



**направление подготовки 09.03.02 «Информационные системы и технологии»,
профиль «Автоматизированные системы обработки информации и управления»**

КУРСОВОЙ ПРОЕКТ

Тема: Разработка оконной игры "змейка" на языке C# с помощью Windows Forms

Дисциплина: Объектно-ориентированное программирование (C#)

Выполнил: студент группы 221-374
Орлова Е.А.

Дата, подпись _____
(Дата) (Подпись)

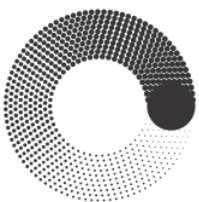
Проверила: _____
(Оценка)

Дата, подпись _____
(Дата) (Подпись)

Замечания:

**Москва
2023**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**



**Факультет информационных технологий
Кафедра Информатики и информационных технологий**

**направление подготовки 09.03.02 «Информационные системы и
технологии»,
профиль «Автоматизированные системы обработки информации и
управления»**

УТВЕРЖДАЮ

Зав. каф., к.т.н. Е.В. Булатников

« » _____ 2023 г.

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Студент Орлова Екатерина Артемьевна группа 221—374

**Тема: Разработка оконной игры "змейка" на языке C# с помощью Windows
Forms**

1. **Срок представления работы к защите** « » _____ 2023 г.
2. **Исходные данные для выполнения работы:**
3. **Содержание курсового проекта:** введение, аналитическая часть,
практическая часть, библиографический список
4. **Перечень графического материала:** диаграмма UML, листинги
5. **Литература и прочие материалы, рекомендуемые студенту для
изучения:** указаны в библиографическом списке
6. **Дата выдачи задания:** « » _____ 2023 г.
7. **Руководитель:** _____ / _____ /
(Подпись) (ФИО)
8. **Задание к исполнению принял:** _____ / _____ /
(Подпись) (ФИО)

Введение

Курсовая работа представляет собой реализацию классической игры "Змейка" на языке программирования C# с использованием технологии Windows Forms. Проект организован с применением принципов объектно-ориентированного программирования (ООП) для достижения более четкой структуры и повышения читаемости кода.

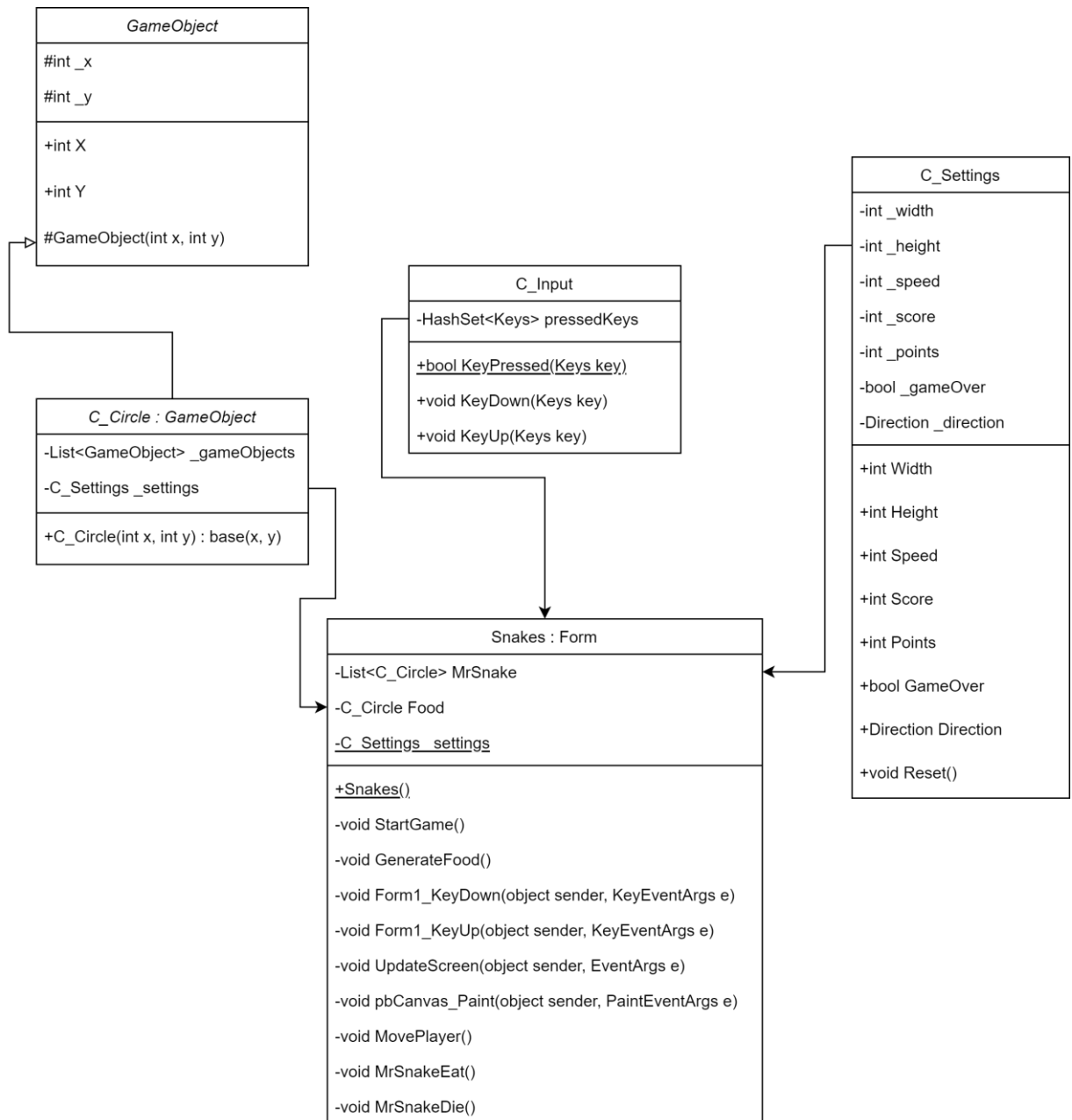
Основные компоненты проекта включают в себя классы C_Circle, C_Settings, C_Input, и Snakes, а также класс GameObject, представляющий базовый объект игры.

Проект активно использует коллекции, такие как List, для управления сущностями на игровом поле. Класс C_Settings инкапсулирует настройки игры, такие как размеры элементов, скорость движения, и текущее направление движения змеи.

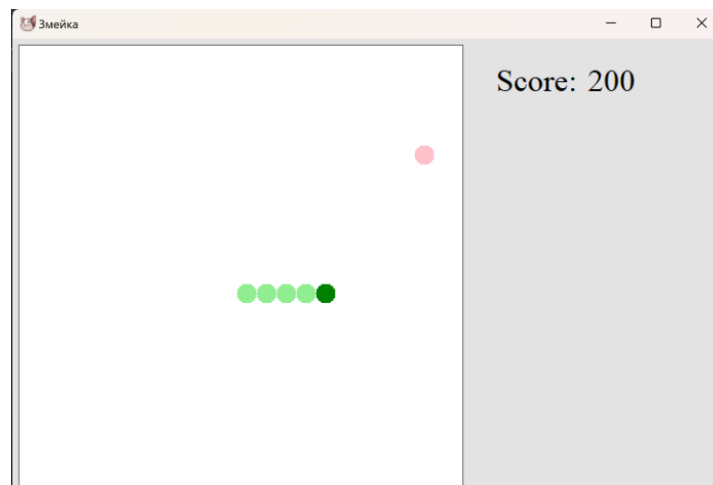
Реакция на пользовательский ввод обеспечивается с использованием статического класса C_Input, который отслеживает нажатия клавиш и управляет игровым процессом в зависимости от действий пользователя.

Проект также реализует обработку логики игры, включая движение змеи, её питание (поедание еды), отслеживание столкновений и управление игровым циклом через таймер.

Диаграмма UML



Игра



Листинг кода

Файл GameObject.cs:

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Snake
{
    internal abstract class GameObject
    {
        protected int _x;
        protected int _y;

        public int X
        {
            get { return _x; }
            set { _x = value; }
        }

        public int Y
        {
            get { return _y; }
            set { _y = value; }
        }

        // Конструктор абстрактного класса GameObject.
        protected GameObject(int x, int y)
        {
        }
    }
}
```

Файл C_Circle.cs:

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Snake
{
    internal class C_Circle : GameObject
    {
        private List<GameObject> _gameObjects;
        private C_Settings _settings;

        // Конструктор класса C_Circle.
        public C_Circle(int x, int y) : base(x, y)
        {
            _x = x;
            _y = y;
        }
    }
}
```

Файл C_Settings.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Snake
{
    public enum Direction { Left, Right, Up, Down }
```

```
internal class C_Settings
{
    private int _width;
    private int _height;
    private int _speed;
    private int _score;
    private int _points;
    private bool _gameOver;
    private Direction _direction;

    public int Width
    {
        get { return _width; }
        set { _width = value; }
    }

    public int Height
    {
        get { return _height; }
        set { _height = value; }
    }

    public int Speed
    {
        get { return _speed; }
        set { _speed = value; }
    }

    public int Score
    {
        get { return _score; }
        set { _score = value; }
    }
}
```

```

public int Points
{
    get { return _points; }
    set { _points = value; }
}

public bool GameOver
{
    get { return _gameOver; }
    set { _gameOver = value; }
}

public Direction Direction
{
    get { return _direction; }
    set { _direction = value; }
}

```

// Конструктор класса C_Settings.

```

public C_Settings()
{
    _width = 20;
    _height = 20;
    _speed = 10;
    _score = 0;
    _points = 50;
    _gameOver = false;
    _direction = Direction.Down;
}

```

// Метод сброса настроек игры.

```

public void Reset()
{
    _score = 0;
}

```



```

        _gameOver = false;
        _direction = Direction.Down;
    }
}
}

```

Файл C_Input.cs:

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Snake
{

    public static class C_Input
    {
        private static HashSet<Keys> pressedKeys = new
HashSet<Keys>();

        // Метод определения, была ли клавиша нажата.
        public static bool KeyPressed(Keys key)
        {
            return pressedKeys.Contains(key);
        }

        // Метод обработки нажатия клавиши.
        public static void KeyDown(Keys key)
        {
            pressedKeys.Add(key);
        }
    }
}

```

```

    }

    // Метод обработки отпускания клавиши.
    public static void KeyUp(Keys key)
    {
        pressedKeys.Remove(key);
    }
}
}

```

Файл Form1.cs:

```

using Snake.Properties;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Snake
{
    public partial class Snakes : Form
    {
        private List<GameObject> MrSnake = new List<GameObject>();
        private C_Circle Food = new C_Circle(0, 0);
        private C_Settings _settings;

        // Конструктор формы Snakes.
        public Snakes()
        {
            InitializeComponent();

```

```

        _settings = new C_Settings();

        this.KeyDown += Form1_KeyDown;
        this.KeyUp += Form1_KeyUp;
        this.Focus();

        game_timer.Interval = 1000 / _settings.Speed;
        game_timer.Tick += UpdateScreen;
        game_timer.Start();

        StartGame();
    }

```

// Метод начала игры.

```

    private void StartGame()
    {
        lblGameOver.Visible = false;

        _settings.Reset();

        MrSnake.Clear();
        C_Circle head = new C_Circle(15, 5);
        MrSnake.Add(head);

        lblScore.Text = _settings.Score.ToString();
        GenerateFood();
    }

```

// Метод генерации еды на поле.

```

    private void GenerateFood()
    {
        int maxXPos = pbCanvas.Size.Width / _settings.Width;
        int maxYPos = pbCanvas.Size.Height / _settings.Height;
    }

```

```
Random random = new Random();  
Food.X = random.Next(0, maxXPos);  
Food.Y = random.Next(0, maxYPos);  
}
```

// Обработчик нажатия клавиши на форме.

```
private void Form1_KeyDown(object sender, EventArgs e)  
{  
    C_Input.KeyDown(e.KeyCode);  
}
```

// Обработчик отпускания клавиши на форме.

```
private void Form1_KeyUp(object sender, EventArgs e)  
{  
    C_Input.KeyUp(e.KeyCode);  
}
```

// Метод обновления экрана и игровой логики.

```
private void UpdateScreen(object sender, EventArgs e)  
{  
    if (_settings.GameOver)  
    {  
        if (C_Input.KeyPressed(Keys.Enter))  
        {  
            StartGame();  
        }  
    }  
    else  
    {  
        if (C_Input.KeyPressed(Keys.Right) &&  
_settings.Direction != Direction.Left)  
        {  
            _settings.Direction = Direction.Right;  
        }  
        if (C_Input.KeyPressed(Keys.Left) &&
```

```

_settings.Direction != Direction.Right)
    {
        _settings.Direction = Direction.Left;
    }
    if (C_Input.KeyPressed(Keys.Down) &&
_settings.Direction != Direction.Up)
    {
        _settings.Direction = Direction.Down;
    }
    if (C_Input.KeyPressed(Keys.Up) &&
_settings.Direction != Direction.Down)
    {
        _settings.Direction = Direction.Up;
    }

    if (C_Input.KeyPressed(Keys.R))
    {
        MrSnakeDie();
    }

    MovePlayer();
}

pbCanvas.Invalidate();
}

```

// Метод движения змеи и обработки столкновений.

```

private void MovePlayer()
{
    for (int i = MrSnake.Count - 1; i >= 0; i--)
    {
        if (i == 0)
        {
            switch (_settings.Direction)

```

```

{
    case Direction.Left:
        MrSnake[i].X--;
        break;
    case Direction.Right:
        MrSnake[i].X++;
        break;
    case Direction.Down:
        MrSnake[i].Y++;
        break;
    case Direction.Up:
        MrSnake[i].Y--;
        break;
}
for (int j = 1; j < MrSnake.Count; j++)
{
    if (MrSnake[i].X == MrSnake[j].X &&
MrSnake[i].Y == MrSnake[j].Y)
    {
        MrSnakeDie();
    }
}

if (MrSnake[0].X == Food.X && MrSnake[0].Y ==
Food.Y)
{
    MrSnakeEat();
}
else
{
    MrSnake[i].X = MrSnake[i - 1].X;
    MrSnake[i].Y = MrSnake[i - 1].Y;
}

```

```

        int MaxXPos = pbCanvas.Size.Width / _settings.Width;
        int MaxYPos = pbCanvas.Size.Height /
_settings.Height;

        if (MrSnake[0].X < 0)
            MrSnake[0].X = MaxXPos - 1;
        if (MrSnake[0].X >= MaxXPos)
            MrSnake[0].X = 0;
        if (MrSnake[0].Y < 0)
            MrSnake[0].Y = MaxYPos - 1;
        if (MrSnake[0].Y >= MaxYPos)
            MrSnake[0].Y = 0;
    }
}

```

// Метод отрисовки на поле змеи и еды.

```

private void pbCanvas_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    // Отрисовка змеи и еды.
    if (!_settings.GameOver)
    {
        Brush MrSnakeColour;
        for (int i = 0; i < MrSnake.Count; i++)
        {
            if (i == 0)
            {
                MrSnakeColour = Brushes.Green;
            }
            else
            {
                MrSnakeColour = Brushes.LightGreen;
            }

```

```

        g.FillEllipse(MrSnakeColour,
            new Rectangle(MrSnake[i].X *
                _settings.Width, MrSnake[i].Y * _settings.Height,
                    _settings.Width, _settings.Height));
    }
    g.FillEllipse(Brushes.Pink,
        new Rectangle(Food.X * _settings.Width, Food.Y *
            _settings.Height,
                _settings.Width, _settings.Height));
}

```

// Отображение сообщения о конце игры.

```

    else
    {
        string gameOver = $"Конец игры\nВаш счёт:
{_settings.Score}\nНажмите Enter";
        lblGameOver.Text = gameOver;
        lblGameOver.Visible = true;
    }
}

```

// Метод обработки поедания еды змеей.

```

private void MrSnakeEat()
{
    C_Circle food = new C_Circle(0, 0);
    food.X = MrSnake[MrSnake.Count - 1].X;
    food.Y = MrSnake[MrSnake.Count - 1].Y;

    MrSnake.Add(food);

    _settings.Score += _settings.Points;
}

```



```
        lblScore.Text=_settings.Score.ToString();

        GenerateFood();
    }

// Метод обработки смерти змеи.
    private void MrSnakeDie()
    {
        _settings.GameOver = true;
    }
}
```

Библиографический список

1. Шилдт Г. "C# 4.0: The Complete Reference" — М.: Вильямс, 2010. — 992 с.
2. Троелсен Э. "Программирование на C# 7 и .NET Core 2 для профессионалов" — Питер, 2018. — 1272 с.
3. Albahari J., Albahari B. "C# 7.0 in a Nutshell: The Definitive Reference" — O'Reilly Media, 2017. — 1072 с.
4. Freeman A., Freeman S. "Head First Design Patterns" — O'Reilly Media, 2004. — 694 с.
5. Nystrom R. "Game Programming Patterns" — Genever Benning, 2014. — 354 с.
6. Martin R.C. "Clean Code: A Handbook of Agile Software Craftsmanship" — Prentice Hall, 2008. — 464 с.