



# UNIVERSIDAD DE GRANADA

Periféricos y dispositivos de interfaz humana

## Práctica 2: Uso de bibliotecas de programación de interfaces de usuario en modo texto

---

**Autor:** Juan José Martínez Águila

**Correo:** jjmartagui@correo.ugr.es



GRADO EN INGENIERÍA INFORMÁTICA

Curso 2022 - 2023

# Índice

1. Programas de ejemplo	2
2. Juego Pong	3

## 1. Programas de ejemplo

Primero mostraremos el funcionamiento de los programas descritos anteriormente en el guión de prácticas:



Figura 1: Programa del movimiento de la pelota

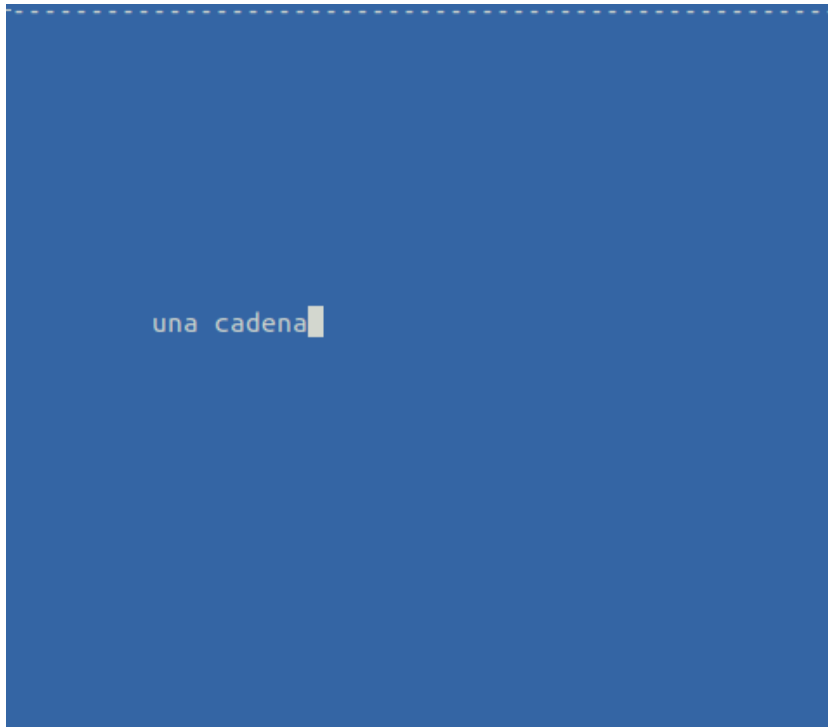


Figura 2: Programa de creación de una nueva ventana y imprimir una cadena

## 2. Juego Pong

Primero he creado dos variables globales para las medidas del campo de juego, y luego las demás variables necesarias como la posición de la pelota, los jugadores, booleanos de condición de parada de juego,...

```
C pelotita.c  F pelotita  C pong.c  X  C ventana.c
C pong.c > @ main()
1 #include <ncurses.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <unistd.h>
5 #define DELAY 100000
6
7 // Dimensiones del campo de juego
8 const int ANCHO = 60;
9 const int ALTO = 30;
10
11
12 // Coordenadas iniciales de la pelota y las paletas
13 int pelotax = ANCHO / 2;
14 int pelotay = ALTO / 2;
15 int jugador1x = 1;
16 int jugador1y = ALTO / 2 - 2;
17 int jugador2x = ANCHO - 2;
18 int jugador2y = ALTO / 2 - 2;
19 int puntos1 = 0;
20 int puntos2 = 0;
21 int rows, cols;
22 bool continuar = true;
23 bool seguir = true;
24 // Velocidad de la pelota
25 int movpelotax = -1;
26 int movpelotay = -1;
27
```

Figura 3: Programa de creación de una nueva ventana y imprimir una cadena

Después vamos a crear una función llamada campo que se encargará de pintar el campo.

```
C pelotita.c  F pelotita  C pong.c  X  C ventana.c
C pong.c > @ main()
28 // Función para dibujar el campo de juego
29 void campo(){
30     // Limpia la pantalla
31     clear();
32     // Dibuja los bordes
33     for (int i = 0; i < ANCHO; i++){
34         mvprintw(0, i, "-");
35         mvprintw(ALTO - 1, i, "-");
36     }
37     for (int i = 1; i < ALTO - 1; i++){
38         mvprintw(i, 0, "|");
39         mvprintw(i, ANCHO - 1, "|");
40     }
41     // Dibuja la pelota y las paletas
42     mvprintw(pelotay, pelotax, "0");
43     for (int i = 0; i < 4; i++){
44         mvprintw(jugador1y + i, jugador1x, "|");
45         mvprintw(jugador2y + i, jugador2x, "|");
46     }
47
48     mvprintw(ALTO + 5, 1, "%d", puntos1);
49     mvprintw(ALTO + 5, ANCHO - 1, "%d", puntos2);
50
51     // Refresca la pantalla
52     refresh();
53 }
54
```

Figura 4: Programa de creación de una nueva ventana y imprimir una cadena

Ahora comenzará el juego, usaremos todas las funciones proporcionadas por Incurses para iniciar como `initscr()` y crearemos la primera nueva ventana donde pondremos la información de los controles.

```
// NUEVA VENTANA
if (has_colors() == FALSE) {
    endwin();
    printf("El terminal no tiene soporte de color\n");
    exit(1);
}

start_color();
init_pair(1, COLOR_YELLOW, COLOR_GREEN);
init_pair(2, COLOR_BLACK, COLOR_WHITE);
init_pair(3, COLOR_WHITE, COLOR_BLUE);
clear();
refresh();
getmaxyx(stdscr, rows, cols);

noecho();
curs_set(0);
WINDOW * instrucciones = newwin(rows, cols, 0, 0);
wbgd(instrucciones, COLOR_PAIR(2));
mvprintw(instrucciones, 1, 1, "Realizado por Juan José Martínez Aguilera, github JuanJGG69");
mvprintw(instrucciones, 2, 1, "Controles jugador izquierda, arriba w y abajo s");
mvprintw(instrucciones, 3, 1, "Controles jugador derecha, arriba p y abajo l");
mvprintw(instrucciones, 4, 1, "El juego acaba cuando algun jugador pulse la tecla e o quien llegue a tres puntos antes");
wrefresh(instrucciones);
getch();
```

Figura 5: Código de la ventana de información con las instrucciones

```
Realizado por Juan José Martínez @-Agulla, github JuanjoGG69
Controles jugador izquierda, arriba w y abajo s
Controles jugador derecha, arriba p y abajo l
El juego acaba cuando algun jugador pulse la tecla e o quien llegue a tres puntos antes
```

Figura 6: Ventana con las instrucciones

Para mover los jugadores se hará mediante un switch pidiendole una tecla al usuario y si coincide con las teclas de movimiento se decrementará o aumentará el eje de la y.  
El movimiento de la pelota nos basaremos en el programa dado en el guión y importante poner un `timeout(0)` para que no se espere a que el usuario pulse una tecla. Utilizaremos una variable para la posición de pelota actual y otra para su movimiento que tomará los valores de 1 o -1 en función de si choca con la paredes de arriba o abajo o con los jugadores

```
pelotax += movpelotax;
pelotay += movpelotay;

if(pelotay <= 0 || pelotay >= ALTO - 1){
    movpelotay = -movpelotay;
}

if(pelotax <= 0 || pelotax >= ANCHO - 1){
    if(pelotax >= ANCHO - 1){
        puntos1++;
    }
    if(pelotax <= 0){
        puntos2++;
    }
    pelotax = ANCHO / 2;
    pelotay = ALTO / 2;
}

if (pelotax == jugador1x + 1 && (pelotay >= jugador1y && pelotay < jugador1y + 4)){
    movpelotax = -movpelotax;
}

if (pelotax == jugador2x - 1 && (pelotay >= jugador2y && pelotay < jugador2y + 4)){
    movpelotax = -movpelotax;
}
```

Figura 7: Código para controlar el choque de la pelota

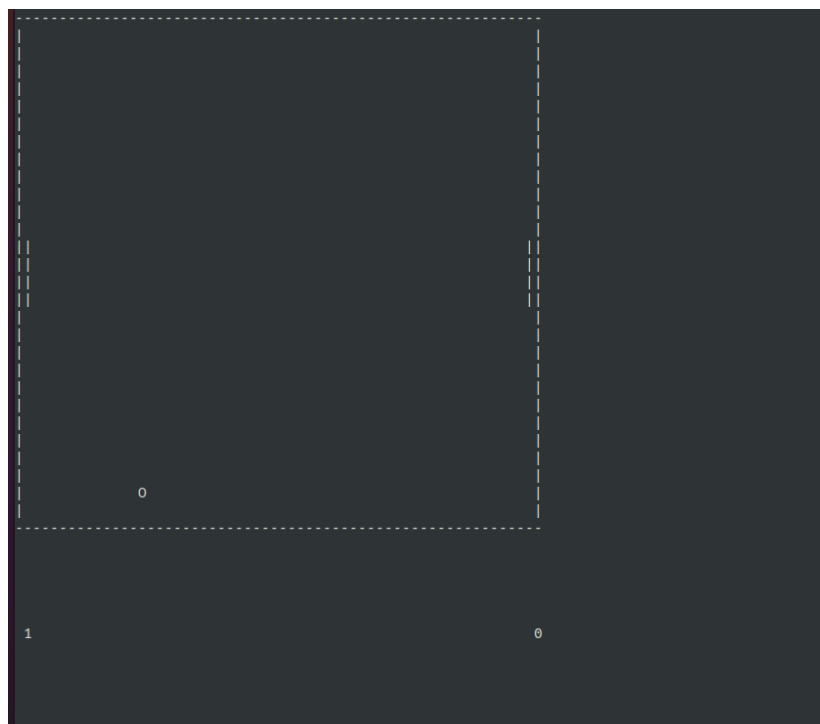


Figura 8: Juego en funcionamiento

Por último el código de la pantalla de final de juego que dice la puntuación de cada jugador y si pulsa la tecla c se reinicia la partida y si es cualquier otra se finaliza el juego.

```

if(puntos1 == 3 || puntos2 == 3){
    timeout(-1);
    WINDOW * final = newwin(rows, cols, 0, 0);
    wbkgd(final, COLOR_PAIR(3));
    mvwprintw(final, 1, 1, "GAME OVER");
    mvwprintw(final, 2, 1, "Jugador derecha:");
    mvwprintw(final, 2, 20, "%d", puntos1);
    mvwprintw(final, 3, 1, "Jugador izquierda:");
    mvwprintw(final, 3, 20, "%d", puntos2);
    mvwprintw(final, 4, 0, "Si desea iniciar una partida pulse la tecla c, si desea salir pulse cualquier otra tecla");
    wrefresh(final);
    char tecla = getch();
    if(tecla != 'c'){
        seguir = false;
    }
    else{
        timeout(0);
        puntos1 = 0;
        puntos2 = 0;
    }
}

```

Figura 9: Código de la segunda pantalla

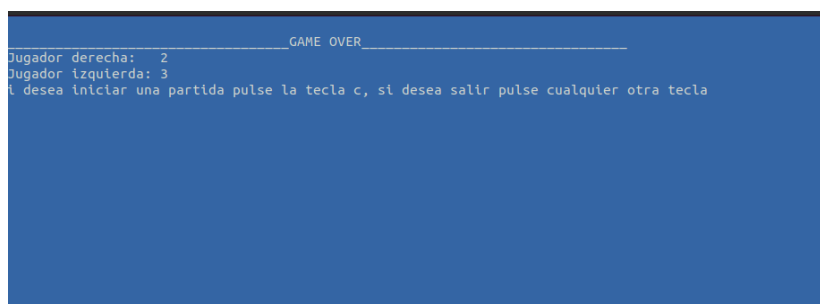


Figura 10: Segunda pantalla para finalizar o reiniciar el juego