

## Configuración de Allegro5 con CLion (WINDOWS)

## ÍNDICE

<b>1</b>	<b>Instalación de la librería Allegro5</b>	<b>3</b>
<b>2</b>	<b>Configuración del proyecto</b>	<b>4</b>

## 1 INSTALACIÓN DE LA LIBRERÍA ALLEGRO5

Para instalar la librería Allegro5 deberemos descargar la librería desde la [página de descargas](#).

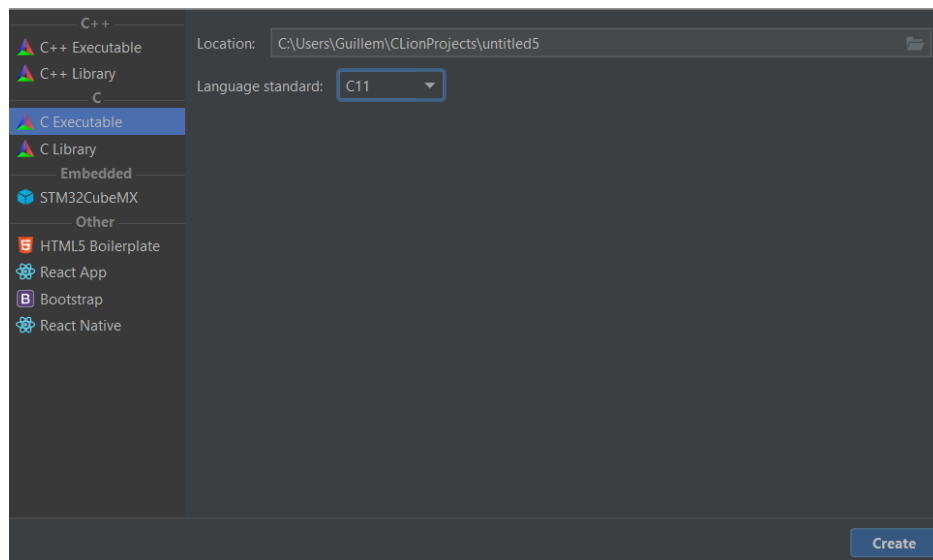
The screenshot shows the Allegro.cc website. The header includes the logo { allegro.cc } and the tagline 'game developing community network'. The left sidebar contains navigation links: About, Depot, Files, Forums, Manual, Members, News, Search, and Wiki. Below these are 'Projects' with a list of categories and counts: Action (252), Adventure (22), Arcade (216), Board Games (39), Demos (24), Emulators (10), Other (68), Puzzle (139), Role Playing (40), Sports (25), Strategy (58), and Utilities (46). The 'Resource Directory' section lists 'Code Gallery' (3) and 'Comments' (12). The main content area is titled 'Files' and shows version tabs for 4.2, 4.4, and 5.0. The 'Current Version' is 5.0.10. Under 'Windows Binaries', it states that it is highly recommended to use one of the binary packages. A list of available packages is shown, with 'MinGW 4.7.0 - zip, 7z' highlighted. Other packages include MinGW 4.5.0, 4.5.2, 4.6.1 (TDM), 4.6.2, MSVC 9, 10, and 11. Under 'Documentation & Examples', there are links for the 'Reference Manual: Online, PDF' and 'Examples: Source & Windows Binaries'. The 'Source Code' section is also visible at the bottom.

Una vez descargado el fichero zip, lo extraeremos en una ubicación conocida. Si nos dirigimos al directorio donde se haya extraído, debe haber tres carpetas: bin, include y lib. Posteriormente necesitaremos mover estas carpetas.

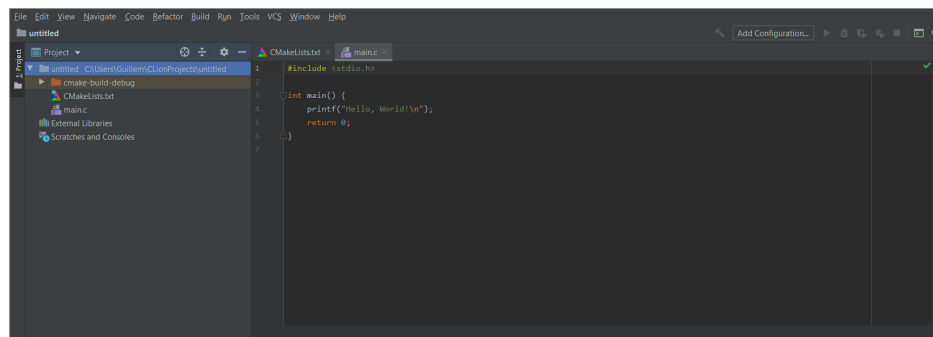
## 2 CONFIGURACIÓN DEL PROYECTO

Una vez instalado [CLion](#), crearemos y configuraremos un nuevo proyecto para poder utilizar Allegro.

Primero, crearemos un nuevo proyecto, del tipo *C executable* y estándar de lenguaje C11, tal como podemos observar en la imagen siguiente:

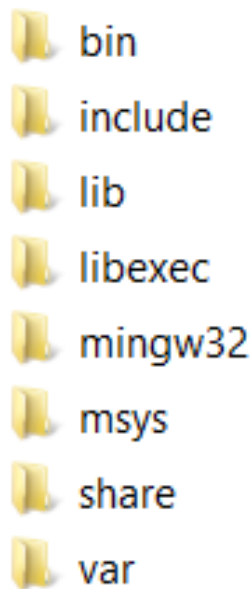


Una vez creado, deberíamos obtener un proyecto con la siguiente estructura:



Con tal de poder compilar satisfactoriamente, deberemos instalar [Mingw](#), un toolchain que entre otros incluye GCC, que deberemos añadir a CLion. Para instalar correctamente Mingw y configurar CLion podéis seguir [este tutorial](#).

Ahora debemos ir donde hemos extraído el fichero zip de Allegro justo al inicio de la explicación. Si lo recordamos, había tres carpetas. Deberemos copiarlas y pegarlas en la misma dirección donde se encuentre MinGW. Debería quedar un listado de carpetas como el siguiente:



Ahora tenemos que modificar nuestro fichero CMakeLists.txt para poder utilizar la librería Allegro.

El fichero tiene que quedar igual que en la imagen, a excepción del nombre del ejecutable (en este caso, untitled5):

```
cmake_minimum_required(VERSION 3.15)
project(untitled5 C)

set(CMAKE_C_STANDARD 11)
set(ALLEGRO_DIR "C:\\MinGW")
include_directories(${ALLEGRO_DIR}/include)
link_directories(${ALLEGRO_DIR}/lib)

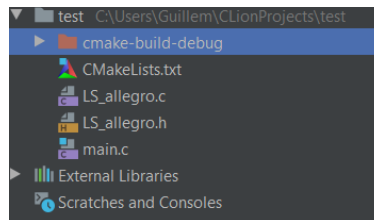
add_executable(untitled5 main.c)

target_link_libraries(untitled5 -lmingw32 -lallegro-5.0.10-monolith-md-debug)
```

Finalmente, recargaremos la configuración pulsando *Reload changes*.

En este punto nuestro proyecto ya puede utilizar Allegro, pero dado que ciertas funciones de éste están fuera del alcance de la asignatura, se os facilita una librería, LS\_Allegro, la cual deberemos añadir.

Esta librería está formada por dos ficheros, LS\_Allegro.c y LS\_Allegro.h, los cuales se pueden encontrar en el eStudy. Una vez descargados, los añadiremos a nuestro proyecto:



Como hemos añadido estos ficheros manualmente, tenemos que indicarle al compilador que forman parte del ejecutable. Para esto, modificaremos el comando *add\_executable* de CMakeLists.txt:

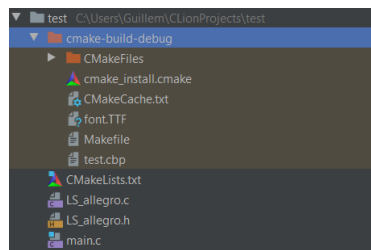
```
cmake_minimum_required(VERSION 3.15)
project(untitled5 C)

set(CMAKE_C_STANDARD 11)
set(ALLEGRO_DIR "C:\\MinGW")
include_directories(${ALLEGRO_DIR}/include)
link_directories(${ALLEGRO_DIR}/lib)

add_executable(untitled5 main.c LS_Allegro.c LS_Allegro.h)

target_link_libraries(untitled5 -lmingw32 -lallegro-5.0.10-monolith-md-debug)
```

Después, añadiremos el fichero *font.tff* (también disponible en el eStudy) a la carpeta *cmake-build-debug*:



Finalmente, reemplazamos el contenido del main por el del *main.c* proporcionado en el eStudy:

```
1 #include <stdio.h>
2 #include "LS_allegro.h"
3
4 int main(void){
5
6     int nSortir = 0;
7
8     //Inicialitzem Allegro
9     LS_allegro_init( nAmplitud: 800, nAltura: 600, sNomFinestra: "Projecte 2");
10
11
12     //Bucle infinit del joc
13     while(!nSortir){
14
15         //Escoltem el teclat esperant la tecla ESC
16         if(LS_allegro_key_pressed( nKey: ALLEGRO_KEY_ESCAPE)){
17             nSortir = 1;
18         }
19
20         //Donem l'ordre d'escriure el text de benvinguda
21         al_draw_textf( font: LS_allegro_get_font(NORMAL), color: LS_allegro_get_color(WHITE), x: 140, y: 100, flags: 0, format: "%s", "Benvingut a Allegro!");
22
23         //Pintem la pantalla de la finestra gràfica
24         LS_allegro_clear_and_paint(BLACK);
25     }
26
27     //Tanquem la finestra gràfica
28     LS_allegro_exit();
29
30     return 0;
31 }
```

Si ejecutamos el proyecto, podemos ver que Allegro funciona correctamente:

