

Häufig gestellte Fragen zum Umgang mit Z88 Aurora

- ***Nach dem Installieren startet das Programm nicht, Fehlermeldung "Das Programm hat einen Fehler festgestellt und muss beendet werden" / Pfade des Optionsmenüs lassen sich nicht editieren/ STL/STEP Import mit Geocoon, TETGEN/NETGEN starten nicht.***

Bitte installieren und benutzen Sie das Programm entweder als Administrator oder als normaler Benutzer. BITTE NICHT ZWISCHEN DEN KONTEN WECHSELN! Wenn Sie als Administrator installieren und als Benutzer das Programm verwenden kann es Problemen mit den Zugriffsrechten der Benutzerkontensteuerung geben!

- ***Eingabefenster liegen außerhalb des Monitors:***

Die voreingestellte Standardgröße von Z88 Aurora beim Start ist 800x600. Sie haben zwei Möglichkeiten diese zu ändern, entweder editieren Sie in der Datei z88.fcd (Verzeichnis z88aurorav1/bin) die Parameter "IW_DRAWAR" und "IH_DRAWAR" oder Sie verwenden eine, der im Optionsmenü unter dem Reiter "Ansicht" voreingestellten Bildschirmauflösungen.

- ***Eine Geometriedatei lässt sich nicht korrekt einlesen/vernetzt nicht:***

Manchmal kommt es beim Ausschreiben einer Datei aus dem CAD Programm zu Fehlern innerhalb dieser Datei. Die in Z88 Aurora hinterlegte Bibliothek zum 3D Datenaustausch für Oberflächen- und Solidmodellierung "OpenCascade" ist daher nicht in der Lage, diese Dateien zu verarbeiten, was wiederum zu Darstellungsfehler oder Abbrüche bei der Vernetzung führen kann.

- ***Vernetzung bricht ab:***

Z88 Aurora ist eine Weiterentwicklung von Z88V13 mit einer erweiterten Benutzeroberfläche für das Prä- und Postprozessing. Die Vernetzung erfolgt mit Hilfe zweier externer Freeware Programme, für deren Abläufe keine Garantie übernommen werden kann.

- ***Bildschirm flackert, Darstellung ist fehlerhaft, Graphik ruckelt bei großen Strukturen:***

Darstellungsprobleme in der Graphik rühren meist von der Graphikkarte her, insbesondere aus einer suboptimale Unterstützung von Open GL Funktionalitäten heraus. Abhilfe können separat installierte Treiber des jeweiligen Graphikkartenherstellers schaffen. Speziell bei Windows 7 können Konflikten mit den Darstellungseinstellungen (Aero-Oberfläche) des Betriebssystems auftreten. In diesem Fall die Funktionalität in Windows bitte für den Gebrauch von Z88 Aurora deaktivieren.

- ***Aurora kann Dateien/Hilfedokumente/Videos nicht öffnen:***

Im Optionsmenü können unter dem Reiter "Pfade" die entsprechenden Pfade gewählt werden, näheres siehe im Installationshandbuch.

- **Die eingestellten Pfade werden nicht übernommen:**

Dieser Fehler trat bisher nur bei Windows Vista auf und ist auf die Benutzerkontensteuerung zurückzuführen. In diesem Fall führen Sie Z88 Aurora bitte als Administrator aus.

- **Fehlermeldung "libgtk ... wurde nicht gefunden":**

Auf dem Rechner wurde nicht das von Z88 Aurora benötigte GTK Paket installiert. Eventuell sind durch andere Programmpakete (z.B. Gimp) ältere GTK-Versionen vorhanden, die nicht alle Funktionalitäten von Z88 Aurora unterstützen. Installieren Sie deshalb das auf der Homepage vorgeschlagene GTK Paket, näheres findet sich im Installationshandbuch.

- **Bei gleichzeitigen Beaufschlagung von mehreren Kräften auf einen Knoten wird nur eine Kennzeichnung dargestellt :**

Momentan können mehrere Lasten auf einen Knoten aufgebracht werden, die einzelnen Lasten werden jedoch übereinander dargestellt, was dazu führt, dass nur eine Farbe sichtbar ist. Abhilfe schafft hier die Option "Randbedingungen" im der Ansichtsmenü der Textmenüleiste. Hier kann jede einzelne Randbedingung separat dargestellt werden.

- **Fehlermeldung: Datei MSVCR80.dll konnte nicht gefunden werden :**

Aus dem <Installationsverzeichnis>\bin von **GTK** müssen die Dateien
Libglib.dll
Libobject.dll
Zlib1.dll

in das <Installationsverzeichnis>\bin\win64 von Z88 Aurora **KOPIERT** werden. Die Dateien müssen danach in **BEIDEN** Verzeichnissen vorhanden sein. Ein Verschieben der Dateien kann zu anderweitigen Fehlern führen.

- **Fehlermeldung: Cannot open Pango Font! STOP!
(Nur für Z88Aurora V1 und Z88AuroraV1a – weitere Informationen siehe Seite 3)**

Ursache: Die Standardschriftart Arial ist nicht auffindbar. Öffnen Sie das Installationsverzeichnis z88aurorav1/bin/[Betriebssystem] und editieren Sie die Datei **z88.fcd**. Sie finden dort die Passage:

WINDOW START

```
IW_DRAWAR      640
IH_DRAWAR      512
PANGO_FONT     Arial
PANGO_SIZE     10
DEPTH_SIZE     1
WINDOW END
```

Ersetzen Sie **Arial** hinter **PANGO_FONT** durch z.B. **Sans** und speichern Sie die Datei.

Ergänzungen von Prof. Rieg:

- **Wie reagiere ich auf die Startmeldung von Z88Aurora "kann PANGO-Font nicht öffnen"?**

Diese Meldung tritt meist nur bei Mac OS X und LINUX in früheren Z88Aurora-Versionen auf.

Einfache Abhilfe: Z88Aurora V1b installieren.

Hintergrund: Windows, Mac OS X und LINUX verwenden intern jeweils unterschiedliche Font-Management-Systeme. OpenGL selbst bietet dafür keinerlei Unterstützung; das macht GTK+. Intern geht GTK+ entweder auf die WinAPI oder auf das X11-System. Die internen X11-Aufrufe sind aber nicht mehr ganz auf dem aktuellen Stand und machen besonders bei neueren LINUX- bzw. Mac OS X-Betriebssystemen Schwierigkeiten.

Da dies seit Jahren zu Problemen führt, haben wir uns ab der Aurora-Version V1b für eine Radikalkur entschieden: Wir haben unser eigenes Font-System programmiert, das wahlweise einen Rasterfont oder einen Vektorfont bietet: Der Rasterfont ist Standard; der Vektorfont ist besonders für Personen mit Sehschwächen geeignet und wird mit

VFONT_USE 1

in der Steuerdatei Z88.FCD aktiviert (mit **VFONT_SCALE** und **VFONT_THICK** kann dann nach Wunsch skaliert werden: ausprobieren). Z88.FCD liegt im jeweiligen **bin**-Unterverzeichnis.

- **Z88Aurora startet mit einer roten Zeichenfläche bzw. rechnet offensichtlich falsch!**

Z88Aurora ist in C programmiert und verwendet als Dezimaltrenner daher den Punkt - wie das in der englischen Sprache selbst, aber bei praktisch allen Programmiersprachen eben so ist. Viele deutsche Betriebssysteme tun hier des Guten zuviel und verwenden als Dezimaltrenner das in Deutschland übliche Komma. Dann wird die Steuerdatei Z88.FCD (im jeweiligen **bin**-Verzeichnis) falsch interpretiert, was zu falschen Farben führt, und die Rechenmodule rechnen natürlich auch falsch. Diese Problematik tritt eigentlich nur bei LINUX und u.U. Mac OS X auf.

Abhilfe: In die `.bashrc` bzw. `.bashprofile` eintragen:

```
export LANG=C
```

```
export PATH=$PATH:. (also Doppelpunkt Punkt)
```

(nur für Mac OS X)

```
export DYLD_LIBRARY_PATH=$HOME/z88aurorav1/addons/geocon/mac
```

Oder Z88Aurora von einer Konsole starten und vorher eben diese Befehle eingeben. Diese Befehle sind nichts Besonderes, sondern ganz normaler UNIX-Standard.

- **Z88Aurora ist Freeware - warum nicht gleich Open Source?**

Dies hat mehrere Gründe:

1. rechtliche Gründe, die hier nicht weiter diskutiert werden sollen.

2. Z88Aurora ist eines der wenigen freien FEA-Programme, die sozusagen "out-of-the-box" weitestgehend identisch unter Windows, LINUX und Mac OS X arbeiten - und zwar im native mode! Das klingt sehr einfach, ist in der Programmierung aber relativ tricky, und es sind viele Besonderheiten intern zu beachten.

Obwohl wir mit ANSI-C als Programmiersprache sowie mit GTK+ für die GUI und OpenGL für die 3D-Grafik arbeiten, muss man sehr tiefgehende Kenntnisse der WinAPI und des X11-Systems haben, um damit klarzukommen. Nur ein Beispiel von sehr vielen: Es gibt im Internet keine fertige Anbindung GTK+ und OpenGL an Windows für den 64-Bit Modus; das haben wir selbstgeschrieben. Mein Team und ich haben hier in aller Bescheidenheit über viele Jahre sehr viel Know-How angesammelt, das man nicht in zwei Sätzen weitergeben kann. Umgekehrt stelle ich aber den Anspruch an OpenSource-Software, dass sie mit einem **configure** und einem **make** dann zu übersetzen ist - das wäre bei unserem System nicht hinzukriegen, und daher lassen wir es lieber ganz. Denn es gibt gerade bei Finite-Elemente-OpenSource-Programmen im Internet dahingehend sehr schlechte Beispiele, die selbst Fortgeschrittene wie wir nicht sauber (oder eher gar nicht) übersetzt bekommen. Genau diesen Frust wollen wir unseren Benutzern nicht zumuten; für die Interessierten unter Ihnen gibt es die OpenSource-Version Z88 V13.0A, die jeweils extra für Windows (mit der WinApi) und LINUX/Mac OS X (mit GTK+) programmiert ist: Die Rechenmodule und das Grafikgrundsystem sind denen von Z88Aurora sehr ähnlich und lassen sich ohne Weiteres für Windows, Linux und Mac OS X übersetzen.

- **Warum ist Z88Aurora für Mac OS X nicht mit dem typischen "Mac-Look-and-feel" programmiert, wie man das in der Apple-Welt erwartet? Können Sie das nicht ?**

Wir würden schon können, aber: Das typische und sehr schöne "Mac-Look-and-feel" setzt als Programmier-Paradigma OpenGL, die sog. Cocoa-API und als Sprache Objective-C voraus. OpenGL ist kein Thema, aber die Cocoa-API basiert auf dem früheren Next-Step und ist sehr deutlich anders als die WinAPI oder GTK+. Objective-C ist ein C mit Objektorientierung, meines Erachtens gelungener als C++, aber es ist eben weder C noch C++.

Da Apple schon seit geraumer Zeit ein X11-System in Mac OS X integriert hat, verwenden wir dies und sparen uns eine komplette Neubearbeitung. Aber selbst dann sind in Z88Aurora für Mac eine ganze Menge Mac-Spezifika einprogrammiert.

Ferner müssen sich auf diese Art Benutzer, die auf mehreren Betriebssystemen arbeiten (z.B. Privat-Mac, Dienst-PC) nicht jedes Mal umgewöhnen.

- **Warum verwenden Sie für die Programmierung denn C und OpenGL und nicht die "modernerer" C#, .NET und DirectX?**

Wenn es um pure Rechenpower geht, sind C und FORTRAN nach wie vor ungeschlagen. Im Augenblick (vielleicht ändert sich das in der Zukunft) brauchen nach meinen Untersuchungen Sprachen wie C# bzw. Java mindestens doppelt so lange für gleichartigen Programmcode wie C - ich habe das versuchsweise an mehreren Z88Aurora-Moduln intensiv unter-

sucht. DirectX ist von der inneren Struktur OpenGL ziemlich ähnlich, sodass wir damit kein Problem hätten. Vorteile würde DirectX gegenüber OpenGL bei unseren Aufgaben nicht bringen. Momentan scheint es so zu sein, dass DirectX mehr den Computerspielmarkt dominiert, wohingehend OpenGL die Nummer 1 bei der technischen Grafik ist. Und mit WebGL (das ein abgespecktes OpenGL ist) wird damit auch der riesige Markt der Smartphones erfasst.

- **Warum verwenden Sie GTK+?**

Wenn man eine derartig umfangreiche Software programmiert, gibt es eigentlich drei Paradigmen: GTK+, .NET oder Qt. Die vierte, die sog. GLUT-Library, ist eigentlich keine Option, da sie nur zum Lernen von OpenGL, aber nicht für das Schreiben von hochwertigen OpenGL/GUI-Programmen gedacht ist (wenngleich das manche machen). Bei den drei anderen ist es so:

- GTK+ ist Open Source, ist in C geschrieben und existiert für Windows und X11. Durch die C-Schnittstelle ist es mit einer zwischengeschalteten Bibliothek wie z.B. gtkglext (die wir aus mehreren Gründen nicht verwenden) sehr einfach, OpenGL anzusteuern.
- .NET ist Eigentum von Microsoft, wird typischerweise mit C# angesteuert und ist für Windows gedacht, obwohl das Mono-Projekt Beachtliches leistet. OpenGL-Ansteuerung ist nicht einfach, DirectX-Ansteuerung aber auch nicht.
- Qt ist sehr leistungsfähig, aber die rechtliche Situation ist uns nicht ganz klar bzw. könnte in Zukunft u.U. unklar werden - vorsichtig formuliert.