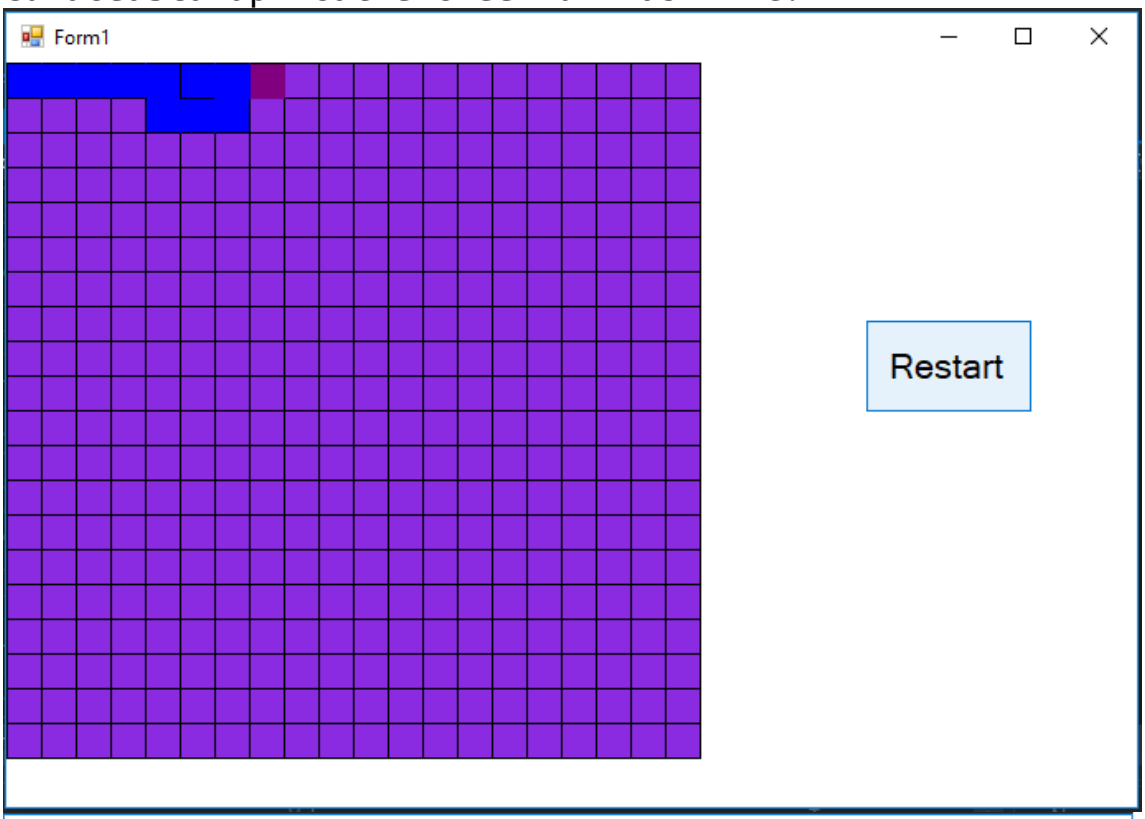
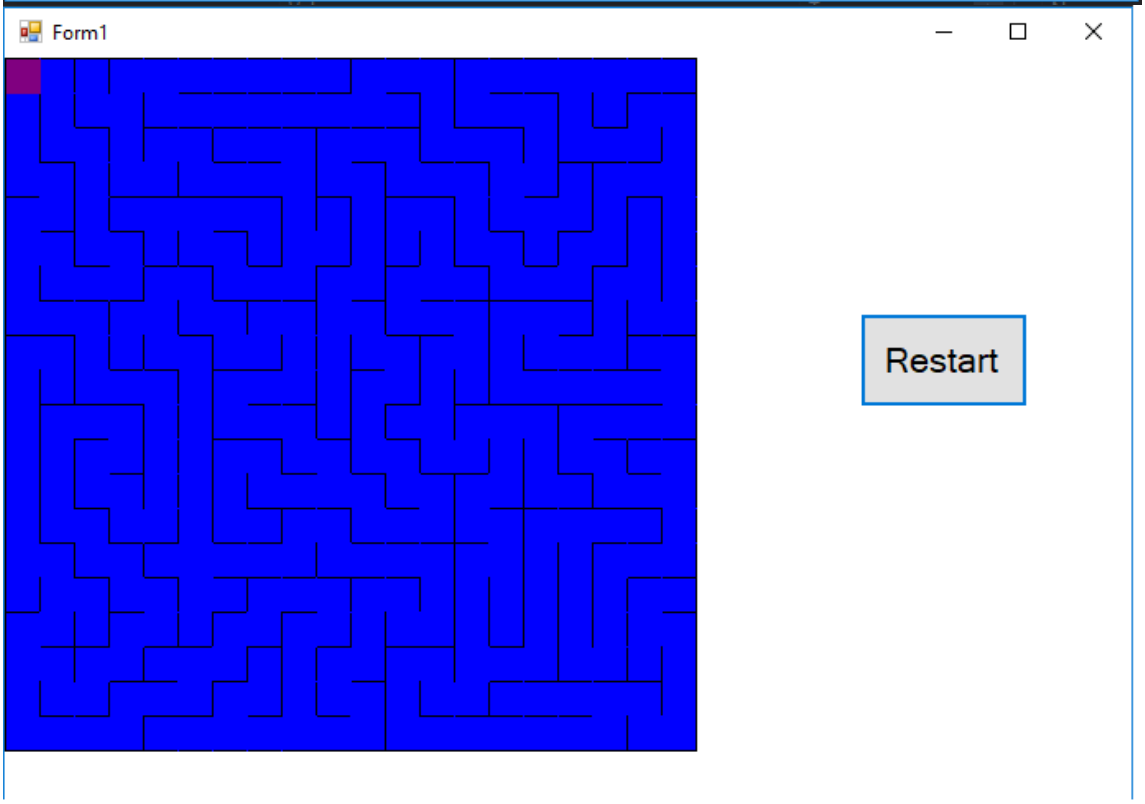
Maze Generator

Cu aceasta aplicatie creez un labirint.





```
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 WindowsFormsApp2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        Cell current,next;
        Cell[,] cells;
        Stack<Cell> drum = new Stack<Cell> { };
        Pen creion = new Pen(Color.Red);
        Graphics desen;
        int n, m, w;
        Random rnd = new Random();

        public void timer_Tick(object sender, EventArgs e)
        {
            current.current = true;
            current.visited = true;
            for(int k=0;k<n;k++)
                for(int p=0;p<m;p++)
                     cells[k, p].display(desen, w, n, m);
            current.verVecini(cells);
            int rand = rnd.Next(0,current.k);

            if (current.k > 0)
            {
                current.changed = true;
                next = current.vecini[rand];
                int dir = -1;

                if (current.i > next.i) dir = 3;
                else if (current.i < next.i) dir = 1;
                else if (current.j > next.j) dir = 0;
                else if (current.j < next.j) dir = 2;

                current.walls[dir] = false;
                if (dir >= 2)
                    next.walls[dir - 2] = false;
                else
                    next.walls[dir + 2] = false;

                drum.Push(current);
            }
            else if(drum.Count()>0)
```

```
            {
                next = drum.Pop();
                current.changed = true;
            }
            current.current = false;
            current = next;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            button1.Text = "Restart";
            n = 20;
            m = 20;
            w = (this.Height-50) / m;
            cells = new Cell[n + 1, m + 1];
            desen = this.CreateGraphics();
            Timer t = new Timer();
            t.Interval =100;
            t.Tick +=new EventHandler(timer_Tick);
            t.Start();

            for (int i = 0; i < n; i++)
                for (int j = 0; j < m; j++)
                    cells[i, j] = new Cell(i, j);

            desen.FillRectangle(new SolidBrush(Color.BlueViolet), 0, 0, n * w, m * w);

            current = cells[0, 0];
        }
    }
    public class Cell
    {
        public int i, j, k, size;
        public bool visited, current,changed;
        public bool[] walls;
        Pen creion = new Pen(Color.Black);
        SolidBrush pensula = new SolidBrush(Color.Blue);
        public Cell[] vecini;

        public Cell(int i, int j)
        {
            this.i = i;
            this.j = j;
            this.visited = false;
            this.changed = true;
            this.walls = new bool[4];
            for (int k = 0; k < 4; k++)
                this.walls[k] = true;
            vecini = new Cell[4];
        }
        public void display(Graphics Desen, int w, int n, int m)
        {
            this.size = n;

            if (this.changed == true) {
                if (this.walls[0])
                    Desen.DrawLine(creion, i * w, j * w, i * w + w, j * w);
                else
```

```
                    Desen.DrawLine(new Pen(Color.Blue), i * w, j * w, i * w + w, j * w);
                if (this.walls[1])
                    Desen.DrawLine(creion, i * w + w, j * w, i * w + w, j * w + w);
                else
                    Desen.DrawLine(new Pen(Color.Blue), i * w + w, j * w, i * w + w, j *
w + w);

                if (this.walls[2])
                    Desen.DrawLine(creion, i * w + w, j * w + w, i * w, j * w + w);
                else
                    Desen.DrawLine(new Pen(Color.Blue), i * w + w, j * w + w, i * w, j *
w + w);

                if (this.walls[3])
                    Desen.DrawLine(creion, i * w, j * w + w, i * w, j * w);
                else
                    Desen.DrawLine(new Pen(Color.Blue), i * w, j * w + w, i * w, j * w);
                if (visited) Desen.FillRectangle(pensula, i * w + 1, j * w + 1, w-1, w-
1);
                this.changed = false;
            }
            if (current) Desen.FillRectangle(new SolidBrush(Color.Purple), i * w + 1, j *
w + 1, w, w);
        }
        public void verVecini(Cell[,] cells)
        {
            k = 0;
            Cell top, bottom, left, right;

            if(this.i+1 <= this.size)
            {
                top = cells[this.i + 1, this.j];
                if (top != null)
                    if (!top.visited)
                        vecini[k++] = top;
            }

            if (this.i-1 >= 0)
            {
                bottom = cells[this.i - 1, this.j];
                if (bottom != null)
                    if (!bottom.visited)
                     vecini[k++] = bottom;
            }

            if (this.j+1 <= this.size)
            {
                right = cells[this.i, this.j + 1];
                if(right != null)
                    if (!right.visited)
                        vecini[k++] = right;
            }

            if (this.j - 1 >= 0)
            {
                left = cells[this.i, this.j - 1];
                if (left != null)
                    if (!left.visited)
```

```
                        vecini[k++] = left;
                }
            }

        }
    }
```