**Северо-Осетинский Государственный Университет  
Факультет математики и информационных технологий  
  
  
  
  
  
  
  
  
  
  
  
Курсовая работа  
Аркадная игра в жанре космического шутера “SpacePilot”**

**Студент 1 курса 11 группы  
Т.О.Баскаев**

**Владикавказ 2023**

Оглавление

[**Введение** 3](#_Toc515644717)

[**Глава 1. Теоретическая часть** 3](#_Toc515644718)

[Постановка задачи 3](#_Toc515644719)

[**Глава 2. Практическая часть**  4](#_Toc515644721)

[Руководство пользователя 4](#_Toc515644725)

[Руководство системного администратора 4](#_Toc515644726)

[Руководство программиста 5](#_Toc515644727)

[Заключение 9](#_Toc515644728)

[Список литературы 9](#_Toc515644729)

## 

## **Введение**

Аркадная игра - это тип видеоигры, который обычно характеризуется простым и интуитивно понятным геймплеем, быстрым темпом и высоким уровнем азартности. Термин "аркадная игра" происходит от игровых автоматов (аркадных автоматов), которые были популярны в 70-80-х годах прошлого столетия. Они обычно представляли собой кабинеты с экраном и кнопками или джойстиками для управления персонажем в игре. Аркадные игры обычно не стремятся к реализму, а скорее к созданию увлекательного и захватывающего игрового опыта. Они часто имеют простую и понятную механику, что делает их доступными для широкой аудитории. Такие игры могут включать платформеры, шутеры, файтинги, гонки и другие жанры.

Космический шутер (англ. space shooter) - это жанр видеоигр, в которых игрок управляет космическим кораблем или другим летательным аппаратом и сражается с противниками в космической среде. Целью игры обычно является уничтожение врагов, выживание и достижение наивысшего счета или выполнение определенных задач. Космические шутеры могут представляться в различных форматах и стилях. Некоторые игры этого жанра предлагают более аркадный подход, где игроку предоставляется максимальная свобода движения и возможность использовать различные виды оружия и способности для борьбы с врагами. Другие космические шутеры могут иметь более реалистичные модели полета и физики, а также предлагать более сложный игровой процесс, где тактика и стратегия играют важную роль.

Так как мне интересно было создать полноценную игру мой выбор пал именно на аркаду.

## **Постановка** **задачи**

Создать полнофункциональную аркаду – космический шутер с помощью технологии WinForms. Создание окна игры, игрового поля, космического корабля с возможностью управления и стрельбы, целей – метеоритов, реализовать анимацию взрыва, реализовать постепенное усложнение игры. Реализовать логику управления кораблем с помощью клавиатуры или мыши. Обработать столкновения игрового персонажа с врагами или другими препятствиями.

## **Глава 1. Теоретическая часть [1]**

Теоретическая часть создания аркады в жанре космического шутера на WinForms включает в себя следующие аспекты:

1. Архитектура приложения. Можно применить принципы объектно-ориентированного программирования (ООП) для создания модульной и расширяемой структуры кода. Разделить игровую логику, графический интерфейс и взаимодействие с пользователем на отдельные компоненты.

2. Графический движок. WinForms предоставляет некоторые базовые средства для рисования, но для более сложных графических эффектов и анимации вам может потребоваться расширять их функциональность или использовать сторонние библиотеки.

3. Управление игровым объектом. Определить логику управления космическим кораблем игрока. Можно реализовать обработчики событий ввода пользователя и связь их с перемещением и стрельбой корабля.

4. Моделирование движения объектов. Определить логику движения космических кораблей врагов и других игровых объектов. Это может включать представление объектов как спрайты или специальные графические элементы. Установить параметры скорости и траектории для движения объектов и обеспечьте их обновление на каждом кадре игры.

5. Обнаружение столкновений. Реализовать систему обнаружения столкновений между объектами в игре. Проверять пересечения границ и коллизии между спрайтами кораблей и других игровых объектов.

6. Счет и прогресс игры. Разработка логики подсчета очков и отслеживания прогресса игрока. Система где сложность игры будет возрастать по мере продвижения игрока.

7. Интерфейс пользователя. Создать интерфейс пользователя, который будет отображаться на главном окне игры. Добавьте кнопки для начала игры, паузы или перезапуска.

8. Обработка игровых событий. Обработка события, связанные с началом и окончанием игры, паузой, победой или поражением игрока. Управление потоком игры, запуская и останавливая игровую логику по мере необходимости.

Это общая теоретическая основа, которая лежит в основах создания аркады в жанре космического шутера на WinForms. В процессе разработки можно вносить дополнительные детали и функции в соответствии пользовательскими потребностями и творческим видением игры.

## **Глава 2. Практическая часть**

## **Словесное описание**

Вся игра основывается на управлении ракетой, уклоняясь от метеоритов и стреляя по ним лазером. Счет увеличивается с течением времени, а игра завершается при столкновении ракеты с метеоритом

## **Руководство пользователя**

1. Запуск игры: Чтобы запустить игру, откройте приложение и войдите в главное меню.

2. Главное меню: в главном меню вы увидите кнопку "I'm ready". Нажмите на нее, чтобы начать игру.

3. Управление ракетой: Во время игры вы управляете ракетой, используя клавиши W, A, S и D на клавиатуре. Клавиши W и S отвечают за движение ракеты вверх и вниз соответственно, а клавиши A и D - за движение влево и вправо.

4. Стрельба лазером: Чтобы стрелять лазером, используйте клавишу X на клавиатуре. У вас есть ограниченное количество зарядов лазера, которое отображается на экране.

5. Уклонение от метеоритов: Ваша задача - уклоняться от метеоритов, которые движутся по экрану. Если ракета столкнется с метеоритом, игра окончится.

6. Подбитые метеориты: Если ваш лазер попадает в метеорит, он будет подбит. Подбитые метеориты будут отображаться с анимацией взрыва. Ваша цель - подбить как можно больше метеоритов, чтобы заработать очки.

7. Счет и время: в верхней части экрана отображается ваш текущий счет и время игры. Чем дольше вы продержитесь и чем больше метеоритов подбито, тем выше будет ваш счет.

8. Завершение игры: Если ваша ракета столкнулась с метеоритом, игра окончится. Вы увидите сообщение с вашим окончательным счетом. Чтобы начать новую игру, нажмите кнопку "OK".

9. Выход в меню: Если вам нужно выйти из текущей игры и вернуться в главное меню, нажмите кнопку "Выход" в правом верхнем углу экрана.

10. Завершение игры: Если вы хотите полностью выйти из игры, закройте приложение.

## **Руководство системного администратора**

Для запуска игры вам понадобится exe файл WindowsFormsApp3.exe, 16 МБ памяти. Рекомендуется в настройках также изменить время задержки при нажатии клавиш до минимума.

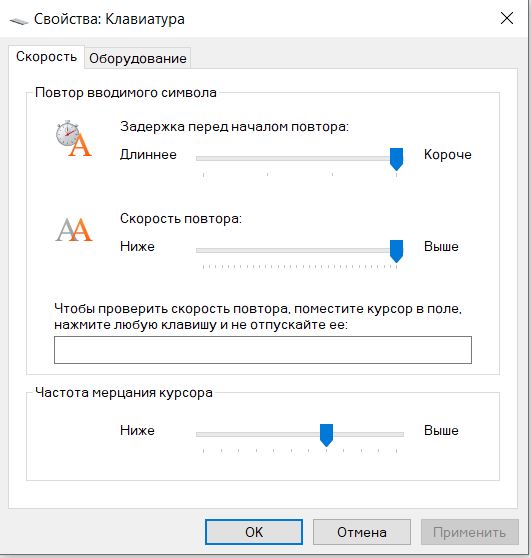


Рис.1 Клавиатура

## **Руководство программиста**

Словесное руководство для понимания кода и его функциональности:

1. Общая структура:

- Код разделен на несколько разделов, используя ключевые слова `namespace` и `class`.

- Внутри класса `Form1` определены различные переменные, методы и события, относящиеся к игре.

2. Инициализация:

- В конструкторе `Form1` устанавливаются настройки формы игры, включая отключение кнопок управления окном и установку фокуса на форму.

- Создаются элементы пользовательского интерфейса, такие как метки (label) для отображения счета и заряда лазера.

3. Управление:

- Используя событие `KeyDown`, отслеживается нажатие клавиш на клавиатуре.

- Если нажата клавиша "X", создается и отображается лазер на экране.

- Также обрабатываются нажатия клавиш "W", "S", "A" и "D" для перемещения объекта "Rocket" вверх, вниз, влево и вправо соответственно.

4. Таймеры:

- Используются таймеры (`Timer`) для выполнения определенных задач с определенной частотой.

- `timer1` отвечает за создание новых метеоров на экране.

- `timer2` отвечает за анимацию движения метеоров и обнаружение столкновений с объектом "Rocket".

5. Обработка столкновений:

- При столкновении метеора с объектом "Rocket" происходит остановка таймеров и вывод сообщения о завершении игры.

- Изображение "Rocket" и метеора меняются на изображение взрыва.

6. Ускорение игры:

- Таймер `Score` отвечает за увеличение счета игры и ускорение движения и появления метеоров по мере прохождения времени.

7. Интерфейс пользователя:

- Используются элементы пользовательского интерфейса, такие как кнопки и метки, для отображения информации и взаимодействия с игрой.

8. Метеоры и лазеры: Создаются и отображаются метеоры на экране.

При нажатии клавиши "X" проверяется столкновение лазера с метеорами, и при обнаружении столкновения метеоры удаляются и заменяются изображением взрыва.

9. Закрытие игры: При нажатии кнопки "button1" создается и отображается форма меню ("Form2"), а текущая форма игры закрывается.

10. Готовность игрока: При щелчке по элементу pictureBox1 запускаются таймеры и начинается игра.

Вот краткое описание переменных и их назначения:

1. `disappearTimer` (Timer): Таймер для управления исчезновением лазера после выстрела.

2. `explosionTimer` (Timer): Таймер для анимации взрыва метеоритов при попадании лазера.

3. `PubLaser` (PictureBox): Переменная, хранящая текущий лазер на форме.

4. `PubMeteor` (List<PictureBox>): Список метеоритов, в которые попал лазер.

5. `x` (int): Переменная, хранящая заряд лазера.

6. `time` (int): Переменная, хранящая время игры (счет).

7. `pictureBoxes` (List<PictureBox>): Список метеоритов на форме.

8. `random` (Random): Объект Random для генерации псевдослучайных чисел.

9. `disappearTimer` (Timer): Таймер для управления исчезновением лазера после выстрела.

10. `Score` (Timer): Таймер для отслеживания и увеличения счета игры.

В коде также есть обработчики событий, которые выполняются при определенных действиях:

1. `Form1\_KeyDown`: Обработчик события нажатия клавиши на клавиатуре. В данном случае он обрабатывает нажатия клавиш W, A, S, D и X, отвечающих за управление игровым персонажем (ракета) и выстрелом лазера.

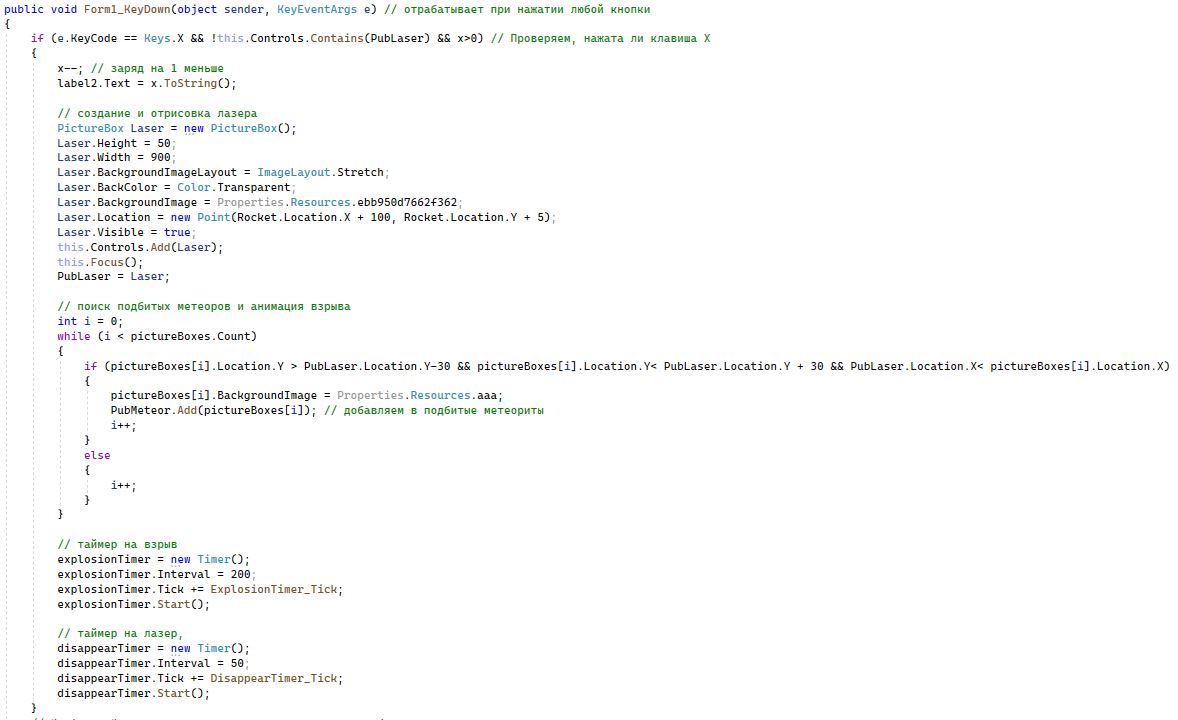


Рис.2 Реализация части функции при нажатии на X

2. `timer1\_Tick`: Обработчик события срабатывания таймера `timer1`. Он создает новый метеорит и добавляет его на форму.

Рис.3 Реализация таймера Добавления нового метеора

3. `timer2\_Tick`: Обработчик события срабатывания таймера `timer2`. Он отвечает за анимацию движения метеоритов, проверку столкновений и удаление метеоритов, вышедших за пределы формы.

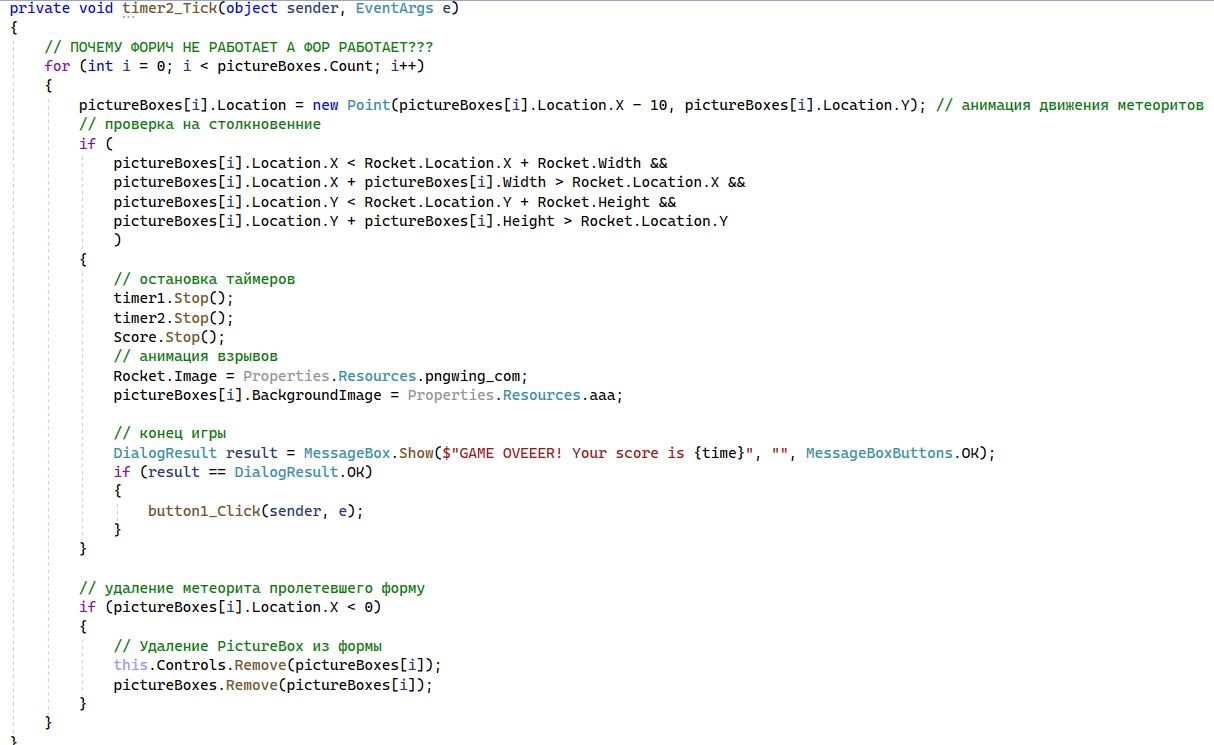


Рис.4 Реализация таймера на анимацию движения метеоров

4. `Score\_Tick`: Обработчик события срабатывания таймера `Score`. Он увеличивает счет игры и ускоряет движение и появление метеоритов.

5. `pictureBox1\_Click`: Обработчик события щелчка по `pictureBox1`. Он активирует игровые таймеры, начинает подсчет счета и удаляет кнопку "I'm ready".

Кроме того, в коде происходит настройка элементов пользовательского интерфейса, таких как форма, метки (`label1` и `label2`), фон, кнопки и изображения.

Для игры заранее создаются две формы, выставляются статические объекты, для которых WinForms автоматически создает код.

## **Заключение**

## В процессе создания игры на платформе WinForms, я, в качестве разработчика, использовал язык программирования C# и инструменты, предоставляемые средой разработки Visual Studio. Игра представляет собой простую аркадную игру, где игрок управляет ракетой и стреляет лазером, уничтожая метеориты.

## В ходе работы я реализовал функциональности, такие как обработка нажатия клавиш, создание и движение объектов на форме, проверку столкновений, анимацию взрывов и отображение информации о счете и заряде лазера. Таймеры использовались для управления временными интервалами и ускорения игры.

## В целом, задача по созданию игры на WinForms была успешно выполнена. В процессе работы я приобрел опыт работы с элементами управления формы, обработкой событий, работой с графическими ресурсами и управлением временем. Полученный результат - это игра, которая обладает базовой функциональностью и может быть доработана и расширена в дальнейшем.

## Создание игр - это увлекательный процесс, который требует творческого подхода и умения сочетать различные аспекты программирования. Я надеюсь, что данная игра на WinForms представляет интерес для пользователей и может стать отправной точкой для дальнейшего развития и совершенствования игровых проектов.

# **Список литературы**

1. Официальная документация Майкрософт по WinForms https://learn.microsoft.com/ru-ru/