Omer Boyaci

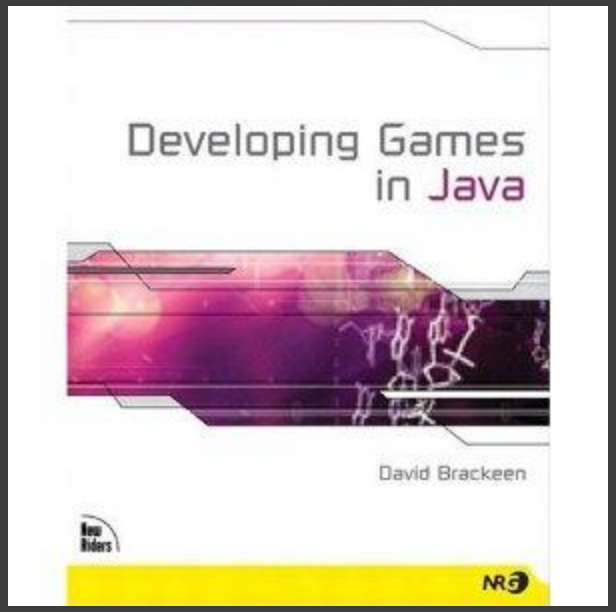
COMPUTER GRAPHICS THROUGH GAME PROGRAMMING

Resources

- http://forums.sun.com/forum.jspa?forumID=406
- http://www.javagaming.org/

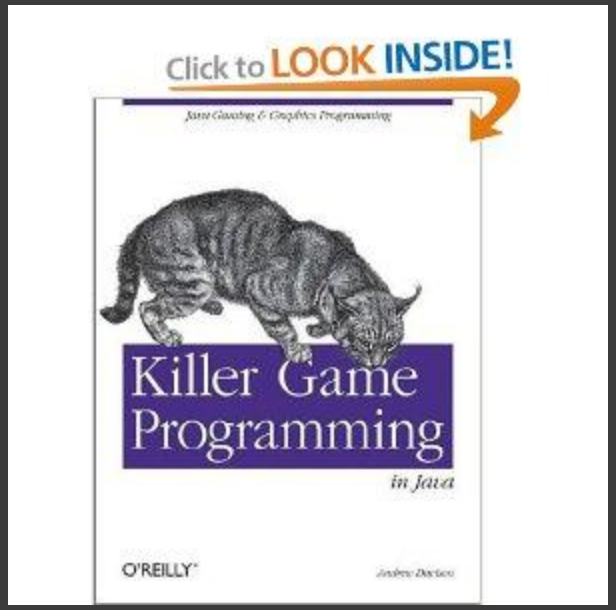
Books

- Developing Games in Java **Andrew Davison**



Books

- **Killer Game Programming in Java Andrew Davison**



Java Game Types

- Applet
- Windowed
- Full screen exclusive mode (FSEM)

FSEM

- J2SE 1.4 introduced full-screen exclusive mode (FSEM). It suspends the normal windowing environment and allows an application to access the underlying graphics hardware more directly. FSEM permits techniques such as page flipping and provides control over the screen's resolution and image depth. The principal aim of FSEM is to accelerate graphics-intensive applications, such as games.

Class JFrame

- Most windows are an instance or subclass of this class
- Provides title bar
- Provides buttons to minimize, maximize and close the application

Some Jframe methods

- – add
  - Adds a component to the JFrame
- – setDefaultCloseOperation
  - **Dictates how the application reacts when the user clicks the close button**
- – setSize
  - **Specifies the width and height of the window**
- – setVisible
  - **Determines whether the window is displayed (true) or not (false)**

Outline

- The {update, render, sleep} animation loop
- Starting and terminating an animation
- Double buffering
- Active rendering
- Animation control based on a user's requested FPS
- The management of inaccuracies in the timer and sleep operations
- Combining FPS and game state *updates per second (UPS)*
- Game pausing and resumption
- User interaction

Animation Example (JFrame)

```
import javax.swing.*;

public class Game1 extends JFrame{

    public Game1(){
        super("Game Application 1");
        GamePanel1 gp1 = new GamePanel1();
        add(gp1);
        setSize(640,480);
        setVisible(true);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        gp1.startGame();
    }

    public static void main(String[] args) {
        Game1 g1 = new Game1();
    }

}
```

Animation Example (GamePanel1)

```
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Point;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;

public class GamePanel1 extends JPanel {

    int x, y;
    int upsc, fpsc;
    long startTime;
    Font f = new Font("SansSerif", Font.BOLD, 24);
    Point[] points = new Point[300];
    Random r = new Random();

    public GamePanel1() {

        for (int i = 0; i < points.length; i++) {
            points[i] = new Point();
            points[i].x = r.nextInt(640);
            points[i].y = r.nextInt(480);
        }
    }
```

```
public void startGame() {
    startTime = System.currentTimeMillis();
    long frameST;
    while (true) {
        frameST = System.nanoTime();
        update();
        repaint();
        frameST  = (System.nanoTime() - frameST)/1000000;
        sleep(10-(int)frameST);
        upsc++;
        if (System.currentTimeMillis() - startTime > 1000) {
            System.out.println("UPS:" + upsc + " FPS:" + fpsc);
            startTime = System.currentTimeMillis();
            upsc = 0;
            fpsc = 0;
        }
    }
}

private void update() {
    if (upsc % 10 == 0)  x++;
    if (upsc % 10 == 0)  y++;
    for (int i = 0; i < points.length; i++) {
        points[i].x = r.nextInt(640);
        points[i].y = r.nextInt(480);
    }
    //for (int x=0;x<Integer.MAX_VALUE/200;x++);
}
```

Animation Example (GamePanel1)

```
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        //System.out.println("paintComponent is called."+ pc++);
        for (int rc = 0; rc < 10; rc++) {
            for (int xc = 0; xc < 20; xc++) {
                for (int yc = 0; yc < 20; yc++) {
                    g.fillRect(x + xc + rc * 20, y + yc + rc * 20, 1, 1);
                }
            }
        }
        /*g.setFont(f);
        for (int i=0;i<10;i++){
        g.drawString("Hello World", x+i*30, y+i*30);
        }
         * */
        /*for (Point p : points){
        g.fillRect(p.x, p.y, 1, 1);
        }*/
        fpsc++;
    }

    private void sleep(int i) {
        try {
            if (i>0) Thread.sleep(i);
        } catch (InterruptedException ex) {
            Logger.getLogger(GamePanel1.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```