



UNIVERSIDAD DE GRANADA

Práctica 2 de Periféricos y Dispositivos de Interfaz Humana

Curso 2024-2025

Francisco Quiles Ramírez

Índice:

Instalación.....	3
Ejemplo de Hola Mundo:.....	4
Ejemplo ventana.....	5
Ejemplo pelota.....	5
Ejemplo de pelota que rebota.....	6
Ejemplo 2 de pelota que rebota.....	6
Ejemplo pong.....	7

Instalación

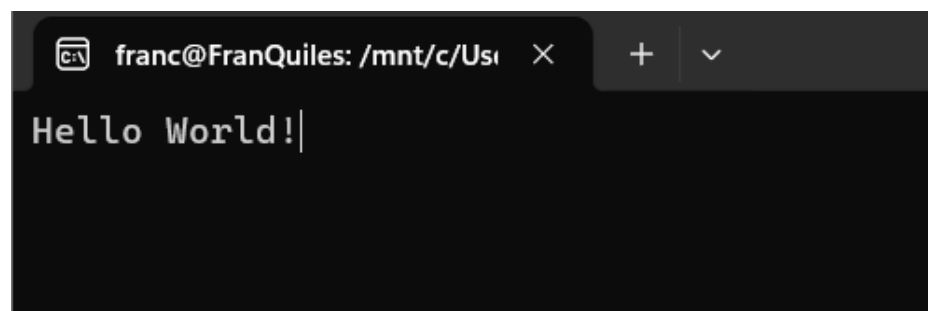
En esta práctica vamos a utilizar la biblioteca ncurses que provee una API que permite escribir interfaces basadas en texto, se van a probar diferentes programas ya dados y finalmente hacer un juego de pong con pantalla de inicio y final. Lo primero será la instalación de la biblioteca ncurses como podemos ver a continuación:

```
franc@FranQuiles:/mnt/c/Users/franc/Desktop/Cuarto - 2°Cuatri/Perifericos/PDIH/P2$
sudo apt-get install libncurses5-dev libncursesw5-dev
[sudo] password for franc:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'libncurses-dev' instead of 'libncurses5-dev'
Note, selecting 'libncurses-dev' instead of 'libncursesw5-dev'
The following additional packages will be installed:
  libncurses6
Suggested packages:
  ncurses-doc
The following NEW packages will be installed:
  libncurses-dev libncurses6
0 upgraded, 2 newly installed, 0 to remove and 31 not upgraded.
Need to get 496 kB of archives.
After this operation, 2761 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu noble/main amd64 libncurses6 amd64 6.4+20240
113-1ubuntu2 [112 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 libncurses-dev amd64 6.4+20
240113-1ubuntu2 [384 kB]
Fetched 496 kB in 1s (924 kB/s)
Selecting previously unselected package libncurses6:amd64.
(Reading database ... 51650 files and directories currently installed.)
Preparing to unpack .../libncurses6_6.4+20240113-1ubuntu2_amd64.deb ...
Unpacking libncurses6:amd64 (6.4+20240113-1ubuntu2) ...
Selecting previously unselected package libncurses-dev:amd64.
Preparing to unpack .../libncurses-dev_6.4+20240113-1ubuntu2_amd64.deb ...
Unpacking libncurses-dev:amd64 (6.4+20240113-1ubuntu2) ...
Setting up libncurses6:amd64 (6.4+20240113-1ubuntu2) ...
Setting up libncurses-dev:amd64 (6.4+20240113-1ubuntu2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
```

Ahora vamos a probar con diferentes ejemplos ya implementados, a continuación podemos ver cómo compilarlo y ejecutarlo y finalmente el resultado que debería mostrar.

Ejemplo de Hola Mundo:

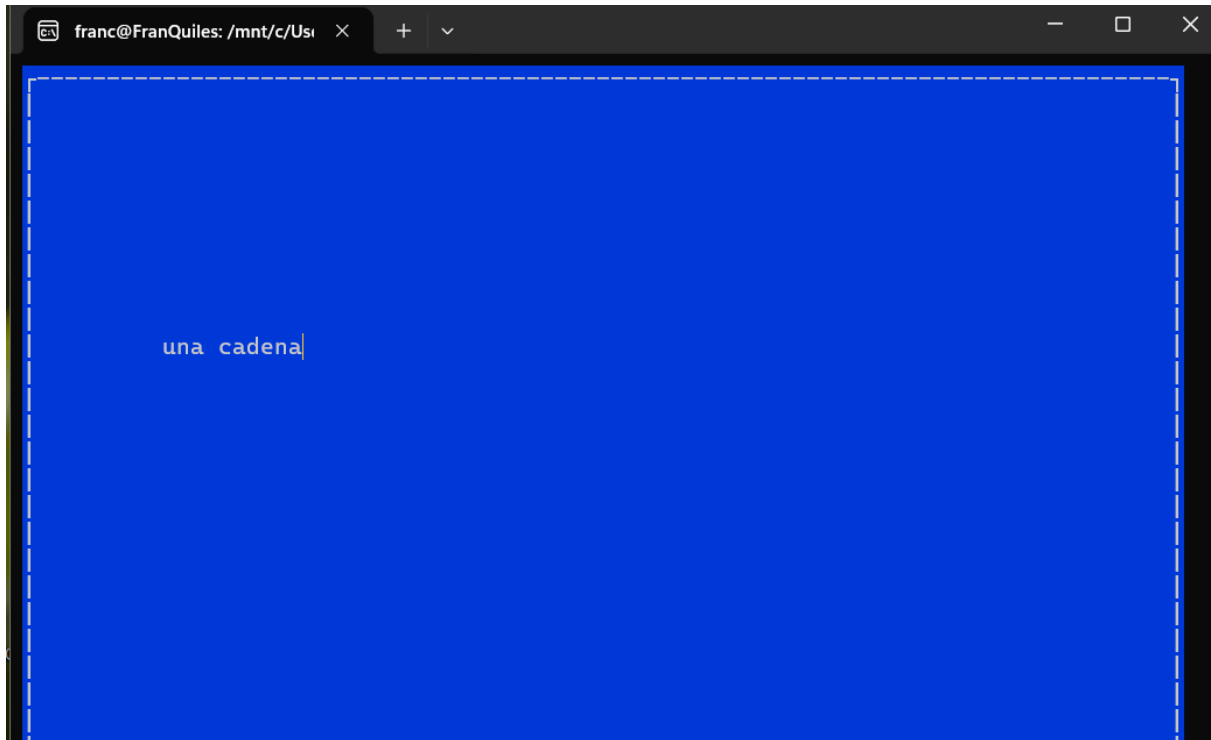
```
gcc hello.c -o hello -lncurses
./hello
```



Ejemplo ventana

```
gcc ventana.c -o ventana -lncurses
```

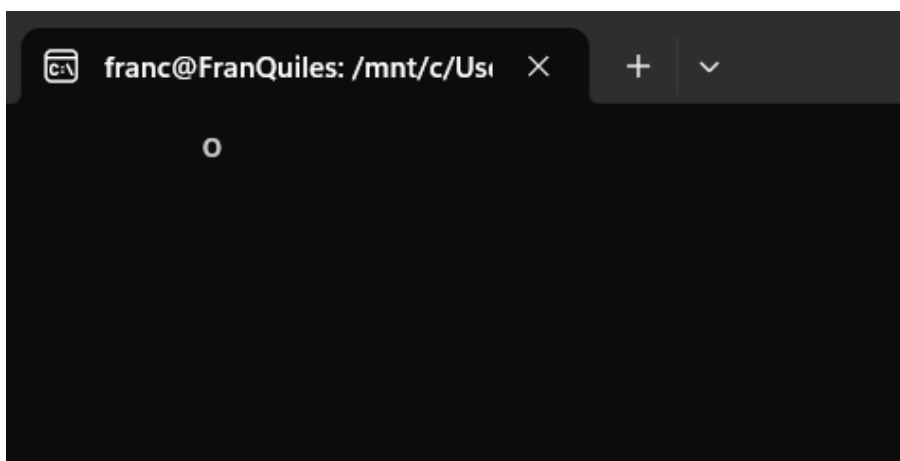
```
./ventana
```



Ejemplo pelota

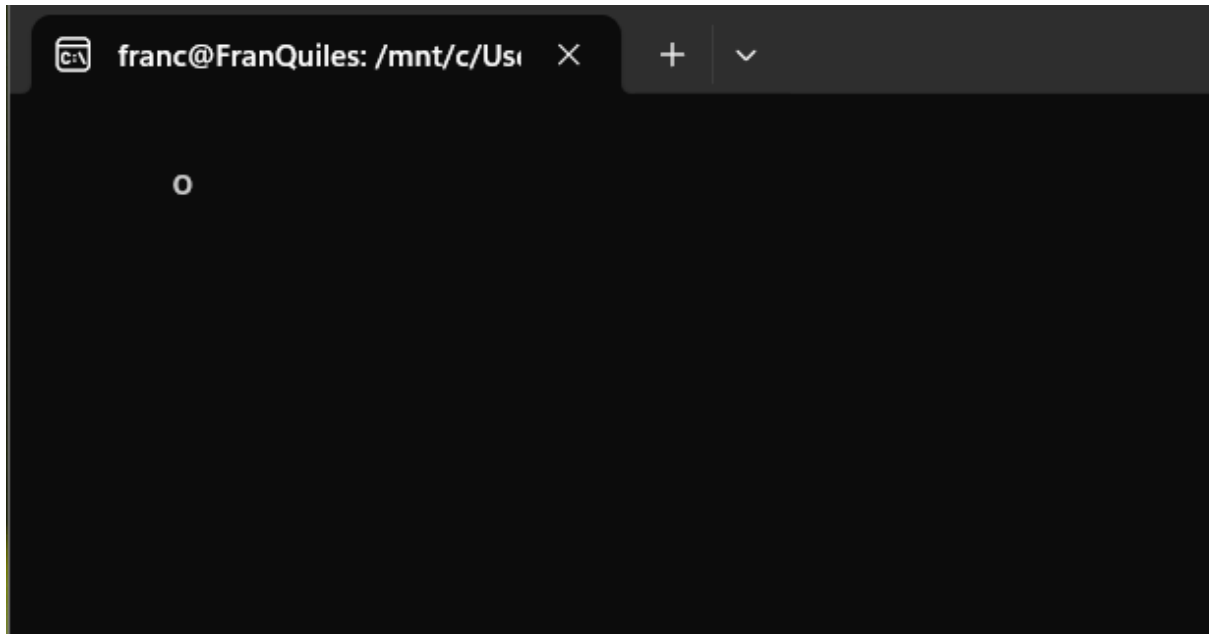
```
gcc -o pelotita pelotita.c -lncurses
```

```
./pelotita
```



Ejemplo de pelota que rebota

```
gcc -o rebota rebota.c -lncurses  
./rebota
```



Ejemplo 2 de pelota que rebota

```
gcc -o rebota2 rebota2.c -lncurses  
./rebota2
```

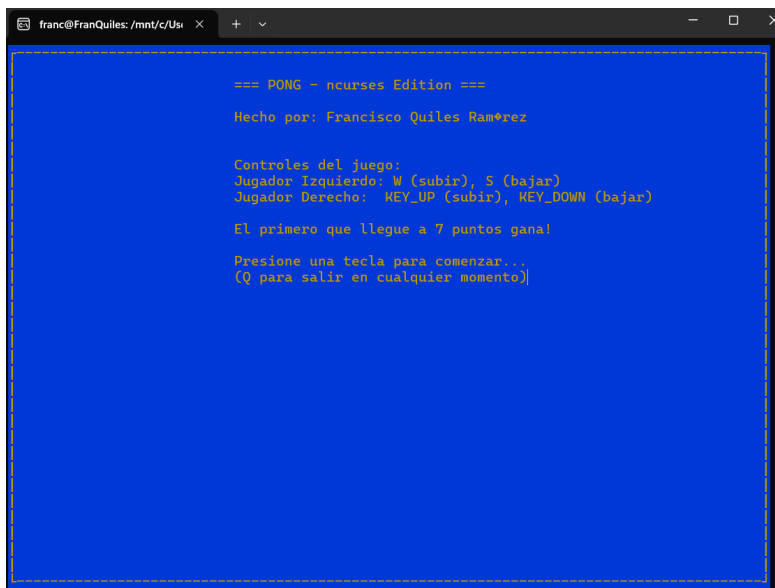


Ejemplo pong

Se ha creado un juego de pong sencillo siguiendo los ejemplos dados, donde dos jugadores podrán mover su “pala” de arriba a abajo internado devolver la pelota al rival y así sumar puntos hasta haber un ganador .

Para empezar se ha hecho una función `mostrarPantallaInicio()` la cual muestra una pantalla de bienvenida al jugador antes de iniciar el juego. Utiliza `ncurses` para crear una ventana con un fondo azul y texto amarillo, centrado horizontalmente. Esta pantalla mostrará el título del juego, el autor del proyecto, los controles del jugador izquierdo y derecho, unas reglas básicas y el mensaje para comenzar.

Para implementarlo primero se crea una ventana a pantalla completa, se le aplica un color personalizado (`COLOR_PAIR(4)`), se escribe el contenido con `mvwprintw`, se espera que el jugador pulse una tecla (`getch()`) y finalmente se cierra la ventana con `delwin`.



```
franc@FranQuiles: /mnt/c/Usr x + v
=== PONG - ncurses Edition ===
Hecho por: Francisco Quiles Ramirez

Controles del juego:
Jugador Izquierdo: W (subir), S (bajar)
Jugador Derecho:  KEY_UP (subir),  KEY_DOWN (bajar)

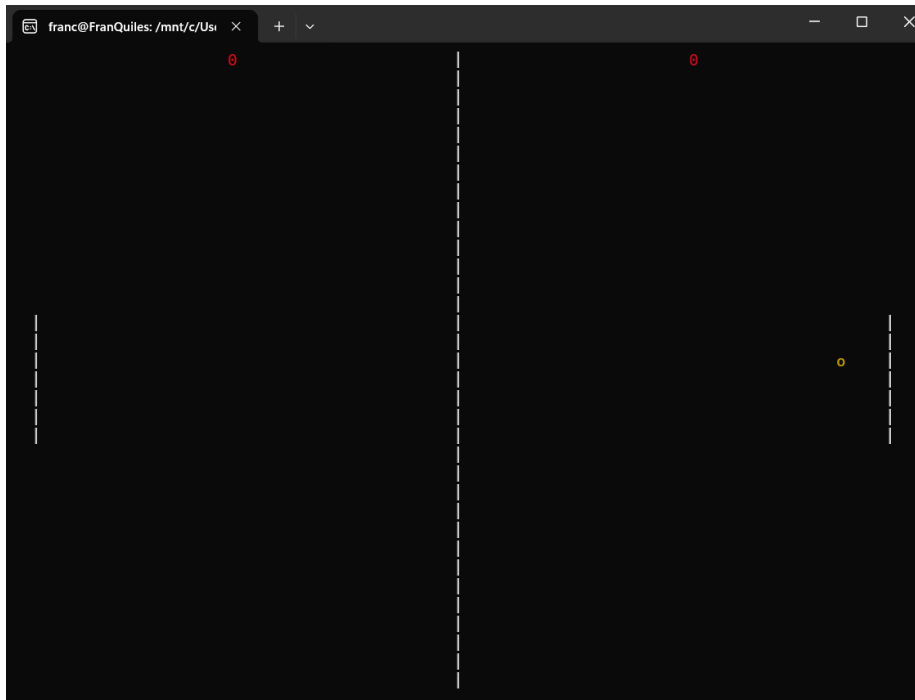
El primero que llegue a 7 puntos gana!

Presione una tecla para comenzar...
(Q para salir en cualquier momento)
```

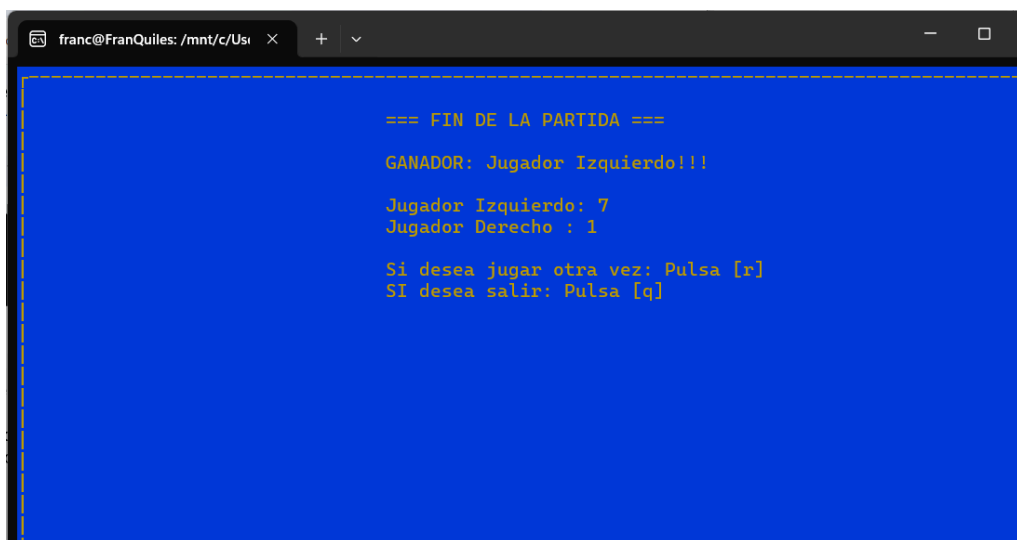
Ahora vamos con la implementación del `main` el cual es el núcleo del juego donde se inicializa todo y se ejecuta el bucle principal que permite jugar, mover las palas, detectar colisiones, hacer puntos y finalizar la partida.

Tras llamar a la pantalla de inicio, se inicializan los distintos valores que usaremos más adelante, ahora entraremos en el bucle principal que se ejecutará hasta que el usuario pulse “q” o alguno gane, cada ciclo de este bucle será un frame del juego. Se dibujarán las palas, pelota, línea del medio y marcador, a los que además se le han asignado algún color, ahora se hace una captura de las teclas que permiten mover las palas o salir del juego. Lo siguiente será el movimiento y colisiones, donde se calcula la posición próxima de la pelota, se comprueba si ha chocado con la pala, borde o ha sido punto y se cambia la dirección en consecuencia, por lo tanto si la pelota sale de la pantalla, se suma un punto al rival y se reposiciona la pelota.

Finalmente si un jugador llega a 7 puntos, se sale del bucle y se muestra la pantalla final



Al igual que al principio tendremos ahora otra pantalla implementada en la función `mostrarResmuen()` la cual mostrará el ganador y el resultado obtenido en la partida, además de la opción de volver a jugar pulsando la tecla `r` o salir del juego pulsando la tecla `q`. La implementación será similar a la de la pantalla de inicio añadiendo la opción de poder volver a repetir donde se inicializan los valores a los iniciales si se desea repetir o saldrá si se desea finalizar.



Para compilarlo y poder probarlo se usará:

```
gcc -o p2 p2.c -lcurses  
./p2
```