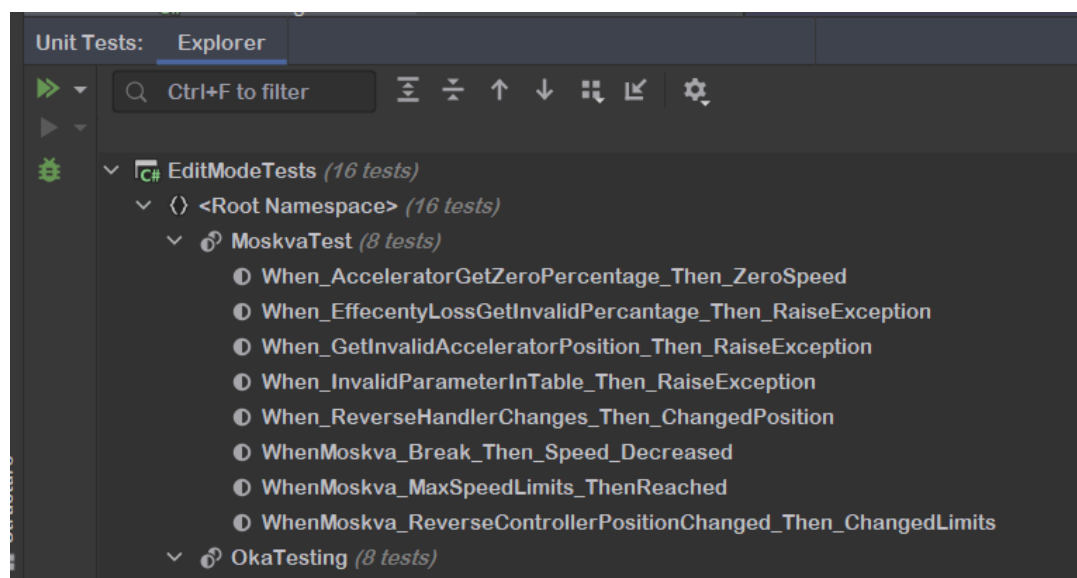


Рисунок 3.7 Поезд с навешенными сущностями

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

### Вывод к разделу 3

В данном разделе была спроектирована структура подсистемы движения симулятора московского метрополитена. На основе анализа предметной области и требований заказчика был спроектированы функционал и средства для поддержки и расширения системы.

В ходе разработки подсистемы были освоены навыки работы с игровым движком Unity, с его компонентами и средствами. Более глубоко изучен язык C#, платформа .Net и процесс тестирования программного обеспечения.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

## Заключение

В ходе выполнения дипломной работы была реализована подсистема управления поездом в симуляторе машиниста московского метрополитена. В начале была сформулирована цель, определены задачи и функции, которые необходимо разработать, чтобы выполнить поставленную цель. Также были рассмотрены существующие аналоги системы, их достоинства и недостатки, которые были учтены при реализации. Итогом стало создание технического задания, которое содержит требования к разрабатываемой системе. Был проведен анализ бизнес-процессов информационной системы для реализации подсистемы управления поездом в симуляторе машиниста. Также была разработана модель данных. Выбраны языки программирования и инструментальные средства, благодаря которым реализованы основные сущности и элементы.

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

## Библиографический список

1. Подвижной состав Московского метрополитена, полное описание структуры и основных механизмов [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/metro-wagon> (Дата обращения: 09.03.2022);
2. Справочник по станциям Московского Метрополитена, полный список всех существующих станций с расписание поездов на них [Электронный ресурс] URL: <https://www.mosmetro.ru/passengers/information/stations/> (Дата обращения: 14.03.2022);
3. Электродепо для поездов метро [Электронный ресурс] URL: <https://stroi.mos.ru/elektrodep-o-dlia-poiez-dov-mietro> (Дата обращения: 04.04.2022);
4. Приказ Минтруда России от 06.10.2015 N 691н «Об утверждении профессионального стандарта Специалист по управлению составом метро» [Электронный ресурс] URL: <https://mintrud.gov.ru/docs/mintrud/orders/2315> (Дата обращения: 18.04.2022);
5. Постановление Минтруда РФ от 21.08.1998 N 37 «Об утверждении Квалификационного справочника должностей руководителей, специалистов и других служащих» [Электронный ресурс] URL: <https://mintrud.gov.ru/docs/mintrud/orders/1956> (Дата обращения: 18.05.2022);
6. Москва-2020: поезд будущего в метрополитене столицы [Электронный ресурс] URL: <https://ria.ru/20170930/1505838490.html> (Дата обращения: 18.05.2022);
7. Тайны метро: принципы движения поездов и подвижных составов [Электронный ресурс] URL: <https://autoreview.ru/articles/moskva-2020-poezd-buduschego-v-metropolitene-stolicy> (Дата обращения: 18.05.2022);

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

8. Локальные нормативные акты [Электронный ресурс] URL: <http://www.hr-ok.ru/kadrconsalt/articles/6842/> (Дата обращения: 18.05.2022);
9. ГОСТ Р 7.0.97-2016 [Электронный ресурс] URL: <https://docs.cntd.ru/document/1200142871> (Дата обращения: 11.05.2022);
10. ГОСТ Р 7.2.12-2011 [Электронный ресурс] URL: <https://docs.cntd.ru/document/107518387> (Дата обращения: 13.05.2022);
10. Принцип работы системы движения поезда «Номерной» [Электронный ресурс] URL: <https://traffic-rules.com/ru/trains/b-177-264> (Дата обращения: 18.05.2022);

|      |      |          |         |      |  |      |
|------|------|----------|---------|------|--|------|
|      |      |          |         |      | Проектирование подсистемы управления поездом | Лист |
|      |      |          |         |      |  |      |
| Изм. | Лист | № докум. | Подпись | Дата |  |      |

## **ПРИЛОЖЕНИЕ 1. Техническое задание**

### **1. Общие сведения**

#### **1.1. Полное наименование системы и ее условное обозначение**

Полное наименование – Подсистемы управления поездом в симуляторе машиниста московского метрополитена.

Условное обозначение – СУП.

#### **1.2. Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты**

Заказчик – Кафедра ИТ ИЭИТУС БГТУ им. В. Г. Шухова и  
ИП Щербаков Кирилл Вадимович.

Разработчик – Богунов Артем Александрович.

#### **1.3. Основание для разработки.**

Программное средство разрабатывается в рамках дипломного проектирования на основании учебного плана направления 09.03.02– «Информационные системы и технологии».

#### **1.4. Плановые сроки.**

Начала работ – 01.09.2020

Окончание работ – 31.05.2021.

#### **1.5. Источник финансирования.**

Кафедра ИТ ИЭИТУС БГТУ им. В. Г. Шухова и  
ИП Щербаков Кирилл Алексеевич.

#### **1.6. Порядок оформления и предъявления заказчику результатов работ**

Техническое задание, эскизный проект, рабочая документация, схема работы;

Технический проект предъявляется отдельно по завершении разработки;

По завершении технического проекта система предоставляется заказчику для первоначального тестирования.

### **2. Назначение и цели создания системы**

#### **2.1. Назначение системы**

Основной задачей системы служит ознакомление заинтересованных пользователей в устройстве и работе московского метрополитена, а также в особенностях работы машиниста метро. Так же созданная система может использоваться как тренажер для машинистов, так как воссозданный процесс взаимодействия очень близок к реальному взаимодействию машиниста метрополитена.

## 2.2. Цели создания системы

- Разработка подсистемы, отвечающей за движение поезда в игровом пространстве по заранее определенному пути.
- Создание второстепенных систем, задействованных в управлении поезда.
- Реализация различных способов управления движением поезда.

## 3. Характеристики разрабатываемой информационной системы

### 3.1. Краткие сведения о разрабатываемой информационной системе

В процессе взаимодействия с игрой пользователь имеет возможность взять на себя роль машиниста одного из трех поездов и проехать по линии московского метро. Система управления и взаимодействия с второстепенными модулями у каждого поезда различается.

## 4. Требования к системе

### 4.1. Требования к функционированию системы

Система должна предоставлять возможность свободного и логически правильного управления поездом, а также всеми его компонентами.

### 4.2. Требования к информационной и программной совместимости

Разработанный сервис работает под Windows, Linux

### 4.3. Требования к программной документации

- Наличие сопроводительной документации.
- Разрабатываемые модуль должен быть самодокументирован, т.е. тексты программ должны содержать все необходимые комментарии.

### 4.4. Требования к составу и содержанию работ по вводу информационной системы

При подготовке к вводу в эксплуатацию ИС заказчик должен определить ответственных лиц за внедрение и проведение опытной эксплуатации.

### **5. Состав и содержание работ по созданию системы**

| Стадия                        | Этапы  |
|-------------------------------|--|
| Эскизный проект               | Разработка предварительных проектных решений по всей системе<br>Разработка документации  |
| Техническое задание           | Разработка ТЗ<br>Согласование ТЗ<br>Утверждение ТЗ   |
| Технический и рабочий проекты | Разработка системы<br>Разработка программной документации<br>Испытания системы   |
| Ввод в действие               | Подготовка объекта автоматизации к вводу системы в действие<br>Подготовка персонала<br>Проведение опытной эксплуатации<br>Проведение проверочных испытаний |

### **6. Требования к составу и содержанию работ по подготовке информационной системы к вводу в действие.**

Для подготовки сервиса к вводу в действие необходимо связать программные абстракции с существующими в проекте связями

Требования к документированию.

Перечень документов, подлежащих к разработке:

план-график работ по разработке системы;

ведомость технического проекта;

пояснительная записка к техническому проекту;

общее описание системы;



описание организационного, информационного, программного и технического обеспечения системы.

## **7. Требования к документированию**

Перечень документов, подлежащих к разработке:

- план-график работ по разработке системы;
- ведомость технического проекта;
- пояснительная записка к техническому проекту;
- общее описание системы;
- описание организационного, информационного, программного и технического обеспечения системы;