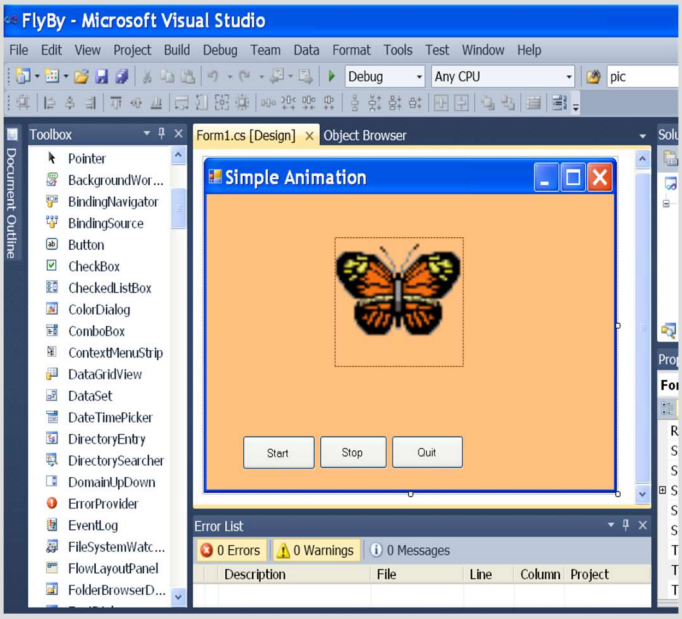
Windows Programming

bucks new university

.NET Windows Programming Using C#



Application Programming (CO453)

Part C – Weeks 9-13

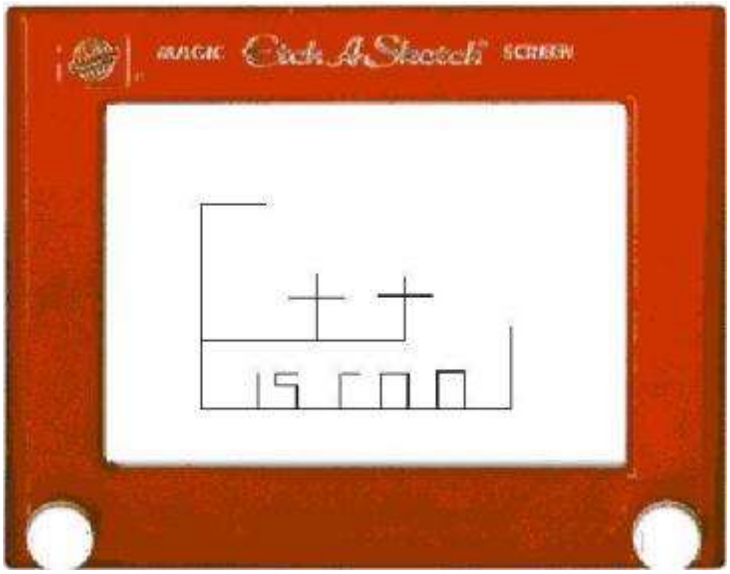
C# Windows Programming
Unit 7

Project

7: ETCH-A-SKETCH : The Project

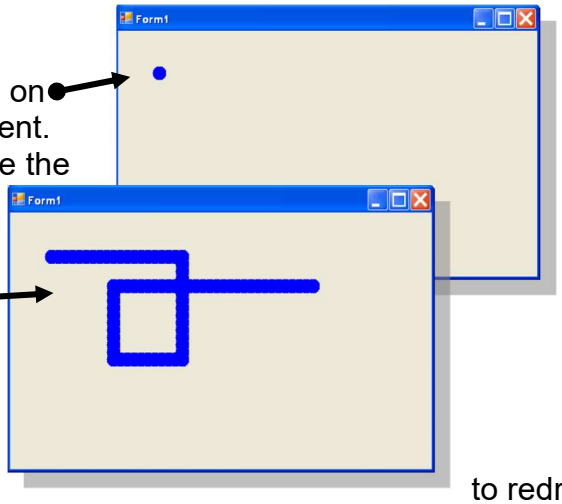
Now use your C# .NET Windows knowledge to create a prototype Etch-a-Sketch program. It will differ from the one shown here in that:

- The whole form will be used for the display
- Arrow keys will be used to draw on the screen
- Various other keys will be used to implement other functions



Starting

- Open the Windows Application called **EtchaSketch**
- Run the program and see a circle drawn on the form. This does nothing at the moment.
- What we want to do is to be able to move the circle around .. but to leave a trace behind showing where the circle has been so that we can then draw lines in the EtchaSketch way



Bitmaps

- The form **Paint()** method can be used to redraw the circle in a new position .. like the bouncing balls we did earlier .. but how can we remember the previous circle position?
- One answer is to use a **Bitmap** object which acts like a snapshot of the form .. then we keep adding to this bitmap as we move the circle.
- Look at the code in the **Form1_Load()** method (see below). This code runs when the form is first loaded and sets up a bitmap with the circle in it and makes this the background image for the form.

```
private void Form1_Load(object sender, EventArgs e)
{
        bm = new Bitmap(this.Width, this.Height);  // create a form-size bitmap
        Graphics g = Graphics.FromImage(bm);       // get a graphic object for the bitmap
        g.FillEllipse(Brushes.Blue, x, y, 20, 20); // put a circle in the bitmap
        this.BackgroundImage = bm;                 // use bitmap as the form background
}
```

Basic Project

1. Change the **Text** property of the form to something suitable
2. Maximize the form's **WindowState** property
3. Now display the **Form1_Paint()** method .. to do this go to the **Properties** window .. select the **Events** lightning button .. double-click the **Paint** event.
4. In the **Form1_Paint()** method put some code to do **only these 2 things**:
   - get a graphic object for the Bitmap
   - put a circle in the Bitmap

   (see **Form1_Load()**)
5. Copy the **ProcessCmdKey()** method from a previous exercise and modify it to use **Up**, **Down**, **Left** and **Right** arrow keys to change the values of **x** and **y**
   - be sure to use **Refresh();** after each change .. to redraw the screen
6. Add code so that the **C** key clears the screen (see the Appendix or the Graphics section for help with this). Again be sure to also use **Refresh();**
7. Add code so the **Escape** key brings up a **MessageBox** that asks the user if they are sure they want to Finish and only if Yes is clicked the program ends

Extension Work-1

1. Add code so that hitting the **B** key keeps increasing the size of the drawing circle .. but stops at some maximum value.
2. Add code so that hitting the **S** key keeps decreasing the size of the drawing circle .. but stops at some minimum value
3. Pressing the **F1** key changes the **colour** of the circle used for the drawing. Repeated pressing of F1 produces other colours.

Extension Work-2

1. Set up 4 more keys that will draw in the 4 diagonal directions (up-left, up-right, down-left, down-right)
2. Pressing the **F2** key changes the colour of the window background. Repeated pressing produces other background colours (see p64)

Extension Work-3

1. Add a red border around the form, text and two white circles to imitate the appearance of the Etch-a-Sketch toy (see the picture on the previous page).
2. When the drawing circle reaches the edge of the border (left, right, top or bottom) it should reappear at the other side
3. The program starts with a suitable Splash Screen that precedes the main EtchaSketch form.
4. This splash screen has a list of all the key presses .. and when a key is chosen from the list a label displays text explaining what that key does.

Deliverables

In your **log book** you should include:

- Fully commented source code
- Author name, date and project title in the comments
- Sample outputs from your program execution
- Class Diagram
- Completed Test Plan
- Comments on problems encountered, successes, etc.

Assessment of Project

- **Basic Project** + some deliverables (40%)
- **Extension Work 1** + most deliverables (15%)
- **Extension Work 2** + all deliverables (20%)
- **Extension Work 3** + all deliverables (25%)

Marks adjusted according to quality of work submitted and help given