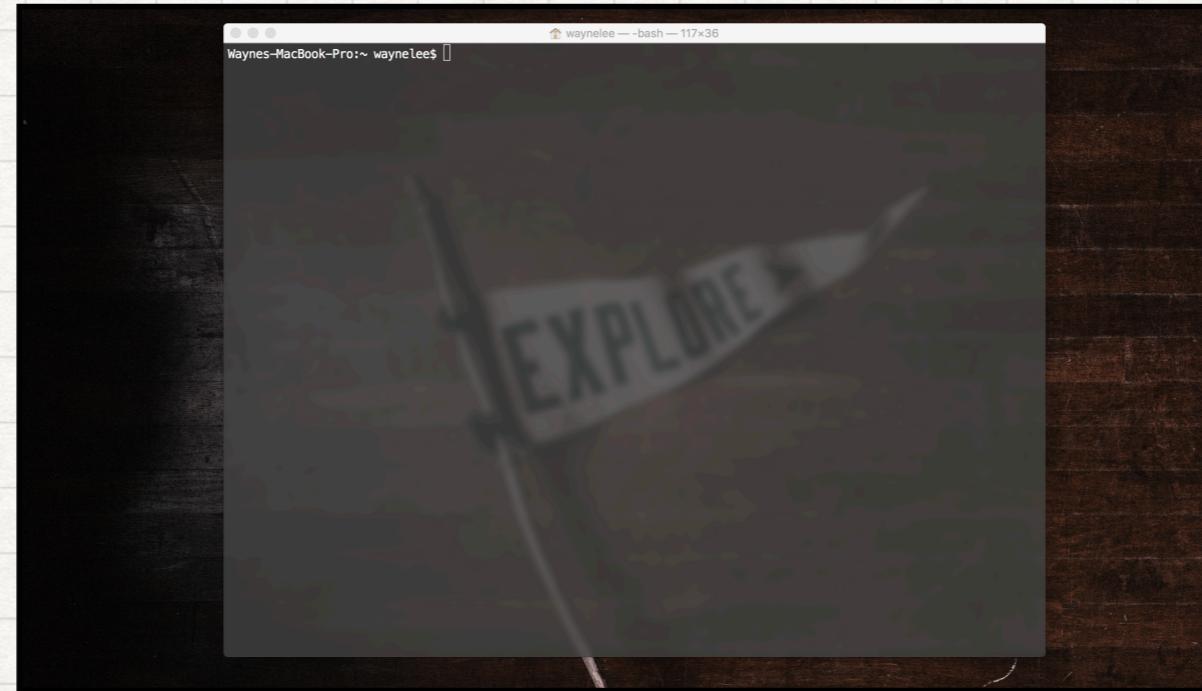


I2P-FINAL PROJECT
ALLEGRO

INSTALLATION

MACOS + XCODE

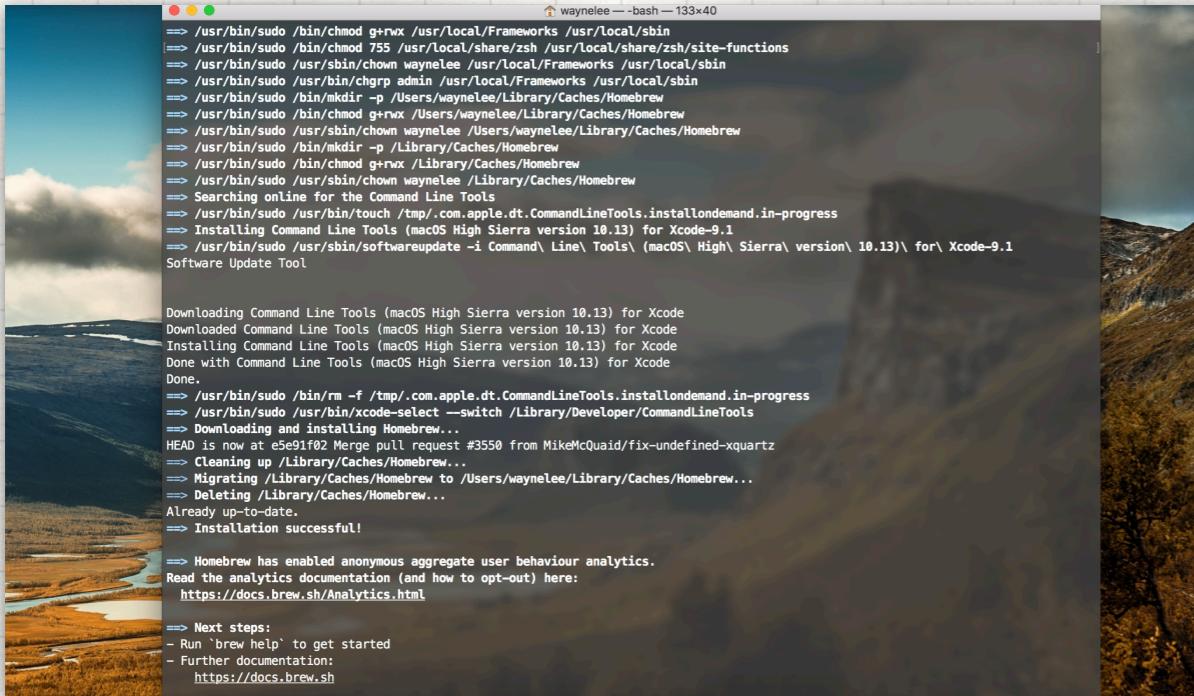
1. Open your “terminal” (終端機)



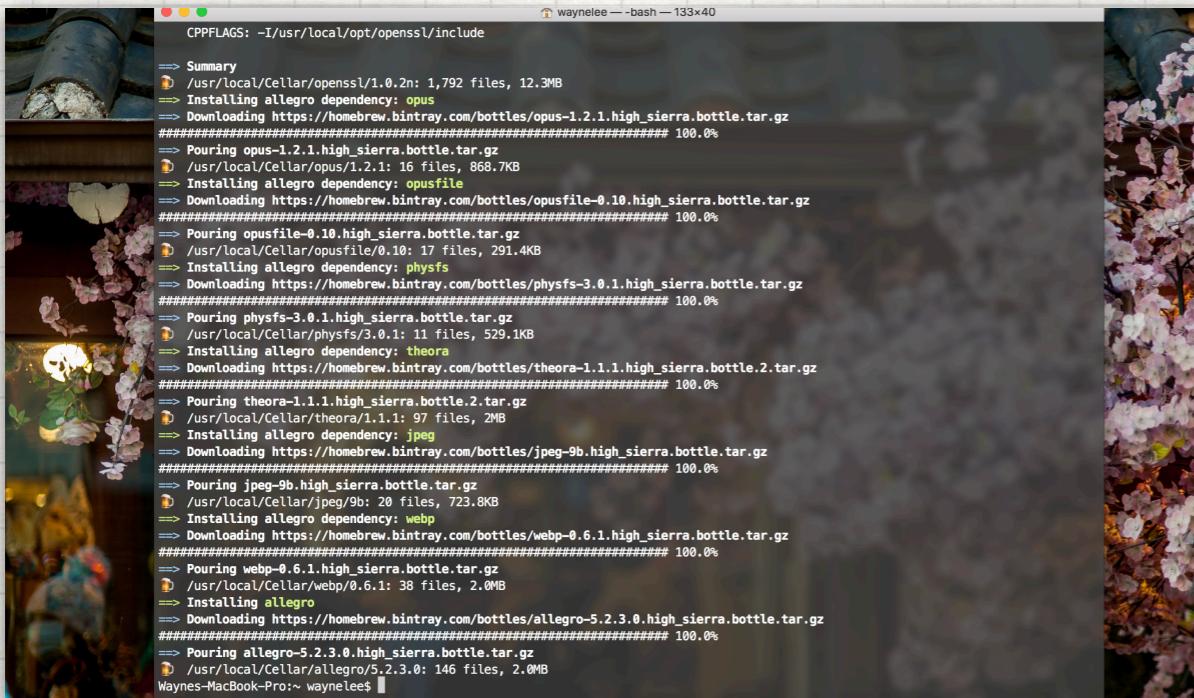
2. Type `/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"` in your terminal, then press enter.

3. Wait until the installation is done.

ALLEGRO



Once you're done, you should see something like this.

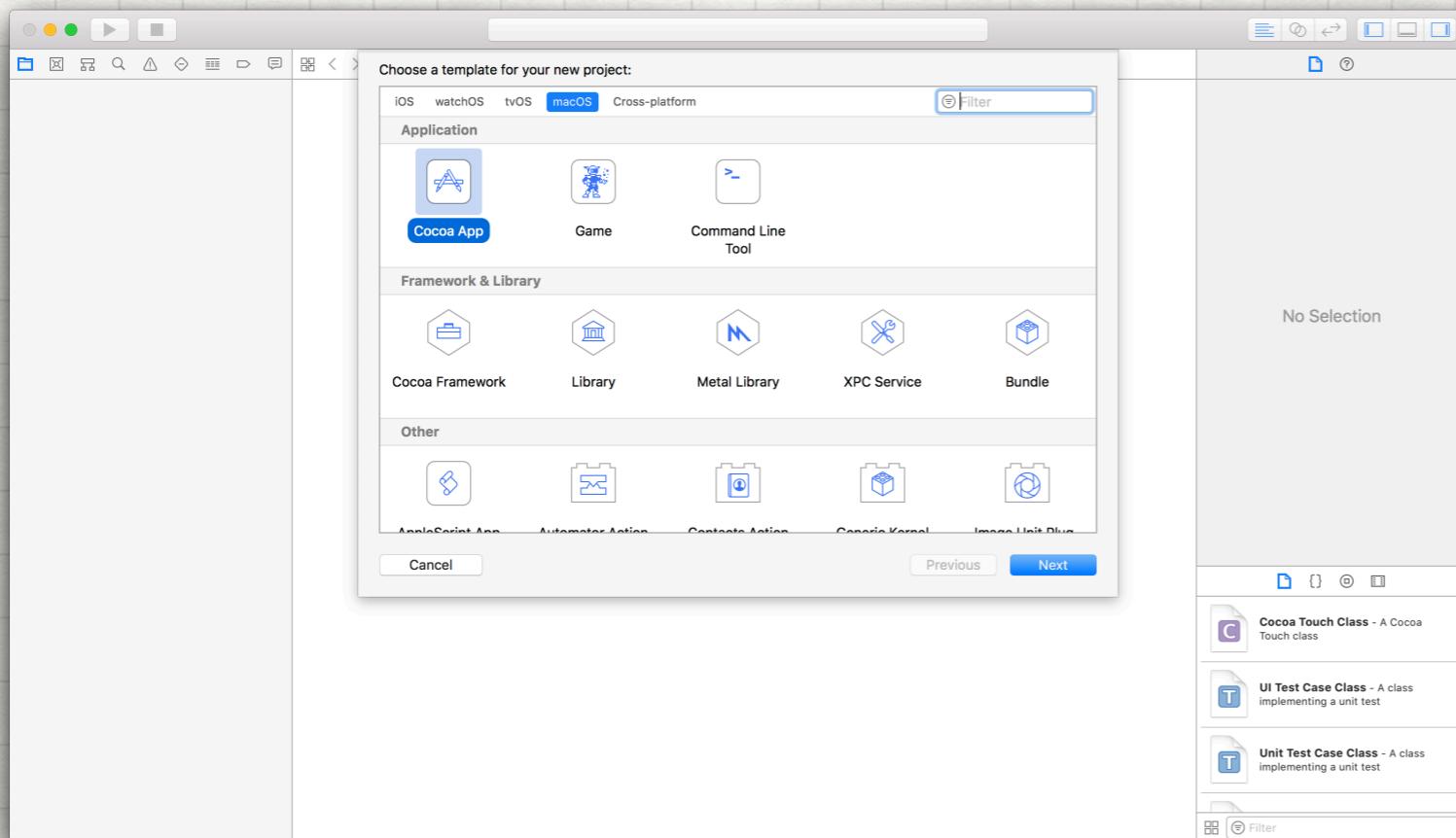


4. Type **brew install allegro** in your terminal, then press enter.

ALLEGRO

5. Open Xocde, which requires version 8.0 or higher

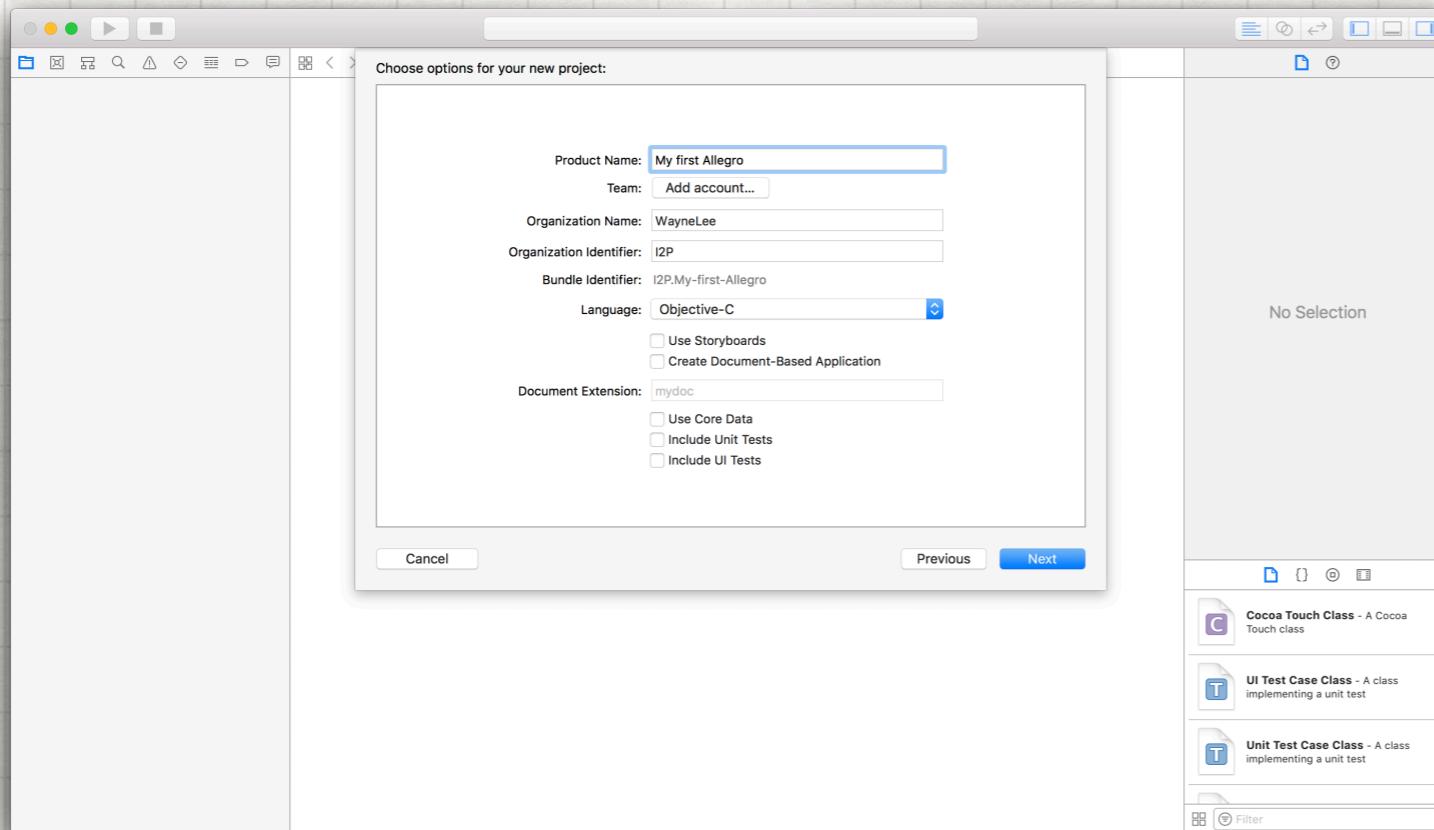
6. Choose **create a new Xcode project**



7. Choose **macOS -> Cocoa App -> Next**

ALLEGRO

8. Fill the **Product Name** , **Organization Name**, choose **Objective-C** as language



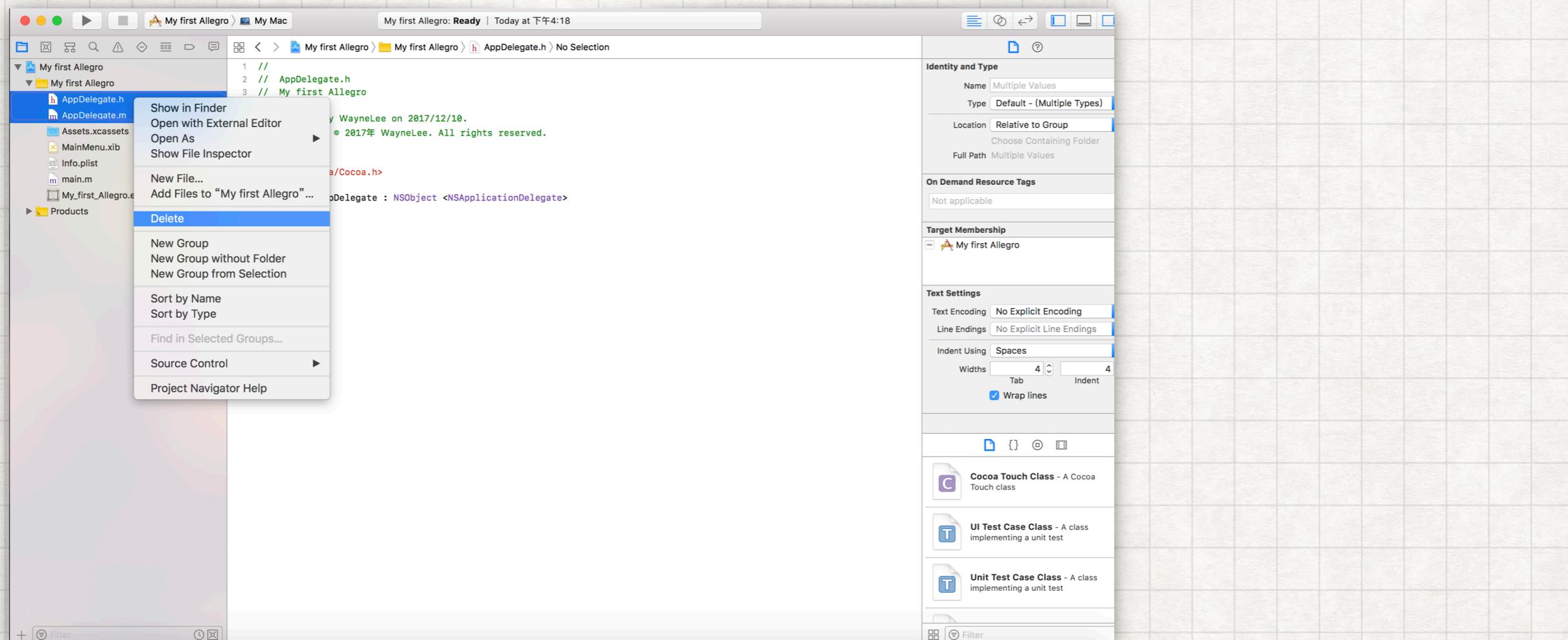
9. Cancel the **source control**, we will not use git here.
(Ignore this step if you haven't meet **source control** option)

ALLEGRO

10. After create the project, **delete** this two (or four) file.

AppDelegate.h, App Delegate.m (ViewController.h, ViewController.m)

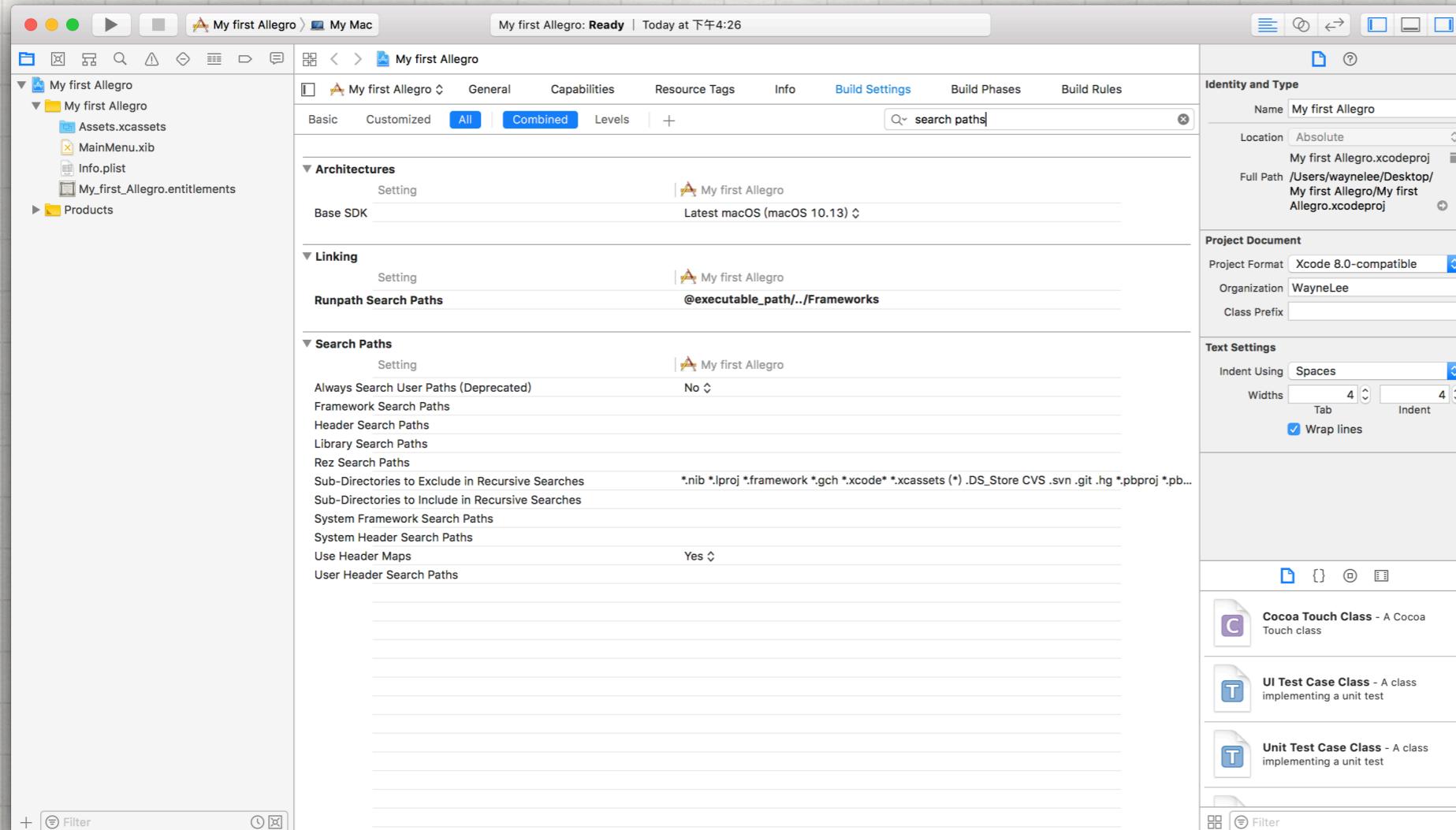
If you do not see ViewController(.m or .h), then ignore this two files.



11. Delete **Main.m** as well. Note that, choose **move to trash** while deleting.

ALLEGRO

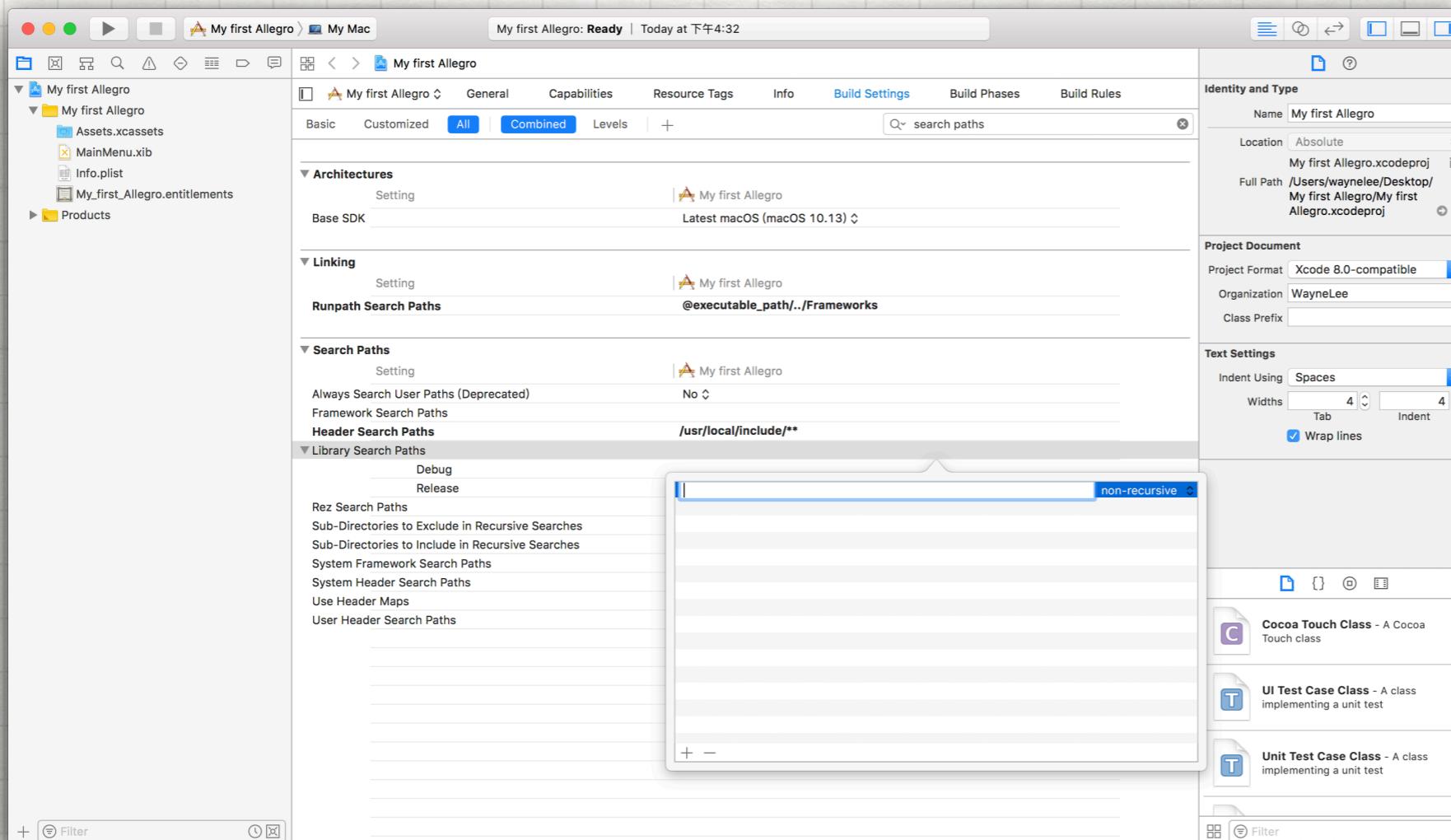
12. Go into Building settings, press the all label.



13. Search “search paths” (somewhere under building settings)

ALLEGRO

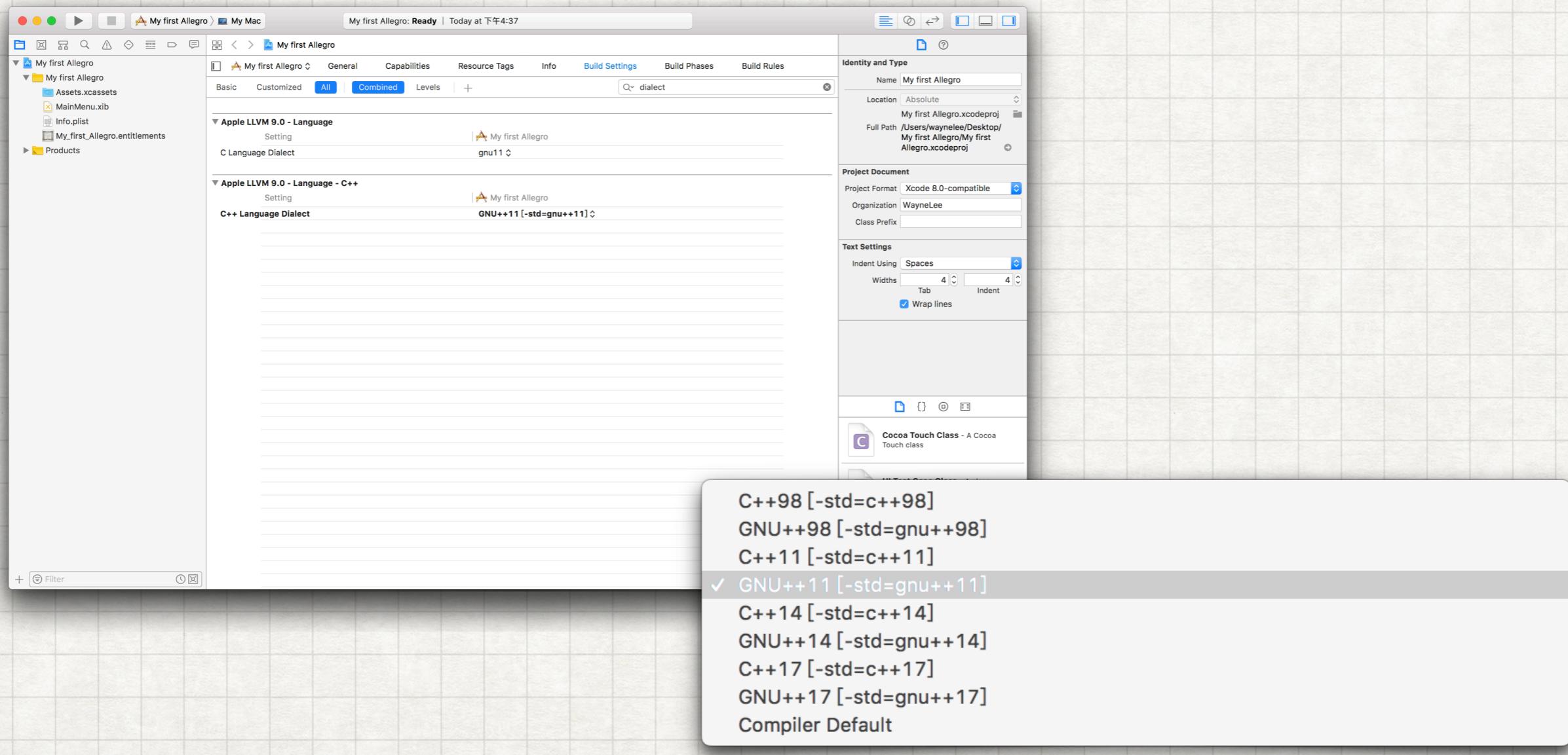
14. Find **Header Search Paths**, then double click on it. Press the **+** and type **/usr/local/include**, change non-recursive to **recursive**.



Do the same thing with **Library Search Paths**, the path is **/usr/local/lib**, also **recursive**.

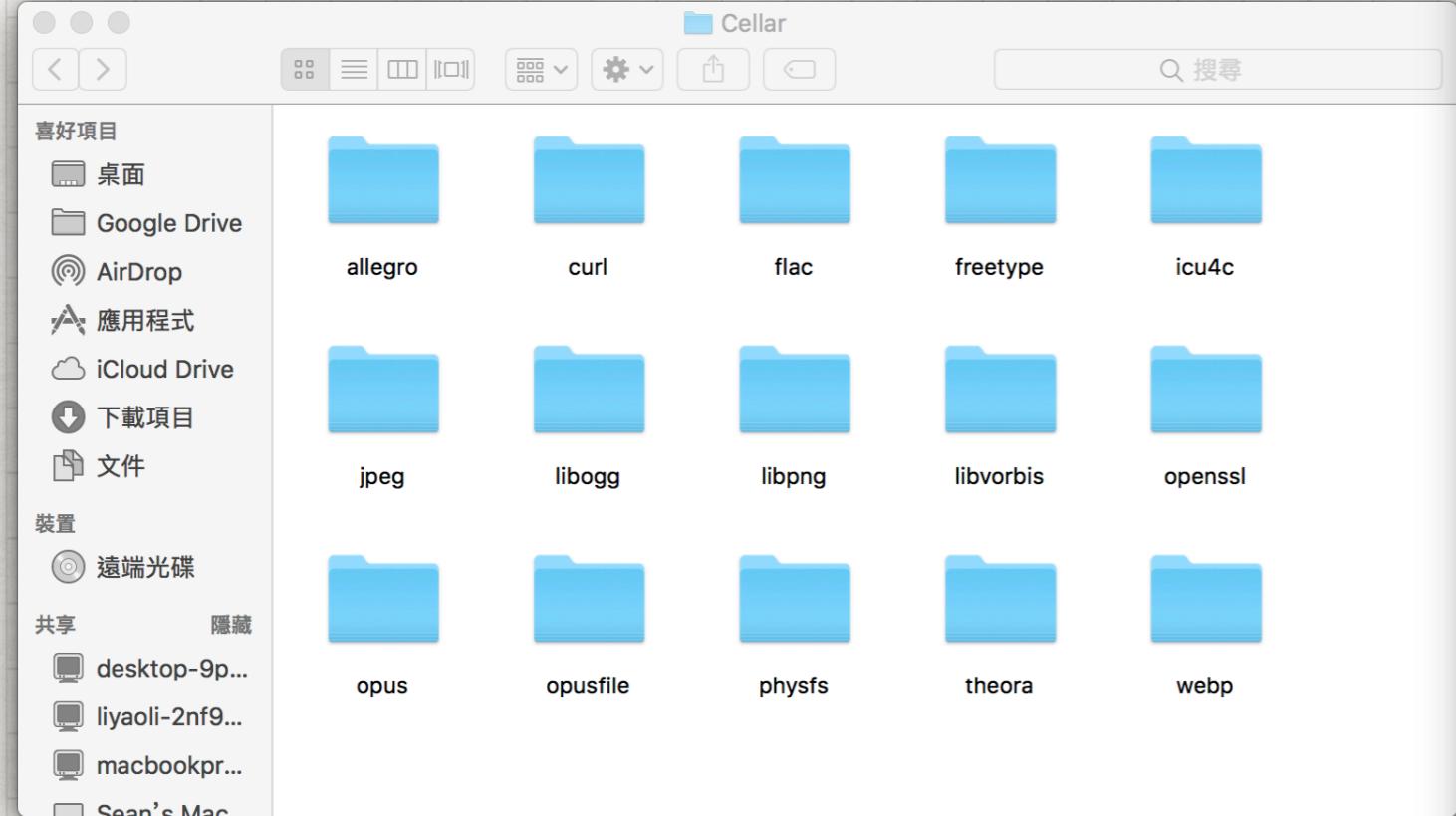
ALLEGRO

15. Search “**dialect**” (somewhere under building settings), then turn **C++ Language dialect** into **GNU++11**



ALLEGRO

16. Go back to terminal. Type `/usr/local/Cellar`, press enter. Type `open .`, press enter.



```
Waynes-MacBook-Pro:~ waynelee$ cd /usr/local/Cellar/
Waynes-MacBook-Pro:Cellar waynelee$ open .
Waynes-MacBook-Pro:Cellar waynelee$
```

ALLEGRO

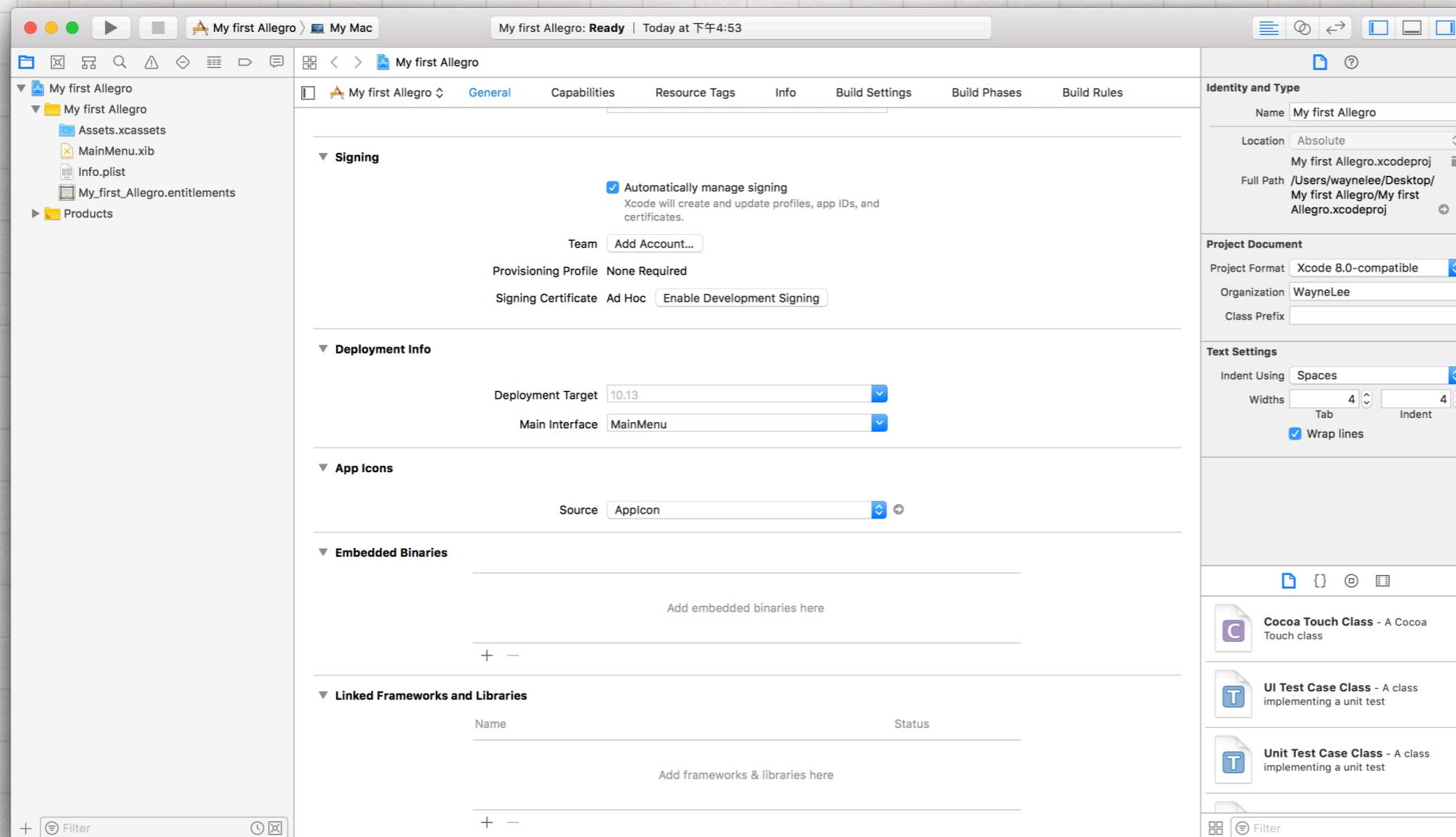
17. Enter allegro -> 5.2.3.0 -> lib

18. Sort by Date modified, choose the files that modified today

	▶	pkgconfig	今天下午3:43	-- 檔案夾
		liballegro_video.5.2.3.dylib	今天下午3:43	46 KB Dynamic Library
		liballegro_ttf.5.2.3.dylib	今天下午3:43	31 KB Dynamic Library
		liballegro_primitives.5.2.3.dylib	今天下午3:43	82 KB Dynamic Library
		liballegro_physfs.5.2.3.dylib	今天下午3:43	25 KB Dynamic Library
		liballegro_memfile.5.2.3.dylib	今天下午3:43	17 KB Dynamic Library
		liballegro_main.5.2.3.dylib	今天下午3:43	13 KB Dynamic Library
		liballegro_image.5.2.3.dylib	今天下午3:43	50 KB Dynamic Library
		liballegro_font.5.2.3.dylib	今天下午3:43	37 KB Dynamic Library
		liballegro_dialog.5.2.3.dylib	今天下午3:43	57 KB Dynamic Library
		liballegro_color.5.2.3.dylib	今天下午3:43	28 KB Dynamic Library
		liballegro_audio.5.2.3.dylib	今天下午3:43	86 KB Dynamic Library
		liballegro_acodec.5.2.3.dylib	今天下午3:43	55 KB Dynamic Library
		liballegro.5.2.3.dylib	今天下午3:43	770 KB Dynamic Library
		liballegro_acodec.5.2.dylib	2017年10月10日上午9:39	29 byte 替身
		liballegro_acodec.dylib	2017年10月10日上午9:39	27 byte 替身
		liballegro_audio.5.2.dylib	2017年10月10日上午9:39	28 byte 替身
		liballegro_audio.dylib	2017年10月10日上午9:39	26 byte 替身
		liballegro_color.5.2.dylib	2017年10月10日上午9:39	28 byte 替身
		liballegro_color.dylib	2017年10月10日上午9:39	26 byte 替身
		liballegro_dialog.5.2.dylib	2017年10月10日上午9:39	29 byte 替身
		liballegro_dialog.dylib	2017年10月10日上午9:39	27 byte 替身
		liballegro_font.5.2.dylib	2017年10月10日上午9:39	27 byte 替身
		liballegro_font.dylib	2017年10月10日上午9:39	25 byte 替身
		liballegro_image.5.2.dylib	2017年10月10日上午9:39	28 byte 替身
		liballegro_image.dylib	2017年10月10日上午9:39	26 byte 替身
		liballegro_main.5.2.dylib	2017年10月10日上午9:39	27 byte 替身
		liballegro_main.dylib	2017年10月10日上午9:39	25 byte 替身
		liballegro_memfile.5.2.dylib	2017年10月10日上午9:39	30 byte 替身
		liballegro_memfile.dylib	2017年10月10日上午9:39	28 byte 替身
		liballegro_physfs.5.2.dylib	2017年10月10日上午9:39	29 byte 替身
		liballegro_physfs.dylib	2017年10月10日上午9:39	27 byte 替身

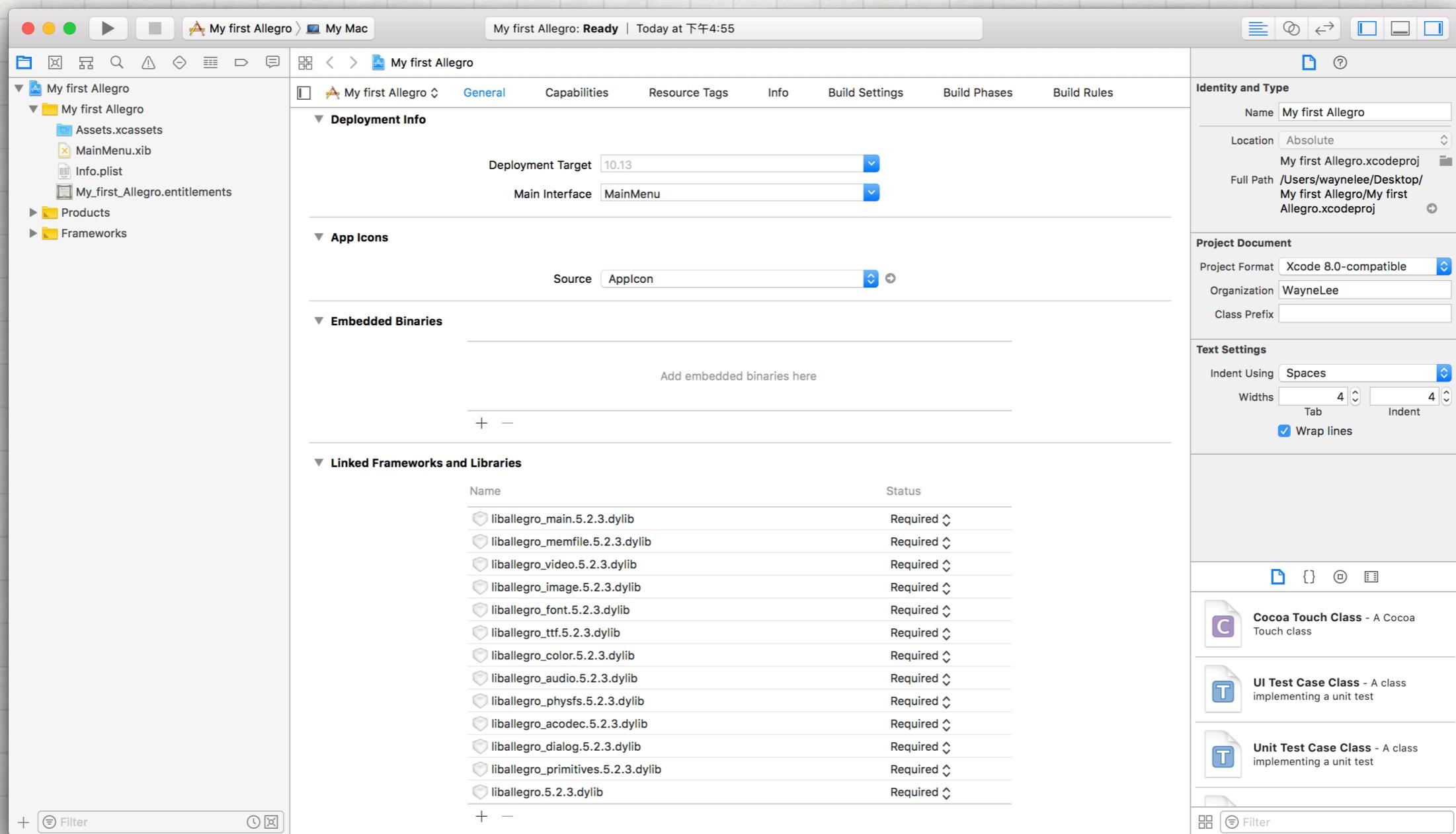
ALLEGRO

19. Go back to Xcode, find general.



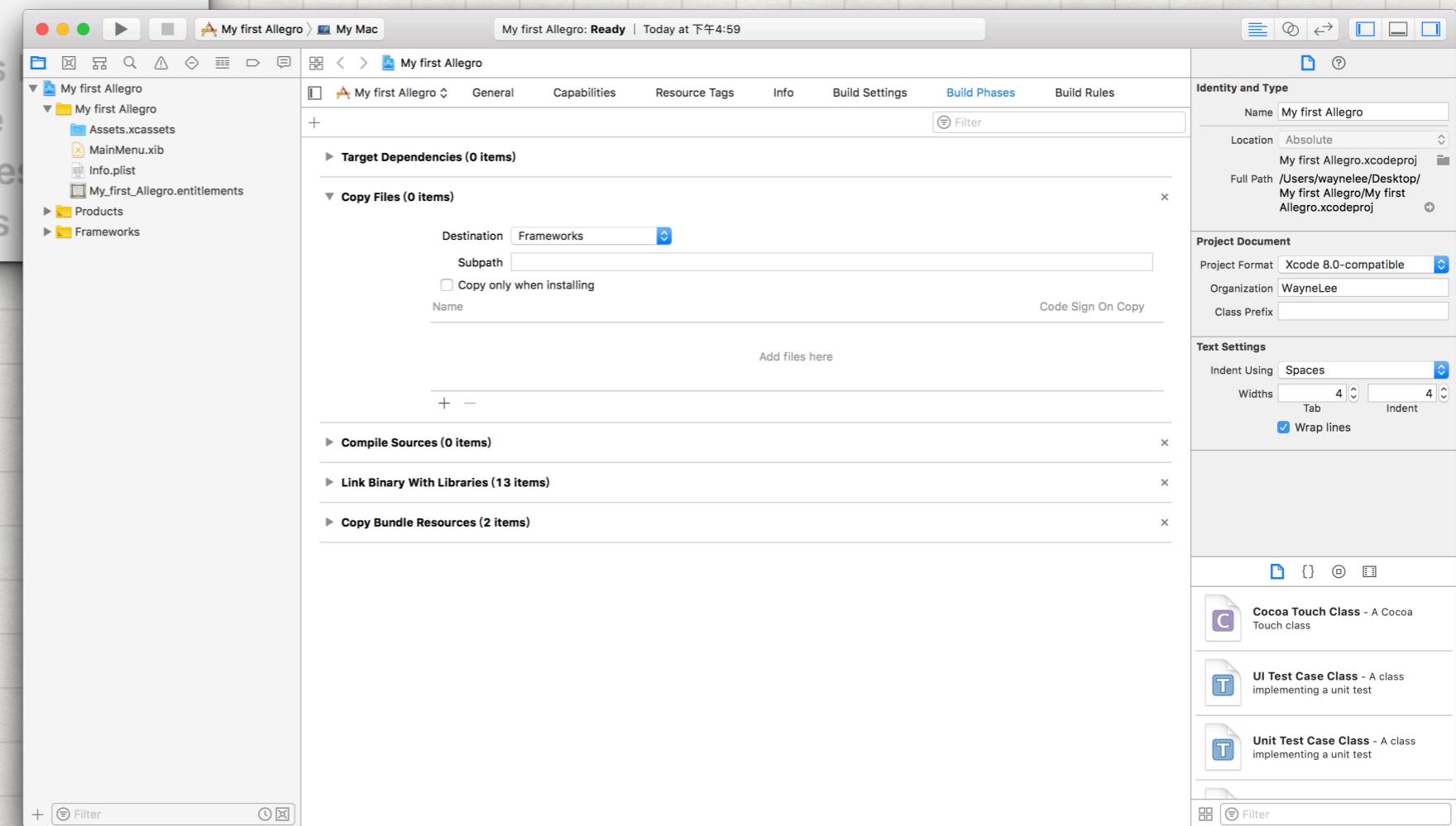
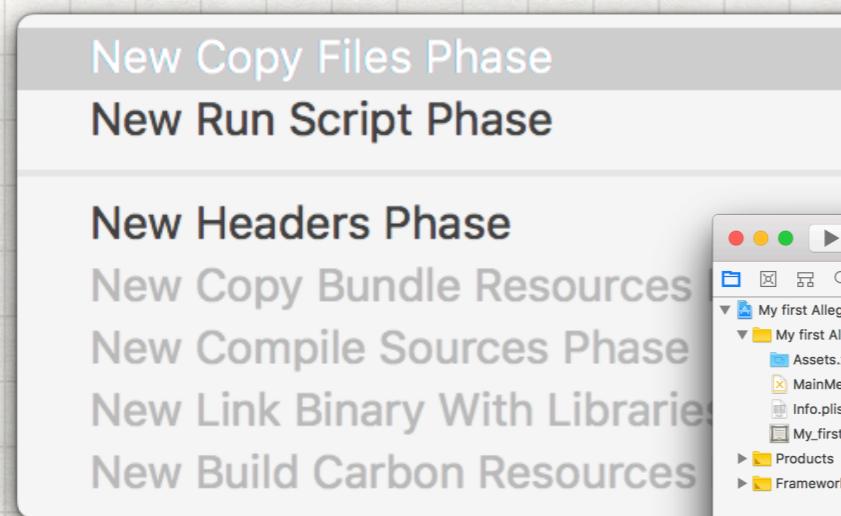
ALLEGRO

20. Move all the selected file to **Linked Frameworks and Libraries**.



ALLEGRO

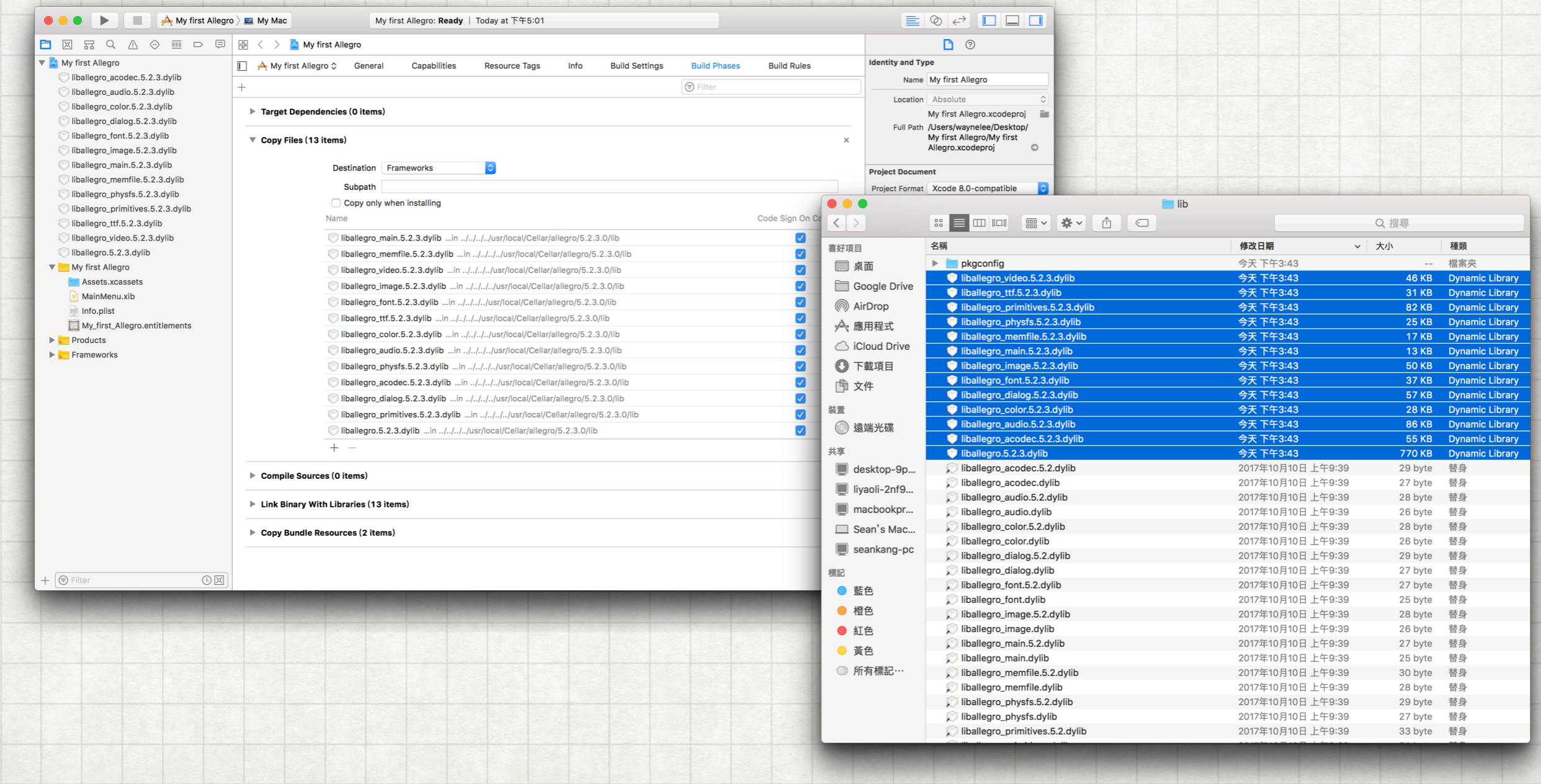
21. Go to **Build phases** (the one next to Building setting), press **+** , choose **New Copy files phase**



22. Find **Copy files**,
change **destination**
from **resources** to
Frameworks

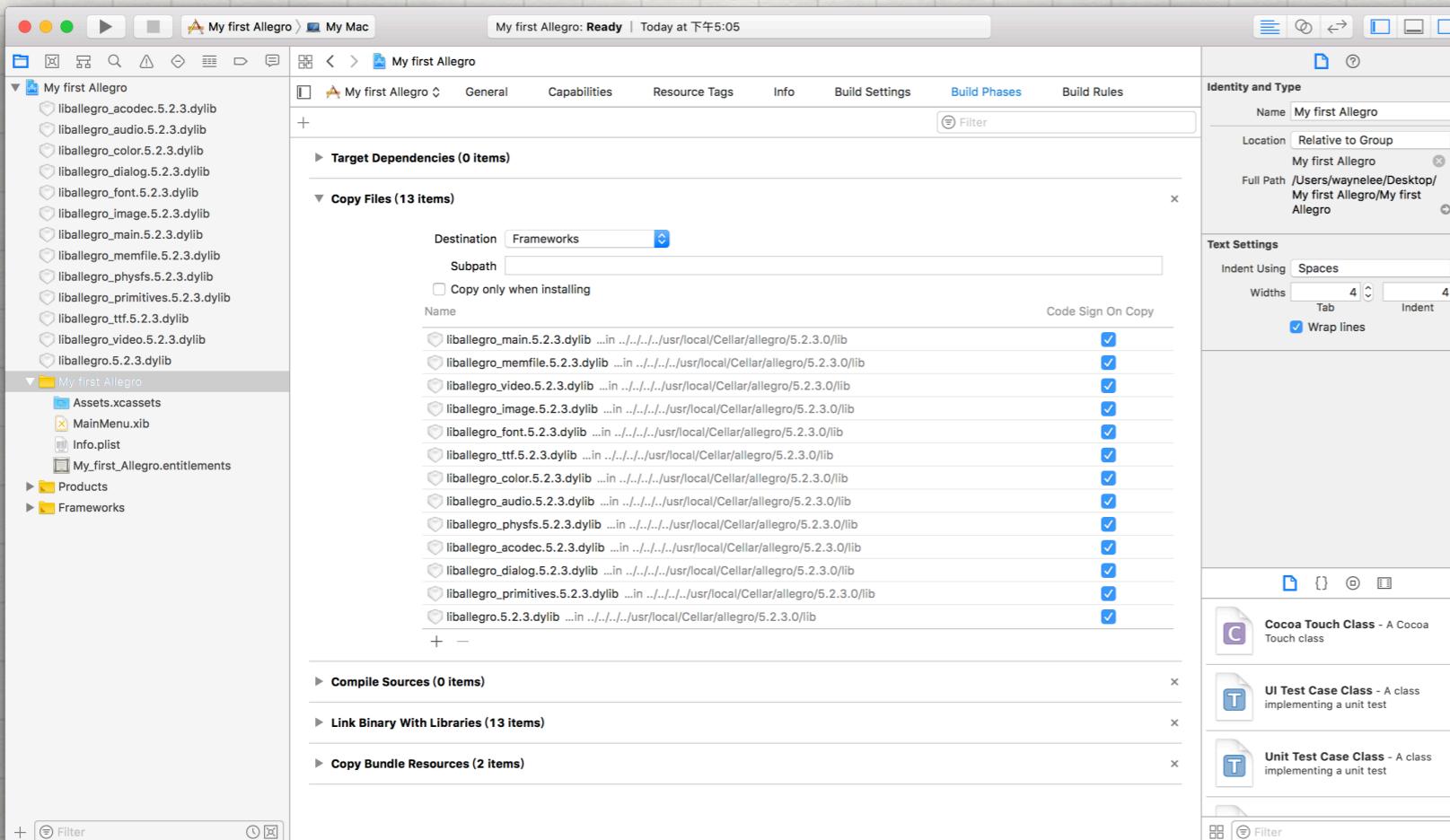
ALLEGRO

23. Then drag the files you selected before to here.



ALLEGRO

24. Congratulations! You've done with the settings. Now lets get started !
Find "My first Allegro " and right click it to new file.



25. New file -> C++ file -> main.cpp -> create

ALLEGRO

You can now build your first allegro program. Here's an example code.

The screenshot shows the Xcode IDE interface with a project named "My first Allegro". The main area displays the "main.cpp" file content:

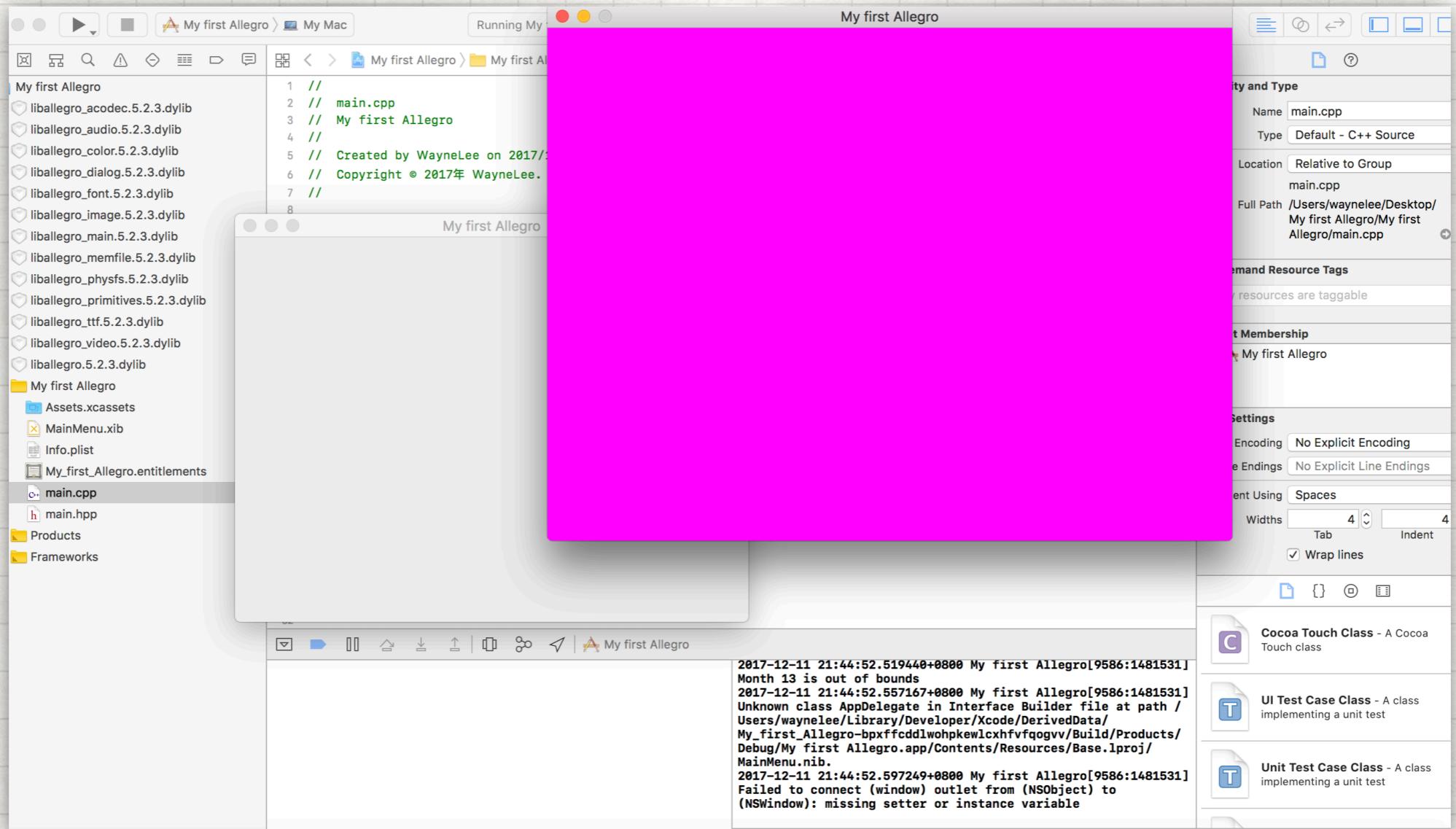
```
1 //  
2 //  main.cpp  
3 // My first Allegro  
4 //  
5 // Created by WayneLee on 2017/12/10.  
6 // Copyright © 2017年 WayneLee. All rights reserved.  
7 //  
8  
9  
10 #include "main.hpp"  
11 #include <allegro5/allegro.h>  
12 #include <allegro5/allegro_native_dialog.h>  
13  
14 int main(int argc, char **argv){  
15  
16     al_init();  
17  
18     al_create_display(640, 480);  
19  
20     al_clear_to_color(al_map_rgb(255, 0, 255));  
21  
22     al_flip_display();  
23  
24     al_rest(5.0f);  
25  
26     return 0;  
27 }  
28  
29 /*  
30 #include <stdio.h>  
31 #include <allegro5/allegro.h>  
32
```

The "Identity and Type" panel on the right shows the file is a "Default - C++ Source" file named "main.cpp". The "Text Settings" panel shows "No Explicit Encoding" and "Wrap lines" checked. The bottom output window shows the assembly and memory dump of the application's memory.

[AllegroAppDelegate app_main:] + 23	0x00007fff3caa56d8
13 Foundation	
__NSThread__start__ + 1197	0x00000001004d86b9
_pthread_body + 340	0x00000001004d8565
_libsystem_pthread.dylib	
_pthread_body + 0	0x00000001004d7c55
16 libsystem_pthread.dylib	
thread_start + 13	
Hello world!!!	
See you, world	

ALLEGRO

When the window pop out, you're ready for the final project!



- If there's any problem with the installation on MacOS, you're welcome to ask in iLMS.
- Note that, if you meet an compiled error about "Arguments is too long", go to -> building settings -> Search "search paths" -> change /usr/local/lib into "non-recursive".



Fighting !