

SDL (Version 2.x) mit Windows und Dev-C++ (ab Version 5.2)

Klaus Kusche, 2016

Zutaten

Du brauchst 4 Dinge, die du bei mir zusammengezippt in einen File herunterladen kannst:

- Den Ordner “SDL2”, der die **SDL-Header-Files** für den Compiler enthält.
- Die **SDL Library** zum Linken deines Programmes: **libSDL2.dll.a**
- Die **SDL DLL** zum Ausführen deines **.exe**-Files mit SDL-Grafik: **SDL2.dll**
- Die beiden von mir programmierten Files **sdllinterf.c** und **sdllinterf.h**, die eine **Zwischenschicht** zwischen deinem Programm und der SDL bilden, damit du nicht direkt auf die SDL-Funktionen zugreifen musst. In **sdllinterf.h** ist dokumentiert, wie du meine Funktionen für die Grafik aufrufst.

Anlegen und Compilieren von SDL-Programmen

- Leg in Windows einen **neuen Ordner** an, und zwar **ohne Umlaute und Zwischenräume** im Ordernamen und im gesamten Pfadnamen zum Ordner!
- Leg in Dev-C++ ein **neues Projekt** an (“Datei” | “Neu” | “Projekt...”) (keinen einzelnen neuen Quellfile, denn zu unserem Programm gehören ja jetzt mehrere **.c**- und **.h**-Files!), und zwar als Konsol-Projekt (“Console Application”) (wir wollen ja Ausgaben und Fehlermeldungen zur Laufzeit im DOS-Fenster sehen!).
Der Projektname (wird auch Name des **.exe**-Files!) und alle im Projekt verwendeten Dateinamen sollen ebenfalls keine Umlaute oder Zwischenräume enthalten.
Stelle als Default-Sprache C ein, wenn das Projekt keinen einzigen C++-File enthält, und stelle C++ ein, wenn du ein C++-Programm schreiben willst!
Gib als Speicherort für das Projekt den **neu angelegten (leeren) Ordner** an.
- **Kopiere alle oben genannten Dateien** und den Header-Ordner **SDL2** (also den Inhalt meines Zip-Files) in deinen neuen Projekt-Ordner (und ev. auch die Musterlösung oder Demo, die du zum Ausprobieren verwenden willst, also z.B. **demo0.c**).
- Du kannst entweder im Editor den Code der Demo oder Musterlösung in das automatisch erzeugte **main.c** hineinkopieren oder das automatisch erzeugte **main.c** aus dem Projekt entfernen (links in der Projekt-Ansicht: Rechtsklick auf **main.c**, “Datei entfernen”) und stattdessen den Demo-**.c**-File ins Projekt aufnehmen (links in der Projekt-Ansicht: Rechtsklick auf das Projekt selbst, “Zum Projekt hinzufügen”).
Die Operationen “Datei aus Projekt entfernen” und “Datei zum Projekt hinzufügen” stehen bei den bei uns installierten DevCpp-Versionen im Rechtsklick-Menü der Projekt-Ansicht. In neueren Dev-Cpp-Versionen findet man sie nicht mehr im Rechtsklick-Menü, sondern es sind eigene Buttons in der Toolbar.
- Füge auch **sdllinterf.c** und **.h** (und ev. weitere eigene .c / .h-Files) zum Projekt dazu, indem du links in der Projekt-Ansicht einen Rechtsklick auf das Projekt selbst machst und “Zum Projekt hinzufügen” wählst.
Die SDL-**.h**-Files aus dem Ordner **SDL2** brauchst du nicht zu deinem Projekt dazutun!

- Dein C/C++-Programm muss ein **#include** auf meinen **sdllinterf.h** machen (mit "**sdllinterf.h**", nicht mit **<sdllinterf.h>**!).
- Geh in den **Projekteigenschaften** (links in der Projekt-Ansicht: Rechtsklick auf das Projekt selbst, "Projekt Optionen") auf das vierte Karteiblatt "Parameter". Klick dort auf den Button "Bibliothek / Objekt hinzufügen" und wähle die **libSDL2.dll.a** aus. Sie sollte dann im Feld "Linker" angezeigt werden.
- Mit diesen Einstellungen sollte sich eine Musterlösung bzw. dein Programm fehlerfrei compilieren, linken und starten lassen.

Hinweise

- Diese Anleitung gilt für meinen Zip-File, der Version 2 der SDL ("SDL2") enthält. Für Version 1 gibt es eine andere Anleitung und einen anderen Zip-File.

Der Header sdllinterf.h ist gleich geblieben, d.h. dein Programm bzw. die Musterlösungen funktionieren unverändert mit Version 1 und Version 2.

Die SDL-Anbindung in **sdllinterf.c** hat sich hingegen stark geändert.

Du musst also die zur jeweiligen SDL-Version gehörende Variante von **sdllinterf.c** verwenden und das gesamte Projekt frisch compilieren und linken.

- Wenn du dein Programm im DOS-Fenster starten willst:
Der **.exe**-File eines Projekts heißt so wie das Projekt, nicht so wie dein **.c**-File.
Der **.exe** läuft nur, wenn der File **SDL2.dll** im selben Ordner liegt! (siehe unten)
- Wenn du ein bestehendes Projekt im Dev-C++ zum Weiterarbeiten öffnen willst, musst du den **.dev-Projektfile doppelklicken** bzw. im Dev-C++ öffnen, nicht den **.c**-File!
- Mein "offizieller" Zip-File enthält die 32-Bit-Versionen von **libSDL2.dll.a** und **SDL2.dll**. Ob man die 32-Bit-Version oder die 64-Bit-Version verwenden muss, hängt nicht vom Windows ab, sondern vom Compiler im Dev-C++.

Für SDL-Programme muss daher im Dev-C++ auch unter 64 Bit Windows der 32 Bit Compiler ("**MinGW32**" oder die 32-Bit-Version des **TDM-GCC**) verwendet werden, nicht der 64 Bit Compiler ("**TDM-GCC x64**").

Passen Compiler und SDL-Files nicht zusammen, liefert der Linker seltsame Fehlermeldungen.

Eine vollständige 64-bit-Dev-C++-Installation installiert beide Compiler.

Der für das Projekt zu verwendende Compiler wird in den Projekteigenschaften (nicht in den allgemeinen Dev-C++ Compilereinstellungen!) eingestellt.

Stelle dort den 32-bit-Compiler (Release) ein!

Wenn du den 64-bit-Compiler verwenden willst:

Auf meiner Webseite (C-Programmierung / SDL) ist auch eine Version des Zip-Files für den 64 Bit Compiler, aber sie ist nicht getestet.

- Wenn du einen **.exe**-File bauen willst, den man kopieren und verwenden kann, ohne die **SDL2.dll** mitzukopieren:
 - 1.) Lade die **libSDL2-32.a** (oder die 64 Bit Version) von meiner Webseite herunter (<http://www.computerix.info/prog-gra.html>, ziemlich weit unten), gib sie in deinen Projektordner, und trage sie in den Projekteigenschaften bei den Linker-Parametern **statt** der **libSDL2.dll.a** ein (**libSDL2.dll.a** rauslöschen!).

2.) Trage zusätzlich folgende Zeile in den Projekteigenschaften bei den Parametern im Feld "Linker" ein:

```
-static -lmingw32 -mwindows -lm -ldinput8 -ldxguid -ldxerr8  
-luser32 -lgdi32 -lwinmm -limm32 -lole32 -loleaut32 -lshell32  
-lversion -luuid -static-libgcc -static-libstdc++
```

Wenn du das Projekt jetzt neu kompilierst, sollte ein sehr großer (einige MB) **.exe**-File entstehen, der ohne **.dll**'s funktioniert.