МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

(ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение»

Направление подготовки 09.03.04 «Программная инженерия»

ОТЧЕТ ПО ПРАКТИКЕ

|  |
| --- |
|  |
| *(наименование типа практики)* |
| *(полное наименование профильной организации)*  Выполнил обучающийся /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / |
| *(подпись) (ФИО, курс, номер группы)*  Дата сдачи отчета: « » 201 г. |
| Дата аттестации « » 201 г. |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Руководитель практики от  ИЖГТУ имени М.Т. Калашникова / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ |
| *(подпись) (И.О. Фамилия, должность, ученая степень)*  Заведующий кафедрой /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ |
| *(подпись) (И.О. Фамилия, должность, ученая степень)* |

Оглавление

[Введение 2](#_Toc168508566)

[Основная часть 4](#_Toc168508567)

[Описание учебного проекта 4](#_Toc168508568)

[Математическая постановка 5](#_Toc168508569)

[Описание программной реализации 6](#_Toc168508570)

[Код программы, комментарии и пример работы 8](#_Toc168508571)

[Реализация сцены с игрой: 8](#_Toc168508572)

[Реализация сцены главного меню: 11](#_Toc168508573)

[Пример работы 12](#_Toc168508574)

[Заключение 13](#_Toc168508575)

[Список литературы 14](#_Toc168508576)

# Введение

SpaceShooter — это стрелялка для iOS, которая бросает вызов игрокам в уничтожении пришельцев. Игра отличается возможностью отслеживать гироскоп телефона для управления космическим кораблем, которые делают ее увлекательной и захватывающей.

**Цель:**

• Разработать двухмерную стрелялку в стиле SpaceShooter для iOS-устройств с использованием языка программирования Swift.

**Задачи:**

1. Создать простую игру для iOS с использованием Swift и SwiftUI.
2. Применить основные концепции Swift, такие как типы данных, переменные, константы и функции.
3. Изучить основы SwiftUI, такие как создание пользовательского интерфейса, привязка данных и модификаторы.
4. Понять жизненный цикл приложения SwiftUI.
5. Реализовать игровой цикл, включая обновление состояния игры и обработку пользовательского ввода.
6. Попрактиковаться в отладке и решении проблем с кодом.
7. Получить практический опыт разработки приложений для iOS.

# Основная часть

Описание учебного проекта

*Цель:*

• Разработать двухмерную стрелялку в стиле SpaceShooter для iOS-устройств с использованием языка программирования Swift.

*Ожидаемый результат:*

Игра будет иметь следующие особенности:

1. Игрок управляет космическим кораблем.
2. Космический корабль стреляет по вражеским кораблям.
3. Игра генерирует бесконечное количество вражеских кораблей.
4. Игра отслеживает счет и выводит его на экран.
5. Игра имеет привлекательную графику и звуковые эффекты.

*Геймплей:*

Игрок берёт на себя роль пилота космического корабля, который должен защитить Землю от вторжения инопланетян. Космический корабль оснащен мощным оружием, которое можно использовать для уничтожения вражеских кораблей.

Игрок может выбрать сложность игры в зависимости от своих навыков. Игра генерирует бесконечное количество вражеских кораблей, гарантируя, что действие никогда не закончится. Игроки должны использовать свои навыки и стратегию, чтобы не проиграть и набрать как можно больше очков.

Математическая постановка

Код демонстрирует использование следующих концепций, связанных с физикой:

1. SpriteKit использует движок физики, основанный на Box2D, который позволяет объектам в игре взаимодействовать с физическим миром (например, гравитация).
2. Код использует Core Motion Framework для получения данных об ускорении устройства. Это позволяет игроку управлять космическим кораблем, наклоняя устройство.

Также в коде используются следующие утверждения и константы:

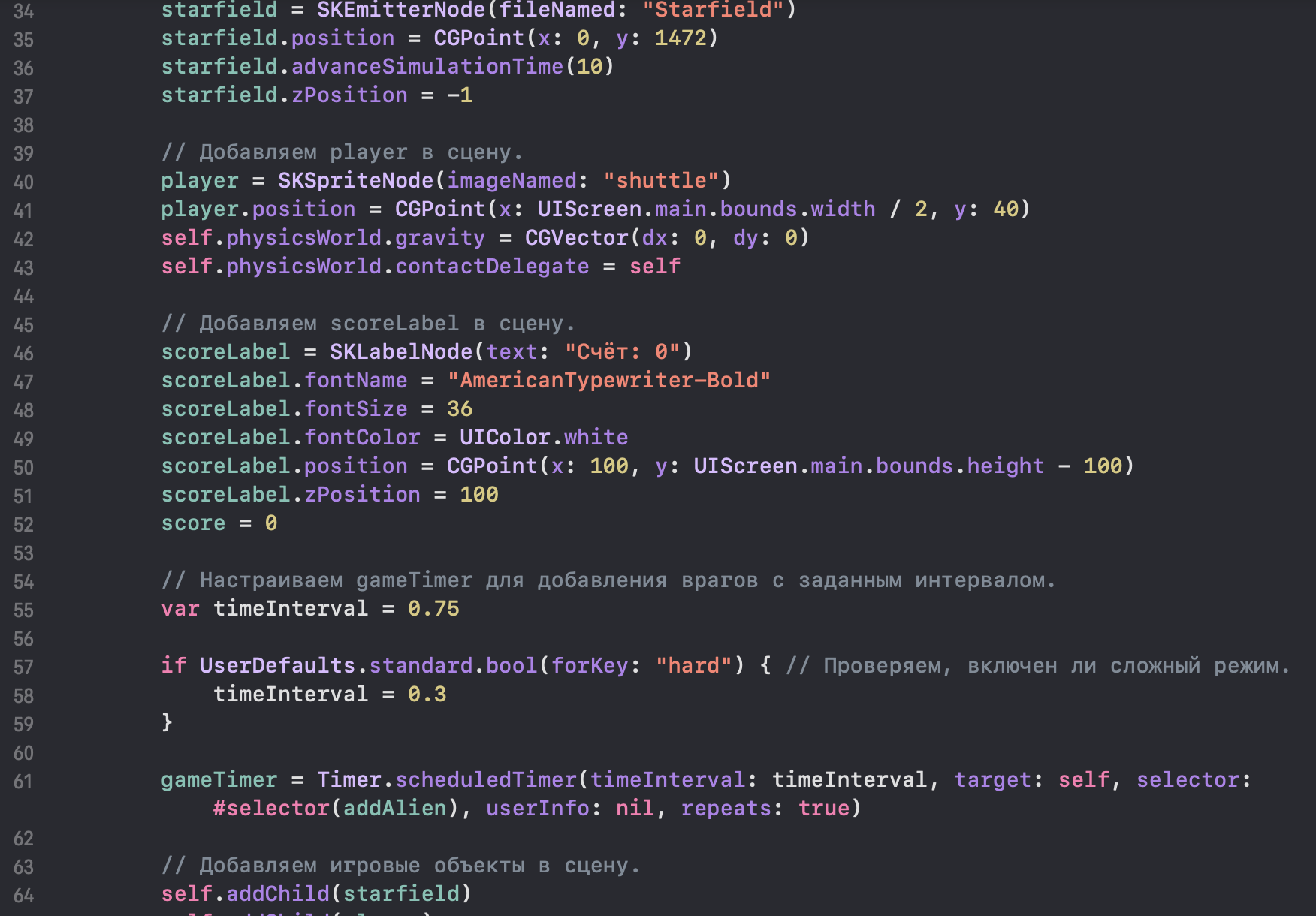
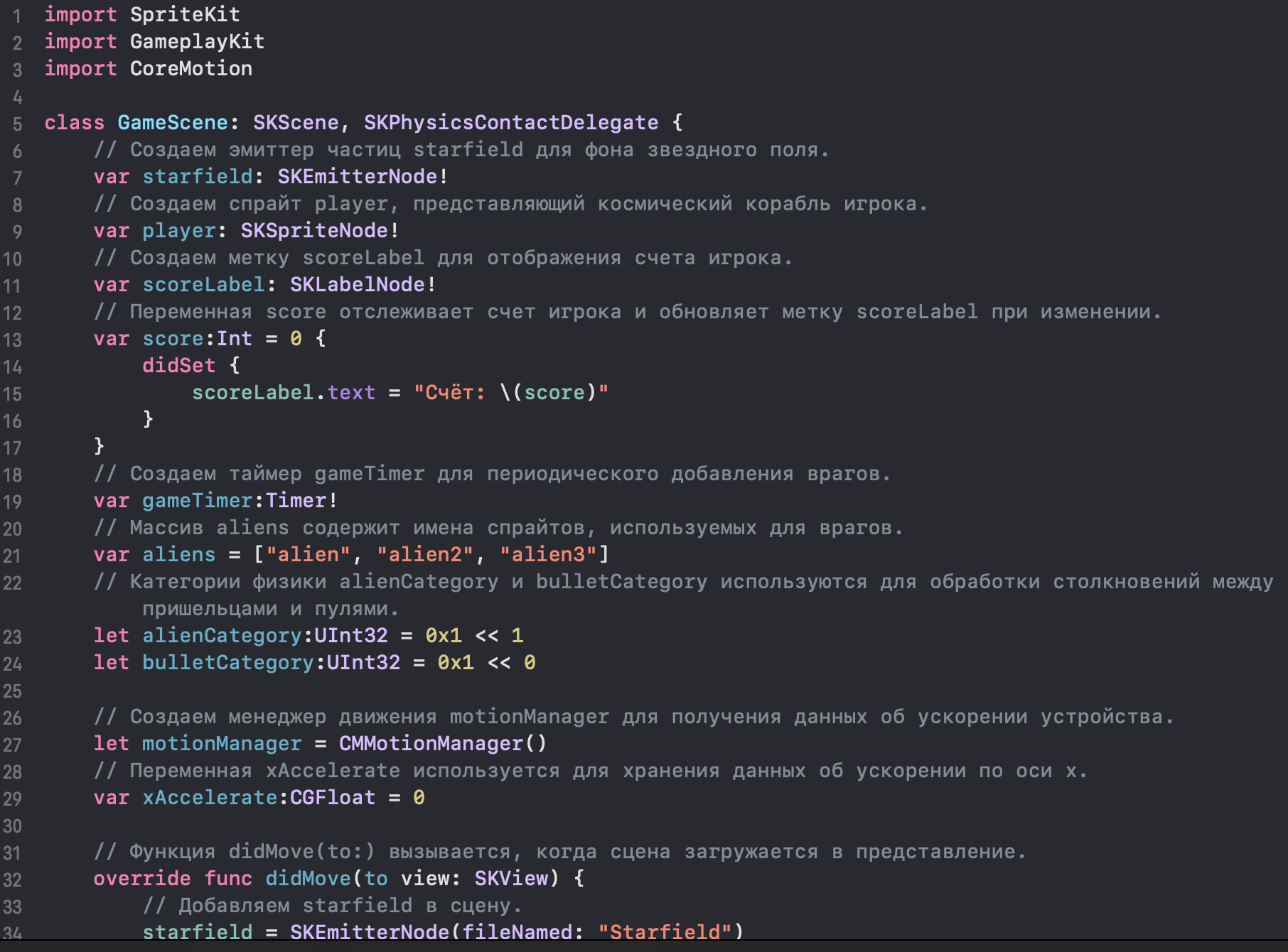
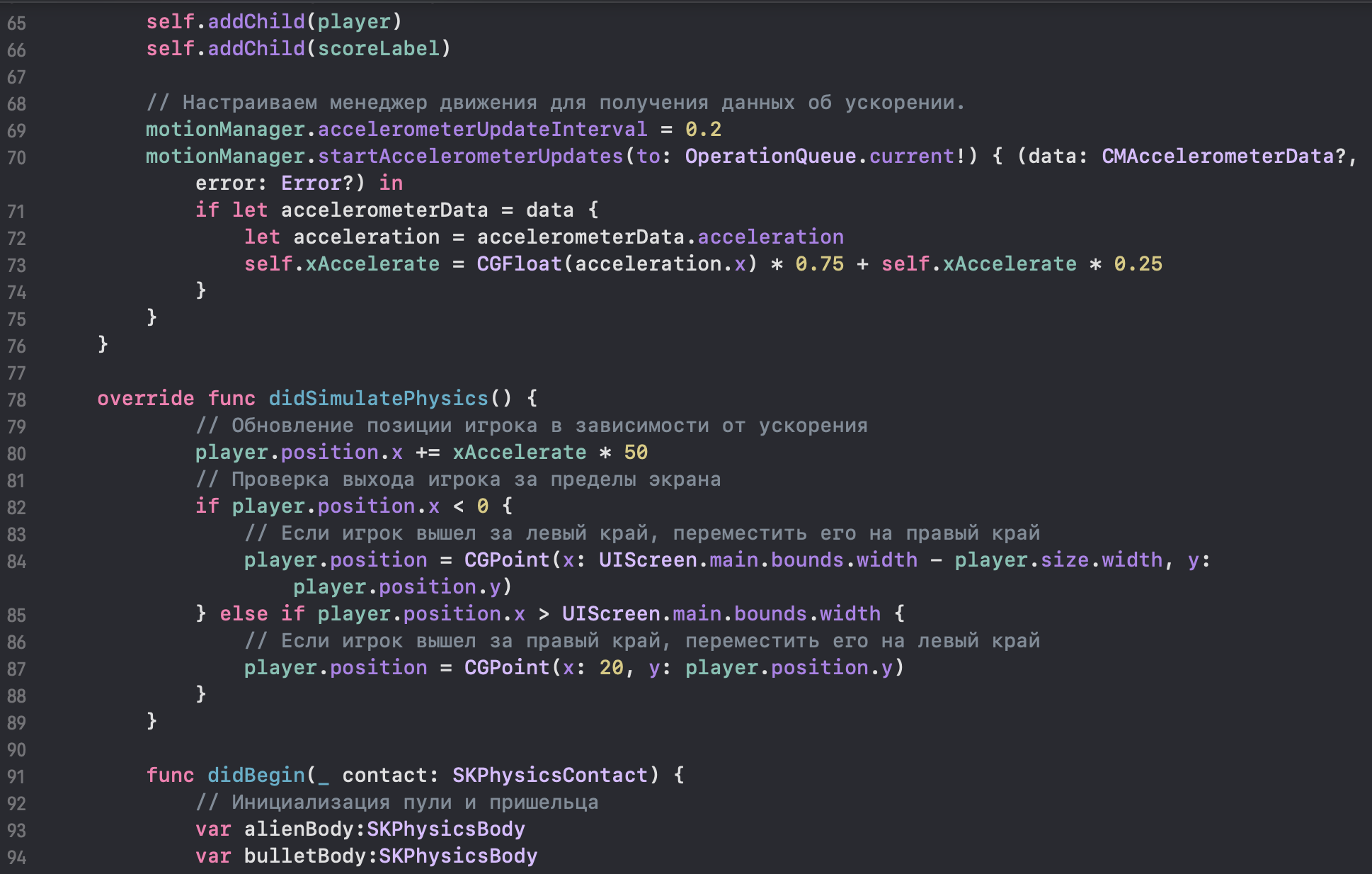
1. self.physicsWorld.gravity = CGVector(dx: 0, dy: 0). Отключает гравитацию в игровом мире, что позволяет космическому кораблю свободно двигаться в космосе.
2. alienCategory:UInt32 = 0x1 << 1 и bulletCategory:UInt32 = 0x1 << 0. Эти константы используются для определения категорий физических тел для инопланетян и пуль, соответственно. Это позволяет игре обнаруживать столкновения между инопланетянами и пулями.
3. motionManager.accelerometerUpdateInterval = 0.2. Устанавливает интервал обновления данных акселерометра на 0,2 секунды, обеспечивая плавное управление космическим кораблем.
4. player.position.x += xAccelerate \* 50. Эта строка кода обновляет положение космического корабля по оси X на основе ускорения, умноженного на константу 50. Это позволяет игроку контролировать скорость и направление движения космического корабля.

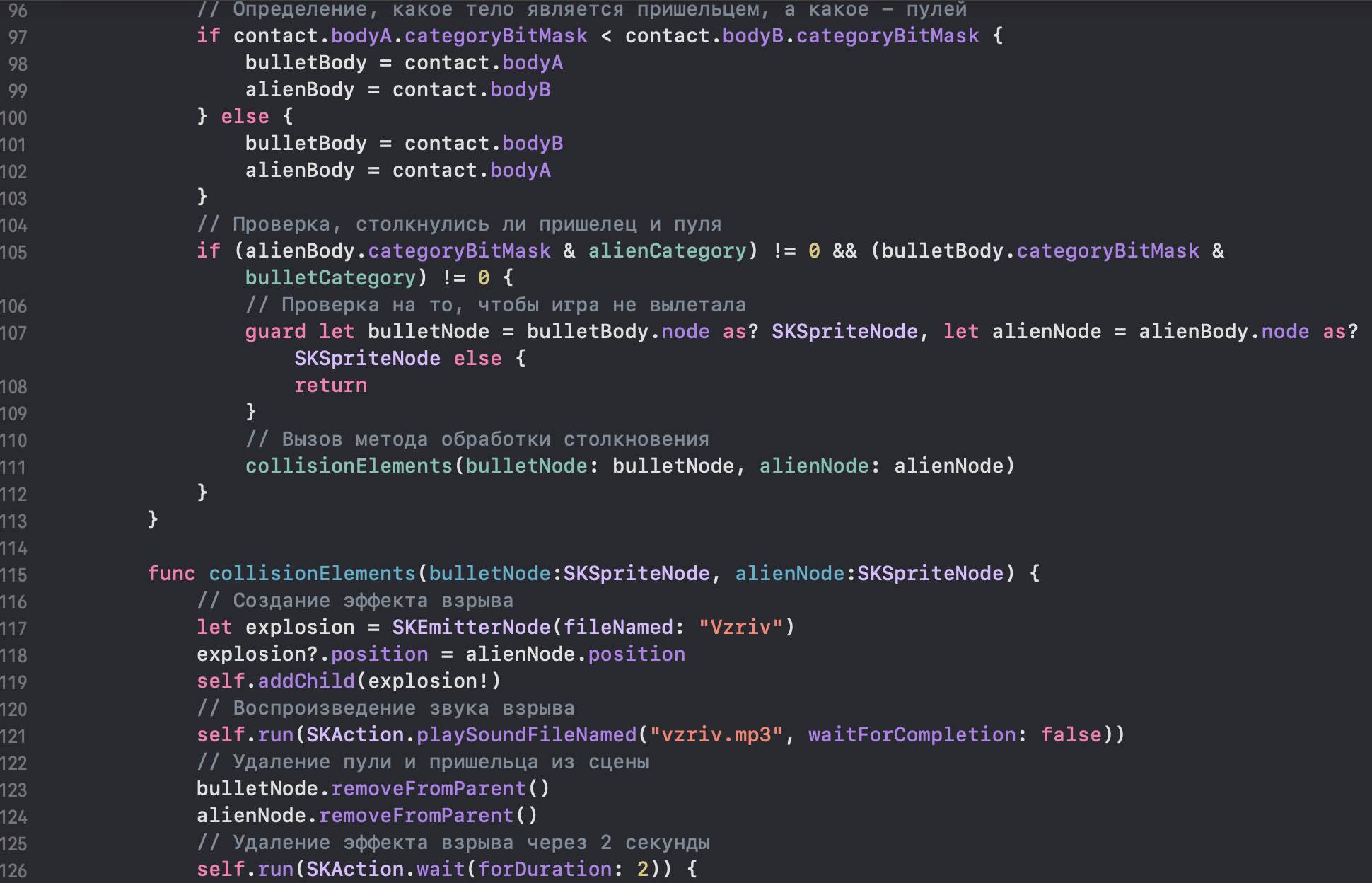
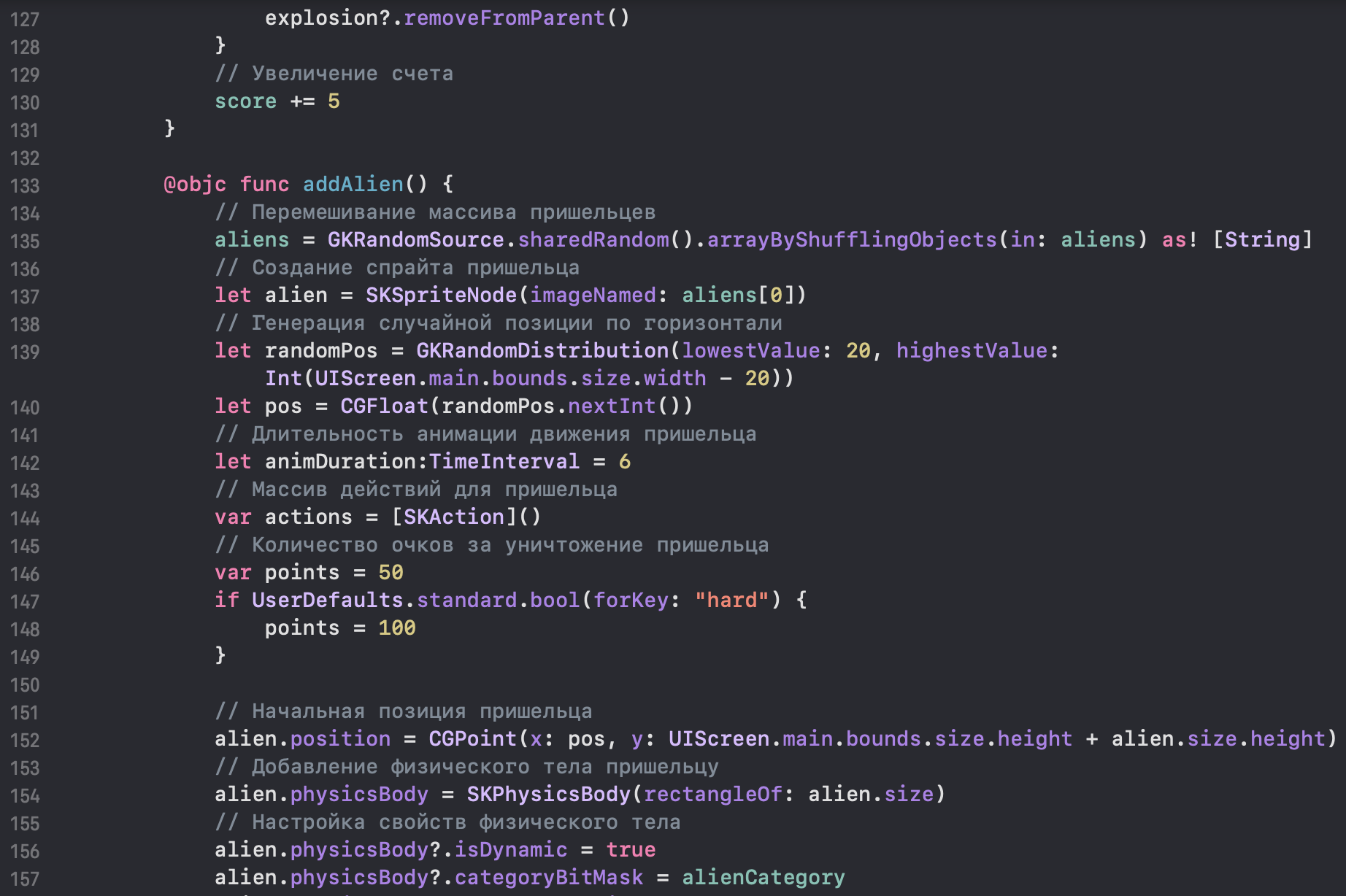
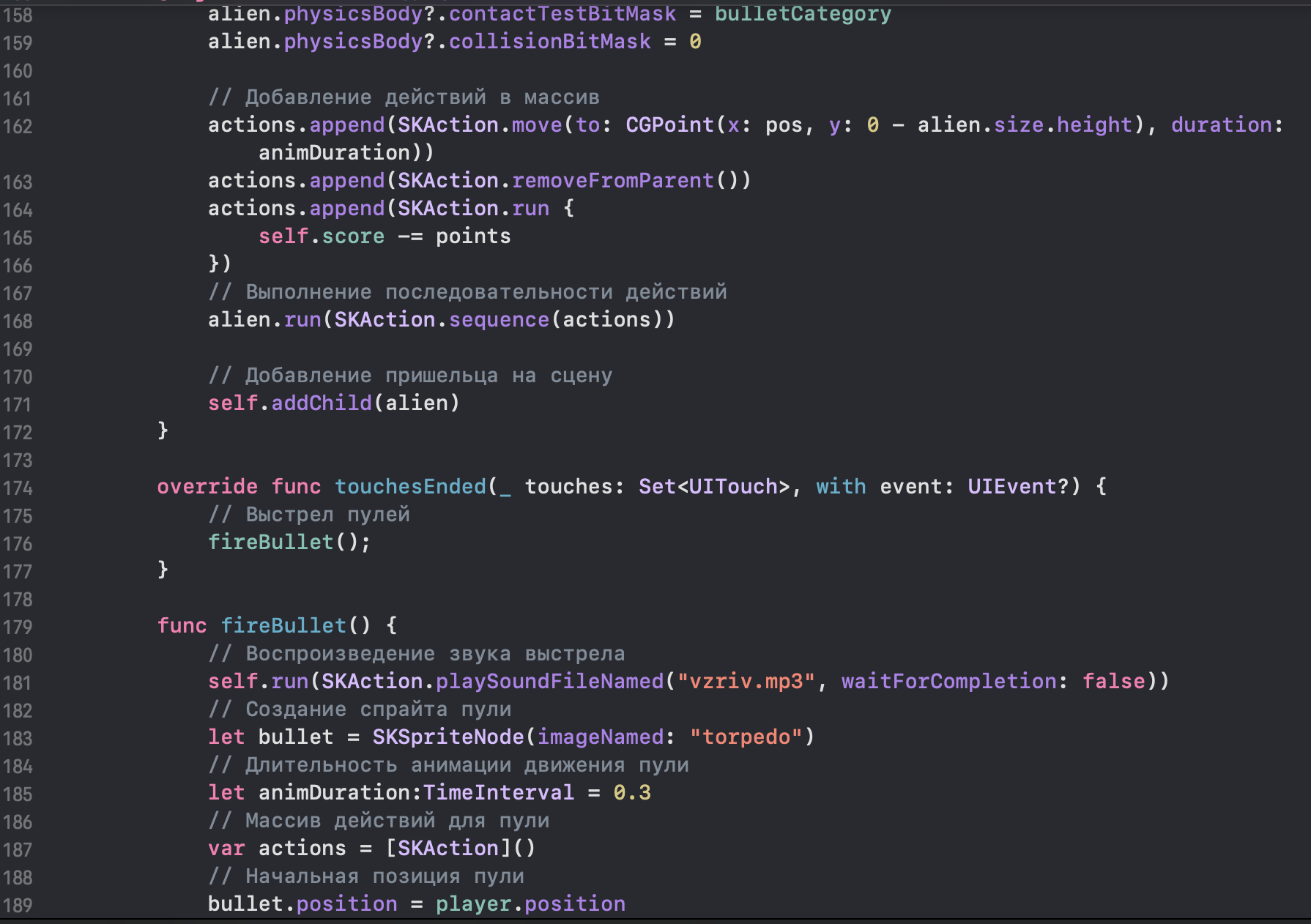
Описание программной реализации

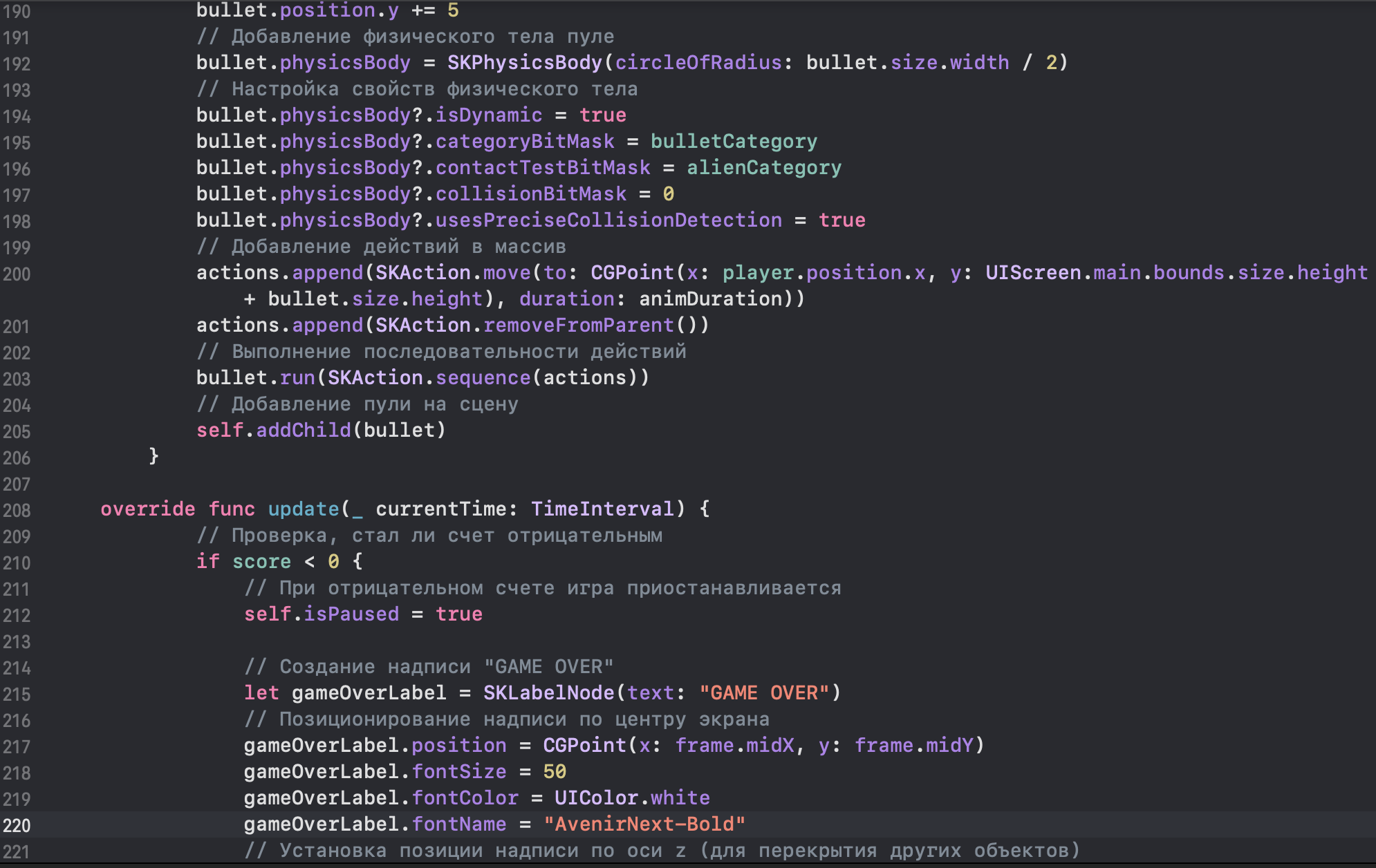
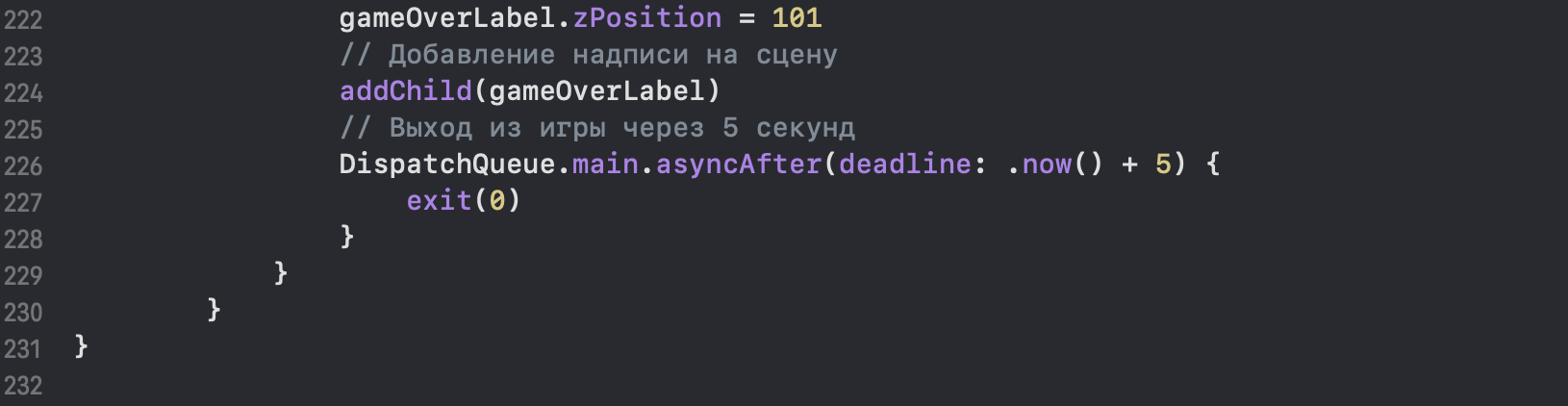
1. Настройка игровой среды
   1. Создать новый проект в Xcode с использованием шаблона Game.
   2. Настроить параметры проекта, такие как название игры, идентификатор пакета и целевые устройства.
   3. Импортировать необходимые библиотеки, такие как SpriteKit и GameplayKit.
2. Создание игровой сцены
   1. Создать новый класс GameScene, который является подклассом SKScene.
   2. Добавить фоновое изображение или анимацию для создания игрового окружения.
3. Создание игрового персонажа
   * 1. Создать спрайт для игрового персонажа (например, космический корабль).
     2. Настроить физическое тело для персонажа, чтобы разрешить физические взаимодействия.
     3. Добавить персонаж на игровую сцену.
4. Создание врагов
   * 1. Создать массив или список различных типов врагов (инопланетные корабли).
     2. Настроить физические тела для врагов и определить их поведение (движение).
     3. Добавить врагов на игровую сцену в случайных позициях.
5. Реализация стрельбы
   * 1. Создать спрайт для пуль.
     2. Настроить физическое тело для пуль и определить их поведение (скорость).
     3. Добавить логику для стрельбы пулями из персонажа игрока.
6. Обработка столкновений
   * 1. Реализовать функцию didBegin(\_ contact:) для обработки столкновений между пулями и врагами.
     2. Уничтожить пули и врагов при столкновении.
     3. Увеличить счет игрока при уничтожении врагов.
7. Управление персонажем
   * 1. Реализовать управление персонажем игрока с помощью акселерометра.
8. Создание игрового цикла
   * 1. Реализовать функцию update(\_ currentTime:) для обновления игрового состояния.
     2. Перемещать врагов.
     3. Проверять столкновения и обновлять счет.
9. Добавление звуковых эффектов
   * 1. Импортировать звуковые файлы для выстрелов, взрывов и фоновой музыки.
     2. Реализовать логику для воспроизведения звуковых эффектов в соответствующих событиях игры.
10. Тестирование и настройка
    * 1. Протестировать игру, чтобы обнаружить и устранить любые ошибки или проблемы.
      2. Настроить игровой баланс: сложность врагов, количество очков за уничтожение.

Код программы, комментарии и пример работы

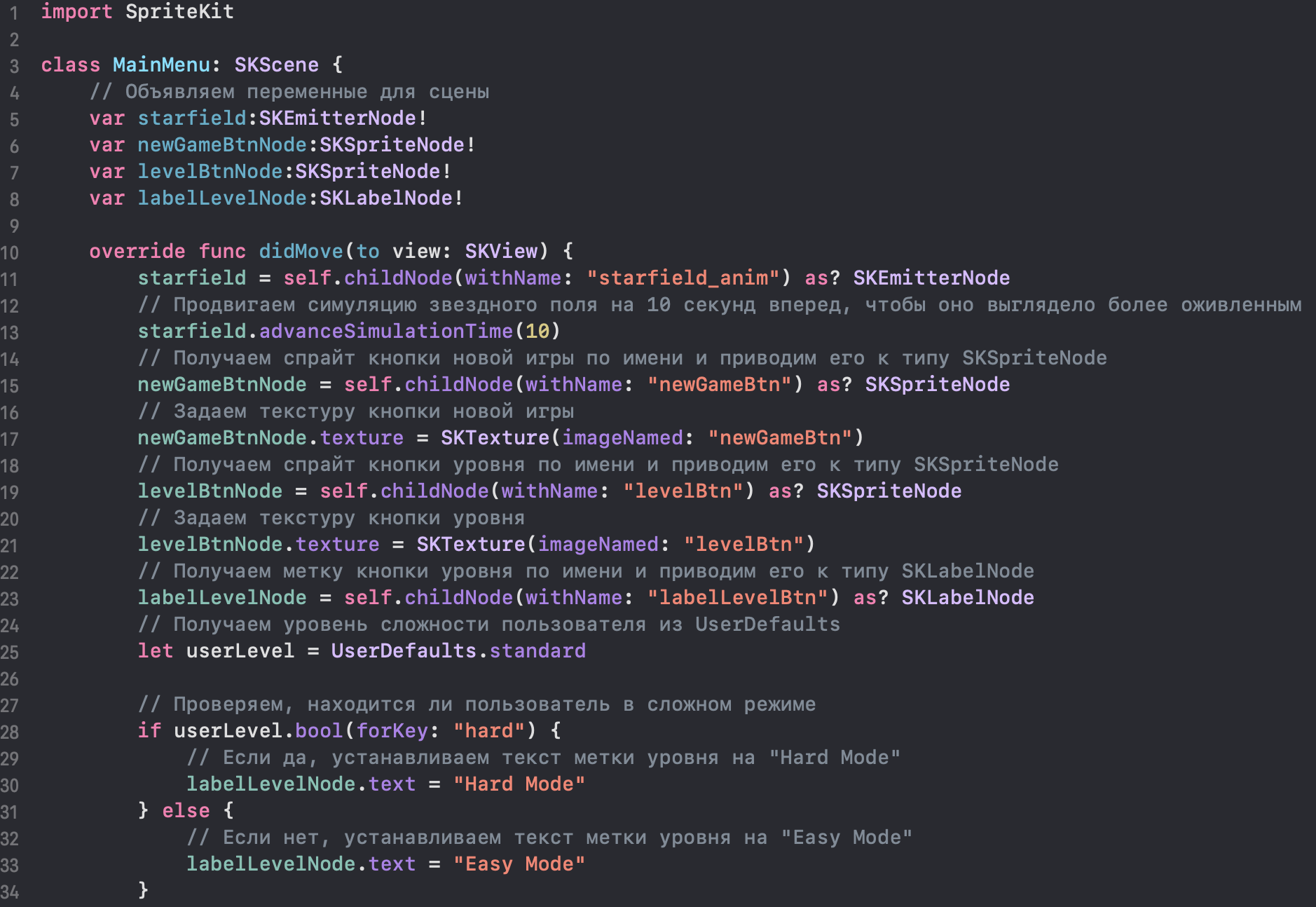
Реализация сцены с игрой:

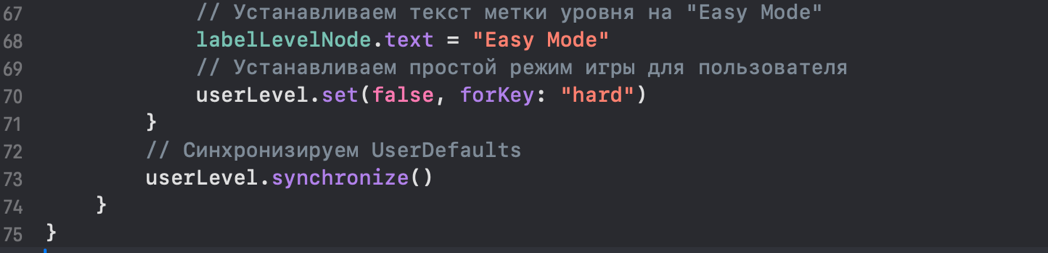
 

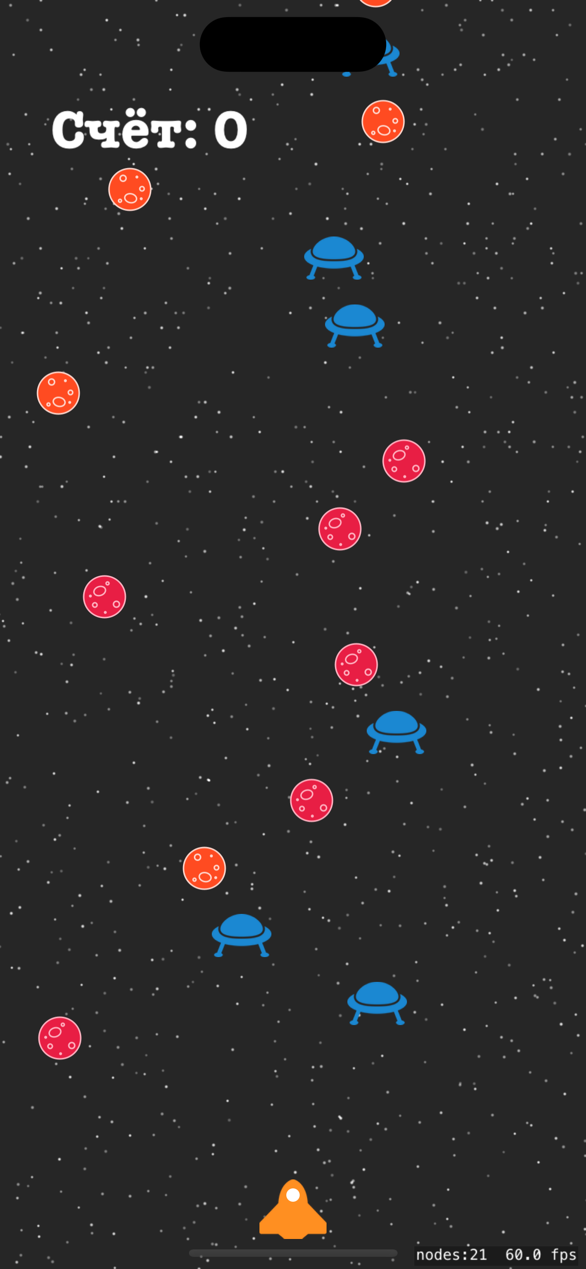
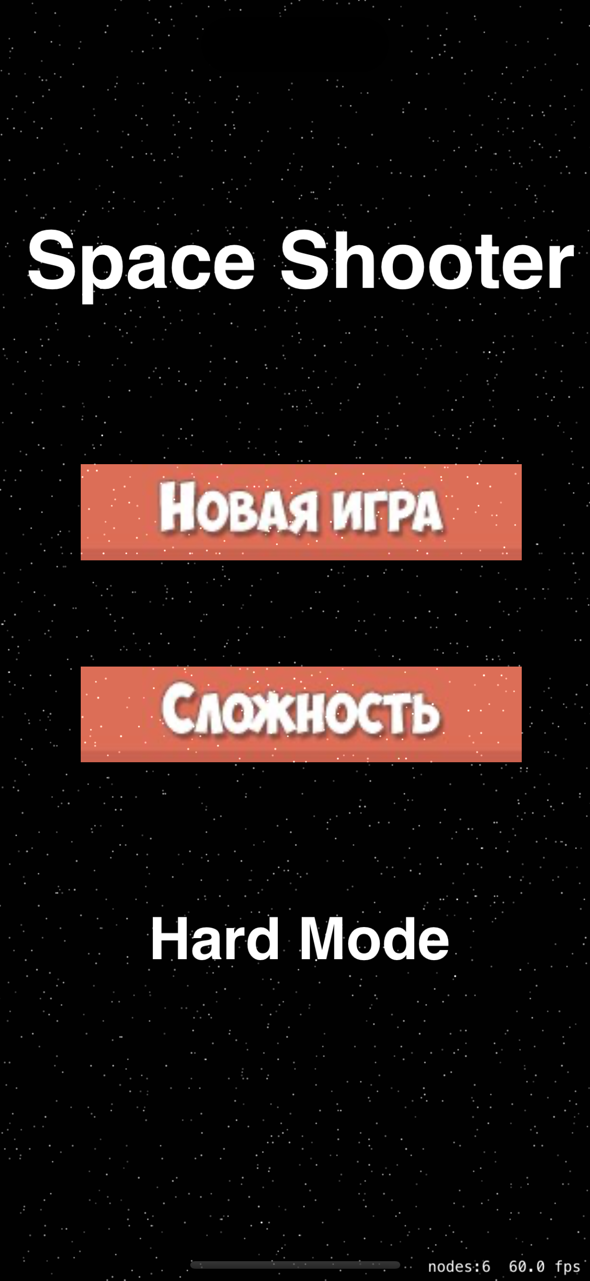
  

Реализация сцены главного меню:



Пример работы

# Заключение

В заключение хочется отметить, чего я добился в течение выполнения данной проектной работы:

* + 1. Создал двухмерную стрелялку в стиле SpaceShooter для iOS-устройств с использованием языка программирования Swift и SwiftUI.
    2. Применил основные концепции Swift, такие как типы данных, переменные, константы и функции.
    3. Изучил основы SwiftUI, такие как создание пользовательского интерфейса, привязка данных и модификаторы.

Я выполнил все поставленные мною задачи, а также достиг цели. Мне понравилась разработка игры. Она помогла мне улучшить понимание отладки и понять логику создания подобных двухмерных игр.

Проект также обладает перспективой развития, в него можно добавить:

1. Большое разнообразие пришельцев.
2. Сделать усложнение в зависимости от набранных очков.
3. Добавить уровни.
4. Добавить различные облики для корабля и пуль.
5. Добавить таблицу лидеров по очкам.

# Список литературы

1. Создание игры на Swift // YouTube [Электронный ресурс] // URL: <https://clck.ru/3B6fqg>
2. Создание игры на Swift // ITProger [Электронный ресурс] // URL: <https://clck.ru/3B6fux>
3. Векторные иконки // Flaticon [Электронный ресурс] // URL: https://clck.ru/3B6fxs