

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»
(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)

ОТЧЕТ о курсовой работе

Тема Объектно-ориентированная разработка программ с графическим
пользовательским интерфейсом «сверху-вниз» «Игра «Помогите! На меня падает всякое»»

Обучающегося группы И943Б Любарчука А.Ф.
группа Фамилия и инициалы

Направление подготовки /
специальность 09.03.01 Информатика и вычислительная техника
индекс полное наименование направления подготовки / специальности

Направленность
образовательной программы Автоматизированные системы обработки
информации и управления
профиль / специализация / магистерская программа

Дисциплина (модуль) Информационные технологии и программирование

Руководитель: _____
подпись

_____ Землянская Е.Р.
ученая степень, ученое звание Фамилия ИО

Оценка: _____
« » _____ 20__ г.

Обучающийся: _____
подпись
_____ Любарчук А.Ф.
Фамилия ИО

« » _____ 20__ г.

РЕФЕРАТ

Отчет 20 с., 7 рис., 5 источн., 1 прил.

ООП, ГРАФИЧЕСКИЙ ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС,
НАСЛЕДОВАНИЕ, C#, WINDOWS FORMS

Объект разработки – интерактивное приложение с графическим пользовательским интерфейсом.

Целью курсовой работы является разработка интерактивного приложения с графическим пользовательским интерфейсом с использованием фреймворка .Net и технологии Windows Forms.

В процессе работы проводились исследования предметной области, были определены основные требования к разрабатываемому приложению, сформирована иерархия классов, выбраны средства разработки, спроектирован пользовательский интерфейс и структура приложения.

В результате разработки было создано интерактивное приложение с графическим пользовательским интерфейсом «TitanCutter».

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 Постановка задачи.....	6
2 Описание иерархии классов.....	8
2.1 Класс GameObject.....	8
2.2 Класс Player.....	9
2.3 Класс Titan.....	9
2.4 Класс FallingObstacle.....	10
2.5 Класс GameEngine.....	10
2.6 Класс GameForm.....	10
2.7 Класс MainMenuForm.....	11
2.8 Класс HighScoresManager.....	11
2.9 Класс ScoreEntry.....	11
2.10 Демонстрация иерархии классов.....	12
3 Используемые мультимедийные ресурсы и библиотеки.....	14
4 Демонстрация работы.....	16
ЗАКЛЮЧЕНИЕ.....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	20
ПРИЛОЖЕНИЕ А Исходные файлы курсовой работы.....	21

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяют следующие определения, обозначения и сокращения:

GameEngine — программный компонент, реализующий основную игровую логику, включая обработку игровых событий, управление объектами и подсчёт очков.

GUI (Graphical User Interface) — графический пользовательский интерфейс, обеспечивающий взаимодействие пользователя с программой с помощью визуальных элементов управления.

Player — игровой объект, управляемый пользователем с помощью клавиатуры.

SFX (Sound Effects) — звуковые эффекты, сопровождающие игровые события (удары, столкновения и т. п.).

Windows Forms (WF) — технология разработки графических пользовательских интерфейсов в среде .NET Framework, используемая для создания оконных приложений.

ВВЕДЕНИЕ

Курсовая работа посвящена объектно-ориентированной разработке программных приложений с графическим пользовательским интерфейсом с использованием языка программирования C# и фреймворка .NET. В качестве среды разработки используется Microsoft Visual Studio, а в качестве технологии создания пользовательского интерфейса применяется Windows Forms.

В рамках курсовой работы выполняется вариант “Игра «Помогите! На меня падает всякое»”, предполагающий разработку игрового приложения жанра аркада по типу «снег на голову», в котором реализуется взаимодействие пользователя с динамическими объектами в реальном времени. Выбор игровой тематики обусловлен возможностью наглядной демонстрации принципов объектно-ориентированного программирования, обработки пользовательского ввода, работы с графикой и звуком. Разработка игрового приложения позволяет объединить теоретические знания и практические навыки, полученные в процессе изучения дисциплины. Особое внимание уделяется корректной организации архитектуры программы и разделению ответственности между программными компонентами. Также рассматриваются вопросы обеспечения отзывчивости интерфейса и устойчивой работы приложения в реальном времени.

Целью курсовой работы является разработка интерактивного приложения с графическим пользовательским интерфейсом с использованием фреймворка .NET и технологии Windows Forms.

Для достижения цели необходимо решить следующие задачи:

- описать основные требования к разрабатываемому приложению;
- составить иерархию классов;
- разработать приложение;
- продемонстрировать работоспособность приложения.

1 Постановка задачи

В рамках курсовой работы требуется разработать интерактивное игровое приложение в жанре аркада, реализующее вариант «Игра “Помогите! На меня падает всякое”». Приложение должно функционировать в среде операционной системы Windows и обладать графическим пользовательским интерфейсом, обеспечивающим удобное взаимодействие пользователя с игровым процессом.

Игровой процесс заключается в управлении персонажем, расположенным в нижней части экрана. Персонаж может перемещаться влево и вправо с помощью клавиатуры. С верхней части игрового поля с определённой периодичностью падают различные объекты. Часть объектов представляет собой противников, которых необходимо уничтожать с помощью атаки, активируемой клавишей пробела. Удержание клавиши атаки допускается и приводит к непрерывному выполнению атакующих действий.

Помимо противников в игровом процессе присутствуют опасные объекты, представляющие собой падающие элементы окружения. Столкновение персонажа с такими объектами приводит к завершению игрового процесса. Таким образом, от игрока требуется одновременно уничтожать противников и избегать опасных препятствий, что повышает сложность и динамику игры.

Перед началом игрового процесса отображается информационный экран с описанием элементов управления. Данный экран предназначен для ознакомления пользователя с правилами и управлением и закрывается при нажатии любой клавиши клавиатуры, после чего начинается игра. В процессе игры пользователь может поставить игровой процесс на паузу и продолжить его позднее.

В игре осуществляется подсчёт статистических показателей, включающих время выживания, количество уничтоженных противников и общий игровой счёт. По завершении игры пользователю предоставляется возможность сохранить достигнутый результат. Сохранённые результаты

отображаются в отдельном разделе приложения в виде таблицы рекордов, отсортированной по убыванию набранных очков.

Приложение должно содержать главное меню, предоставляющее пользователю доступ к основным функциям программы. В главном меню реализована возможность начала новой игры, перехода к настройкам приложения, просмотра таблицы рекордов и завершения работы программы. Интерфейс меню должен быть интуитивно понятным и визуально оформленным с использованием фонового изображения.

В приложении используется графическое и звуковое сопровождение. Графическая часть включает отображение фона, персонажа и игровых объектов с использованием растровых изображений. Звуковое сопровождение включает фоновую музыку и звуковые эффекты, воспроизводимые при выполнении игровых действий. Предусмотрена возможность включения и отключения звукового сопровождения через настройки приложения.

В результате были сформированы требования, согласно которым будет разработано приложение:

- реализация игрового процесса в реальном времени;
- управление персонажем с помощью клавиатуры;
- поддержка удержания клавиши атаки;
- отображение фонового изображения и игровых объектов;
- воспроизведение фоновой музыки и звуковых эффектов;
- наличие главного меню и экрана настроек;
- реализация таблицы рекордов с сохранением результатов;
- возможность приостановки и возобновления игры;
- корректная обработка завершения игрового процесса.

2 Описание иерархии классов

В процессе разработки приложения была спроектирована объектно-ориентированная архитектура, основанная на разделении ответственности между классами, отвечающими за игровую логику, отображение, управление пользовательским вводом и хранение данных.

В результате были реализованы следующие классы:

- **GameObject** — базовый класс, описывающий общий интерфейс игровых объектов;
- **Player** — класс, описывающий управляемого пользователем персонажа;
- **Titan** — класс, описывающий противников, падающих сверху игрового поля;
- **FallingObstacle** — класс, описывающий опасные падающие объекты;
- **GameEngine** — класс, реализующий основную игровую логику;
- **GameForm** — класс главного игрового окна;
- **MainMenuForm** — класс главного меню приложения;
- **HighScoresManager** — класс, обеспечивающий сохранение и загрузку таблицы рекордов;
- **ScoreEntry** — класс, описывающий одну запись в таблице рекордов.

Далее будет приведено описание членов каждого из этих классов и интерфейса.

2.1 Класс **GameObject**

Класс **GameObject** является базовым классом для всех объектов, участвующих в игровом процессе.

Данный класс содержит следующие защищённые поля:

- **PointF Position** — используется для хранения текущего положения объекта на игровом поле;
- **SizeF Size** — определяет размеры объекта;
- **Float Speed** — определяет скорость перемещения объекта.

Также класс содержит следующие публичные методы:

- `void Update()` — используется для обновления состояния объекта при каждом игровом тике;
- `void Draw(Graphics g)` — используется для отрисовки объекта на экране с использованием переданного графического контекста.

2.2 Класс Player

Класс `Player` наследуется от класса `GameObject` и описывает игрового персонажа, управляемого пользователем.

Данный класс содержит следующие публичные свойства:

- `bool IsMovingLeft` — определяет, выполняется ли перемещение персонажа влево;
- `bool IsMovingRight` — определяет, выполняется ли перемещение персонажа вправо.

Класс реализует следующие публичные методы:

- `void Update()` — переопределяет метод базового класса и реализует логику перемещения персонажа в зависимости от текущего состояния клавиш управления;
- `RectangleF GetBounds()` — возвращает прямоугольную область, занимаемую персонажем, для использования при проверке столкновений.

2.3 Класс Titan

Класс `Titan` наследуется от класса `GameObject` и описывает противников, падающих сверху игрового поля.

Данный класс содержит следующие поля:

- `bool IsAlive` — определяет состояние противника;
- `int ScoreValue` — определяет количество очков, начисляемых за уничтожение противника.

Класс содержит следующие методы:

- `void Update()` — реализует вертикальное перемещение противника вниз по экрану;
- `bool CheckCollision(RectangleF playerBounds)` — определяет факт столкновения с персонажем.

2.4 Класс FallingObstacle

Класс FallingObstacle наследуется от класса GameObject и описывает опасные падающие объекты, которых необходимо избегать.

Данный класс не содержит дополнительных полей, однако использует базовую функциональность класса GameObject.

Основной метод класса void Update() — реализует падение объекта с заданной скоростью.

2.5 Класс GameEngine

Класс GameEngine отвечает за управление состоянием игры и реализацию основной игровой логики.

Данный класс содержит следующие поля:

- Player player — ссылка на объект управляемого персонажа;
- List<GameObject> objects — коллекция всех активных объектов;
- bool Paused — определяет состояние паузы;
- bool GameOver — определяет завершение игрового процесса;
- int Score — хранит текущее количество очков;
- int KilledTitans — хранит количество уничтоженных противников;
- DateTime startTime — для расчёта времени выживания.

Класс содержит следующие публичные методы:

- void Update() — обновляет состояние всех игровых объектов и выполняет проверку столкновений;
- void SpawnOne() — создаёт новый падающий объект;
- void PlayerAttack() — обрабатывает атаку игрока;
- void Render(Graphics g) — выполняет отрисовку игровых объектов.

2.6 Класс GameForm

Класс GameForm является главным игровым окном приложения.

Данный класс отвечает за:

- инициализацию игрового движка;
- обработку пользовательского ввода;

- управление таймерами игрового цикла;
- отображение игрового процесса;
- воспроизведение фоновой музыки.

Основные методы класса:

- OnPaint(PaintEventArgs e) — выполняет отрисовку игрового поля и элементов интерфейса;
- GameForm_KeyDown(KeyEventArgs e) — обрабатывает нажатия клавиш;
- GameForm_KeyUp(KeyEventArgs e) — обрабатывает отпускание клавиш.

2.7 Класс MainMenuForm

Класс MainMenuForm реализует главное меню приложения.

Данный класс обеспечивает переход между основными режимами работы программы и содержит методы обработки нажатий кнопок меню, а также реализацию полноэкранного режима по запросу пользователя.

2.8 Класс HighScoresManager

Класс HighScoresManager предназначен для работы с таблицей рекордов.

Класс содержит следующие публичные статические методы:

- List<ScoreEntry> LoadEntries() — загружает список рекордов из файла;
- void SaveEntry(string name, int score) — сохраняет новый результат.

2.9 Класс ScoreEntry

Класс ScoreEntry описывает одну запись в таблице рекордов.

Класс содержит следующие публичные свойства:

- string Name — имя игрока;
- int Score — количество набранных очков;
- DateTime Date — дата сохранения результата.

2.10 Демонстрация иерархии классов

Разработанная иерархия наглядно продемонстрирована на диаграмме классов, приведённой на рисунке 1.

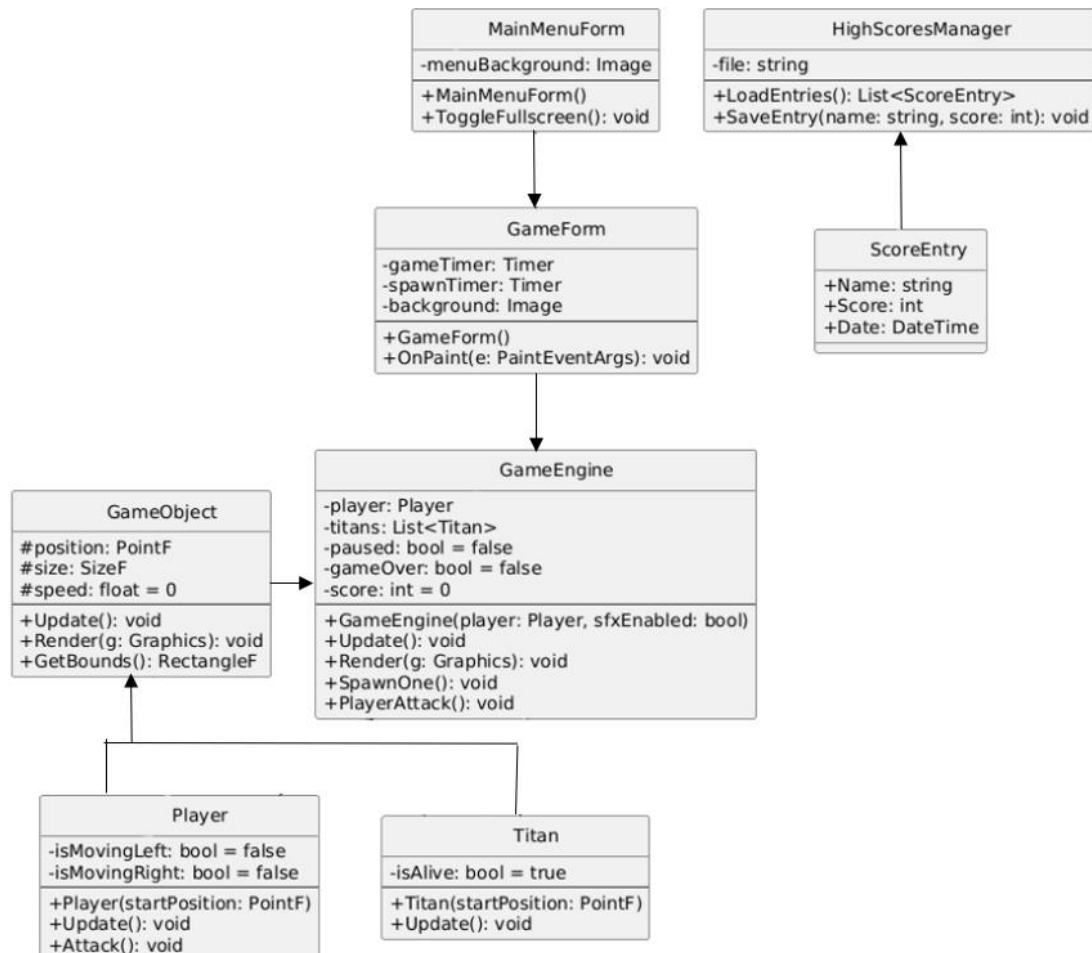


Рисунок 1 – Диаграмма классов иерархии «TitanCutter»

На данной диаграмме отображены классы GameObject, Player, Titan, GameEngine, GameForm, MainMenuForm, HighScoresManager и ScoreEntry. Класс GameObject является базовым и используется для описания общих характеристик игровых объектов. От данного класса наследуются классы Player и Titan, что отражено отношением наследования на диаграмме.

Для каждого класса на диаграмме перечислены его поля и методы. Перед каждым членом класса указан спецификатор доступа, где символ «-» обозначает закрытый доступ (private), символ «+» — открытый доступ (public), символ «#» — защищённый доступ (protected).

Поля классов записаны в формате: спецификатор доступа имя поля: тип =

значение по умолчанию. Методы представлены в формате спецификатор доступа имя метода (параметр: тип), тип возвращаемого значения.

Класс GameEngine инкапсулирует основную игровую логику, управляет состоянием игры, объектом игрока и коллекцией противников, а также реализует методы обновления игрового процесса и обработки атак.

Классы GameForm и MainMenuForm отвечают за графический пользовательский интерфейс приложения, отображение игровых сцен и взаимодействие с пользователем.

Класс HighScoresManager предназначен для управления таблицей рекордов, а класс ScoreEntry используется для хранения информации об отдельной записи результата.

Связи между классами отображают реальные зависимости компонентов приложения, включая отношения наследования, агрегации и ассоциации. Все связи на диаграмме выполнены в ортогональной форме, что соответствует требованиям ГОСТ и обеспечивает наглядность структуры приложения.

Разработанная иерархия классов позволяет реализовать поставленную задачу в соответствии со сформулированными требованиями.

3 Используемые мультимедийные ресурсы и сторонние библиотеки

В процессе разработки приложения были использованы ресурсы, обеспечивающие визуальное и звуковое сопровождение игрового процесса и библиотеки .Net.

Для визуального оформления приложения применялись графические изображения в формате PNG. В частности, использовались изображения фона игрового окна, фона главного меню, а также спрайты игровых объектов, включая персонажа игрока и падающие объекты. Все изображения используются для отрисовки элементов интерфейса и игровых сцен. Скриншоты применяемых графических ресурсов приведены в приложении к пояснительной записке.

Для звукового сопровождения игры использовались аудиофайлы в формате WAV. Музыкальная композиция применяется в качестве фоновой музыки во время игрового процесса. Дополнительные звуковые эффекты используются для сопровождения игровых событий, таких как атака персонажа и взаимодействие с объектами. Звуковые ресурсы загружаются и воспроизводятся с использованием стандартных средств платформы .NET. Скриншоты используемых аудиофайлов также приведены в приложении.

В качестве шрифтов применялись стандартные системные шрифты операционной системы Windows, в частности шрифт Segoe UI, который используется для отображения текстовой информации и элементов интерфейса. Дополнительные пользовательские шрифты в приложении не применялись.

Сторонние библиотеки в процессе разработки приложения не использовались. Для реализации функциональности были задействованы стандартные библиотеки платформы .NET, включая пространства имён System.Windows.Forms, System.Drawing, System.Media, System.IO и System.Collections.Generic, которые обеспечивают работу с графическим

интерфейсом, графикой, звуком, файловой системой и коллекциями данных соответственно.

Использование приведённых ресурсов и библиотек позволит упростить процесс разработки приложения.

4 Демонстрация работы

При входе в приложение доступно меню с возможностями начать игру, зайти в настройки, зайти в список результатов, выйти из игры, представлено на рисунке 2.

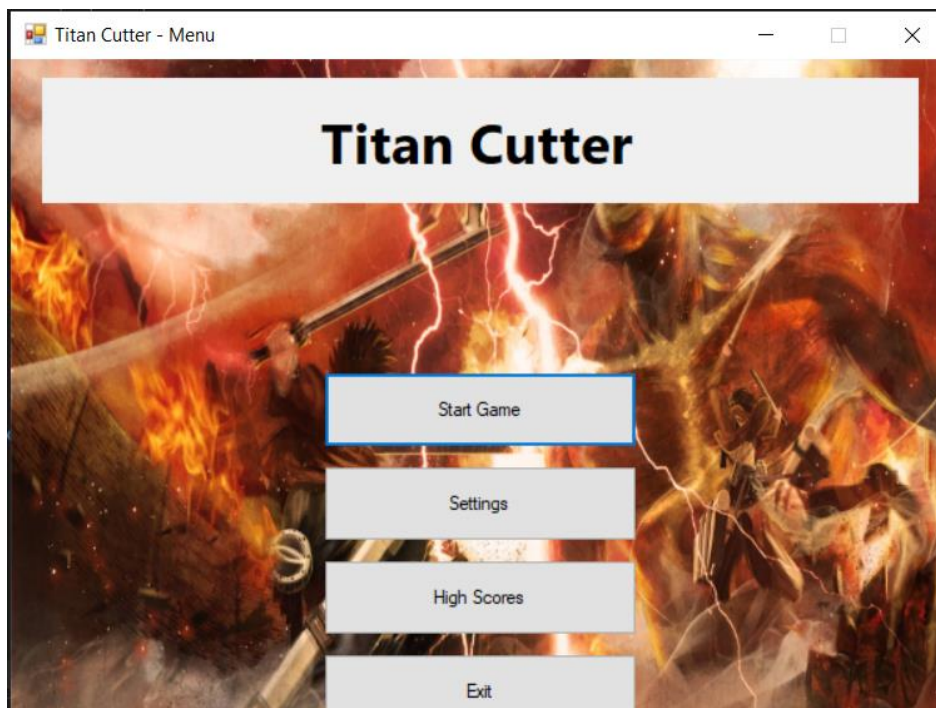


Рисунок 2 – Главное меню игры

При входе в пункт Settings, можно включить или выключить музыку и звуки, а также поменять сложность игры, представлено на рисунке 3.

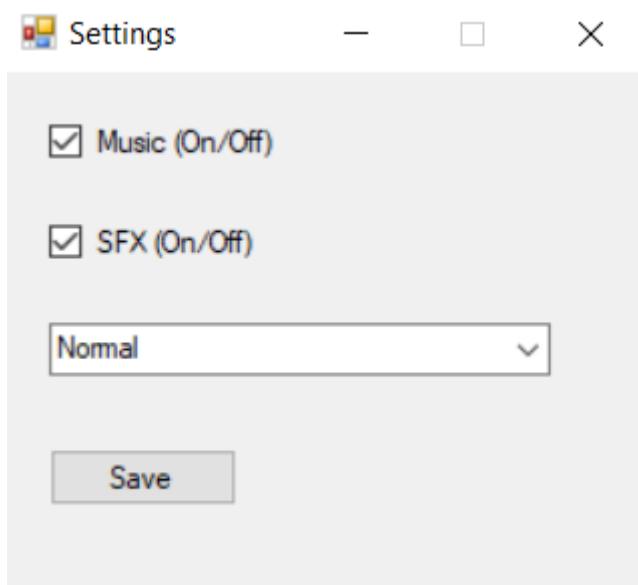


Рисунок 3 – Настройки

При входе в пункт High Scores, можно увидеть результаты игр прошлых игроков, представлено на рисунке 4.

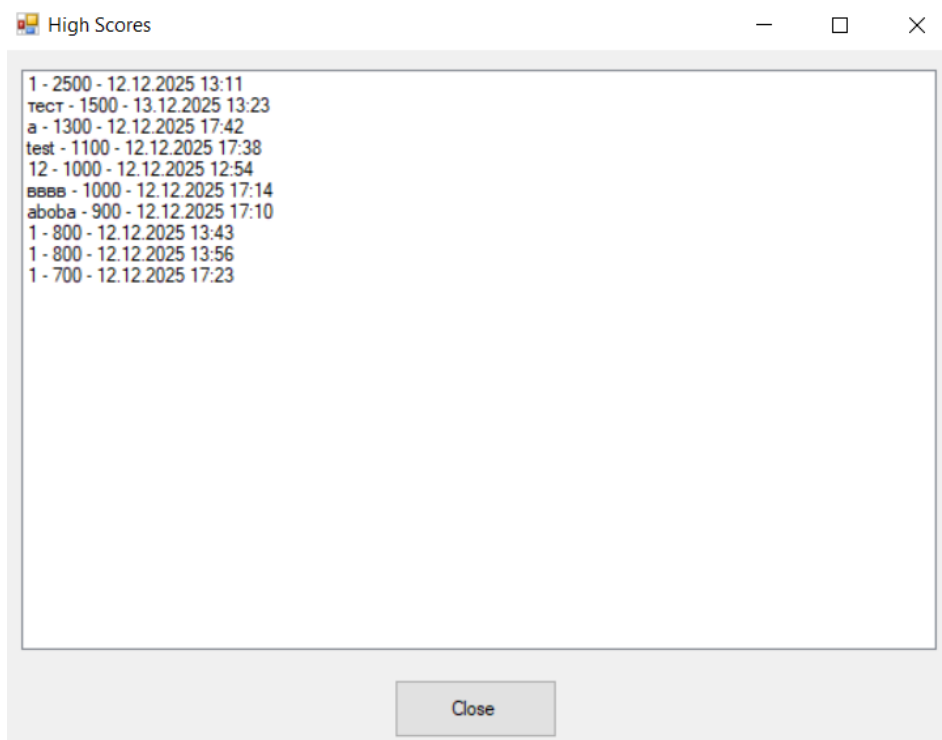


Рисунок 4 – Результаты игр

При входе в пункт Start Game показываются клавиши управления, представлено на рисунке 5.

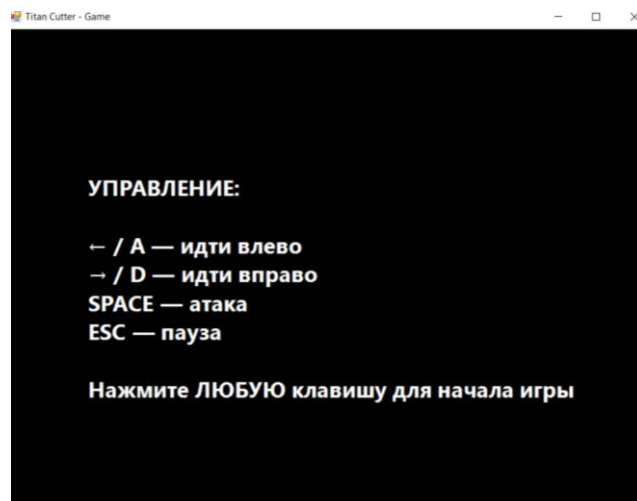


Рисунок 5 – Управление

После нажатия на любую кнопку клавиатуры начинается игра, где с помощью атаки надо убивать титанов и уворачиваться от камней, представлено на рисунке 6.

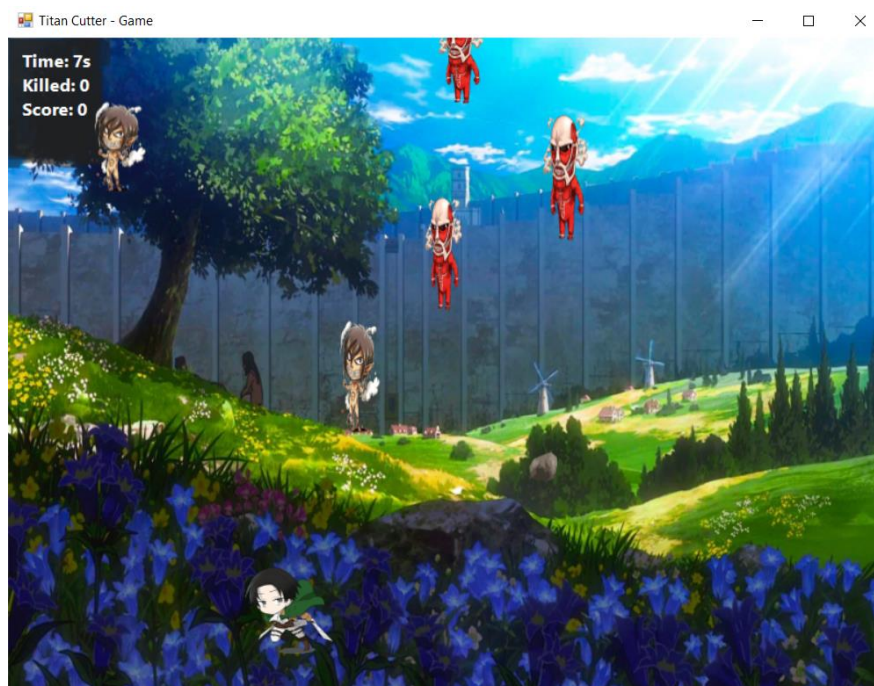


Рисунок 6 – Демонстрация игры

При нажатии клавиши escape во время игры ставится пауза, представлено на рисунке 7.

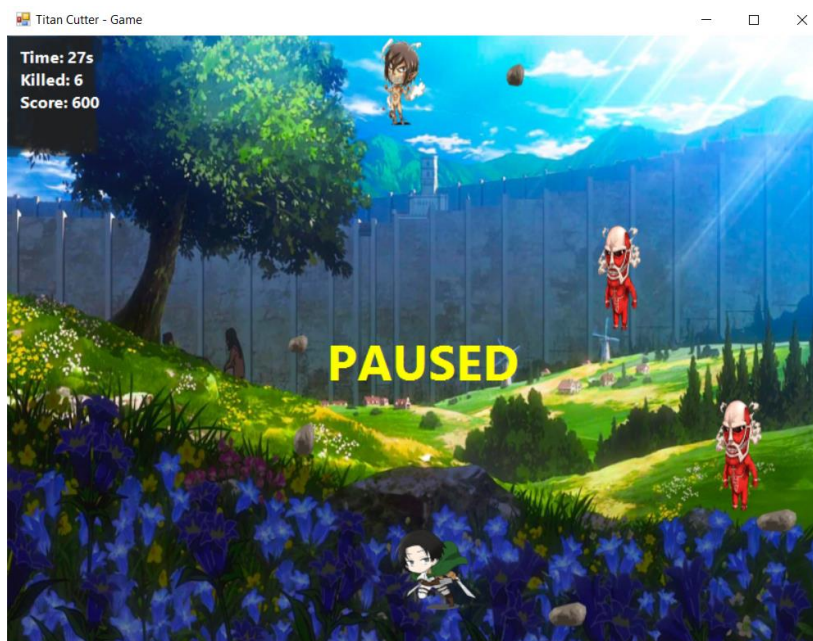


Рисунок 7 - Пауза

Демонстрация возможностей приложения подтверждает его соответствие выдвинутым требованиям и поставленной задаче.

ЗАКЛЮЧЕНИЕ

В рамках курсовой работы было разработано приложение с графическим пользовательским интерфейсом «Игра “Помогите! На меня падает всякое”».

В ходе выполнения курсовой работы были определены требования к разрабатываемому приложению. Были описаны правила игры, основные функции, требования к интерфейсу, а также к работе с графикой и звуком.

В процессе проектирования была разработана иерархия классов приложения. В результате была создана структура программы, включающая игровые объекты, игровой движок и формы пользовательского интерфейса, что упростило реализацию логики игры и дальнейшую доработку приложения.

В ходе разработки приложения была реализована игра с использованием фреймворка .NET и технологии Windows Forms. Приложение обеспечивает обработку пользовательского ввода с клавиатуры, отображение графических элементов, воспроизведение звуковых эффектов и фоновой музыки, а также корректную работу игрового процесса.

Также была проведена демонстрация работоспособности приложения. В результате было подтверждено, что все основные функции работают корректно, а приложение соответствует поставленным требованиям.

Все поставленные задачи выполнены, цель достигнута. Исходные тексты разработанной программы представлены в приложении А.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Microsoft. Документация по Windows Forms. – URL:
<https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/> (дата обращения: 13.12.2025).
- 2 Microsoft. Класс SoundPlayer (System.Media). – URL:
<https://learn.microsoft.com/ru-ru/dotnet/api/system.media.soundplayer> (дата обращения: 13.12.2025).
- 3 Fandom. Attack on Titan. – URL:
https://attackontitan.fandom.com/ru/wiki/Атака_Титанов (дата обращения: 13.12.2025).
- 4 Microsoft. Язык программирования C#. Руководство. – URL:
<https://learn.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 13.12.2025).
- 5 Wikipedia. Диаграмма классов UML. – URL:
https://ru.wikipedia.org/wiki/Диаграмма_классов (дата обращения: 13.12.2025).

ПРИЛОЖЕНИЕ А

Исходные файлы курсовой работы

Исходные файлы приложения представлен в виде ссылки для скачивания: <https://disk.yandex.ru/d/sJ5Uta2EYoXfPA> или <https://github.com/4ector2/TitanCutter> .