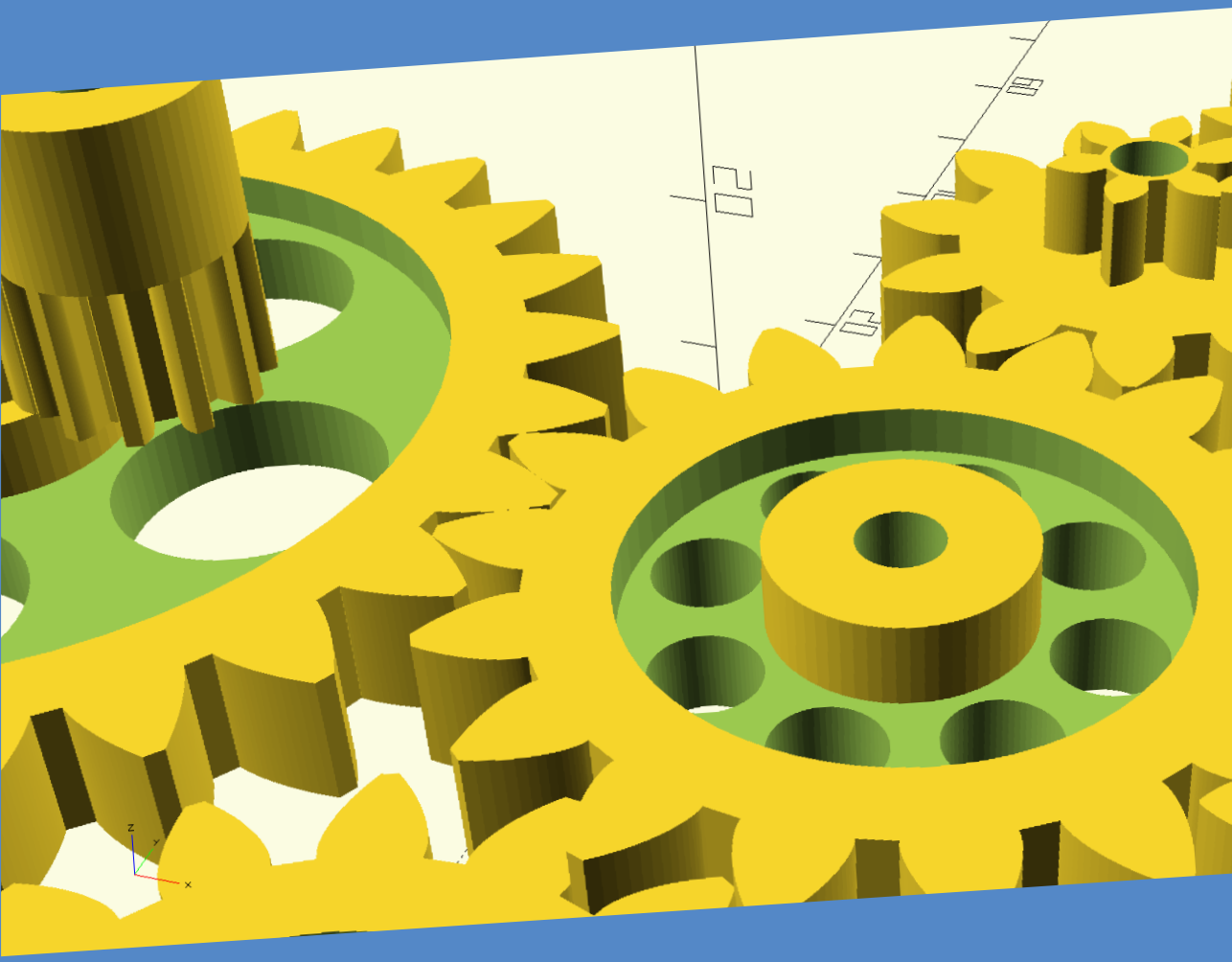
elektor books

# Technisches Konstruieren mit OpenSCAD

## Modelle für 3D-Druck, CNC-Fräsen, Prozesskommunikation und Dokumentation erstellen



Tam Hanna

elektor
design › share › sell

# Technisches Konstruieren mit OpenSCAD

## Modelle für 3D-Druck, CNC-Fräsen, Prozesskommunikation und Dokumentation erstellen

Tam HANNA

elektor

1. Auflage 2020



Umschlaggestaltung: Elektor, Aachen

Satz und Aufmachung: D-Vision, Julian van den Berg | Oss (NL)
Druck: Ipskamp Printing, Enschede, Niederlande

ISBN 978-3-89576-396-0
Ebook 978-3-89576-397-7
Epub 978-3-89576-398-4

Elektor-Verlag GmbH, Aachen
www.elektor.de

# Kapitel 1 • Wieso OpenSCAD?

Wenn es um 3D-Designsoftware Programme geht, denkt man instinktiv an Autocad, Rhino, Google SketchUp oder - die Älteren - 3D Studio Max. Außer Frage steht, dass mit all diesen Produkten großartige Designs geschaffen wurden und werden. Außer Frage steht aber auch, dass sich "Schlomo Normaldesigner" vom Denkprozess her radikal von "Schlomo Normalentwickler" unterscheidet - der beste Beleg dafür ist die in Bild 1-1 gezeigte Definition der Coder Colors, die für ein so hässliches Farbschema stehen, dass es nur dem Gehirn eines Softwareentwicklers entstammen kann.


*Bild 1-1. Die Definition des (mittlerweile ein wenig in Vergessenheit geratenen) Begriffs der Coder Colors ist wenig charmant.*
*(Bildquelle: http://www.pouet.net/topic.php?which=5540&page=1)*

Witzigerweise ist diese ingenieurwissenschaftliche Denkweise nicht nur unter Informatikern verbreitet. Der Autor arbeitet zu Ehren des amerikanischen Modeunternehmens Icy Beats LLC (siehe http://www.bopsync.com/) als technischer Konsulent und hat mit der Eigentümerin ein ähnliches Erlebnis gehabt.

Die Dame ist eine exzellente und erfahrene Modedesignerin, kommt mit 3D-Modelliersoftware aber genauso wenig zurecht wie der Autor. Ursache dafür ist, dass sowohl die Modedesignerin als auch der Ingenieur "mit ihren Händen arbeiten" und deshalb naturgemäß immer in Handbewegungen und Handhandlungen denken.

OpenSCAD unterscheidet sich von den in der Einleitung genannten Systemen insofern, als es eine "schrittweise orientierte" Vorgehensweise zu Tage legt. Da ein Bild oft mehr als 1000 Worte besagt, zeigt Bild 1-2 einen "angeschnittenen" Quader neben dem zu seiner Generierung vorgesehenen Quellcode. Bitte halten Sie sich an dieser Stelle noch nicht mit

der Analyse der Feinheiten des Codes auf - wichtig ist nur die Feststellung, dass der Quader aus der "Subtraktion" zweier Quader entsteht.

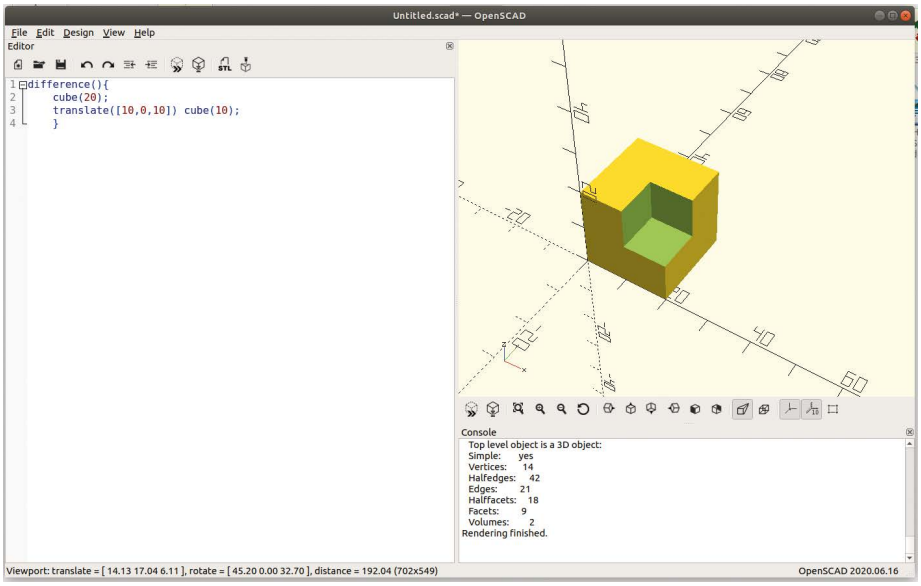



*Bild 1-2. Ein Bild sagt mehr als tausend Worte.*

Neben der insbesondere für Ingenieure "logischeren" Vorgehensweise spricht für OpenSCAD, dass die Sprache eine Art Objektorientierung umsetzt. Bestes Beispiel dafür sind die in den Bildern 1-3..1-5 gezeigten Ausschnitte aus dem unterirdischen Bunker des Autors. In beiden Fällen wurden "Halterungen" verwendet, die - im Bereich des Unterteils - aus identischem Code stammen. Der einzige Unterschied zwischen den beiden Teilen bestand darin, dass sie verschieden parametriert wurden.

*Bild 1-3. Kein Bunker ist komplett, wenn es seinen Nutzern an Seifen- und Lotion-Spendern fehlt.*

*Bild 1-4. Das gilt naturgemäß auch für den Baderaum...*

*Bild 1-5. ..dumm nur, dass die bestgeeigneten Spender in unterschiedlichen Größen vorlagen.*

Wenn Sie schon immer Lust dazu hatten, "ingenieurwissenschaftliche" Teile zu entwerfen, finden Sie mit OpenSCAD ein großartiges Modellierungssystem. Im Lehrbuch werde ich Ihnen die Grundlagen und auch einige fortgeschrittene Anwendungsszenarien vorstellen. Dabei werde ich mich allerdings auf die eigentliche Arbeit mit OpenSCAD fokussierten und peripher ein wenig auf Realisierung mit 3D-Druckern eingehen. Explizit nicht möchte ich in diesem Lehrbuch eine Einführung in "mechanical Engineering" geben - wer dies sucht, findet im Appendix mit den Literaturhinweisen eine Liste von Büchern auf verschiedenen Niveau- und Kompetenzstufen.

### 1.1 Was brauchen wir?

Vom geschätzten Leser verlangt der Autor wenig – räumliches Vorstellungsvermögen ist, logischerweise, hilfreich. Sicherer Umgang mit einer Schieblehre ist empfehlenswert; wer sich mit einem Akkuschrauber und/oder einer Bohrstation auskennt, hat ebenfalls ein leichteres Leben. Elektronikkenntnisse sind explizit nicht erforderlich!

Aus systemtechnischer Sicht benötigen wir auch nicht viel. OpenSCAD ist nicht anspruchsvoll, der Editor funktioniert auch auf langsamen Rechnern problemlos. Der Autor illustriert die Einrichtung des Programms unter Windows und Linux; unter MacOS lässt sich analog arbeiten. Scheut man den Aufwand der Eigen-Kompilation, so stehen ältere Versionen als Fertigpaket zur Verfügung.

Wer explizit für OpenSCAD eine Workstation kauft, soll auf extrem hohe Single Thread-Leistung achten. Bild 1-6 zeigt, dass sich der Gutteil der Achtkernworkstation des Autors bei der Arbeit mit OpenSCAD langweilt, während ein Kern unter Volldampf steht.

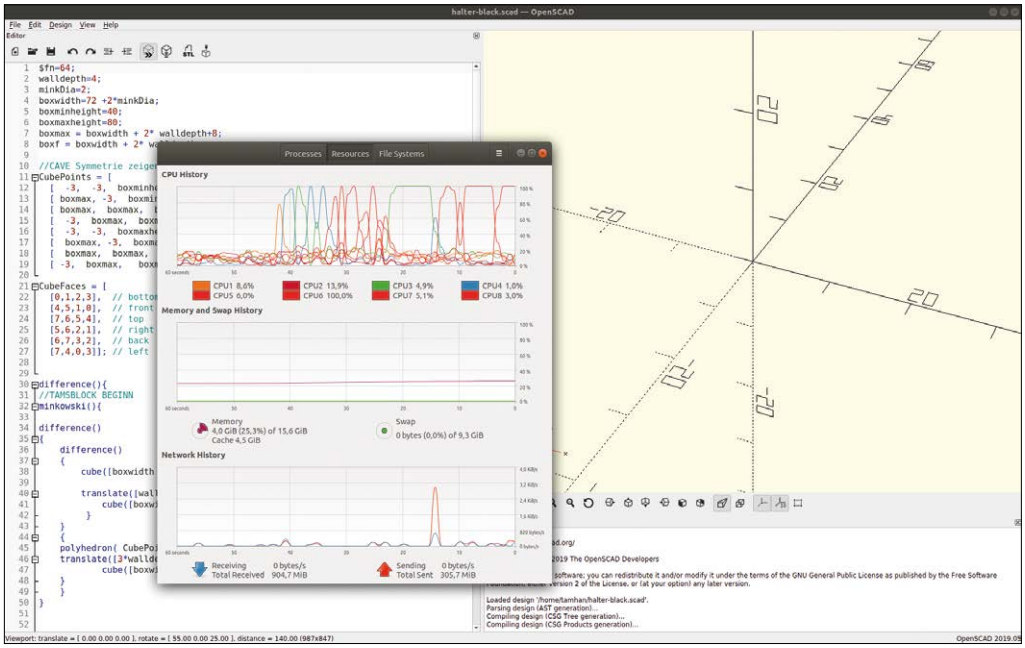



*Bild 1-6. Parallelisierung: Fehlanzeige.*

Damit zurück zu den Humanfaktoren: OpenSCAD ist eine Programmiersprache. Der Autor dieser Zeilen geht davon aus, dass sie mit C, Java, Javascript oder Pascal gearbeitet haben - andere Sprachen, egal ob objektorientiert oder nicht, taugen ebenfalls.

Sonst ist eigentlich nur Lust am Designen erforderlich - der Autor bittet sie, das Werk als Ganzes zumindest zu "überfliegen", um einen Grip für die in OpenSCAD enthaltenen Funktionen zu erhalten. Das Auswendiglernen der Syntax ist außerhalb des akademischen Bereichs wenig sinnvoll - es spricht nichts dagegen, während der Arbeit mit OpenSCAD sowohl

auf Vorlagen als auch auf dieses Buch und auf das Cheatsheet (https://www.openscad.org/cheatsheet/) zurückzugreifen.

**Wer eine kann, lernt die zweite leichter**
Die Erfahrung lehrt, dass eine Person, die mit einer Programmiersprache firm ist, sich vergleichsweise schnell in die (meisten) anderen einarbeiten kann. Der Autor kann dies auch aus seiner eigenen Lebenserfahrung bestätigen - er begann seine Karriere mit Pascal, um sich nebenbei C anzueignen und hat seither auch mit exotischeren Sprachen wie Python keine großen Probleme gehabt.

Finden sie sich partout mit einer älteren Version von OpenSCAD konfrontiert, so ist die Situation nicht allzu schlimm. Das OpenSCAD-Entwicklerteam stand einer politischen Partei nahe, die in Österreich mit Sicherheit die nächsten zehn Jahre keine nennenswerte Regierungsverantwortung bekommen wird und dementsprechend nicht sonderlich viel Geld zu verteilen hat. Daraus folgt – logischer-, aber auch irgendwie erfreulicherweise - eine sehr langsame Weiterentwicklung der Sprache.

Über die Frage, welcher 3D-Drucker für die Bedürfnisse eines Unternehmens oder einer Person besser geeignet ist, werden härtere Kriege ausgetragen als über die Form der perfekten Frauenfüße oder die beste Zigarre.

Der Autor dieser Zeilen verwendet hausintern meistens den bei Conrad für ungefähr 175 Euro erhältlichen Renkforce RF100 V2 - ein kleines kompaktes Fertiggerät, das (insbesondere mit nachgerüsteter Bauteilkühlung, siehe YouTube-Kanal des Autors) problemlos funktioniert und Teile in einer Größe von bis zu 12 × 12 × 12 cm fertigt.

Wenn Sie einen anderen 3D-Drucker haben und mit diesen souverän umgehen können, ist dies naturgemäß kein Hindernis. Wir führen unsere Experimente in Kapitel fünf mit CURA durch, weil der Autor immer CURA verwendet. Verwenden Sie einen anderen Slicer und können Sie mit ihm schon umgehen, so ist dies auch kein Problem.

Wer keinen 3D-Drucker hat und auch keinen Platz (und keine Wartungszeit) für so ein Geraffel auftreiben kann, kann stattdessen auf die Dienste diverser Modelltischlereien setzen. So gut wie jedes Ladengeschäft, das 3D-Drucker hat, nimmt Druckaufträge entgegen und "söldnert" - freundliche Herangehensweise vorausgesetzt - äußerst gern.

### 1.2 Wer schreibt hier?

Dass elektronische Komponenten und (Rüstungs-) Elektronik das Leben eines einsamen rassischen Halbarabers in Österreich vor vielen Jahren bereichert haben, sollte spätestens nach Lektüre meines Mikrocontroller-Lehrbuchs bekannt sein. Die technische Weiterentwicklung gab in den letzten 20 Jahren reiche Geschenke - waren Oszilloskope und 32-Bit-Prozessoren einst unerschwinglich, sind sie nun auch für ein "kleines" Budget leistbar. Farbdisplays sind mittlerweile sogar so preiswert, dass das Unternehmen des Autors sie zur Freude aller Leut‘ in Humidore (!!!) einbaut (siehe Bild 1-7).


*Bild 1-7. Farbdisplay in der Zigarrenkiste: dank HygroSage für Jedermann erreichbar.*

Bei all dieser Weiterentwicklung blieb immer die Frage offen, wie man "hauseigene" mechanische Teile realisiert - eine wunderbare Steuerung für eine Drohne nutzt einem auch beim Vorhandensein von Bürgerkrieg und Käufer nichts, wenn man sie nicht realisieren kann.

Als der Autor noch im aktiven Dienst war, war für jedes Teil eine lange und holprige Reise in einem ZAZ-965 Zaporizhzhets der Fahrbereitschaft erforderlich, um danach mit der Modelltischlerei zu sprechen. Aus einem kleinen Holm oder Trägerteil wurde eine Aufgabe, die hochqualifiziertes technisches Personal hart forderte.

3D-Drucker haben den "technischen" Teil dieser Frage mittlerweile beantwortet - in der BRD bekommt man zugegebenermaßen kleine, aber problemlos funktionierende Geräte, wie in der Einleitung gesagt, für weniger als 200 Euro.

Offen blieb die Frage, wie man den theoretischen bzw. designtechnischen Teil der Frage löst. OpenSCAD ist nach Ansicht des Autors die geradezu ideale Antwort darauf – wenn Sie schon immer nach einem Weg suchten, um mechanische Teile - seien dies Gehäuse oder Halterungen - zu entwerfen, so finden Sie hier einen komfortablen Weg. Der Autor möchte das Leben mit OpenSCAD jedenfalls nicht mehr missen - ob bei der Einrichtung seiner Immobilie, beim Vorbereiten von Proof of Concepts für Gerichtsverhandlungen oder einfach beim Designen eines Ersatzknopfs für das geliebte LeCroy-Oszilloskop eines französischen Elektronikers: OpenSCAD ist ein Produkt, das immer und immer wieder Freude macht.

In diesem Sinne: nichts wie los! Und mögen ihre Designs – egal ob zivil oder militärisch - immer perfekt funktionieren!

# Kapitel 2 • OpenSCAD bereitstellen

An dieser Stelle geht der Autor davon aus, dass Sie Experimente mit OpenSCAD durchführen wollen. Hierzu benötigen wir – logischerweise – eine Version der Software.

Wie so viele andere "neuartige" ingenieurwissenschaftliche Programme ist auch OpenSCAD quelloffen. Daraus ergibt sich zwar einerseits, dass sie den Quellcode des Produkts ansehen und gefundene Fehler selbst beheben können. Andererseits bedeutet dies allerdings auch, dass es mehrere Darreichungsformen gibt.

In diesem Kapitel wollen wir uns die Bereitstellung unter Windows und Linux ansehen. Wer unter MacOS arbeiten möchte, findet im Internet diverse Anweisungen sowohl zur Kompilation als auch zur Verwendung bereitgestellter Pakete.

## 2.1 OpenSCAD installieren: Linux, fertiges Paket

Wer sich einige Zeit mit Linux auseinandersetzt, kennt Kommandos wie apt-get mit Sicherheit. Sie befragen im Internet oder im lokalen Netzwerk lebende "Paketquellen-Server" nach einem Paket, um es danach herunterzuladen und zu installieren.

Da die meisten Distributoren mehr oder weniger umfangreiche Überprüfungen der in ihren Paketquellen gelisteten Pakete durchführen, gilt, dass die dort enthaltenen Pakete oft nicht besonders aktuell sind - im Fall von Ubuntu bekommt man vom Server beispielsweise eine Version aus dem Jahre 2015. Quereleien zwischen dem OpenSCAD- und dem Ubuntu-Team sorgen dafür, dass manche Releases von Ubuntu komplett ohne OpenSCAD auskommen müssen.

**Paket-Management, ganz leicht.**
Wenn Sie sich schon immer dafür interessiert haben, was im Gehirn eines Paket-Managers vorgeht, so empfiehlt der Autor den auf den Chemnitzer Linuxtagen im Jahr 2018 gehaltenen und unter https://chemnitzer.linux-tage.de/2018/de/programm/beitrag/285 als Video bereitstehenden Vortrag.

Wer unter Ubuntu mit einem halbwegs aktuellen OpenSCAD arbeiten möchte, besucht im ersten Schritt die URL https://www.openscad.org/downloads.html und scrollt dann bis zum Abschnitt "Other Linux" hinunter. Klicken Sie danach auf die in Bild 2-1 gezeigten Link, um eine rund 35 MB große Datei herunterzuladen.

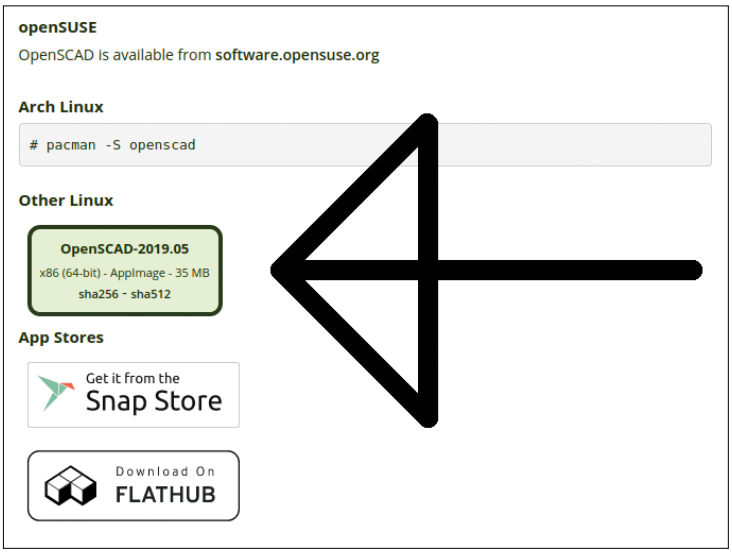



*Bild 2-1. Dieser Link führt ins Glück.*

Beachten Sie, dass "fertige" OpenSCAD-Pakete für Linux nur noch für 64-Bit-Versionen des Betriebssystems zur Verfügung stehen. Wer unbedingt mit einer 32-Bit-Variante arbeiten muss oder möchte, muss - wie weiter unten besprochen - eine manuelle Kompilation durchführen.

AppImage-Dateien sind ein "neuartiges" Verpackungsprogramm, das Linux-Applikationen samt den zu ihnen gehörigen Bibliotheken verpackt. Der Start von OpenSCAD erfolgt durch die Eingabe der folgenden beiden Kommandos:

```
tamhan@TAMHAN18:~/Downloads$ chmod +x OpenSCAD-2019.05-x86_64.AppImage
tamhan@TAMHAN18:~/Downloads$ sudo ./OpenSCAD-2019.05-x86_64.AppImage
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to ‚/tmp/runtime-root‘
. . .
```

Der erste Aufruf weist OpenSCAD dabei das Attribut ausführbar zu, während die darauffolgende Aktivierung die "eigentliche" Anweisung zum Start von OpenSCAD erteilt. Das Programm reagiert darauf mit der Einblendung des OpenSCAD-Startbildschirms, den Sie durch Klick auf das Verlassen-Symbol "deaktivieren".

Je nach Distribution gibt es auch noch die Möglichkeit, OpenSCAD in den Programmstarter zu integrieren - da der Autor dieser Zeilen gerne auf Kommandozeilenebene arbeitet, sei hier auf die Dokumentation des jeweiligen Systems verwiesen. Alternativ dazu können Sie auch die im folgenden Schritt besprochenen Kompilationsanweisungen abarbeiten, die OpenSCAD "vollständig" in ihr System integriert.

### 2.2 OpenSCAD installieren: Linux, Kompilation.

Unsere als fertiges Paket heruntergeladene OpenSCAD-Version 2019.05 war zwar ein funktionaler Versionsstand, der allerdings nicht sonderlich aktuell ist. Wer OpenSCAD aus dem Quellcode kompiliert, bekommt erstens die aktuelle Variante und kann den Compiler zweitens dazu anweisen, eine experimentelle Version der OpenSCAD-Beschreibungssprache bereitzustellen. Dadurch können Sie von zusätzlichen Features profitieren, an die man "so" nicht herankommt.

Ärgerlicherweise besteht OpenSCAD aus einer ganzen Gruppe von Komponenten, weshalb die Kompilation ein wenig Arbeit voraussetzt.

**Der Komponenten-Zoo.**
Wenn Sie wissen wollen, welche Bibliotheken im Hintergrund von OpenSCAD arbeiten, sei die URL https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/Building_OpenSCAD_from_Sources wärmstens empfohlen.

Wie so gut wie alle anderen Open Source-Projekte "größerer" Dimension erfolgt auch die Verwaltung von OpenSCAD unter Verwendung des git-Versionskontrollsystems. Unsere erste Aufgabe besteht darin, nach folgendem Schema sicherzustellen, dass git auf ihrer Workstation installiert ist:

```
tamhan@TAMHAN18:~/Downloads$ sudo apt-get install git
```

Im nächsten Schritt weisen wir das git-Kommando dazu an, die aktuelle Version des Quellcodes aus dem Repository herunterzuladen. Wundern Sie sich nicht darüber, dass der Autor diesen Befehl in seinem Stammverzeichnis eingibt - so gut wie alle git-Repositories erzeugen im Rahmen des Downloads ein neues Verzeichnis, in dem die eigentlichen Inhalte unterkommen:

```
tamhan@TAMHAN18:~$ cd ~/
tamhan@TAMHAN18:~$ git clone https://github.com/openscad/openscad.git
Cloning into ‚openscad‘...
. . .
Receiving objects: 100% (64355/64355), 62.52 MiB | 5.11 MiB/s, done.
Resolving deltas: 100% (45989/45989), done.
```

Der verwendete Befehl lädt die wirklich letzte aktuelle Version herunter - dies ist insofern "von Nachteil", als dass es während der Entwicklung (selten) zum Hochladen von nicht funktionierenden oder nicht kompilierbaren Versionen kommt. In diesem Fall empfiehlt es sich, das angelegte Verzeichnis zu löschen und danach anzuweisen, über die Branch-Funktion eines der unter https://github.com/openscad/openscad/tree/master/releases bereitstehenden Releases herunterzuladen.

Sei dem wie es sei, kehren wir im nächsten Schritt ins OpenSCAD-Arbeitsverzeichnis zurück:

```
tamhan@TAMHAN18:~$ cd openscad/
```

Git-Repositories sind seit langer Zeit zur Abbildung von Vernetzungsbeziehungen befähigt. Dies ist für uns insofern nachteilig, als wir innerhalb des OpenSCAD-Ordners die folgenden beiden Befehle eingeben müssen, um eine vollwertige Quellcodeumgebung bereitzustellen:

```
tamhan@TAMHAN18:~/openscad$ git submodule init
Submodule ‚libraries/MCAD‘ (https://github.com/openscad/MCAD.git) registered
for path ‚libraries/MCAD‘
tamhan@TAMHAN18:~/openscad$ git submodule update
. . .
Cloning into ‚/home/tamhan/openscad/libraries/MCAD‘...
Submodule path ‚libraries/MCAD‘: checked out
‚a7be3d623669d635b7249a327cfce5796ea200b3‘
```

Der explizite Hinweis auf die verschachtelten Beziehungen in git-Repositories ist insofern wichtig, als git-Verweigerer gern den in Bild 2-2 gezeigten Download-Knopf verwenden. Leider ist seine Verwendung nicht zielführend, da das dabei bereitgestellte Archiv die verbundenen Repositories nicht mitbringt.

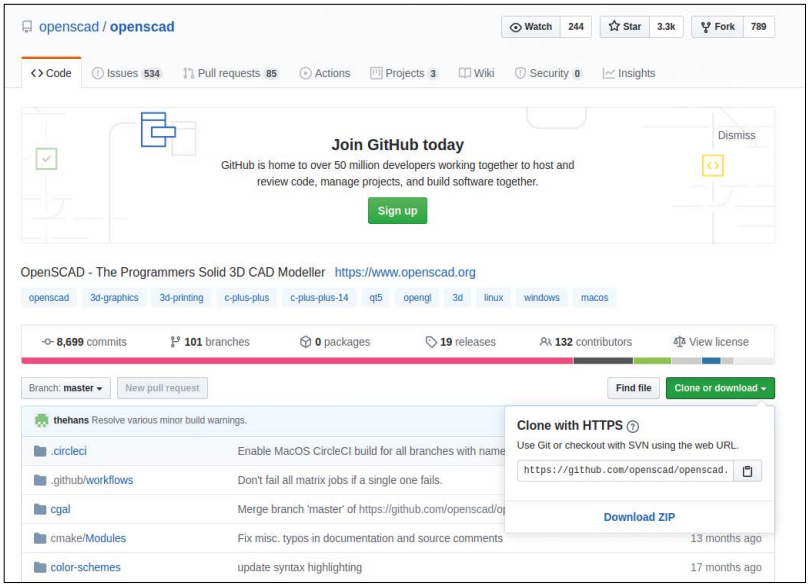



*Bild 2-2. Git-Flüchtlinge, aufgepasst: dieser Knopf verspricht Unheil.*

Im nächsten Schritt müssen wir die sonstigen Abhängigkeiten bereitstellen:

```
tamhan@TAMHAN18:~/openscad$ sudo ./scripts/uni-get-dependencies.sh
. . .
Use ‚sudo apt autoremove‘ to remove them.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
installed qt5-default to enable qmake
```

Das OpenSCAD-Entwicklerteam unterstützt uns bei dieser Aufgabe durch ein Skript, das auf halbwegs aktuellen Versionen von Ubuntu die notwendigen Elemente automatisch unter Verwendung von Paketmanager und anderen Werkzeugen herunterlädt. Kommt es dabei zu Problemen, finden Sie Hilfe unter https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/Building_on_Linux/UNIX#Installing_dependencies

An dieser Stelle ist ein Neustart des Systems empfehlenswert. Danach lässt sich das Vorhandensein der nötigen Beziehungen durch folgendes Kommando überprüfen:

```
tamhan@TAMHAN18:~/openscad$ ./scripts/check-dependencies.sh
```

Auf der Workstation des Autors führt seine Abarbeitung zum in Bild 2-3 gezeigten Ergebnis.

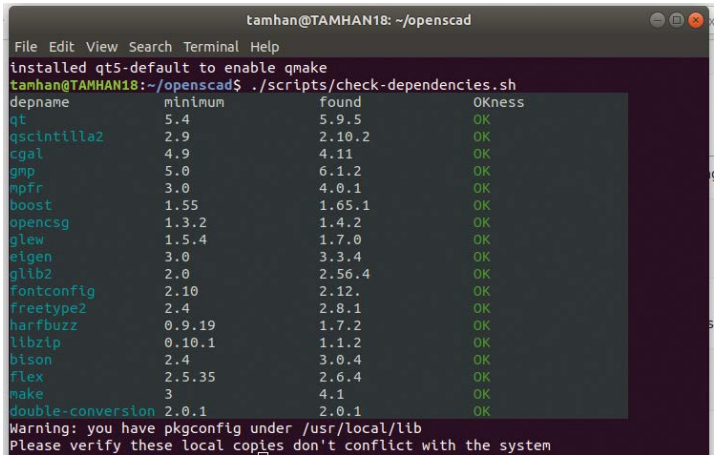



*Bild 2-3. Die notwendigen Abhängigkeiten sind zur Kompilation bereit.*

Sei dem wie es sei, wir müssen im nächsten Schritt einen Durchlauf des qmake-Kommandozeilenwerkzeug befehligen. Es hat die Aufgabe, die auf dem Rechner befindlichen Quellcodes in eine für den Compiler "verarbeitbare" Struktur zu bringen und diverse Skriptdateien und sonstige Elemente bereitzustellen, auf die der Compiler während seiner Arbeit zurückgreifen wird. Möchten Sie eine "gewöhnliche" Version von OpenSCAD kompilieren, so reicht die folgende Eingabe:

```
tamhan@TAMHAN18:~/openscad$ qmake
Info: creating stash file /home/tamhan/openscad/.qmake.stash
Project MESSAGE: If you‘re building a development binary, consider adding
CONFIG+=experimental
. . .
to the PKG_CONFIG_PATH environment variable
No package ‚lib3MF‘ found
Project MESSAGE: 3MF Import/Export disabled
```

Wundern Sie sich nicht darüber, wenn das Werkzeug während der Abarbeitung des qmake-Befehls eine Gruppe von Status-Ausgaben auswirft, die sich auf das Fehlen diverser Import-Export-Bibliotheken beziehen. Sofern qmake keinen wirklich ernsthaften Fehler auswirft, dürfte alles problemlos funktionieren.

So sie OpenSCAD kompilieren, um von modernen bzw. experimentellen Funktionen zu profitieren, sieht der Aufruf von qmake etwas anders aus. Sie müssen in diesem Fall ein zusätzliches Attribut anliefern, das über den CONFIG-Kommandozeilenparameter in das System wandert:

```
tamhan@TAMHAN18:~/openscad$ qmake CONFIG+=experimental
. . .
```

Wir können an dieser Stelle eine Kompilation unserer Applikation anweisen. Das Übergeben des Parameters -B weist make dabei dazu an, alle schon vorhandenen make-files zu eliminieren. Dies ist zum Beispiel dann wichtig, wenn sie zwischen der Kompilation einer gewöhnlichen und einer experimentellen Version von OpenSCAD wechseln:

```
tamhan@TAMHAN18:~/openscad$ make -B
. . .
```

Wegen des doch nicht unerheblichen Codeaufwands ist es empfehlenswert, die Bereitstellung bzw. die durch make ausgelöste eigentliche Kompilation so stark wie möglich zu parallelisieren. Hierzu müssen Sie den Parameter -j übergeben - wer wie der Autor an einer Achtkernworkstation sitzt, gibt folgendes Kommando ein:

```
tamhan@TAMHAN18:~/openscad$ make -B -j 8
. . .
418 translated messages, 24 fuzzy translations, 7 untranslated messages.
itstool missing, won‘t apply translations to openscad.appdata.xml
tamhan@TAMHAN18:~/openscad$
```

Während der Abarbeitung wirft make mitunter – wie in Bild 2-4 gezeigt – Fehler, die auf Probleme mit den Übersetzungsdateien hinweisen. Diese sind für das Funktionieren von OpenSCAD allerdings belanglos.

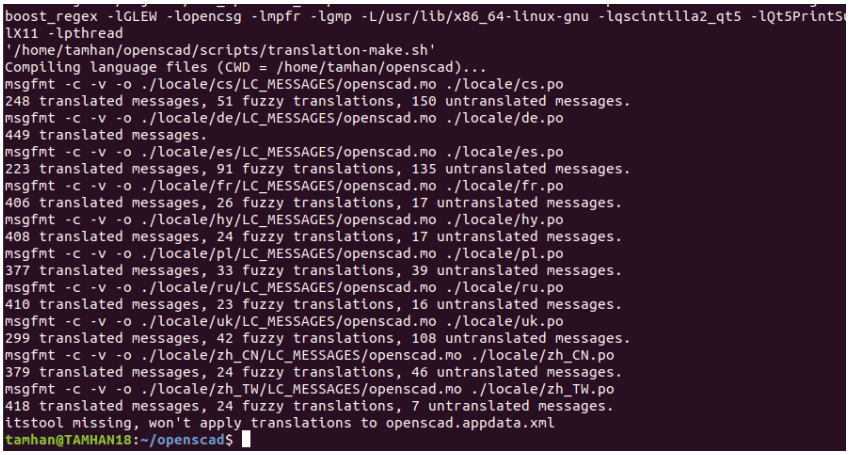



*Bild 2-4. Der make-Lauf zeigt sich durchaus kommunikativ.*

**Wieso qmake und make?**
OpenSCAD basiert auf dem C++-Cross-Plattform-Framework QT. Dieses bietet Entwicklern nicht nur Bibliotheken und einen GUI-Stack an, sondern erweitert den C++-Sprachstandard auch um Funktionen wie ein Signal-Slot-System.

Um mit diesen Attributen ausgestattete Applikationen mit dem gewöhnlichen Compiler des Betriebssystems verarbeiten zu können, ist ein zusätzlicher Arbeitsschritt erforderlich. Ein als qmake bezeichnetes Projekt kümmert sich darum, die Makefiles auf eine Art und Weise aufzubauen, die eine Aktivierung der Qt-eigenen Hilfsinfrastruktur anweist. Nach der Abarbeitung des qmake-Kommandos findet sich im Projektverzeichnis ein Makefile, das für die eigentliche Kompilation sorgt.

Nach dem erfolgreichen Durchlaufen des Kommandos hat der Compiler eine ausführungsfertige Binärdatei erzeugt. Im Interesse der Sicherheit ist es an dieser Stelle empfehlenswert, zwei "häufige" Ablageorte von OpenSCAD-Binaries zu bereinigen:

```
tamhan@TAMHAN18:~/openscad$ rm /usr/local/lib/openscad
rm: cannot remove ,/usr/local/lib/openscad‘: No such file or directory
tamhan@TAMHAN18:~/openscad$ rm /usr/local/share/openscad
rm: cannot remove ,/usr/local/share/openscad‘: No such file or directory
```

Wenn sie wie hier die Meldung "No such file or directory" bekommen, so war die jeweilige Datei nicht vorhanden. Bekommen Sie stattdessen einen Berechtigungsfehler, so müssen Sie mit sudo nachlegen.

Wer unter einem von Debian abgeleiteten Betriebssystem - Ubuntu gehört explizit dazu - arbeitet, muss an dieser Stelle das checkinstall-Werkzeug herunterladen. Es kümmert sich darum, die Binärdatei in eine für die Debian-Paketverwaltung verarbeitbare Paketierung zu bringen: