МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования  
«Ивановский государственный энергетический университет

имени В.И. Ленина»

Кафедра Программного обеспечения компьютерных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ**

**Дисциплина: Объектно-ориентированное программирование**

**Тема: Игра “Звездный разрушитель”**

Обучающийся:

студент гр. 2-41 М. В. Прохоров

*(подпись)*

Проверил:

к.т.н., доцент А.Л. Алыкова

*(подпись)*

Оценка:

*(оценка промежуточной аттестации)*

Иваново 2024

# 

# 

[**Постановка задачи**](#_gjdgxs) **3**

[**Техническое задание**](#_gjlwrzj7v4ej) **3**

[Название программного продукта](#_z8k1uyd5a2ir) 3

[Основание разработки](#_1fob9te) 3

[**Общие требования к системе**](#_3znysh7) **3**

[Требования к функциональным характеристикам](#_2et92p0) 3

[Требования к информационному обеспечению](#_tyjcwt) 4

[Требования к лингвистическому обеспечению](#_3dy6vkm) 4

[Требования к программному обеспечению](#_1t3h5sf) 4

[Требования к техническому обеспечению](#_4d34og8) 4

[**Этапы разработки**](#_2s8eyo1) **4**

[**Объектно-ориентированный анализ предметной области**](#_17dp8vu) **6**

[Разработка статической объектной модели](#_3rdcrjn) 6

[**Разработка динамической модели приложения**](#_26in1rg) **13**

[**Особенности реализации**](#_y4yk2pzgkubs) **15**

[**Руководство пользователя**](#) **23**

[**Инструкции по использованию**](#) **23**

# 

# Постановка задачи

Необходимо разработать игру “Звездный разрушитель”, в которой необходимо управлять космическим кораблем с помощью мыши, уворачиваться от вражеских объектов и уничтожать их с помощью пушки, нажимая на ПРОБЕЛ. Помимо этого в игре должно быть реализовано:

1. Сохранение рекордов в файл;
2. Возможность установки на паузу;
3. Масштабирование.

# Техническое задание

## Название программного продукта

Полное название программного продукта: «Star Destroyer», сокращенное: «Звездный разрушитель».

## Основание разработки

Основанием для разработки является «Учебный план по программе бакалавриата. Направление 09.03.04 Программная инженерия. Профиль: Разработка программно-информационных систем».

# Общие требования к системе

## Требования к функциональным характеристикам

1. Отображение начального меню игры;
2. Перемещение космического корабля перетаскиванием с помощью мыши;
3. Враги:
   1. Астероиды, летящие на космический корабль;
   2. Вражеские корабли, движущиеся поперек космического корабля и стреляющие из своих пушек;
4. Возможность постановки на паузу;
5. Возможность стрельбы из пушки;
6. Ограниченное количество патронов, которые пополняются со временем;
7. Сохранение счета и имени игрока при потери 3 единиц здоровья;
8. Вывод списка рекордов в отдельном окне.

## Требования к информационному обеспечению

Для сохранения данных о рекордах используется файл records.dat.

К системе прилагается документация по правилам работы с программой.

## **Требования к лингвистическому обеспечению**

По умолчанию программный продукт имеет русскую версию интерфейса. Система не имеет встроенную поддержку многоязычности.

## **Требования к программному обеспечению**

Клиентское приложение поддерживает следующие ОС: Windows (Windows 7 и выше); Linux дистрибутивы, основанные на ядре debian.

## **Требования к техническому обеспечению**

Для корректной работы клиентского требуется 200 Мб свободного дискового пространства, 400 Мб ОЗУ.

Для игры требуется клавиатура с мышью или тачпадом.

# **Этапы разработки**

1. Загрузка изображений объектов игры;
2. Создание базовых классов;
3. Создание начального меню, окна игры и меню окончания игры;
4. Создание классов игровых объектов;
5. Написание логики игры:
   1. Управление космическим кораблем игрока;
   2. Появление и поведение врагов;
   3. Уничтожение врагов и космического корабля игрока;
   4. Сохранение рекорда и имение игрока.
6. Написание справки.

# **Объектно-ориентированный анализ предметной области**

## **Разработка статической объектной модели**

Основным классом является класс **GameObject**, от которого наследуется большинство классов в приложении. В его конструкторе определяются координаты и размер игрового объекта. Также класс содержит такие поля, как speed (скорость объекта), hp (очки здоровья), Sprite (изображение), score (сколько очков будет приносить), а также методы IsCollide (определение столкновения с другим объектом), virtual Draw (отрисовка игрового объекта), abstract Fly (метод, изменяющий положение игрового объекта за тик времени).

Класс **StarShip** - это класс космического корабля игрока, который наследуется от GameObject. В его конструкторе принимаются начальные координаты корабля, определяется спрайт (изображение) космического корабля, hp (количество очков здоровья). Определяются методы MouseDown, MouseMove, MouseUp, которые вызываются при возникновении соответствующих событий. Если очки здоровья становятся равны нулю, игра завершается.

Класс **Rocket** - класс вражеской ракеты, которая появляется в рандомном месте через каждый интервал времени. Наследуется от класса GameObject. В его конструкторе принимаются начальные координаты, определяется спрайт, скорость, очки здоровья и количество очков, начисляемых игроку за уничтожение ракеты выстрелом. Переопределен метод Fly, который определяет изменение координат в зависимости от скорости каждый тик таймера. Если очки здоровья равны нулю, ракета уничтожается.

Класс **EnemyShip** является вражеским кораблем, появляющийся реже, чем ракета, в рандомной стороне в верхней части экрана. Наследуется от класса GameObject. В конструкторе принимаются начальные координаты, определяется спрайт, скорость, очки, начисляемые при уничтожении, очки здоровья и направление движения (направо или налево, 1 и -1 соответственно). Переопределяется метод Fly, который определяет изменение координат в зависимости от скорости и направления движения каждый тик таймера. Определено статическое свойство, которое отвечает за скорострельность - количество тиков таймера, которое проходит между выстрелами вражеского корабля. Если очки здоровья равны нулю, корабль уничтожается.

Класс **Bullet** - это класс пули, который наследуется от GameObject. В конструкторе принимаются начальные координаты, определяется цвет и скорость пули. Переопределяется функция Draw, так как у пули нет спрайта, она является обычным прямоугольником с цветом color.

Класс **MyBullet** наследуется от класса Bullet и является пулей, выпущенной космическим кораблем игрока. В конструктор наследуемого класса Bullet передается цвет пули, а также переопределяется метод Fly. Пуля создается при нажатии на пробел и летит в сторону противников. При попадании в ракету или вражеский корабль отнимает 1 очко здоровья противника.

Класс **EnemyBullet** наследуется от класса Bullet и является пулей, выпущенной вражеским кораблем. В конструктор наследуемого класса Bullet передается цвет пули, а также переопределяется метод Fly. Выпускается вражеским кораблем, и при попадании в космический корабль игрока отнимает 1 очко здоровья.

Класс **Heart** наследуется от класса GameObject и является обозначением одного очка здоровья игрока - сердечко. Сердечки отрисовываются к углу экрана в том количестве, сколько очков здоровья у игрока.

В классах игровых объектов также определены статические свойства - ширина и высота.

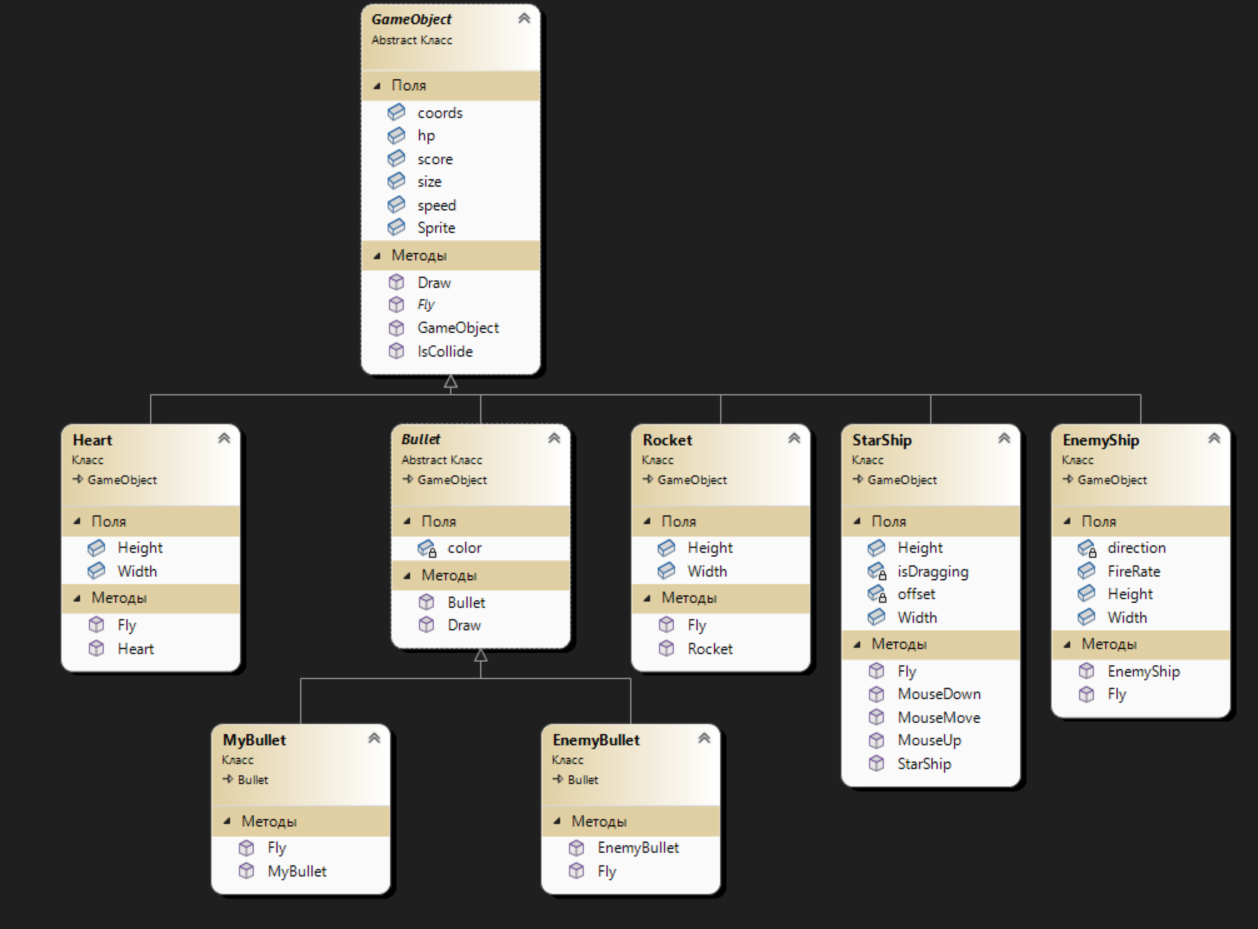
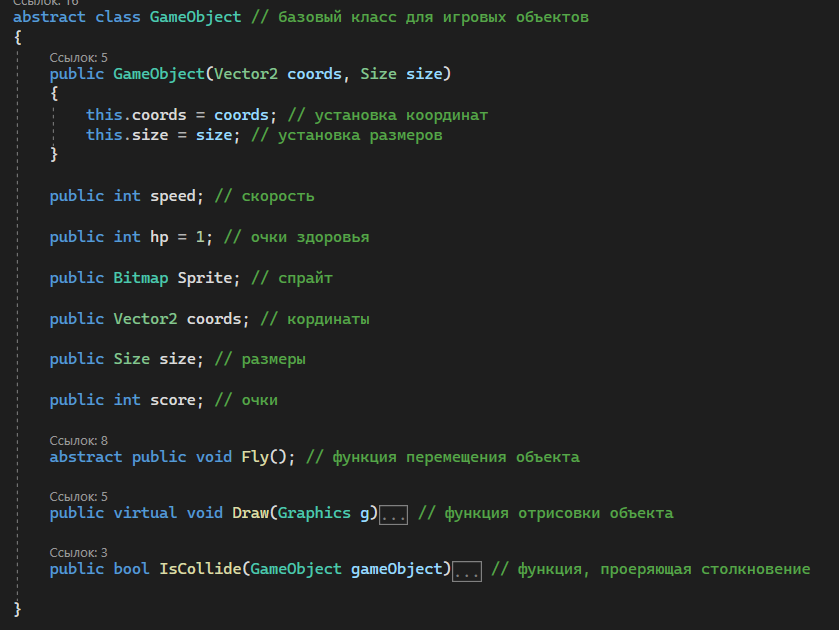


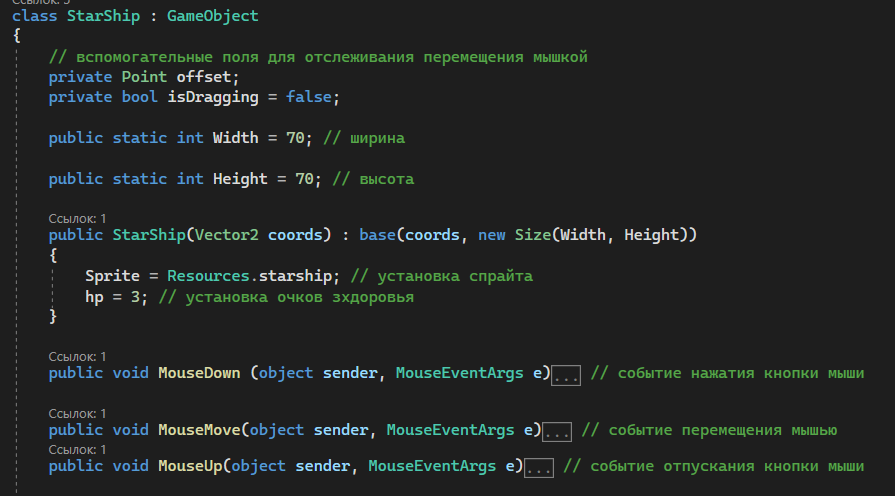
Рисунок 1. Диаграмма классов

**Реализация объектов**

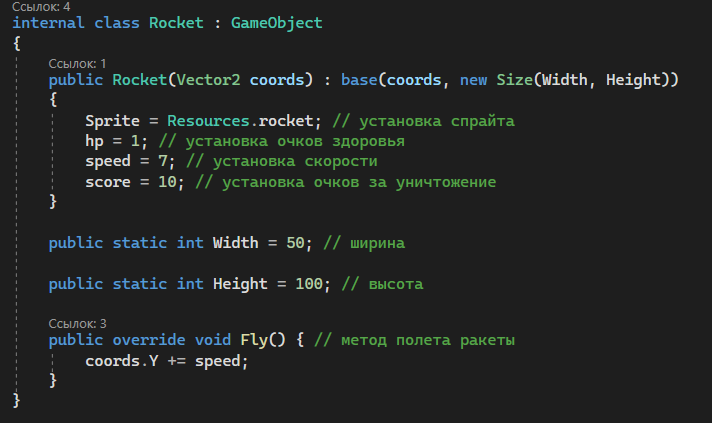
**class GameObject**

****

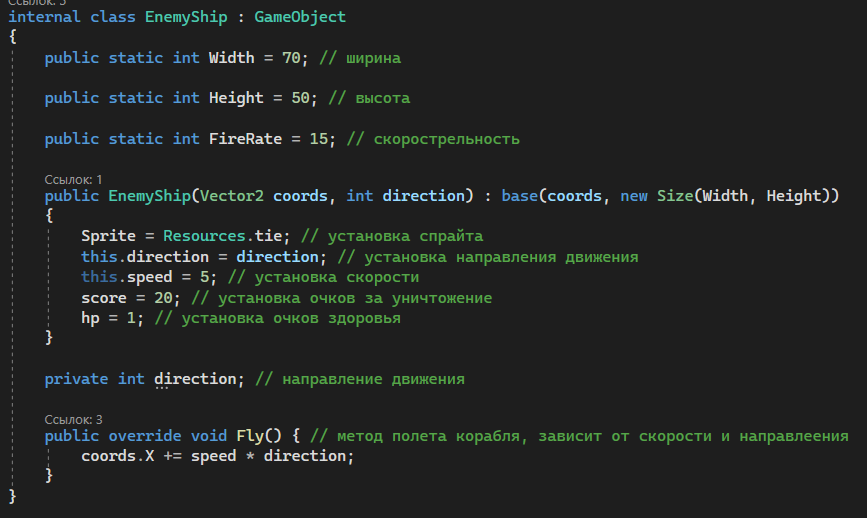
**class StarShip**

****

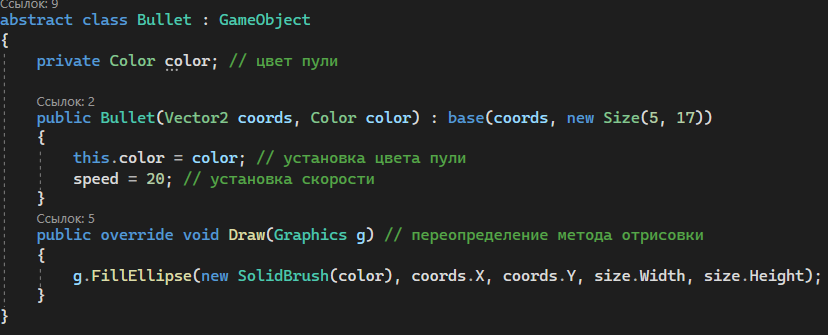
**class Rocket**

****

**class EnemyShip**

****

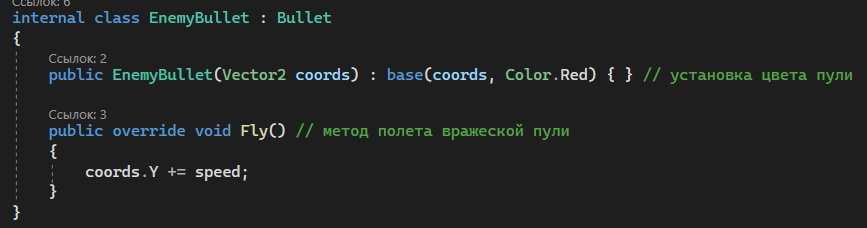
**class Bullet**

****

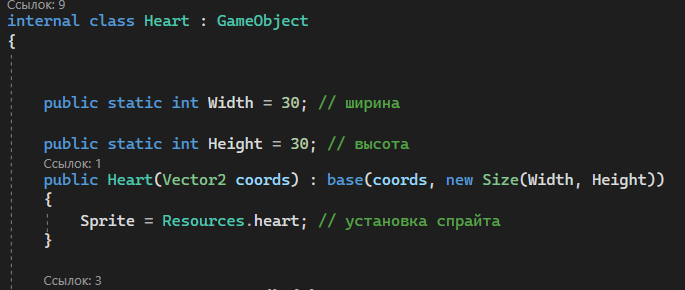
**class MyBullet**

****

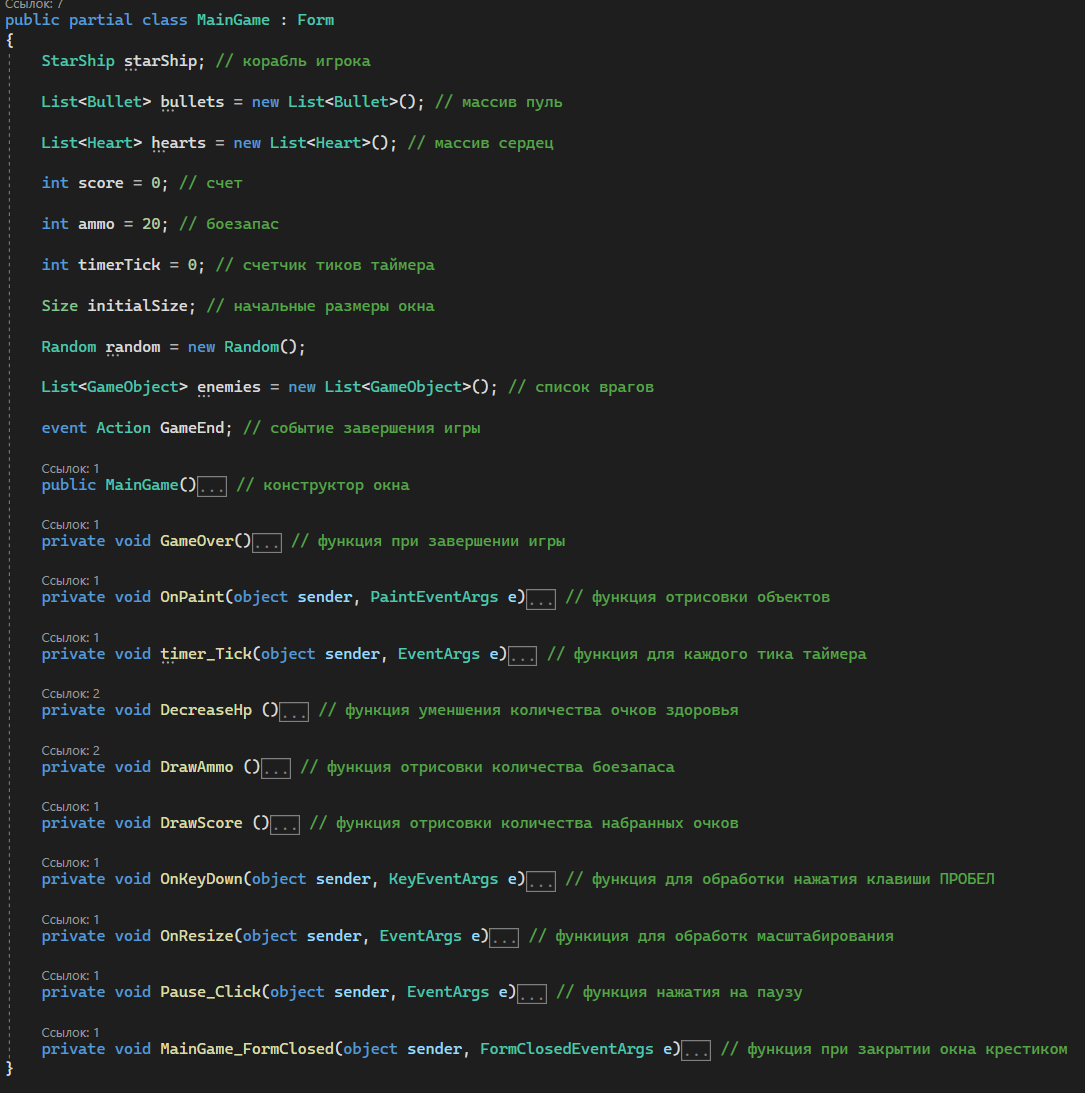
**class EnemyBullet**

****

**class Heart**

****

**class MainGame**

****

# **Разработка динамической модели приложения**

При запуске приложения, пользователь видит стартовое меню. На нем изображены 3 кнопки - “Играть”, “Рекорды” и “О программе”. При нажатии на кнопку “Рекорды” отображается окно со списком игроков и их рекордов. При нажатии на кнопку “О программе” отображается окно с описание программы. При нажатии на кнопку “Играть” закрывается главное меню, открывается окно и игрой.

Внизу отображается космический корабль игрока, который игрок может перетаскивать мышью по всему окну. Периодически в рандомных местах верхней стороны окна будут появляться ракеты, летящие в сторону игрока, от которых можно увернуться, либо уничтожить выстрелом из пушки. За уничтожение ракеты начисляется 10 очков. Для выстрела необходимо нажать клавишу ПРОБЕЛ - и корабль игрока выпустит пулю, которая будет лететь в сторону врагов. Количество пуль ограничено, но оно увеличивается со временем. Иногда будут появляться вражеские космические корабли в верхней части экрана, вылетать справа или слева и двигаться в противоположную сторону по горизонтали, периодически выпуская по 2 пули за выстрел. Эти пули двигаются в сторону игрока, и при попадании в корабль игрока отнимают по 1 единице здоровья. Вражеские корабли также можно уничтожить из пушки, либо проигнорировать. При уничтожении вражеского корабля начисляется 20 очков. Текущий счет отображается в левом верхнем углу окна. Игрок может поставить игру на паузу, нажав кнопку в правом верхнем углу. Количество единиц здоровья корабля игрока изначально равняется трем. Очки здоровья отображаются в левом нижнем углу в виде сердечек. Если игрок теряет все очки здоровья, то он проигрывает, и появляется окно сохранения рекорда, которое предлагает игроку ввести его имя. При вводе имени рекорд сохраняется в файл.

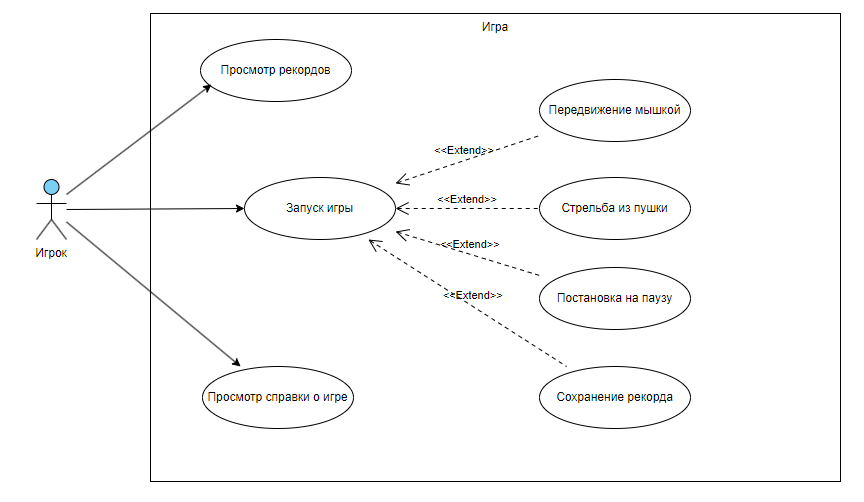


Рисунок 2. Диаграмма прецедентов

# 

# **Особенности реализации**

Основная логика игры заложена в окне игры Game.cs. Инициализируются все переменные и списки объектов.

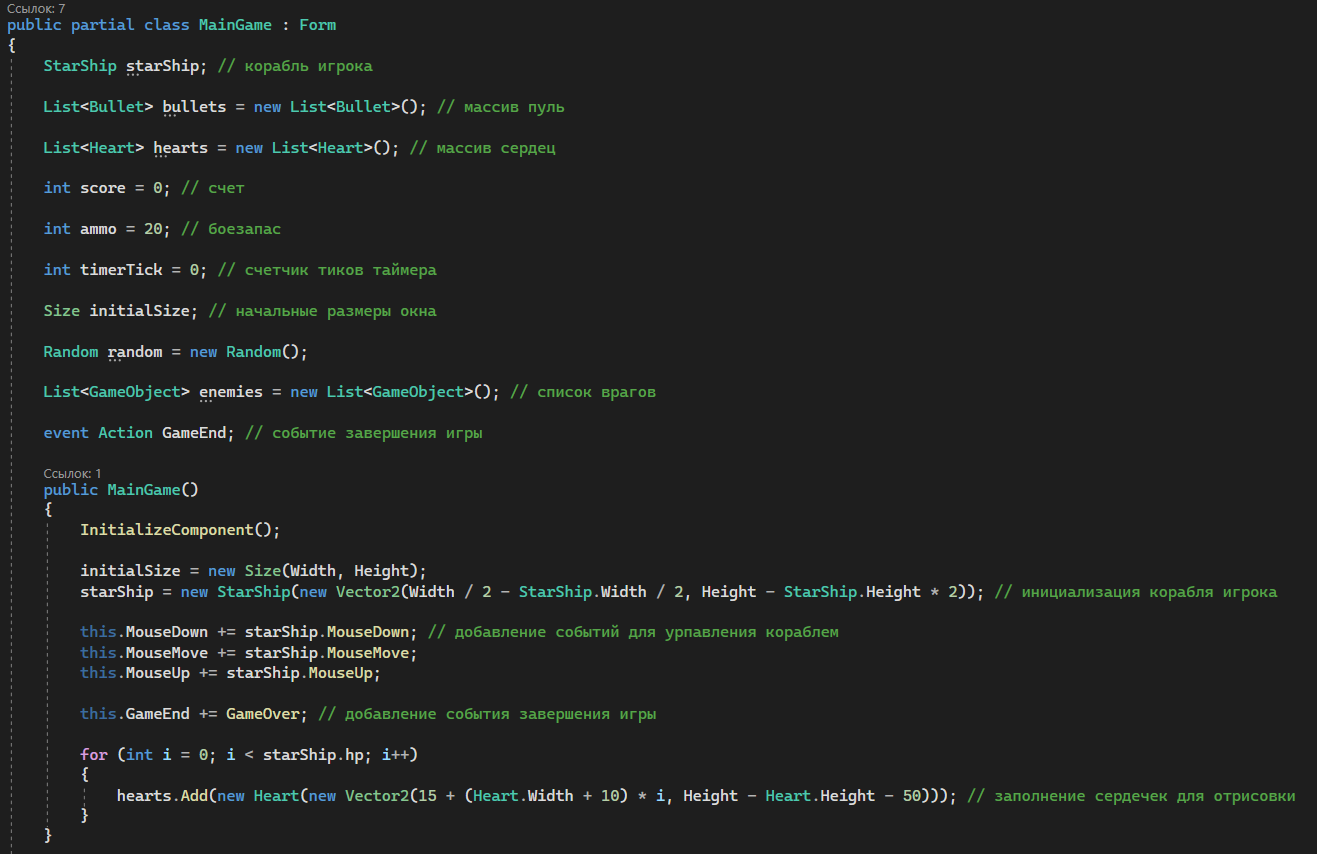


Рисунок 3. Инициализация игровых объектов.

Игра реализована по следующему принципу: несколько раз в секунду с определенным интервалом вызывается основная функция **timer\_Tick**, которая обновляет состояния всех игровых объектов, и обновляет экран вызовом функции Refresh(), в результате чего срабатывает функция **OnPaint**, которая и отрисовывает все объекты в циклах.

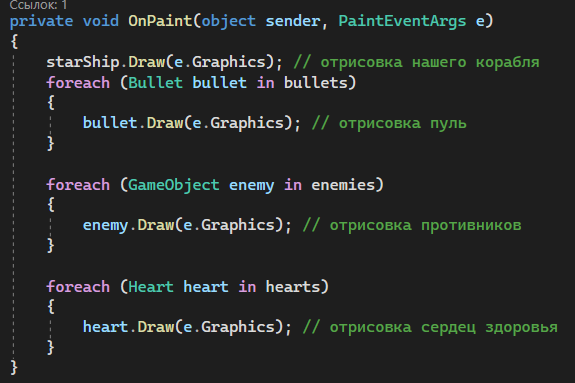


Рисунок 4. Функция OnPaint.

Функция **timer Tick** реализует 4 основных этапа:

1. Спавн врагов на каждый 30 тик таймера, причем с вероятностью 0.25 появится вражеский космический корабль. В массив врагов добавляется соответствующий экземпляр класса. Также каждые 20 тиков добавляется 1 пуля в боезапас.

# 

Рисунок 5. Спавн врагов.

1. Обработка полета врагов, а также выстрелов вражеских кораблей. Если врезаемся в противника, уменьшаем наше hp.

# 

Рисунок 6. Обработка полета врагов.

1. Обработка полета пуль, а также попадание пуль в противника или в корабль игрока.

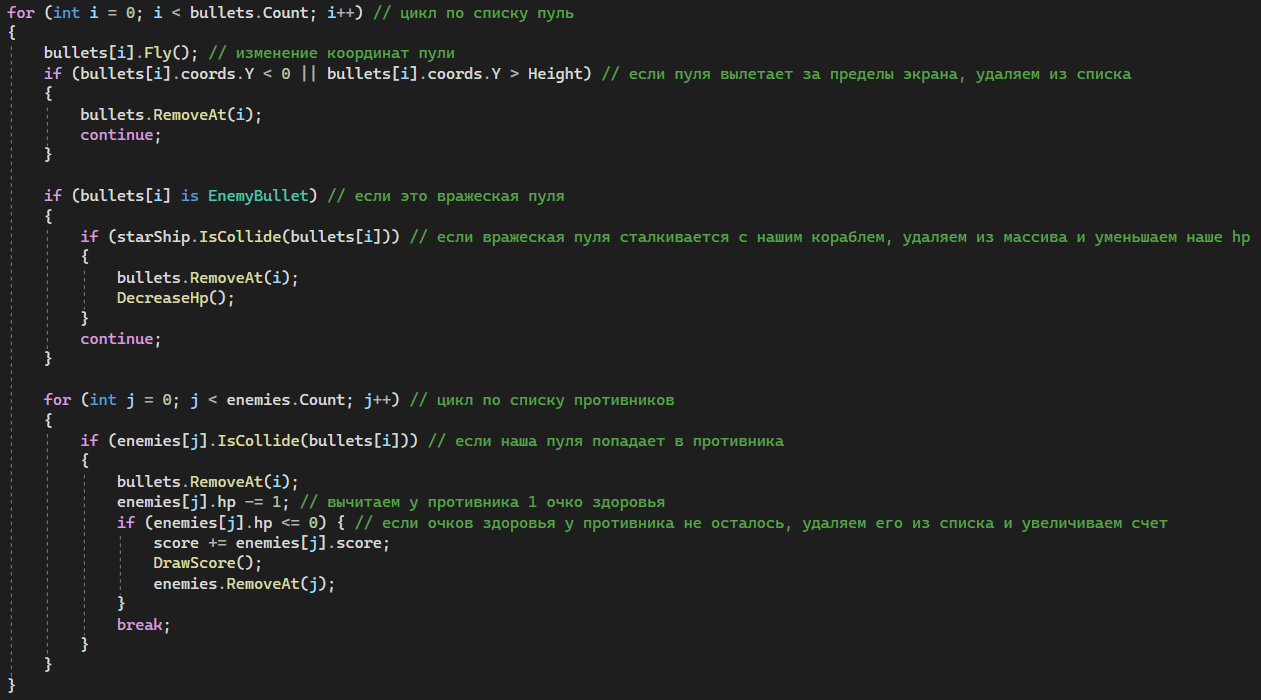


Рисунок 7. Обработка полета пуль.

1. Вызов встроенной функции Refresh() - обновление экрана



Рисунок 8. Обновление экрана.

Функция OnKeyDown обрабатывает нажатие клавиши ПРОБЕЛ и реализует выстрел.

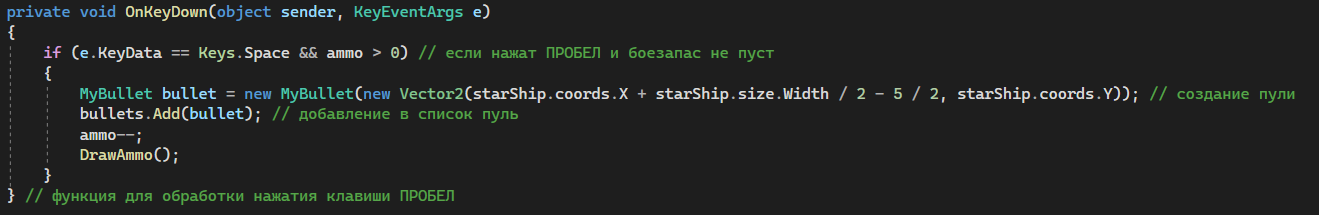


Рисунок 9. Выстрел по нажатию пробела.

В реализации присутствует собственное событие - GameEnd. Оно вызывается, когда игра завершается, то есть игрок потерял все очки здоровья. Оно вызывает функцию GameOver, которая открывает окно записи рекорда.

# 

Рисунок 10. Завершение игры.

Игра поддерживает масштабирование: координаты игровых объектов соблюдают пропорции сторон экрана при растягивании.

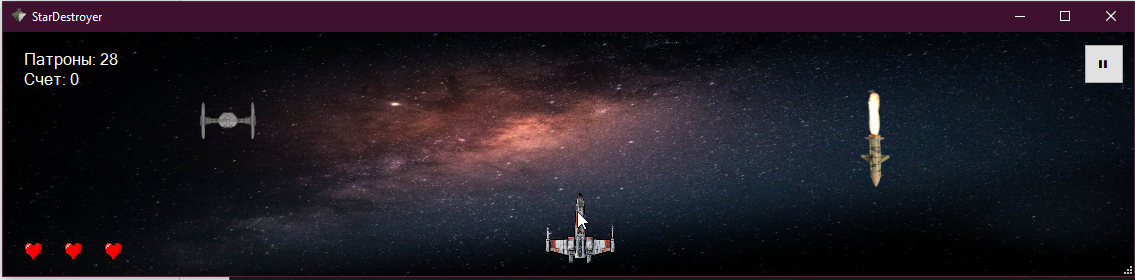


Рисунок 11. Масштабирование экрана.

В игре также реализовано сохранение рекордов в файл с помощью сериализации.

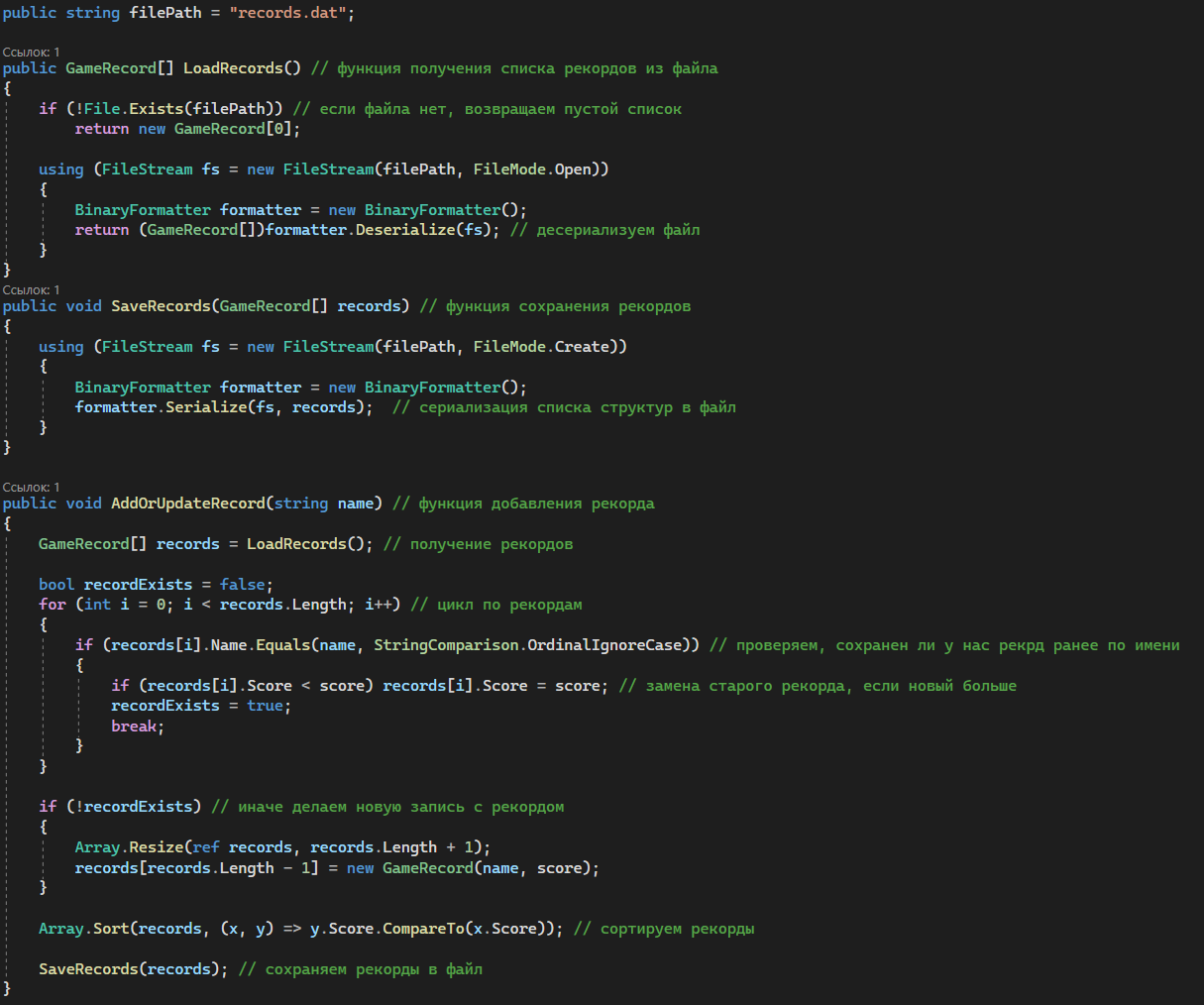


Рисунок 12. Функционал сериализации и десериализации массива структур в файл.

В процессе игры может возникнуть необходимость поставить на паузу - это реализовано в приложении.

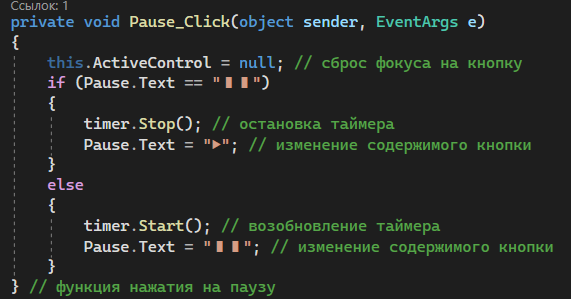


Рисунок 13. Установка на паузу (реализация).

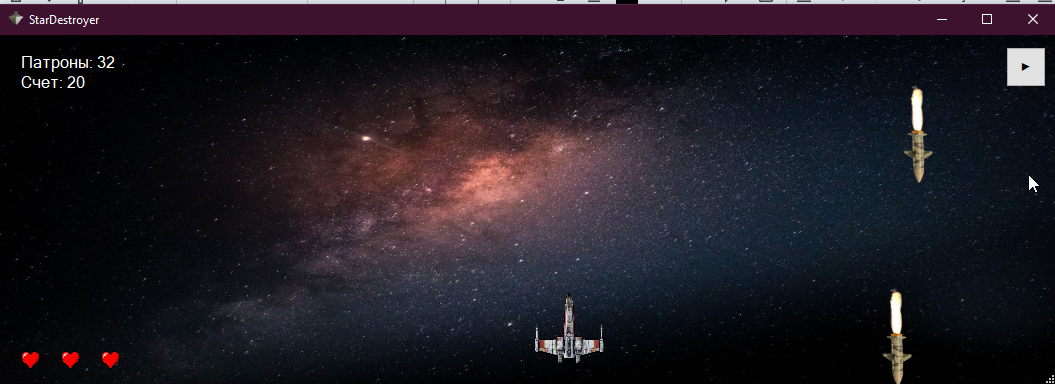
****

Рисунок 13. Установка на паузу (результат).

# 

Рисунок 14. Диаграмма последовательности при взаимодействии пули с объектами.

# **Руководство пользователя**

# **Инструкции по использованию**

Для запуска приложения необходимо запустить файл StarDestroyer.exe из папки Game. Откроется стартовое меню. Внешний вид приложения представлен на рисунке 3.

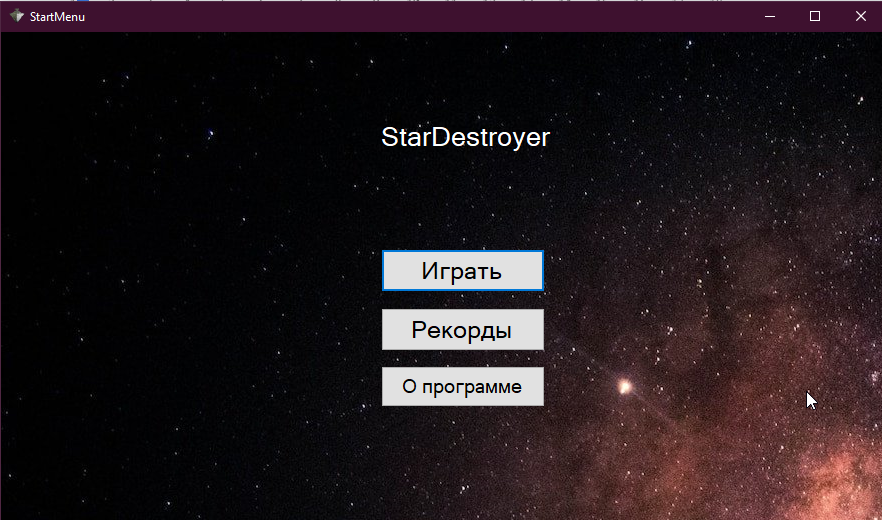


Рисунок 15. Внешний вид приложения.

Вы можете посмотреть справку по игре, нажав кнопку “О программе”.

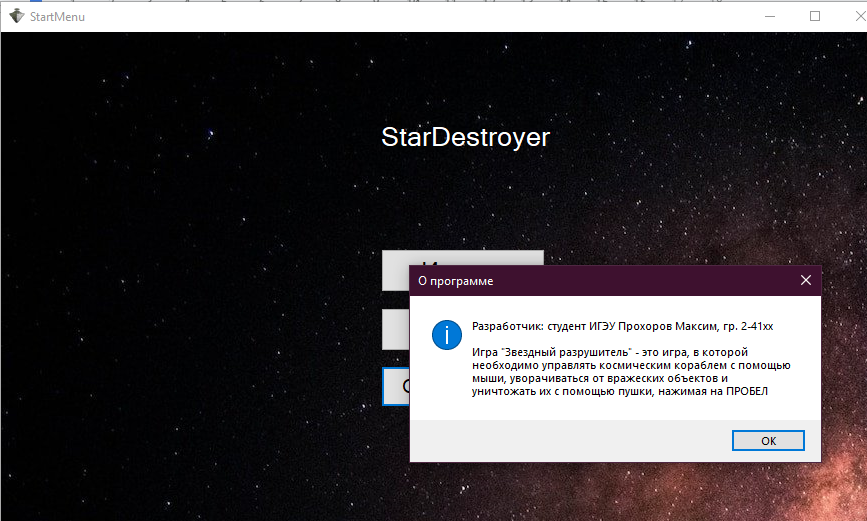


Рисунок 16. О программе.

Для того, чтобы запустить игру, надо нажать кнопку “Играть”. Откроется окно с игрой.

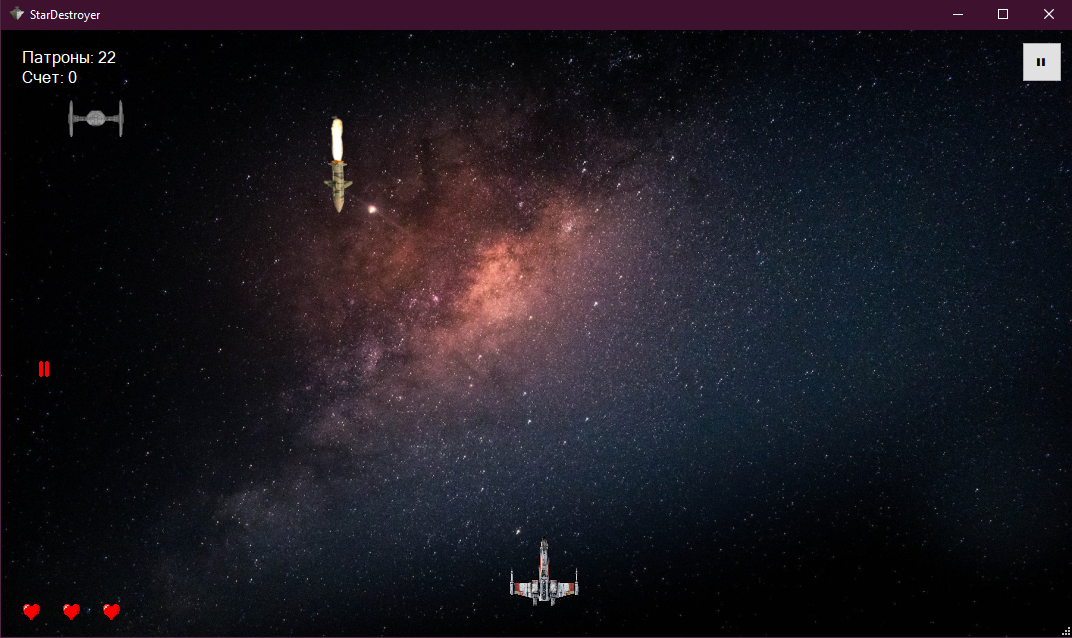


Рисунок 17. Запуск игры.

Чтобы управлять кораблем, необходимо перетаскивать космический корабль мышкой. Чтобы стрелять, надо нажимать на ПРОБЕЛ.

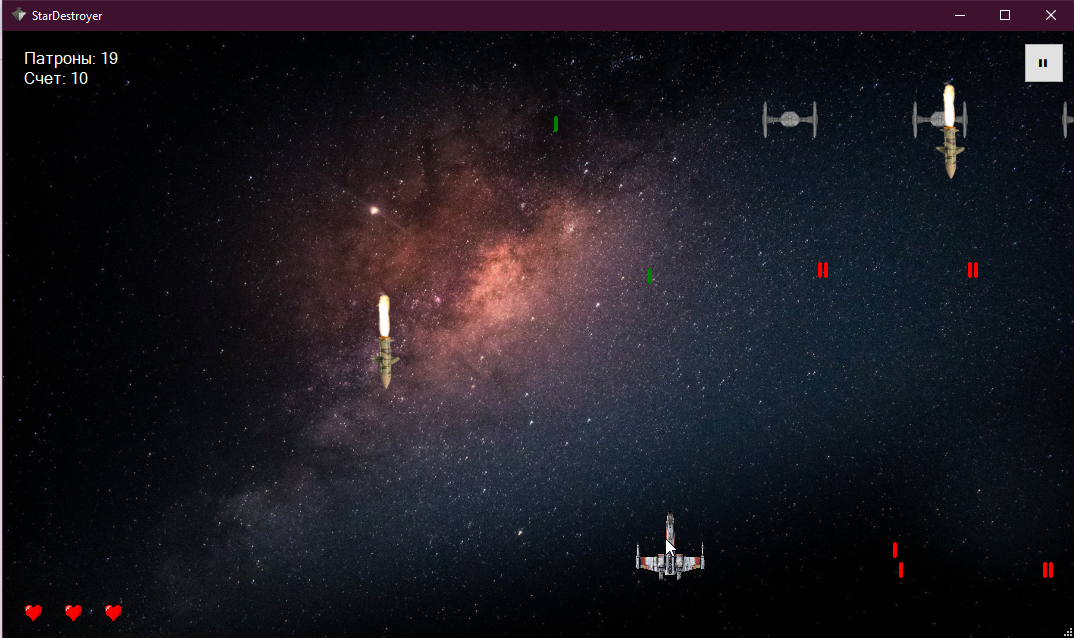


Рисунок 18. Игровой процесс.

При завершении игры, чтобы сохранить рекорд, надо ввести имя и нажать “Сохранить”.

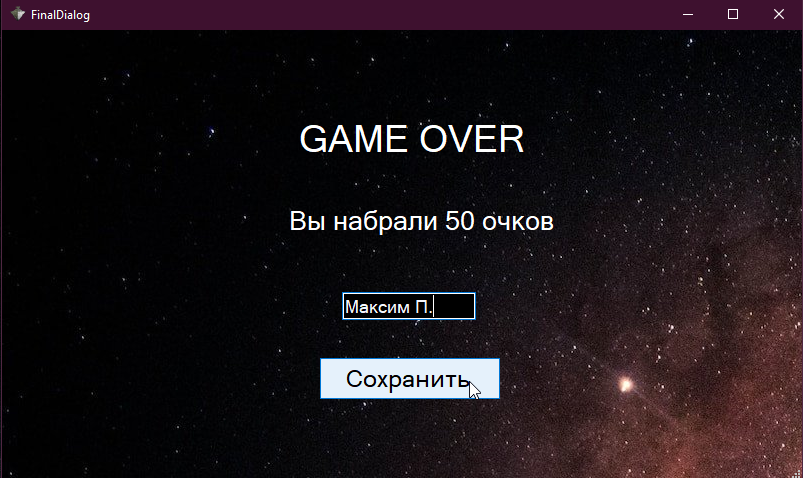


Рисунок 19. Сохранение рекорда.

Для просмотра списка рекордов, надо нажать кнопку “Рекорды” на главном меню приложения.

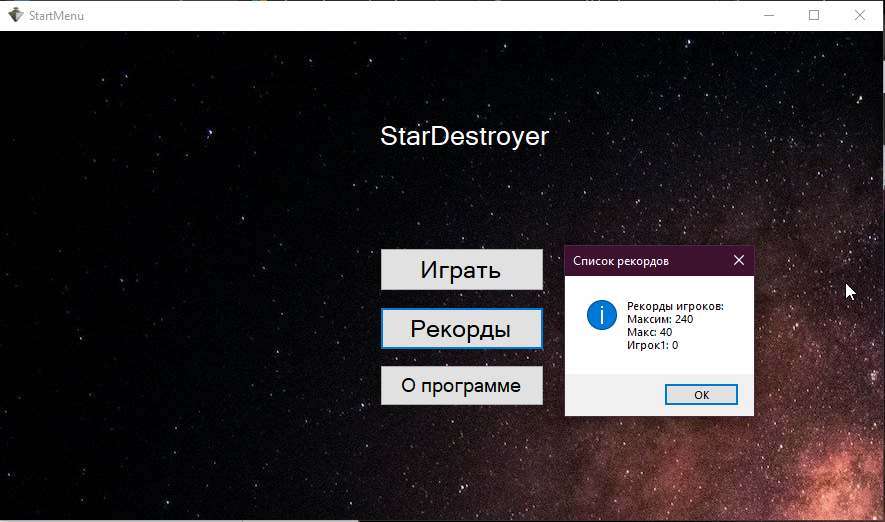


Рисунок 20. Список рекордов.