**Графика (C#)**

Требования к проекту:

* Проект WindowsForm.
* Необходимо описать правила игры: в прилагаемом документе Word и/или сделать в приложении кнопку «?» и подгружать на отдельной форме правила игры..
* В обязательном порядке создать класс с игровыми методами.

**Задание**

1. Придумать свою игру-кликер или игру-бродилку. В обязательном порядке имеется игровое поле *n x n* с разного рода полями (минимум 2-3 типа полей), каждое из которых отличается от других цветом и числовым знамением в расчетном (игровом) массиве. Для кликера: придумать правила, по которым за время (или за кол-во шагов) нужно перевести через клик кнопки из одних состояний в другие состояния. Для бродилки: за время или кол-во шагов перемещать игрока по полю, игрок ходит по «белым» полям и может собирать только «зеленые» клетки (очки/жизни/золото и т.д.), на «красных» он гибнет, в «черные» ходить не может (или может при наборе очков их разбивать).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Игровой массив int | | | | | Соответствие числовому (игровому) массиву и визуализации на форме | Визуализация (button) | | | | |
| 0 | 0 | 0 | 3 | 2 |  |  |  |  |  |
| 1 | 0 | 0 | 0 | 0 |  |  |  |  |  |
| 2 | 0 | 1 | 0 | 2 |  |  |  |  |  |
| 0 | 1 | 0 | 0 | 0 |  |  |  |  |  |
| 2 | 0 | 2 | 1 | 0 |  |  |  |  |  |

1. Необходимо написать класс Game с заданными полями (размер игрового поля, массивы) и методами (конструкторами по умолчанию и с передачей размеров поля, созданием игрового поля, заполнением игрового поля значения, визуализацией (визуализация всегда независима!), игровыми методами).

**Подсказка**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Buttons\_Array\_class

{

class MyButtons

{

Button[,] bt; //игровое поле (визуальная часть)

int[,] bt\_numb; //игровое поле (расчетная часть)

int n = 3;

Random rnd = new Random();

public MyButtons(int a)

{

n = a;

bt = new Button[n, n];

bt\_numb = new int[n, n];

}

public void Bt\_visual(Button bt\_example, Form frm)

{

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

bt[i, j] = new Button();

bt[i, j].Height = bt\_example.Height;

bt[i, j].Width = bt\_example.Width;

bt[i, j].Left = bt\_example.Left + j \* bt\_example.Width;

bt[i, j].Top = bt\_example.Top + i \* bt\_example.Height;

bt[i, j].Parent = frm;

bt[i, j].Name = i.ToString() + "\_" + j.ToString();

bt\_numb[i, j] = rnd.Next(1,3);

bt[i, j].Text = bt\_numb[i, j].ToString();

bt[i, j].Click += new EventHandler(bt\_click); // Объявление «клика» для массива создаваемых кнопок

//bt[i, j].MouseDown += new MouseEventHandler(bt\_MDown);

}

}

bt\_example.Hide();

//EO\_buttons();

ButtonsColor();

}

private void EO\_buttons()

{

bt\_numb[0, 0] = 0;

bt[0, 0].Enabled = false;

bt\_numb[n-1, n-1] = 3;

bt[n - 1, n - 1].Enabled = false;

}

private void ButtonsColor()

{

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

if (bt\_numb[i, j] == 2)

bt[i, j].BackColor = Color.Red;

if (bt\_numb[i, j] == 1)

bt[i, j].BackColor = Color.LightGreen;

if (bt\_numb[i, j] == 0)

bt[i, j].BackColor = Color.Aquamarine;

if (bt\_numb[i, j] == 3)

bt[i, j].BackColor = Color.Black;

if (bt\_numb[i, j] == 5)

bt[i, j].BackColor = Color.DarkOrange;

}

}

}

//Программирование «клика» для массива создаваемых кнопок

void bt\_click(object sender, EventArgs e)

{

var current = sender as Button;

//current.BackColor = Color.Gray;

if (current.BackColor == Color.Green)

{

current.BackColor = Color.LightGreen;

current.Text = "1";

}

if (current.BackColor == Color.Red)

{

current.BackColor = Color.Green;

current.Text = "2 --> 1";

}

}

}

}