

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Государственный университет морского и речного флота имени адмирала С.О. Макарова»**

**(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)**

**Институт водного транспорта**

**Кафедра вычислительных систем и информатики**

ОТЧЕТ ПО РАСЧЕТНО-ГРАФИЧЕСКОЙ РАБОТЕ

по дисциплине:

«Синтез программных систем»

на тему:

«Разработка игры жанра "Симулятор" - "Жизнь почтальона"»

Работу выполнил:  
студент группы ИС-42  
Дель А.А.

Работу принял: доцент кафедры ВСиИ,  
к.т.н., Журавлёв А.Е.

Санкт-Петербург

2024

Оглавление

[Введение 4](#_Toc31728)

[1. Общие сведения, обзор аналогов и формирование требований к проектируемому программному средству 6](#_Toc6762)

[1.1. Общие сведения об игре 6](#_Toc32024)

[1.2. Обзор аналогов 6](#_Toc9449)

[1.3. Сводная таблица оценки 12](#_Toc24409)

[1.4. Требования к проекту 13](#_Toc30222)

[1.5. Задачи 13](#_Toc3276)

[2. Логика живых объектов 15](#_Toc28974)

[2.1. Создание заготовки живых объектов 15](#_Toc24232)

[2.2. Реализация функционала 16](#_Toc18697)

[2.3. Подключение к мировой системе 16](#_Toc10606)

[3. Создание скриптов окружения 18](#_Toc16852)

[3.1. Создание заготовки игрового окружения 18](#_Toc17230)

[3.2. Реализация функционала игрового окружения 18](#_Toc14479)

[3.3. Создание скрипта окружения с общемировой системой 19](#_Toc7852)

[4. Создание скриптов объектов взаимодействия 20](#_Toc16869)

[4.1. Создание заготовок объектов взаимодействия 20](#_Toc8699)

[4.2. Реализаия функционала объектов взаимодействия 21](#_Toc1319)

[4.3. Налаживание связи между объектами взаимодействия, где это требуется. 22](#_Toc28487)

[5. Оптимизация игрового мира 23](#_Toc23597)

[5.1. Поиск проблемных областей кода 23](#_Toc8476)

[5.2. Измерение влияния и существенности проблемы на игровой мир и его системы 23](#_Toc18200)

[5.3. Оптимизация проблемной части кода 23](#_Toc28601)

[6. Руководство программиста 25](#_Toc31564)

[6.1. Игровые настройки 26](#_Toc2211)

[6.2. Игра 33](#_Toc20254)

[6.3. Диаграммы UML 35](#_Toc29722)

[Заключение 38](#_Toc4105)

[Приложение 39](#_Toc32518)

# Введение

Игры жанра «Симулятор» позволяют игрокам войти в виртуальный мир, который имитирует реальность или определенный аспект жизни. Они позволяют игрокам почувствовать себя в роли профессионала, испытать адреналин от опасных ситуаций или просто насладиться повседневной жизнью виртуального персонажа.

Большинство симуляторов фокусируются (полностью или частично) на имитации реальной жизненной деятельности. Тем не менее, многие симуляторы имитируют "фантастические" ситуации, например симуляторы космических сражений. Основными подвидами игровых симуляторов являются: узкотематические, широкотематические.

В основном узкотематические являются симуляторами технических средств, в частности: автомобилями, поездами, самолетами и вертолетами.

Широкотематическими являются симуляторы выполнения различных заданий. Пользователю предстоит выполнение определенных задач определенной профессии. Практически отсутствуют сложные технические средства. Управление становится дополнительным игровым действием вместо основного. Подобные проекты выделяются разноплановых геймплеем.

Широкотематические симуляторы условно разделяют на 3 основные группы:

- Конструкторы (Construction simulations) — моделирование и строительства городов, зданий, техники и тд;

- Симуляторы жизни (Life simulations) — имитация жизни человека;

- Симуляторы профессий (Professional simulations) и Социальные симуляторы (Social simulations) — имитация профессиональной и социальной деятельности;

«Жизнь почтальона» - однопользовательская игра в 3D-мире, в которой игрок берет на себя роль почтальона. Игра представляет собой симулятор профессии в фантастическом, основанном на реальном мире. Игроку предстоит доставлять посылки и письма по поселку «Торковичи», покупать вспомогательное оборудование и технику, сталкиваться с другими существами и, по необходимости, противостоять им.

# Общие сведения, обзор аналогов и формирование требований к проектируемому программному средству

## Общие сведения об игре

Игра «Postman life» представляет собой бродилку в открытом 3D мире, где игроки берут на себя роль почтальона в поселке городского типа.

В игровом мире будут функциональные здания, такие как почта и магазин, жилые и производственные здания, а также различные пасхалки и элементы окружения.

## Обзор аналогов

1. Totally Reliable Delivery Service (TRDS)

Totally Reliable Delivery Service - это основанная на физике игра-симулятор, разработанная We're Five Games и изданная tinyBuild. Действие игры разворачивается в открытом мире, где игроки берут на себя роль водителей доставки в хаотичном и непредсказуемом мире.

Одной из ключевых особенностей игры является многопользовательский режим, который позволяет до четырех игрокам играть вместе онлайн или локально. Игроки могут работать сообща, чтобы завершить доставку, или соревноваться друг с другом, чтобы узнать, кто сможет доставить больше посылок.

Игра доступна для покупки на нескольких платформах, включая Windows, Nintendo Switch, PlayStation 4 и Xbox One. Цена варьируется в зависимости от платформы, но обычно составляет от 15 до 20 долларов.

Минимальные требования для Windows:

Операционная система: Windows 7 или более поздняя версия

Процессор: Intel Core i5 2300 или эквивалент

Оперативная память: 4 ГБ ОЗУ

Графика: NVIDIA GeForce GTX 560 или эквивалент

DirectX: Версия 11

Объем памяти: 3 ГБ свободного места

В настоящее время игра не предлагает кроссплатформенную игру, что означает, что игроки на разных платформах не могут играть вместе. Однако игроки могут играть с другими на одной платформе, онлайн или локально.

В целом, Totally Reliable Delivery Service - это веселая и занимательная игра-симулятор, основанная на физике, в которую лучше всего играть с друзьями. Открытый мир, многопользовательский режим и низкие системные требования делают игру доступной для игроков на различных платформах и устройствах.



Рис. 1.1. Скриншот игры Totally Reliable Delivery Service

1. Route Me Mail and Delivery Co (RMMaD)

Route Me Mail and Delivery Co - это игра для доставки, разработанная TinyDino Games. Действие игры разворачивается в открытом 2D мире, где игроки берут на себя роль почтальона, доставляющего почту и посылки в различные места.

В игре нет многопользовательского режима. В игре представлены различные маршруты и режимы, включая новый режим под названием "Почтовая жизнь", который добавляет в игру основные боевые действия и новые карты.

Route Me Mail and Delivery Co доступна для покупки на платформе Windows через Steam. Цена игры составляет $9,99.

Минимальные системные требования для Windows:

ОС: Windows 7 или более поздняя версия

Процессор: Intel Core i3 2,4 ГГц или эквивалент

Оперативная память: 4 ГБ ОЗУ

Графика: NVIDIA GeForce 8800GT или эквивалент

DirectX: Версия 9.0c

Объем памяти: 1 ГБ свободного места

В целом, Route Me Mail and Delivery Co - это игра для доставки, которая предлагает игрокам множество маршрутов и режимов для изучения. Открытый мир и относительно низкие системные требования делают ее доступной для игроков на платформе Windows. Однако отсутствие многопользовательского режима и кроссплатформенности игры может быть недостатком для некоторых игроков.



Рис. 1.2. Скриншоты игры Route Me Mail and Delivery Co

1. Flying Neko Delivery (FND)

Flying Neko Delivery - это основанная на физике игра для доставки, разработанная Righteous Weasel Games. Действие игры разворачивается в красочном и причудливом мире, где игроки берут на себя роль летающего кота, доставляющего посылки в различные места.

В игре нет многопользовательского режима. В игре представлено множество уровней, каждый с различными задачами и препятствиями для преодоления.

Flying Neko Delivery доступна для покупки на нескольких платформах, включая Windows, Mac и Linux. Цена игры составляет 9,99 долларов.

Минимальные системные требования для Windows:

Операционная система: Windows 7 или более поздняя версия

Процессор: Intel Core 2 Duo 2,0 ГГц или эквивалент

Оперативная память: 2 ГБ ОЗУ

Графика: NVIDIA GeForce 8800GT или эквивалент

DirectX: Версия 9.0c

Объем хранилища: 500 МБ свободного места

В настоящее время игра не поддерживает кроссплатформенную игру, что означает, что игроки на разных платформах не могут играть вместе. Однако игроки могут играть с другими на той же платформе, онлайн или локально.

В целом, Flying Neko Delivery - это основанная на физике игра для доставки, которая предлагает игрокам множество уровней и задач для изучения. Причудливый мир и относительно низкие системные требования делают ее доступной для игроков на нескольких платформах. Однако отсутствие многопользовательского режима игры может быть недостатком для некоторых игроков.



Рис. 1.3. Скриншот игры Flying Neko Delivery

1. Lake

Lake - приключенческая игра, основанная на повествовании, разработанная Gamious. Действие игры разворачивается в маленьком городке Провиденс-Оукс, где игроки берут на себя роль Мередит Вайс, инженера-программиста, которая возвращается в свой родной город, чтобы на две недели взять на себя обязанности по доставке почты своего отца.

В игре присутствует среда открытого мира, позволяющая игрокам исследовать город и взаимодействовать с его жителями. Однако в игре нет многопользовательского режима.

Lake доступен для покупки на нескольких платформах, включая Windows, Mac и Nintendo Switch. Цена игры составляет $19,99.

Минимальные системные требования для Windows:

Операционная система: Windows 7 или более поздняя версия

Процессор: Intel Core i3 2,0 ГГц или эквивалент

Оперативная память: 4 ГБ ОЗУ

Графика: NVIDIA GeForce 560 или эквивалент

DirectX: Версия 11

Объем хранилища: 5 ГБ свободного места

В целом, Lake - это приключенческая игра, основанная на повествовании, которая предлагает игрокам возможность исследовать небольшой городок и делать выбор, влияющий на исход истории. Открытый мир и относительно низкие системные требования делают ее доступной и приятной для игроков на нескольких платформах. Однако отсутствие многопользовательского режима игры может быть недостатком для некоторых игроков.



Рис. 1.4. Скриншот игры Lake

1. Cloudpunk

Cloudpunk - приключенческая игра в стиле киберпанк, разработанная ION Lands. Действие игры разворачивается в городе Нивалис, где игроки берут на себя роль Нарнии, нового водителя доставки для Cloudpunk, компании по доставке, которая работает в подземном мире города.

В игре присутствует среда открытого мира, позволяющая игрокам исследовать город и его различные районы, выполняя миссии по доставке. Однако в игре нет многопользовательского режима.

Cloudpunk доступен для покупки на нескольких платформах, включая Windows, PlayStation 4, Xbox One и Nintendo Switch. Цена игры варьируется в зависимости от платформы, но обычно составляет от 24,99 до 29,99 долларов.

Минимальные системные требования для Windows:

Операционная система: Windows 7 или более поздняя версия

Процессор: Intel Core i3-3220 или эквивалент

Оперативная память: 4 ГБ RAM

Графика: NVIDIA GeForce GTX 560 или эквивалент

DirectX: Версия 11

Объем памяти: 6 ГБ свободного места

В целом, Cloudpunk - это приключенческая игра в стиле киберпанк, которая предлагает игрокам возможность исследовать огромный город и делать выбор, влияющий на исход истории. Открытый мир и относительно низкие системные требования делают ее доступной и приятной для игроков на нескольких платформах. Однако отсутствие многопользовательского режима игры может быть недостатком для некоторых игроков.



Рис. 1.5. Скриншот игры Cloudpunk

## Сводная таблица оценки

Таблица 1. Сводная таблица оценки аналогов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | TRDS | RMMaD | FND | Lake | Cloudpunk |
| 1. Открытый мир | 10/10 | 08/10 | 10/10 | 10/10 | 10/10 |
| 2. Мультиплеер | 10/10 | 0/10 | 0/10 | 0/10 | 0/10 |
| 3. Цена | 05/10 | 09/10 | 07/10 | 07/10 | 07/10 |
| 4. Минимальные требования | 06/10 | 10/10 | 08/10 | 07/10 | 06/10 |
| 5. Кроссплатформенность | 07/10 | 02/10 | 05/10 | 05/10 | 07/10 |
| Итог | 7,6/10 | 5,8/10 | 6/10 | 5,8/10 | 6/10 |

## Требования к проекту

В связи с выставленными оценками и учётом всех позитивных и негативных аспектов каждого конкурента нашего проекта, предлагаем следующие решения:

Таблица 2. Требования к проекту

|  |  |
| --- | --- |
| 1. Открытый мир | Проект Postman Life подразумевает открытый мир изначально. Он будет огромным, разнообразным и полным живых существ. По этому фактору мы сможем дать отпор конкурентам благодаря движку - Godot Engine, в который предустановлена отличная оптимизация 3D пространства, что даст больше свободы в строительстве игрового мира. |
| 2. Цена | Наша игра будет практически бесплатной, как минимум - дешевле чем у конкурентов. |
| 3. Минимальные требования | Благодаря новому движку и его инструментарию есть широкий функционал оптимизации игры для слабых ПК. |
| 4. Кроссплатформенность | Godot Engine - среда разработки проекта Postman Life позволяет экспортировать его практически на любую платформу: Android, IOS, Web, Windows, VR. Проект Postman Life будет портирован на: Android и Windows. |
| Порт на IOS и VR не будет осуществлен, так как у команды разработки нет устройств для тестирования. |
| Порт на Web не будет осуществлен, так как это потребует создания параллельного проекта с перебором функционала, что не вписывается в график разработки. |

## Задачи

Ведущий разработчик (Дель Александр Андреевич)

Задача:

Создание и внедрение скриптов в проект игры «Жизнь почтальона» для оживления виртуальных сцен.

Шаги выполнения:

1. Логика живых объектов

* Создать заготовки живых объектов
* Реализовать функционал
* Подключить к общей мировой системе

1. Создание скриптов окружения:

* Реализовать функционал игрового окружения
* Связать скрипт окружения с общемировой системой

1. Создание скриптов объектов взаимодействия:

* Создать заготовки объектов взаимодействия
* Реализовать функционал объектов взаимодействия
* Наладить связь между объектами взаимодействия, где это требуется

1. Оптимизация игрового мира

* Найти проблемные области кода
* Измерить влияние и существенность проблемы на игровой мир и его системы
* Оптимизировать проблемную область кода

1. Руководство Разработчика

# Логика живых объектов

Список всех живых объектов с общим описанием функционала приведен в таблице ниже.

Таблица 1. Сущности игры

|  |  |
| --- | --- |
| Сущность | Функционал |
| НЛО | Летает, может украсть и уничтожить посылку |
| Чайка | Летает в воздухе, приземлиться не может. С определенным шансом может украсть посылку, находящуюся вне инвентаря игрока |
| Корова | Мычит |
| Собака | Лает |
| Курица | Орет |
| Призрак | Воспроизводит скример для игрока |
| Гигантское НЛО | Появляется раз в 10 дней, издает громкий звук |

## Создание заготовки живых объектов

Создание заготовки живого объекта включает в себя добавление модели в игру и его базовая настройка: настройка коллайдеров (область столкновений), рейкастов (по сути - зрение объекта), источники звука, таймеры и прочее. Для каждой сущности эти параметры будут индивидуальными. К примеру, сущность «Птица» имеет следующие базовые настройки (Рис. 2.1): модель, коллайдер, таймер, источники звука.

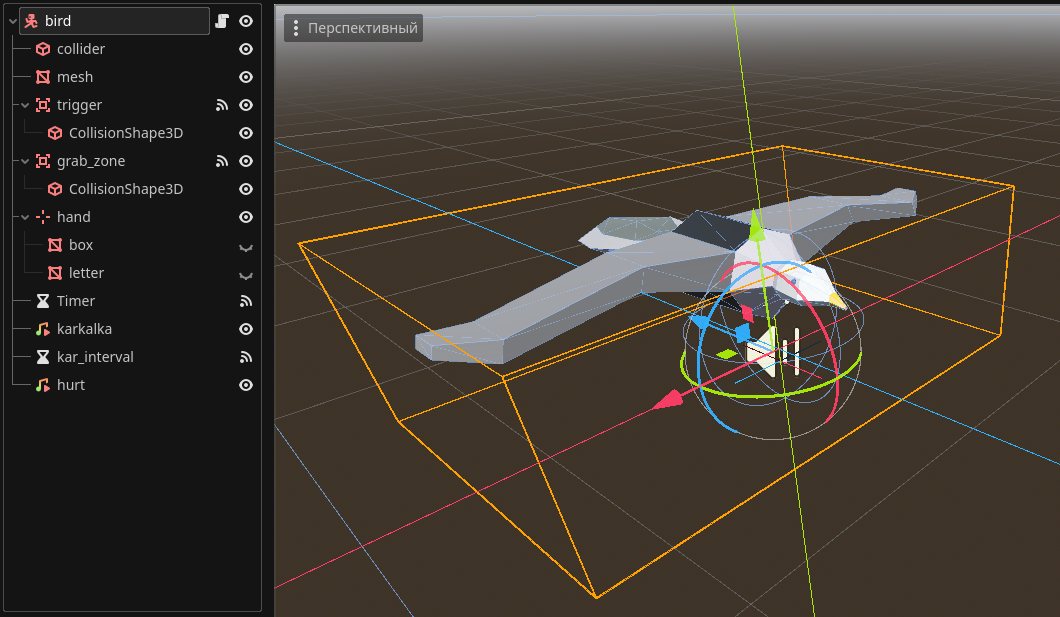


Рис. 2.1. Заготовка сущности - «Птица»

## Реализация функционала

Была написана логика всех живых объектов для игры «Жизнь почтальона». Реализация игрового функционала всех сущностей приведена в приложении в соответствующих скриптах. Реализация функционала включает в себя:

* Реализация появления и исчезновения сущностей в процессе игры;
* Реализация перемещения сущностей (по необходимости);
* Реализация звуков сущностей, в том числе вариативность;
* Балансировка количества сущностей.

## Подключение к мировой системе

Была написана логика подключения всех живых объектов к мировой системе для игры «Жизнь почтальона». Всего подобным образом было разработано 7 различных живых существ: Собака, Корова, Курица, НЛО, Гигантское НЛО, Призрак, Чайка.

Подключение сущностей к мировой системе включает в себя:

* Реализация появления сущностей в игровом мире;
* Реализация взаимодействия сущностей с игровым миром;
* Реализация взаимодействия сущностей с игроком.

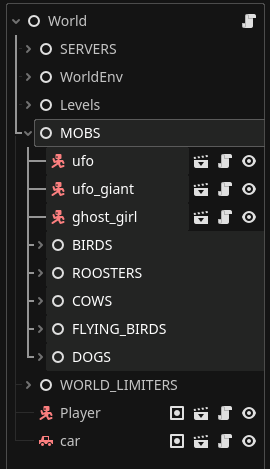


Рис. 2.2. Подключенные сущности к игровому миру

# Создание скриптов окружения

Игровое окружение — это очень важная составляющая любого проекта. Именно оно создает игровую атмосферу. С помощью окружения можно умело создавать разнообразные ситуации, погружающие игрока в виртуальный мир. Окружение — это своеобразное тело игры, в которое помещены персонажи.

## [Создание заготовки игрового окружения](#_Toc22522)

На Рис. 3.1 изображена заготовка окружения в игре «Жизнь почтальона». Заготовка игрового окружение подразумевает готовый игровой мир с точки зрения подключения неба, соответствующих шейдеров и анимации.

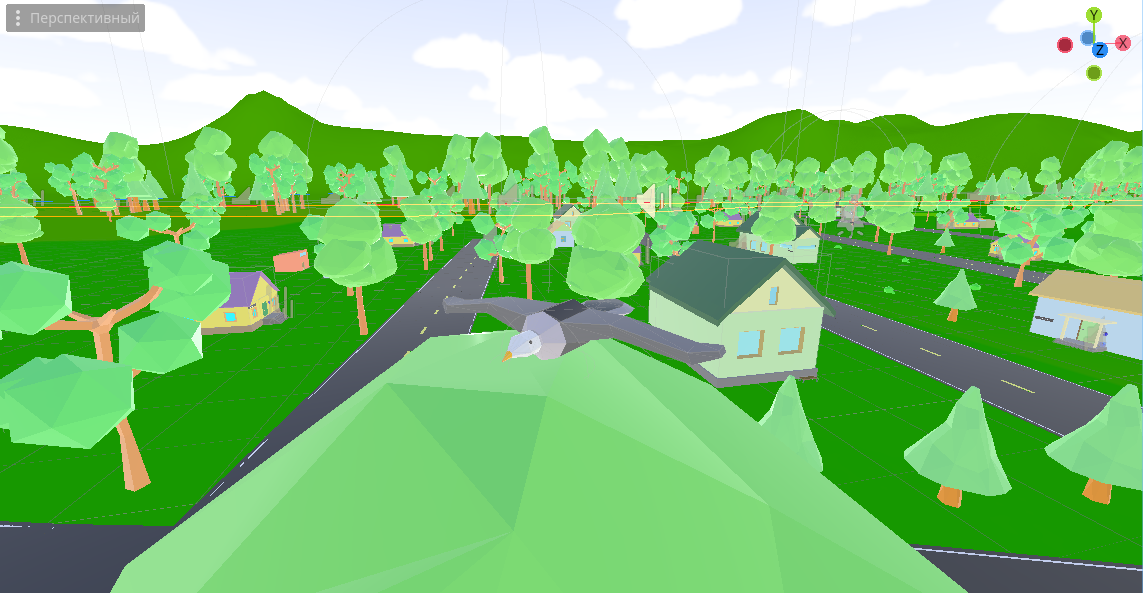


Рис. 3.1. Заготовка окружения в игре «Жизнь почтальона»

## [Реализация функционала игрового окружения](#_Toc21729)

На Рис. 3.2 отражен главный функционал окружения в игре «Жизнь почтальона» - смена дня и ночи, в виде анимации. Смена дня и ночи происходит постепенно в течении дня и является бесшовной, то есть игрок не заметит перехода между временем суток и сменой анимации.

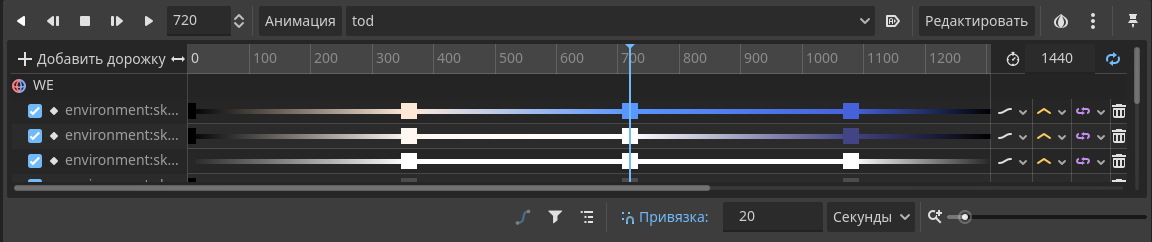


Рис. 3.2. Функционал окружения в игре «Жизнь почтальона»

## [Создание скрипта окружения с общемировой системой](#_Toc6105)

Была написана логика окружения для игры «Жизнь почтальона». Она включает в себя счет и отображение игрового времени и погода, а также изменение окружения в зависимости от них.



Рис. 3.3. Код окружения в игре «Жизнь почтальона»

# Создание скриптов объектов взаимодействия

## [Создание заготовок объектов взаимодействия](#_Toc13646)

Заготовка игрового объекта включает в себя добавление в проект модели или ее создание, а также нанесение на нее коллайдеров. Для каждого объекта заготовка будет представлять индивидуальный набор параметров. Для примера, так выглядит заготовка подбираемого предмета - «Молочка»:

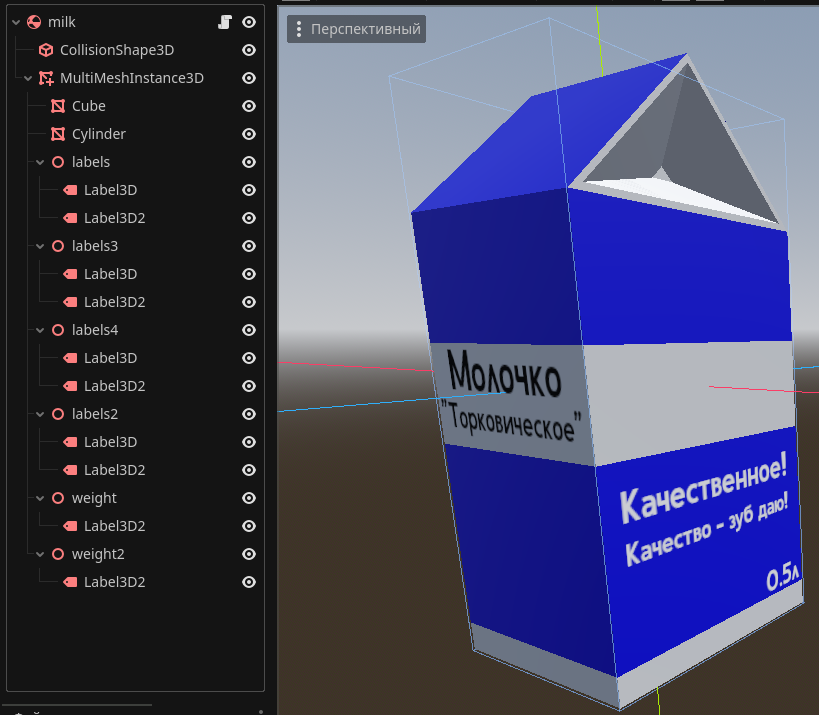


Рис. 4.1. Заготовка «Молочка» в игре «Жизнь почтальона»

## [Реализаия функционала объектов взаимодействия](#_Toc5762)

Реализация функционала объектов взаимодействия включает в себя следующие параметры:

* Описание параметров, таких как id, вес и имя;
* Описание процедуры взаимодействия (взятия).

Для каждого из объектов функционал будет индивидуальным. Простейший описан для объекта молочко:

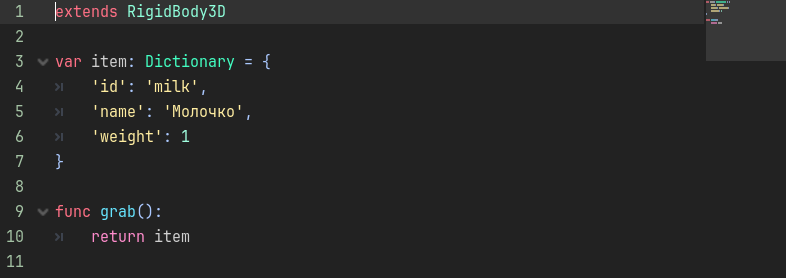


Рис. 4.2. Реализация функционала объекта «Молочко»

Пример реализации более сложного функционала для объекта «Мина»:

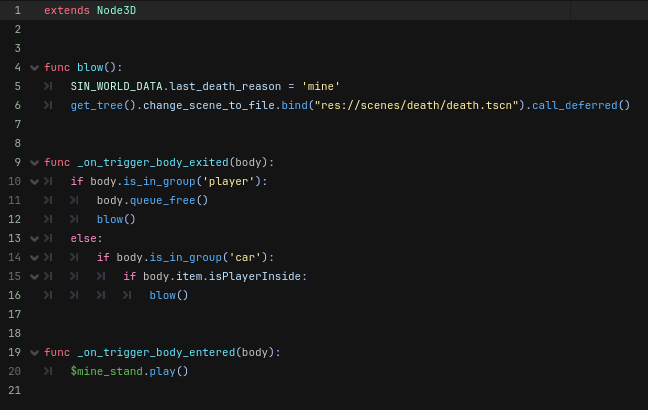


Рис. 4.3. Реализация функционала объекта «Мина»

## [Налаживание связи между объектами взаимодействия, где это требуется](#_Toc20573).

Налаживание связи подразумевает использование сигналов, которые позволяют из одного участка кода обращаться к другому, при условии что они никак не связаны (Рис. 4.3):

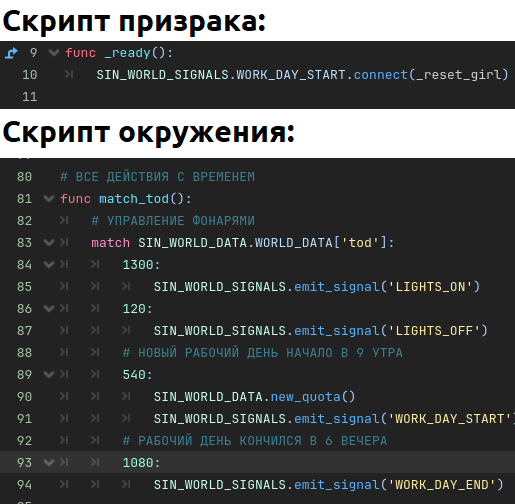


Рис. 4.4. Связь призрака с окружением в игре «Жизнь почтальона»

# Оптимизация игрового мира

## [Поиск проблемных областей кода](#_Toc6135)

В ходе разработки игры было создано и позже, обнаружено огромное множество неэффективного кода. В процессе оптимизации игрового мира эти проблемные области кода были исправлены.

Примером неэффективного кода может служить ошибка №121, отраженная на Рис. 5.1.

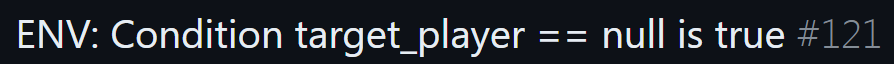


Рис. 5.1. Пример ошибки, отражающей неэффективный код

## [Измерение влияния и существенности проблемы на игровой мир и его системы](#_Toc30307)

Измерение существенности влияния ошибки (Рис. 5.2).

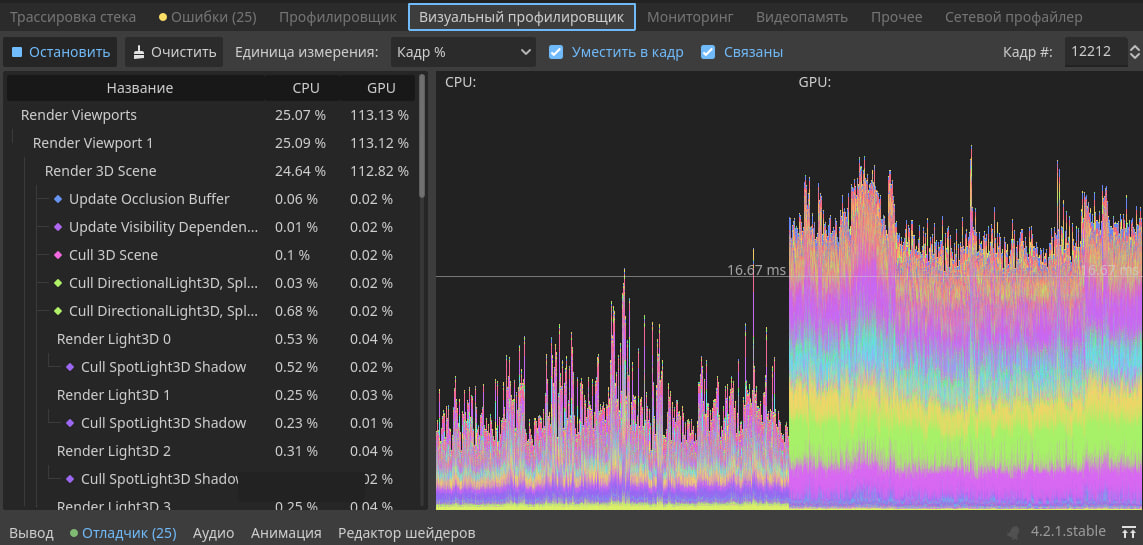


Рис. 5.2. Измерение существенности влияния ошибки

## [Оптимизация проблемной части кода](#_Toc20508)

Исправлением неэффективного кода стало избавление логики от постоянного обновления переменных и в целом удаления конечного автомата (Рис. 5.3).

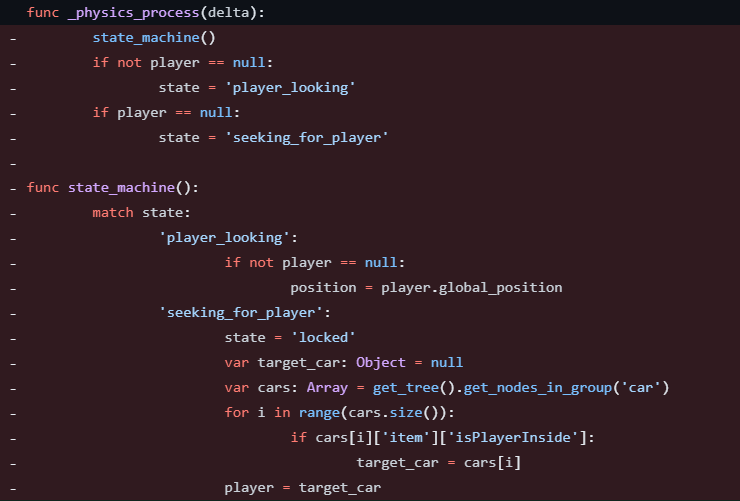


Рис. 5.3. Пример решения ошибки, исправления неэффективного кода

# Руководство программиста

При загрузке приложение появляется главное меню (Рис. 6.1), из которого пользователь может перейти в саму игру, ее настройки или выйти из игры.



Рис. 6.1. Меню приложения

Таким образом это окно выглядит в окне иерархии узлов. Существуют четыре кнопки, три из которых видны всегда, а одна дополнительная «Продолжить» - появляется только при наличии сохранения.



Рис. 6.2. Меню приложения - иерархия узлов

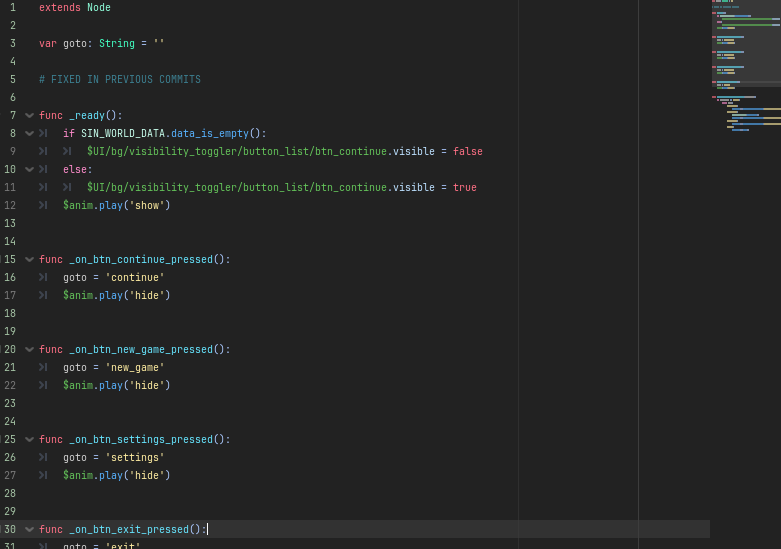


Рис. 6.3. Меню приложения - код

Код связывает сигналы кнопок с их логикой. Так, нажимая на каждую из них меняется переменная goto, по сути является id окна для перехода.

## Игровые настройки

При переходе в настойки у игрока появляется возможность детально настроить графику и звуки под свои потребности. Для возврата в меню необходимо нажать кнопку «Назад». В настройках есть 3 раздела: «Аудио», «Графика» и «Дополнительно». Для перемещения между разделами необходимо воспользоваться меню слева. Все настройки сохраняются и применяются автоматически при нажатии кнопки «Назад».

По умолчанию при переходе в настройки, открывается раздел «Аудио» (Рис. 6.4). Здесь пользователь может настроить общую громкость звуков и отдельно громкость музыки.

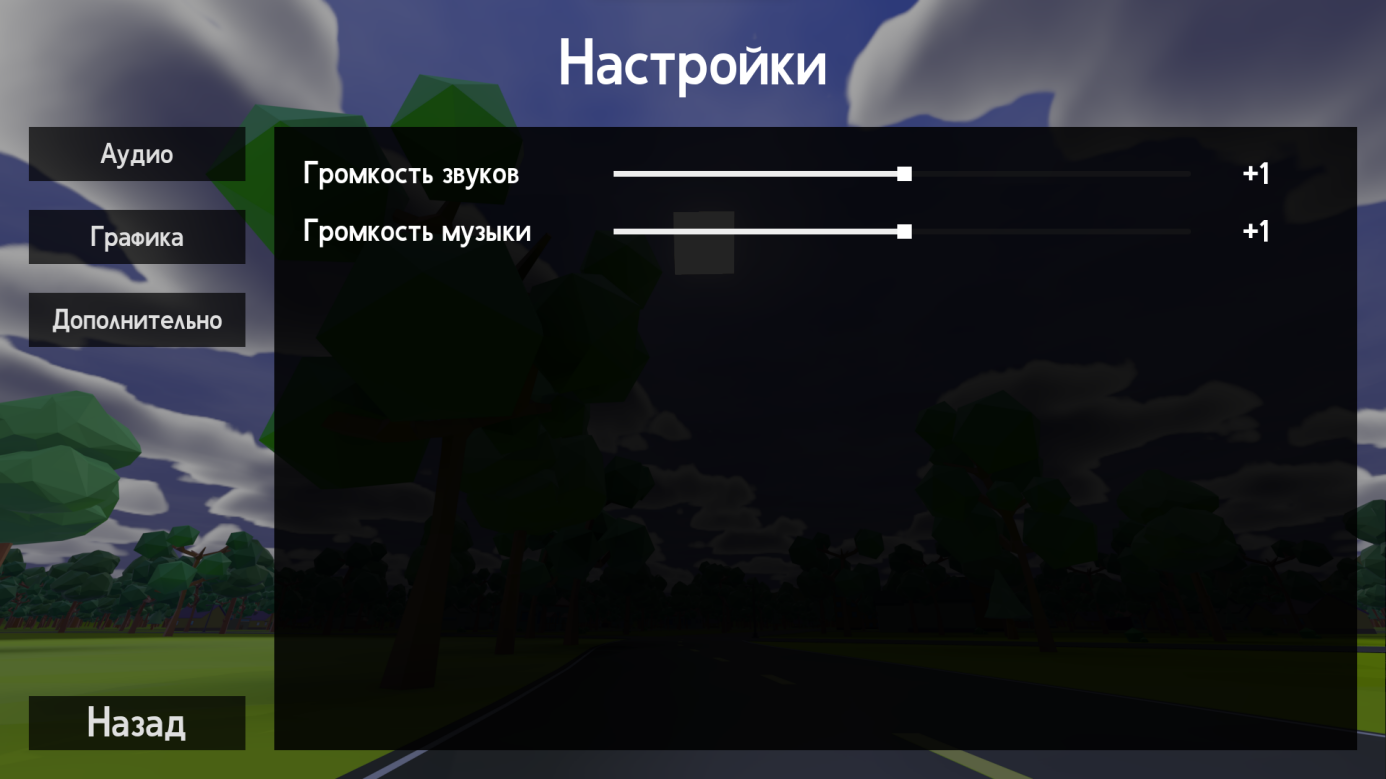


Рис. 6.4. Раздел «Аудио» в настройках приложения

Таким образом выглядит окно «Настройки» в иерархии узлов (Рис. 6.5).

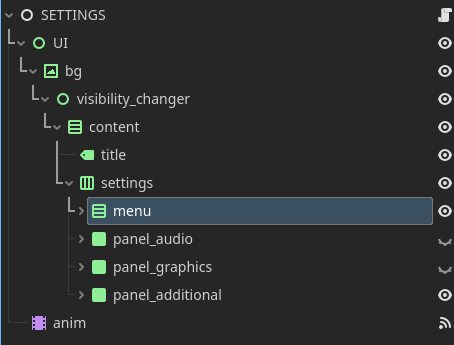


Рис. 6.5. Настройки - иерархия узлов

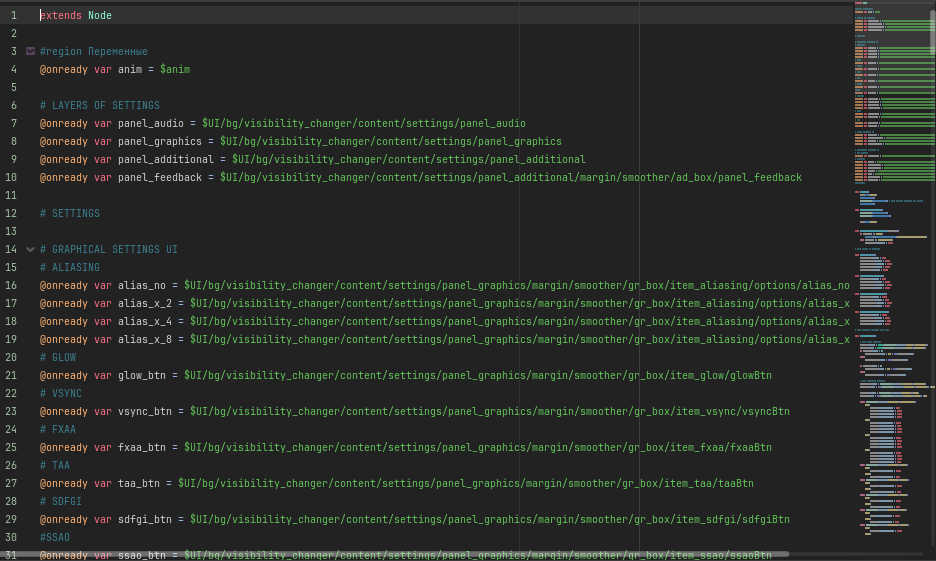


Рис. 6.6. Настройки Аудио - код

В разделе «Графика» (Рис. 6.7) у пользователя есть возможность детально настроить внутриигровую графику. Увеличение значений настроек может способствовать понижению производительности и FPS внутри игры.

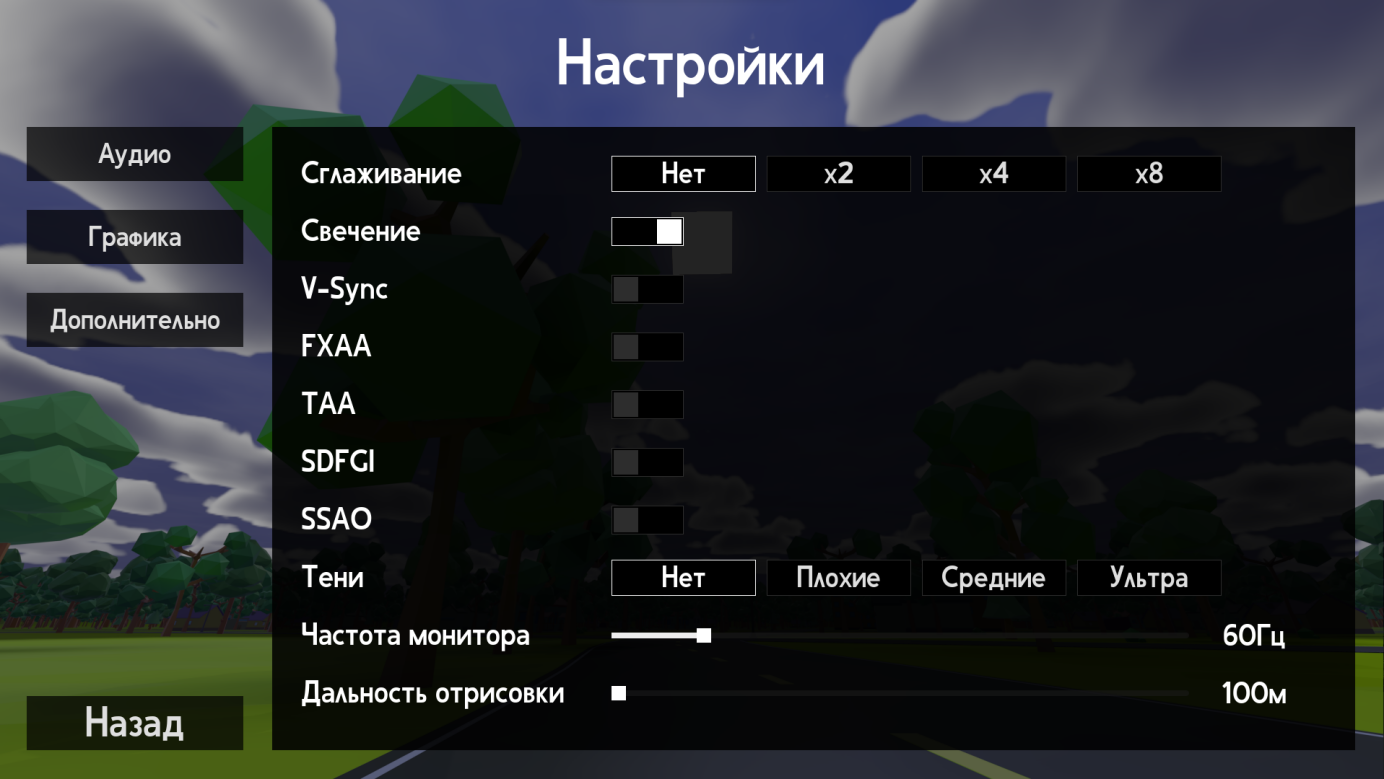


Рис. 6.7. Раздел «Графика» в настройках приложения

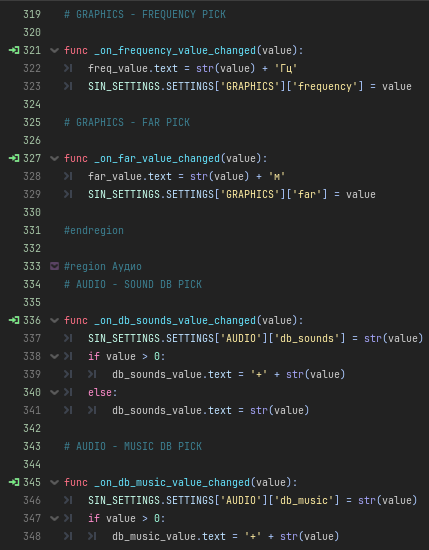


Рис. 6.8. Настройки Графики - код

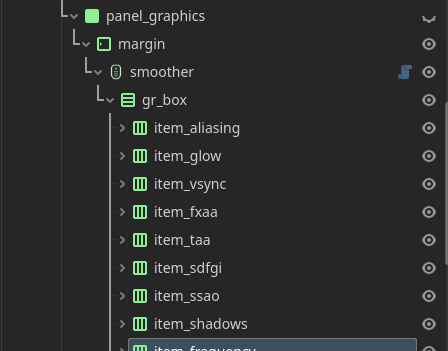


Рис. 6.9. Настройки Графики - иерархия узлов

В разделе «Дополнительно» (Рис. 6.10) есть возможность включить счетчик кадров (FPS) и отправить обращение разработчикам через специальную форму (Рис. 6.13). Для отправки обращения необходимо описать проблему или предложение в поле ввода и нажать кнопку «Отправить». Если пользователь хочет получить обратную связь - он может ввести данные для обратной связи в отдельном поле, до отправки обращения. Отправить пустое обращение нельзя.

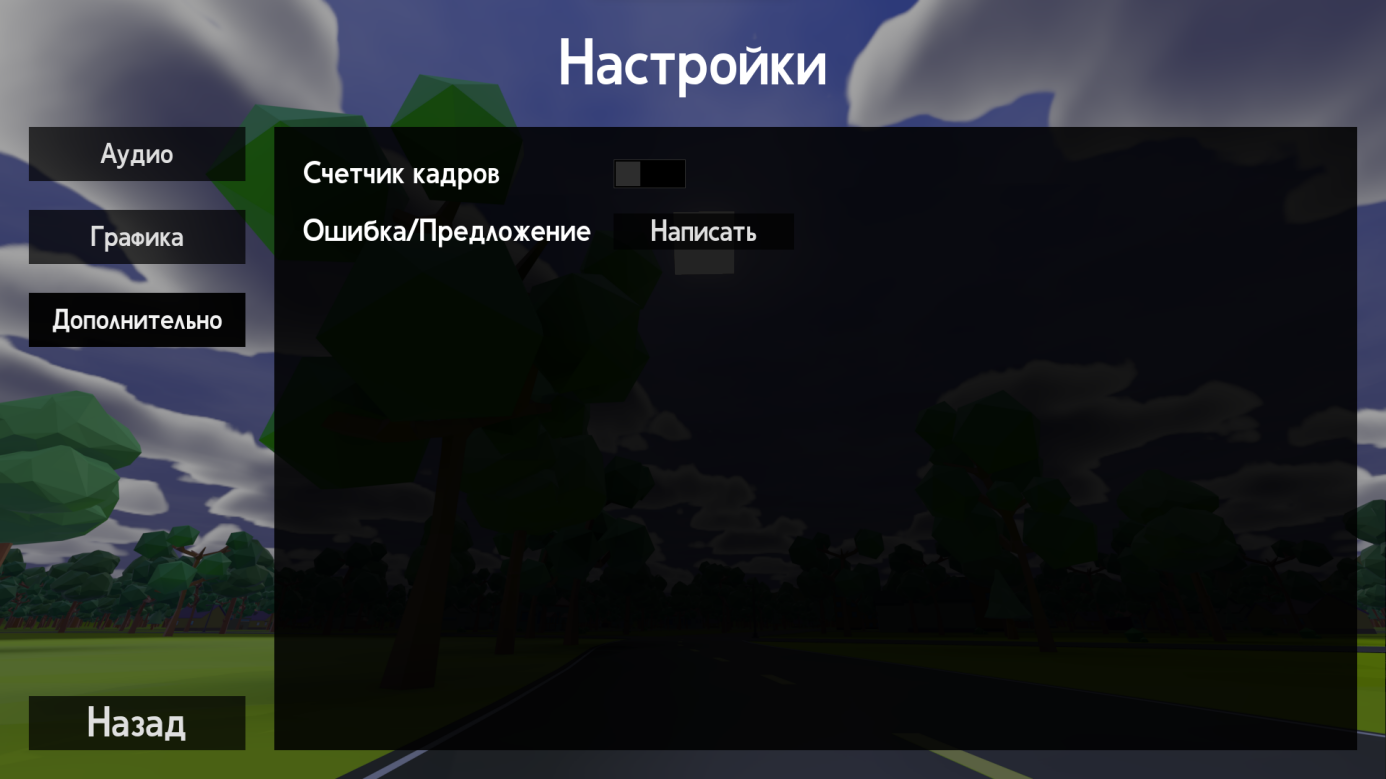


Рис. 6.10. Раздел «Дополнительно» в настройках приложения

Вот так выглядит окно настроек «Дополнительно» в окне иерархии узлов (Рис 3.5).

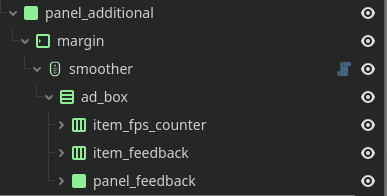


Рис. 6.11. Настройки Графики - иерархия узлов

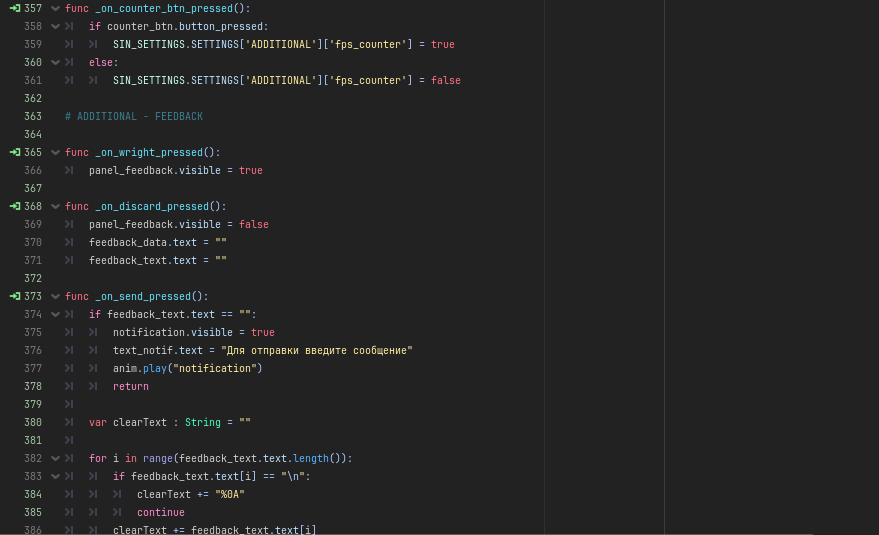


Рис. 6.12. Настройки «Дополнительно» - код

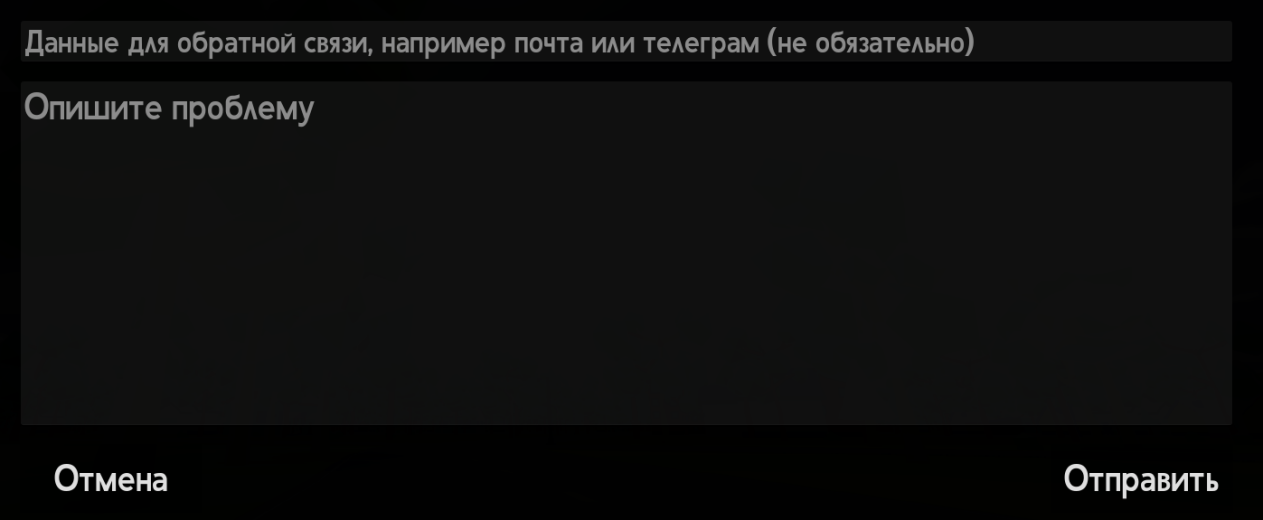


Рис. 6.13. Форма отправки обращения к разработчикам

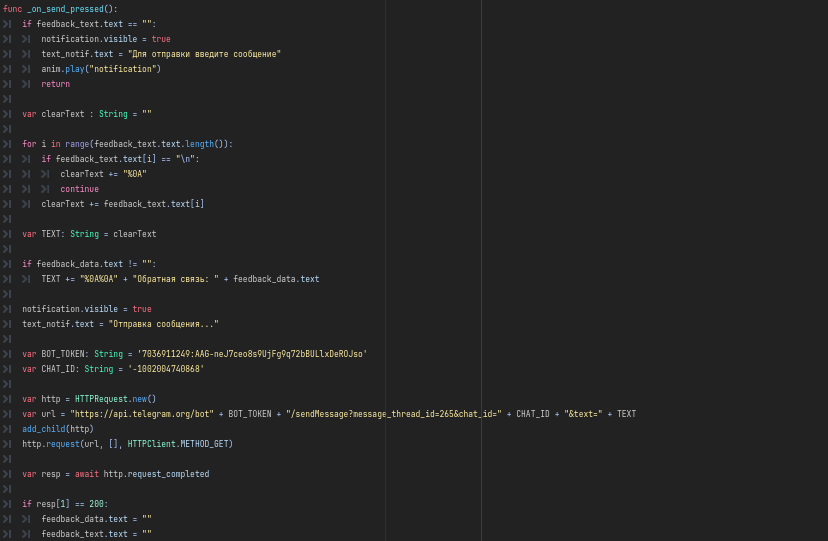


Рис. 6.14. Форма отправки обращения к разработчикам - код

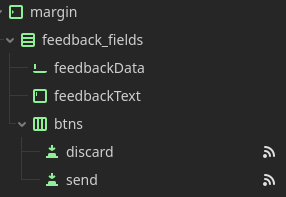


Рис. 6.15. Форма отправки обращения к разработчикам - иерархия узлов

Все настройки графики, аудио и дополнительные - сохраняются в отдельный файл- settings.dat (Рис. 6.16).



Рис. 6.16. Файл, содержащий настройки

В коде сохранение настроек выглядит следующим образом (Рис. 6.17).

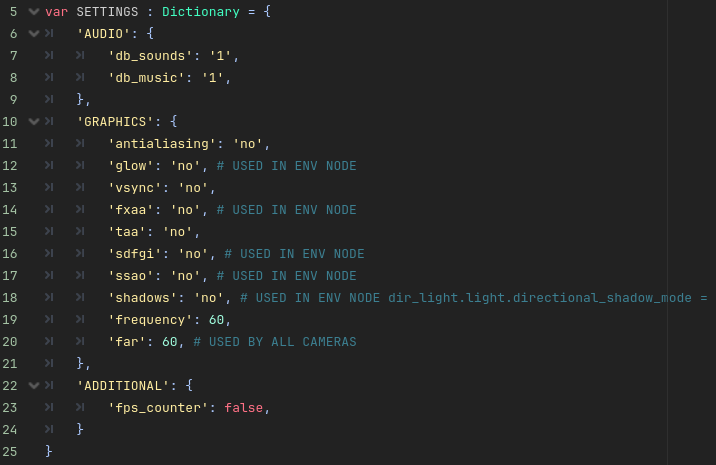


Рис. 6.17. Словарь значений файла «Настройки»

## Игра

Для перехода в саму игру пользователь, находясь в главном меню, должен нажать кнопку «Новая игра» или «Продолжить», если у него уже имеется сохранение. После нажатия на кнопку, пользователь переходит на экран загрузки (Рис. 6.18), который автоматически переключиться в игровой мир, после окончания настройки и загрузке всех ресурсов. Длительность загрузки зависит от производительности устройства.



Рис. 6.18. Экран загрузки игры

Функциональный код загрузочного экрана выглядит следующим образом (Рис. 6.19). В коде реализованы подсказки, в виде случайного элемента взятого из массива и динамическая подгрузка требуемого файла сцены, с помощью функции \_process().

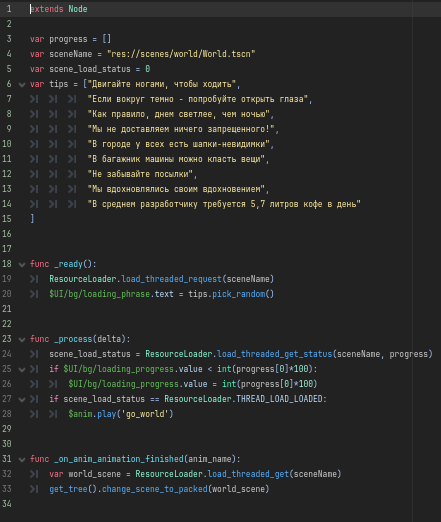


Рис. 6.19. Экран загрузки игры - Код

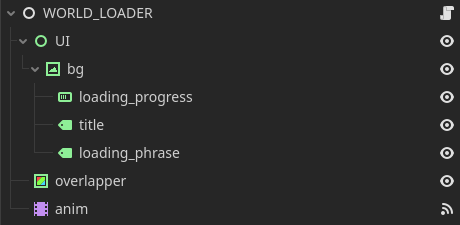


Рис. 6.20. Экран загрузки игры - Иерархия узлов

## Диаграммы UML

Диаграмма USE-CASE:

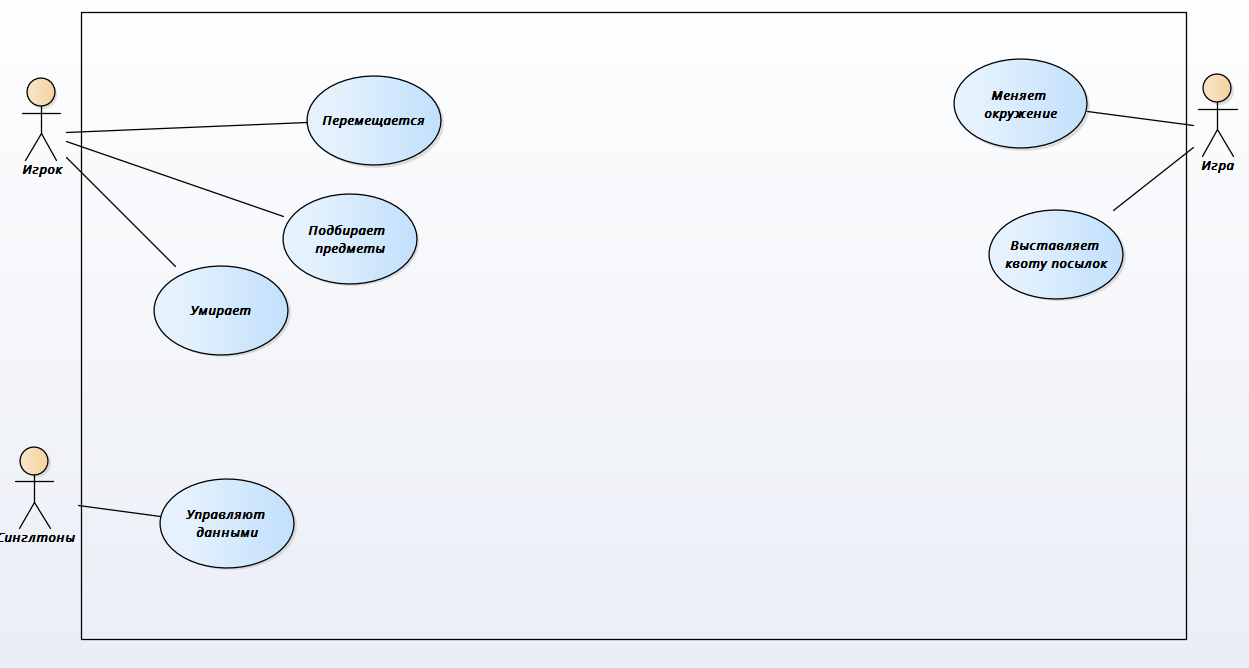


Рис. 6.21. Диаграмма вариантов использования

Диаграмма классов:

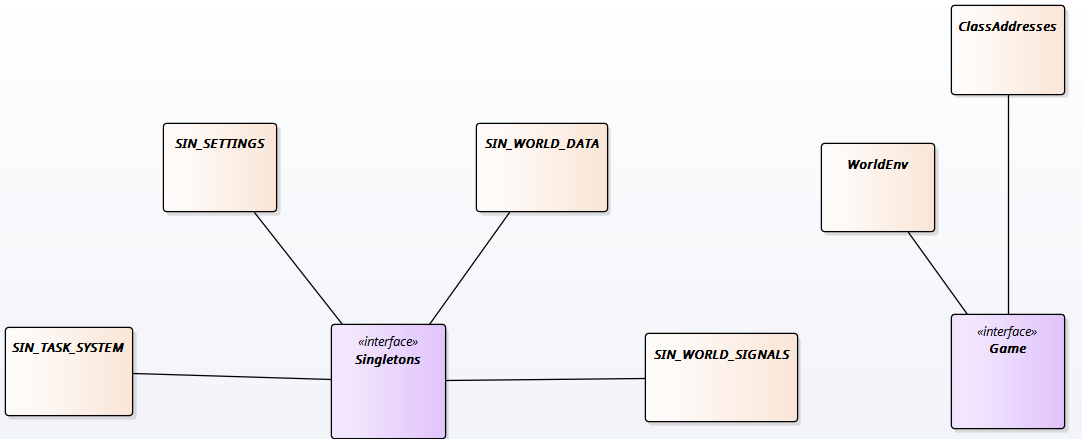


Рис. 6.22. Диаграмма классов и интерфейсов

Диаграмма компонентов:

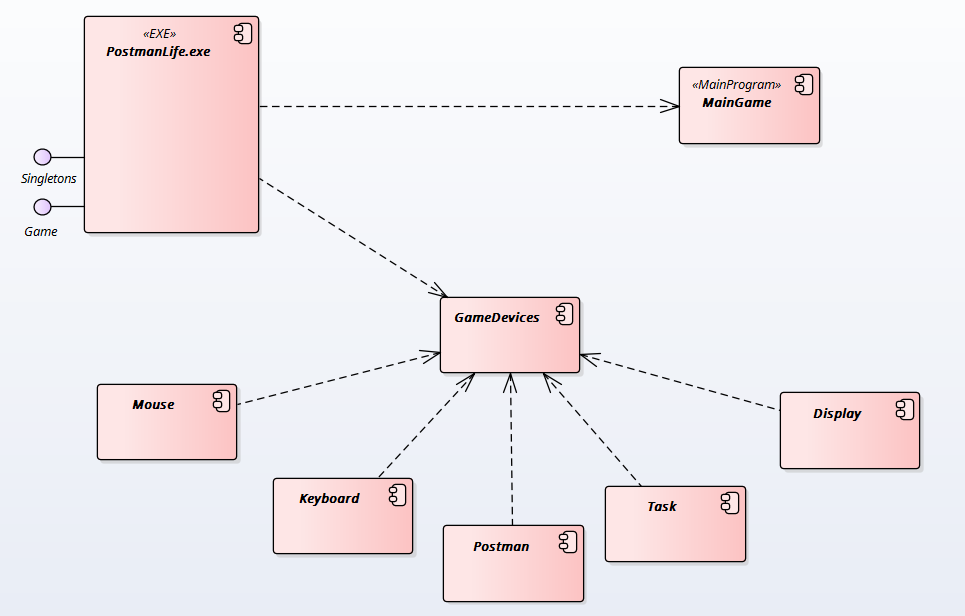


Рис. 6.23. Диаграмма компонентов

Диаграмма бизнес-логики доставки посылки:

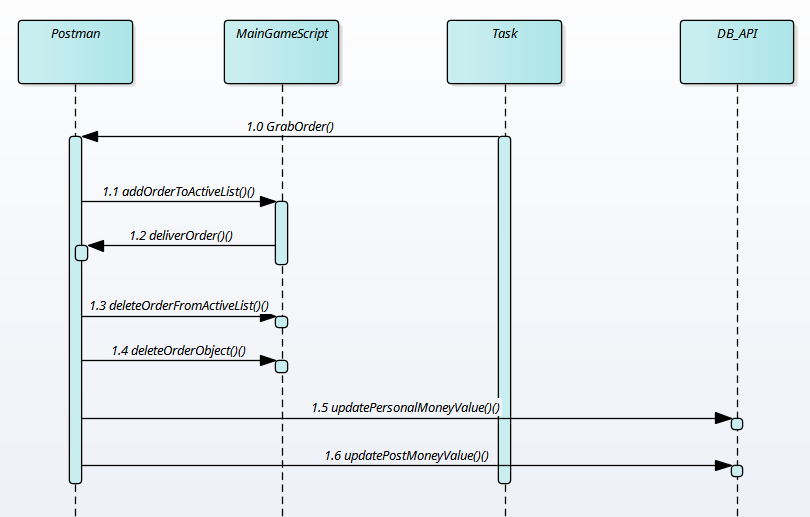


Рис. 6.24. Диаграмма бизнес-логики доставки посылки

Диаграмма бизнес-логики управления персонажем:

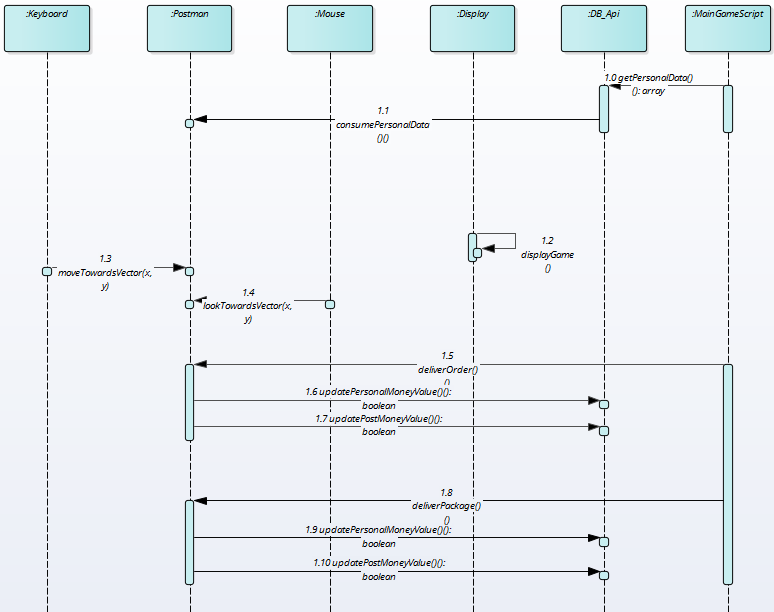


Рис. 6.25. Диаграмма бизнес-логики управления персонажем

# Заключение

В результате работы над расчетно-графической работой была создана игра жанра «Симулятор» - «Жизнь почтальона». Игра имеет дружественный интерфейс, широкие настройки графики и стабильную производительность.

Стоит отметить, что в силу сроков на данный момент в приложении реализован не весь задуманный функционал, а только базовый и наиболее важный. Тем не менее, этого хватает для полноценной игры.

В ходе работы на расчетно-графической работой были укреплены знания по работе с системой управления приложением и контроля версий - GitHub для организации и контроля процесса разработки.

Также были получены навыки работы со средой разработки Godot Engine и средой построения UML-диаграмм Enterprise Architect

# Приложение

Листинг класса...