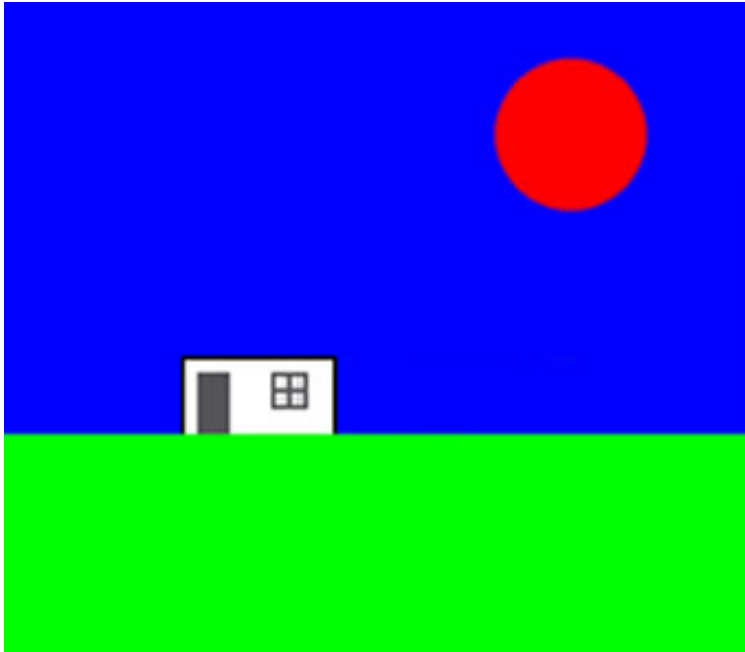


题目：

仿照下面图形在窗体上做图（图形大小自定），当按下鼠标左键后，太阳从左向右移动。当移动到接近右侧边界时停止移动。



第一步：

按照图示实现窗体 **Paint** 事件的响应函数：

```
private void Form1_Paint(object sender, PaintEventArgs e) {  
    Graphics g = e.Graphics;  
    SolidBrush gb = new SolidBrush(Color.FromArgb(0, 255, 0));  
    SolidBrush bb = new SolidBrush(Color.Blue);  
    Pen bp = new Pen(Color.Black, 2.5f);  
    g.FillRectangle(bb, 0, 0, 340, 196);  
    g.FillRectangle(new SolidBrush(Color.White), 80, 161, 70, 35);  
    g.DrawRectangle(bp, 80, 161, 70, 35);  
    g.FillRectangle(new SolidBrush(Color.FromArgb(87, 86, 94)), 87.5f, 168, 14,  
28);  
  
    g.DrawRectangle(bp, 121, 169, 8, 8);  
    g.DrawRectangle(bp, 121 + 8, 169, 8, 8);  
    g.DrawRectangle(bp, 121, 169 + 8, 8, 8);  
    g.DrawRectangle(bp, 121 + 8, 169 + 8, 8, 8);  
    g.FillRectangle(gb, 0, 196, 340, 99);  
    // draw sun  
    g.FillEllipse(new SolidBrush(Color.Red), sunWidth, 24, 70, 70);  
}
```

```
}
```

第二步：

设置私有字段 `sunWidth` 以保存太阳的横坐标。在构造函数中赋初值，并设置双重缓冲，以避免太阳移动时的闪烁。

```
public Form1() {  
    InitializeComponent();  
    this.Size = new Size(340, 295);  
    this.sunWidth = 15;  
    this.DoubleBuffered = true;  
}  
  
private float sunWidth;
```

第三步：

实例化定时器，设定其 `interval` 为 100ms，在其 `Tick` 事件发生时响应太阳横坐标的移动和窗体的重绘：

```
private void timer1_Tick(object sender, EventArgs e) {  
    if (sunWidth < 250)  
        sunWidth = sunWidth + 10;  
    this.Invalidate();  
}
```

第四步：

在窗体 `Click` 事件的响应函数中加入打开计时器：

```
private void Form1_Click(object sender, EventArgs e) {  
    timer1.Start();  
}
```

