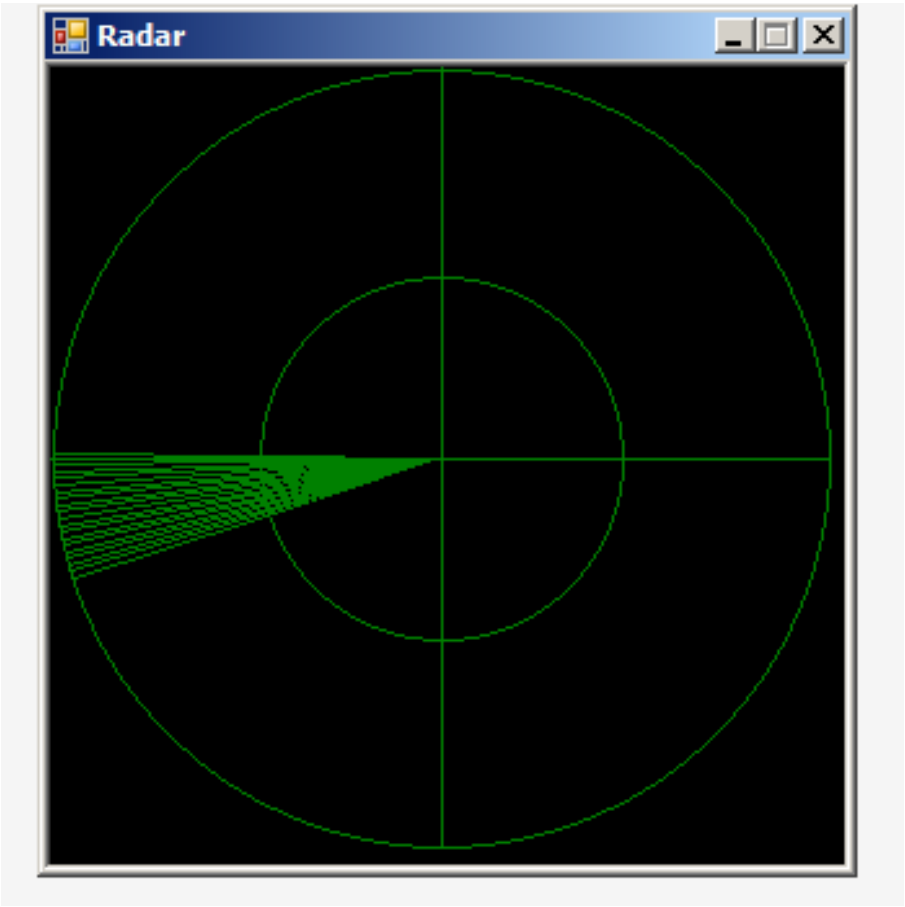
Radar

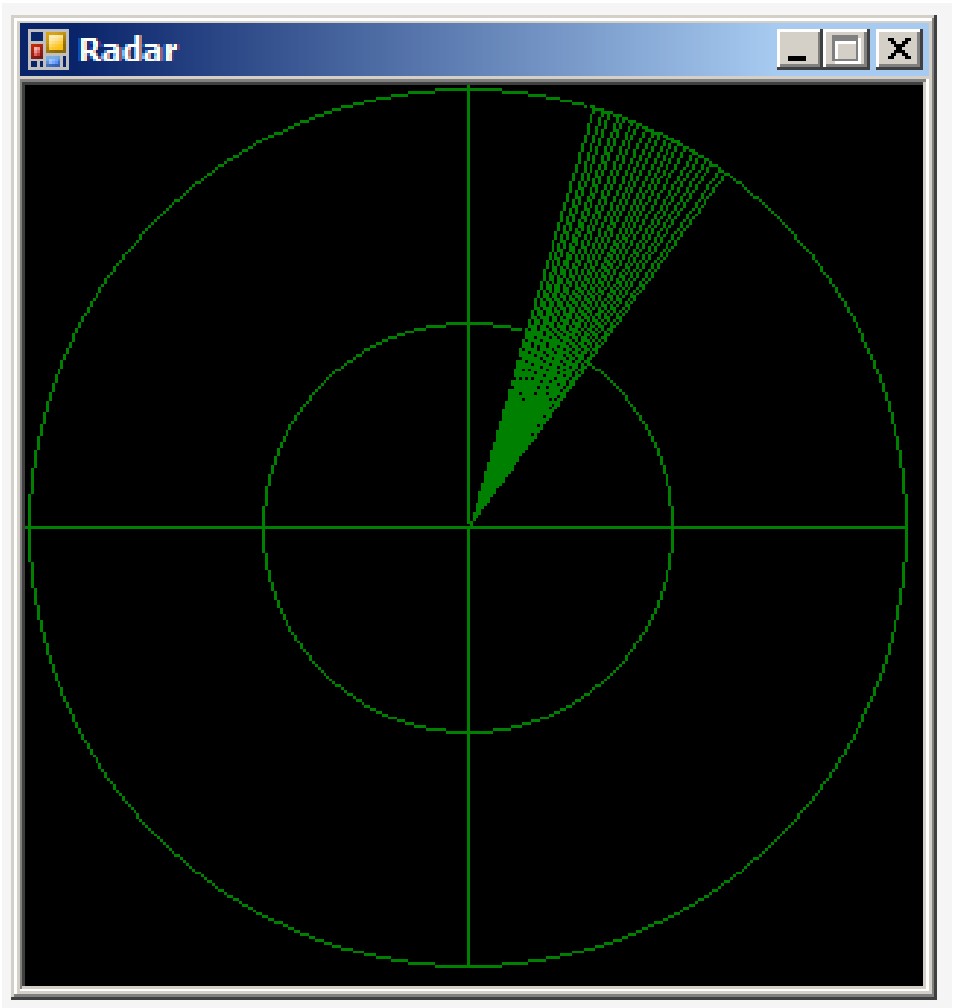
Lucrarea a 5-a

Marton Elias-Natanael

1.Prezentarea aplicatiei:

Aceasta aplicatie a fost realizata in Visual Studio 2017, folosind limbajul de programare C#. Dupa cum ii spune si numele, aplicatia este o copie virtuala a unui aparat radar (regasit pe nave maritime sau avioane).





2.Prezentarea codului:

Aplicatia deseneaza pe formul negru din fundal doua cercuri prin centrul carora trec doua axe care impart cercul in patru cadrane (asemanator cercului trigonometric). Un fascicul de linii cu originea in centrul axelor si extremitatea pe cercul mare, de o anumita deschidere, se invarte in jurul centrului axelor, controlat fiind de un timer cu interval relativ mic (5 milisec).

```
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 app5
{
    public partial class Form1 : Form
    {

        int w = 300, h = 300, ax = 150;
        int u; // in grade
        int cx, cy; // centrul cercului
        int x, y; // coordonate ax

        int tx, ty, lim = 20;

        Pen p = new Pen(Color.Green);
        Pen n = new Pen(Color.Black);
        Graphics desen;

        private void deseneaza_ax()
        {
            int tu = (u - lim) % 360;

            if (u >= 0 && u <= 180)
            {
                x = cx + (int)(ax * Math.Sin(Math.PI * u / 180));
                y = cy - (int)(ax * Math.Cos(Math.PI * u / 180));
            }
            else
            {
                x = cx - (int)(ax * -Math.Sin(Math.PI * u / 180));
                y = cy - (int)(ax * Math.Cos(Math.PI * u / 180));
            }

            if (tu >= 0 && tu <= 180)
            {
                tx = cx + (int)(ax * Math.Sin(Math.PI * tu / 180));
                ty = cy - (int)(ax * Math.Cos(Math.PI * tu / 180));
            }
            else
            {
                tx = cx - (int)(ax * -Math.Sin(Math.PI * tu / 180));
                ty = cy - (int)(ax * Math.Cos(Math.PI * tu / 180));
            }

            desen.DrawEllipse(p, 1, 1, w, h);
            desen.DrawEllipse(p, 81, 81, w - 160, h - 160);

            desen.DrawLine(p, cx, 0, cx, h);
            desen.DrawLine(p, 0, cy, w, cy);
```

```
            desen.DrawLine(n, cx, cy, tx, ty);
            desen.DrawLine(p, cx, cy, x, y);

            u++;
            if (u == 360)
                u = 0;
        }

        private void t_Tick(object sender, EventArgs e)
        {
            deseneaza_ax();
        }



        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            this.BackColor = Color.Black;
            desen = this.CreateGraphics();

            cx = w / 2 +1;
            cy = h / 2 +1;

            u = 0;

        }
    }
}
```