

CT255 [2D games in Java]

Week#2 Sample Solution

The application class (single instance)

```
import java.awt.*;
import javax.swing.*;

public class MovingSquaresApplication extends
    JFrame implements Runnable {

    // member data
    private static final Dimension WindowSize = new Dimension(600,600);
    private static final int NUMGAMEOBJECTS = 30;
    private GameObject[] GameObjectsArray = new
GameObject[NUMGAMEOBJECTS];
    private boolean isInitialised = false;

    // constructor
    public MovingSquaresApplication() {
        //Create and set up the window.
        this.setTitle("Threads and Animation");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Display the window, centred on the screen
        Dimension screensize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
        int x = screensize.width/2 - WindowSize.width/2;
        int y = screensize.height/2 - WindowSize.height/2;
        setBounds(x, y, WindowSize.width, WindowSize.height);
        setVisible(true);

        // create+initialise (via their constructor) the GameObject instances
        for (int i=0; i< NUMGAMEOBJECTS; i++) {
            GameObjectsArray[i] = new GameObject();
        }

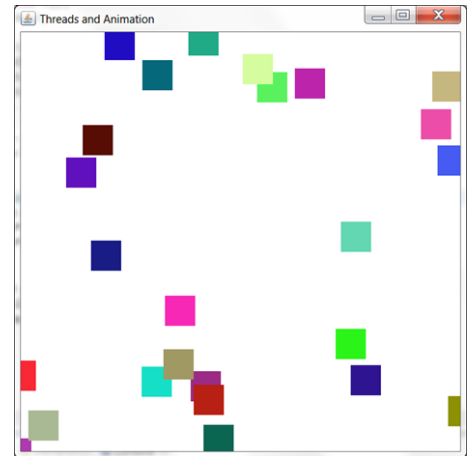
        isInitialised = true;

        // create and start our animation thread
        Thread t = new Thread(this);
        t.start();
    }

    // thread's entry point
    public void run() {
        while ( true ) {

            // 1: sleep for 1/50 sec
            try {
                Thread.sleep(20);
            } catch (InterruptedException e) { }

            // 2: animate game objects
            for (int i=0;i<NUMGAMEOBJECTS; i++)
```



```

        GameObjectsArray[i].move();

        // 3: force an application repaint
        this.repaint();
    }
}

// application's paint method
public void paint(Graphics g) {
    if (!isInitialised)
        return;

    // clear the canvas with a big white rectangle
    g.setColor(Color.WHITE);
    g.fillRect(0, 0, WindowSize.width, WindowSize.height);

    // redraw all game objects
    for (int i=0;i<NUMGAMEOBJECTS; i++)
        GameObjectsArray[i].paint(g);
}

// application's entry point
public static void main(String[] args) {
    MovingSquaresApplication w = new MovingSquaresApplication();
}
}

```

The game object class (instantiated once for each coloured square)

```

import java.awt.*;

public class GameObject {

    // member data
    private double x,y;
    private Color c;

    // constructor
    public GameObject() {
        x = Math.random()*600;
        y = Math.random()*600;
        int r = (int) (Math.random()*256);
        int g = (int) (Math.random()*256);
        int b = (int) (Math.random()*256);
        c = new Color(r,g,b);
    }

    // public interface
    public void move() {
        x += 10*(Math.random()-Math.random());
        y += 10*(Math.random()-Math.random());
    }

    public void paint(Graphics g) {
        g.setColor(c);
        g.fillRect((int)x, (int)y, 40, 40);
    }
}

```