**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БАШКОРТОСТАН**

Государственное бюджетное профессиональное образовательноеучреждение

«Уфимский колледж статистики, информатики и вычислительной техники»

Проектная работа

Тема: Программа для Windows Forms «Игра Змейка»

Дисциплина: Прикладное программирование

Выполнил:

Студент гр. 16П-2

\_\_\_\_\_\_\_\_\_\_\_Батурин Н. С.

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2018г.

Проверил преподаватель

\_\_\_\_\_\_\_\_\_\_\_Зиганшина Р. С.

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2018г.

Уфа 2018

Цель работы: Написать игру «Змейка», используя Windows Forms и приобретенные навыки владения языком программирования C#.

Теоретический материал

Змейка - Это 2D игра целью, которой является поедание фруктов, расширяющих длину змеи до тех пор пока она не заполнит все поле.

Проигрышем считается выход за пределы поля или поедания туловища змеи.

Использованные элементы Windows Forms:

FieldPnl - поле

Tmr - таймер

menuStrip1 - меню

difficultyToolStripMenuItem - сложность

viewToolStripMenuItem - вид

showGridToolStripMenuItem - сетка

gameToolStripMenuItem - игра

startToolStripMenuItem - старт

stopToolStripMenuItem - стоп

pauseToolStripMenuItem - пауза

statusStrip1 - состояние

TSSL1 - счет

GameOverLbl – конец игры

PAKTCLbl - инфо

PEOETELbl - инфо

themeToolStripMenuItem - тема

brightToolStripMenuItem - светлая

darkToolStripMenuItem - темная

easyToolStripMenuItem - легко

mediumToolStripMenuItem - средне

hardToolStripMenuItem - тяжело

TSSL2 - размерность

Код из файла Form1.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AdvSnake

{

public partial class Form1 : Form

{

int widInt = 10, heiInt = 10, moves = 0, score = 0;

double wid = 10.0, hei = 10.0;

List<Point> SnakeBody;

Point fruit;

Random rnd = new Random();

bool gridField = false, gamedo = false, pause = false;

Way KeyWay = Way.None;

Difficulty dif = Difficulty.Easy;

Pen MyPen;

Color ClrBack, ClrFore;

Brush brushFruit = Brushes.DarkOrange;

enum Way

{

None,

Up,

Right,

Left,

Down,

}

enum Difficulty

{

Easy = 1,

Medium = 2,

Hard = 3

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (pause && e.KeyData == Keys.Escape)

pauseToolStripMenuItem.PerformClick();

else if (gamedo && !pause)

{

switch (e.KeyData)

{

case Keys.Up:

if (KeyWay != Way.Down)

KeyWay = Way.Up;

break;

case Keys.Right:

if (KeyWay != Way.Left)

KeyWay = Way.Right;

break;

case Keys.Left:

if (KeyWay != Way.Right)

KeyWay = Way.Left;

break;

case Keys.Down:

if (KeyWay != Way.Up)

KeyWay = Way.Down;

break;

case Keys.Escape:

pauseToolStripMenuItem.PerformClick();

moves--;

break;

}

moves++;

}

else if(GameOverLbl.Visible)

{

switch (e.KeyData)

{

case Keys.Escape:

if (PEOETELbl.Visible)

Close();

else

{

if (pause)

startToolStripMenuItem.PerformClick();

else

pauseToolStripMenuItem.PerformClick();

}

break;

case Keys.Enter:

if (PEOETELbl.Visible)

Close();

break;

default:

GameOverLbl.Visible = false;

PAKTCLbl.Visible = false;

PEOETELbl.Visible = false;

FieldPnl.Refresh();

break;

}

}

}

public Form1()

{

InitializeComponent();

MinimumSize = new Size(Width - FieldPnl.Width + 200, Height - FieldPnl.Height + 200);

wid = FieldPnl.Width / 10.0;

hei = FieldPnl.Height / 10.0;

widInt = (int)Math.Truncate(wid);

heiInt = (int)Math.Truncate(hei);

SnakeBody = new List<Point>() { new Point(widInt / 2, heiInt / 2), new Point(widInt / 2, heiInt / 2),

new Point(widInt / 2, heiInt / 2), new Point(widInt / 2, heiInt / 2) };

ClrBack = Color.WhiteSmoke;

ClrFore = Color.Black;

fruit = NewFruit();

MyPen = new Pen(Color.LimeGreen, 10f)

{

StartCap = LineCap.Round,

EndCap = LineCap.Triangle,

LineJoin = LineJoin.Round

};

TSSL1.Text = $"Score: {score}";

}

public Point NewFruit()

{

wid = FieldPnl.Width / 10.0;

hei = FieldPnl.Height / 10.0;

widInt = (int)Math.Truncate(wid);

heiInt = (int)Math.Truncate(hei);

Point tmp = new Point();

tmp.X = rnd.Next(0, widInt);

tmp.Y = rnd.Next(0, heiInt);

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

{

if (SnakeBody[i].X == tmp.X && SnakeBody[i].Y == tmp.Y)

{

Point p = NewFruit();

tmp = new Point(p.X, p.Y);

i = 0;

}

}

return new Point(tmp.X, tmp.Y);

}

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

{

if (gamedo)

{

if (pause)

{

pause = false;

Tmr.Start();

startToolStripMenuItem.Enabled = false;

stopToolStripMenuItem.Enabled = true;

pauseToolStripMenuItem.Text = "Pause";

}

else

{

pause = true;

Tmr.Stop();

startToolStripMenuItem.Enabled = false;

stopToolStripMenuItem.Enabled = false;

pauseToolStripMenuItem.Text = "Continue";

}

}

}

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

{

darkToolStripMenuItem.Checked = false;

brightToolStripMenuItem.Checked = true;

ClrBack = Color.WhiteSmoke;

ClrFore = Color.Black;

MyPen.Color = Color.LimeGreen;

brushFruit = Brushes.OrangeRed;

FieldPnl.Refresh();

}

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

{

darkToolStripMenuItem.Checked = true;

brightToolStripMenuItem.Checked = false;

ClrBack = Color.DimGray;

ClrFore = Color.Silver;

MyPen.Color = Color.Cyan;

brushFruit = Brushes.Gold;

FieldPnl.Refresh();

}

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

{

var b = sender as ToolStripMenuItem;

easyToolStripMenuItem.Checked = false;

mediumToolStripMenuItem.Checked = false;

hardToolStripMenuItem.Checked = false;

b.Checked = true;

dif = (Difficulty)Convert.ToInt32(b.AccessibleDescription);

if (dif == Difficulty.Easy)

Tmr.Interval = 150;

else if (dif == Difficulty.Medium)

Tmr.Interval = 100;

else if (dif == Difficulty.Hard)

Tmr.Interval = 50;

}

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

{

Tmr.Start();

gamedo = true;

pauseToolStripMenuItem.Enabled = true;

stopToolStripMenuItem.Enabled = true;

this.FormBorderStyle = FormBorderStyle.FixedSingle;

Graphics g = FieldPnl.CreateGraphics();

fruit = NewFruit();

g.FillEllipse(brushFruit, fruit.X \* 10, fruit.Y \* 10, 10, 10);

TSSL1.Text = $"Score: {score}";

if (GameOverLbl.Visible)

{

GameOverLbl.Visible = false;

PAKTCLbl.Visible = false;

PEOETELbl.Visible = false;

}

startToolStripMenuItem.Enabled = false;

difficultyToolStripMenuItem.Enabled = false;

}

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

{

if (showGridToolStripMenuItem.Checked)

{

showGridToolStripMenuItem.Checked = false;

gridField = false;

}

else

{

showGridToolStripMenuItem.Checked = true;

gridField = true;

}

FieldPnl.Refresh();

}

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

{

Tmr.Stop();

gamedo = false;

FormBorderStyle = FormBorderStyle.Sizable;

moves = 0;

score = 0;

KeyWay = Way.None;

wid = FieldPnl.Width / 10.0;

hei = FieldPnl.Height / 10.0;

widInt = (int)Math.Truncate(wid);

heiInt = (int)Math.Truncate(hei);

SnakeBody.Clear();

SnakeBody = new List<Point>() { new Point(widInt / 2, heiInt / 2), new Point(widInt / 2, heiInt / 2),

new Point(widInt / 2, heiInt / 2), new Point(widInt / 2, heiInt / 2) };

FieldPnl.Refresh();

stopToolStripMenuItem.Enabled = false;

startToolStripMenuItem.Enabled = true;

pauseToolStripMenuItem.Enabled = false;

difficultyToolStripMenuItem.Enabled = true;

}

private void Tmr\_Tick(object sender, EventArgs e)

{

if (gamedo)

{

if (!pause && KeyWay != Way.None)

{

int x = SnakeBody[0].X, y = SnakeBody[0].Y;

switch (KeyWay)

{

case Way.Up:

y--;

break;

case Way.Right:

x++;

break;

case Way.Left:

x--;

break;

case Way.Down:

y++;

break;

}

if (y < 0 || y >= heiInt || x < 0 || x >= widInt)

{

GameOver();

return;

}

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

if (SnakeBody[i].X == x && SnakeBody[i].Y == y)

{

GameOver();

return;

}

Point newHead = new Point(x, y);

SnakeBody.Insert(0, newHead);

Graphics g = FieldPnl.CreateGraphics();

if (SnakeBody[0].X == fruit.X && SnakeBody[0].Y == fruit.Y)

{

if (SnakeBody.Count == widInt \* heiInt)

{

MessageBox.Show("You won", "Congratulations!");

GameOver();

return;

}

fruit = NewFruit();

g.DrawLines(MyPen, SnakeBody.Select(p => new Point(p.X \* 10 + 5, p.Y \* 10 + 5)).ToArray());

g.FillEllipse(brushFruit, fruit.X \* 10, fruit.Y \* 10, 10, 10);

TSSL1.Text = $"Score: {++score}";

}

else

{

int lastI = SnakeBody.Count - 1;

g.FillRectangle(new Pen(ClrBack).Brush, SnakeBody[lastI].X \* 10, SnakeBody[lastI].Y \* 10, 11, 11);

g.FillRectangle(new Pen(ClrBack).Brush, SnakeBody[lastI - 1].X \* 10, SnakeBody[lastI - 1].Y \* 10, 11, 11);

if (gridField)

{

g.DrawRectangle(new Pen(ClrFore), SnakeBody[lastI].X \* 10, SnakeBody[lastI].Y \* 10, 10, 10);

g.DrawRectangle(new Pen(ClrFore), SnakeBody[lastI - 1].X \* 10, SnakeBody[lastI - 1].Y \* 10, 10, 10);

}

SnakeBody.RemoveAt(lastI);

g.DrawLines(MyPen, SnakeBody.Select(p => new Point(p.X \* 10 + 5, p.Y \* 10 + 5)).ToArray());

}

}

else

{

if (moves == 0)

FieldPnl.CreateGraphics().FillEllipse(MyPen.Brush, widInt \* 5, heiInt \* 5, 10, 10);

}

}

}

private void FieldPnl\_Paint(object sender, PaintEventArgs e)

{

wid = FieldPnl.Width / 10.0;

hei = FieldPnl.Height / 10.0;

widInt = (int)Math.Truncate(wid);

heiInt = (int)Math.Truncate(hei);

TSSL2.Text = $"{widInt}, {heiInt}";

Pen forepen = new Pen(ClrFore);

FieldPnl.BackColor = ClrBack;

if (gridField)//grid

{

for (int i = 0; i <= heiInt; i++)//vertical lines

e.Graphics.DrawLine(forepen, new Point(0, i \* 10), new Point(FieldPnl.Width, i \* 10));

for (int i = 0; i <= widInt; i++)//horizontal lines

e.Graphics.DrawLine(forepen, new Point(i \* 10, 0), new Point(i \* 10, FieldPnl.Height));

}

e.Graphics.FillPolygon(forepen.Brush, new[] {//void zone

new Point(widInt \* 10, 0),

new Point(FieldPnl.Width, 0),

new Point(FieldPnl.Width, FieldPnl.Height),

new Point(0, FieldPnl.Height),

new Point(0, heiInt \* 10),

new Point(widInt \* 10, heiInt \* 10)

});

e.Graphics.DrawLines(MyPen, SnakeBody.Select(p => new Point(p.X \* 10 + 5, p.Y \* 10 + 5)).ToArray());//snake

if (!pause && gamedo && !GameOverLbl.Visible && moves == 0)//stand still

e.Graphics.FillEllipse(MyPen.Brush, widInt \* 5, heiInt \* 5, 10, 10);

if (gamedo)//fruit

e.Graphics.FillEllipse(brushFruit, fruit.X \* 10, fruit.Y \* 10, 10, 10);

}

private void FieldPnl\_SizeChanged(object sender, EventArgs e)

{

wid = FieldPnl.Width / 10.0;

hei = FieldPnl.Height / 10.0;

widInt = (int)Math.Truncate(wid);

heiInt = (int)Math.Truncate(hei);

SnakeBody = new List<Point>() { new Point(widInt / 2, heiInt / 2), new Point(widInt / 2, heiInt / 2),

new Point(widInt / 2, heiInt / 2), new Point(widInt / 2, heiInt / 2) };

FieldPnl.Refresh();

}

private void GameOver()

{

stopToolStripMenuItem.PerformClick();

FieldPnl.CreateGraphics().Clear(ClrBack);

GameOverLbl.Visible = true;

PAKTCLbl.Visible = true;

PEOETELbl.Visible = true;

}

}

}

Результаты:

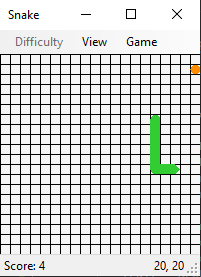


Рис. 1. Вид программы при: Светлой теме и включенным отображение сетки.

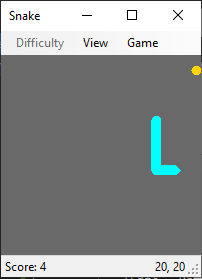


Рис. 2. Вид программы при: Темной теме и выключенным отображение сетки.

Вывод: Я написал игру «Змейка», используя Windows Forms и приобретенные навыки владения языком программирования C#.