

Gaming Canvas

HotGame KiểmHiệp Hay 2013

chanlong.sgame.vn/Downloads+Games

WebGame Nhập Vai nổi tiếng Giang Hồ Chân Long Giáng Thế. Tung Chương!



[Back](#) | [Tutorial Home](#) | [Next](#)

The GameCanvas brings to the table two very important features that solve two problems that previously existed when using MIDP 1.0. First feature is now a single buffer for each GameCanvas that is created. This is important because it not only minimizes the heap usage your game functionality; but also, the game can now be controlled in one single loop. Let us take a look at the old way of looping through a game using MIDP 1.0.

```
public void theGameCanvas extends Canvas implements Runnable {
    public void run() {
        while (true) {
            repaint(); // update game display
        }
        public void paint(Graphics g) {
            // painting // redraw occurs here
        }
        protected void keyPressed(int keyCode) {
            // obtain user inputs
        }
    }
}
```

As you can see there are basically three different areas of functionality, painting of the screen, run() area and key input. all of which run on different threads. Because of the different threads the final display to the user may sometimes seem jerky especially for arcade/action type games that require a lot of graphics and interactiveness. And also it somewhat awkward to keep track of the three different functionalities in three different places.

Now with MIDP 2 and the implementation of GameCanvas, everything is a lot more cleaner, easier to use and more efficient. Aside from running in single thread the GameCanvas no longer waits for a keyPressed event instead it used the a technique called polling, meaning you can determine which keys were pressed at any point time by using the getKeyState() method provide by the GameCanvas. With the use of the buffer in GameCanvas the gaming technique called double buffering graphics is done automatically. All you have to do is call the flushGraphics() method to output the graphics to the display. Double Buffering is a technique that is used to avoid flickering on the display by simply drawing a temporary image off set from the actually display and when completed the image is then shown in visible display area.

Basic GameCanvas Example

The following is a simple example of GameCanvas being used, in this example the string character **x** moves according to the users input.

GameCanvas

```
import javax.microedition.lcdui.*;
import javax.microedition.lcdui.game.*;

public class ExampleGameCanvas extends GameCanvas implements Runnable {
    private boolean isPlay; // Game Loop runs when isPlay is true
    private long delay; // To give thread consistency
```

```
private int currentX, currentY; // To hold current position of the 'X'
private int width;             // To hold screen width
private int height;            // To hold screen height

// Constructor and initialization
public ExampleGameCanvas() {
    super(true);
    width = getWidth();
    height = getHeight();
    currentX = width / 2;
    currentY = height / 2;
    delay = 20;
}

// Automatically start thread for game loop
public void start() {
    isPlay = true;
    Thread t = new Thread(this);
    t.start();
}

public void stop() { isPlay = false; }

// Main Game Loop
public void run() {
    Graphics g = getGraphics();
    while (isPlay == true) {

        input();
        drawScreen(g);
        try { Thread.sleep(delay); }
        catch (InterruptedException ie) {}
    }
}

// Method to Handle User Inputs
private void input() {
    int keyStates = getKeyStates();

    // Left
    if ((keyStates & LEFT_PRESSED) != 0)
        currentX = Math.max(0, currentX - 1);

    // Right
    if ((keyStates & RIGHT_PRESSED) != 0 )
        if ( currentX + 5 < width)
            currentX = Math.min(width, currentX + 1);

    // Up
    if ((keyStates & UP_PRESSED) != 0)
        currentY = Math.max(0, currentY - 1);

    // Down
    if ((keyStates & DOWN_PRESSED) != 0)
        if ( currentY + 10 < height)
            currentY = Math.min(height, currentY + 1);
}
```

```
// Method to Display Graphics
private void drawScreen(Graphics g) {
    g.setColor(0xffffffff);
    g.fillRect(0, 0, getWidth(), getHeight());
    g.setColor(0x0000ff);
    g.drawString("X",currentX,currentY,Graphics.TOP|Graphics.LEFT);
    flushGraphics();
}
}
```

Main Midlet

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class ExampleGameCanvasMidlet extends MIDlet {
    private Display display;

    public void startApp() {
        display = Display.getDisplay(this);
        ExampleGameCanvas gameCanvas = new ExampleGameCanvas();
        gameCanvas.start();
        display.setCurrent(gameCanvas);
    }

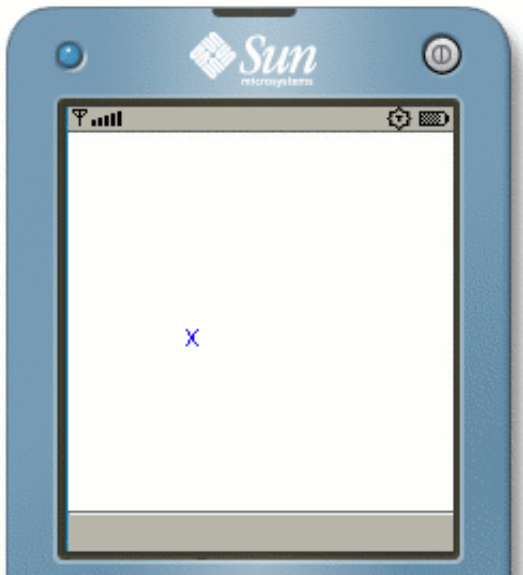
    public Display getDisplay() {
        return display;
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
        exit();
    }

    public void exit() {
        System.gc();
        destroyApp(false);
        notifyDestroyed();
    }
}
```

Simple GameCanvas Example Emulator Screen Shot



j2meSalsa goodies

Execute Game canvas Sample online

Download demo code for deploying directly to WTK

Execute on your browsers left frame.

[Click here](#) to download Zip File Containing WTK compatible file structure.

Download Ringtones - Free

www.Mobogenie.com/Download

Android SmartPhone PC Manager. One-Click Download! (Windows Only)



[Back](#) | [Tutorial Home](#) | [Next](#)

3 comments



Leave a message...

Best ▾

Community

Share



chirag • 4 months ago

how do u move a game canvas in a loop?

^ | ▾ Reply Share ›



Floryfrancismsrit • a year ago

Useful !!! Thank you !

^ | ▾ Reply Share ›



Ishan1211 • 2 years ago

This is very useful. Thank you.

^ | ▾ Reply Share ›

Subscribe

Add Disqus to your site

[site comments powered by Disqus](#)

This page is a part of a frames based web site. If you have landed on this page from a search engine [click here](#) to view the complete page.