

```

public void paint(Graphics g)
{
    g.translate(getInsets().left, getInsets().top);
    int height = getSize().height - getInsets().top;
    int width = getSize().width - getInsets().left;
    size = Math.min(height, width) * 2;

    int x0 = 0, y0 = size/2;

    setTitle("Hilbert curve of order "+Order);
    for (int order = 0; order <= Order; ++order)
    {
        size = size / 2;
        x0 = x0 + (size / 2);
        y0 = y0 - (size / 2);

        /* Note that rounding errors will accumulate, making
           higher order curves a bit off. */
        t = new Turtle(g, x0, y0);

        g.setColor(colours[order]);
        A(order, size);
    }
}

```

```
public void A(int i, int size)
    {
        if (i > 0)
{
    D(i-1, size);
    t.lineTo(t.getX() - size, t.getY());
    A(i-1, size);
    t.lineTo(t.getX(), t.getY() + size);
    A(i-1, size);
    t.lineTo(t.getX() + size, t.getY());
    B(i-1, size);
}
    }
```

```
public void B(int i, int size)
{
    if (i > 0)
{
    C(i-1,size);
    t.lineTo(t.getX(), t.getY() - size);
    B(i-1,size);
    t.lineTo(t.getX() + size, t.getY());
    B(i-1,size);
    t.lineTo(t.getX(), t.getY() + size);
    A(i-1,size);
}
}
```

```
public void C(int i, int size)
    {
        if (i > 0)
        {
            B(i-1,size);
            t.lineTo(t.getX() + size, t.getY());
            C(i-1,size);
            t.lineTo(t.getX(), t.getY() - size);
            C(i-1,size);
            t.lineTo(t.getX() - size, t.getY());
            D(i-1,size);
        }
    }
```

```
public void D(int i, int size)
{
    if (i > 0)
{
    A(i-1,size);
t.lineTo(t.getX(), t.getY() + size);
    D(i-1,size);
t.lineTo(t.getX() - size, t.getY());
    D(i-1,size);
t.lineTo(t.getX(), t.getY() - size);
    C(i-1,size);
}
}
```