Using SDL2 with eForth Windows

version 1.0 - dimanche 27 octobre 2024

Autor

- Marc PETREMANN

Contents

Installing eForth and SDL2

Installing the Forth language development environment for SDL2 requires only two components:

- Brad NELSON's eForth version for Windows;
- the SDL2 library in a dll file;
- A good text file editor.

It is a very compact environment which combines, thanks to eForth, an interpreter and a compiler.

Installing eForth for Windows

The eForth version for Windows is available here:
https://eforth.appspot.com/windows.html

Download the uEf64-7.0.7.20.exe version. This is a 64-bit version. It is very stable and very robust. The file is only 265 KB.

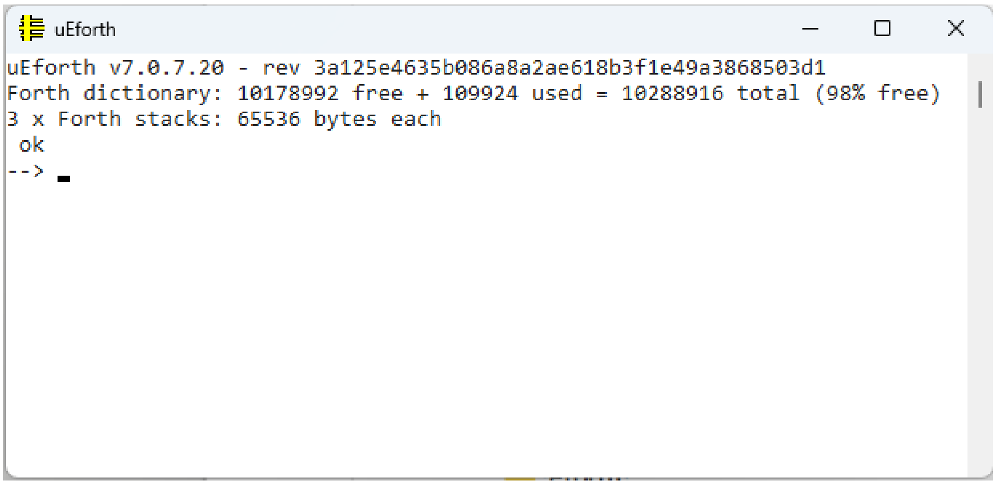
Create an eforth folder in your usual workspace:

📁 eforth

Copy the previously downloaded file into this folder:

📁 eforth
    📄 uEf64-7.0.7.20.exe

Run this program from this **eforth folder** . Often, Windows issues a warning. If so, accept the execution of this program. You should end up with this window:



```
uEforth v7.0.7.20 - rev 3a125e4635b086a8a2ae618b3f1e49a3868503d1
Forth dictionary: 10178992 free + 109924 used = 10288916 total (98% free)
3 x Forth stacks: 65536 bytes each
 ok
-->
```

Figure 1: eForth window

There you have it! You have your hands on a Forth version with three stacks:

- a stack of data
- a return stack
- a stack for real numbers

Each stack has a data space of 64KB. Since each element weighs 64 bits (8 bytes) in the stack, that's 8000 values that can be stacked!

The dictionary has 10,178,992 bytes of free space! So we have more than enough development space.

eForth is an interpreter AND a compiler. The very first word at your disposal, just to see what's under the hood, is `words,` which when executed displays this, summarized in the first three lines:

```
FORTH graphics argv argc visual set-title page at-xy normal bg fg ansi
editor list copy thru load flush update empty-buffers buffer block save-
buffers
default-use use open-blocks block-id scr block-fid file-exists? needs
required…
...etc.
```

All these words make up the language for compiling and running programs written in the Forth language.

Installing the SDL2 library

Access to the SDL2 site: https://www.libsdl.org/

This site provides a lot of resources and documentation to understand the use of the SDL2 library. WARNING: the C language functions are adapted to eForth Windows. We will see this later.

To download the latest SDL2 version:
https://github.com/libsdl-org/SDL/releases/tag/release-2.30.8

Get the **SDL2-2.30.8-win32-x64.zip file** .

Open this zip file and transfer the single **SDL2.dll file** to the eforth directory:

```
🗀 eforth
      🗎 uEf64-7.0.7.20.exe
      🗎 SDL2.dll
```

That's it! There's nothing else to do on the installation side. Now let's see how to prepare the development environment.

File organization

SDL2 subdirectory :

- 🗀 eforth
  - 🗎 uEf64-7.0.7.20.exe
  - 🗎 SDL2.dll
  - 🗀 SDL2

We will fill this directory with the files available here:
https://github.com/MPETREMANN11/SDL2-eForth-windows/tree/main/SDL2

Retrieve only these files:

- 🗀 eforth
  - 🗎 uEf64-7.0.7.20.exe
  - 🗎 SDL2.dll
  - 🗀 SDL2
    - 🗎 SDL2.fs
    - 🗎 SDLconstants.fs
    - 🗎 main.fs
    - 🗎 tests.fs

Contents of these files:

- **SDL2.fs** contains all the words accessing the SDL library contained in **SDL2.dll** ;
- **SDLconstants.fs** contains a number of commonly used constants specific to their use with words defined in the `SDL2` vocabulary ;
- **main.fs** is the main script responsible for aggregating the various components of your application;
- **tests.fs** catch-all file for performing tests.

WARNING: The content of these files is likely to change constantly on the Github repository. It is therefore strongly recommended to monitor their content.

There is one last file to install in the **eforth directory** :

- 🗀 eforth
  - 🗎 uEf64-7.0.7.20.exe
  - 🗎 SDL2.dll
  - 🗀 SDL2
    - 🗎 SDL2.fs
    - 🗎 SDLconstants.fs
    - 🗎 main.fs
    - 🗎 tests. fs
  - 🗎 SDL2.fs

Contents of this **SDL2.fs file**

```
\ pre-load tools
\ s" tools/dumpTool.fs" required
\ load SDL2
s" SDL2/main.fs" included
```

Consider the contents of this file as a batch process, but executable only from eForth.

To check if the SDL2 library is working properly, launch eForth, then enter this command:

```
include SDL2.fs
```

This command loads the contents of the **SDL2.fs** file located in the root of the eforth subdirectory . Then type:

```
SDL2 vlist
```

You should see the words defined in the SDL2 vocabulary appear, here the first three lines:

```
SDL.CreateWindow SDL.init SetRenderDrawColor Quit RenderPresent RenderClear
PollEvent Init GetError GetCursor GetCPUCount GetBasePath GetAudioStatus
DestroyWindow DestroyRenderer CreateWindow CreateRenderer
SDL_MAX_LOG_MESSAGE
...etc…
```

ATTENTION: The content of this vocabulary is constantly evolving. It only represents the compilation of the definitions described in the **SDL2/SDL2.fs** file .

The main.fs file

This is the file containing the script responsible for aggregating the components of your applications. Example of the content of this file:

```
\ load SDL2 library
s" SDL2.fs"     included

\ load SDL2 tests
s" tests.fs"    included
```

You can modify its content without any worries. Let's take the case where you want to test the display of windows. You create a file **testWindows.fs** in which you will put the window tests with the words of the `SDL2` vocabulary. Here is how to integrate this file **testWindows.fs** into **main.fs** :

```
\ load SDL2 library
s" SDL2.fs"     included

\ load SDL2 tests
\ s" tests.fs"    included

\ load windows tests with SDL2
s" testWindowss.fs" included
```

The word \ comments out the rest of the line. The next time **SDL2.fs** is loaded which is in the **eforth** root directory , only the contents of the files **SDL2/SDL2.fs** and **SDL2/testWindows.fs** will be processed by eForth.

So you can easily chain tests or portions of code together for the final application.

To avoid accidentally overwriting your work, it is advisable to create several subdirectories, for example sandbox where you can do lots of small tests:

- 🗀 eforth
    - 🗎 uEf64-7.0.7.20.exe
    - 🗎 SDL2.dll
    - 🗀 SDL2
        - 🗎 SDL2.fs
        - 🗎 SDLconstants.fs
        - 🗎 main.fs
        - 🗎 tests. fs
    - 🗎 SDL2. fs
    - 🗀 sandbox
        - 🗎 SDL2.fs
        - 🗎 SDLconstants.fs
        - 🗎 main.fs
        - 🗎 tests. fs
    - 🗎 sandbox.fs

Example of the contents of **sandbox.fs** :

```
\ s" tools/dumpTool.fs" required
\ load sandbox
s" sandbox/main.fs" included
```

So, when launching eForth, you just need to enter:

```
include sandbox.fs
```

It's up to you to organize yourself to be as efficient as possible during your developments.

Ressources

- **eForth for Windows**
  project created and maintained by Brad NELSON
  https://eforth.appspot.com/windows.html
- **Analyseur de code FORTH**
  transforms undocumented FORTH code into a version with syntax highlighting and hyperlinks to known words.
  https://analyzer.arduino-forth.com/

GitHub

- **SDL2 eForth windows project**
  SDL2 project codes and documentations
  https://github.com/frenchie68/Z79Forth

SDL

- **Simple DirectMedia Layer**
  SDL library development platform providing access to audio, keyboard, mouse and graphics hardware resources via OpenGL and Direct3D.
  https://www.libsdl.org/

Index