

**HARDWARE-SKINS**

# Echtes Look and Feel

Virtuelle Android-Geräte lassen sich mit eigenen Hardware-Skins ausstatten.

**M**it dem AVD (Android Virtual Device) Manager [1], der mit Android Studio [2] geliefert wird, können virtuelle Android-Geräte erstellt werden, die sich zum Testen bei der App-Entwicklung eignen. Solche Geräte lassen sich optional mit einem Hardware-Skin ausstatten ([Bild 1](#), [Bild 2](#)).

Einen Hardware-Skin einzusetzen ist dann sinnvoll, wenn die Hardware-Buttons des Geräts grafisch emuliert werden sollen, aber auch, wenn Sie einen realistischen Eindruck darüber bekommen wollen, wie die fertige App auf einem speziellen Modell aussehen wird.

Da der AVD Manager nur eine recht begrenzte Anzahl an Skins mitbringt, demonstriert dieser Artikel am Beispiel des Samsung Galaxy S6 [3] und des Grafikeditors Paint.net [4], wie Sie beliebige Skins erstellen können.

## Erstellen der Hintergrundgrafik

Als Erstes fertigen Sie eine Grafik des Zielgeräts an. Oft werden Sie in den Online-Produktkatalogen der Hersteller fündig; so stellt etwa Samsung Bilder im Format 900 x 600 Pixeln mit reichlich weißem Leerraum zur Verfügung ([Bild 3](#)). Dabei ist für den Emulator vor allem die Größe des Bildschirmbereichs interessant.

Öffnen Sie die Grafik in Paint.net, messen Sie mit dem Werkzeug *Rechteckige Auswahl* den Bildschirmbereich aus (hier: 256 x 455 Pixel) und rechnen Sie dann per Dreisatz auf die Pixelanzahl des physikalischen Geräts hoch. Anschließend passen Sie die Bildgröße über *Bild, Größe ändern* entsprechend an. Im vorliegenden Fall muss um den Faktor

**Samsungs** Produktbild für das blaue Galaxy S6 ([Bild 3](#))

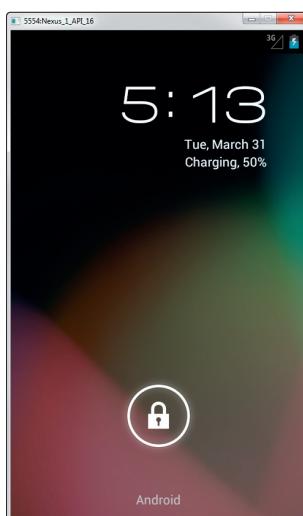
5,625 vergrößert werden, um das Zielformat von 1440 x 2560 Pixeln zu erreichen.

Stellen Sie als Nächstes die Grafik frei. Produktfotos stehen meist vor einfarbigem Hintergrund, sodass einfaches Zuschneiden ausreicht. Dazu setzen Sie mit dem Werkzeug *Zauberstab* eine Maske auf den Hintergrund, wählen *Bearbeiten, Auswahl umkehren* und schließlich *Bild, Auf Markierung zuschneiden*.

Darauf setzen Sie im Fenster *Farben* die Sekundärfarbe auf transparent (Transparent-Alpha = 0) und fügen über *Bild, Leinwandgröße* jeder Seite einen Leerraum von circa fünf Prozent der längsten Seite der Grafik hinzu; dieser wird für die Anzeige der seitlichen Hardware-Buttons benötigt. Bei Bedarf können Sie das Bild noch über *Korrekturen* oder *Effekte* anpassen – so lassen sich über *Effekte, Unschärfe und Weichzeichner, Oberflächenunschärfe* Artefakte beseitigen, die durch die Bildvergrößerung entstanden sind.



Das Google Nexus One mit ...  
([Bild 1](#))



... und ohne Hardware-Skin.  
([Bild 2](#))

## Pixelmaße des Zielgeräts

Legen Sie anschließend mit dem Werkzeug *Rechteck* eine schwarze Fläche über den Bildschirmbereich. Wenn alles stimmt, hat diese Fläche die genauen Pixelmaße des Zielgeräts, wobei kleine Abweichungen nicht weiter tragisch sind. Notieren Sie sich bei dieser Gelegenheit auch die Koordinaten der linken oberen Ecke des Rechtecks – die Angabe wird später bei der Konfiguration benötigt.

Sie können diese Werte in der Statusleiste des Grafikeditors ablesen, nachdem Sie eine Rechteckmaske (Werkzeug *Rechteckige Auswahl*) über die Fläche gelegt haben ([Bild 4](#)).

Speichern Sie die fertige Grafik ([Bild 5](#)) unter dem Namen *background.png* in einem neuen Ordner *galaxy\_s6* ab. Für das Querformat drehen Sie das noch geöffnete Bild nach links (*Bild, Um 90° entgegen dem Uhrzeigersinn drehen*) und spei-



ichern es unter `background_land.png` ab. Beachten Sie, dass Sie so erstellte Grafiken nicht ohne Zustimmung des Urhebers weitergeben oder veröffentlichen dürfen.

## Erstellen der Button-Grafiken

Für die Hardware-Buttons benötigen Sie teiltransparente Overlay-Grafiken. Diese werden vom Emulator bei einem Mouseover-Ereignis eingeblendet sowie beim Anklicken mit der Maus nochmals überlagert. Ein guter Ausgangspunkt sind die Grafiken der vorhandenen Skins im Verzeichnis `C:\Program Files\Android\Android Studio\plugins\android\lib\device-art-resources\`, die Sie bei Bedarf mit Paint.net weiter anpassen oder ergänzen müssen.

Um beispielsweise eine Overlay-Grafik für den Menü-Button zu erstellen, gehen Sie wie folgt vor: Öffnen Sie zunächst die Hintergrundgrafik (`background.png` oder `background-`

`land.png`) und wählen Sie *Ebenen, Neue Ebene hinzufügen*. Öffnen Sie dann die Grafik `button.png` aus dem Nexus-One-Skin und kopieren Sie sie in die neu erzeugte Ebene. Ziehen Sie die Grafik auf die gewünschte Größe und verschieben Sie sie an den Zielort auf der Hintergrundgrafik. Nun können Sie wiederum die Zielkoordinaten ablesen (**Bild 6**). Da der Farbkontrast in diesem Fall zu schwach ist, ändern Sie über *Korrekturen, Farnton/Sättigung* die Werte für Farnton und Helligkeit. Zuletzt kopieren Sie die Button-Grafik in die Zwischenablage, wählen *Datei, Neu, OK*, fügen die Grafik ein, und speichern sie im Ordner `galaxy_s6` ab. Die fertigen Grafiken zeigt **Tabelle 1**.

## Konfiguration

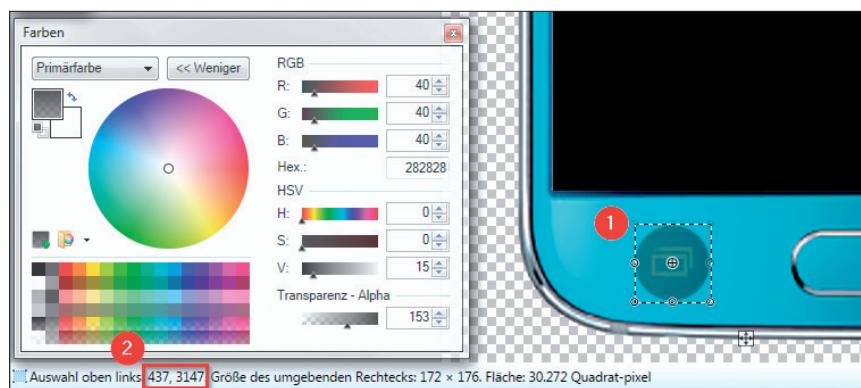
Die Konfiguration des Skins erfolgt über die Datei `layout` (**Listing 1**). Sie gliedert sich in einen Abschnitt `parts`, in den ►

► Tabelle 1: Die angefertigten Grafiken

Grafik Hochformat	<code>background.png</code>	<code>button.png</code>	<code>home.png</code>	<code>power.png</code>	<code>volumedown.png</code>	<code>volumeup.png</code>
Grafik Querformat	<code>background_land.png</code>	(wie Hochformat)	<code>home_land.png</code>	<code>power_land.png</code>	<code>volumedown_land.png</code>	<code>volumeup_land.png</code>
Funktion	Hintergrund	Button Menü, Button Zurück	Button Startseite	Button Ein/Aus	Button Lautstärke verringern	Button Lautstärke erhöhen

Werte über den Bildschirm und die Anordnung der Buttons relativ zum Hintergrund eingetragen werden, und einen Abschnitt *layouts* mit Angaben zur Gestaltung des Emulatorfensters.

Die meisten Werte sind selbsterklärend. Welche Buttons emuliert werden können, ersehen Sie aus [Tabelle 2](#). Sie sind nicht offiziell dokumentiert und müssen bei Bedarf aus dem Android-Quellcode ermittelt werden (suchen Sie in [github.com/android/platform\\_external\\_qemu/blob/master/android/skin/file.c](https://github.com/android/platform_external_qemu/blob/master/android/skin/file.c) nach dem Array `keyin-`



**Ablesen** der Koordinaten (2) für die Overlay-Grafik des Menü-Buttons (1) ([Bild 6](#))

### Listing 1: Konfigurationsdatei layout (Teil 1)

```
# Bildschirm und Anordnung der Buttons
parts {
    # Physikalische Eigenschaften des Geräts
    device {
        # Bildschirm
        display {
            width 1440 # Breite des Bildschirms in Pixeln
            height 2560 # Höhe des Bildschirms in Pixeln
        }
    }

    # Layout im Hochformat
    portrait {
        # Hintergrundgrafik
        background {
            image background.png
        }

        # Buttons
        buttons {
            # Button "Ein/Aus"
            power {
                image power.png
                x     1738
                y     1226
            }

            # Button "Lautstärke erhöhen"
            volume-up {
                image volumeup.png
                x     35
                y     516
            }

            # Button "Lautstärke verringern"
            volume-down {
                image volumedown.png
                x     35
                y     903
            }
        }
    }

    # Button "Startseite"
    home {
        image home.png
        x     787
        y     3157
    }

    # Button "Zurück"
    back {
        image button.png
        x     1349
        y     3147
    }

    # Button "Menü"
    soft-left {
        image button.png
        x     437
        y     3147
    }
}

# Layout im Querformat
landscape {
    background {
        image background_land.png
    }

    buttons {
        power {
            image power_land.png
            x     1226
            y     43
        }

        volume-up {
```

*fo\_table*). Beachten Sie, dass der Menü-Button über *soft-left* emuliert wird. Wenn Grafikelemente rotiert werden, verändern sie ihre x-/y-Koordinaten. Im Beispiel erfolgt eine Rotation des Bildschirmbereichs um 270° im Uhrzeigersinn ins Querformat. Da die Rotation um den linken oberen Punkt erfolgt, muss der Bildschirm anschließend um seine eigene Breite nach unten verschoben werden; addieren Sie also zum y-Wert die Bildschirmbreite in Pixeln.

## Hoch- und Querformat

Neben den üblichen Layouts für Hoch- und Querformat können Sie weitere Layouts definieren. Für Geräte mit Hardware-Tastatur wären beispielsweise getrennte Layouts für

Hochformat mit geöffneter Hardware-Tastatur und Hochformat mit geschlossener Hardware-Tastatur sinnvoll. Damit der Emulator zwischen diesen beiden Zuständen unterscheiden kann, müssen Sie im Abschnitt *layouts* das Hardware-Ereignis *EV\_SW* wie folgt definieren:

```
layouts {
    portrait_keyboard_closed
    {
        ...
        # Linux Event Code für "Hardware-Tastatur
        # geschlossen"
        event EV_SW:0:1
```



Listing 1: Konfigurationsdatei layout (Teil 2)

```
image volumeup_land.png
x      516
y      1746
}

volume-down {
    image volumedown_land.png
    x      903
    y      1746
}

home {
    image home_land.png
    x      3157
    y      787
}

back {
    image button.png
    x      3147
    y      422
}

soft-left {
    image button.png
    x      3147
    y      1331
}

# Layout des Emulatorfensters
layouts
{
    # Hochformat
    portrait {
        width 1946
        # Breite der Hintergrundgrafik in Pixeln
    }

    # Querformat
    landscape {
        width 3574
        height 1946
        color 0x00ffff

        part1 {
            name portrait
        }

        part2 {
            name device
            # Verschiebung des Bildschirms innerhalb
            # der Hintergrundgrafik
            x      252
            y      512
        }
    }

    # Querformat
    orientation {
        width 3574
        height 1946
        color 0x00ffff

        part1 {
            name landscape
        }

        part2 {
            name device
            x      512
            y      1694
            # Ausgemessener Wert (254) +
            # Display-Breite (1440)
            rotation 3
            # Rotation um 3 * 90° = 270° im Uhrzeigersinn
        }
    }
}
```

### Links zum Thema

- AVD Manager  
<http://developer.android.com/tools/help/avd-manager.html>
- Samsung-Galaxy-Produktseite  
[www.samsung.com/de/consumer/mobile-devices/smartphones](http://www.samsung.com/de/consumer/mobile-devices/smartphones)
- Android Emulator Skin File Specification  
<http://android.googlesource.com/platform/external/qemu.git/+/master/docs/ANDROID-SKIN-FILES.TXT>
- Linux input protocol event types  
[www.kernel.org/doc/Documentation/input/event-codes.txt](http://www.kernel.org/doc/Documentation/input/event-codes.txt)
- Paint.net  
[www.getpaint.net](http://www.getpaint.net)

```

    ...
}

portrait_keyboard_open
{
    ...

    # Linux Event Code für "Hardware-Tastatur geöffnet"
    event EV_SW:0:0
    ...

}
}

```

Das Ereignis wird dann bei Initialisierung des Layouts ausgelöst. EV steht für *event*, SW bedeutet *stateful binary switches*. Das Ereignis wird zum Beispiel auch bei Laptop-Computern ausgelöst, wenn das Gerät zugeklappt wird. Die erste Ziffer bezeichnet die Art des Ereignisses, die zweite den Wert (hier: 1 = wahr, 0 = falsch).

Der Emulator kann weitere Hardware-Ereignisse auslösen, von denen aber die wenigsten eine Bedeutung für Hardware-

Tabelle 2: Emulierbare Hardware-Buttons

Name des Buttons	Beschreibung
<b>Mobilgeräte</b>	
back	Zurück
camera	Kamera
home	Startseite
power	Ein/Aus
search	Suche
soft-left	Von Software belegte Taste links (meist Menü)
soft-right	Von Software belegte Taste rechts
volume-down	Lautstärke verringern
volume-up	Lautstärke erhöhen
<b>Standard- und Telefon-Tastatur</b>	
(0 – 9, a – z)	Ziffern und Buchstaben (ohne Umlaute)
ALT	Alt links
ALT2	Alt rechts
AT	@
CAP	Umschalten (Shift) links
CAP2	Umschalten (Shift) rechts
COMMA	,
DEL, del	Rücktaste (Backspace)
ENTER	Eingabe (Return)
PERIOD	.
phone-dial	Anrufen
phone-hangup	Anruf beenden
pound	#

Name des Buttons	Beschreibung
SLASH	/
SPACE	Leerzeichen
star	*
SYM	Symbole umschalten
<b>Direktionales Steuer-Pad (D-Pad)</b>	
dpad-center	Mitte
dpad-down	Ab
dpad-left	Links
dpad-right	Rechts
dpad-up	Auf
<b>Audio/Video</b>	
bookmarks	Lesezeichen / Favoriten
channel-down	Vorheriger Kanal
channel-up	Nächster Kanal
dvr	Digitaler Videorekorder
epg	Elektronisches Programmverzeichnis
ffwd	Vorspulen
next	Nächste Aufnahme
pause	Pause
play	Abspielen
prev	Vorherige Aufnahme
rev	Zurückspulen
stop	Anhalten
tv	Fernseher
window	Fensterfunktionen

Skins haben. Sie können eine Liste dieser Ereignisse über die Kommandozeile des Emulators abrufen, indem Sie sich per Telnet mit dem emulierten Gerät verbinden (*telnet localhost 5554*) und anschließend den Befehl *event types* ausführen. Weitere Details erfahren Sie über das Kommando *help event*.

Die komplette Spezifikation zur *layout*-Datei können Sie in der Android Emulator Skin File Specification einsehen [5].

## Anwenden des Skins

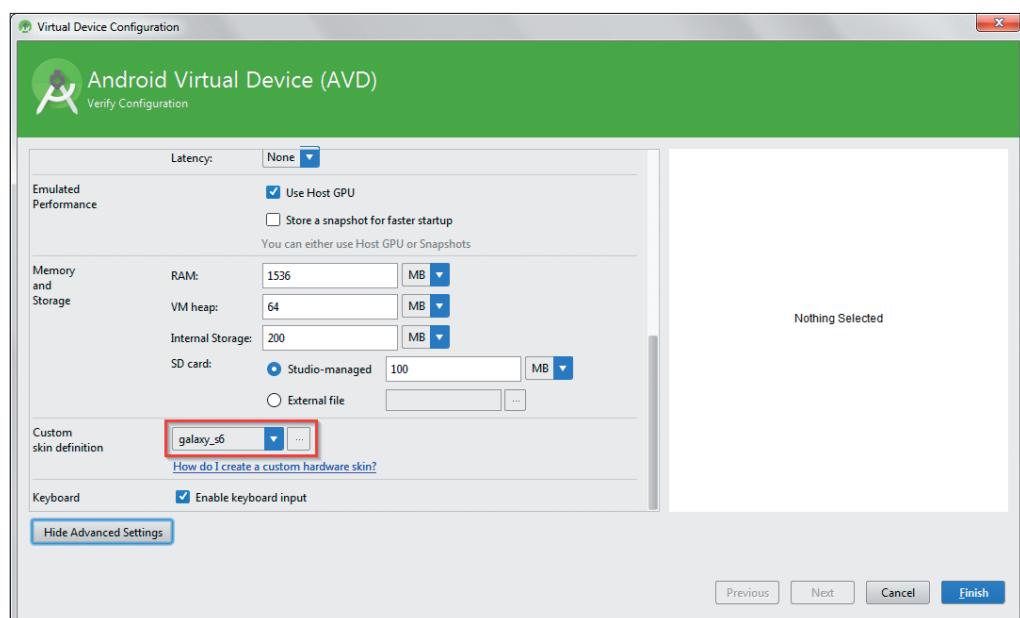
Benutzen können Sie den Skin, indem Sie ihn einem virtuellen Gerät zuweisen. Dazu öffnen Sie im AVD Manager die Eigenschaftsseite des Geräts, klicken auf *Show Advanced Settings* und wählen den Ordner *galaxy\_s6* aus (Bild 7). Anschließend erscheint das Gerät im neuen Gewand (Bild 8).

Neben dem Emulatorfenster kann man den Skin auch in der Vorschau von Android Studio nutzen. Wenn Sie dies anstreben, entwerfen Sie am besten passende Grafiken für die Schatten und fügen den folgenden XML-Code in die Datei *C:\Program Files\Android\Android Studio\plugins\android\lib\device-art-resources\device-art.xml* ein:



```
<device id="galaxy_s6"
name="Samsung Galaxy S6">
<orientation name="port"
size="1946,3574"
screenPos="252,512"
screenSize="1440,2560"
shadow="shadow.png"
back="background.png"/>
<orientation name="land"
size="3574,1946"
screenPos="512,254"
screenSize="2560,1440"
shadow="shadow_land.png"
back="background_land.
png"/>
</device>
```

**Fertig:** Ein hübsches Smartphone mit klickbaren Buttons, hier mit Mouseover-Effekt am Ein/Aus-Button (Bild 8)



Zuweisen des Skins (Bild 7)

Anzeige in Android Studio funktioniert auch nicht immer zuverlässig, und ein Skin bringt keinen echten Mehrwert, da die Vorschau in erster Linie dem Entwurf von Seitenlayouts dient und keine Funktionalität zum Anklicken von Buttons mitbringt.

## Fazit

Ein virtuelles Android-Gerät mit passendem Hardware-Skin sieht schön aus, emuliert alle Hardware-Buttons und vermittelt einen plastischen Eindruck einer laufenden App. Der Aufwand für die Erstellung eines Skins ist besonders bei Geräten mit wenigen Buttons relativ gering.

Verzichten Sie allerdings auf einen Skin in der Android-Studio-Vorschau und berücksichtigen Sie beim Erstellen von Bildern das Urheberrecht.

[1] AVD (Android Virtual Device) Manager:

<http://developer.android.com/tools/help/avd-manager.html>

[2] Android Studio: <http://developer.android.com>

[3] Samsung Galaxy S6: [www.samsung.com/de/consumer/mobile-devices/smartphones/galaxy-s/SM-G920FZKADBT](http://www.samsung.com/de/consumer/mobile-devices/smartphones/galaxy-s/SM-G920FZKADBT)

[4] Paint.net: [www.getpaint.net](http://www.getpaint.net)

[5] Android Emulator Skin File Specification: <http://android.googlesource.com/platform/external/qemu.git/+/master/docs/ANDROID-SKIN-FILES.TXT>



**Martin Gossen**

ist als IT-Berater bei der IKS GmbH tätig und befasst sich unter anderem mit Themen der Usability und Sicherheit im Bereich Web und Mobile.

[m.gossen@iks-gmbh.com](mailto:m.gossen@iks-gmbh.com)