

Como bloquear o screen saver em Java ME

Este artigo demonstra uma maneira de impedir o screen saver de funcionar durante a execução de uma aplicação.

Se sua aplicação não requer pressionamento de teclas constantemente, depois de algum tempo o screen saver será iniciado pelo sistema operacional automaticamente.

Para certificar-se de que a luz do display está acesa, use o método `setLights` antes de o screen saver ser acionado, logo chame esse método em loop, já que o screen saver não é desabilitado, apenas interrompido, ou seja, após alguns segundos ele será acionado novamente.

```
import com.nokia.mid.ui.DeviceControl;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class BacklightWorkaround extends MIDlet {

    private SimpleCanvas canvas;

    /**
     * Mantém a luz de fundo acesa (ativa em intervalos de tempos de 5000ms)
     */
    class LightThread extends Thread {
        public void run() {
            while(true){

                DeviceControl.setLights(0, 100);
                try {
                    Thread.sleep(5000);
                } catch (InterruptedException ex) {
                    ex.printStackTrace();
                }
            }
        }
    }

    private class SimpleCanvas extends Canvas implements CommandListener{
        private Command exitCmd;
        private MIDlet midlet;

        public SimpleCanvas(MIDlet midlet) {
            this.midlet = midlet;
            exitCmd = new Command("Exit",Command.EXIT, 1);
            addCommand(exitCmd);
            setCommandListener(this);
        }
        public void paint(Graphics g) {
            g.drawString("Let there be light.", 0, 0, Graphics.LEFT|Graphics.TOP);
        }

        public void commandAction(Command command, Displayable displayable) {
            if(command == exitCmd){
                midlet.notifyDestroyed();-----
            }
        }
    }
}
```

```
public void startApp() {  
    if(canvas == null){  
        canvas = new SimpleCanvas(this);  
        new LightThread().start();  
    }  
  
    Display.getDisplay(this).setCurrent(canvas);  
}  
  
public void pauseApp() { }  
  
public void destroyApp(boolean unconditional) { }  
}
```