

Uses a thread to Animate a ball



```
using System;
using System.Drawing;
using System.Threading;
using System.Windows.Forms;

public class AnimateBall : Form {
    private int x, y;
    private Button suspend = new Button();
    private Button resume = new Button();
    private Button abort = new Button();
    Thread t;

    public AnimateBall() {
        BackColor = Color.White;
        abort.Text = "Abort";
        suspend.Text = "Suspend";
        resume.Text = "Resume";

        int w = 20;
        suspend.Location = new Point(w, 200);
        resume.Location = new Point(w += 10 + suspend.Width, 200);
        abort.Location = new Point(w += 10 + resume.Width, 200);

        abort.Click += new EventHandler(Abort_Click);
        suspend.Click += new EventHandler(Suspend_Click);
        resume.Click += new EventHandler(Resume_Click);

        Controls.Add(suspend);
        Controls.Add(resume);
        Controls.Add(abort);

        t = new Thread(new ThreadStart(Run));
```

```

t.Start();
}
protected void Abort_Click(object sender, EventArgs e) {
t.Abort();
}
protected void Suspend_Click(object sender, EventArgs e) {
t.Suspend();
}
protected void Resume_Click(object sender, EventArgs e) {
t.Resume();
}
protected override void OnPaint( PaintEventArgs e ) {
Graphics g = e.Graphics;
g.FillEllipse(Brushes.Red, 100 ,y, 4 ,4);
base.OnPaint(e);
}
public void Run() {
int dx=2, dy=2;
y = 1;

while (true) {
for(int i=0; i<140; i++) { y+=dy; Invalidate();
Thread.Sleep(10); } dx = -dx; dy = -dy; } } public static void
Main( ) { Application.Run(new AnimateBall()); } } [/csharp]

```