



Android 4.4 in einer Oracle VM VirtualBox
installieren

Android 4.4 in einer Oracle VM VirtualBox installieren

Inhalt

Installations- Vorbereitungen	3
Virtuelle Maschine in Oracle VM VirtualBox einrichten.....	4
Android 4.4 auf die Festplatte installieren.....	9
Android 4.4 starten und erste Einstellungen vornehmen.....	19
Android 4.4 Grundkonfiguration	24
Android 4.4 mit der Maus bedienen	31

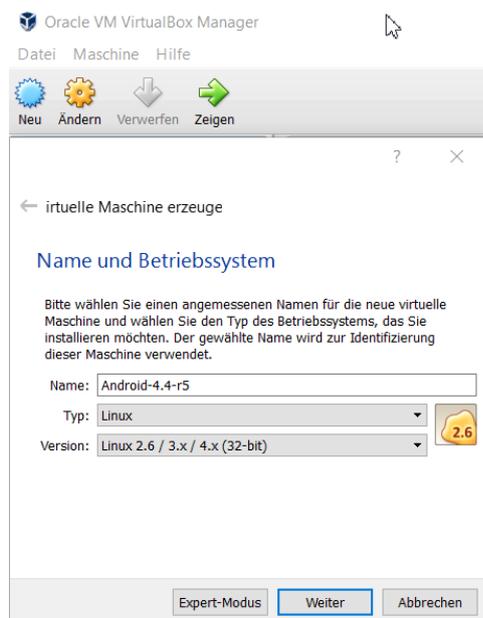
Installations- Vorbereitungen

Um dieses HowTo schreiben zu können, wurde das Ganze in einer Virtuellen Umgebung mit Oracle VM VirtualBox nachgestellt.

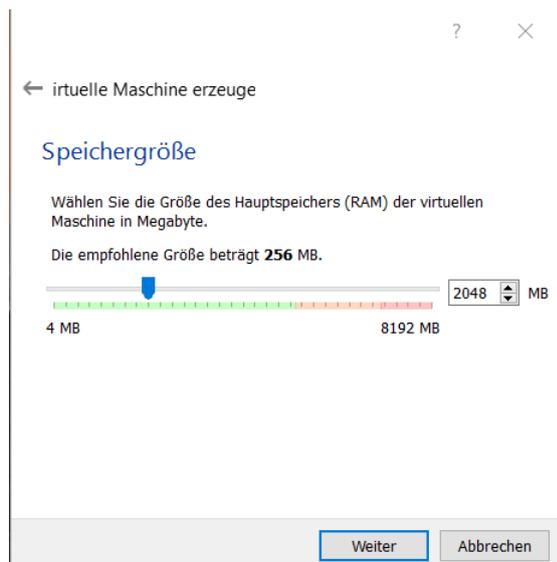
- ↳ Oracle VM VirtualBox -- <https://www.virtualbox.org>
- ↳ Android Image Download -- <http://www.android-x86.org/download>

Virtuelle Maschine in Oracle VM VirtualBox einrichten

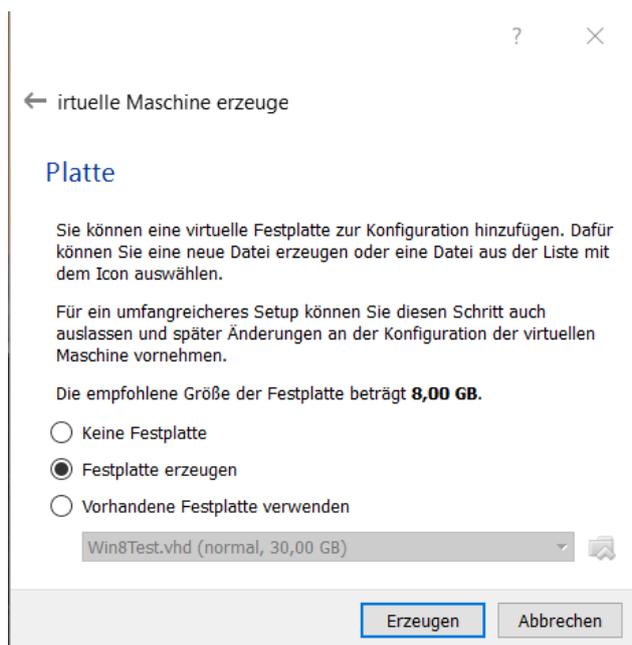
Habe nun alle Images durchgetestet, (Version Android 4.4) funktioniert bis jetzt am besten in der Virtuellen Maschine, bei alle anderen wie z.B. Android 5.1 und 6.0 gibt es immer wieder Probleme Apps aus dem Google Play Store herunterzuladen. Vielleicht kann ja jemand diese Howto gebrauchen. 😊



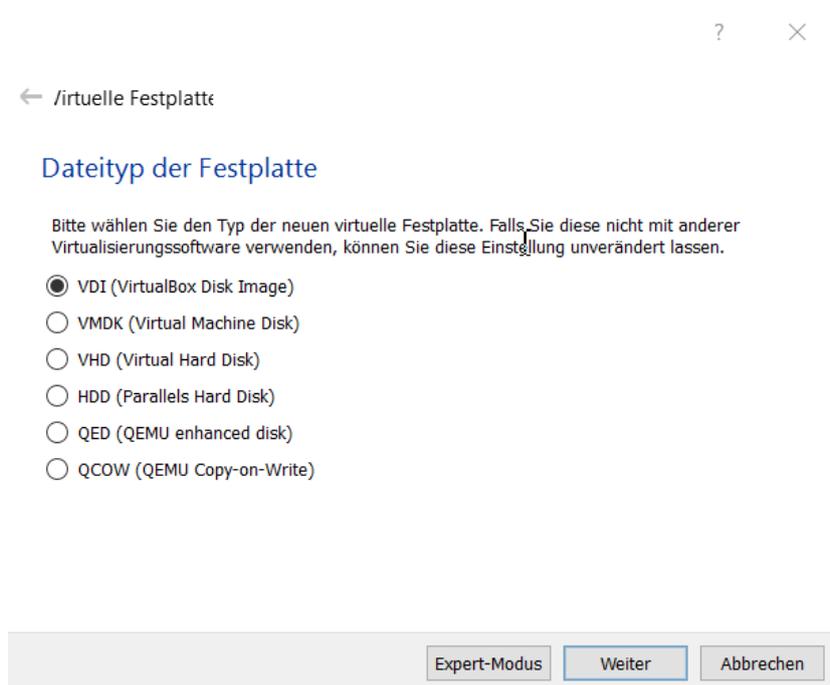
Nachdem Oracle VM VirtualBox installiert ist, legen wir eine neue Virtuelle Maschine an, vergeben einen aussagekräftigen Namen, wählen als Betriebssystem Linux aus und wählen die zu installierte Version - Linux 2.6/3.x/4.x (32-bit) aus.



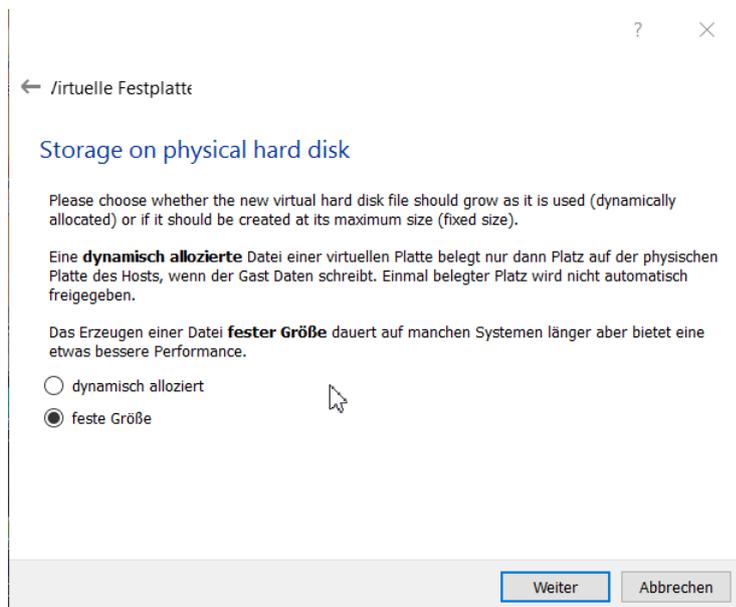
Nachdem wir auf Weiter geklickt haben vergeben wir zwischen 1024 und 2048 MB RAM.



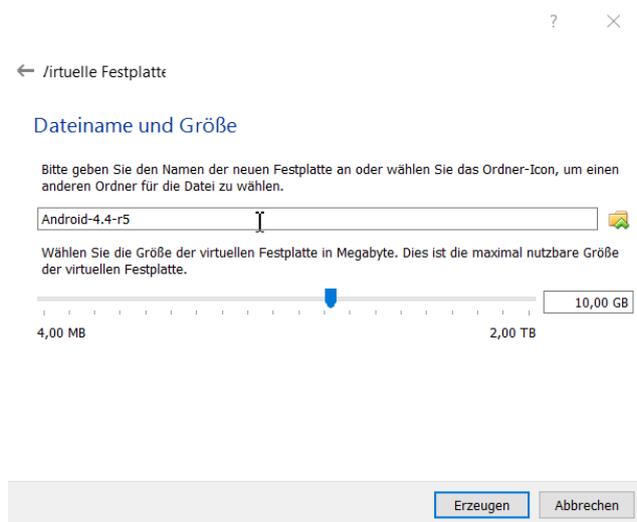
Danach erzeugen wir eine neue Festplatte.



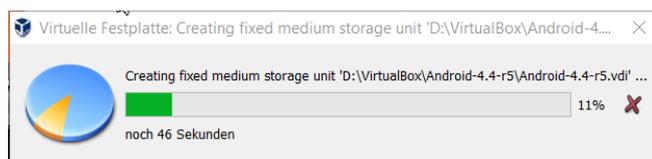
Welche Art der Festplatte ist egal, habe mich hier für das interne Format VDI entschieden.



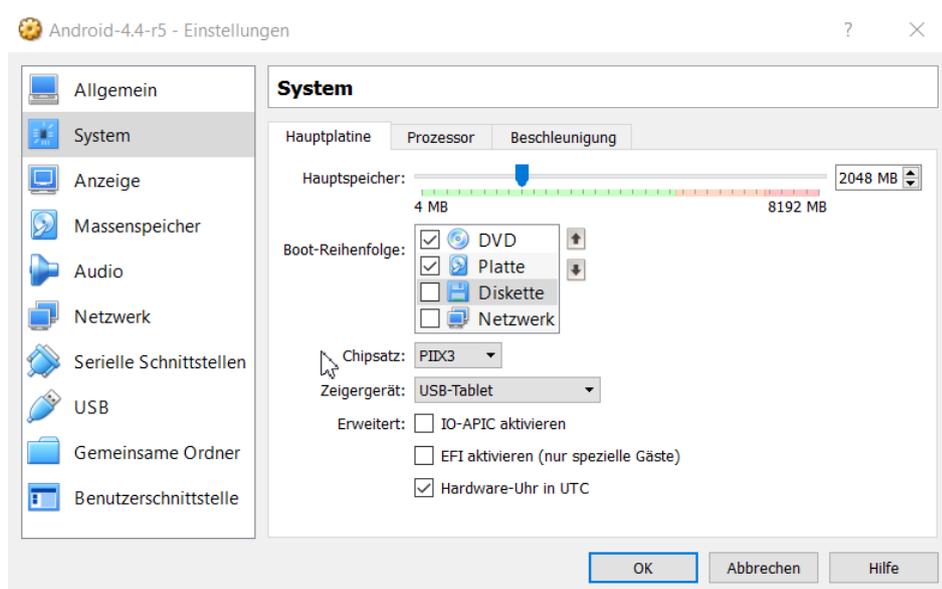
Wir wählen eine feste Größe für die Festplatte aus.



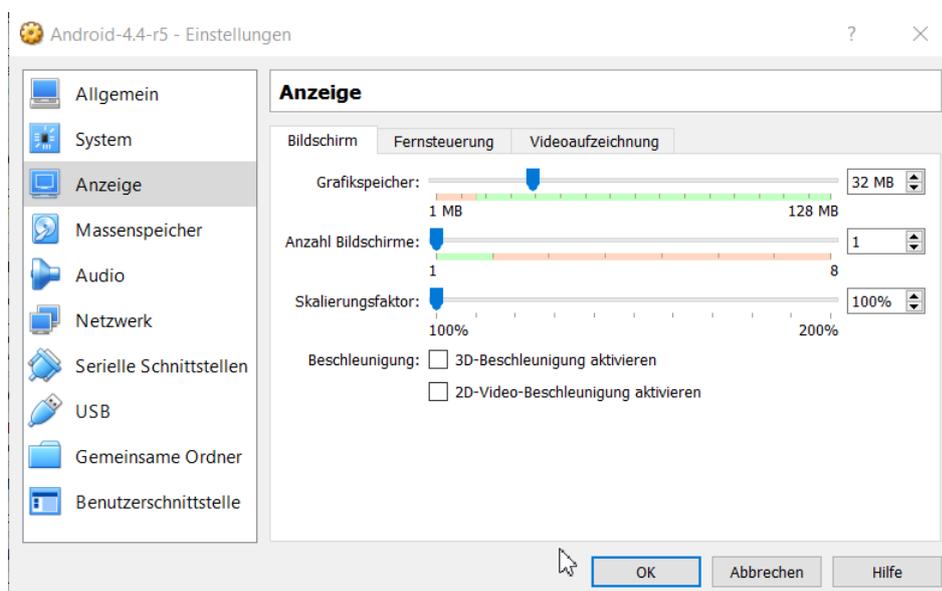
Die Größe der Festplatte definiere ich jetzt zwischen 8 und 10 GB.



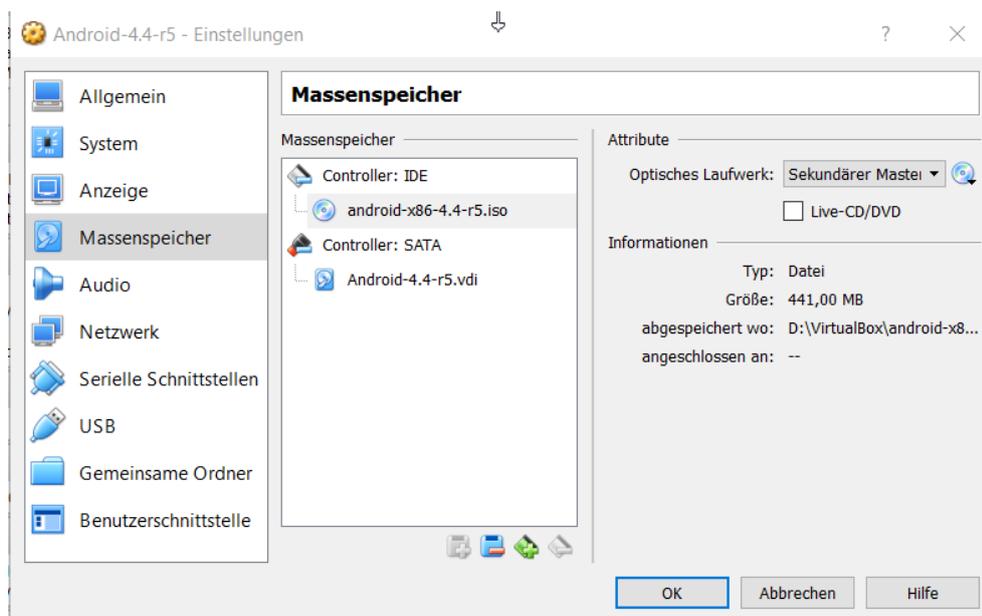
Danach wird die Festplatte lokal auf dem Computer angelegt.



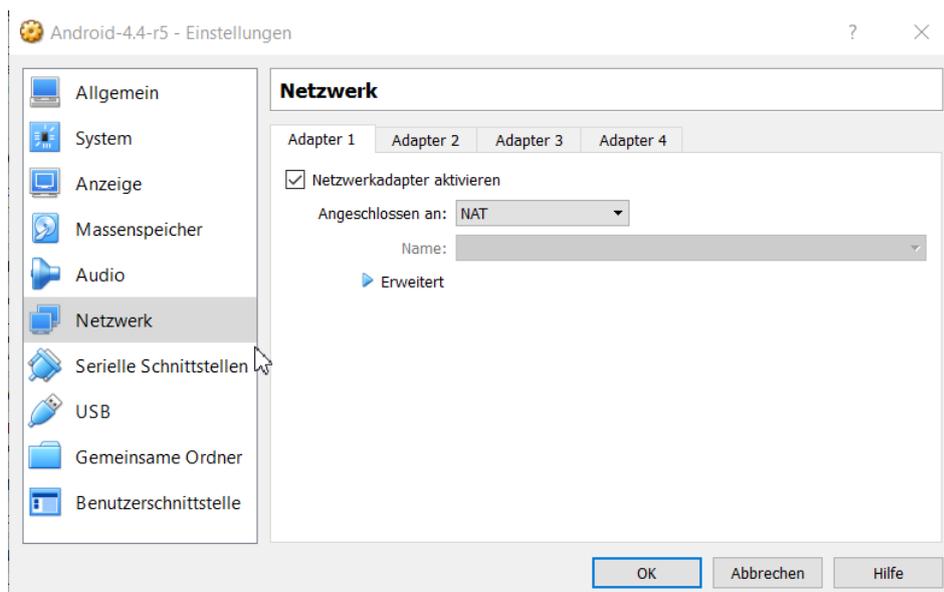
Bei den Grundeinstellungen der Virtuellen Maschine, nehme ich nun im Reiter **System** das Häkchen für das **Disketten Laufwerk** raus und verschiebe es unter die Festplatte.



Bei der Einstellung der Anzeige lege ich 32 MB für den Grafikspeicher fest.



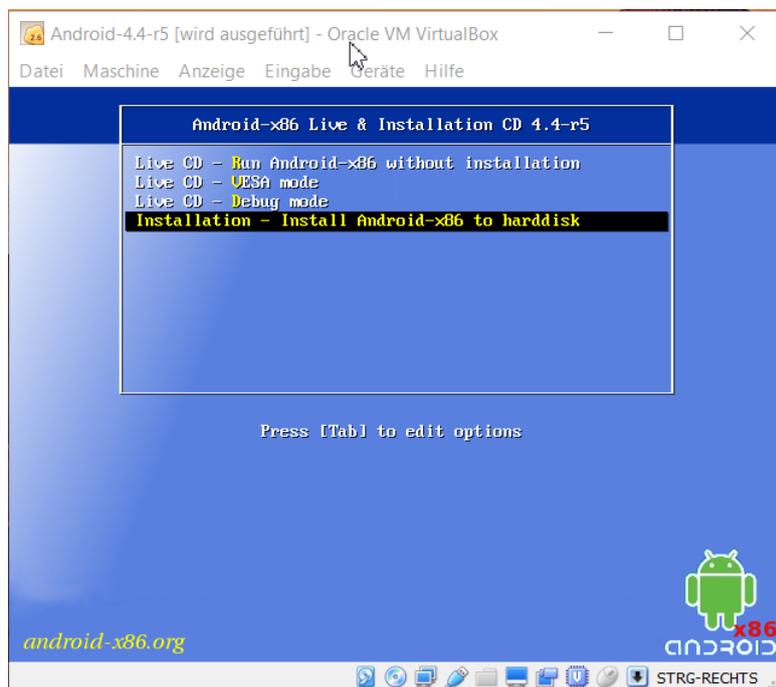
Nachdem das Android ISO Image heruntergeladen wurde, hänge ich das ISO Image hier im CDROM Laufwerk ein.



Die Netzwerkeinstellung lasse ich für die Installation auf **NAT** stehen, später kann man diese Einstellung auf **Netzwerkbrücke** umstellen.

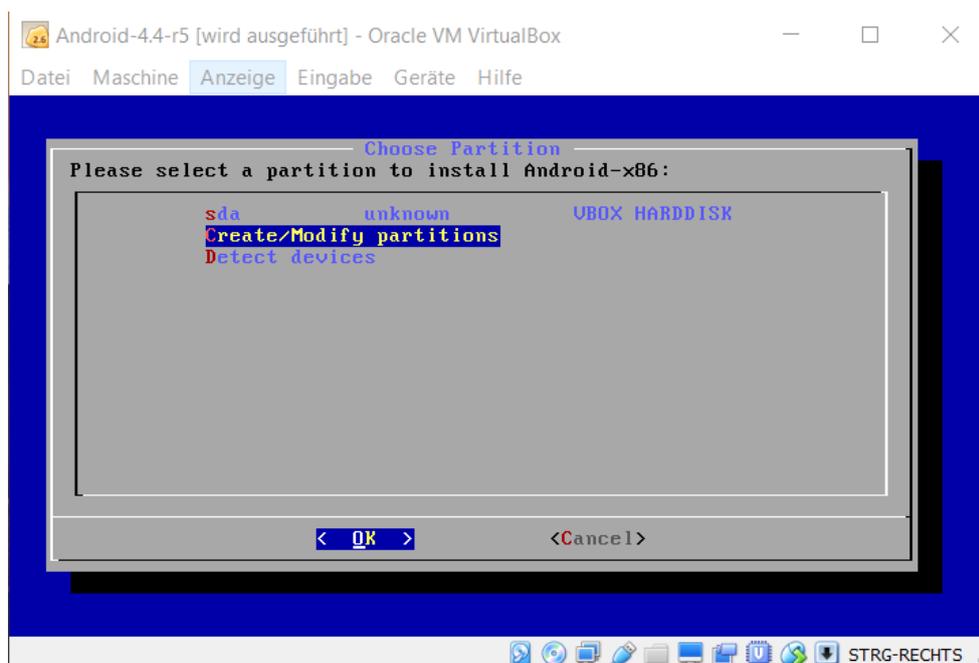
- ↳ NAT -- bekommt eine interne Virtuelle IP-Adresse zugewiesen.
- ↳ Netzwerkbrücke – nutzt einen internen DHCP Server, wenn vorhanden.

Android 4.4 auf die Festplatte installieren



Nachdem nun die Virtuelle Maschine angelegt und gestartet wurde, klicke ich mit der Maus in die Virtuelle Maschine und wähle den letzte Eintrag aus „**Install Android-x86 to harddisk**“.

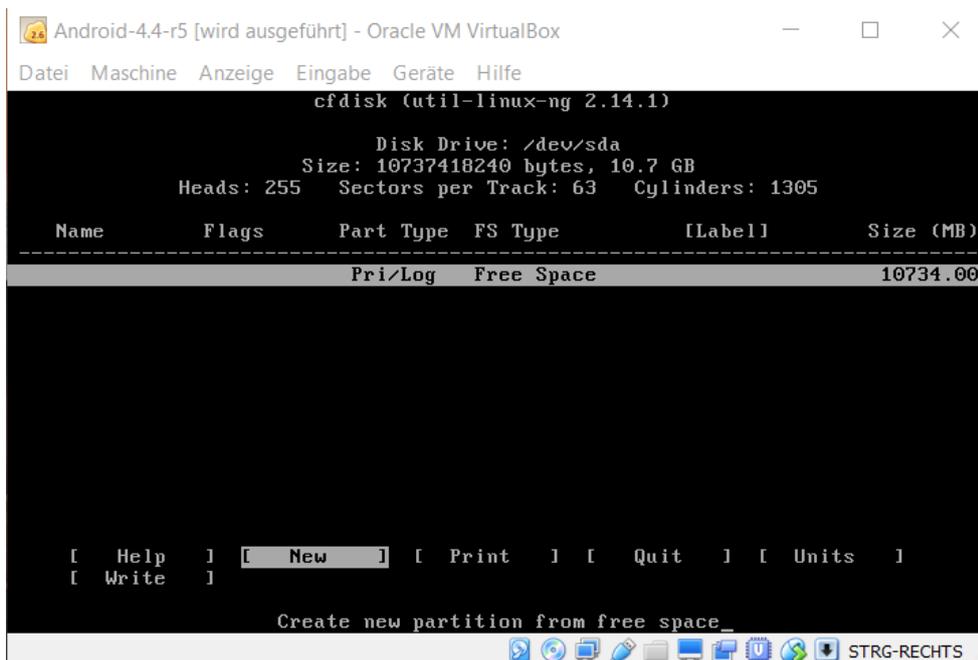
Kann sein das die Virtuelle Maschine die Maus fängt, dieses Maus fangen kann man lösen, in dem man auf der Tastatur die **Rechte STRG Taste** drückt.



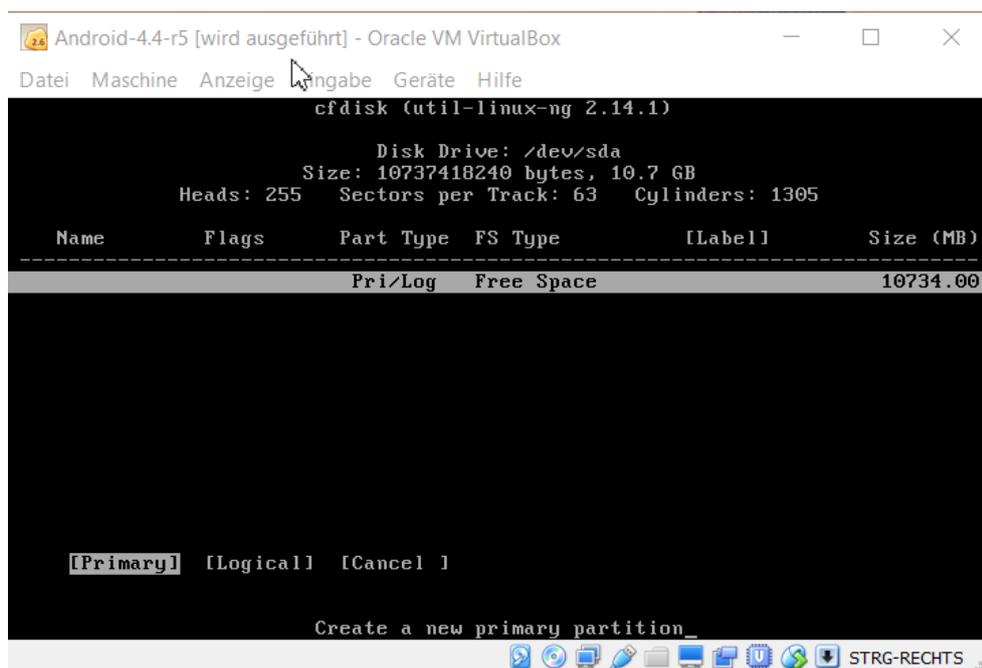
Nachdem das ISO Image in den Arbeitsspeicher geladen wurde, wählen wir mit der Pfeiltaste oben/unten den Eintrag **Create/Modify partitions** und klicken weiter mit der Enter Taste.



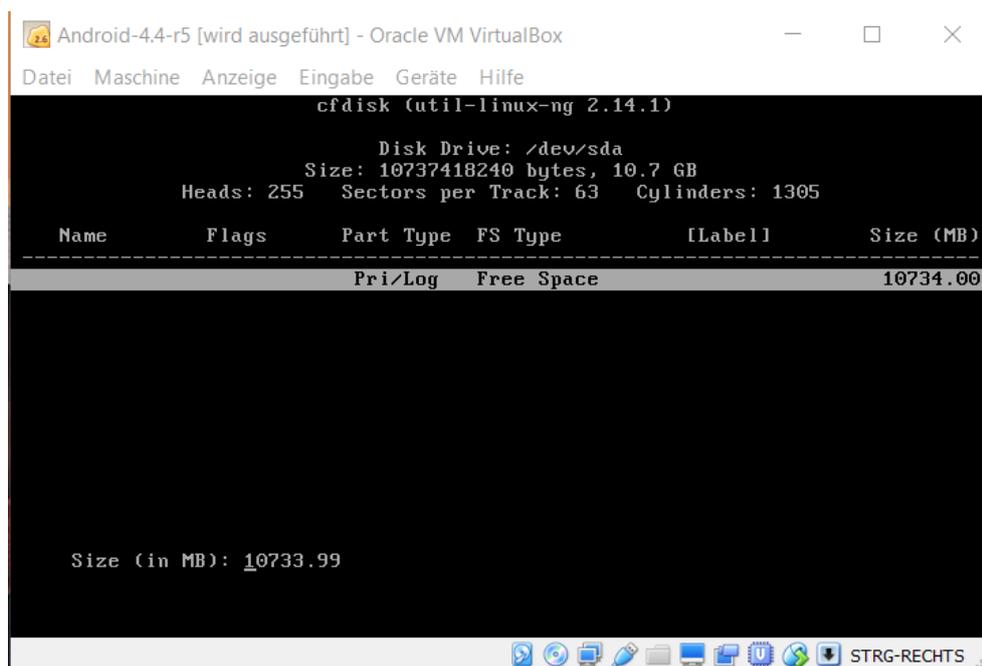
Wir wollen keinen GPT Datenträger anlegen, wählen **No** und klicken mit Enter weiter.



Mit der Pfeiltaste links/rechts wählen wir den Eintrag **New** aus und klicken wieder auf die Enter Taste um weiter zu kommen.



Bestätigen den Eintrag **Primary** mit Enter.



Wir wollen die ganzen **10 GB** Festplatte für die Installation nutzen und klicken hier ebenfalls auf Enter.

```

Android-4.4-r5 [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe
cfdisk (util-linux-ng 2.14.1)

Disk Drive: /dev/sda
Size: 10737418240 bytes, 10.7 GB
Heads: 255 Sectors per Track: 63 Cylinders: 1305

Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
sda1      Primary   Linux      10734.00

[ Bootable ] [ Delete ] [ Help ] [ Maximize ] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Toggle bootable flag of the current partition_
  
```

Wir springen in das nächste Fenster und wählen den Eintrag **Bootable** mit Enter aus.

```

Android-4.4-r5 [wird ausgeführt] - Oracle VM VirtualBox
Datei Maschine Anzeige Eingabe Geräte Hilfe
cfdisk (util-linux-ng 2.14.1)

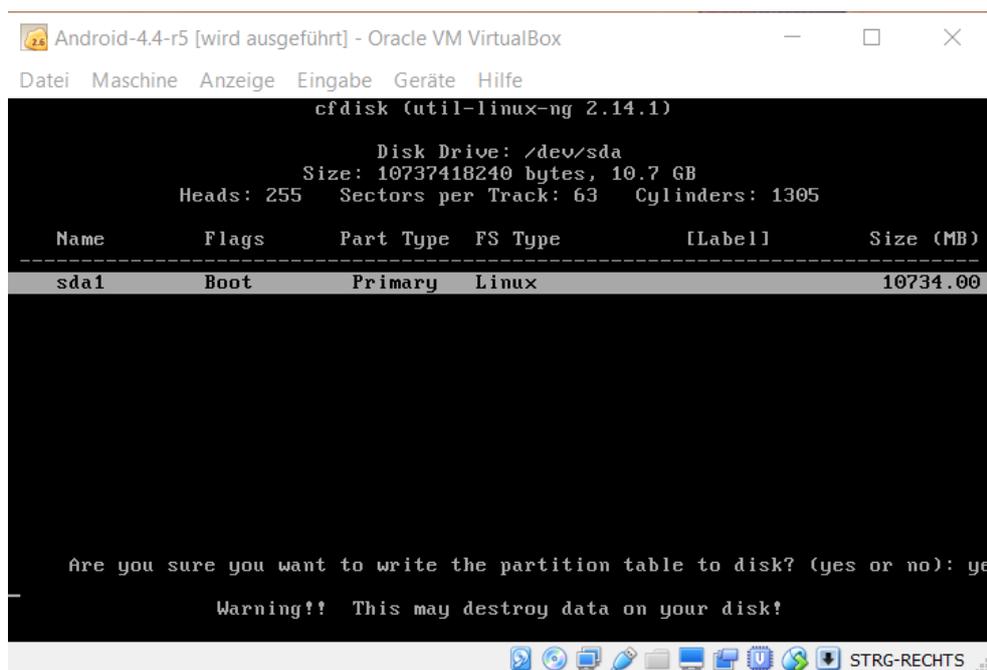
Disk Drive: /dev/sda
Size: 10737418240 bytes, 10.7 GB
Heads: 255 Sectors per Track: 63 Cylinders: 1305

Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
sda1      Boot       Primary   Linux      10734.00

[ Bootable ] [ Delete ] [ Help ] [ Maximize ] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Write partition table to disk (this might destroy data)_
  
```

Nach dem Bestätigen mit der Enter Taste sollte hier unter Flags der Eintrag **Boot** erscheinen, danach klicken wir mit der Pfeiltaste links/rechts soweit bis wir auf den Punkt **Write** stehen und bestätigen auch diese Auswahl mit der Enter Taste.



```

cfdisk (util-linux-ng 2.14.1)

Disk Drive: /dev/sda
Size: 10737418240 bytes, 10.7 GB
Heads: 255 Sectors per Track: 63 Cylinders: 1305

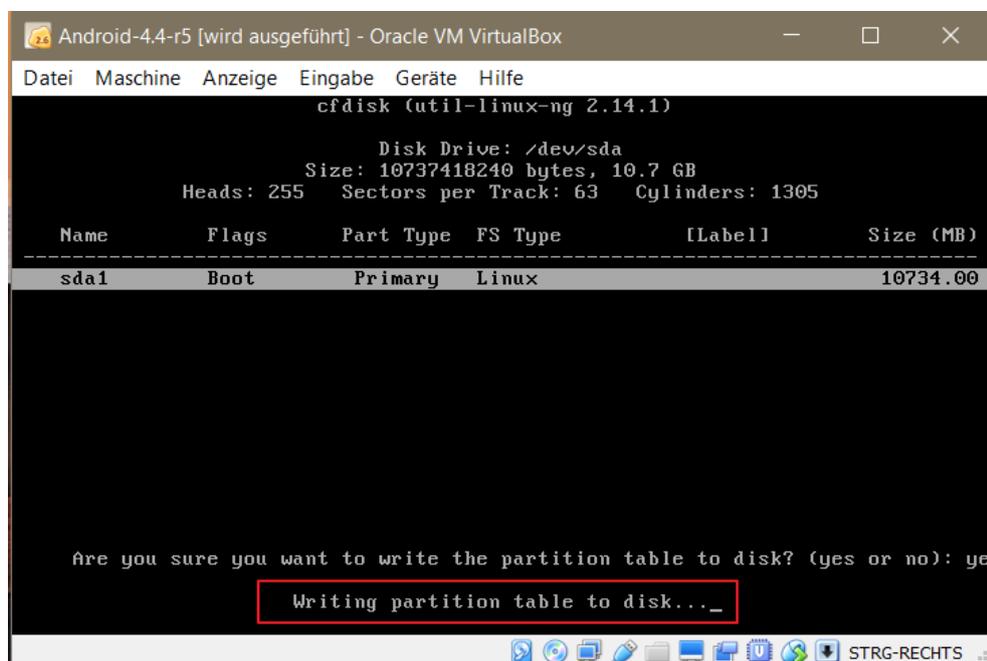
Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
sda1     Boot      Primary   Linux      [Label]    10734.00

Are you sure you want to write the partition table to disk? (yes or no): ye

Warning!! This may destroy data on your disk!
  
```

Nun wird man gefragt ob man die Informationen auf die Festplatte schreiben möchte, dies bestätigen wir in dem wir **yes** eintippen und mit Enter bestätigen.

Das **y** ist natürlich auf der deutschen Tastatur das **z**.



```

cfdisk (util-linux-ng 2.14.1)

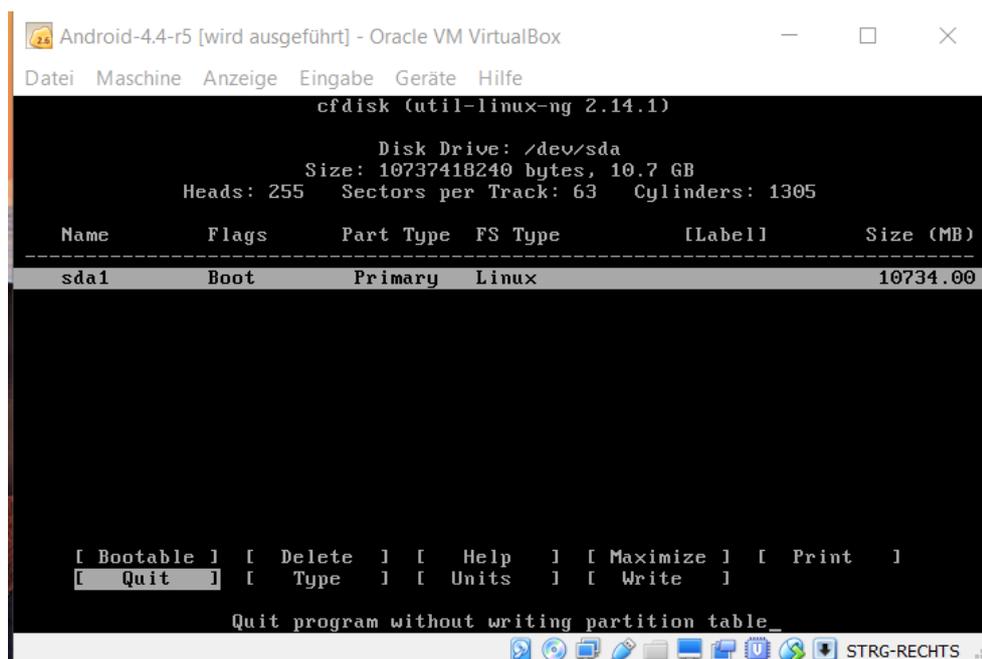
Disk Drive: /dev/sda
Size: 10737418240 bytes, 10.7 GB
Heads: 255 Sectors per Track: 63 Cylinders: 1305

Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
sda1     Boot      Primary   Linux      [Label]    10734.00

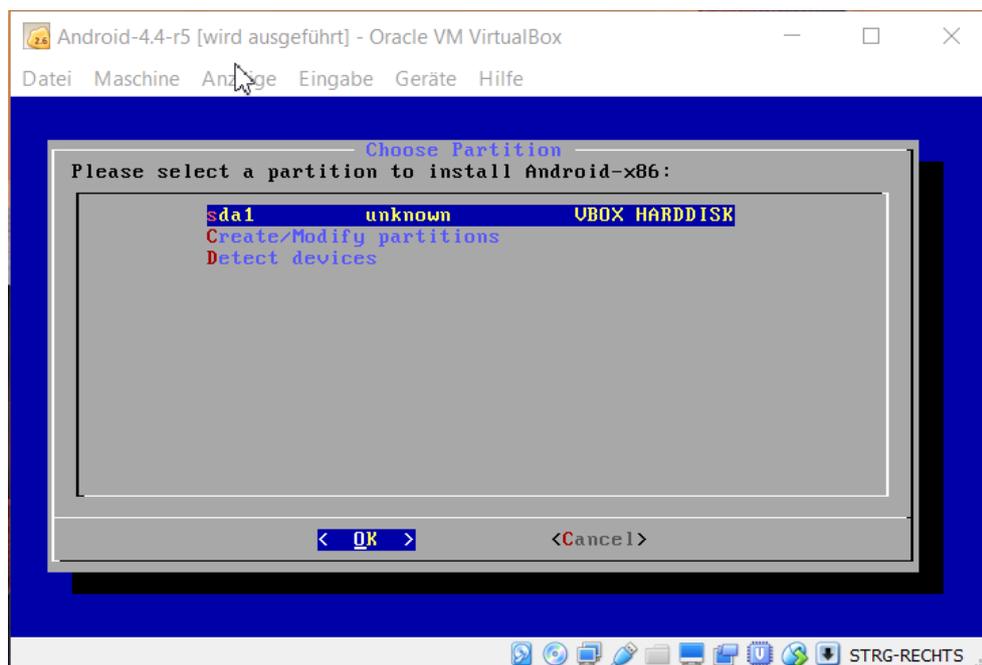
Are you sure you want to write the partition table to disk? (yes or no): ye

Writing partition table to disk..._
  
```

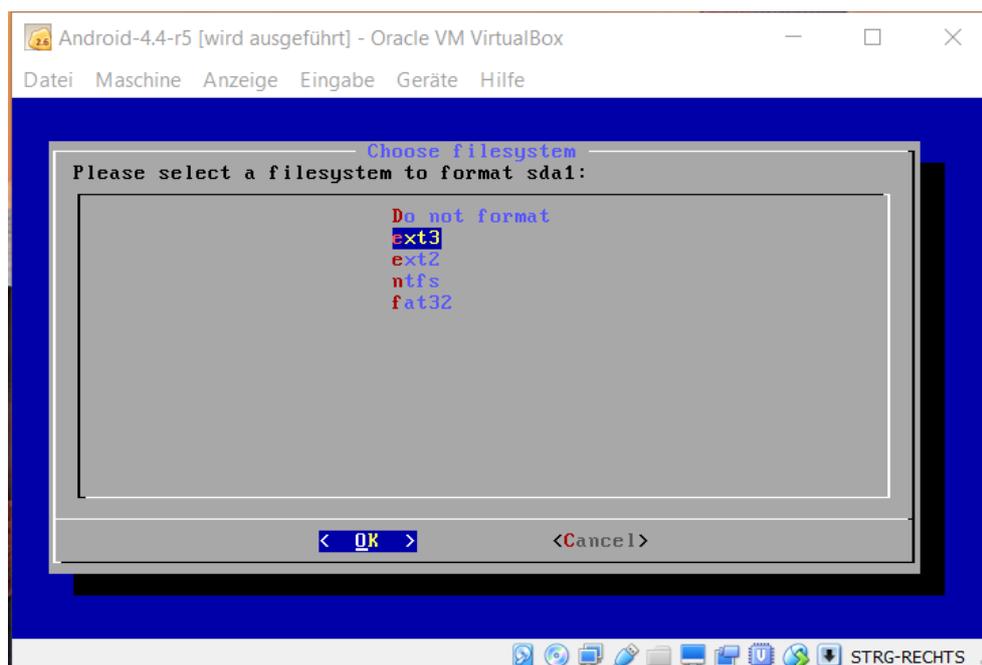
Nun werden die Information auf die Festplatte in die Partitionstabelle geschrieben.



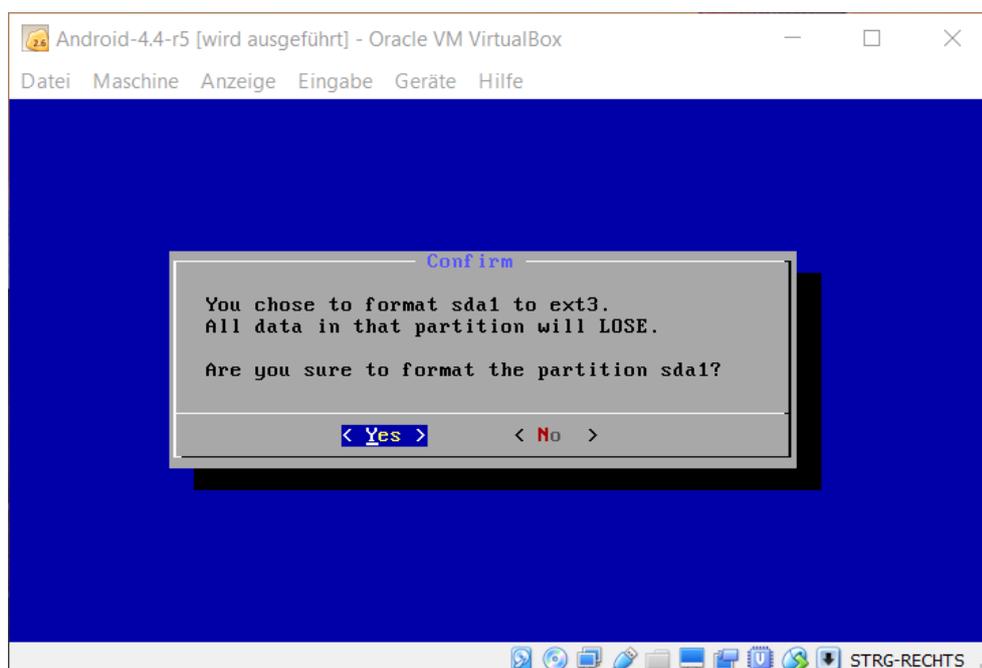
Nachdem die Informationen geschrieben wurden, klicken wir wieder mit der Pfeiltaste links/rechts auf den Eintrag **Quit** und anschließend die Enter Taste.



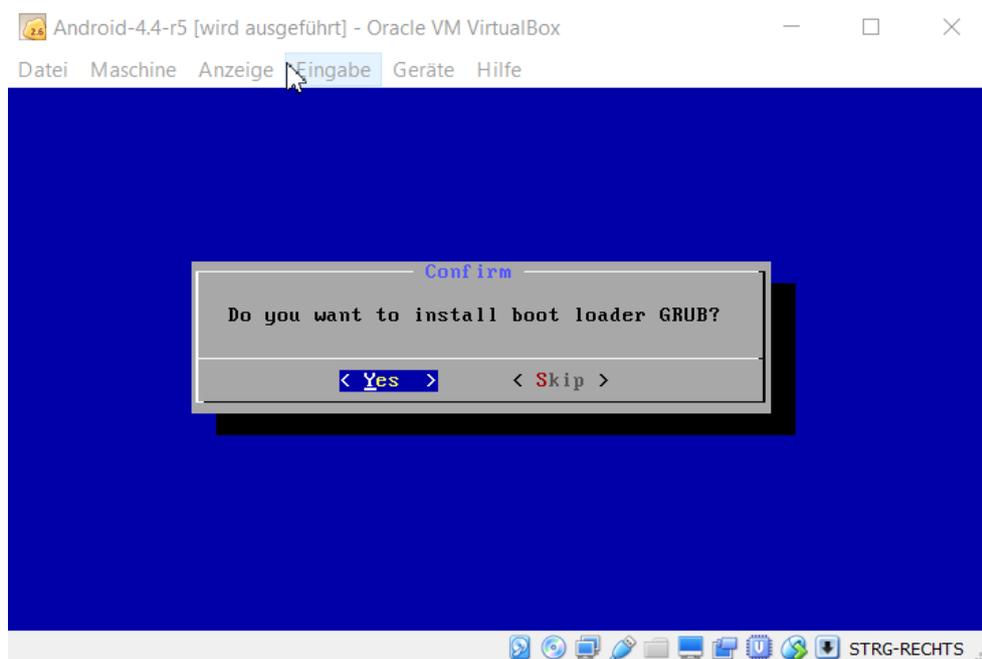
Nun starten wir die Installation von **Android 4.4**, hier wählen wir unsere neu erstellte Festplatte aus und bestätigen mit Enter.



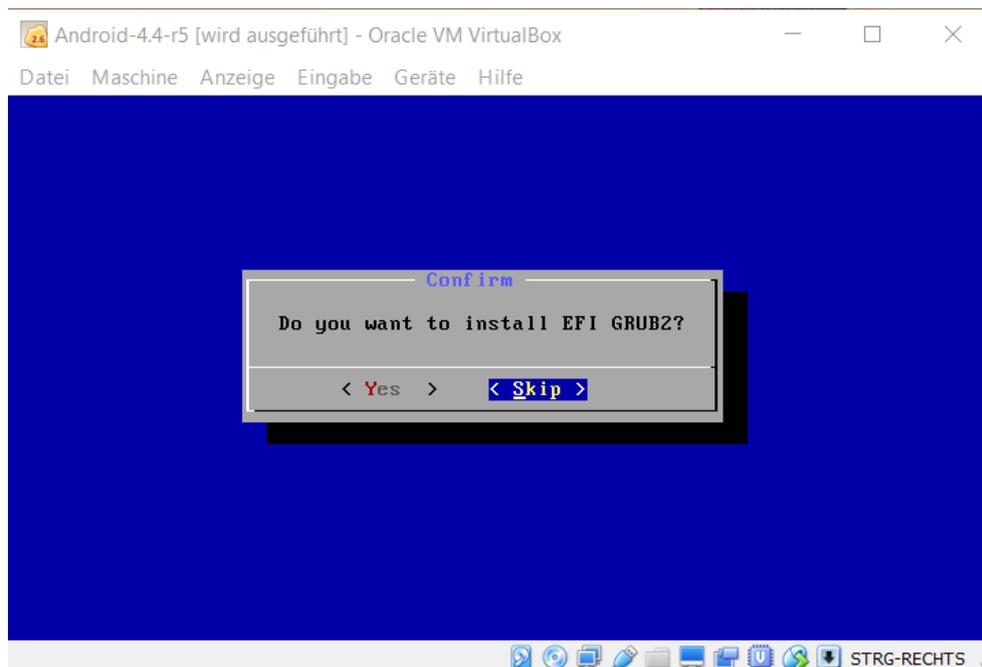
Wählen mit der Pfeiltaste oben/unten das Formatierungsformat **ext3** aus und klicken auf Enter.



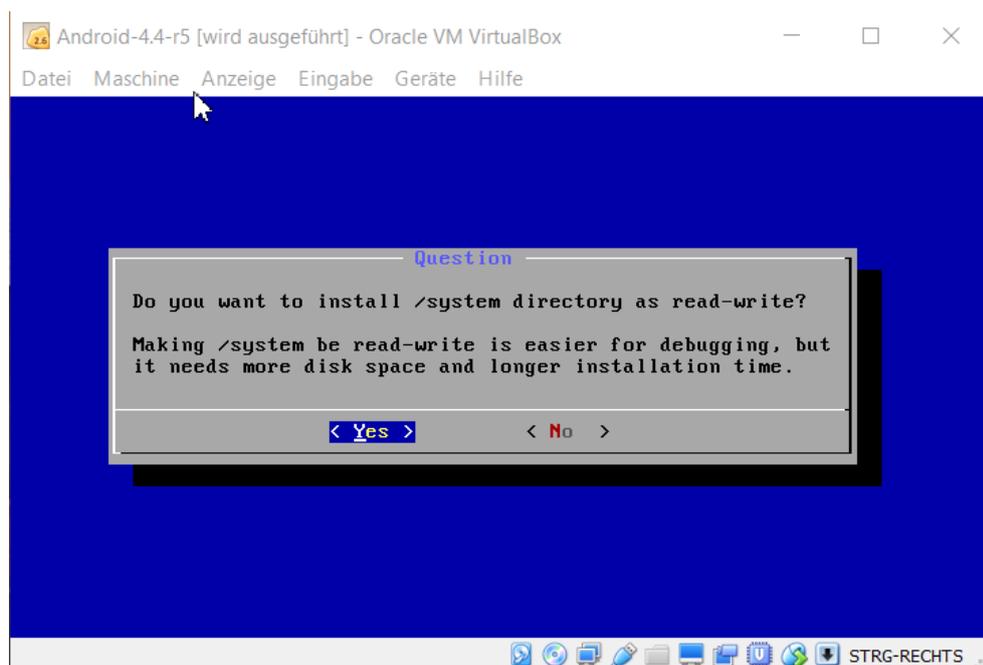
Im nächsten Schritt wählen wir **Yes**/Enter aus um die Festplatte zu formatieren.



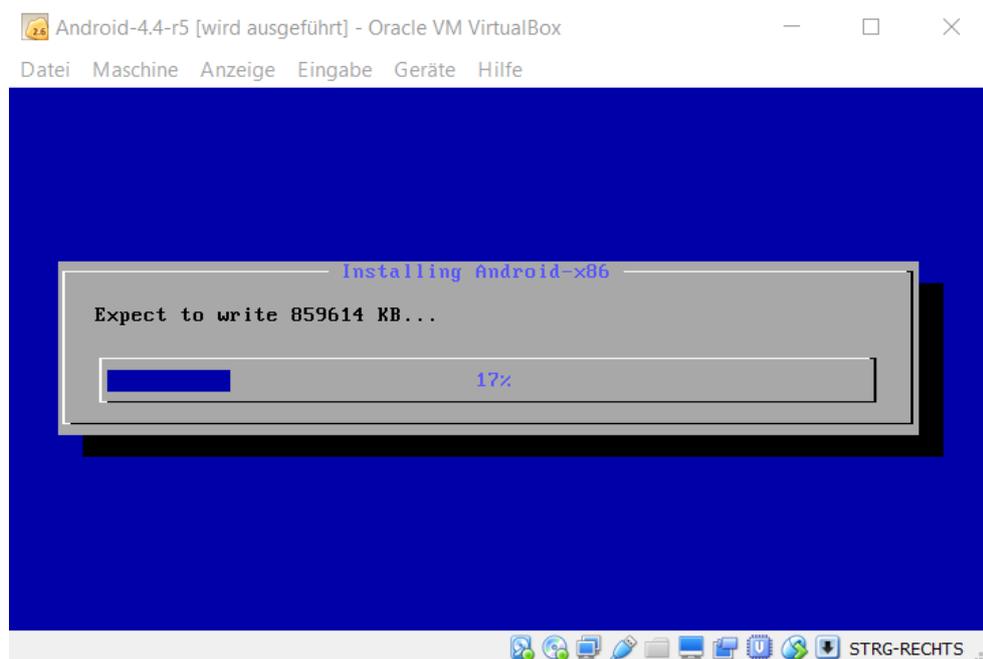
Auch hier wählen wir **Yes**/Enter aus um einen **Bootloader GRUB** zu installieren.



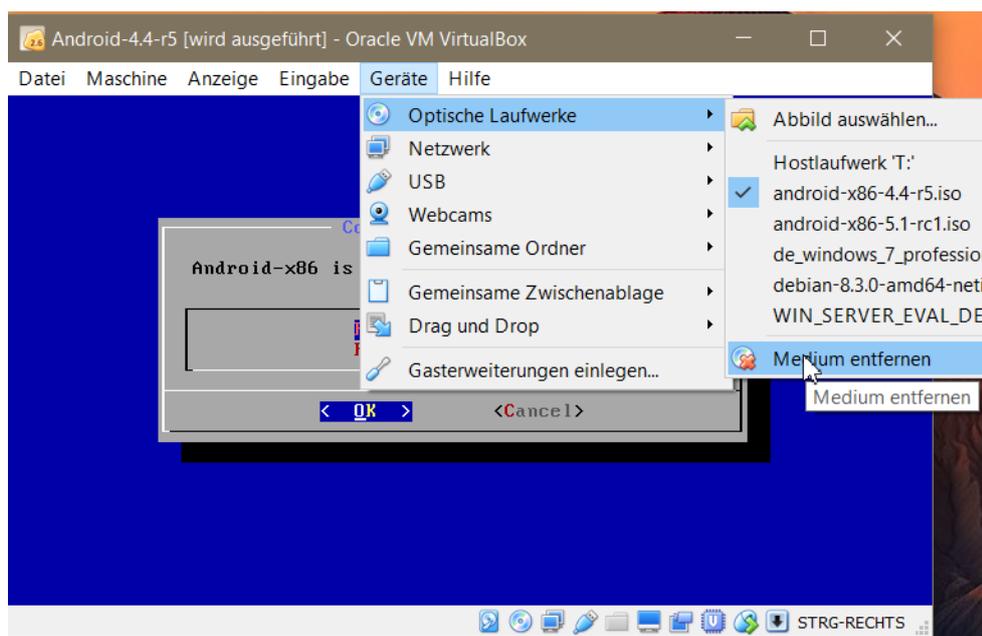
Wir wollen keinen Secureboot mit UEFI, daher überspringen wir diesen Part mit Skip.



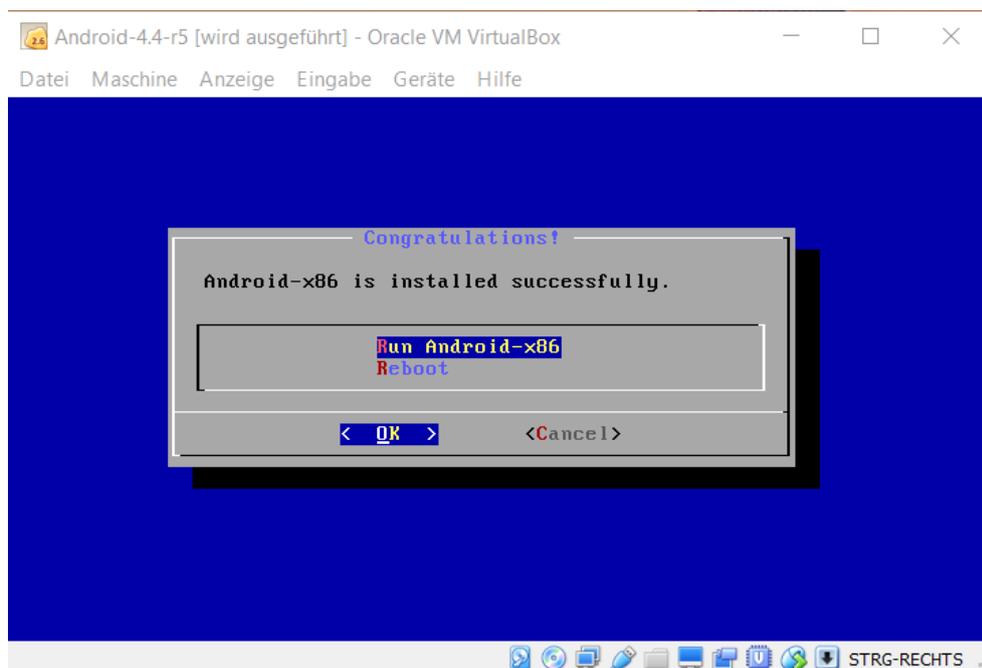
Endlich geschafft, mit **Yes**/Enter installieren wir Android 4.4 auf die Festplatte.



Dieser Schritt dauert keine Minute, dann haben wir ein lauffähiges Android System.

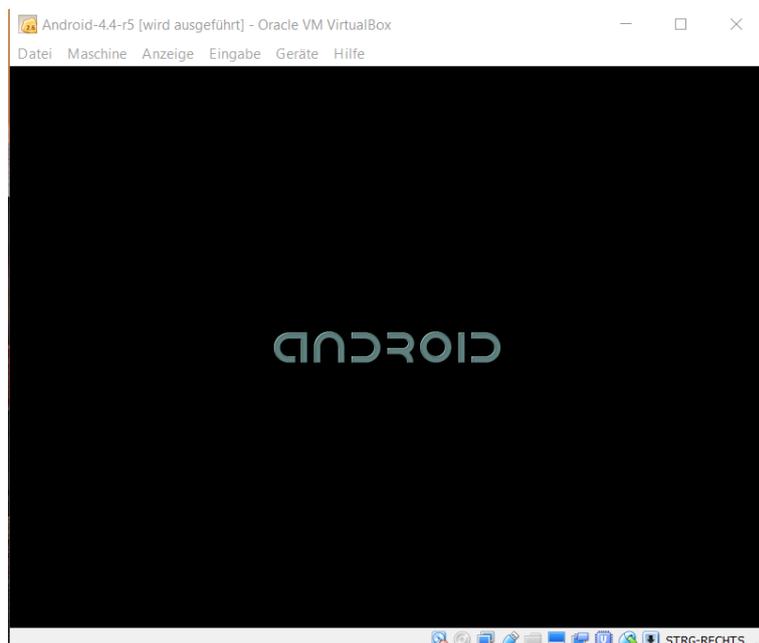


Bevor wir das Android starten können, entfernen wir noch das ISO Images aus dem CDROM Laufwerk, wenn Ihr hier eine Meldung bekommt, dann klickt auf „Medium entfernen - erzwingen“

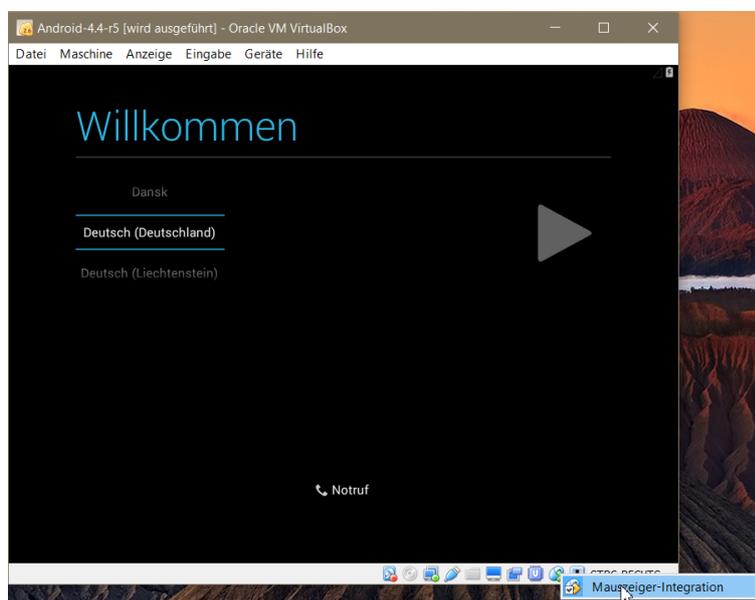


Nun endlich können wir das Android System mit Enter starten.

Android 4.4 starten und erste Einstellungen vornehmen

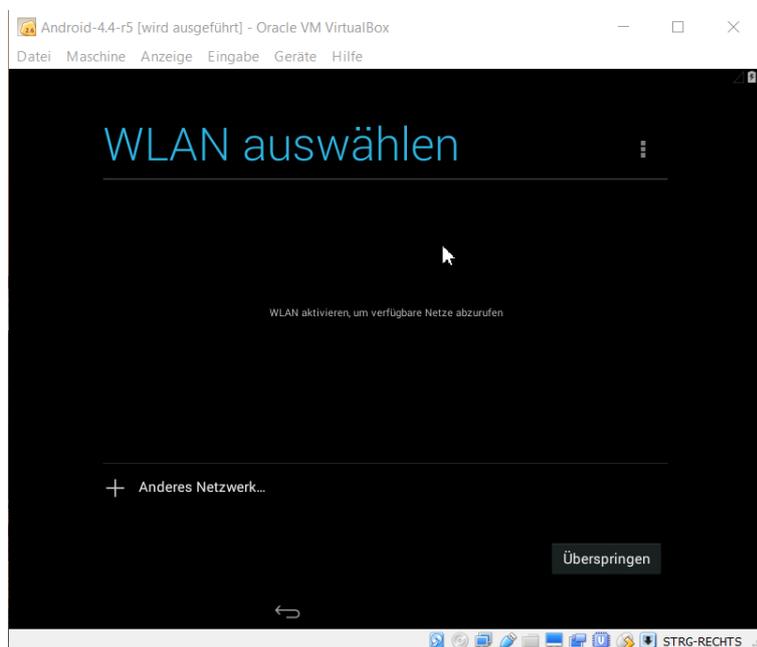


Dieser Schritt könnte zwischen 30 Sekunden und 1 Minuten dauern bis das Betriebssystem vollständig geladen wurde.

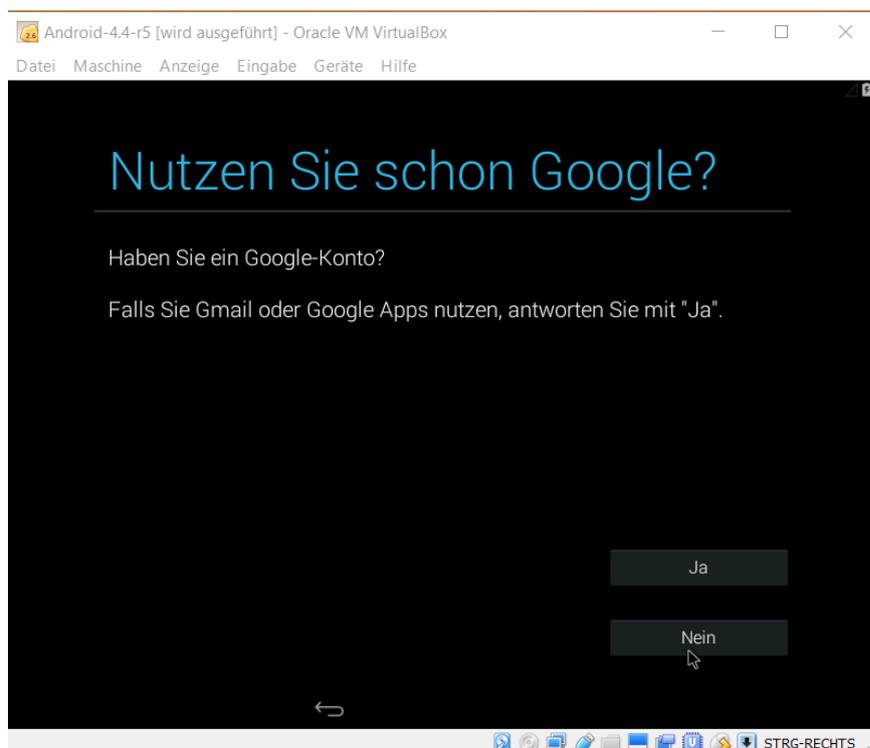


Um nun mit der Oberfläche in der Virtuellen Maschine arbeiten zu können muss hier die Mausintegration deaktiviert werden, rechte Maustaste auf das kleine Maussymbol und den Eintrag Mauszeiger - Integration anklicken. Nun kann man in die Virtuelle Maschine reinklicken und die Oberfläche mit der Pfeiltaste oben/unten und der Maus bedienen, wenn man diese Oberfläche wieder verlassen möchte, die **rechte STRG Taste** würde die Maus wieder frei geben. **Bitte beachten, jedes Mal wenn die Virtuelle Maschine ausgeschaltet wurde und neu gestartet wurde, muss diese Mausintegration deaktiviert werden.**

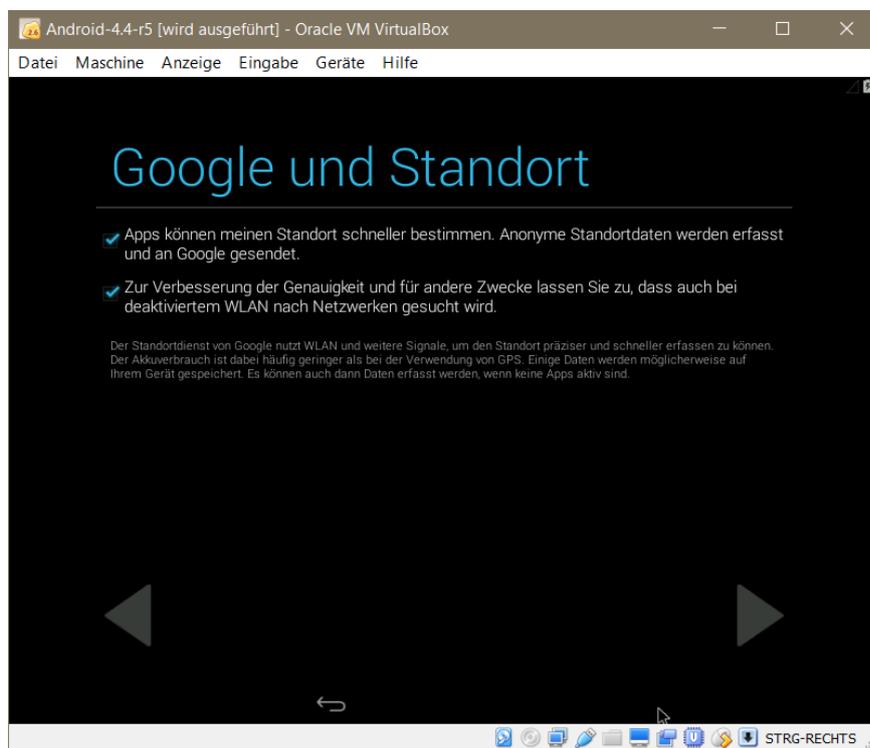
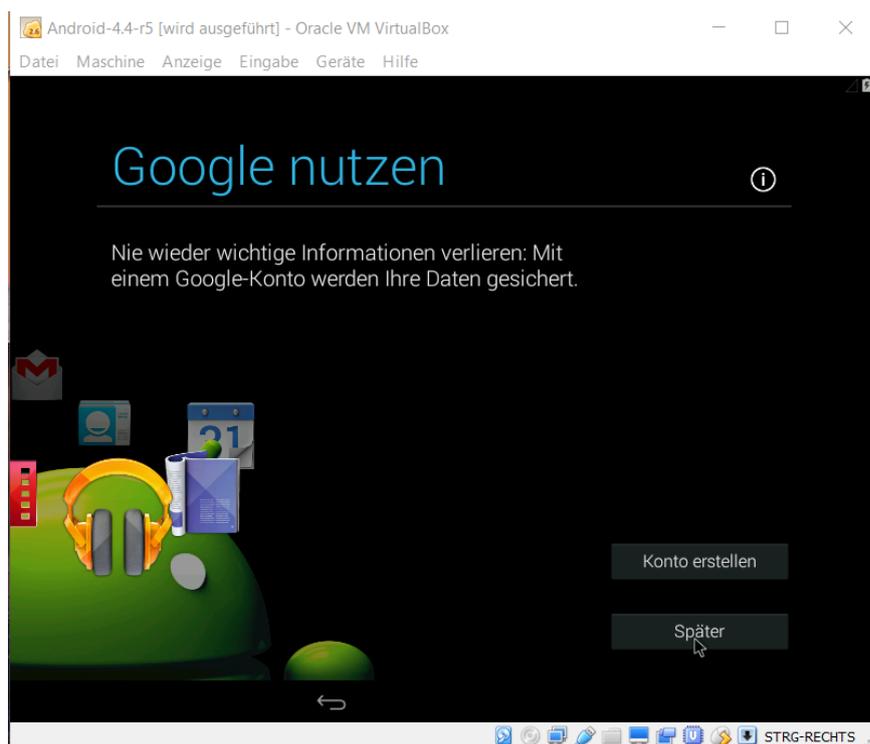
Nun wählen wir Deutsch (Deutschland) als Sprache aus und mit der Maustaste klicken wir den rechten Pfeil an um weiter zu kommen.

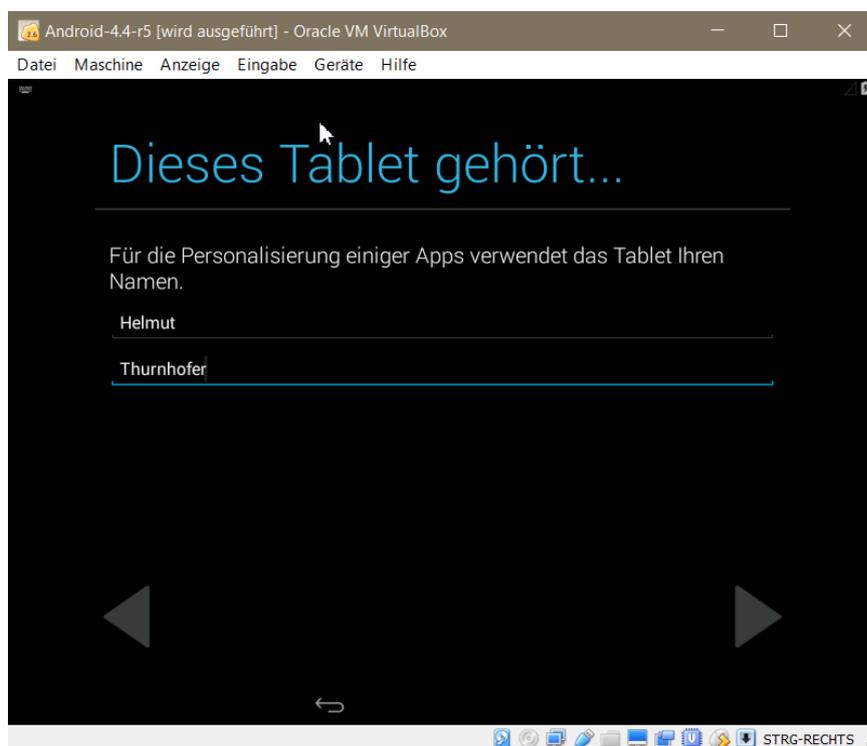
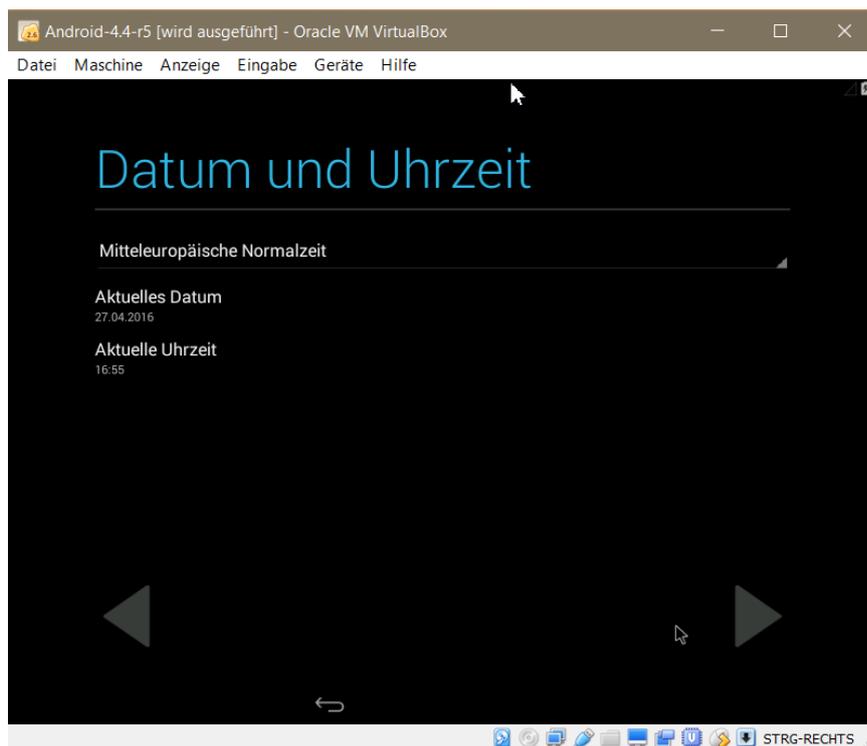


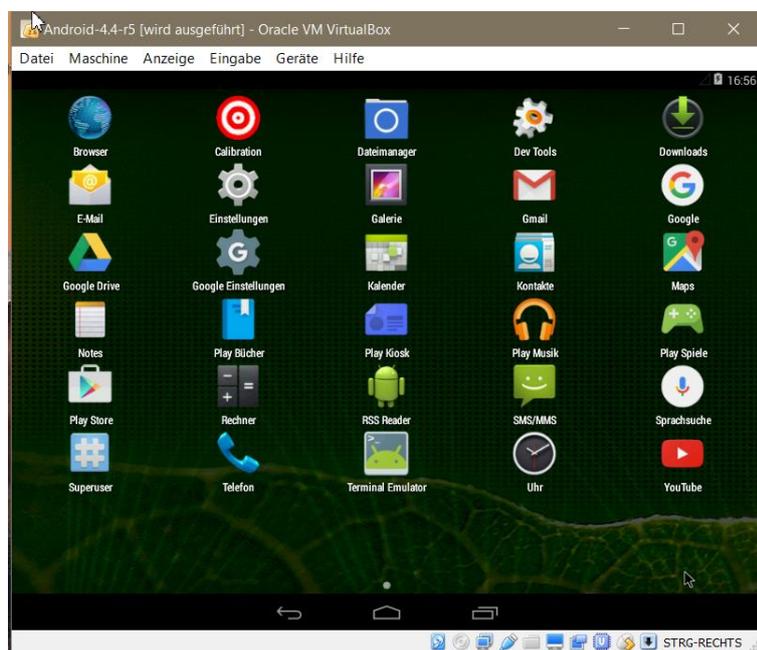
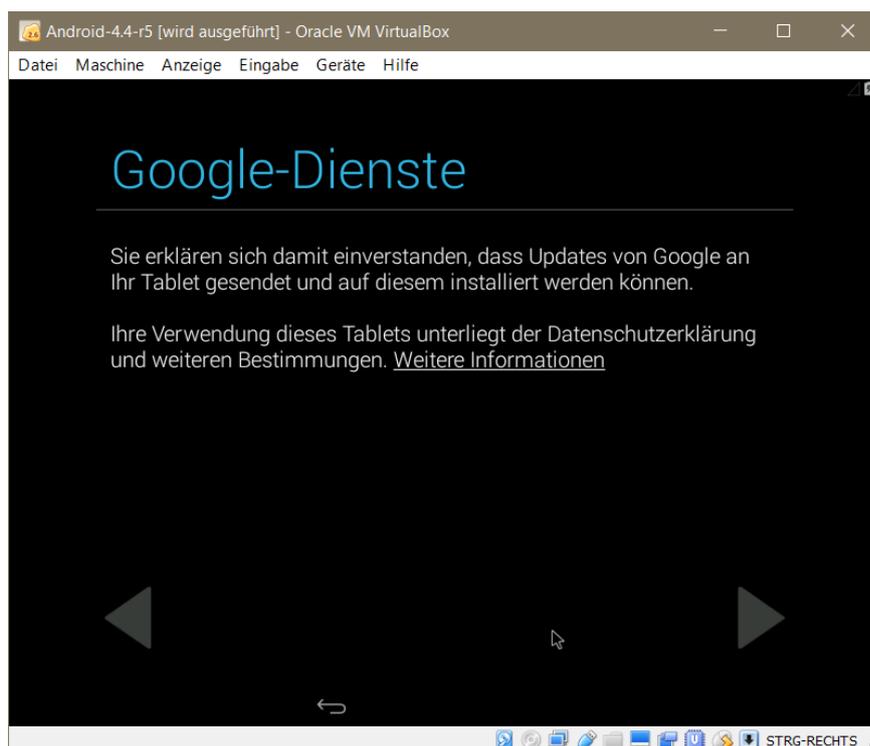
Die WLAN Einstellungen überspringen wir, wir haben kein WLAN, das ist eine Virtuelle Maschine. 😊



Im den nachfolgenden Screenshots richtet man sich sein Google Konto ein, um sich aus dem Google Play Store Apps herunterladen zu können.



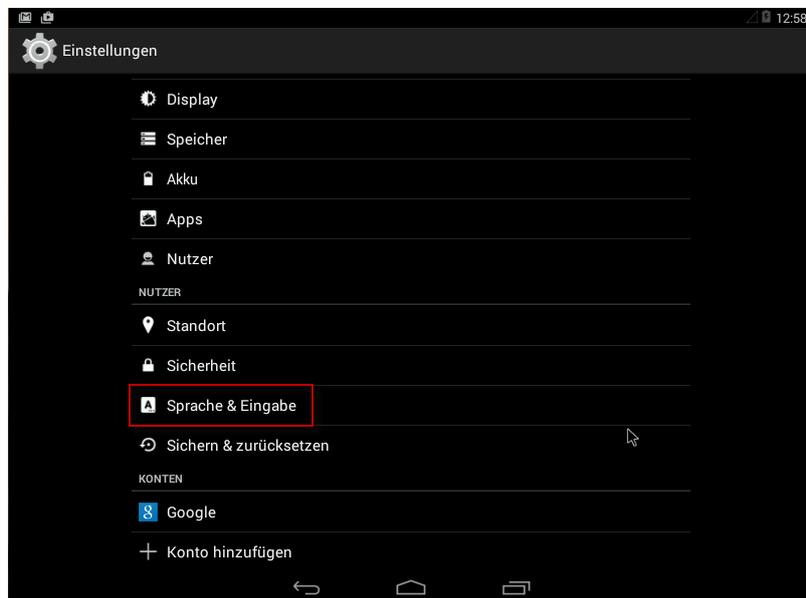




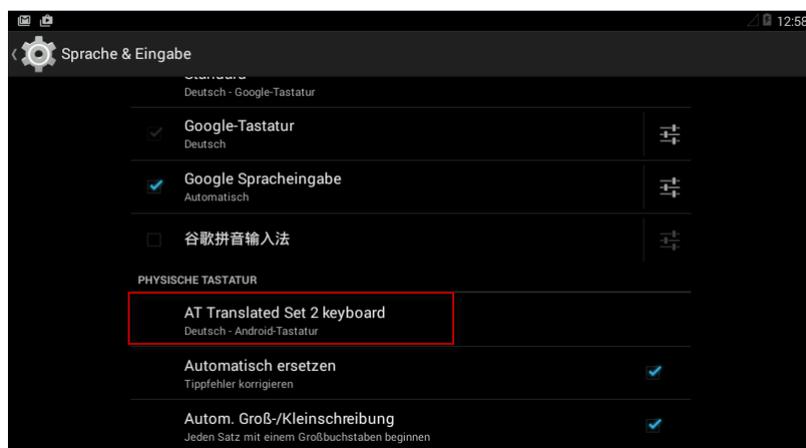
So wenn man nun alle Einstellungen gemacht hat, sieht man irgendwann diesen Desktop, wow sieht fast so aus wie auf mein Handy. 😊

Android 4.4 Grundkonfiguration

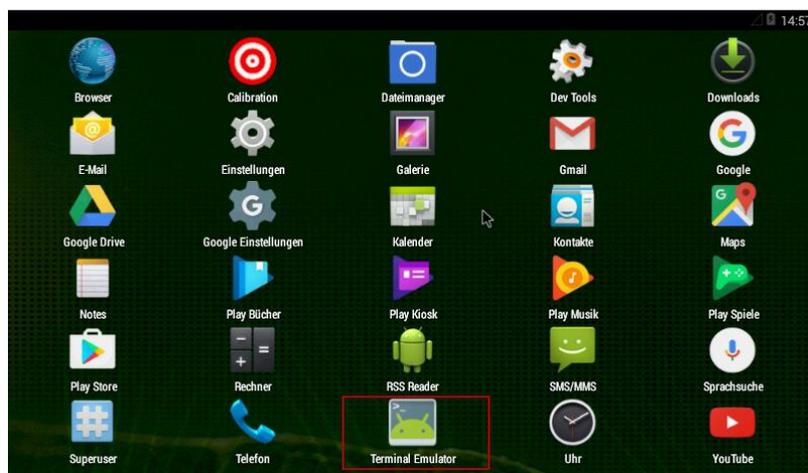
Um nun mit Android in der Virtuellen Maschine anständig arbeiten zu können habe ich noch ein paar Anpassungen gemacht.



Um in der Konsole das Deutsche Tastaturlayout benutzen zu können, muss noch eine Einstellung konfiguriert werden. Dazu klickt man auf **Einstellungen** → **Sprache & Eingabe**

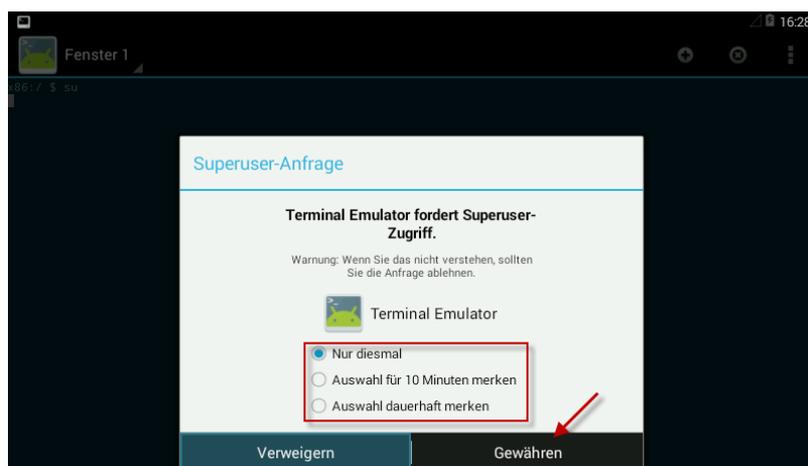


Im nächsten Fenster scrollt man ein bisschen nach unten und wählt die Physische Tastatur aus hier habe ich die **Deutsch –Android Tastatur** gewählt.



Im nächsten Schritt möchte ich gerne für die Dokumentation die Bildschirmauflösung von 800x600 Pixel in 1024x768 Pixel verändern, dazu öffne ich einen Terminal Emulator und mache mich zum root Benutzer und gewähre den Zugriff für die Sitzung.

su



Danach starte ich die Virtuelle Maschine neu mit dem Befehl

reboot



```
GNU GRUB version 0.97 (639K lower / 1047488K upper memory)

Android-x86 4.4-r5
Android-x86 4.4-r5 (Debug mode)
Android-x86 4.4-r5 (Debug nomodeset)
Android-x86 4.4-r5 (Debug video=LVD3-1:d)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
before booting, or 'c' for a command-line.
```

Nachdem die Virtuelle Maschine neu startet, muss man schnell sein und mit der Pfeiltaste oben/unten die Auswahl zu verändern um den Timer von 5 Sekunden anzuhalten.

Danach wähle ich „e“ für den edit Modus aus.

```
GNU GRUB version 0.97 (639K lower / 1047488K upper memory)

kernel /android-4.4-r5/kernel quiet root=/dev/ram0 androidboot.hardware=
initrd /android-4.4-r5/initrd.img

Use the ↑ and ↓ keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.
```

Danach nochmal den Buchstaben „e“ um die Kommandozeile anzupassen.

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time cancels. ENTER
at any time accepts your changes.]

<dboot.hardware=android_x86 SRC=/android-4.4-r5 vga=ask
```

Jetzt schreibe ich an das Ende **vga=ask** hin und klicke auf Enter um zur vorherigen Ansicht zu gelangen.

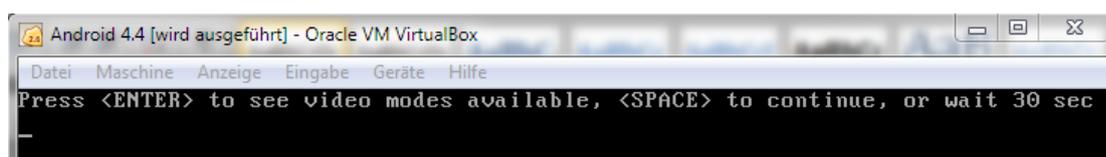
```

GNU GRUB version 0.97 (639K lower / 1047488K upper memory)

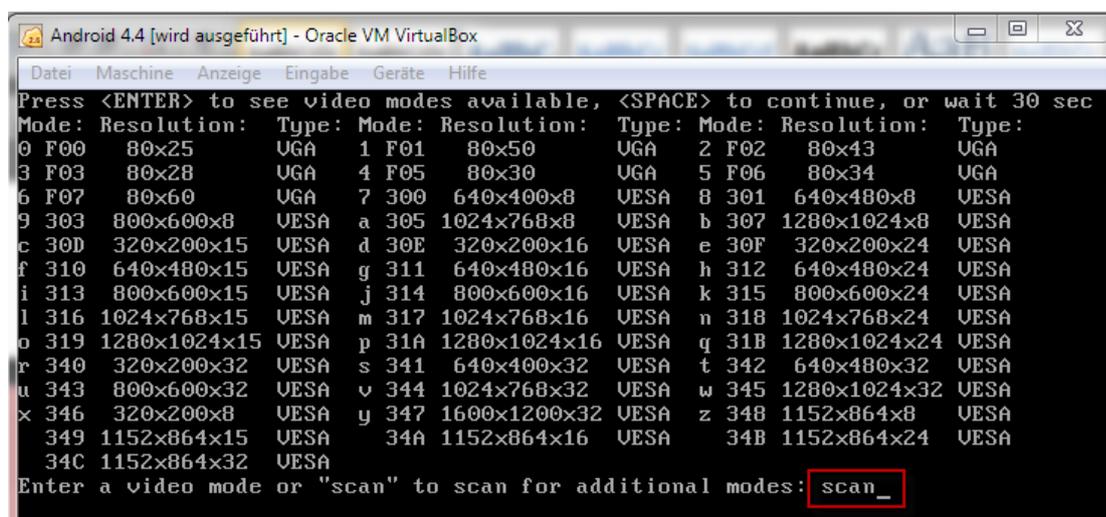
kernel /android-4.4-r5/kernel quiet root=/dev/ram0 androidboot.hardware=
initrd /android-4.4-r5/initrd.img
  
```

Use the ↑ and ↓ keys to select which entry is highlighted.
 Press 'b' to boot, 'e' to edit the selected command in the boot sequence, 'c' for a command-line, 'o' to open a new line after ('O' for before) the selected line, 'd' to remove the selected line, or escape to go back to the main menu.

Um nun den Kernel zu booten tippe ich den Buchstaben „b“ für boot.



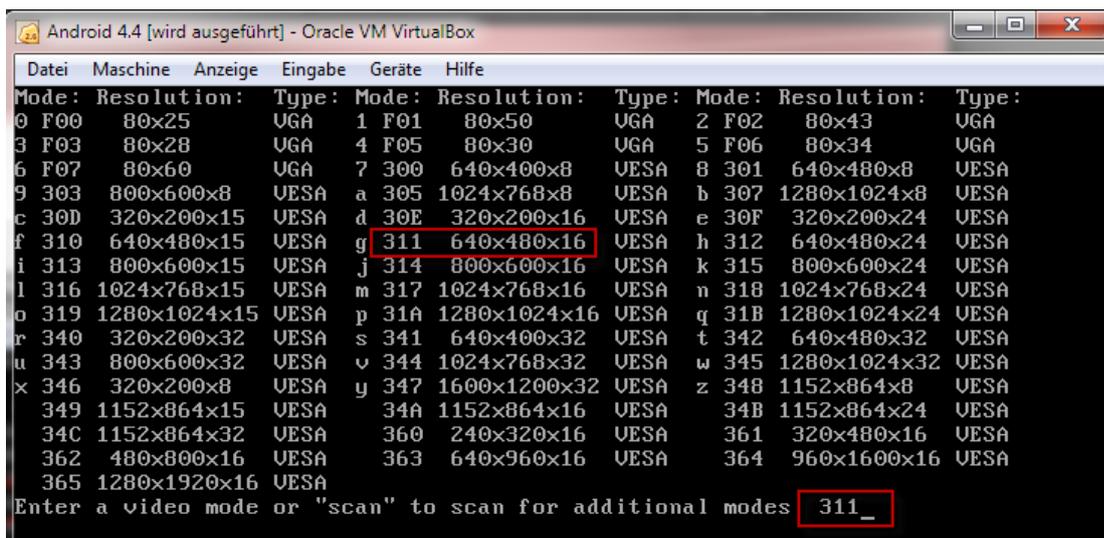
Danach greift der Befehl vga=ask und man kann sehen welche Auflösungen unterstützt werden, hier Enter klicken um eine Auswahl zu erhalten.



Mode:	Resolution:	Type:	Mode:	Resolution:	Type:	Mode:	Resolution:	Type:
0 F00	80x25	UGA	1 F01	80x50	UGA	2 F02	80x43	UGA
3 F03	80x28	UGA	4 F05	80x30	UGA	5 F06	80x34	UGA
6 F07	80x60	UGA	7 300	640x400x8	VESA	8 301	640x480x8	VESA
9 303	800x600x8	VESA	a 305	1024x768x8	VESA	b 307	1280x1024x8	VESA
c 30D	320x200x15	VESA	d 30E	320x200x16	VESA	e 30F	320x200x24	VESA
f 310	640x480x15	VESA	g 311	640x480x16	VESA	h 312	640x480x24	VESA
i 313	800x600x15	VESA	j 314	800x600x16	VESA	k 315	800x600x24	VESA
l 316	1024x768x15	VESA	m 317	1024x768x16	VESA	n 318	1024x768x24	VESA
o 319	1280x1024x15	VESA	p 31A	1280x1024x16	VESA	q 31B	1280x1024x24	VESA
r 340	320x200x32	VESA	s 341	640x400x32	VESA	t 342	640x480x32	VESA
u 343	800x600x32	VESA	v 344	1024x768x32	VESA	w 345	1280x1024x32	VESA
x 346	320x200x8	VESA	y 347	1600x1200x32	VESA	z 348	1152x864x8	VESA
349	1152x864x15	VESA	34A	1152x864x16	VESA	34B	1152x864x24	VESA
34C	1152x864x32	VESA						

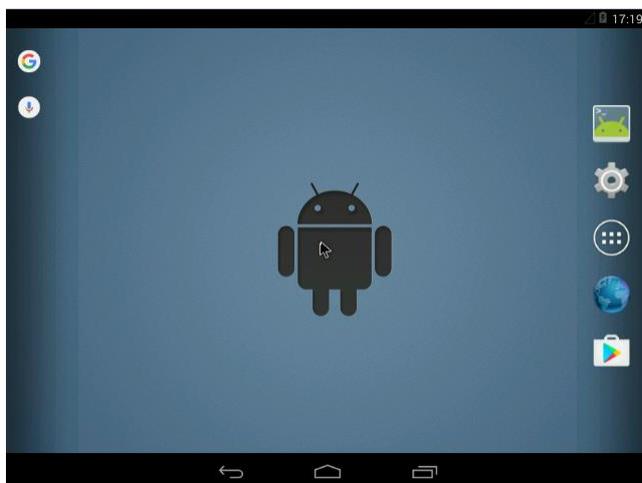
Enter a video mode or "scan" to scan for additional modes: scan_

Um jetzt für seine Umgebung die richtige Auflösung zu finden gibt es noch den Befehl „scan“

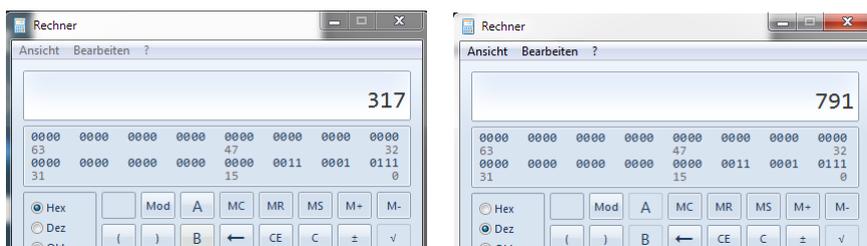


Nun bekommt man eine schöne Liste, welche Auflösungen funktionieren würden. Unter Mode findet man die Hexadezimalzahl, diese muss später umgewandelt werden in eine Dezimalzahl.

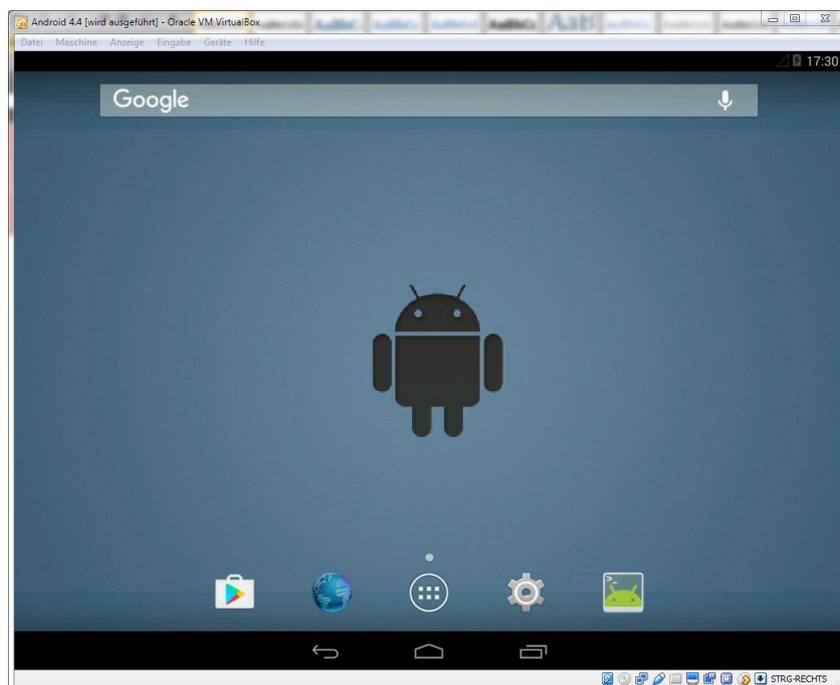
Beispiel: Man möchte ein etwas kleineres Tablet symbolieren $640 \times 480 \times 16 = \text{Hex } 311 = \text{Dez } 785$, nun tippt man hinter modes: 311 ein und bestätigt per Enter Taste.



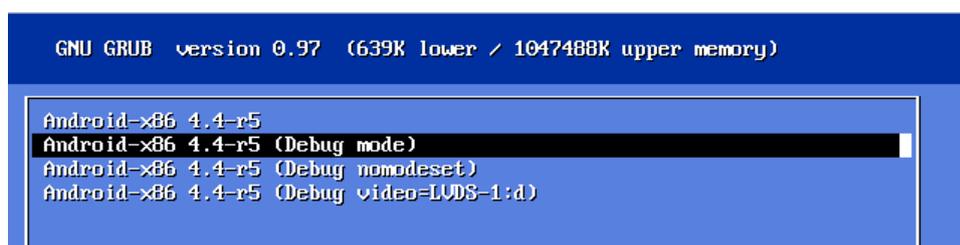
Man würde folgendes Ergebnis bekommen. Wenn man nun aber unsere geplante Auflösung von $1024 \times 768 \times 16 = \text{Hex } 317 = \text{Dez } 791$ haben möchte, öffnet man ein Rechner – Ansicht – Programmierer wählt Hex aus und tippt die 317 ein, danach switch man um auf Dez und erhält die 791.



Dies können wir nun fix konfigurieren, im ersten Schritt würde ich die Auflösung testen, wenn das funktioniert, starten wir die Virtuelle Maschine neu um die konfiguration vorzunehmen.



Und klicken diesmal den Debug Mode an.



```
[ 1.185778] EXT2-fs (sda1): warning: mounting unchecked fs, running e2fsck is
recommended
found at /dev/sda1
[ 1.205958] input: AT Translated Set 2 keyboard as /devices/platform/i8042/ser
rio0/input/input3
[ 1.216093] usb 2-1: new full-speed USB device number 2 using ohci-pci

Type 'exit' to continue booting...

Running busybox ash...
/android # [ 1.325484] tsc: Refined TSC clocksource calibration: 3103.649 MHz
[ 1.423217] usb 2-1: New USB device found, idVendor=80ee, idProduct=0021
[ 1.424532] usb 2-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
[ 1.426156] usb 2-1: Product: USB Tablet
[ 1.426518] usb 2-1: Manufacturer: VirtualBox
[ 1.442072] input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/
usb2/2-1/2-1:1.0/0003:80EE:0021.0001/input/input5
[ 1.444379] hid-generic 0003:80EE:0021.0001: input,hidraw0: USB HID v1.10 Mou
se [VirtualBox USB Tablet] on usb-0000:00:06.0-1/input0
[ 2.325871] Switched to clocksource tsc

/android #
/android #
/android #
/android #
```

Nun startet Android im Debug Modus, wenn nach kurzer Zeit keine Cursor zu sehen ist dann ein paarmal die Enter Taste drücken.

Bitte folgende Befehle eingeben um die *menu.lst* Datei zu bearbeiten.

```
on
    move          Relocate an existing mount point
    remount       Remount a mounted filesystem, changing flags
    ro/rw        Same as -r/-w

There are filesystem-specific -o flags.

/android # mount -o remount,rw /mnt
/android # vi /mnt/grub/menu.lst _
```

mount -o remount,rw /mnt
vi /mnt/grub/menu.lst

```
default=0
timeout=6
splashimage=/grub/android-x86.xpm.gz
root (hd0,0)

title Android-x86 4.4-r5
    kernel /android-4.4-r5/kernel quiet root=/dev/ram0 androidboot.hardware=
    initrd /android-4.4-r5/initrd.img

title Android-x86 4.4-r5 (Debug mode)
    kernel /android-4.4-r5/kernel root=/dev/ram0 androidboot.hardware=androi
    initrd /android-4.4-r5/initrd.img
```

Wenn nun der vi gestartet wurde sucht man sich die erste Zeile aus und navigiert mit den Cursor bis ganz ans Ende der Zeile, danach klickt man den Buchstaben „i“ für insert.

```
quiet root=/dev/ram0 androidboot.hardware=android_x86 SRC=/android-4.4-r5 vga=791
j
```

Nun trägt man den Dezimalwert für die Auflösung von 1024x768 ein (vga=791)

Mit der ESC Taste kommt man aus dem insert Modus, danach speichert man die Einstellungen mit :wq (write, quit) -- Auf der deutschen Tastatur **SHIFT + Ö, wq**

```
/android # cd /
# umount /mnt
# reboot -f
```

Nun wächelt man ins Root Verzeichnis und unmountet das */mnt* Verzeichnis wieder. Nach einen Reboot, startet die VM immer mit einer Auflösung von 1024x768.

Tastaturbelegung:

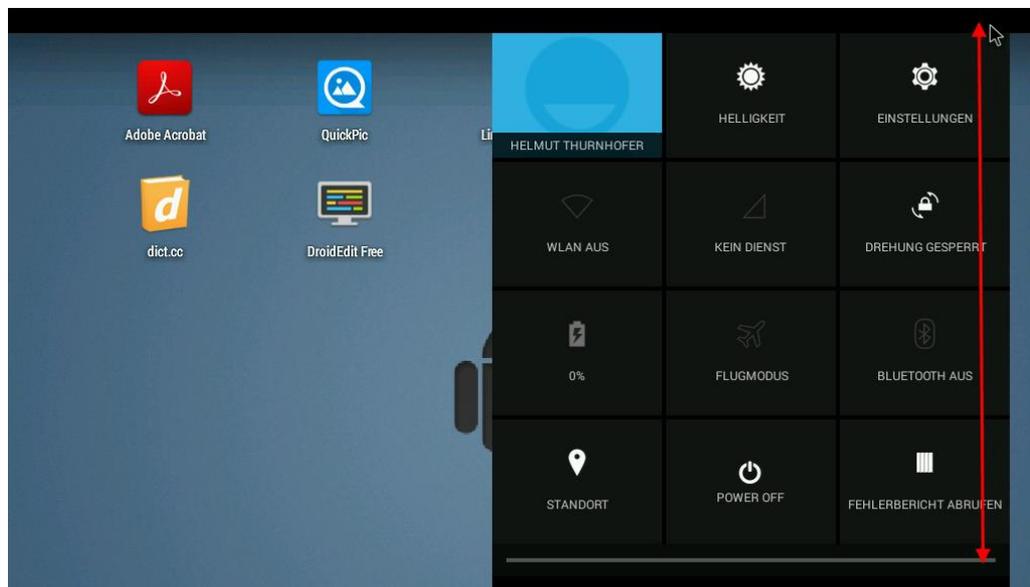
Das β = das Minus Zeichen

Rechts neben dem β ist gleich das = Zeichen.

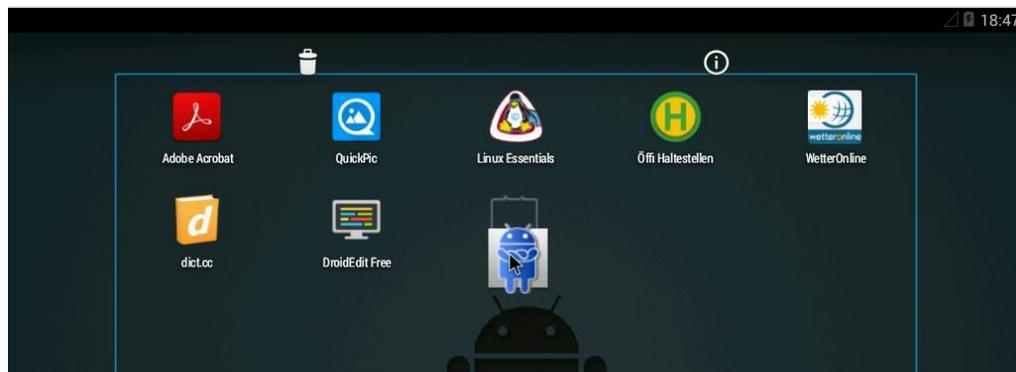
das Minus Zeichen = / Vorwärts Slash

Android 4.4 mit der Maus bedienen

Um mit der Maus arbeiten zu können benötigt es ein bisschen, bis man alles herausgefunden hat.



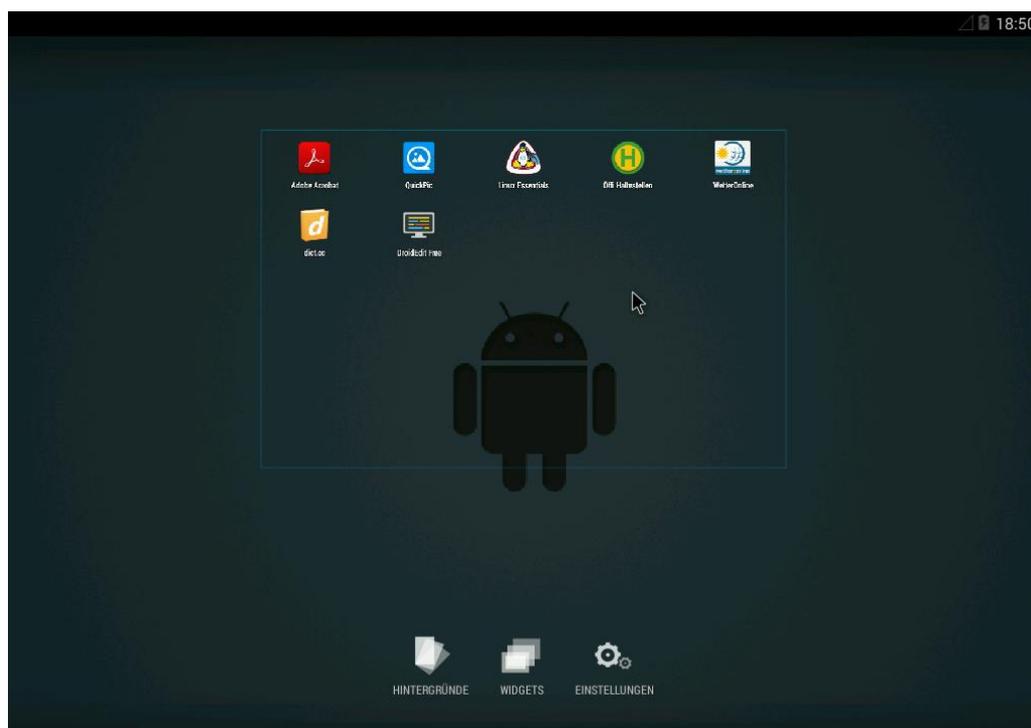
Mit der linken Maustaste die obere Leiste anklicken und mit gedrückter Maustaste langsam nach unten ziehen.



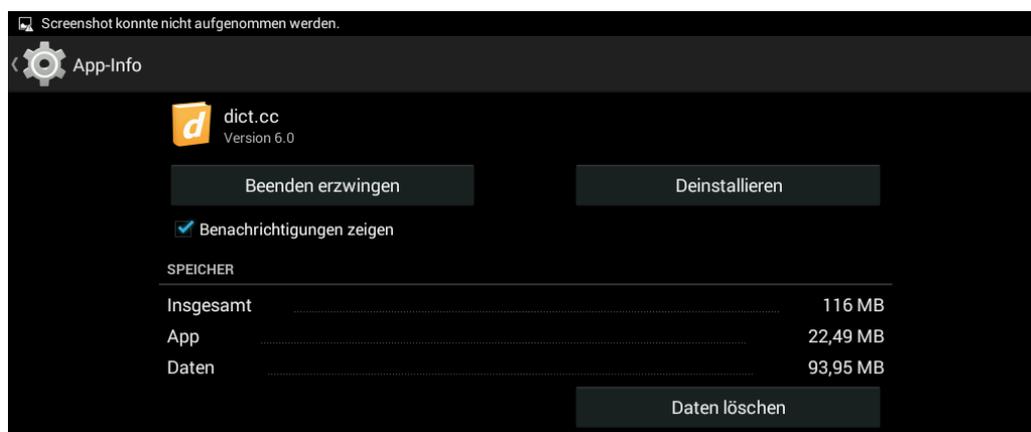
Um irgendeine Icon verschieben zu können, klickt man diese lange mit der rechten Maustaste an und verschiebt sie dann wo hin man möchte.



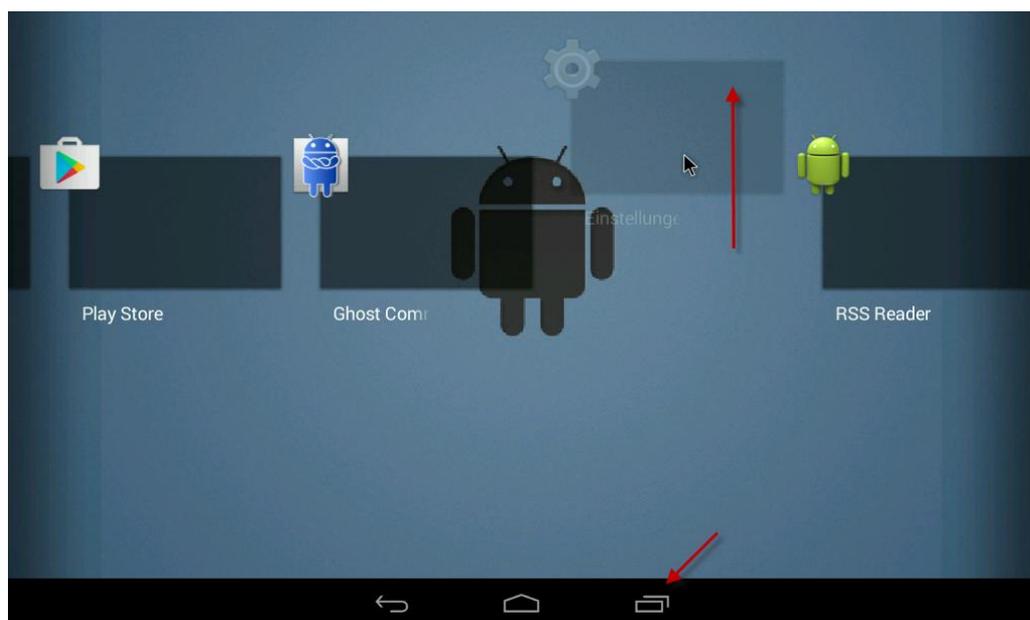
Um Verknüpfung wieder zu löschen, klickt man diese ebenfalls lange mit der rechten Maustase an und zieht sie nach oben bis ein rote Kreuz erscheint.



Lange mit der rechten Maustaste auf den Desktop, und man erhält weiter Auswahlmöglichkeiten.



Apps deinstallieren funktioniert nur über Einstellungen – Apps – App herausuchen – Deinstallieren



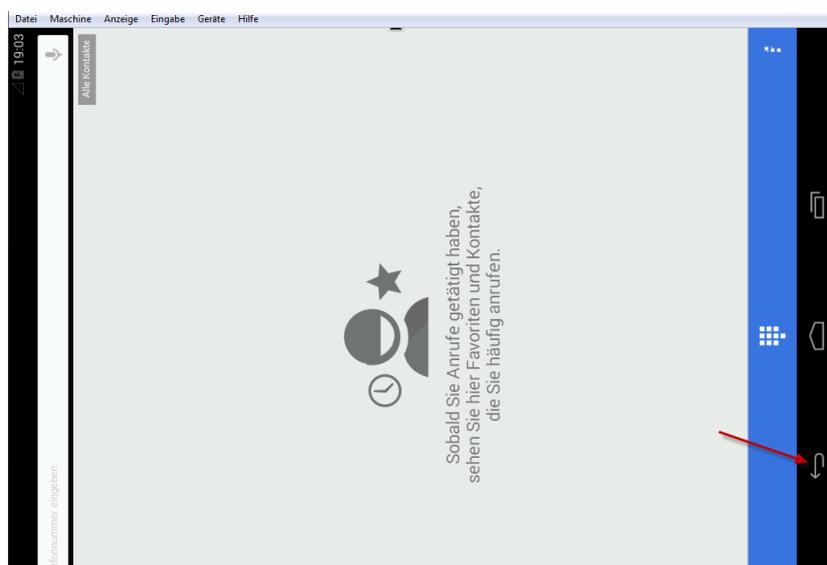
Apps wieder schließen nachdem Sie geöffnet wurden, unten rechts auf das Registersymbol und mit der linken Maustaste die Apps nach oben verschieben.



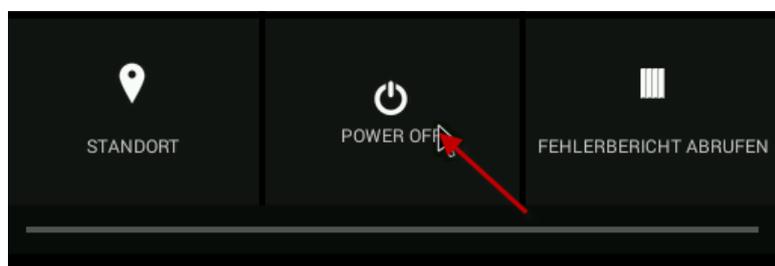
Zusatznachrichten kann man sich ebenfalls mit der linken Maustaste herunterziehen und die Liste mit den drei Balken/Strichen bereinigen.

Mir ist aufgefallen das manche Vorinstallieren Apps nicht funktionieren, diese stürzen immer nach dem Start ab (Gallery, Google Maps) Hier kann man in Google Play Store nach alternativen umsehen die funktionieren.

Manche Apps drehen sich automatisch nach dem öffnen, obwohl die Funktion abgeschaltet wurde. Hier muss man dann mit der Maus versuchen die App wieder zu schließen. Ein Highlight. 😊



Grundlegen bin ich aber ganz zufrieden, ob nun alles funktioniert wie auf einem echten Smartphone oder Tablet, mag ich zu bezweifeln. Trotzdem hat man hier ein gute Alternative um selbst Apps zu entwickeln oder nur rum zu spielen.



Über den guten alten Power off Butten beende ich mein Android in der VM wieder.

Viel Spaß damit
Helmut Thurnhofer