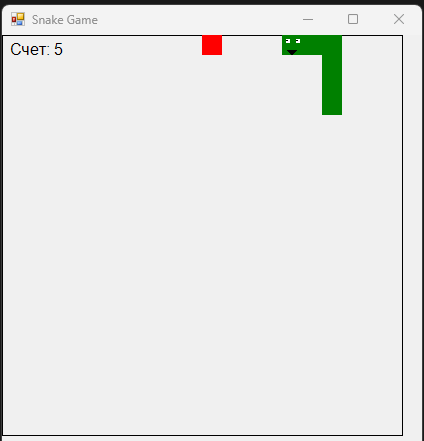
Создание игры "Змейка (Snake)", похожей на классическую версию, которая была на кнопочных телефонах Nokia. Ниже приведен пример реализации этой игры на C# с использованием Windows Forms.



**Шаги по созданию игры "Змейка"**

1. **Создайте новый проект Windows Forms Application** в Visual Studio.
2. **Настройка формы**:
   * Установите следующие свойства для формы:
     + **Text**: Snake Game
     + **ClientSize**: 400, 400 (или любую другую подходящую величину)
     + **DoubleBuffered**: True (чтобы избежать мерцания при перерисовке)
3. **Код игры**. Вот пример кода для игры "Змейка":

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Media; // Добавьте это пространство имен для работы со звуком

using System.Windows.Forms;

namespace Snake\_Game

{

public partial class SnakeGame: Form

{

private const int CellSize = 20; // Размер клетки

private const int Width = 20; // Ширина игрового поля

private const int Height = 20; // Высота игрового поля

private List<Point> snake; // Список сегментов змейки

private Point food; // Позиция еды

private int direction; // Направление движения змейки

private Timer timer; // Таймер для управления игровой логикой

private Button newGameButton; // Кнопка для новой игры

private int score; // Счет игрока

private SoundPlayer eatSound; // Звук при поедании еды

private SoundPlayer gameOverSound; // Звук при столкновении

public SnakeGame()

{

InitializeComponent();

InitializeGame(); // Вызов инициализации игры

eatSound = new SoundPlayer( "eat.wav" ); // Звук поедания

gameOverSound = new SoundPlayer( "gameOver.wav" ); // Звук столкновения

}

private void InitializeGame()

{

this.Text = "Snake Game";

// Увеличиваем высоту для счёта

this.ClientSize = new Size( Width \* CellSize + 20, Height \* CellSize + 60 );

this.DoubleBuffered = true;

// Инициализация змейки и направления

snake = new List<Point>

{

new Point(Width / 2, Height / 2) // Начальная позиция змейки в центре

};

direction = 0; // Начальное направление (вправо)

score = 0; // Сброс счета

// Генерация еды

GenerateFood();

// Настройка таймера

timer = new Timer();

// Скорость игры (змеи)

timer.Interval = 300; // Увеличенный интервал обновления (мс)

timer.Tick += Update;

timer.Start();

// Обработчик нажатий клавиш

this.KeyDown += new KeyEventHandler( OnKeyDown );

// Создание кнопки для новой игры

newGameButton = new Button();

newGameButton.Text = "Новая игра";

newGameButton.Location = new Point( Width \* CellSize / 2 - 50, Height \* CellSize / 2 - 20 );

newGameButton.Size = new Size( 100, 40 );

newGameButton.Visible = false; // Скрываем кнопку в начале

newGameButton.Click += NewGameButton\_Click;

this.Controls.Add( newGameButton );

}

private void GenerateFood()

{

Random rand = new Random();

int x, y;

do

{

x = rand.Next( Width );

y = rand.Next( Height );

} while (snake.Contains( new Point( x, y ) )); // Убедимся, что еда не появляется на змейке

food = new Point( x, y );

}

protected override void OnPaint( PaintEventArgs e )

{

base.OnPaint( e );

DrawGame( e.Graphics );

}

private void DrawGame( Graphics g )

{

// Отрисовка рамки

g.DrawRectangle( Pens.Black, 0, 0, Width \* CellSize, Height \* CellSize );

// Отрисовка головы змеи

bool isGameOver = !timer.Enabled; // Проверка, закончена ли игра

// Отрисовка змейки

for (int i = 0; i < snake.Count; i++)

{

g.FillRectangle( Brushes.Green, snake[i].X \* CellSize, snake[i].Y \* CellSize, CellSize, CellSize );

}

// Отрисовка еды

g.FillRectangle( Brushes.Red, food.X \* CellSize, food.Y \* CellSize, CellSize, CellSize );

// Отрисовка счета

g.DrawString( "Счет: " + score, new Font( "Arial", 12 ), Brushes.Black, new PointF( 5, 5 ) );

// Рисуем глаза в зависимости от состояния игры

DrawSnakeEye( g, snake[0], isGameOver );

// Рисуем рот змеи

DrawSnakeMouth( g, snake[0], isGameOver );

}

private void DrawSnakeEye( Graphics g, Point head, bool isGameOver )

{

int eyeSize = 5; // Размер глаза

// После столкновения

if (isGameOver)

{

// Рисуем крестики вместо глаз

int crossSize = 5; // Размер крестика

// Левый крестик

g.DrawLine( Pens.Red, head.X \* CellSize + CellSize / 4 - eyeSize / 2, head.Y \* CellSize + CellSize / 4 - eyeSize / 2,

head.X \* CellSize + CellSize / 4 + eyeSize / 2, head.Y \* CellSize + CellSize / 4 + eyeSize / 2 );

g.DrawLine( Pens.Red, head.X \* CellSize + CellSize / 4 - eyeSize / 2, head.Y \* CellSize + CellSize / 4 + eyeSize / 2,

head.X \* CellSize + CellSize / 4 + eyeSize / 2, head.Y \* CellSize + CellSize / 4 - eyeSize / 2 );

// Правый крестик

g.DrawLine( Pens.Red, head.X \* CellSize + 3 \* CellSize / 4 - eyeSize / 2, head.Y \* CellSize + CellSize / 4 - eyeSize / 2,

head.X \* CellSize + 3 \* CellSize / 4 + eyeSize / 2, head.Y \* CellSize + CellSize / 4 + eyeSize / 2 );

g.DrawLine( Pens.Red, head.X \* CellSize + 3 \* CellSize / 4 - eyeSize / 2, head.Y \* CellSize + CellSize / 4 + eyeSize / 2,

head.X \* CellSize + 3 \* CellSize / 4 + eyeSize / 2, head.Y \* CellSize + CellSize / 4 - eyeSize / 2 );

}

else

{

// Рисуем глаза

// Левый глаз

g.FillEllipse( Brushes.White, head.X \* CellSize + CellSize / 4 - eyeSize / 2, head.Y \* CellSize + CellSize / 4 - eyeSize / 2, eyeSize, eyeSize );

g.FillEllipse( Brushes.Black, head.X \* CellSize + CellSize / 4 - eyeSize / 4, head.Y \* CellSize + CellSize / 4 - eyeSize / 4, eyeSize / 2, eyeSize / 2 );

// Правый глаз

g.FillEllipse( Brushes.White, head.X \* CellSize + 3 \* CellSize / 4 - eyeSize / 2, head.Y \* CellSize + CellSize / 4 - eyeSize / 2, eyeSize, eyeSize );

g.FillEllipse( Brushes.Black, head.X \* CellSize + 3 \* CellSize / 4 - eyeSize / 4, head.Y \* CellSize + CellSize / 4 - eyeSize / 4, eyeSize / 2, eyeSize / 2 );

}

}

private void DrawSnakeMouth( Graphics g, Point head, bool isGameOver )

{

// Проверяем направление движения

if (direction == 1) // Если движение вверх

{

// Не рисуем рот

return;

}

// Рисуем рот

if (isGameOver)

{

// Рисуем грустный смайлик

g.DrawArc( Pens.Black, head.X \* CellSize + CellSize / 4, head.Y \* CellSize + 3 \* CellSize / 4, CellSize / 2, CellSize / 2, 0, -180 ); // Улыбка

//g.FillEllipse( Brushes.Black, head.X \* CellSize + CellSize / 4 + 10, head.Y \* CellSize + CellSize / 4 - 5, 5, 5 ); // Левый глаз

g.FillEllipse( Brushes.Black, head.X \* CellSize + CellSize / 4 + 25, head.Y \* CellSize + CellSize / 4 - 5, 5, 5 ); // Правый глаз

}

else

{

// Рисуем обычный рот

g.FillPolygon( Brushes.Black, new Point[]

{

//new Point(head.X \* CellSize + CellSize / 4, head.Y \* CellSize + CellSize / 4),

//new Point(head.X \* CellSize + 3 \* CellSize / 4, head.Y \* CellSize + CellSize / 4),

//new Point(head.X \* CellSize + CellSize / 2, head.Y \* CellSize + CellSize / 2)

new Point(head.X \* CellSize + CellSize / 4, head.Y \* CellSize + 3 \* CellSize / 4), // Смещаем Y вниз

new Point(head.X \* CellSize + 3 \* CellSize / 4, head.Y \* CellSize + 3 \* CellSize / 4), // Смещаем Y вниз

new Point(head.X \* CellSize + CellSize / 2, head.Y \* CellSize + CellSize) // Смещаем Y вниз

} );

}

}

private void Update( object sender, EventArgs e )

{

// Движение змейки

Point newHead = snake[0];

switch (direction)

{

case 0: // Вправо

newHead.X++;

break;

case 1: // Вверх

newHead.Y--;

break;

case 2: // Влево

newHead.X--;

break;

case 3: // Вниз

newHead.Y++;

break;

}

// Проверка на столкновение с границей или самой собой

if (newHead.X < 0 || newHead.X >= Width || newHead.Y < 0 || newHead.Y >= Height || snake.Contains( newHead ))

{

gameOverSound.Play(); // Воспроизведение звука при столкновении

timer.Stop(); // Остановка игры

newGameButton.Visible = true; // Показываем кнопку новой игры

MessageBox.Show( "Игра окончена! Ваш счет: " + score ); // Сообщение об окончании игры

return;

}

// Добавление новой головы

snake.Insert( 0, newHead );

// Проверка на поедание еды

if (newHead == food)

{

score++; // Увеличиваем счет

eatSound.Play(); // Воспроизведение звука при поедании

GenerateFood(); // Генерация новой еды

// Увеличение скорости игры при увеличении счета

if (score % 5 == 0) // Каждые 5 очков увеличиваем скорость

{

timer.Interval = Math.Max( 50, timer.Interval - 10 ); // Уменьшаем интервал, но не меньше 50 мс

}

}

else

{

// Удаление последнего сегмента, если еда не съедена

snake.RemoveAt( snake.Count - 1 );

}

Invalidate(); // Перерисовывание формы

}

// Обработчик нажатий клавиш. Движение клавишами

private void OnKeyDown( object sender, KeyEventArgs e )

{

// Изменение направления в зависимости от нажатой клавиши

switch (e.KeyCode)

{

case Keys.Right:

case Keys.D:

if (direction != 2) direction = 0; // Вправо

break;

case Keys.Up:

case Keys.W:

if (direction != 3) direction = 1; // Вверх

break;

case Keys.Left:

case Keys.A:

if (direction != 0) direction = 2; // Влево

break;

case Keys.Down:

case Keys.S:

if (direction != 1) direction = 3; // Вниз

break;

case Keys.Escape: timer.Stop(); break; // Остановка игры

case Keys.Space: timer.Start(); break; // Остановка игры

}

}

private void NewGameButton\_Click( object sender, EventArgs e )

{

// Перезапуск игры

this.Controls.Remove( newGameButton ); // Удаляем кнопку

InitializeGame(); // Инициализация новой игры

}

}

} **Описание кода**

1. **Инициализация**:
   * Змейка представлена как список сегментов, где каждый сегмент — это точка (координаты).
   * Еда генерируется в случайной позиции, которая не совпадает с сегментами змейки.
   * Звуковые эффекты при поедании еды и при столкновении.
2. **Движение змейки**:
   * Змейка движется в зависимости от текущего направления. Если змейка достигает еды, то она растет, и генерируется новая еда.
3. **Управление**:
   * Игрок может управлять змейкой с помощью клавиш стрелок и WASD.
4. **Окончание игры**:
   * Если змейка сталкивается с границей или сама с собой, игра останавливается, и выводится сообщение об окончании игры. Также меняется анимация головы.

Анимация головы:

**Запуск игры**

1. Запустите игру ярлык «Snake Game START». Вы увидите игровое поле, на котором будет змейка и еда.
2. Используйте стрелки или WASD для управления змейкой.
3. ESC – для паузы и Spice (Пробел) – продолжить.
4. Когда игра закончится (при столкновении), вы сможете начать новую игру, нажав кнопку "Новая игра".

