

CT255/NGT2 [2D games in Java]

Week#3 Sample Solution

The main application class (single instance)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class InvadersApplication extends JFrame implements
Runnable, KeyListener {
    // member data
    private static String workingDirectory;
    private static boolean isGraphicsInitialised = false;
    private static final Dimension WindowSize = new
Dimension(600,600);
    private static final int NUMALIENS = 30;
    private Sprite2D[] AliensArray = new
Sprite2D[NUMALIENS];
    private Sprite2D PlayerShip;

    // constructor
    public InvadersApplication() {
        //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);
        this.setTitle("Space Invaders! .. (sort of)");

        // load image from disk. Make sure the path is right! For Mac use / rather than \\
        ImageIcon icon = new ImageIcon(workingDirectory + "\\alien_ship_1.png");
        Image alienImage = icon.getImage();

        // create and initialise some aliens, passing them each the image we have loaded
        for (int i=0; i<NUMALIENS; i++) {
            AliensArray[i] = new Sprite2D(alienImage);
        }

        // create and initialise the player's spaceship
        icon = new ImageIcon(workingDirectory + "\\player_ship.png");
        Image shipImage = icon.getImage();
        PlayerShip = new Sprite2D(shipImage);
        PlayerShip.setPosition(300,530);

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

        // send keyboard events arriving into this JFrame back to its own event handlers
        addKeyListener(this);

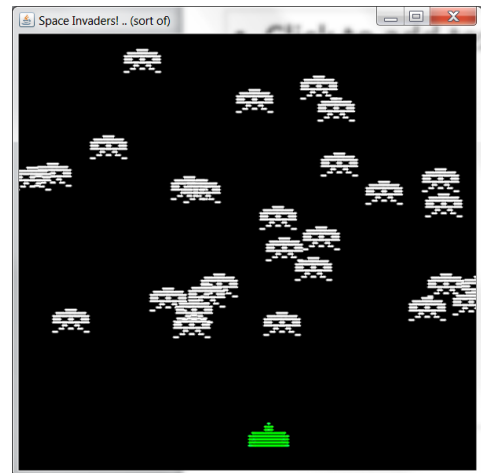
        isGraphicsInitialised = true; // it's now safe to paint the images
    }

    // thread's entry point
    public void run() {
        while ( 1==1 ) { // the game loop
            // 1: sleep for 1/50 sec
            try {
                Thread.sleep(20);
            } catch (InterruptedException e) { }

            // 2: animate game objects
            for (int i=0; i<NUMALIENS; i++)
                AliensArray[i].moveEnemy();

            PlayerShip.movePlayer();

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



```

// Three Keyboard Event-Handler functions
public void keyPressed(KeyEvent e) {
    if (e.getKeyCode()==KeyEvent.VK_LEFT)
        PlayerShip.setXSpeed(-4);
    else if (e.getKeyCode()==KeyEvent.VK_RIGHT)
        PlayerShip.setXSpeed(4);
}
public void keyReleased(KeyEvent e) {
    if (e.getKeyCode()==KeyEvent.VK_LEFT || e.getKeyCode()==KeyEvent.VK_RIGHT)
        PlayerShip.setXSpeed(0);
}
public void keyTyped(KeyEvent e) {}
//
// application's paint method
public void paint(Graphics g) {
    if (isGraphicsInitialised) { // don't try to draw uninitialized objects!
        // clear the canvas with a big black rectangle
        g.setColor(Color.BLACK);
        g.fillRect(0, 0, WindowSize.width, WindowSize.height);

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

        PlayerShip.paint(g);
    }
}
// application entry point
public static void main(String[] args) {
    workingDirectory = System.getProperty("user.dir");
    InvadersApplication w = new InvadersApplication();
}
}

```

The game object class (instantiated once for each alien and once for the player's spaceship)

```

import java.awt.*;
public class Sprite2D {
    // member data
    private double x,y;
    private double xSpeed=0;
    private Image myImage;

    // constructor
    public Sprite2D(Image i) {
        x = Math.random()*600;
        y = Math.random()*600;
        myImage = i;
    }

    // public interface
    public void moveEnemy() {
        x += 10*(Math.random()-Math.random());
        y += 10*(Math.random()-Math.random());
        if (x<0)
            x=0;
        else if (x>600)
            x=600;
        if (y<50)
            y=50;
        else if (y>400)
            y=400;
    }
    public void setPosition(double xx, double yy) {
        x=xx;
        y=yy;
    }
    public void movePlayer() {
        x+=xSpeed;
    }
    public void setXSpeed(double dx) {
        xSpeed=dx;
    }
    public void paint(Graphics g) {
        g.drawImage(myImage, (int)x, (int)y, null);
    }
}
}

```