

//Lab Exercise 5.24.2022

//Draw House

//Author: nmessa

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

            private void radioButtonClicked(object sender)
            {
                //Create a Graphics surface on lblDraw
                Graphics labelSurface = lblDraw.CreateGraphics();

                //Store the width and height of Graphics surface
                int maxX = lblDraw.Size.Width;
                int maxY = lblDraw.Size.Height;

                //Create 8 brushes
                SolidBrush greenBrush = new SolidBrush(Color.Olive);
                SolidBrush blackBrush = new SolidBrush(Color.Black);
                SolidBrush brownBrush = new SolidBrush(Color.Brown);
                SolidBrush magentaBrush = new SolidBrush(Color.SlateGray);
                SolidBrush purpleBrush = new SolidBrush(Color.Purple);
                SolidBrush redBrush = new SolidBrush(Color.Red);
                SolidBrush whiteBrush = new SolidBrush(Color.White);
                SolidBrush yellowBrush = new SolidBrush(Color.Yellow);

                //Define the points array for the roof
                Point[] roofPoints = new Point[] { new Point(maxX / 3 - 30, maxY / 2 + 30),
                                                    new Point(maxX - 62, maxY / 2 + 30),
                                                    new Point(maxX / 2 + 10, maxY / 4)};

                //Convert the sender to a RadioButton
                RadioButton button = (RadioButton)sender;
```

```

//Determine which RadioButton was selected using the Tag property and act accordingly
switch (Convert.ToInt32(button.Tag))
{
    case 1: //set label BackColor to DeepSkyBlue
        lblDraw.BackColor = Color.DeepSkyBlue;
        break;
    case 2: //set label BackColor to LightSkyBlue
        lblDraw.BackColor = Color.LightSkyBlue;
        break;
    case 3: //Draw a red FillEllipse for sun
        labelSurface.FillEllipse(redBrush, maxX - 100, maxY - 370, 80, 80);
        break;
    case 4: //Draw a yellow FillEllipse for sun
        labelSurface.FillEllipse(yellowBrush, maxX - 100, maxY - 370, 80, 80);
        break;
    case 5: //Draw a brown FillPolygon using roofPoints
        labelSurface.FillPolygon(brownBrush, roofPoints);
        break;
    case 6: //Draw a purple FillPolygon using roofPoints
        labelSurface.FillPolygon(purpleBrush, roofPoints);
        break;
    case 7: //Draw a green FillRectangle for house body
        labelSurface.FillRectangle(greenBrush, maxX / 3 - 30, maxY / 2 + 30, maxX - 120, maxY - 10);
        break;
    case 8: //Draw a magenta FillRectangle for house body
        labelSurface.FillRectangle(magentaBrush, maxX / 3 - 30, maxY / 2 + 30, maxX - 120, maxY -
10);
        break;
    case 9: //Draw a black FillRectangle for door
        labelSurface.FillRectangle(blackBrush, maxX / 3 + 30, maxY / 2 + 100,
            maxX - 230, maxY - 270);
        break;
    case 10: //Draw a white FillRectangle for door
        labelSurface.FillRectangle(whiteBrush, maxX / 3 + 30, maxY / 2 + 100,
            maxX - 230, maxY - 270);
        break;
}
}

private void radBlueSky_CheckedChanged(object sender, EventArgs e)
{
    radioButtonClicked(sender);
}

```

```
private void radCyanSky_CheckedChanged(object sender, EventArgs e)
{
    radioButtonClicked(sender);
}

private void radRedSun_CheckedChanged(object sender, EventArgs e)
{
    radioButtonClicked(sender);
}

private void radYellowSun_CheckedChanged(object sender, EventArgs e)
{
    radioButtonClicked(sender);
}

private void radBrownRoof_CheckedChanged(object sender, EventArgs e)
{
    radioButtonClicked(sender);
}

private void radPurpleRoof_CheckedChanged(object sender, EventArgs e)
{
    radioButtonClicked(sender);
}

private void radGreenHouse_CheckedChanged(object sender, EventArgs e)
{
    radioButtonClicked(sender);
}

private void radMagentaHouse_CheckedChanged(object sender, EventArgs e)
{
    radioButtonClicked(sender);
}

private void radBlackDoor_CheckedChanged(object sender, EventArgs e)
{
    radioButtonClicked(sender);
}

private void radWhiteDoor_CheckedChanged(object sender, EventArgs e)
{
    radioButtonClicked(sender);
}
}
```

//Lab Exercise 5.24.2022

//Face Generator

//Author: nmessa

```
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 Face
{
    public partial class Form1 : Form
    {
        bool blnSwitch = true;

        public Form1()
        {
            InitializeComponent();

            private void btnHappy_Click(object sender, EventArgs e)
            {
                tmrWink.Enabled = false;
                btnWink.Enabled = true;

                //Create Graphics surface on PictureBox
                Graphics pictureBoxSurface = picFrame.CreateGraphics();
                pictureBoxSurface.Clear(this.BackColor);

                //Get width and height of graphics surface
                int maxX = picFrame.Size.Width;
                int maxY = picFrame.Size.Height;

                //draw rectangle in background of picturebox
                SolidBrush tanBrush = new SolidBrush(Color.Tan);
                pictureBoxSurface.FillRectangle(tanBrush, 0, 0, maxX, maxY);

                //draw large ellipse for face
                SolidBrush grayBrush = new SolidBrush(Color.LightGray);
                pictureBoxSurface.FillEllipse(grayBrush, 1, 1, maxX - 3, maxY - 3);
```

```

//draw eyes
SolidBrush blueBrush = new SolidBrush(Color.Blue);
pictureBoxSurface.FillEllipse(blueBrush, maxX / 3 - 10, 30, 35, 35);
pictureBoxSurface.FillEllipse(blueBrush, maxX / 2 + 10, 30, 35, 35);

//draw nose
SolidBrush redBrush = new SolidBrush(Color.Red);
pictureBoxSurface.FillEllipse(redBrush, maxX / 2 - 10, maxY / 2, 20, 20);

//draw mouth
Pen pinkPen = new Pen(Color.PaleVioletRed,3);
pictureBoxSurface.DrawArc(pinkPen, maxX / 3 + 10, maxY / 3 + 40, 50, 50, 0, 180);
}

private void btnSad_Click(object sender, EventArgs e)
{
    tmrWink.Enabled = false;
    btnWink.Enabled = true;

    //Create Graphics surface on PictureBox
    Graphics pictureBoxSurface = picFrame.CreateGraphics();
    pictureBoxSurface.Clear(this.BackColor);

    //Get width and height of graphics surface
    int maxX = picFrame.Size.Width;
    int maxY = picFrame.Size.Height;

    //draw rectangle in background of picturebox
    SolidBrush tanBrush = new SolidBrush(Color.Tan);
    pictureBoxSurface.FillRectangle(tanBrush, 0, 0, maxX, maxY);

    //draw large ellipse for face
    SolidBrush grayBrush = new SolidBrush(Color.LightGray);
    pictureBoxSurface.FillEllipse(grayBrush, 1, 1, maxX - 3, maxY - 3);

    //draw eyes
    SolidBrush blueBrush = new SolidBrush(Color.Blue);
    pictureBoxSurface.FillEllipse(blueBrush, maxX / 3 - 10, 30, 35, 35);
    pictureBoxSurface.FillEllipse(blueBrush, maxX / 2 + 10, 30, 35, 35);

    //draw nose
    SolidBrush redBrush = new SolidBrush(Color.Red);
    pictureBoxSurface.FillEllipse(redBrush, maxX / 2 - 10, maxY / 2, 20, 20);

    //draw mouth
    Pen pinkPen = new Pen(Color.PaleVioletRed,3);
    pictureBoxSurface.DrawArc(pinkPen, maxX / 3 + 10, maxY / 3 + 60, 50, 50, 0, -180);
}

```

```

private void btnWink_Click(object sender, EventArgs e)
{
    tmrWink.Enabled = true;
}

private void tmrWink_Tick(object sender, EventArgs e)
{
    //create Graphics surface on PictureBox
    Graphics pictureBoxSurface = picFrame.CreateGraphics();

    //get width and height of PictureBox
    int maxX = picFrame.Size.Width;
    int maxY = picFrame.Size.Height;

    if (blnSwitch) //erase eye
    {
        SolidBrush eraseBrush = new SolidBrush(Color.LightGray);
        pictureBoxSurface.FillEllipse(eraseBrush, maxX / 3 - 10, 30, 35, 35);
        //draw wink
        Pen bluePen = new Pen(Color.Blue, 3);
        pictureBoxSurface.DrawArc(bluePen, maxX / 3 - 10, 30, 35, 35, 0, 180);
        blnSwitch = false;
    }
    else
    {
        //erase wink
        Pen grayPen = new Pen(Color.LightGray, 3);
        pictureBoxSurface.DrawArc(grayPen, maxX / 3 - 10, 30, 35, 35, 0, 180);
        //draw open eye
        SolidBrush blueBrush = new SolidBrush(Color.Blue);
        pictureBoxSurface.FillEllipse(blueBrush, maxX / 3 - 10, 30, 35, 35);
        blnSwitch = true;
    }
}
}
}

```

//Lab Exercise 5.24.2022

//Flying Bird

//Author: nmessa

```
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 Flying_Bird
{
    public partial class Form1 : Form
    {
        //Global variables
        const int MAXIMAGES = 3;

        //Create an array that hold image file names
        string[] imageArray = new string[MAXIMAGES];
        int numImage = 0;

        public Form1()
        {
            InitializeComponent();

            //Slow down animation
            private void btnSlow_Click(object sender, EventArgs e)
            {
                tmrChangeImage.Interval += 100;
            }

            //Speed up animatino
            private void btnFast_Click(object sender, EventArgs e)
            {
                //ensure no negative interval
                if(tmrChangeImage.Interval > 100)
                    tmrChangeImage.Interval -= 100;
            }
        }
    }
}
```

```
//Generate next frame of animation
private void tmrChangeImage_Tick(object sender, EventArgs e)
{

    //Put image into picture box
    picImage.Image = Image.FromFile(imageArray[numImage]);

    //create rollover or advance image
    if (numImage == MAXIMAGES-1)
        numImage = 0;
    else
        numImage++;
}

private void Form1_Load(object sender, EventArgs e)
{
    //Initialize imageArray strings
    imageArray[0] = "bird1.gif";
    imageArray[1] = "bird2.gif";
    imageArray[2] = "bird3.gif";
}
}
```


//Lab Exercise 5.24.2022

//Photo Album

//Author: nmessa

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Photo_Album
{
    public partial class Form1 : Form
    {
        //Global variables
        //Create array of image titles
        string[] arrayWords = new string[] { "Anemone", "Gray Angel", "Sponges",
            "Scorpionfish", "Starfish" };

        //Create an array of image filenames
        string[] myImages = new string[] { "anemone.jpg", "grayangel.jpg", "sponges.jpg",
            "scorpionfish.jpg", "starfish.jpg" };

        int count = -1;

        public Form1()
        {
            InitializeComponent();
        }

        private void btnArrow_Click(object sender, EventArgs e)
        {
            //Advance to next image with rollover
            if (count == 4)
                count = 0;
            else
                count++;

            //Place image name in lblName
            string textToDisplay = arrayWords[count];
            lblName.Text = textToDisplay;

            //Place image in picPhoto
            picPhoto.Image = Image.FromFile(myImages[count]);
        }
    }
}
```

//Lab Exercise 5.24.2022

//Pie Chart Generator

//Author: nmessa

```
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 Pie_Chart
```

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

```
private void btnChart_Click(object sender, EventArgs e)
{
```

```
    //obtain the four values
```

```
    double val1 = Convert.ToDouble(txtValue1.Text);
```

```
    double val2 = Convert.ToDouble(txtValue2.Text);
```

```
    double val3 = Convert.ToDouble(txtValue3.Text);
```

```
    double val4 = Convert.ToDouble(txtValue4.Text);
```

```
    //define graphics surface and max x and y
```

```
    Graphics labelSurface = lblDrawing.CreateGraphics();
```

```
    //Get height and width of graphics surface
```

```
    int maxX = lblDrawing.Size.Width;
```

```
    int maxY = lblDrawing.Size.Height;
```

```
    //calculate the percent of each value (add total and divide each value)
```

```
    double total = val1 + val2 + val3 + val4;
```

```
    double pct1 = val1 / total * 100;
```

```
    double pct2 = val2 / total * 100;
```

```
    double pct3 = val3 / total * 100;
```

```
    double pct4 = val4 / total * 100;
```

```

//display the percentages of each value in a label
this.lblPercentages.Text = pct1.ToString("f1") + "%" + "    " +
pct2.ToString("f1") + "%" + "    " + pct3.ToString("f1") + "%" +
"    " + pct4.ToString("f1") + "%";

//define drawing brushes
SolidBrush yellowBrush = new SolidBrush(Color.YellowGreen);
SolidBrush purpleBrush = new SolidBrush(Color.BlueViolet);
SolidBrush aquaBrush = new SolidBrush(Color.Aqua);
SolidBrush magentaBrush = new SolidBrush(Color.Magenta);

//calculate the degrees to sweep each angle out of 360 degrees
const int DEGREES = 360;
int degrees1 = (int)(DEGREES * pct1 / 100);
int degrees2 = (int)(DEGREES * pct2 / 100);
int degrees3 = (int)(DEGREES * pct3 / 100);
int degrees4 = (int)(DEGREES * pct4 / 100);

//draw pie slices
labelSurface.Clear(this.BackColor);
labelSurface.FillPie(yellowBrush, 0, 0, maxX, maxY, 0, degrees1);
labelSurface.FillPie(purpleBrush, 0, 0, maxX, maxY, degrees1, degrees2);
labelSurface.FillPie(aquaBrush, 0, 0, maxX, maxY, degrees2 + degrees1, degrees3);
labelSurface.FillPie(magentaBrush, 0, 0, maxX, maxY, degrees1 + degrees2 + degrees3, degrees4);
    }
}
}

```

//Lab Exercise 5.24.2022

//Running Turtle

//Author: nmessa

```
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 Running_Turtle
{
    public partial class Form1 : Form
    {
        const int MAXIMAGES = 3;
        //Create an array of image file names
        string[] imageArray = new string[MAXIMAGES];
        int numImage = 0;

        public Form1()
        {
            InitializeComponent();

            private void btnSlow_Click(object sender, EventArgs e)
            {
                tmrChangeImage.Interval = 500;
            }

            private void btnMedium_Click(object sender, EventArgs e)
            {
                tmrChangeImage.Interval = 300;
            }

            private void btnFast_Click(object sender, EventArgs e)
            {
                tmrChangeImage.Interval = 100;
            }
        }
    }
}
```

```
private void tmrChangeImage_Tick(object sender, EventArgs e)
{
    //Put image into PictureBox
    picImage.Image = Image.FromFile(imageArray[numImage]);

    //Advance to next frame with rollover
    if (numImage == MAXIMAGES - 1)
        numImage = 0;
    else
        numImage++;
}

private void Form1_Load(object sender, EventArgs e)
{
    //Initialize array with image file names
    imageArray[0] = "turtle1.bmp";
    imageArray[1] = "turtle2.bmp";
    imageArray[2] = "turtle3.bmp";
}
}
```

//Lab Exercise 5.24.2022

//Skate!!!

//Author: nmessa

```
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 Skate

```
{  
    public partial class Form1 : Form  
    {  
        const int MAXIMAGES = 6;  
  
        //Create an array of image file names  
        string[] imageArray = new string[MAXIMAGES];  
        int intImage = 0;  
  
        public Form1()  
        {  
            InitializeComponent();  
        }  
  
        private void tmrChangeImage_Tick(object sender, EventArgs e)  
        {  
            //display an image from array  
            picImage.Image = Image.FromFile(imageArray[intImage]);  
  
            //determine next image number  
            intImage = (intImage + 1) % MAXIMAGES;  
        }  
  
        private void Form1_Load(object sender, EventArgs e)  
        {  
            //store image file names in array  
            imageArray[0] = "skateboard1.bmp";  
            imageArray[1] = "skateboard2.bmp";  
            imageArray[2] = "skateboard3.bmp";  
            imageArray[3] = "skateboard4.bmp";  
            imageArray[4] = "skateboard5.bmp";  
            imageArray[5] = "skateboard6.bmp";  
        }  
    }  
}
```