Double Buffering

U. Breymann H. Mosemann

Ergänzendes Material zum Buch »Java ME«
http://www.java-me.de

Weil ab MIDP 2.1 Double Buffering von allen Implementierungen unterstützt werden soll und sich damit über kurz oder lang das Problem, Double Buffering selbst zu implementieren, erledigt haben wird, verzichten wir im Buch aus Platzgründen auf weitere Einzelheiten. Für alle, die dennoch wissen möchten, wie man Double Buffering selbst implementieren kann, gibt es die folgende Kurzanleitung.

Bildflackern kommt dadurch zustande, dass Zeichenoperationen bei manchen Geräten direkt auf dem Bildschirm ausgeführt werden. `paint()` ist, wie beschrieben, für den *ganzen* Bildschirm zuständig, muss also den Bildschirm vorher löschen, ehe zum ersten Mal, zum Beispiel nach `showNotify()`, gezeichnet wird. Der Wechsel von Bild löschen und neuzeichnen bewirkt den flackernden Eindruck. Es gibt mehrere Wege, Abhilfe zu schaffen. Eine Möglichkeit ist die Bereitstellung eines zweiten, nicht sichtbaren Bildspeichers (daher »double buffering«), in den die Grafik geschrieben wird. Andere Möglichkeiten werden im nächsten Abschnitt besprochen. Der Inhalt dieses Bildspeichers wird erst nach Ende aller Zeichenoperationen zur Anzeige gebracht, ein Vorgang, der erheblich schneller als die Zeichenoperationen ist und deswegen nicht wahrgenommen wird.

Manche Geräte unterstützen Double Buffering von sich aus, in solchen Fällen braucht man es nicht einzusetzen. Um portabel zu sein, sollte man eine eigene Implementierung von Double Buffering vorsehen und sie nur bei Bedarf, d.h. wenn Double Buffering seitens der Implementierung nicht gegeben ist, einschalten. Das geht so:

1. In die eigene `Canvas`-Klasse werden zusätzlich Attribute für den Bildspeicher eingestellt, zum Beispiel:

```
public class CanvasDoubleBuf extends Canvas
                                    implements Runnable {
    // für Double Buffering
    private boolean doubleBufferON = false;
    private Image img = null;
```

```
    private Graphics buffer = null;
    // ... weitere Attribute
```

2. Die Attribute werden im Konstruktor initialisiert. Dabei wird Speicher nur bei Bedarf zugewiesen, wenn nämlich Double Buffering von der Implementation nicht unterstützt wird:

```
public CanvasDoubleBuf() {
    // bei Bedarf einschalten
    doubleBufferON = !super.isDoubleBuffered();
    if(doubleBufferON) {
      // Puffer der richtigen Größe bereitstellen
      img = Image.createImage(getWidth(), getHeight());
      buffer = img.getGraphics();
    }
    // ...
```

3. Die Funktion `isDoubleBuffered()` muss, da wir Double Buffering einschalten, überschrieben werden. Wegen der Existenz dieser Funktion wird im vorhergehenden Punkt die entsprechende Funktion der Oberklasse `Canvas` aufgerufen.

```
public boolean isDoubleBuffered() { return true;}
```

4. Der Code der bereits fertigen Funktion `paint()` soll erhalten bleiben, wenn Double Buffering nachgerüstet wird. Daher wird `void paint(Graphics g)` einfach in `private void zeichnen(Graphics g)` umbenannt, und wir schreiben eine neue Funktion `paint()`:

```
protected void paint(Graphics g) {
    if(doubleBufferON) {
        zeichnen(buffer);
        g.drawImage(img, 0, 0, 0);
    }
    else {
        zeichnen(g);
    }
}
```

   Die neue Funktion `paint()` sorgt dafür, dass bei eingeschaltetem Double Buffering zuerst in den zweiten Speicher `buffer` geschrieben wird. Danach wird das Bild zur Anzeige gebracht. Die Parameter `(img, 0, 0, 0)` bedeuten in der Reihenfolge von links nach rechts: zu zeichnende Grafik, x-Koordinate, y-Koordinate, Ausrichtung (hier: oben links).

Anmerkung zum Sun WTK

WTK Problem

Das Sun WTK unterstützt Double Buffering. Um den Effekt eines fehlenden Double Buffering sehen zu können, muss es über »Edit« → »Preferences« → »Performance« ausgeschaltet werden. Klicken Sie statt »Double Buffer« den Radio-Button »Immediate« an. Allerdings liefert die Funktion `isDoubleBuffered()` in den Versionen 2.2 und 2.3 beta des WTK weiterhin `true`, weswegen die Variable `doubleBufferON` (siehe Punkt 2 auf Seite 2) zum Ausprobieren auf `true` gesetzt werden sollte.