

Building and using the Allegro Graphics & Gaming Library

Table of Contents

Download and install CodeBlocks with MinGW compiler.....	2
Set Environment Variables.....	3
Installing The Allegro Library	4
Download and install Allegro 4.2.2 Source.....	4
Download and install directX sources.....	5
Confirm that the paths and environment variables are correct.....	6
Build Sources and install Allegro library.....	7
Create project “Hello World” example.....	9
Compile, build, and run the HelloWorld project.....	9
Add liballeg.a to linker settings.....	9
Create a new Hello World project with following source.....	10
Final Screen output.....	11
Running the Demo program.....	12

Download and install CodeBlocks with MinGW compiler

Download and install CodeBlocks with MinGW compiler. Click on link, save to a working directory and then run `codeblocks-8.02mingw-setup.exe`.

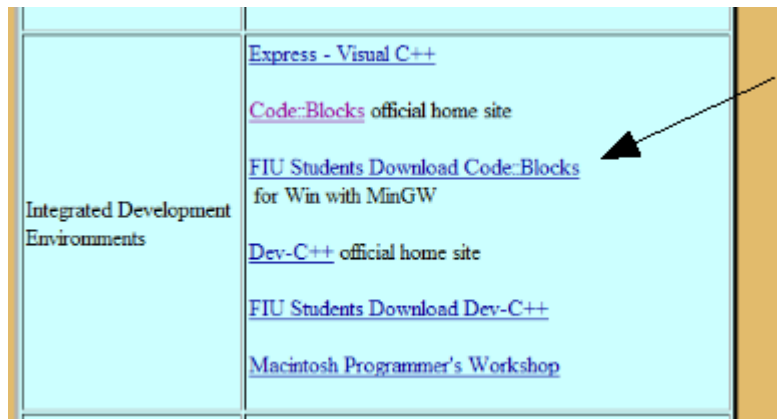


Illustration 1: Class web site selection



Illustration 2: Installation

Install supplies both CodeBlocks and the MinGW gcc compiler in one package.

Information: The MinGW compiler is located in C:\Program Files\CodeBlocks\MinGW\bin.

Set Environment Variables

The final part of the installation involves environment variables.

XP: Click on start, then control panel, then the system icon. Click on the advanced tab and then on the environment variables button.

Vista: Click start, right click Computer, select properties. On the left side of the new window select "advanced system settings". Then on the newest window press Environment Variables.

Two variables will need to be added:

1) MINGDIR set to the path value to (Illustration 4) **C:\Progra~1\CodeBlocks\MinGW**

2) PATH will add the value **C:\Progra~1\CodeBlocks\MinGW\bin**. Note that Path already exists and will then need to be edited/modified. The new path is added after a semicolon. (Illustration 3).

C:\Program Files\CodeBlocks\MinGW\bin is the location of the compiler, but path values cannot have spaces included so non-8dot3 notation is used as shown. This notation is used to specify the Program Files directory without spaces.

Note: Any changes to environment variables will not affect currently opened console windows. You will need to restart the console for the changes to take effect.

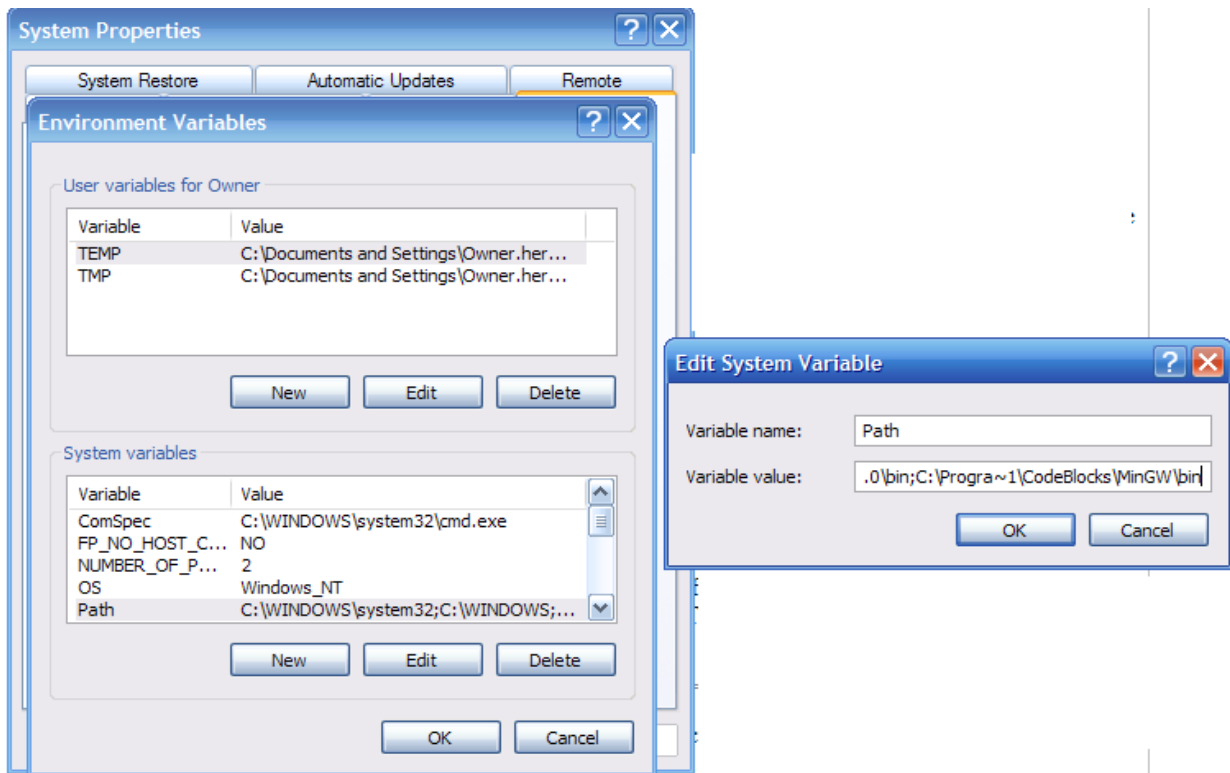


Illustration 3: Modify Path environment variable

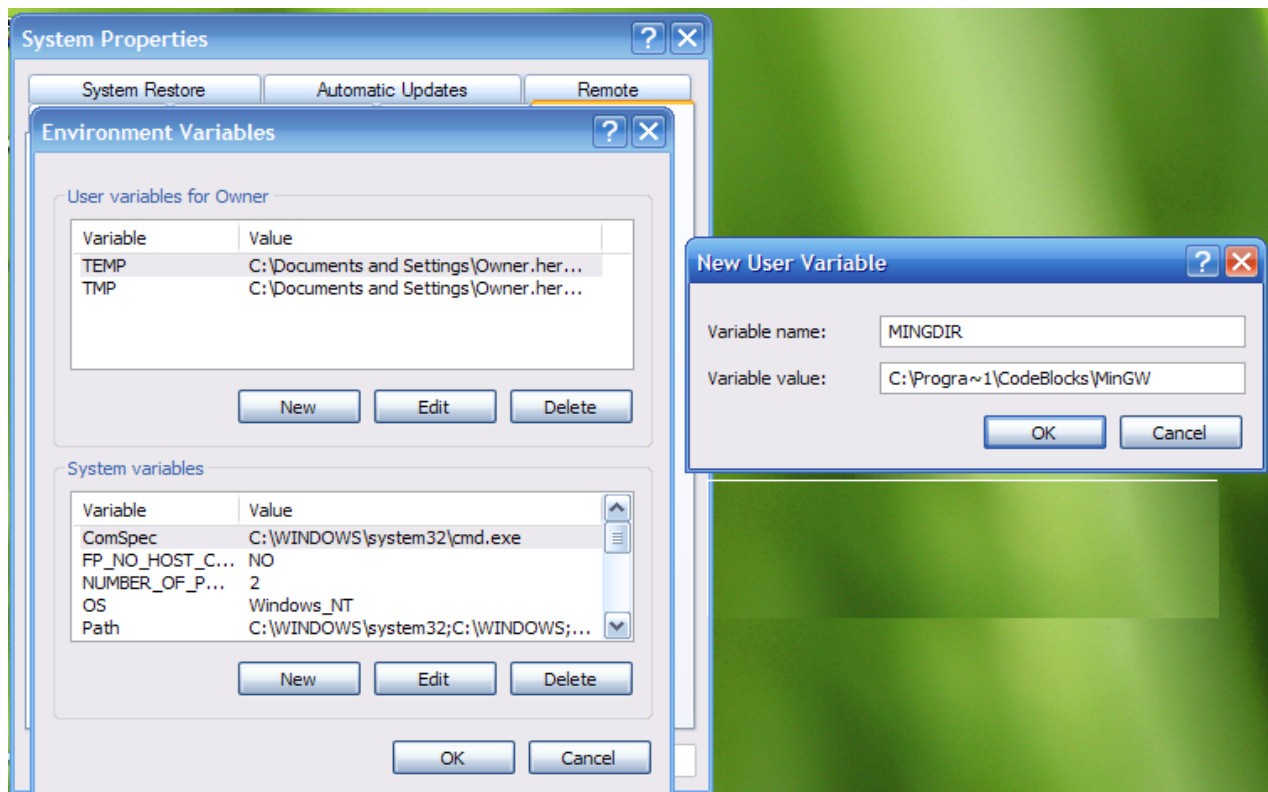


Illustration 4: Add MINGDIR environment variable

Installing The Allegro Library

Download and install Allegro 4.2.2 Source

Download file “all422.zip” from class web site and save in a directory. (Illustration 5)

Unzip all422.zip and allowing it to expand to its own folder Start and follow wizard to unpack into folder.

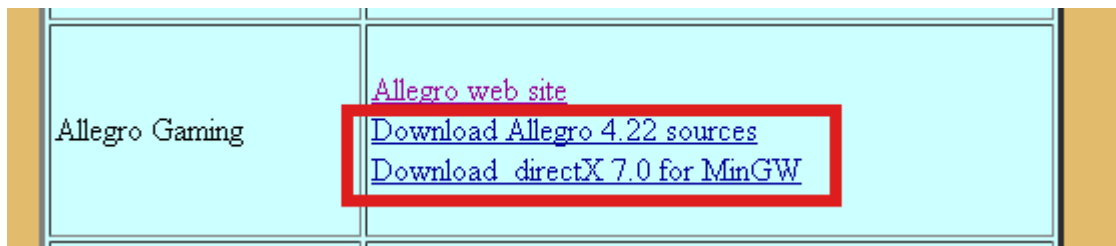


Illustration 5: Class web site source

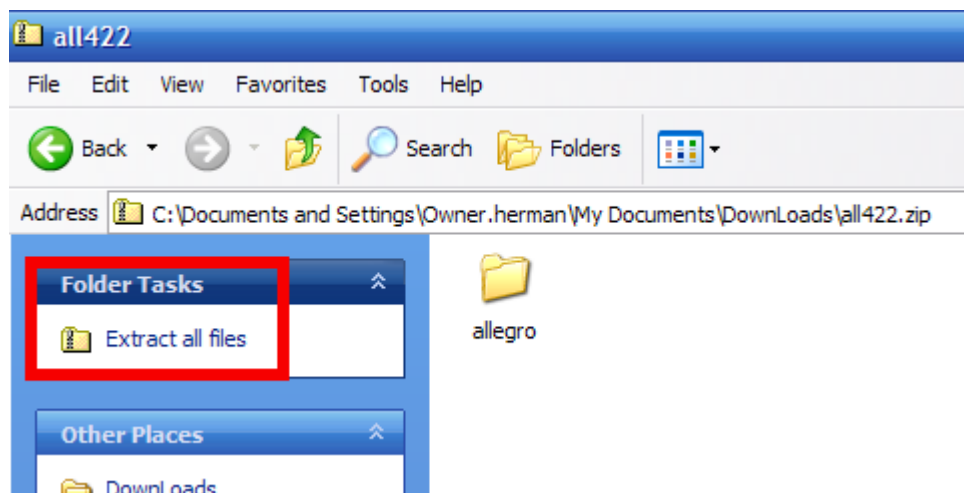


Illustration 6: Unzip all422.zip to folder

Download and install DirectX sources

Download file “dx70_mgw.zip” (Illustration 5) from class web site and save in a directory .

The Windows version of allegro uses this version of DirectX (the standard Windows way of interfacing graphics). Allegro won't compile without it.

NOTE: As shown in Illustration 7, extract the contents to **C:\Program Files\CodeBlocks\MinGW** overwriting all files.

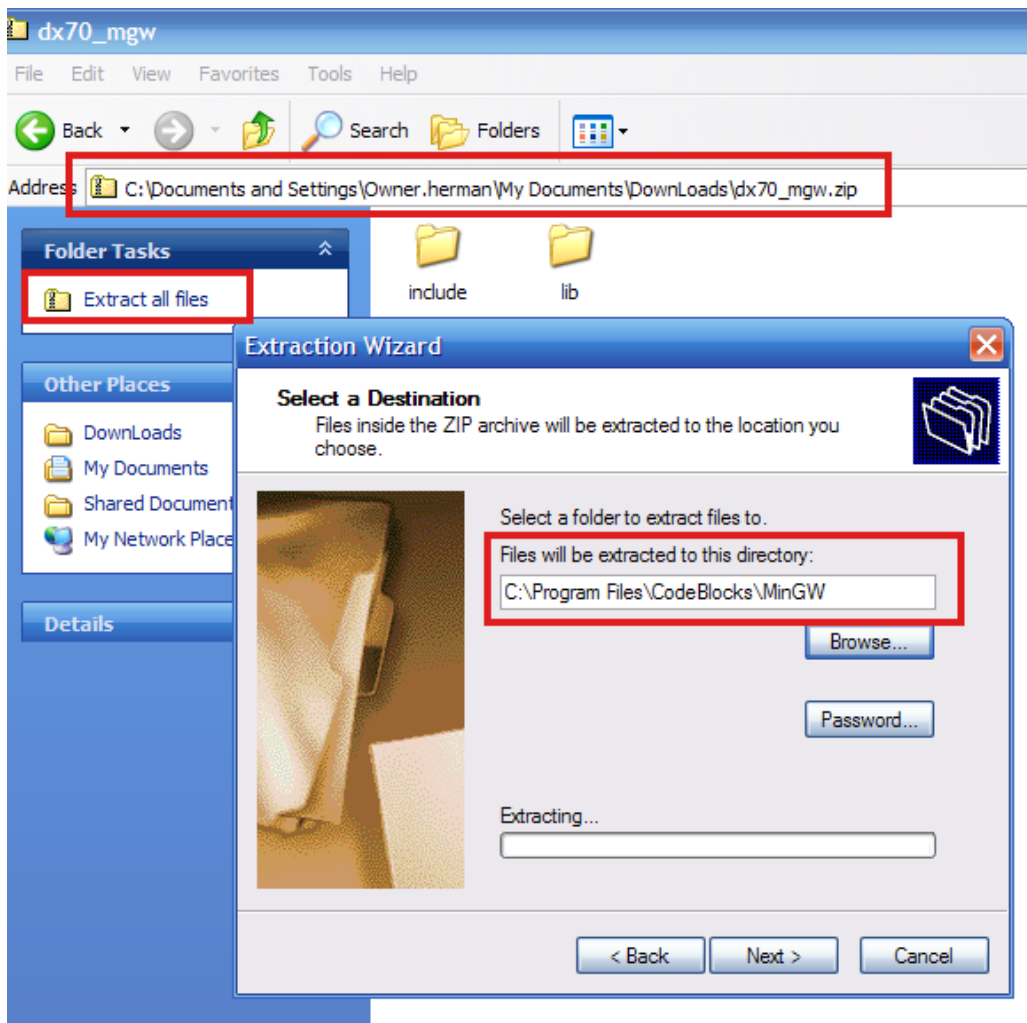
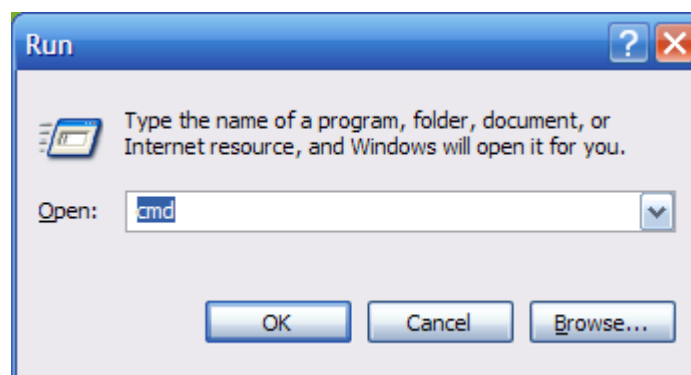


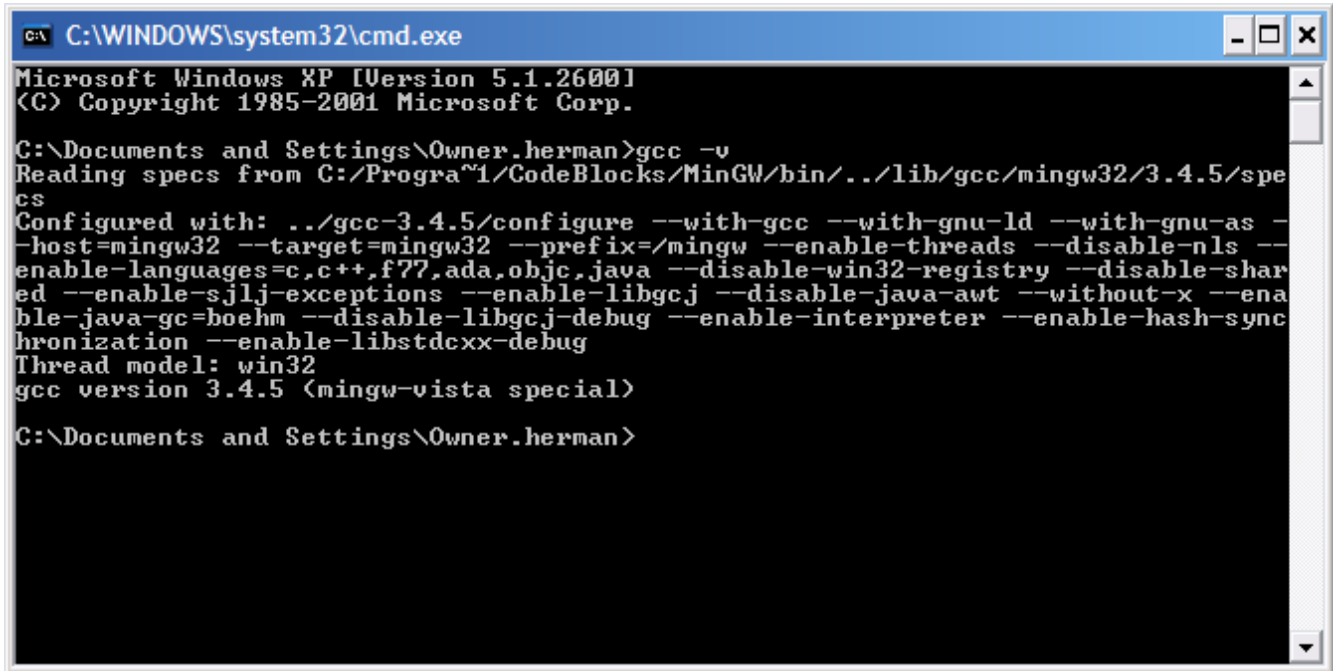
Illustration 7: Extracting directX sources to MinGW directory

Confirm that the paths and environment variables are correct

Start a command window with *Start->Run*



Confirm the gcc compiler is available and working. Type “**gcc -v**” and observe a correct response:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Owner.herman>gcc -v
Reading specs from C:/Progra~1/CodeBlocks/MinGW/bin/./lib/gcc/mingw32/3.4.5/specs
Configured with: ../gcc-3.4.5/configure --with-gcc --with-gnu-ld --with-gnu-as --
-host=mingw32 --target=mingw32 --prefix=/mingw --enable-threads --disable-nls --
enable-languages=c,c++,f77,ada,objc,java --disable-win32-registry --disable-shar
ed --enable-sjlj-exceptions --enable-libgcj --disable-java-awt --without-x --ena
ble-java-gc= Boehm --disable-libgcj-debug --enable-interpreter --enable-hash-syn
chronization --enable-libstdcxx-debug
Thread model: win32
gcc version 3.4.5 (mingw-vista special)

C:\Documents and Settings\Owner.herman>
```

Illustration 8: Confirm gcc is working

Build Sources and install Allegro library

Then inside the command window, **change directory to the unzipped allegro directory.**

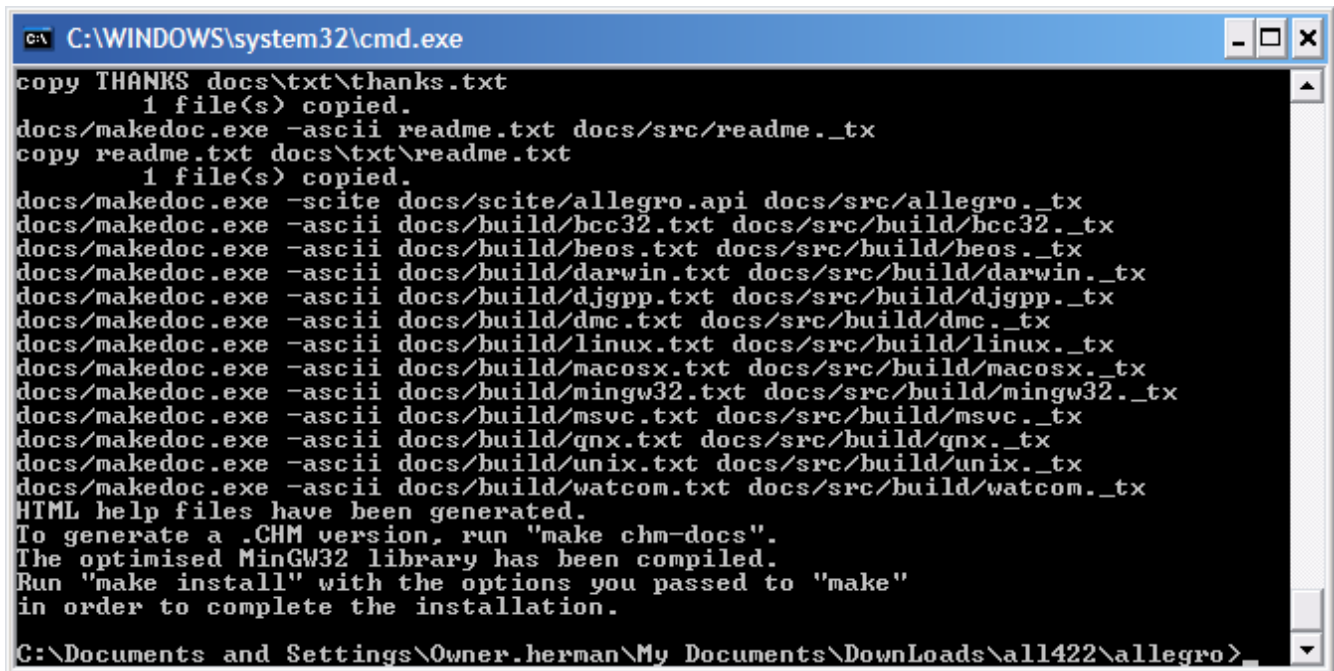
On my computer this is the following:

```
cd C:\Documents and Settings\Owner.herman\My Documents\Downloads\all422\allegro>
```

Now allegro can be compiled. Note that compilation will take some time. Start by typing in the command line:

- **FIX mingw**
- **MINGW32-MAKE**
- **MINGW32-MAKE install**

See Illustration 9 and Illustration 10 below for successful Allegro compile and install results.



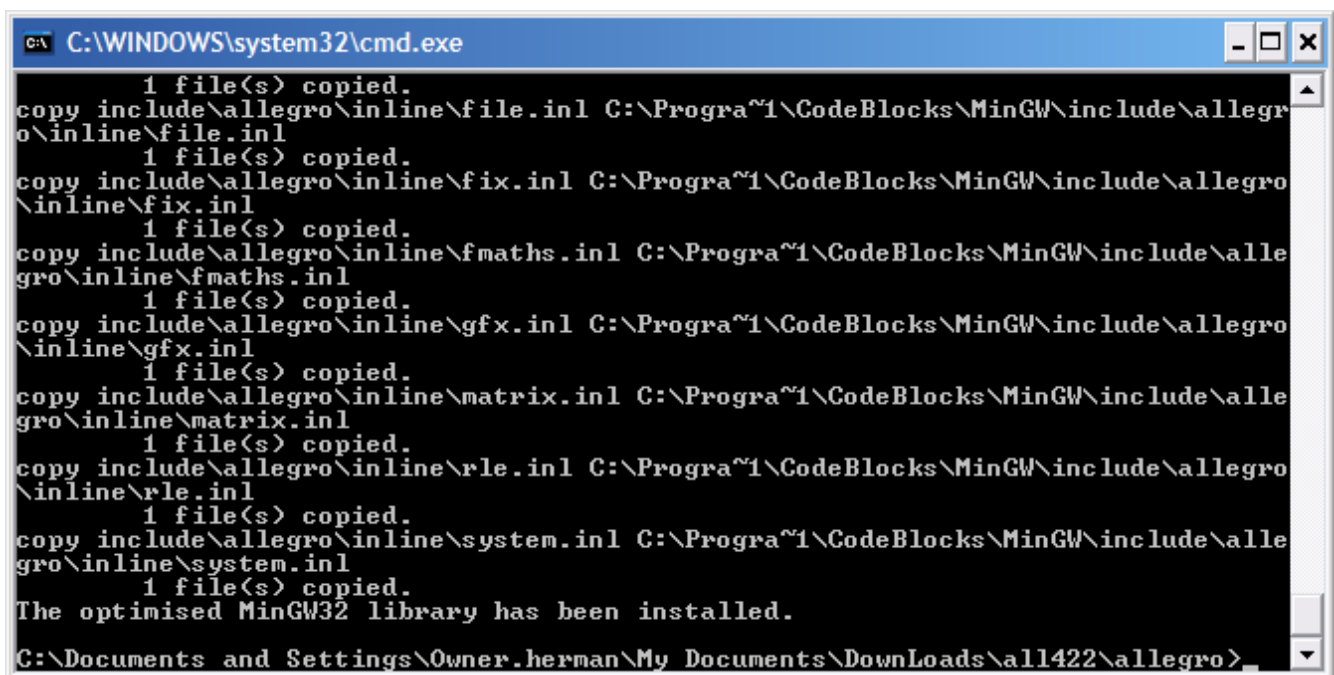
```

C:\WINDOWS\system32\cmd.exe
copy THANKS docs\txt\thanks.txt
    1 file(s) copied.
docs/makedoc.exe -ascii readme.txt docs/src/readme._tx
copy readme.txt docs\txt\readme.txt
    1 file(s) copied.
docs/makedoc.exe -scite docs/scite/allegro.api docs/src/allegro._tx
docs/makedoc.exe -ascii docs/build/bcc32.txt docs/src/build/bcc32._tx
docs/makedoc.exe -ascii docs/build/beos.txt docs/src/build/beos._tx
docs/makedoc.exe -ascii docs/build/darwin.txt docs/src/build/darwin._tx
docs/makedoc.exe -ascii docs/build/djgpp.txt docs/src/build/djgpp._tx
docs/makedoc.exe -ascii docs/build/dmc.txt docs/src/build/dmc._tx
docs/makedoc.exe -ascii docs/build/linux.txt docs/src/build/linux._tx
docs/makedoc.exe -ascii docs/build/macosx.txt docs/src/build/macosx._tx
docs/makedoc.exe -ascii docs/build/mingw32.txt docs/src/build/mingw32._tx
docs/makedoc.exe -ascii docs/build/msvc.txt docs/src/build/msvc._tx
docs/makedoc.exe -ascii docs/build/gnx.txt docs/src/build/gnx._tx
docs/makedoc.exe -ascii docs/build/unix.txt docs/src/build/unix._tx
docs/makedoc.exe -ascii docs/build/watcom.txt docs/src/build/watcom._tx
HTML help files have been generated.
To generate a .CHM version, run "make chm-docs".
The optimised MinGW32 library has been compiled.
Run "make install" with the options you passed to "make"
in order to complete the installation.

C:\Documents and Settings\Owner.herman\My Documents\Downloads\all422\allegro>

```

Illustration 9: Successful Allegro compile (end screen)



```

C:\WINDOWS\system32\cmd.exe
    1 file(s) copied.
copy include\allegro\inline\file.inl C:\Progra~1\CodeBlocks\MinGW\include\allegro\inline\file.inl
    1 file(s) copied.
copy include\allegro\inline\fix.inl C:\Progra~1\CodeBlocks\MinGW\include\allegro\inline\fix.inl
    1 file(s) copied.
copy include\allegro\inline\fmaths.inl C:\Progra~1\CodeBlocks\MinGW\include\allegro\inline\fmaths.inl
    1 file(s) copied.
copy include\allegro\inline\gfx.inl C:\Progra~1\CodeBlocks\MinGW\include\allegro\inline\gfx.inl
    1 file(s) copied.
copy include\allegro\inline\matrix.inl C:\Progra~1\CodeBlocks\MinGW\include\allegro\inline\matrix.inl
    1 file(s) copied.
copy include\allegro\inline\rle.inl C:\Progra~1\CodeBlocks\MinGW\include\allegro\inline\rle.inl
    1 file(s) copied.
copy include\allegro\inline\system.inl C:\Progra~1\CodeBlocks\MinGW\include\allegro\inline\system.inl
    1 file(s) copied.
The optimised MinGW32 library has been installed.

C:\Documents and Settings\Owner.herman\My Documents\Downloads\all422\allegro>

```

Illustration 10: Successful Allegro install

Create project “Hello World” example

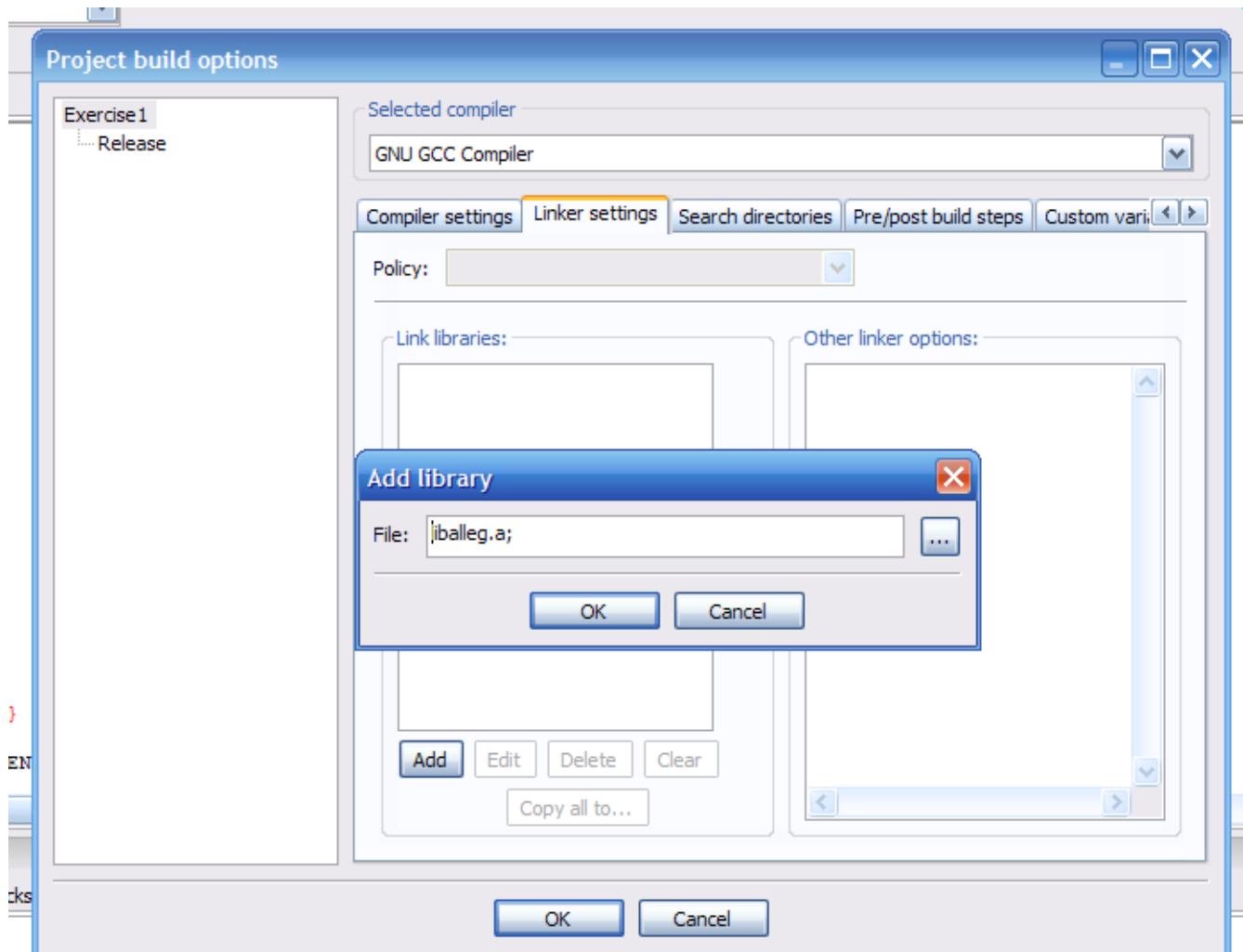
Compile, build, and run the HelloWorld project.

Change line 51 in the source and insert your name. Run example and print screen to turn in as the endpoint of the assignment. Note also that the `liballeg.a` library has to be specified for the Linker as shown in the illustration following the example program.

Turn in a screen capture of the output screen.

Add liballeg.a to linker settings

NOTE! In CodeBlocks, a reference to the allegro library has to be placed in the Build instructions. Go to **Project->Build Options->Linker Settings**, and add `liballeg.a` as shown below by pressing the Add button.



Create a new Hello World project with following source.

```
1 /*
2 *   Example program for the Allegro library, by Shawn Hargreaves.
3 *
4 *   This is a very simple program showing how to get into graphics
5 *   mode and draw text onto the screen.
6 */
7
8 #include <allegro.h>
9
10 int main(void)
11 {
12     int height;
13
14     /* you should always do this at the start of Allegro programs */
15     if (allegro_init() != 0)
16         return 1;
17
18     /* set up the keyboard handler */
19     install_keyboard();
20
21     /* set a graphics mode sized 320x200 */
22     if (set_gfx_mode(GFX_AUTODETECT_WINDOWED, 640, 400, 0, 0) != 0) {
23         if (set_gfx_mode(GFX_SAFE, 320, 200, 0, 0) != 0) {
24             set_gfx_mode(GFX_TEXT, 0, 0, 0, 0);
25             allegro_message("Unable to set any graphic mode\n%s\n", allegro_error);
26             return 1;
27         }
28     }
29
30     /* set the color palette */
31     set_palette(desktop_palette);
32
33     /* clear the screen to white */
34     clear_to_color(screen, makecol(255, 255, 255));
35
36     /* you don't need to do this, but on some platforms (eg. Windows) things
37     * will be drawn more quickly if you always acquire the screen before
38     * trying to draw onto it.
39     */
40     acquire_screen();
41
42     height = SCREEN_H/2;
43     /* write some text to the screen with black letters and transparent background */
44     textout_centre_ex(screen, font, "Hello, world!", SCREEN_W/2, height, makecol(0,0,0), -1);
45
46     height += 2*text_height(font);
47     textout_centre_ex(screen, font, "My Name!", SCREEN_W/2, height, makecol(0,0,0), -1);
48
49     /* you must always release bitmaps before calling any input functions */
50     release_screen();
51
52     /* wait for a keypress */
53     readkey();
54
55     return 0;
56 }
57
58
59 END_OF_MAIN()
60
```

Final Screen output

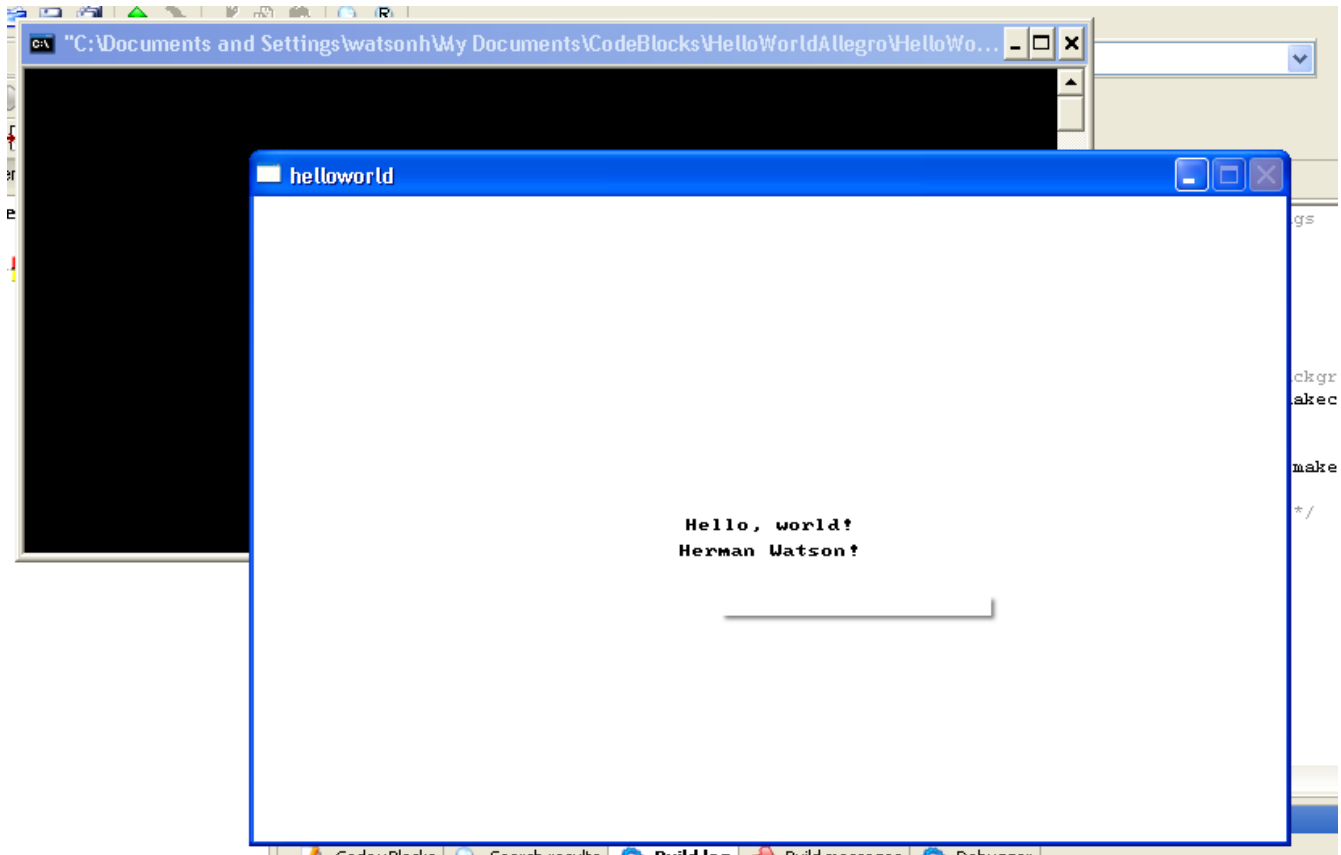


Illustration 11: Final output screen capture

Running the Demo program

Allegro also includes a demonstration game program that is automatically compiled and built when the library is built. This game resides in the **demo** directory (Illustration 12) within the folder where **all422** is unzipped. This folder is the same folder where **all422** was initially unzipped as shown in Illustration 6.

Use explorer to navigate to the **demo** directory and click on the **demo** icon to run the program. Good luck – hope you get a high score.

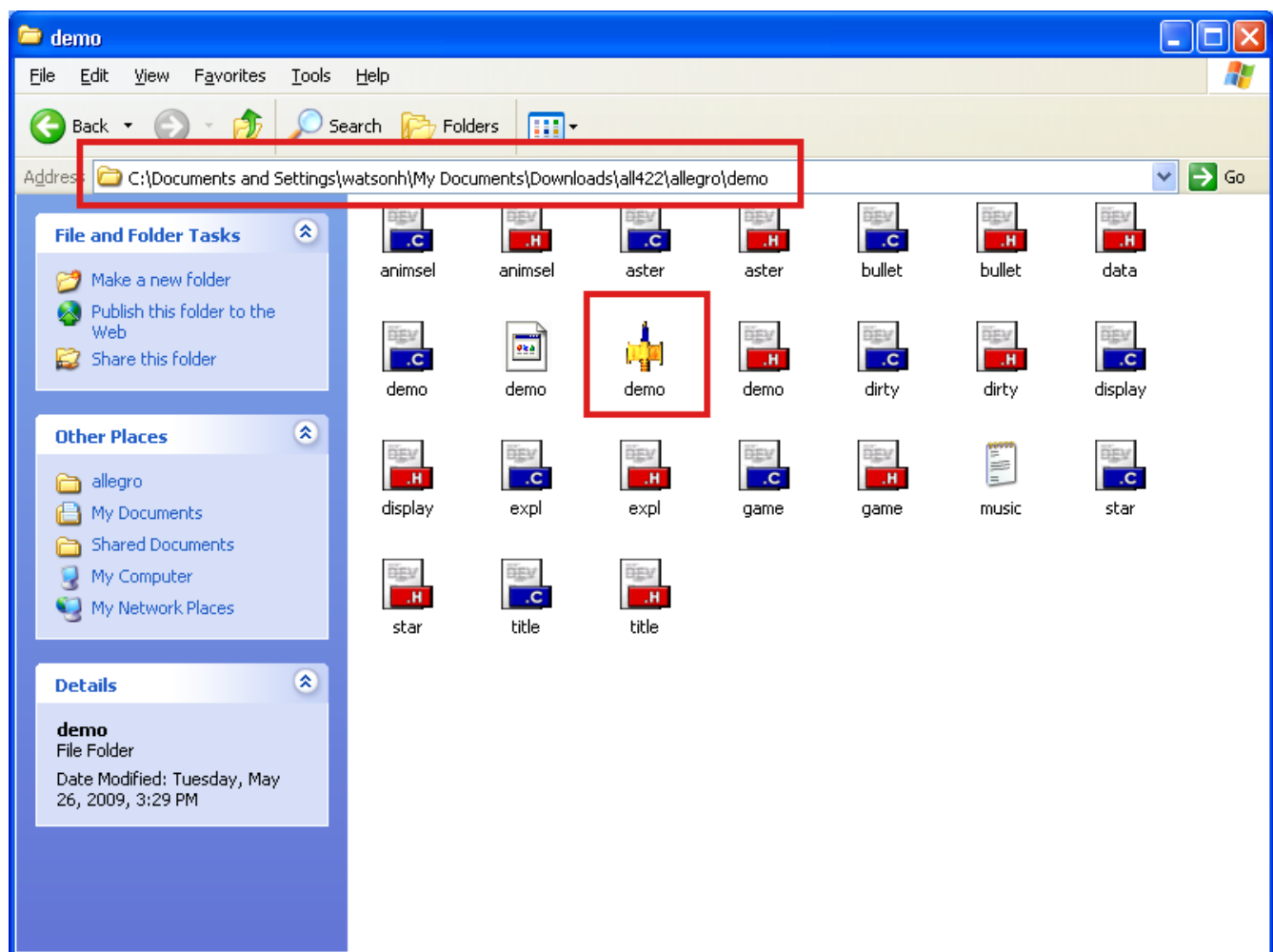


Illustration 12: Location of Allegro demo program