

**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 3](#_Toc15188)

[1. Техническое задание 5](#_Toc21290)

[2. Выполнение технического задания 6](#_Toc1699)

[2.1. Логика живых объектов 6](#_Toc25405)

[2.2. Создание скриптов окружения 6](#_Toc28816)

[2.3. Создание скриптов объектов взаимодействия 7](#_Toc1695)

[2.3.1. Создать заготовки объектов взаимодействия 7](#_Toc21518)

[2.4. Оптимизация игрового мира 8](#_Toc8156)

[3. Руководство программиста 10](#_Toc14158)

[3.1. Игровые настройки 11](#_Toc31444)

[3.2. Игра 18](#_Toc25437)

[3.3. Диаграммы UML 20](#_Toc24164)

[Заключение 23](#_Toc9631)

# 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.

# Техническое задание

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:

* Создать заготовку игрового 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

# Выполнение технического задания

## Логика живых объектов

### 2.1.1 Создать заготовки живых объектов

Была написана логика всех живых объектов для игры «Жизнь почтальона» (Рис. 2.1.1.1).

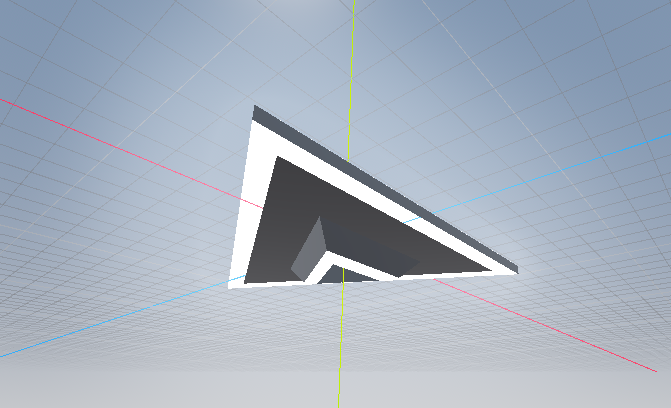


Рис. 2.1.1.1 Заготовка НЛО в игре «Жизнь почтальона»

### 2.1.2 Реализовать функционал

Была написана логика всех живых объектов для игры «Жизнь почтальона» (Рис. 2.1.2.1).

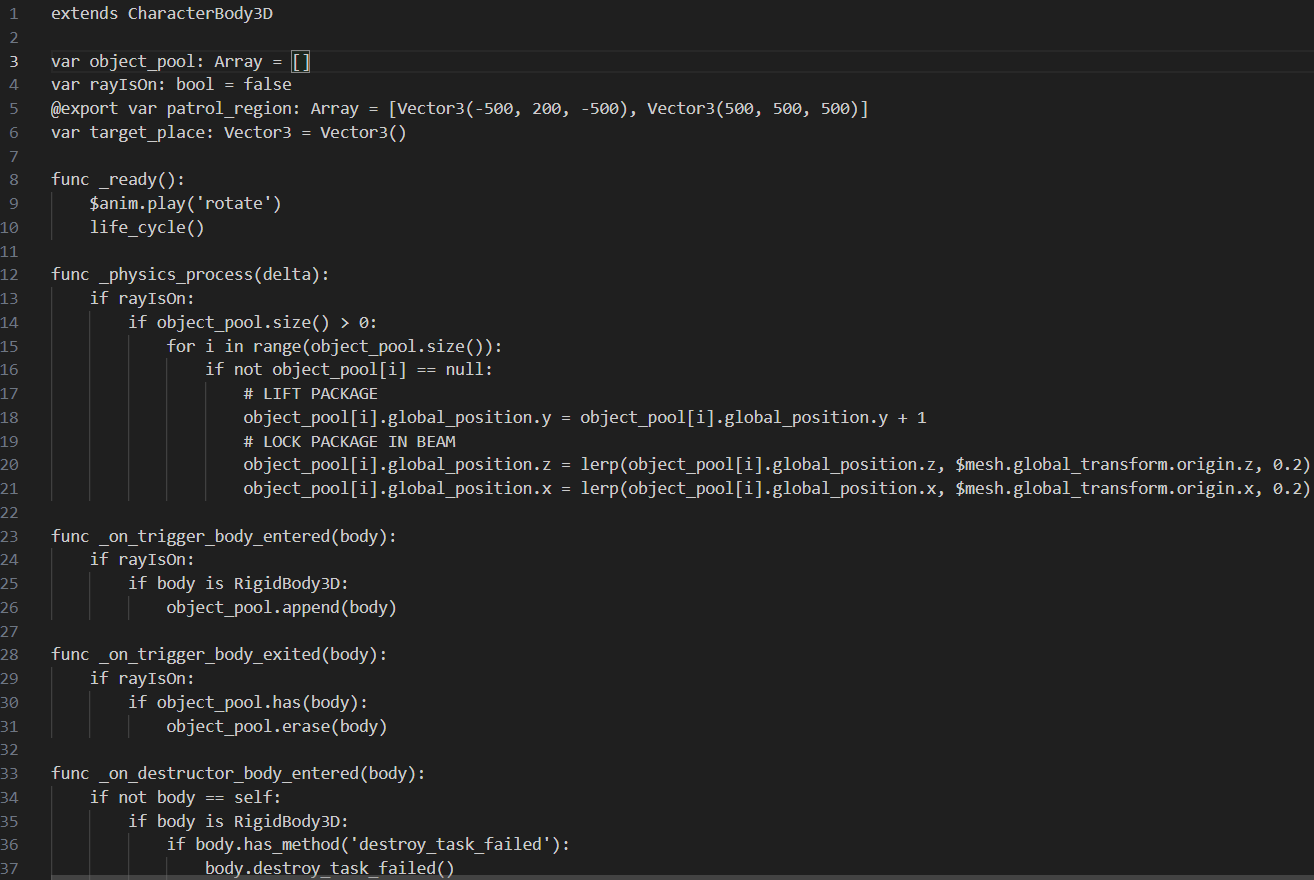


Рис. 2.1.2.1. Код НЛО в игре «Жизнь почтальона»

### 2.1.3 Подключить к мировой системе

Была написана логика подключения всех живых объектов к мировой системе для игры «Жизнь почтальона».

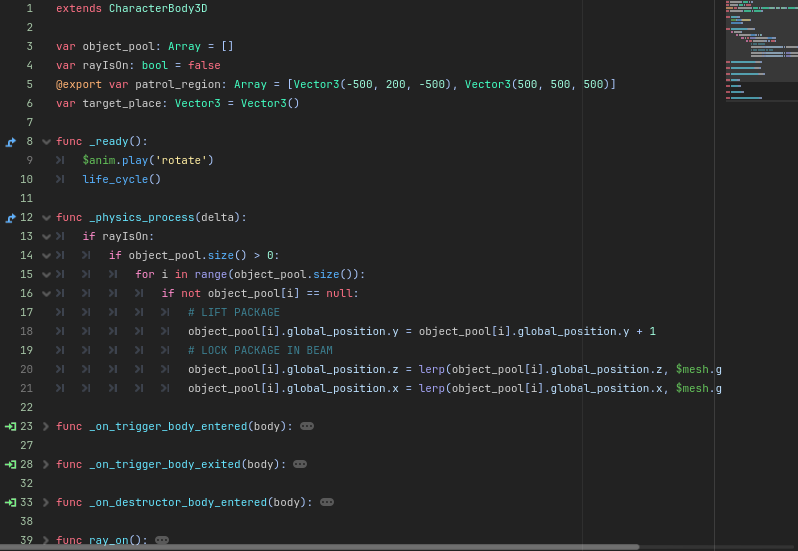


Рис. 2.1.3.1 - Код НЛО в игре «Жизнь почтальона»

Всего подобным образом было разработано 7 различных живых существ: Собака, Корова, Курица, НЛО, Гигантское НЛО, Статуя Грифона, Призрак.

## 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.

### [2.2.1 Создать заготовку игрового окружения](#_Toc22522)

На рисунке 2.2.1.1 изображена заготовка окружения в игре «Жизнь почтальона».

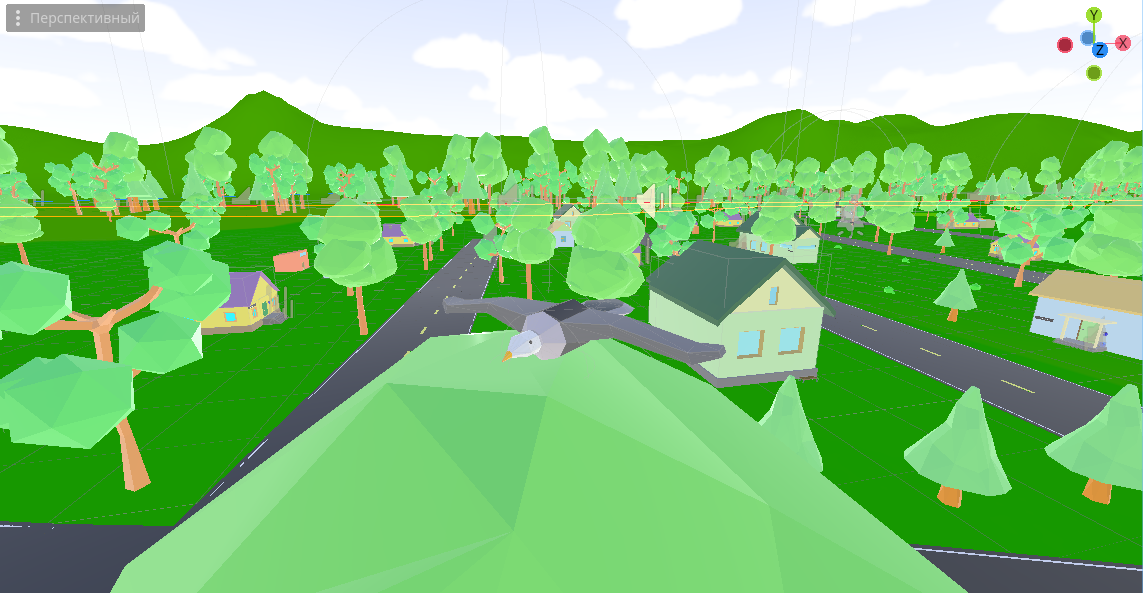


Рис. 2.2.1.1. Заготовка окружения в игре «Жизнь почтальона»

### [2.2.2 Реализовать функционал игрового окружения](#_Toc21729)

На рисунке 2.2.2.1 отражен главный функционал окружения в игре «Жизнь почтальона» - смена дня и ночи, в виде анимации.

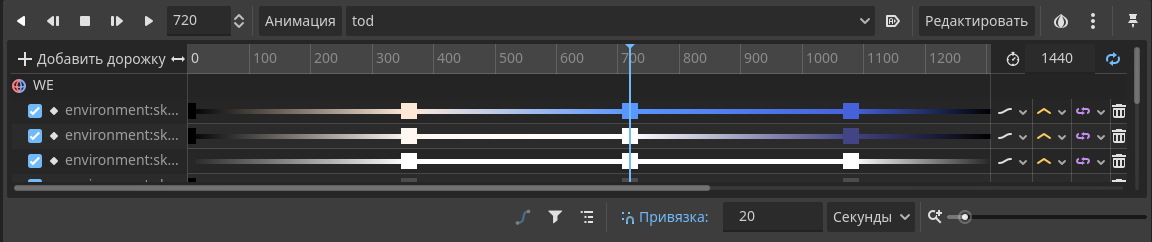


Рис. 2.2.2.1. Функционал окружения в игре «Жизнь почтальона»

### [2.2.3 Создать скрипт окружения с общемировой системой](#_Toc6105)

Была написана логика окружения для игры «Жизнь почтальона».



Рис. 2.2.3.1 Код окружения в игре «Жизнь почтальона»

## Создание скриптов объектов взаимодействия

## [2.3.1. Создать заготовки объектов взаимодействия](#_Toc13646)

Были написаны скрипты для объектов взаимодействия для игры «Жизнь почтальона».

Для примера, так выглядит заготовка подбираемого предмета - «Молочка»:

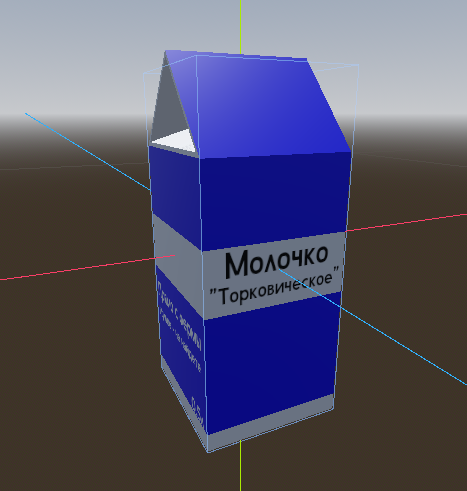


Рис. 2.3.1.1 Заготовка «Молочка» в игре «Жизнь почтальона»

Всего было разработано 8 различных подбираемых предметов.

### [2.3.2. Реализовать функционал объектов взаимодействия](#_Toc5762)

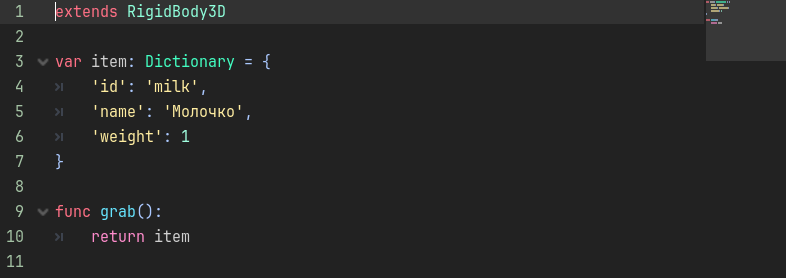


Рис. 2.3.2.1. Код окружения в игре «Жизнь почтальона»

### [2.3.3. Наладить связь между объектами взаимодействия, где это требуется](#_Toc20573).

Налажено общение объектов между собой с помощью сигналов (Рис 2.3.3.1):

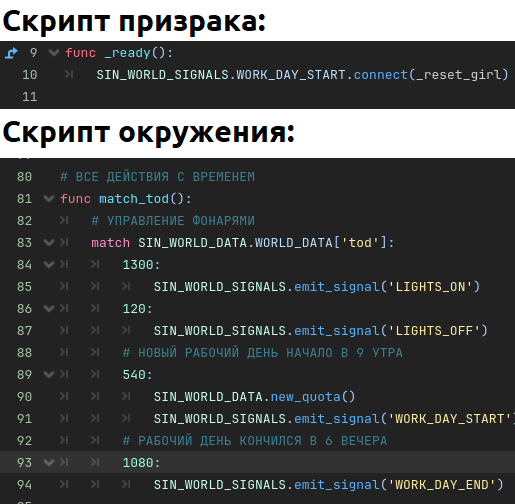


Рис. 2.3.3.1 Связь призрака с окружением в игре «Жизнь почтальона»

## Оптимизация игрового мира

### [2.4.1 Найти проблемные области 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 рисунке 2.4.1.1.

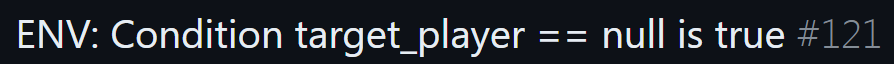


Рис. 2.4.1.1 Пример ошибки, отражающей неэффективный код

### [2.4.2 Измерить влияние и существенность проблемы на игровой мир и его системы](#_Toc30307)

Измерение существенности влияния ошибки (Рис 2.4.2.1).

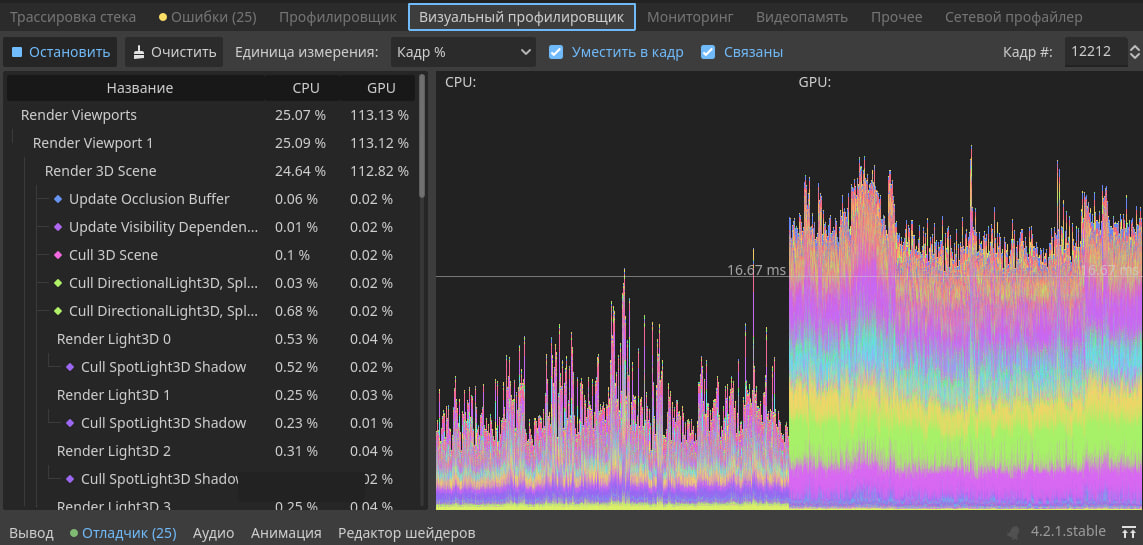


Рис. 2.4.2.1. Измерение существенности влияния ошибки

### [2.4.3. Оптимизировать проблемную часть кода](#_Toc20508)

Исправлением неэффективного кода стало избавление логики от постоянного обновления переменных и в целом удаления конечного автомата (Рис 2.4.3.1).

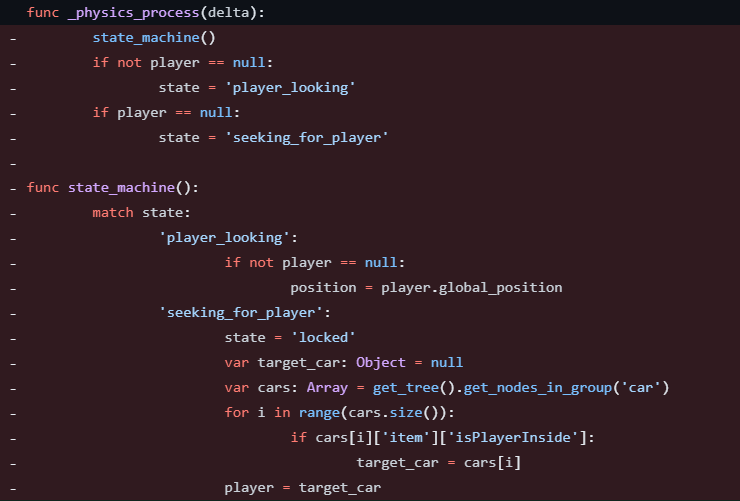


Рис. 2.4.3.1. Пример решения ошибки, исправления неэффективного кода

# Руководство программиста

При загрузке приложение появляется главное меню (Рис. 3.1), из которого пользователь может перейти в саму игру, ее настройки или выйти из игры.



Рис. 3.1. 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. 3.2. Меню приложения - иерархия узлов

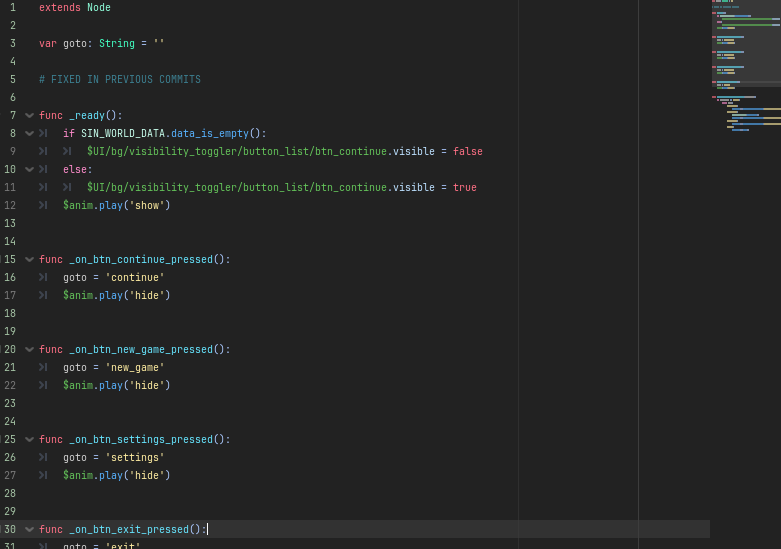


Рис. 3.3. Меню приложения - код

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 3.4). Здесь пользователь может настроить общую громкость звуков и отдельно громкость музыки.

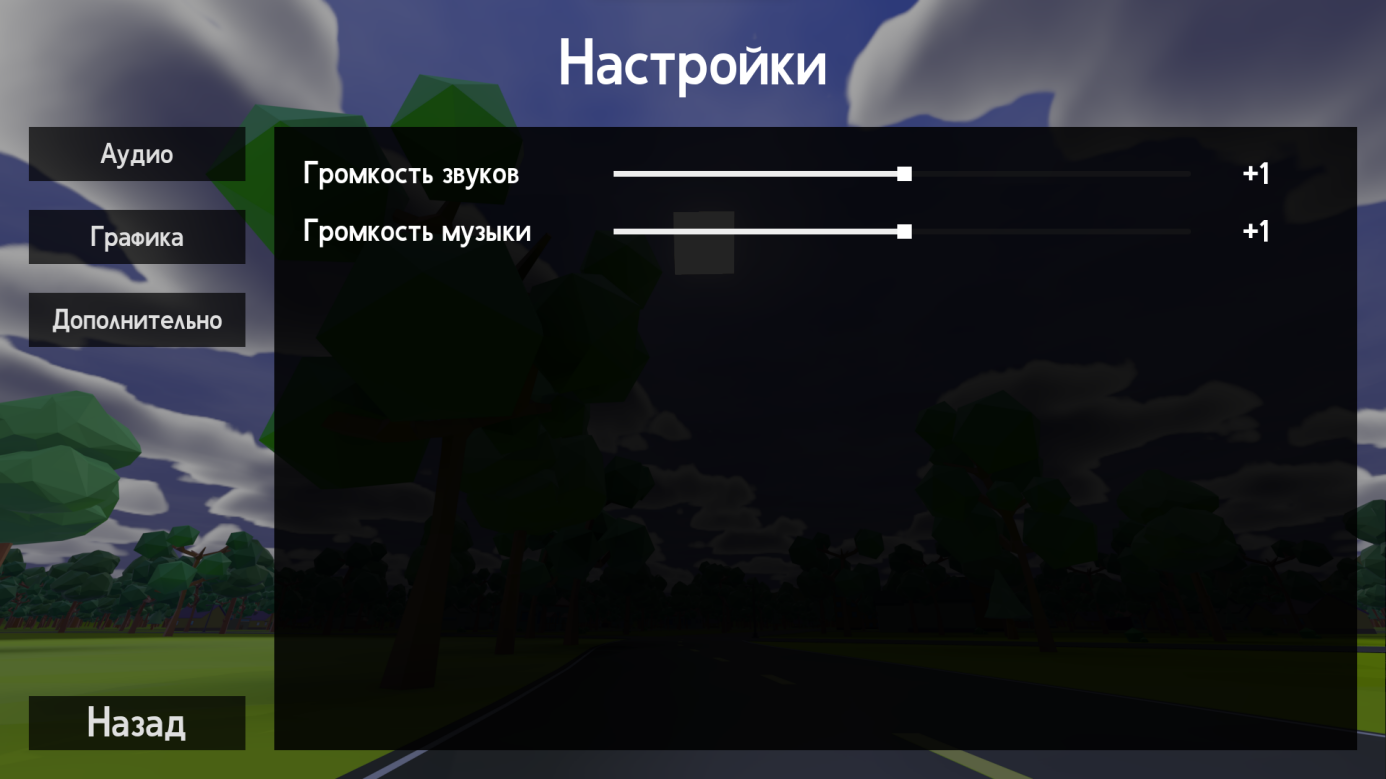


Рис. 3.4. Раздел «Аудио» в настройках приложения

Таким образом выглядит окно «Настройки» в иерархии узлов (Рис 3.5).

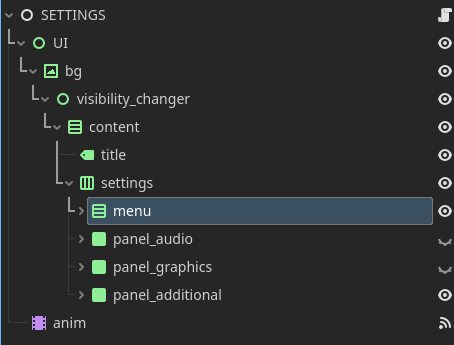


Рис. 3.5. Настройки - иерархия узлов

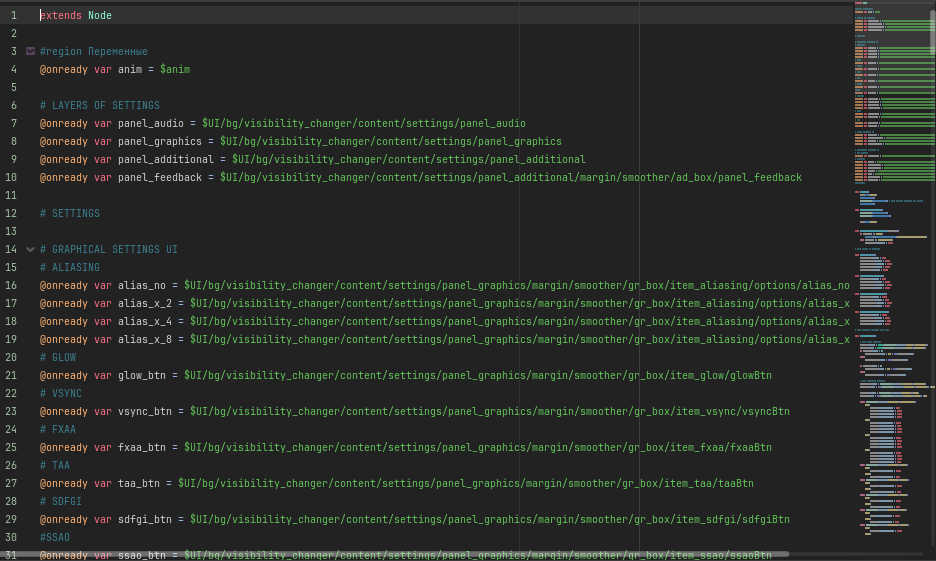


Рис. 3.6. Настройки Аудио - код

В разделе «Графика» (Рис 3.7) у пользователя есть возможность детально настроить внутриигровую графику. Увеличение значений настроек может способствовать понижению производительности и FPS внутри игры.

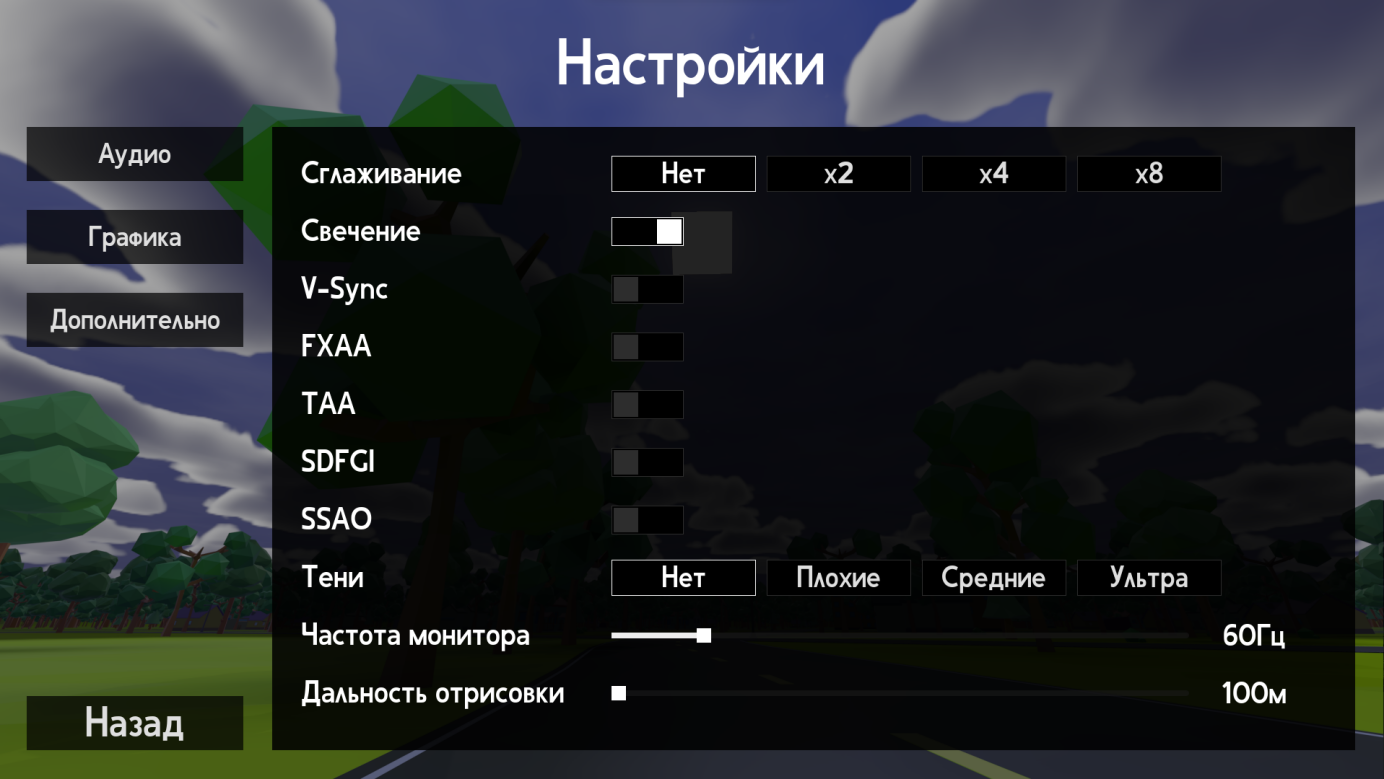


Рис. 3.7. Раздел «Графика» в настройках приложения

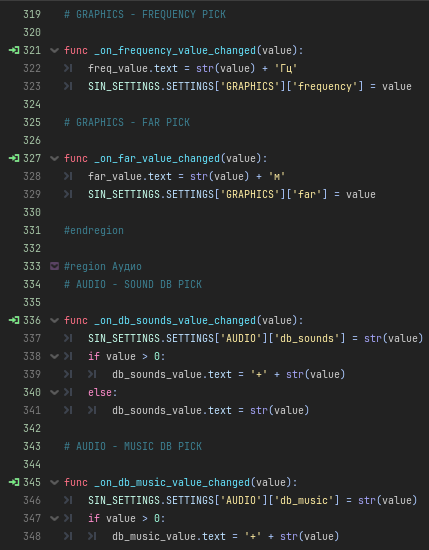


Рис. 3.8. Настройки Графики - код

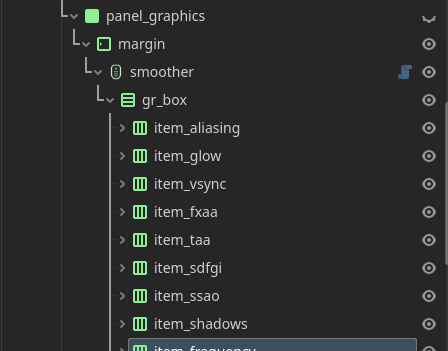


Рис. 3.9. Настройки Графики - иерархия узлов

В разделе «Дополнительно» (Рис 3.10) есть возможность включить счетчик кадров (FPS) и отправить обращение разработчикам через специальную форму (Рис 3.12). 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.

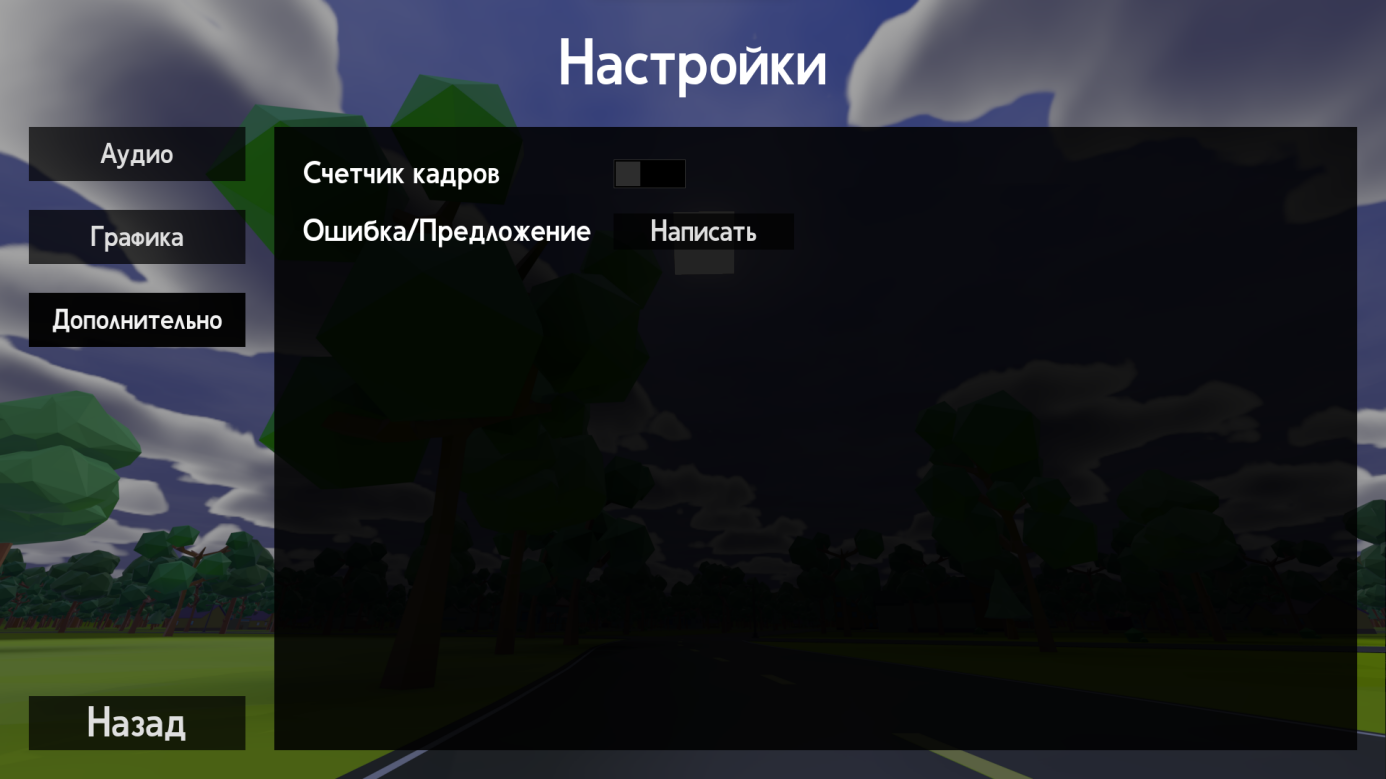


Рис. 3.10. Раздел «Дополнительно» в настройках приложения

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

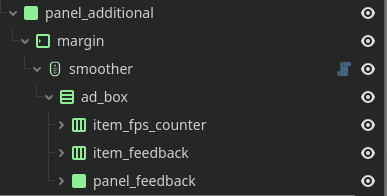


Рис. 3.11. Настройки Графики - иерархия узлов

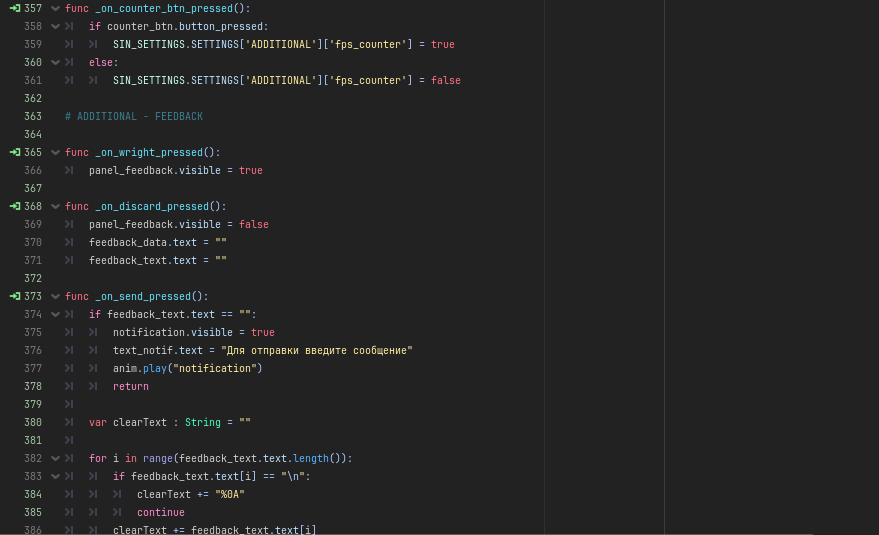


Рис. 3.12. Настройки «Дополнительно» - код

Визуально форма обращения к разработчикам выглядит следующи

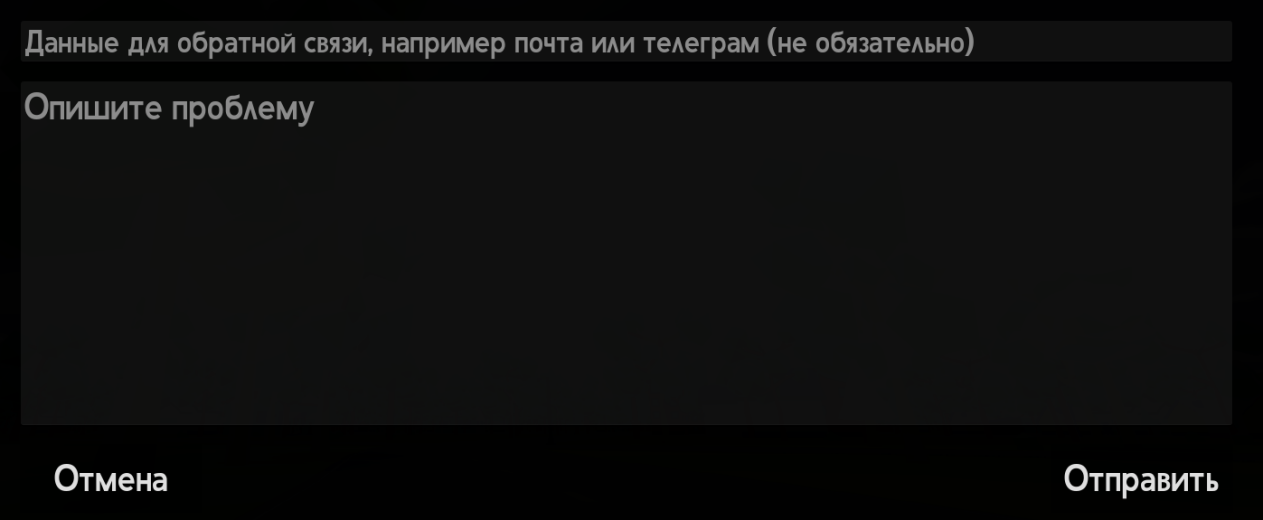


Рис. 3.13. Форма отправки обращения к разработчикам

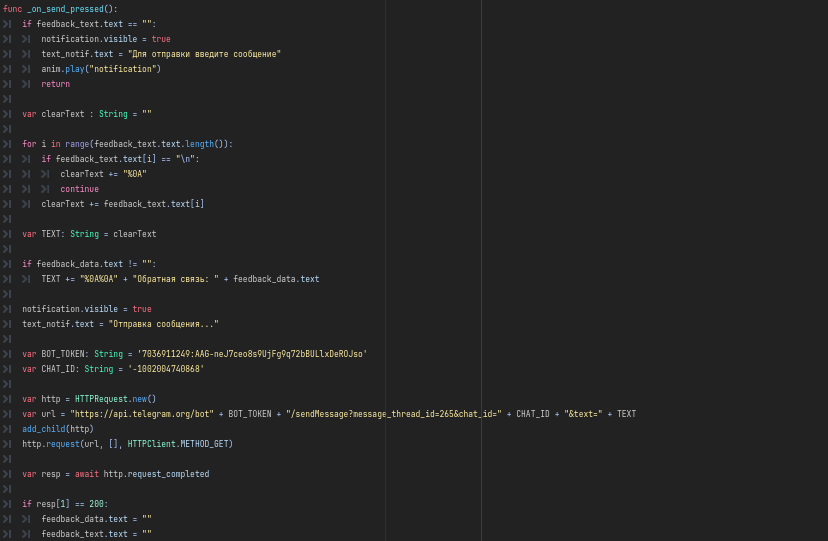


Рис. 3.14. Форма отправки обращения к разработчикам - код

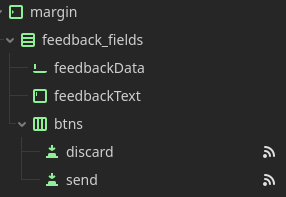


Рис. 3.15. Форма отправки обращения к разработчикам - иерархия узлов

Все настройки графики, аудио и дополнительные - сохраняются в отдельный файл- settings.dat (Рис 3.16).



Рис. 3.16. Файл, содержащий настройки

В коде сохранение настроек выглядит следующим образом (Рис 3.17).

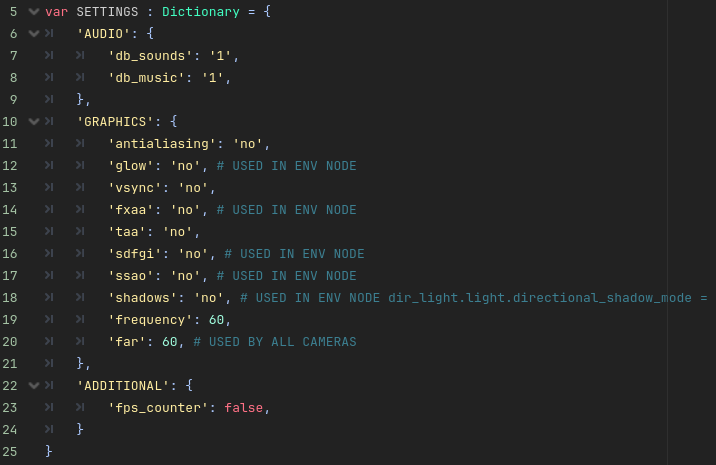


Рис. 3.17. Словарь значений файла «Настройки»

## 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. 3.16), который автоматически переключиться в игровой мир, после окончания настройки и загрузке всех ресурсов. Длительность загрузки зависит от производительности устройства.



Рис. 3.16. Экран загрузки игры

Функциональный код загрузочного экрана выглядит следующим образом (Рис. 3.17). В коде реализованы подсказки, в виде случайного элемента взятого из массива и динамическая подгрузка требуемого файла сцены, с помощью функции \_process().

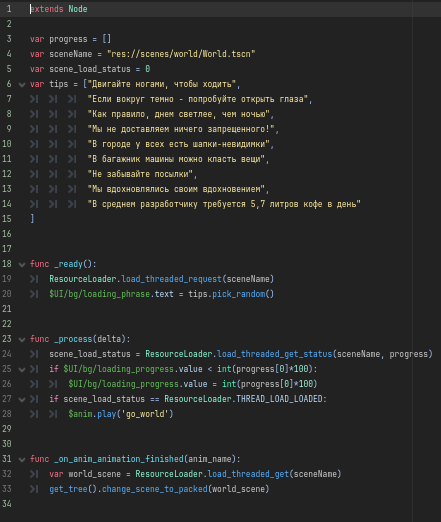


Рис. 3.17. Экран загрузки игры - Код

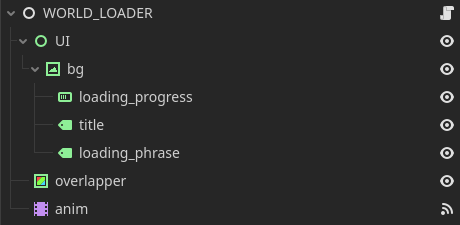


Рис. 3.18. Экран загрузки игры - Иерархия узлов

## Диаграммы UML

Диаграмма USE-CASE:

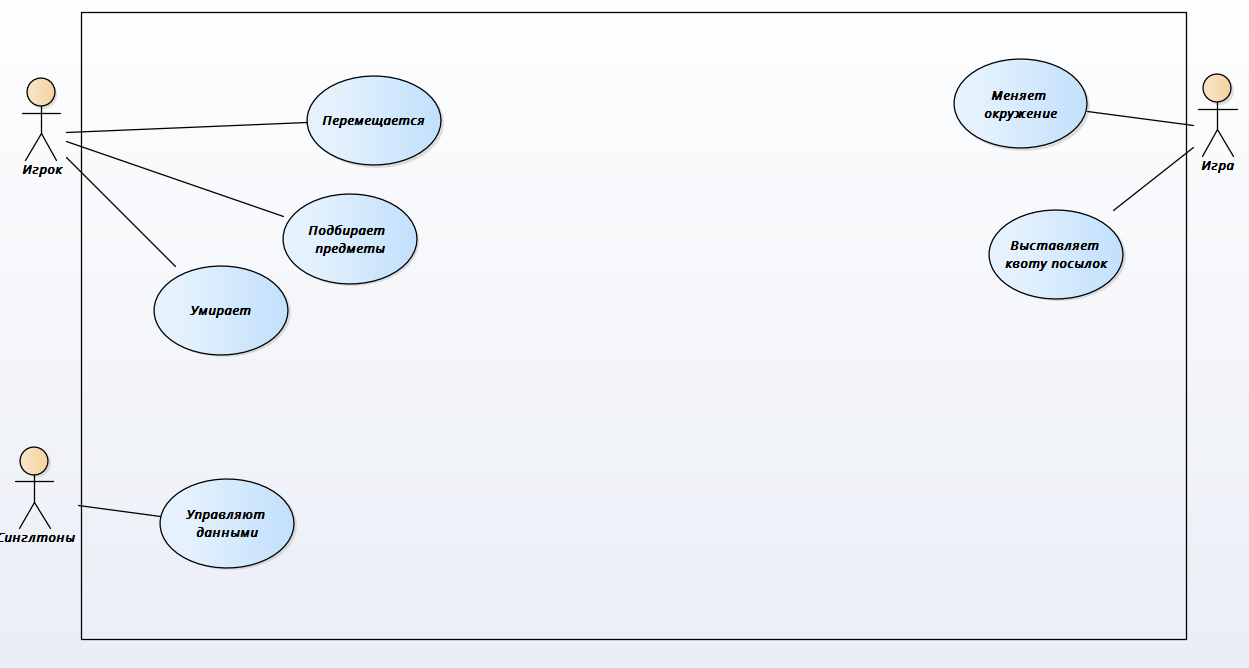


Рис. 3.18. Диаграмма вариантов использования

Диаграмма классов:

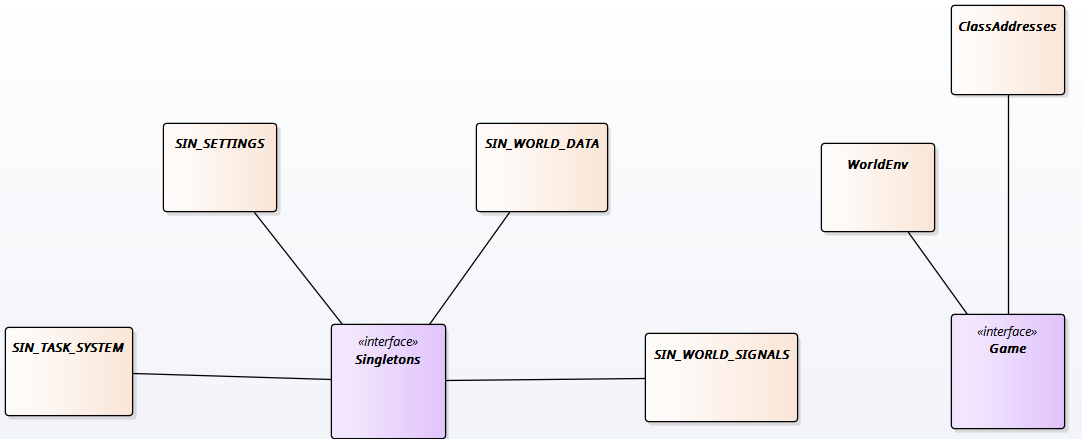


Рис. 3.19. Диаграмма классов и интерфейсов

Диаграмма компонентов:

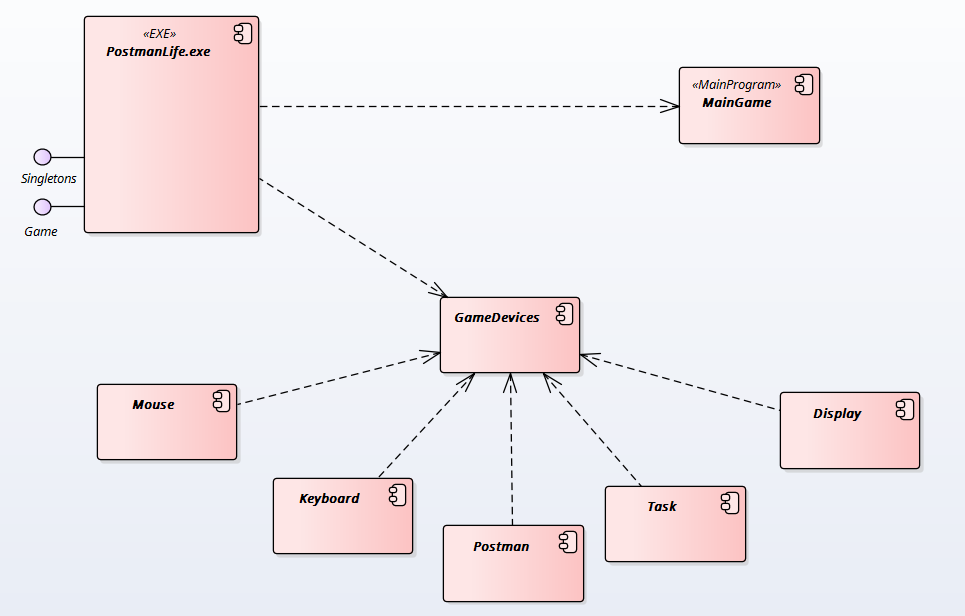


Рис. 3.20. Диаграмма компонентов

Диаграмма бизнес-логики доставки посылки:

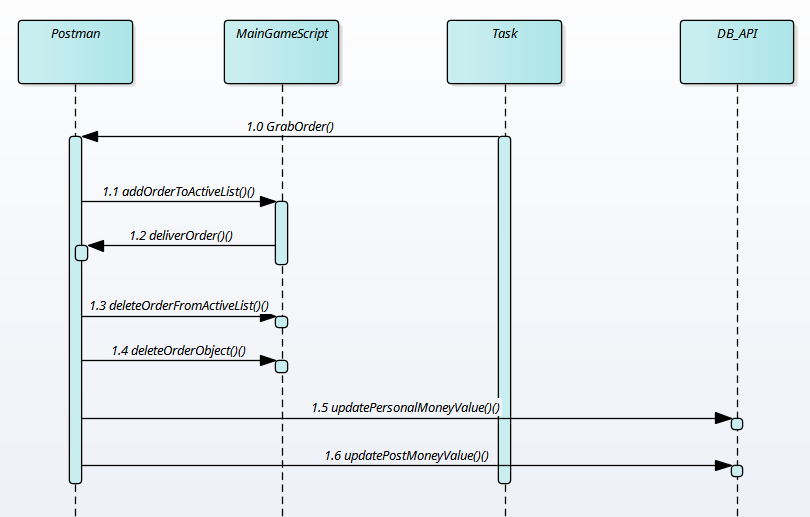


Рис. 3.21. Диаграмма бизнес-логики доставки посылки

Диаграмма бизнес-логики управления персонажем:

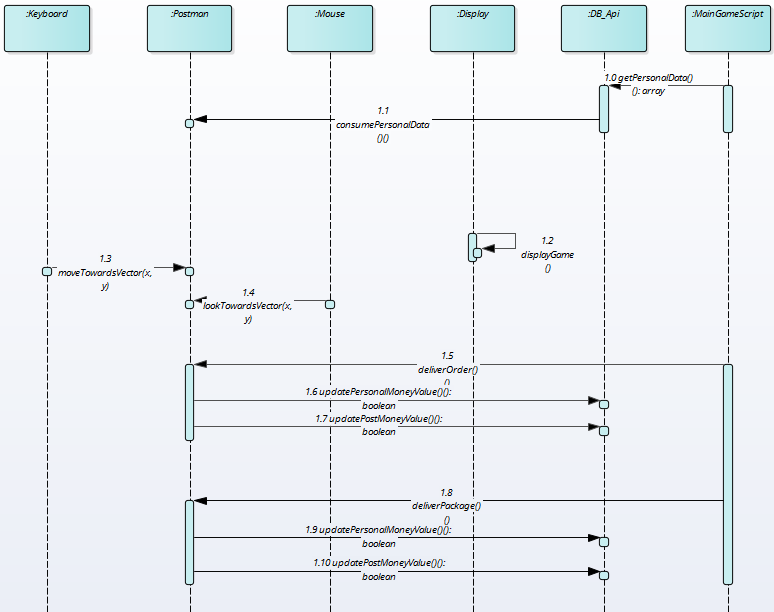


Рис. 3.22. Диаграмма бизнес-логики управления персонажем

# 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 мной 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