

DevC++ e libreria SDL per la grafica

1 RIFERIMENTI

https://lazyfoo.net/SDL_tutorials/lesson01/windows/devcpp/index.php

<https://thenumbat.github.io/cpp-course/sdl2/04/04.html>

2 CONFIGURAZIONE AMBIENTE

2.1 DEV CPP 4.9.2

Scaricare [devcpp 4.9.2](#) (versione [portabile](#) o quella non portabile a seconda delle esigenze).

Devcpp 4.9.2 è completamente libero. Le versioni successive hanno molte nuove funzioni per l'editing, il completamento etc ma hanno problem con il debugger. Possono essere utilizzate ma a proprio rischio e pericolo.

Io ho fatto tutto con la versione Portable di DevCpp per poterla facilmente distribuire.

Scompattare la versione portabile in una directory, ad esempio

c:/ProgrammiPortabili/DevCpp4.9.2

I setting di default dovrebbero andare bene.

Può essere comodo creare un link al programma DevcppPortable.exe sul desktop e associarlo ai file .dev.

2.2 SDL2

Scaricare la libreria grafica [SDL](#) (nel mio caso la versione SDL2-2.0.14). Va presa la versione per Windows MinGW32:



Main	
About	•
Bugs	•
Licensing	•
Credits	•
Feedback	•
Documentation	
Wiki	•
Forums	•
Mailing Lists	•
Download	
SDL Releases	•
SDL Mercurial	•
Bindings	•

SDL version 2.0.14 (stable)

Source Code:

[SDL2-2.0.14.zip](#) - GPG signed
[SDL2-2.0.14.tar.gz](#) - GPG signed

GPG Signature:

The source code to this release has been signed by Sam Lantinga.

You can get the public key from any keyserver with the key id 0xA7763BE6, or directly from Sam's home page: [sloouken-pubkey.asc](#)

The public key fingerprint should be:

pub 10240/A7763BE6 2001-01-05 Sam Lantinga <sloouken@litesdl.org>
Key Fingerprint = 1528 63D2 8053 A579 77D1 E886 30A5 9377 A776 38E6

For more information about public key signatures, see <http://www.gnupg.org/>

Runtime Binaries:

Windows:
[SDL2-2.0.14-win32-x86.zip](#) (32-bit Windows)
[SDL2-2.0.14-win32-x64.zip](#) (64-bit Windows)

Mac OS X:
[SDL2-2.0.14.dmg](#)

Linux:
Please contact your distribution maintainer for updates.

Development Libraries:

Windows:
[SDL2-devel-2.0.14-VC.zip](#) (Visual C++ 32/64-bit)
[SDL2-devel-2.0.14-mingw.tar.gz](#) (MinGW 32/64-bit)

Mac OS X:
[SDL2-2.0.14.dmg](#)

Linux:
Please contact your distribution maintainer for updates.

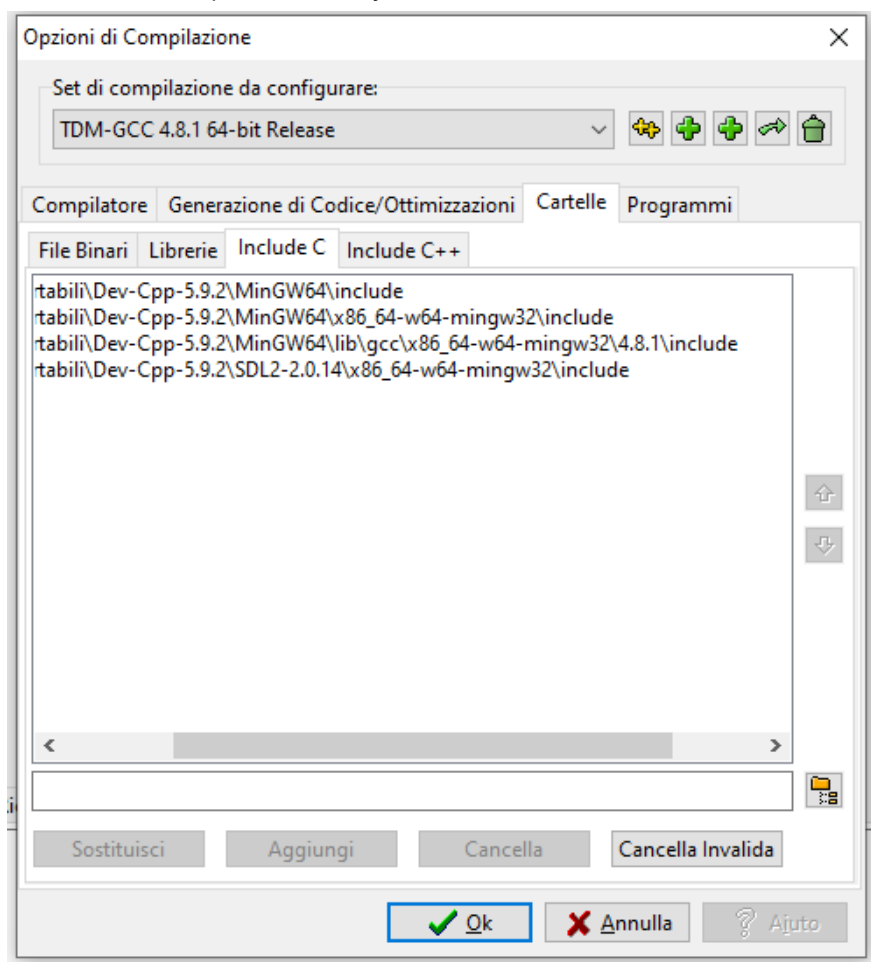
iOS & Android:
Projects for these platforms are included with the [source](#).

Scompattare SDL in una directory comoda, io l'ho messa come sottodirectory di quella dove ho installato DevCpp.

c:\ProgrammiPortabili\Dev-Cpp-5.9.2*.*			
Nome	Est.	Dimens.	Data/Ora
[.]		<DIR>	26/02/20
[AStyle]		<DIR>	26/02/20
[config]		<DIR>	26/02/20
[Help]		<DIR>	26/02/20
[Icons]		<DIR>	26/02/20
[Lang]		<DIR>	26/02/20
[MinGW64]		<DIR>	26/02/20
[SDL2-2.0.14]		<DIR>	21/12/20
[Templates]		<DIR>	26/02/20
ConsolePauser	exe	171.008	13/11/20
copying	txt	18.581	09/02/20
devcpp	map	1.584.205	01/02/20
devcpp.exe	manifest	695	23/08/20
devcppPortable	exe	174.592	08/12/20
NEWS	txt	59.653	01/02/20
NON-USARE-devcpp	exe	3.209.728	01/02/20
Packman	exe	380.928	19/02/20
Packman	map	697.941	19/02/20
SDL-Template	zip	633.451	26/02/20

Avviare DevCpp e aprire Strumenti / Opzioni di compilazione / scheda Cartelle.

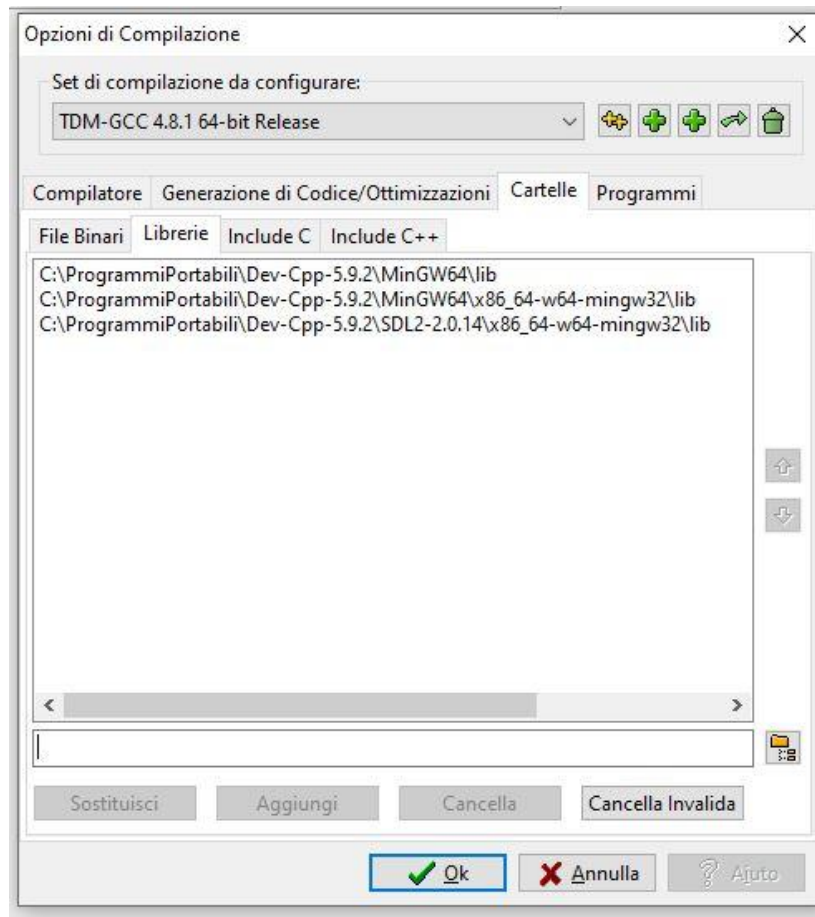
Aggiungere (finestrella in basso) la directory include nelle schede **Include C** e **Include C++**.



Prendere i file dalla sottocartella di SDL **x86_64-w64-mingw32**

c [os] 89.622.184 k di 973.475.796 k liberi			
▼ \AppData\Local\Temp_tc\SDL2-devel-2.0.14-mingw.tar\SDL2-2.0.14\.* *			
Nome	Est.	Dimens.	↓ Data/Ora
↑ [..]		<DIR>	21/12/2020 18:48
[docs]		<DIR>	21/12/2020 18:48
[i686-w64-mingw32]		<DIR>	31/10/2018 16:06
[test]		<DIR>	21/12/2020 18:48
[x86_64-w64-mingw32]		<DIR>	31/10/2018 16:06
BUGS	txt	463	21/12/2020 18:48
COPYING	txt	930	21/12/2020 18:48
CREDITS	txt	1.920	21/12/2020 18:48
README	txt	648	21/12/2020 18:48
README-SDL	txt	433	21/12/2020 18:48
WhatsNew	txt	35.930	21/12/2020 18:48
Makefile		1.161	29/01/2018 07:12
INSTALL	txt	489	05/09/2017 08:10

Fare lo stesso con le la scheda Librerie



3 CREAZIONE DI UN PROGETTO DI BASE

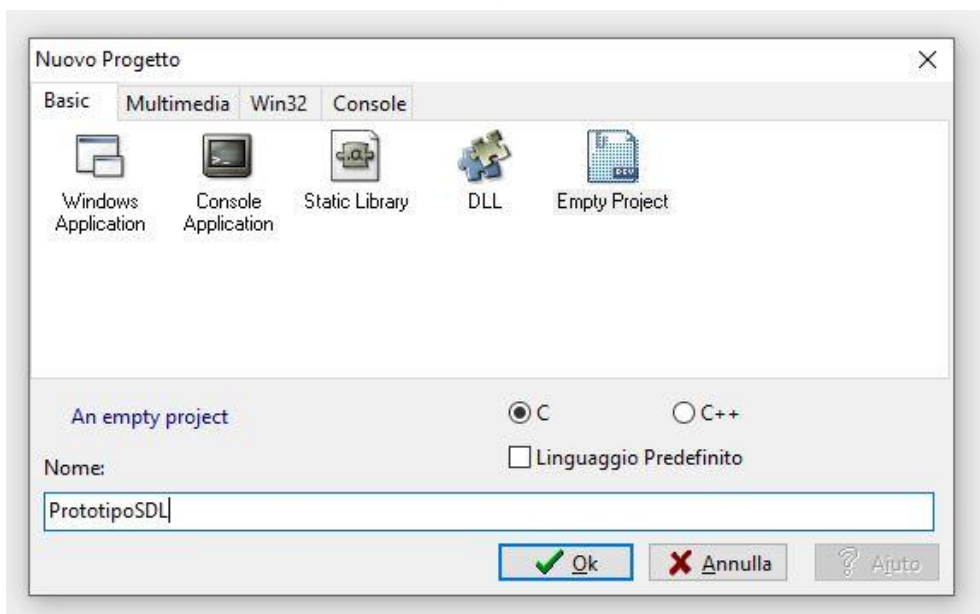
Io ho creato un template nella cartella ProgrammiPortabili/DevCpp4.9.2/Templates di DevCpp, ad esempio SDL-Template

c:\ProgrammiPortabili\Dev-Cpp-5.9.2\Templates*			
Nome	Est	Dimens.	Data/Ora
[.]		<DIR>	26/02/2021 18
[SDL-Template]		<DIR>	26/02/2021 19
1-WinApp	template	234	28/02/2013 17
2-ConsoleApp	template	254	27/02/2013 22
3-StaticLib	template	174	07/07/2014 27
4-DLL	template	340	07/07/2014 27
5-Empty	template	164	15/12/2013 14
ball	bmp	630	06/03/2002 03
ballmask	bmp	190	06/03/2002 03
ConsoleApp_c	txt	196	21/04/2012 23
ConsoleApp_cpp	txt	175	24/02/2013 17
Direct3D	template	220	28/02/2013 17

Copiare nella directory ProgrammiPortabili/DevCpp4.9.2/Templates/SDL-Template la libreria **SDL2.dll** presa dalla sottodir bin di SDL.

Nome	Est	Dimens.	Data/Ora
[.]		<DIR>	21/12/2020 18:48
SDL2	dll	1.561.088	21/12/2020 18:48
sdl2-config		1.443	21/12/2020 18:47

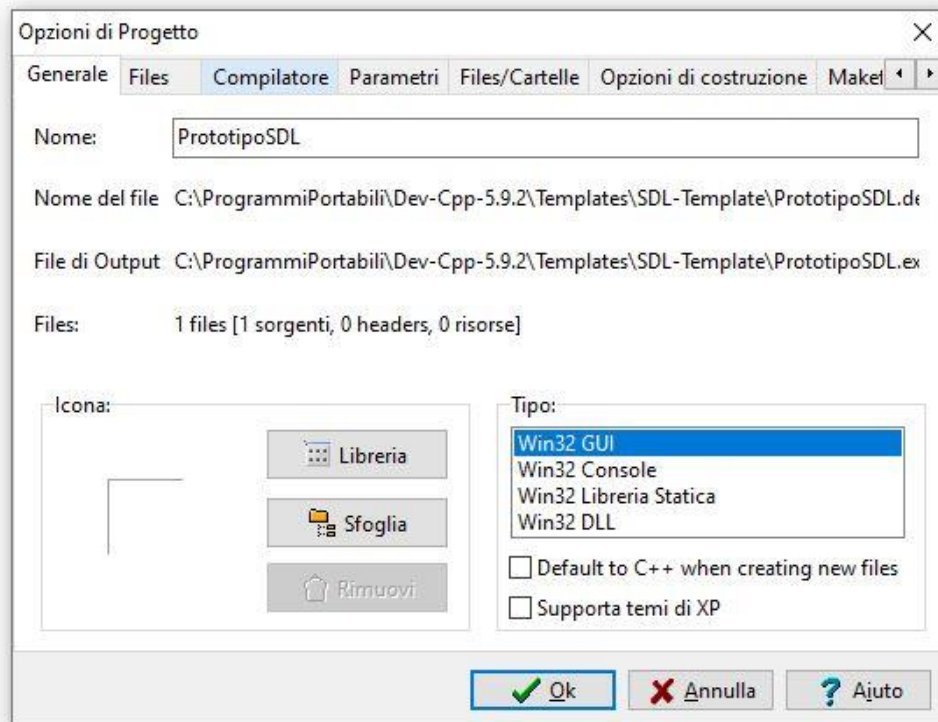
Lanciare **DevCppPortable.exe** e creare un progetto vuoto denominato **PrototipoSDL**



Selezionare Progetto / Opzioni del progetto



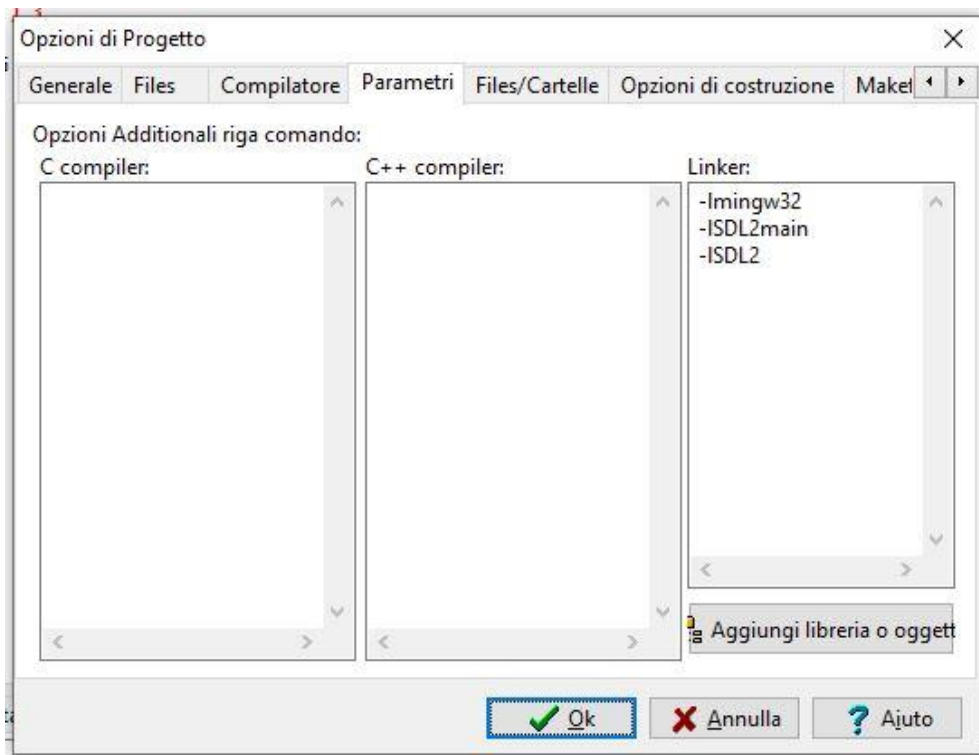
E mettere come tipo Win32GUI



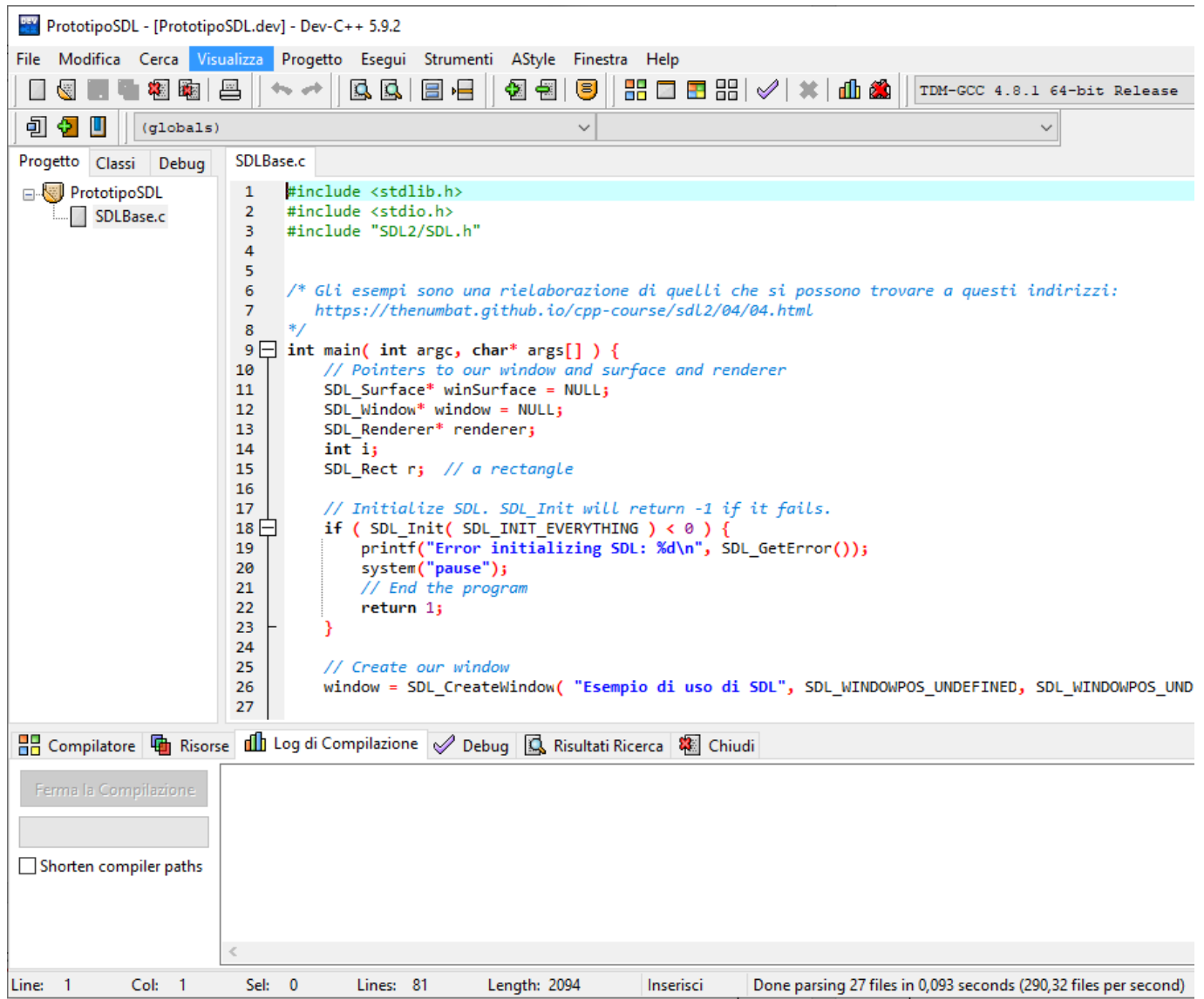
Selezionare le Opzioni del progetto, la scheda Parametri e aggiungere le tre righe (forse si può mettere anche come unica stringa) come in figura:

-lmingw32
-lSDL2main
-lSDL2

ATTENZIONE: sul tutorial in inglese mancano i "2".



Creare un file sorgente, nominarlo **SDLBase.c** e copiare il codice seguente



Il codice disegna alcuni punti, una linea e un rettangolo.

```
#include <stdlib.h>
#include <stdio.h>
#include "SDL2/SDL.h"

/* Gli esempi sono una rielaborazione di quelli che si possono trovare a questi indirizzi:
   https://thenumbat.github.io/cpp-course/sdl2/04/04.html
*/
int main( int argc, char* args[] ) {
    // Pointers to our window and surface and renderer
    SDL_Surface* winSurface = NULL;
    SDL_Window* window = NULL;
    SDL_Renderer* renderer;
    int i;
    SDL_Rect r; // a rectangle
```



```

// Initialize SDL. SDL_Init will return -1 if it fails.
if ( SDL_Init( SDL_INIT_EVERYTHING ) < 0 ) {
    printf("Error initializing SDL: %d\n", SDL_GetError());
    system("pause");
    // End the program
    return 1;
}

// Create our window
window = SDL_CreateWindow( "Esempio di uso di SDL",
    SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED,
    1280, 720, SDL_WINDOW_SHOWN );

// Make sure creating the window succeeded
if ( !window ) {
    printf("Error creating window: %d\n", SDL_GetError());
    system("pause");
    // End the program
    return 1;
}

renderer = SDL_CreateRenderer( window, -1, SDL_RENDERER_ACCELERATED );
if ( !renderer ) {
    printf("Error creating renderer: %d\n", SDL_GetError());
    system("pause");
    return 1;
}

// Clear the window to white
SDL_SetRenderDrawColor( renderer, 255, 255, 255, 255 );
SDL_RenderClear( renderer );

// Set drawing color to black
SDL_SetRenderDrawColor( renderer, 0, 0, 0, 255 );
for(i=0;i<10;i++)
    SDL_RenderDrawPoint(renderer, 100+10*i, 100 );

// Set drawing color to red
SDL_SetRenderDrawColor( renderer, 255, 0, 0, 255 );
SDL_RenderDrawLine( renderer, 200, 200, 300, 400 );

// Set drawing color to blue
SDL_SetRenderDrawColor( renderer, 0, 0, 255, 255 );
r.x = 600;
r.y = 200;
r.w = 50;
r.h = 100;
SDL_RenderDrawRect( renderer, &r );

// Update window
SDL_RenderPresent( renderer );

```



```
// Wait
system("pause");

// Destroy the renderer and the windows
SDL_DestroyRenderer ( renderer );
SDL_DestroyWindow( window );

// Quit SDL
SDL_Quit();

// End the program
return 0;
}
```

4 CLONAZIONE DEL PROGETTO.

E' possibile clonare il progetto di base ed effettuare tutte le modifiche che necessita.

Una versione compressa di comodità è contenuta nel file SDL-Template.zip messo nella directory di base di DevCpp.