Advance Programming Techniques (APT)

Lecture # 23

Ehtisham Rasheed

Department of Computer Science
University of Gurjat, Gujrat

UNIVERSITY OF GUJRAT

UOG

Mouse Events

- Mouse interactions are a fundamental part of GUI programming. In WinForms, almost every control (Button, Label, PictureBox, Panel, Form itself, etc.) supports mouse-related events

- With mouse events, we can detect:

    - When the mouse enters or leaves a control

    - When the user clicks, double-clicks, or presses mouse buttons

    - The mouse's X/Y coordinates

    - Mouse movement, dragging, and scrolling actions

Mouse Events

- Mouse events help you build interactive applications like:

    - Drawing programs

    - Games

    - Image editors

    - Drag-and-drop UIs

    - Hover-based behavior

Important Mouse Events

- MouseEnter

```
btnSubmit.BackColor = Color.LightBlue;
```

- MouseLeave

```
btnSubmit.BackColor = SystemColors.Control;
```

- MouseMove

```
lblCoordinates.Text = $"X: {e.X}, Y: {e.Y}";
```

- MouseDown

```
lblStatus.Text = $"Mouse Down: {e.Button}";
```

- MouseUp

```
lblStatus.Text = $"Mouse Up: {e.Button}";
```

- MouseClick

```
lblStatus.Text = "Mouse Clicked!";
```

- MouseDoubleClick

```
lblStatus.Text = "Mouse Double Click!";
```

- MouseHover

```
btnSubmit.BackColor = Color.LightBlue;
```

Event Sequence Example

1. MouseEnter
2. MouseMove
3. MouseDown
4. MouseUp
5. MouseClick
6. MouseLeave

Difference Between MouseEnter and MouseHover

- MouseEnter

  - It fires *immediately when the mouse pointer enters* the boundary of a control

- Important Points:

  - It fires **only once** when entering

  - No delay

  - Even if the mouse keeps moving inside the control, **MouseEnter does NOT repeat**

  - Great for hover-effects like changing background color

Difference Between MouseEnter and MouseHover

- MouseHover

  - It fires when the mouse pointer **stops moving** and **stays still** over a control for a short time (a few milliseconds)

- Important Points:

  - There is always a **delay** (called *hover time*)

  - It activates **only if the mouse stops moving**

  - If the mouse keeps moving inside the control, MouseHover will NOT trigger

  - Good for tooltips or showing helpful hints

Drawing with mouse

```csharp
private void panel1_MouseDown(object sender, MouseEventArgs e)
{
    isDrawing = true;
    previousPoint = e.Location;
}
```

```csharp
private void panel1_MouseUp(object sender, MouseEventArgs e)
{
    isDrawing = false;
}
```

```csharp
private void panel1_MouseMove(object sender, MouseEventArgs e)
{
    if (isDrawing)
    {
        using (Graphics g = panel1.CreateGraphics())
        {
            g.DrawLine(Pens.Black, previousPoint, e.Location);
        }
        previousPoint = e.Location;
    }
}
```